Center for Cognitive Interaction Technology
Kognitronik und Sensorik
Prof. Dr.-Ing. U. Rückert

Deep Generative Models for Multi-Modal Perception under the Influence of Ambiguity

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEUR (Dr.-Ing.)

der Technischen Fakultät
der Universität Bielefeld

genehmigte Dissertation

von

Timo Korthals

Referent:	Prof. Dr.-Ing. Ulrich Rückert
1. Korreferent:	Prof. Dr. rer. nat. Helge Ritter
2. Korreferent:	Prof. Dr. rer. nat. Thomas Bräunl

Tag der mündlichen Prüfung: 9.11.2021

Bielefeld / 2021

Printed on acid-free paper according to ISO 9706.

Contents

1 Introduction

Living and learning beings are exposed to extensive information perceived through different senses (modalities) every day. However, information about complex perceivable phenomena only becomes reliably if multiple heterogeneous modalities are taken into account. This so-called multi-modal perception helps resolve perceptual ambiguities and lets us attempt to make more accurate predictions, judgments, and inferences. The multi-modal approach motivates machine learning solutions that combine complementary information sources to improve classification or regression outcomes. The related field of research is called multi-modal, multi-view, multi-sensory, or multi-source learning and has received growing interest over the past decades (see "multi-modal" in Fig. 1.1).

One major challenge faced when utilizing multi-modal learning is the handling of heterogeneity between the modalities. Heterogeneity is a property in which the nature of information sources, such as images or sounds, are very different from each other. It may cause unequal dimensions or structures in the data that is emitted by each modality. Even the information content can differ, even when different modalities are used to observe the same phenomena. Therefore, heterogeneity can impede naïve approaches to combinations of information.

In recent years deep learning, a machine learning technique that involves the use of deep neural networks, has become the mainstream approach tackling significant challenges like dealing with the complexity, dimensionality, and the unstructured nature of raw observations. However, while one key to understanding and explaining data in general is the representational learning of its latent distribution, deep learning techniques do commonly lack this demanding feature. On the contrary, distributions and distributional differences between modalities can be explicitly addressed using stochastic generative models. As a consequence, the very recent unification of deep learning and generative models into deep Bayesian generative models or variational auto encoders (VAEs) are of great interest in many scientific disciplines and their applications are becoming increasingly widespread (see "VAE" in Fig. 1.1).

Researchers recently approached the application of deep generative models to multi-modal data (see "multi-modal VAE" in Fig. 1.1) but neglected either the generative nature, ambiguities, or possible drop-out in multi-modal observations. Therefore, the author of this work focused on extending deep generative models that can deal with the ambiguities between different modalities while learning and representing the underlying probability distributions. Furthermore, there is an emerging field in machine and deep learning that involves dealing with missing modalities, but so far, no studies are known to have investigated the applicability and effectiveness of deep generative models in multi-modal settings. Also, the framework for training multi-modal deep generative models has not been sufficiently developed, which motivated the author to explore the combination of multi-modality, deep learning, and generative models in earnest.

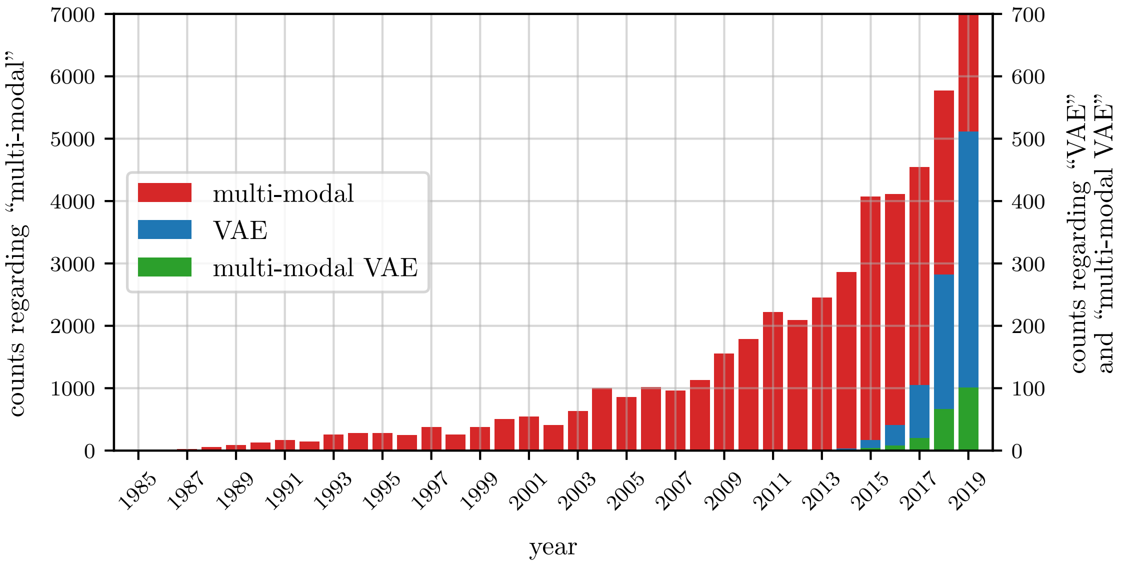


Figure 1.1: Interest in "multi-modal" (left ordinate) and "VAE"/"multi-modal VAE" (right ordinate) has been growing over the past decades. This can be seen, for example, in the increasing number of occurrences in those keywords in publications from the 40 most influential engineering and computer science conferences in robotics (see Table A.1) and AI (see Table A.2) according to gScholar.

Multi-modal perception also plays a significant role in autonomous systems that interact with the world (i.e., robots). Robots are commonly equipped with numerous task-specific sensors to perform exteroceptive and proprioceptive detections of external and internal states. To overcome the complexity involved in combining all these information sources, handcrafted architectures ranging from early to late fusion were proposed over the last decades. They commonly demand the making of naïve simplifications and assumptions about the information's nature, which often involves neglecting the sensors' potentials.

Here again, deep learning (DL) had its advent over the last years in substituting traditional fusion and control architectures, because DL allows learning directly from data with as few constraints as possible. The theoretical approach of this thesis on multi-modal perception also has a high potential for active sensing and intrinsic motivation due to the behavior of the learned distributions during fusion. Therefore, further experiments were carried out, revealing the potential of the proposed approach to reinforcement learning tasks.

2 Contributions and Outline

A multi-modal deep generative model (DGM) approach that combines the frameworks of data-driven deep learning and model-driven generative models is proposed in this work. The thesis is structured as follows (see Fig. 2.1):

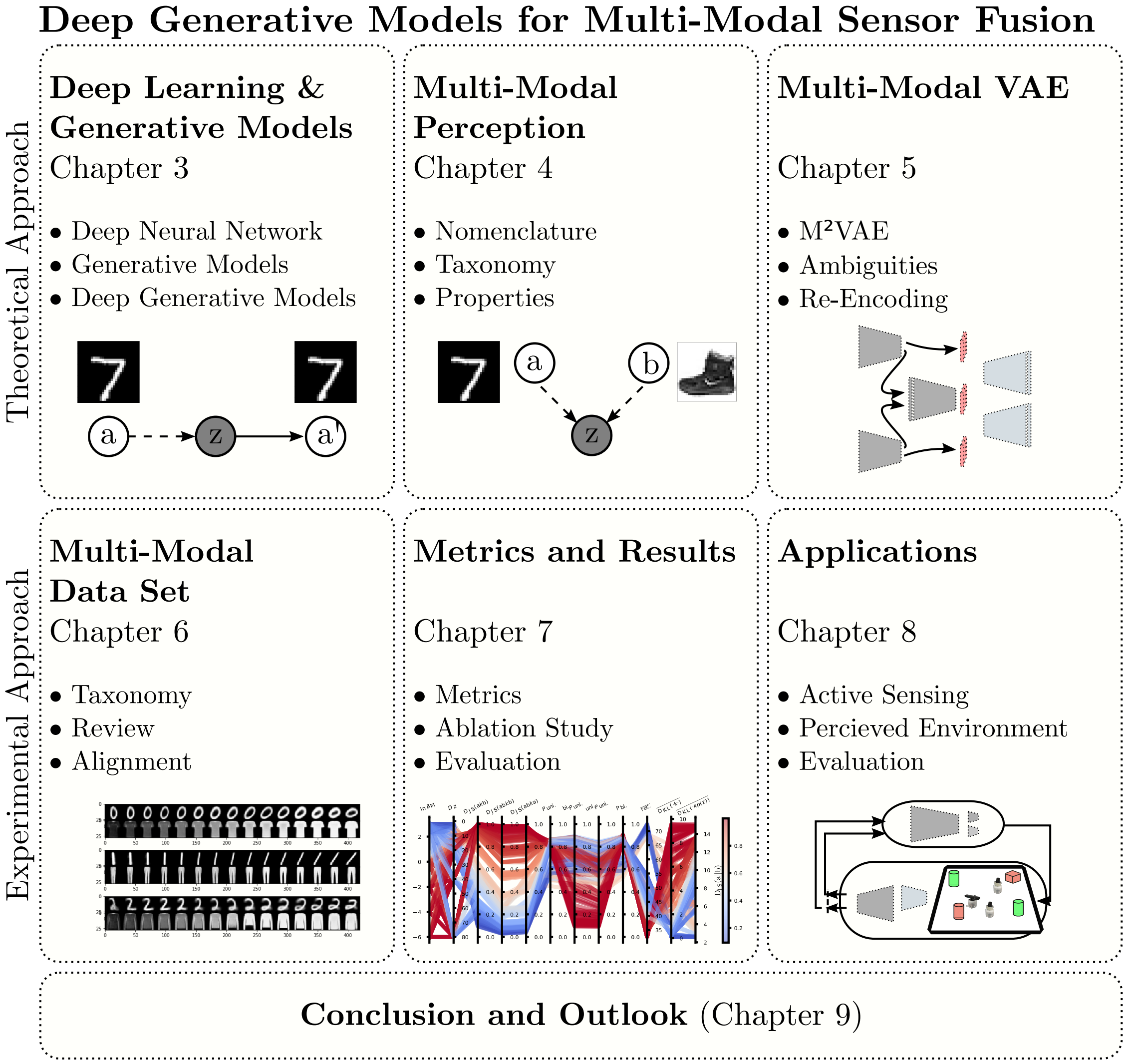


Figure 2.1: Outline of this thesis.

Figure 2.1 highlights the separation of this work into two branches: theoretical and applied. Chapter 3 through 5 comprise the theoretical work from the introduction of DGMs in Chapter 3 and multi-modal perception in Chapter 4 to the main contribution of the multi-modal DGM in Chapter 5. Chapter 6 through 8 comprise the applied work from the discussion and introduction of multi-modal data-sets and their properties in Chapter 6 and the evaluation of the proposed and competitive approaches in Chapter 7 to the discussion of applications in Chapter 8. Finally, the work is comprised and reflected on in Chapter 9. A closer look at the single chapters is carried out in the subsequent paragraphs while all the author's publications that led to the findings in this thesis, are comprised in Table 2.1.

In this thesis, a method for multi-modal learning using both deep learning and generative models is proposed. Therefore, Chapter 3 contains the results of a deep analysis of these two disciplines before establishing the link that forms DGMs. The contributions of Chapter 3 comprise a deep dive into DL and GM plus an overview of the many aspects of DGMs.

The connection of this work's DGM to each problem setting of multi-modal learning is proposed in this work: representation, translation, fusion, and co-learning. Therefore, the multi-modal research domain was investigated during the first studies presented in Chapter 4. This thesis is grounded by means of taxonomy and nomenclature, and an extensive literature summary was conducted on the investigated topics. The contributions of Chapter 4 comprise a multi-modal taxonomy tree, the definition of modality relations and correlations, and an exhaustive literature overview regarding multi-modal DGMs.

In the following research in Chapter 5, a multi-modal DGM is proposed that addresses two problem settings in multi-modal learning: representation and transformation under modality dropout. The training of various multi-modal VAEs approaches suffers from learning non-coherent representation between modalities. This results in inconsistent behaviors when facing irregularities, like modality dropout, during testing. The resulting contributions of Chapter 5 comprise a coherent and exhaustive derivation of the VAE's evidence lower bound (ELBO) in a multi-modal scenario, the extension to arbitrary large sets of modalities, an analysis of the latent space behaviors under ambiguities, and a deep learning architecture to train the network in a tractable fashion.

Many data set collections exist and are publicly available, but a categorization of multi-modal data set properties is still missing in the literature. Therefore, the necessity of defining and generating multi-modal data sets is revealed in Chapter 6. Furthermore, new and comprehensible multi-modal data sets are introduced, which are able to demonstrate the true capabilities of the proposed work. The contributions of Chapter 6 comprise a taxonomy of multi-modal data sets, an exhaustive data set overview plus new comprehensible data sets, and a generative approach to factor alignment.

This work focuses on multi-modal data sets under ambiguities and modality drop out, which demands the investigation of proper measures and metrics with which to quantify model performance. Various measures in the literature exist, but there is a lack of focus on shared representation from different modalities. Therefore, the most common measures are analyzed in Chapter 7 regarding their performance in quantifying coherent shared representation during modality dropout and ambiguous observations. Furthermore, an analysis of the proposed models' hyperparameters plus an ablation study was carried out. The contributions of Chapter 7 comprise an overview of suitable metrics for multi-modal learning under ambiguities and an exhaustive evaluation of the model's performance.

Finally, Chapter 8 focuses on drawing connections between multi-modal and ambiguity resolving (epistemic) sensing. It leads to a proposal about how a multi-modal VAE can be used as a solution in an active sensing scenario. The contribution of Chapter 8 leads to a perceived environment that can be applied to learn active sensing with deep reinforcement learning.

Table 2.1: Publications from the author that led to the findings in this thesis.

[Kor+15] Timo Korthals et al. "Evidence Grid Based Information Fusion for Semantic Classifiers in Dynamic Sensor Networks". In: *Machine Learning for Cyber Physical Systems* 1.1 (2015), p. 6

[Mey+15] Sebastian Meyer zu Borgsen et al. "ToBI-Team of Bielefeld The Human-Robot Interaction System for RoboCup@Home 2015". In: 2015

[Kor+16a] Timo Korthals et al. "Einsatz Event-Basierter Systemarchitektur für Erntemaschinen zur Elektronischen Umfelderkennung". In: *74. Tagung LAND.TECHNIK*. VDI e.V., 2016

[Kor+16b] Timo Korthals et al. "Evidenzkarten-basierte Sensorfusion zur Umfelderkennung und Interpretation in der Ernte". In: *Informatik in der Land-, Forst und Ernährungswirtschaft*. 2016, pp. 15–18

[Kor+16c] Timo Korthals et al. "Occupancy Grid Mapping with Highly Uncertain Range Sensors based on Inverse Particle Filters". In: *ICINCO 2016 - Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics* 2 (2016)

[Mey+16] Sebastian Meyer zu Borgsen et al. "ToBI-Team of Bielefeld The Human-Robot Interaction System for RoboCup@Home 2016". In: 2016

[Kra+16] Mikkel Kragh et al. "Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture". In: *International Conference on Agricultural Engineering*. Aarhus, 2016. URL: `http://conferences.au.dk/uploads/tx_powermail/2016cigr_-_multi-modal_obstacle_detection_and_evaluation_of_evidence_grid_mapping_in_agriculture.pdf`

[Her+16] Stefan Herbrechtsmeier et al. "AMiRo: A Modular & Customizable Open-Source Mini Robot Platform". In: *ICSTCC*. 2016. ISBN: 9781509027200. DOI: `10.1109/ICSTCC.2016.7790746`

[Kor+17a] Timo Korthals et al. “Semantical Occupancy Grid Mapping Framework”. In: *2017 European Conference on Mobile Robots, ECMR 2017*. IEEE, 2017. ISBN: 9781538610961. DOI: `10.1109/ECMR.2017.8098673`

[Kor+17b] Timo Korthals et al. “Towards Inverse Sensor Mapping in Agriculture”. In: *IROS 2017 Workshop on Agricultural Robotics: learning from Industry 4.0 and moving into the future*. Vancouver, 2017

[Bor+17] Sebastian Meyer zu Borgsen et al. “ToBI – Team of Bielefeld: Enhancing Robot Behaviors and the Role of Multi-robotics in RoboCup@Home”. In: *RoboCup 2016: Robot World Cup XX*. ed. by Sven Behnke et al. Cham: Springer International Publishing, 2017, pp. 577–588. ISBN: 978-3-319-68792-6

[Kor+18a] Timo Korthals et al. “Coordinated Heterogeneous Distributed Perception based on Latent Space Representation”. In: *IROS 2018 Second Workshop on Multi-robot Perception-Driven Control and Planning*. 2018. arXiv: `arXiv:1809.04558v1`. URL: `https://arxiv.org/abs/1809.04558`

[Kor+18b] Timo Korthals et al. “Obstacle Detection and Mapping in Agriculture for Process Evaluation”. In: *Frontiers in Robotics and AI Robotic Control Systems* 1.1 (2018). URL: `https://www.frontiersin.org/research-topics/5597/multi-modal-sensor-fusion`

[Kor+18c] Timo Korthals et al. “Path Evaluation via HMM on Semantical Occupancy Grid Maps”. In: *ArXiv e-prints* (2018). arXiv: `1805.02944 [cs.RO]`

[Kor19] Timo Korthals. *M^2VAE - Derivation of a Multi-Modal Variational Autoencoder Objective from the Marginal Joint Log-Likelihood*. 2019. arXiv: `arXiv:1903.07303`. URL: `http://arxiv.org/abs/1903.07303`

[Kor+19a] Timo Korthals et al. *A Perceived Environment Design using a Multi-Modal Variational Autoencoder for learning Active-Sensing*. 2019. arXiv: `1911.00584 [cs.RO]`. URL: `https://sites.google.com/site/dpgmcar2019/home`

[Kor+19b] Timo Korthals et al. “Fiducial Marker based Extrinsic Camera Calibration for a Robot Benchmarking Platform”. In: *European Conference on Mobile Robots, ECMR 2019, Prague, CZ, September 4-6, 2019*. 2019, pp. 1–6

[Kor+19c] Timo Korthals et al. “Jointly Trained Variational Autoencoder for Multi-Modal Sensor Fusion”. In: *22st International Conference on Information Fusion, FUSION 2019, Ottawa, CA, July 2-5, 2019*. 2019, pp. 1–8

[Kor+19d] Timo Korthals et al. “Multi-Modal Generative Models for Learning Epistemic Active Sensing”. In: *2019 IEEE International Conference on Robotics and Automation, ICRA 2019, Montreal, CA, May 20-25, 2019*. Montreal, Canada, 2019

[Kor+19e] Timo Korthals et al. “Multisensory Assisted In-hand Manipulation of Objects with a Dexterous Hand”. In: *2019 IEEE International Conference on Robotics and Automation Workshop on Integrating Vision and Touch for Multimodal and Cross-modal Perception, ViTac 2019, Montreal, CA, May 20-25, 2019*. 2019, pp. 1–2. URL: `http://wordpress.csc.liv.ac.uk/smartlab/icra-2019-vitac-workshop/`

3 Deep Generative Models

In this thesis, a method for multi-modal learning using both deep learning and generative models is proposed. In sections 3.1 and 3.2, the prerequisites for deep neural networks (DNNs) and generative models (GMs) are explained, respectively. Furthermore, two approaches to connect DNNs and GMs to become a DGM are described in Section 3.3. Finally, the VAE, as one of the two approaches, is explained.

3.1 Deep Neural Network

A DNN, which is also known as a feedforward neural network (FFNN) or multilayer perceptron (MLP), is a function approximator composed of artificial neural networks (ANNs) connected in a hierarchical manner. When the number of layers in a FFNN is more than two (i.e., the input and output layer), it is called a deep neural network. Deep learning (DL) is the generic term for the approaches that involve using the DNNs as a learning machine (e.g., a discriminator) for machine learning (ML). The following sections contain a brief explanation of the structure and learning of DNNs.

3.1.1 The Structure of Deep Neural Networks

The single artificial neuron is the basic building block that makes up a DNN. McCulloch et al. [McC+43] described it as the formal modeling of the neurons in the brain's nervous system. It is a function f that takes the input vector $\mathbf{x} = [x_1, \dots, x_I]^\mathsf{T} \in \mathbb{R}^I$ and outputs some h value:

$$h = f(\mathbf{x}) = g\left(\sum_{i=1}^{I} w_i x_i + b\right) = g\Big(\mathbf{w}^\mathsf{T}\mathbf{x} + b\Big) \tag{3.1}$$

The neuron's parameters consist of the weight vector $\mathbf{w} = [w_1, \dots, w_I]^\mathsf{T} \in \mathbb{R}^I$ and bias parameter $b \in \mathbb{R}$, which scale and offset the input (i.e., a linear function). g is called the activation function, that can be a linear (lin), hyperbolic tangent (tanh), sigmoid (sig), or rectified linear unit (ReLU) function, for example[1]. Equation (3.1)

[1] See Ramachandran et al. [Ram+17b] for an evaluation on activation functions for DNN

can be generalized to a layer of multiple neurons that share the input vector $\mathbf{x}$ and output the vector $\mathbf{h} = [h_1, \ldots, h_N]^\mathsf{T}$:

$$\mathbf{h} = \left[g\Big(\sum_i^I w_{1,i} x_i + b_i \Big), \ldots, g\Big(\sum_i^I w_{N,i} x_i + b_i \Big) \right]^\mathsf{T} := g\Big(\mathbf{W}^\mathsf{T} \mathbf{x} + \mathbf{b} \Big) \tag{3.2}$$

In the case of FFNN, Eq. (3.2) is called the layer of the network. FFNNs are structured in such a way that the output of one layer is the input of the next layer. For example, the output $\mathbf{h}_l = f(\mathbf{x}; \theta_l)$ of the l^{th} layer is given by its parameters, $\theta_l := (\mathbf{W}_l, \mathbf{b}_l)$, its activation function, g_l, and the output, $\mathbf{h}_{l-1}$, of the $l^{\text{th}} - 1$ layer, which serves as input $\mathbf{x}$. Because $\mathbf{h}_{l-1}$ depends on the parameters of the $l^{\text{th}} - 1$ layer, the output of the l^{th} layer eventually depends on all the parameters of the layers before it plus the input of the first layer. That is, when the network consists of L layers, the output of the L^{th} layer of the last layer is $\mathbf{h}_L = f(f(\ldots f(\ldots f(f(\mathbf{x}; \theta_1); \theta_2) \ldots; \theta_l) \ldots; \theta_{L-1}); \theta_L)$. In this case, $f(\cdot; \theta_1)$ is called the input layer and $f(\cdot; \theta_L)$ is called the output layer. The other intermediate layers are called hidden layers. A FFNN with at least one hidden layer is called a DNN. For the sake of brevity, each layer's parameters are collectively denoted by θ and the output of the last layer L can be written simply as $\mathbf{h} = f(\mathbf{x}; \theta)$. The network's architecture properties like the layer's activation function, number of neurons (i.e., width), or the number of layers (i.e., depth) are not part of the functions argument because they count as hyperparameters (HPs) that are chosen by design. In general, the same activation function is used in the layers before the output layer while the activation function of the L^{th} layer is selected according to the form of the output. Width and depth are chosen according to the target function's complexity, which needs to be approximated. Figure 3.1 summarizes the above explanation.

Depending on the feature space and nature of input $\mathbf{x}$, a deep neural network can have various architectural designs to improve its performance. In the case of image data, the convolutional neural network (CNN) [Lec+98] is used under the assumption that the modality has a lattice-like form with a strong correlation between adjacent pixels and is translation-invariant. In the case of serial data, a recurrent neural network (RNN) [Ger+99] is used under the assumption that there are one-dimensional variable vectors and strong dependencies in the serial direction.

3.1.2 Learning Objectives

Consider the joint distribution $p(\mathbf{x}, \mathbf{y})$ of a pair of inputs $\mathbf{x}$ and target outputs $\mathbf{y}$, where $p^*(\mathbf{x}, \mathbf{y})$ is the true data distribution. If a neural network (NN) is considered to be a trainable function in ML, the purpose of learning is to adjust the θ parameter so that the output distribution, $p(\mathbf{x}; \theta)$, for any $\mathbf{x}$ input is close to the corresponding

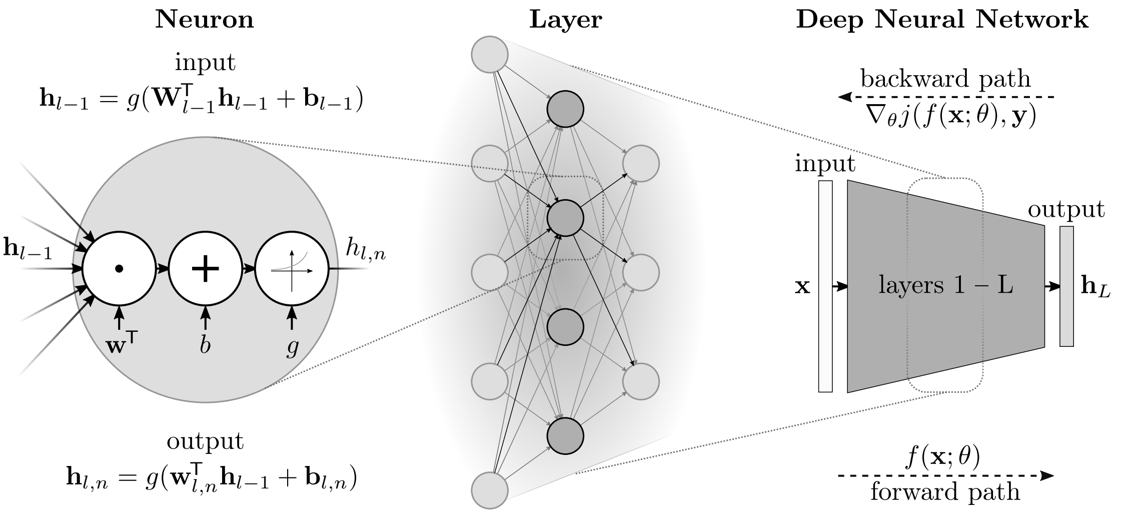

Neuron
input
$\mathbf{h}_{l-1} = g(\mathbf{W}_{l-1}^{\mathsf{T}}\mathbf{h}_{l-1} + \mathbf{b}_{l-1})$
$\mathbf{h}_{l-1}$
$h_{l,n}$
$\mathbf{w}^{\mathsf{T}}$
b
g
output
$\mathbf{h}_{l,n} = g(\mathbf{w}_{l,n}^{\mathsf{T}}\mathbf{h}_{l-1} + \mathbf{b}_{l,n})$
Layer
Deep Neural Network
backward path
$\nabla_\theta j(f(\mathbf{x};\theta),\mathbf{y})$
input
output
$\mathbf{x}$
layers 1 – L
$\mathbf{h}_L$
$f(\mathbf{x};\theta)$
forward path

Figure 3.1: A summary of the structure and notation of a deep neural network (DNN) as a feedforward neural network (FFNN).

target $\mathbf{y}$ under the true data distribution $p^*(\mathbf{x},\mathbf{y})$. This is accomplished by minimizing the error between the output $f(\mathbf{x};\theta)$ and the target $\mathbf{y}$ regarding the input $\mathbf{x}$ using a comparative function $j(f(\mathbf{x};\theta),\mathbf{y})$, where j is a function of θ. This function j is called an error, loss, or objective function in the field of ML. By minimizing the loss, given the input $\mathbf{x}$ and target $\mathbf{y}$ data, the parameter θ can be adjusted such that $f(\cdot;\theta^*)$ eventually approximates the true data distribution $p^*(\mathbf{x},\mathbf{y})$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathrm{E}_{p^*(\mathbf{x},\mathbf{y})}\, j(f(\mathbf{x};\theta),\mathbf{y}) \tag{3.3}$$

Therefore, the learning of the NN for the true data distribution $p^*(\mathbf{x},\mathbf{y})$ is an optimization problem in Eq. (3.3).

In fact, the true data distribution $p^*(\mathbf{x},\mathbf{y})$ is unknown, so the expected value, E, of Eq. (3.3) cannot be directly obtained. The expected value can only be approximated by obtaining the set of observations (i.e., samples) $\mathcal{O} = \{(\mathbf{x}_1,\mathbf{y}_1),\dots,(\mathbf{x}_O,\mathbf{y}_O)\}$ from the true data distribution. Therefore, the optimization problem changes as follows:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, J(\theta|\mathcal{O}) = \sum_{o=1}^{O} j(f(\mathbf{x}_o;\theta),\mathbf{y}_o) \tag{3.4}$$

$J(\theta|\mathcal{O})$ is called the error of observation set $\mathcal{O}$. Because the original purpose of learning is to optimize the expected value in Eq. (3.3) (i.e., the true data distribution) the optimization of Eq. (3.4) may result in learning some θ^* that only fits the training set. This phenomenon is called overtraining and prevents the NN from being generalized to unobserved data that results from $p^*(\mathbf{x}, \mathbf{y})$ but was not part of the training set. To prevent this behavior, $\mathcal{O}$ is separated into an explicit training set $\mathcal{O}_{\text{train}}$ plus a testing $\mathcal{O}_{\text{test}}$ and validation $\mathcal{O}_{\text{val}}$ set. Usually, after the training (i.e., the optimization of Eq. (3.4)) on $\mathcal{O}_{\text{train}}$ is over, the error $J(\theta|\mathcal{O}_{\text{test}})$ is evaluated to check whether the NN has overlearned the training set. Therefore, $J(\theta|\mathcal{O}_{\text{train}})$, $J(\theta|\mathcal{O}_{\text{test}})$, and $J(\theta|\mathcal{O}_{\text{val}})$ are called train, test, and validation errors, respectively.

One goal of ML is to optimize Eq. (3.4) on a training set such that the algorithm generalizes well to unknown and unseen data. Thus, the test error is also known as the generalization error, which needs to be minimized as well. Because the test set is not available at the time of training and exclusively not for optimization, the validation set is used to evaluate the generalization error at the time of training.

The above framework is called supervised learning because the inputs and corresponding targets are given as datasets. A framework for learning just from a set of inputs is called unsupervised learning. Finally, a framework in which a set of inputs plus targets and a set of inputs without targets are given is called semi-supervised learning.

3.1.3 Training Deep Neural Networks

In the case of DNNs, it is intractable to solve Eq. (3.4) analytically because the training error is usually not a convex function concerning the parameters. However, the gradient of Eq. (3.4) for a given parameter and observation set can be calculated using basic calculus. Therefore, identifying the gradient of the error function and updating the parameters in the direction that minimizes the error can be considered:

$$\theta_{e+1} \leftarrow \theta_e - \gamma \left. \frac{\partial J(\theta|\mathcal{O})}{\partial \theta} \right|_{\theta_e} \tag{3.5}$$

Equation (3.5) is called the gradient decent method. In it, γ is the learning rate that determines the change of the parameter in one epoch e.

However, the gradients in the hidden layer cannot be computed. The error can only be computed for the output layer (i.e., the credit allocation problem). In a FFNN, the gradient in the hidden layer is calculated using the backpropagation method, which involves propagating the error calculated in the output layer toward the input layer using the differential method of the composite function.

While Eq. (3.5) updates the parameters concerning the whole training set, it is common in DL to perform gradient descent on only a small and randomly selected batch (i.e., mini-batch) to stabilize the training process. This method is called stochastic gradient decent (SGD). In addition to the above SGD, various optimization algorithms have been proposed, such as the momentum method with an inertia term [Rum+86], Adadelta [Zei12], RMSprop [Tie+12], and Adam [Kin+14a].

3.1.4 Further Remarks on Training Deep Neural Networks

It is known that DNN training does not scale well with an increasing number of layers. Various methods have been proposed to solve this problem.

The initialization of the weights of a neural network has a great influence on its learning progress and the final accuracy (see Locatello et al. [Loc+18]). Therefore, the initialization trick called Xavier [Glo+10] or He [He+15] is used. In this thesis, Xavier was uses for the initialization of DNNs unless otherwise stated. To avoid bias in the treatment of each element of the data in NNs, the data is often standardized beforehand (e.g., min-max normalization or zero-mean/standard-variance).

As the learning of the NN progresses, the distribution of the output of each layer shifts, and the deeper the layer is, the slower the convergence of its parameters becomes. To solve this problem, a method called batch normalization [Iof+15] has been proposed to standardize the outputs of each layer during training.

Several methods have also been proposed to reduce the generalization error in the test set. For example, dropout [Sri+14] is a method of regularization that involves stochastically removing the output of units during training.

3.2 Generative Models

This section contains an overview of generative models, graphical models, and the training of generative models with latent variables.

3.2.1 Generative Model Framework and Learning

The mapping from the data, $\mathbf{x}$, to the targets, $\mathbf{y}$, using DNNs was discussed in Section 3.1. Suppose that the true data distribution in data $\mathbf{x}$ is represented by $p^*(\mathbf{x})$. Because the true data distribution cannot be directly obtained, instead, the stochastic model $p_\theta(\mathbf{x})$ (also referred to as $p(\mathbf{x}|\theta)$) is considered because it can be used to approximate the true data distribution by learning the stochastic parameter θ of the model. Because $p_\theta(\mathbf{x})$ is a stochastic model of the generation process of data $\mathbf{x}$, it is called a generative model. On the contrary, the approach to learning the mapping from input to output as shown in Section 3.1 is called a discriminative model.

One of the advantages of training a generative model is that new data can be generated through sampling. This is because the learned generative model approximates the entire structure of the data distribution. In addition, the use of a generative model enables us to do things, such as make density estimations and fill in missing values, that are impossible with the discriminative model, which only learns input/output mappings. However, the ability of generative models to adequately approximate true data distributions depends on what optimization and modeling methods are used, as well as how the distance between the generative model against the data distribution is measured.

To train a generative model, it is necessary to measure how close it is to the true data distribution (i.e., how well it can be approximated). In general, the closeness of the generative model $p_\theta(\mathbf{x})$ to the data distribution $p^*(\mathbf{x})$ is calculated using the Kullback–Leibler divergence (KLD) D_{KL}.[2] Therefore, the training of the generative model is an optimization problem that involves finding a parameter θ such that the KLD (i.e., Eq. (C.20)) is minimized. This can be rewritten as follows:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathrm{D}_{\mathrm{KL}}(p^*(\mathbf{x}) \| p_\theta(\mathbf{x})) = \underset{\theta}{\operatorname{argmax}}\, \mathrm{E}_{p^*(\mathbf{x})}\, p_\theta(\mathbf{x}) \tag{3.6}$$

Equation (3.6) and (3.3) calculate expectations about the true data distributions, which cannot be directly calculated. To this end, the expected value in Eq. (3.6) can be approximated from observations $\mathcal{O} = \{\mathbf{x}_1, \ldots, \mathbf{x}_O\}$ of the true distribution:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \frac{1}{O} \sum_{o=1}^{O} \log p_\theta(\mathbf{x}_o) = \underset{\theta}{\operatorname{argmax}}\, \log \prod_{o=1}^{O} p_\theta(\mathbf{x}_o) = \underset{\theta}{\operatorname{argmax}}\, \log p_\theta(\mathcal{O}). \tag{3.7}$$

$\log p_\theta(\mathcal{O}) =: \mathrm{L}(\mathcal{O})$ is the likelihood function, and its parameter θ is omitted for brevity. In this way, the method for estimating the value of the parameter that maximizes the likelihood function is called maximum likelihood estimation.

Furthermore, it is assumed, that the parameter θ of the generative model has a distribution as well. Therefore, θ needs to be considered as a random variable, which is generated as $\hat{\theta} \sim p(\theta|\mu)$ using some probability distribution.[3] μ is a hyper parameter that controls the probability distribution of the parameter of the generative model.

The distribution $p(\theta|\mathcal{O}, \mu)$ of the parameter θ under a given observations $\mathcal{O}$ is given by Bayes' theorem:

$$p(\theta|\mathcal{O}, \mu) = \frac{p(\mathcal{O}|\theta)p(\theta|\mu)}{p(\mathcal{O})} \propto p(\mathcal{O}|\theta)p(\theta|\mu) \tag{3.8}$$

Equation (3.8) shows that the distribution of the parameter $p(\theta|\mu)$ changes to $p(\theta|\mathcal{O}, \mu)$ by making the observations $\mathcal{O}$. Thus, $p(\theta|\mu)$ is called the prior distribution (i.e., what is known about θ before making any observation) and $p(\theta|\mathcal{O}, \mu)$ is called the posteriori distribution (i.e., the updated assumption about θ after making an observation). The denominator in Eq. (3.8) is the data distribution, which is independent of the model and its parameters. It only normalizes the nominator to let

[2] see section C.2.3 for notes on divergence properties.

[3] $\sim$ denotes the sampling process which generates a realization $\hat{\mathbf{x}}$ from the distribution $p(\mathbf{x})$ of the random variable $\mathbf{X}$

Eq. (3.8) become a true probability. However, it remains constant and, therefore, negligible for any optimization approach.

By pursuing the goal of finding a model that can re-produce the observations, the parameters θ needs to be optimized such that Eq. (3.8) is maximized. The estimation θ^* that maximizes the posterior distribution is called maximum a posteriori probability (MAP) estimation. By maximizing the logarithmic posterior distribution, which shares the same optima as the posterior distribution because of the logarithm's monotonic behavior, the following optimization approach can be obtained:

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmax}}\, p(\theta|\mathcal{O}, \mu) = \underset{\theta}{\operatorname{argmax}} \log p(\theta|\mathcal{O}, \mu) && (3.9) \\ &= \underset{\theta}{\operatorname{argmax}} \log(p(\mathcal{O}|\theta)p(\theta|\mu)) = \underset{\theta}{\operatorname{argmax}}(\mathrm{L}(\mathcal{O}) + \log p(\theta|\mu)) && (3.10) \end{aligned}$$

Compared to the optimization approach in Eq. (3.7), Eq. (3.10) results in the addition of the term $\log p(\theta|\mu)$. This term bounds θ to be in a spectrum of possible solutions instead of being a single and discrete value. This can be regarded as a regularization term to prevent the overlearning of the parameter θ.

3.2.2 Graphical Model

In generative models, several random variables and probability distributions are used to describe the process of data generation. The generative model studied in Section 3.2.1 uses two random variables, $\mathbf{x}$ and θ. A model that is able to describe the relationship between these random variables is called a graphical model. There are two types used in graphical models: directed non-cyclic graphs and undirected graphs. The former is called a Bayesian network while the latter is called a Markovian probability field or Markov network. In this paper only Bayesian networks, which are directed models, are used when referring to graphical models. In the field of statistics, graphical models are used to analyze the interrelationships between multivariate data, and such statistical methods are called graphical modeling [Kol+09]. In ML, graphical models are simply used as a notion or language to describe the relationships between the variables modeled by the generative models in a graph structure (see Fig. 3.2 for an example).

3.2.3 Training of Generative Models with Latent Variables

When a generative model of observable data is modeled, it is assumed that some variables are hidden in the data in addition to the random variables and parameters of the input data. Such variable is called a latent variable. On the contrary, a

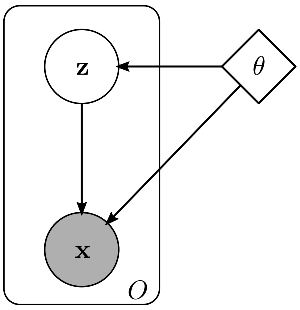


Figure 3.2: A generative model in plate notation containing one latent variable. The plate denotes that O independent latent samples $\mathbf{z}$ are drawn, which generates O observations of $\mathbf{x}$. Following Bishop [Bis07a, Chapter 10], the parameter θ is actually absorbed into $\mathbf{z}$. However, to satisfy the thesis' notation, it is explicitly stated in this figure.

variable whose value is directly observable, such as perceivable data, is called an observable variable.

Consider the learning of a generative model including latent variables $\mathbf{z}$. Let

$$\hat{\mathbf{z}} \sim p_\theta(\mathbf{z}) \text{ and } \hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z}) \tag{3.11}$$

be the generation processes of each variable. Figure 3.2 shows a graphical model of this generation process. In Fig. 3.2, the white circles represent latent variables and the shaded circles represent the observed variables. The joint distribution for all variables in the depicted generative model becomes

$$p_\theta(\mathbf{x}, \mathbf{z}) \overset{\text{(C.1)}}{=} p_\theta(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{x}). \tag{3.12}$$

As in Eq. (3.7), the likelihood of the model for the observed variables needs to be maximized. To do so, the marginalized likelihood $p_\theta(\mathbf{x})$ for the latent variables is obtained as follows:

$$p_\theta(\mathbf{x}) = \int_{\mathcal{Z}} p_\theta(\mathbf{x}, \mathbf{z}) = \int_{\mathcal{Z}} p_\theta(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{x}) = p_\theta(\mathbf{x}) \underbrace{\int_{\mathcal{Z}} p_\theta(\mathbf{z}|\mathbf{x})}_{1}. \tag{3.13}$$

Such an operation of elimination for a specific variable through integration is called marginalization, and $p_\theta(\mathbf{x})$ is called marginal distribution. Furthermore, $p_\theta(\mathbf{x})$ is also called evidence or, in physics, a partition function.

In general, the computation of this marginalization is difficult, because the integration over the whole latent space is intractable. For this reason, an approximating function q is introduced:

$$
\begin{aligned}
\mathrm{L}(\mathbf{x}) = \log p_\theta(\mathbf{x}) &= \int_{\mathcal{Z}} q(\mathbf{z}) \log p_\theta(\mathbf{x}) && \text{(C.6) w/o cond.} \quad (3.14)\\
&= \int_{\mathcal{Z}} q(\mathbf{z}) \log\left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right) && \text{Eq. (C.1)} \quad (3.15)\\
&= \int_{\mathcal{Z}} q(\mathbf{z}) \log\left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\frac{q(\mathbf{z})}{q(\mathbf{z})}\right) && \text{mul. by } 1{=}q/q \quad (3.16)\\
&= \int_{\mathcal{Z}} q(\mathbf{z}) \log\left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{q(\mathbf{z})}\frac{q(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right) && \text{reordered} \quad (3.17)\\
&= \underbrace{\int_{\mathcal{Z}} q(\mathbf{z}) \log\left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{q(\mathbf{z})}\right)}_{\mathcal{L}} + \underbrace{\int_{\mathcal{Z}} q(\mathbf{z}) \log\left(\frac{q(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right)}_{\mathrm{D_{KL}}} && \text{Eq. (C.3)} \quad (3.18)\\
&= \mathcal{L}(\mathbf{x}; q, \theta) + \mathrm{D_{KL}}(q(\mathbf{z}) \| p_\theta(\mathbf{z}|\mathbf{x})) && \text{(C.5) + (C.20)} \quad (3.19)\\
&\geq \mathcal{L}(\mathbf{x}; q, \theta) && \mathrm{D_{KL}} \geq 0 \quad (3.20)
\end{aligned}
$$

As can be seen from Eq. (3.19), the likelihood now possesses parameters θ and the function q that are evaluated as follows: $\mathrm{D_{KL}}$ is the Kullback–Leibler divergence, with $\mathrm{D_{KL}} \geq 0$, which is used to evaluate how well $q(\mathbf{z})$ can approximate $p(\mathbf{z}|\mathbf{x})$. The function $\mathcal{L}$ evaluates the evidence's lower bound (rhs. of Eq. (3.20)) and is, therefore, called the evidence lower bound (ELBO) of the marginal log-likelihood L_θ. $\mathrm{D_{KL}}$ becomes 0, if and only if the two distributions, q and p, are identical ($q \equiv p \Leftrightarrow \mathrm{D_{KL}} = 0$). Therefore, $\mathcal{L} \equiv L$ implicitly means that q perfectly approximates p. This is because $\mathcal{L}$ and $\mathrm{D_{KL}}$ are in equilibrium such that minimizing $\mathrm{D_{KL}}$ is equivalent to the maximization of $\mathcal{L}$ ($\min \mathrm{D_{KL}} \Leftrightarrow \max \mathcal{L}$). Thus, the marginal likelihood can be maximized by consecutively repeating the maximization of the lower bound for θ and the minimization of KLD for q.

ELBO is also called the negative variational free energy [Fri10] or variational lower bound of the marginal likelihood [Bis07a, Figure 9.11]. If $p_\theta(\mathbf{z}|\mathbf{x})$ can be obtained analytically, then Eq. (3.19) can be optimized using the expectation–maximization (EM) algorithm. On the contrary, if $p_\theta(\mathbf{z}|\mathbf{x})$ cannot be obtained analytically, then optimization is performed on the distribution family of $q(\mathbf{z})$ using mean-field approximation (see [Bis07a, Chapter 10.1.1] for further details about the mean-field technique). This method is called variational inference (VI) [Bis07a, Chapter 10.1]. The term "variational" originates from "variational methods" or "calculus of variations" [Bis07a, Chapter 10]. In comparison, standard calculus is concerned with functions that take the value of a variable as the input and return a value of the function as the output. Variational calculus, on the contrary, is concerned with functions, that take a function as the input and return a value of the functional as the output.[4]

3.3 Linking Deep Neural Networks and Generative Models

Generative models are modeled by probability distributions and, therefore, it is infeasible to take data with large dimension and complex structure directly as input. Recently, research into how to combine deep learning and generative models has been performed.

One approach is to perform feature extraction from the raw data using deep neural networks and train a generative model with a simple probability distribution as the input. This approach is called the feature extraction plus generative model approach to deep learning. Neural networks can reveal apt representations of inputs in hidden layers. The criteria for "good representation" were discussed by Bengio et al. [Ben+12], Goodfellow et al. [Goo+16, Chapter 13], and Van Der Maaten et al. [Van+09] and include information content, independence, explicitness, sparseness, invariance, robustness, and smoothness.

The second approach is to parameterize the probability distribution of the GM itself with the DNN. This is the approach with the DGM that can generate high dimensional and complex structured data because the probability distribution is directly parameterized by a DNN.

3.3.1 Deep Generative Model

The deep generative model is a method that combines DL and GMs. A DNN is used to directly define the probability distribution of a GM.

Deep Boltzmann machine [Sal+09] has been proposed as a deep layer generation model for use on undirected graphs. However, one issue with deep Boltzmann machine (DBM) is that they are infeasible to train on high-dimensional data, such as natural images, because the training rule of a DBM is based on the Markov chain Monte Carlo (MCMC) method.

In recent years, DGMs have been proposed for use with complex high-dimensional data. The two dominating methods are the VAE by Kingma et al. [Kin+13] and Rezende et al. [Rez+14], and the generative adversarial network (GAN) by Goodfellow et al. [Goo+14]. The main difference between VAE and GAN is that in VAE,

[4] An example would be the entropy function as defined in C.2.1, which takes a probability distribution $p(x)$ as the input and returns a quantity.

the distribution of the generative models is explicitly assumed, whereas in GAN, the shape of the distribution is implicitly learned. The author of this thesis only investigates VAEs as DGMs, which are carried out in the following sections.

3.3.2 Variational Autoencoder (VAE)

As shown in Fig. 3.2, a generative model with one latent variable and one observed variable is considered. In contrast to Section 3.2.3, the approach of the VAE is to find a stochastic map from the observation $\mathbf{x}$ to the latent representation $\mathbf{z}$ in an unsupervised fashion.[5] Therefore, $q(\mathbf{z})$ becomes $q_\phi(\mathbf{z}|\mathbf{x})$ using the model parameters ϕ. The method of approximating $q(\mathbf{z})$ by mapping from $\mathbf{x}$ to $\mathbf{z}$ is called learned variational inference because $q_\phi(\mathbf{z}|\mathbf{x})$ can be regarded as approximating the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$. The log-likelihood from Section 3.2.3 can then be rewritten as follows:

$$\begin{aligned}
\mathrm{L}(\mathbf{x}) = \log p_\theta(\mathbf{x}) &= \int_{\mathcal{Z}} q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) = \dots && (3.21)\\
&= \mathcal{L}(\mathbf{x};\phi,\theta) + \mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \geq \mathcal{L}(\mathbf{x};\phi,\theta) \qquad \text{sec. 3.2.3} && (3.22)
\end{aligned}$$

As explained in Section 3.2.3, the direct maximization of the marginal likelihood is not possible because the true posterior $p(\mathbf{z}|\mathbf{x})$ is unknown in general and, therefore, the KLD cannot be calculated. Fortunately, the maximization of the ELBO is equivalent to the minimization of the KLD in Eq. (3.22), which coincidentally depend on the same parameters. This ELBO can be further transformed into the following formula:

$$\begin{aligned}
\mathcal{L}(\mathbf{x};\phi,\theta) &= \int_{\mathcal{Z}} q_\phi(\mathbf{z}|\mathbf{x}) \log\left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right) && (3.23)\\
&= \int_{\mathcal{Z}} q_\phi(\mathbf{z}|\mathbf{x}) \log\left(\frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right) \qquad \text{Eq. (C.1)} && (3.24)\\
&= \int_{\mathcal{Z}} q_\phi(\mathbf{z}|\mathbf{x}) \log\left(\frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right) && (3.25)\\
&\quad + \int_{\mathcal{Z}} q_\phi(\mathbf{z}|\mathbf{x}) \log(p_\theta(\mathbf{x}|\mathbf{z})) && (3.26)\\
&= -\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) + \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log(p_\theta(\mathbf{x}|\mathbf{z})) \quad \text{(C.20)+(C.14)} && (3.27)
\end{aligned}$$

Equation (3.27) can now be transformed into an objective function as follows: The marginal likelihood can be maximized during the maximization of the ELBO by learning the parameters ϕ and θ of a DNN. Therefore, Eq. (3.27) needs to be negated

[5] Note that no labels $\mathbf{y}$ are introduced in this section, and the VAE only learns from the observable data $\mathbf{x}$.

so it can be optimized using gradient decent techniques perform an optimization on it. The prior of the latent representation can be freely chosen and independent of any parameter: $p_\theta(\mathbf{z}) \equiv p(\mathbf{z})$. The reconstructed observation $\mathbf{x}'$ of the generative process $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{z})$ is indexed with a prime to distinguish it from the actual observation $\mathbf{x}$. Finally, the objective for the set of observations can be written as follows:

$$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}}\, J(\theta, \phi|\mathcal{O}) \tag{3.28}$$

$$= \underset{\theta,\phi}{\operatorname{argmin}} \sum_{o=1}^{O} \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}_o) \| p(\mathbf{z}))}_{\text{Regularization}} - \underbrace{\mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x}_o)} \log(p_\theta(\mathbf{x}|\hat{\mathbf{z}}_o))}_{\text{Reconstruction}} \tag{3.29}$$

With Eq. (3.29) as the objective function, the marginal likelihood can be maximized while minimizing the negative ELBO by learning the parameters ϕ and θ. The first summand of Eq. (3.29) can be interpreted as a regularization term while the second one is the negative reconstruction error as known from auto encoder (AE) [Hin+94]. For each observation, $q_\phi(\mathbf{z}|\mathbf{x})$ is considered as a stochastic map that encodes the input $\mathbf{x}$ to the latent variables $\hat{\mathbf{z}}$ and $p_\theta(\mathbf{x}|\mathbf{z})$ as a stochastic map that decodes the latent variables $\hat{\mathbf{z}}$ to the reconstructed input $\mathbf{x}'$. This approach is called variational auto encoder (VAE).

In this paragraph, the representation of the encoder and decoder by a DNN are summarized. To keep the calculation of the KLD tractable, the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ as well as the prior $p(\mathbf{x})$ were chosen to be Gaussian distributions.[6] The prior is a D-dimensional multivariate normal distribution $p(\mathbf{z}) := \mathcal{N}(\mathbf{0}, \mathbf{I})$, while the encoder is parameterized by a DNN as follows:

$$\boldsymbol{\mu}_o = f_{\boldsymbol{\mu}}(f_{\text{enc.}}(\mathbf{x}_o)) \text{ with } \boldsymbol{\mu}_o \in \mathbb{R}^D \tag{3.30}$$

$$\log \boldsymbol{\sigma}_o^2 = f_{\boldsymbol{\sigma}}(f_{\text{enc.}}(\mathbf{x}_o)) \text{ with } \boldsymbol{\sigma}_o \in \mathbb{R}^D \tag{3.31}$$

$$\hat{\mathbf{z}}_o \sim q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\Big(\mathbf{z}; \boldsymbol{\mu}_o, \operatorname{diag}\Big(\boldsymbol{\sigma}_o^2\Big)\Big) \tag{3.32}$$

$f_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\sigma}}$ are two separated linear neural network layers with D neurons, while $f_{\text{enc.}}$ is a DNN with an arbitrary number of layers and $\mathbf{x}$ as an input. The decoder $p_\theta(\mathbf{x}|\mathbf{z})$ can be parameterized by a DNN similar to Eq. (3.32) but is eased in practice[7] to become the following:

$$\mathbf{x}'_o = f_{\text{out}}(f_{\text{dec.}}(\hat{\mathbf{z}}_o)) \tag{3.33}$$

[6] The KLD needs to be calculated by numerically solving its integral. However, it has a closed-form solution for multivariate and uni-modal Gaussian distributions (see C.2.5).

[7] Kingma et al. [Kin+13] mentioned that the output of the decoder has to be a full stochastic map if one wants to satisfy the derivation (e.g., acquire a Bernoulli distribution for binarized image data). However, in practice, the decoder's output exclusively predicts the mean value as the best representative of the stochastic map.

$f_{\text{dec.}}$ is, again a DNN with an arbitrary number of layers with $\mathbf{z}$ as the input while f_{out} is an output layer with a specific activation function that satisfies the value range of the input data. Because the probability distributions in Eq. (3.32) and Eq. (3.33) are parameterized with the DNNs, the parameters are, in fact, the weights and biases of the DNNs.

To train the DNNs that model the VAE, it is necessary to compute the gradient of the ELBO in Eq. (3.29) for the parameters θ and ϕ. However, while the reconstruction error term in Eq. (3.29) can be computed using common losses as used in AE (e.g., binary cross-entropy (BCE) or mean squared error (MSE)), the loss cannot be back propagated to the input through the stochastic map into the latent space. This is because of the actual stochastic sampling $\hat{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ in the network's forward path, for which no gradient can be defined. Kingma et al. [Kin+13] and Rezende et al. [Rez+14] circumvented this issue using a method called the reparameterization trick: Let $\hat{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ be a sampling process of a Gaussian distribution that is parametrized by a DNN. One can reparametrize the sampling from a normal distribution[8] $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using the network's predictions as $\hat{\mathbf{z}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$. Then the gradient, which is necessary for the parameter updates, of the reconstruction error term regarding θ and ϕ can be pulled into the expectation operator:

$$\begin{aligned} \nabla_{\theta,\phi} \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x}_o)} \log(p_\theta(\mathbf{x}|\mathbf{z}_o)) &= \nabla_{\theta,\phi} \mathrm{E}_{\mathcal{N}(\mathbf{0},\mathbf{I})} \log(p_\theta(\mathbf{x}|\boldsymbol{\mu}_o + \boldsymbol{\sigma}_o \odot \epsilon)) && (3.34) \\ &= \mathrm{E}_{\mathcal{N}(\mathbf{0},\mathbf{I})} \nabla_{\theta,\phi} \log(p_\theta(\mathbf{x}|\boldsymbol{\mu}_o + \boldsymbol{\sigma}_o \odot \epsilon)). && (3.35) \end{aligned}$$

Therefore, the gradient can be calculated because it becomes independent of the stochastic sampling. The expected value can then be calculated using Monte Carlo sampling

$$\mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x}_o)} \log(p_\theta(\mathbf{x}|\mathbf{z})) \simeq \frac{1}{S} \sum_{s=1}^{S \to \infty} \log(p_\theta(\mathbf{x}|\boldsymbol{\mu}_o + \boldsymbol{\sigma}_o \odot \epsilon_s)), \tag{3.36}$$

while Kingma et al. [Kin+13] stated, that a sample size of $S = 1$ and the reconstruction loss calculation similar to the training of AEs is sufficient.

Finally, the gradient of the loss from the KLD can be back propagated without any issues because the loss is directly calculated between a fixed prior $p(\mathbf{z})$ and the output of the deterministic output layers $f_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\sigma}}$ (see C.2.5) with

$$\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}_o) \| p(\mathbf{z})) = \frac{1}{2} \sum_{d=1}^{D} \log \sigma_d^2 + \sigma_d^2 + \mu_d^2 - 1 \tag{3.37}$$

The findings of Eq. (3.37) and Eq. (3.36) allows for the analytical calculation of the objective function. Therefore, Eq. (3.29) can be optimized using conventional optimization algorithms such as SGD.

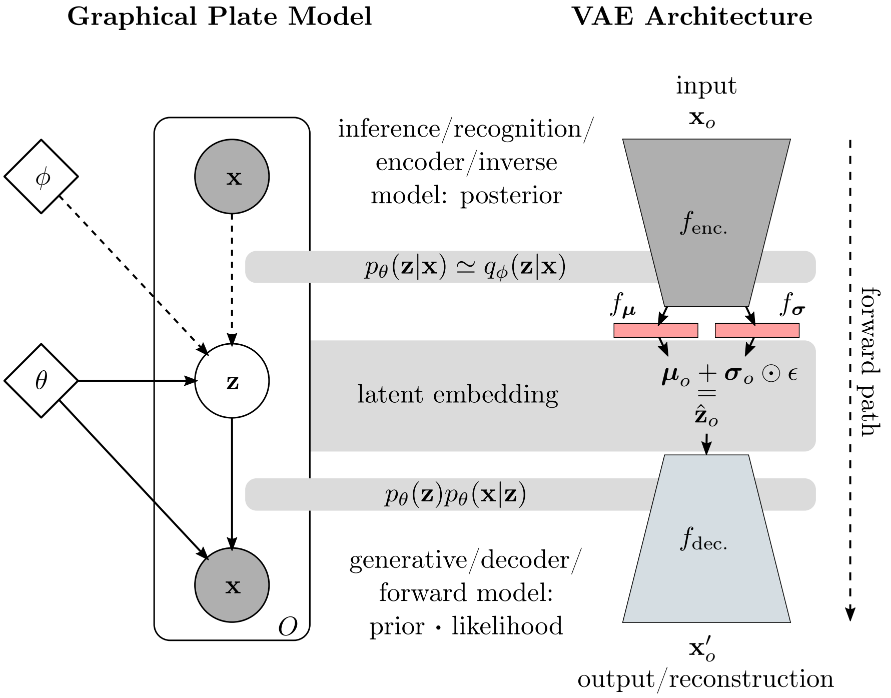


Figure 3.3: The generative model (GM) of the variational auto encoder (VAE) (left) plus the associated VAE architecture as DNNs (right).

Figure 3.3 (left) shows the graphical model of VAE. The GM is the same as in Fig. 3.2, but the approximate distribution $q_\phi(\mathbf{z}|\mathbf{x})$ (i.e., the inference model) is represented by the dotted line. Figure 3.3 (right) shows a possible structure of a DNN in the VAE with the encoder network, single layers for mean and logarithmic variance, the Gaussian sampling in the latent space, and the decoder network.

Figure 3.4 shows the results of training the VAE with the MNIST database of handwritten digits by LeCun et al. [LeC+98] after various epochs and then generating images $\mathbf{x}'$ for selected samples $\mathbf{z}$ using the decoder $p_\theta(\mathbf{x}|\mathbf{z})$. In this example, the VAE learns a latent embedding of the MNIST data set in a 2D latent space.[9] The VAE learns to cluster similar samples, which are, in the case of the MNIST data set, images with the same numerical information but differing writing styles. The clustering happens naturally because the VAE tries to reduce the confusion between

[8] It is worth mentioning that the normal distribution of the reparameterization trick, despite the same form, has nothing to do with the prior in Eq. (3.32).

[9] The dimension of the latent space, denoted as D_z, can be of arbitrary size. It is generally smaller than the input dimension $D_\mathrm{z} \ll D_\mathrm{x}$ to maintain the hourglass/bottleneck architecture and to force the VAE to learn a latent embedding.

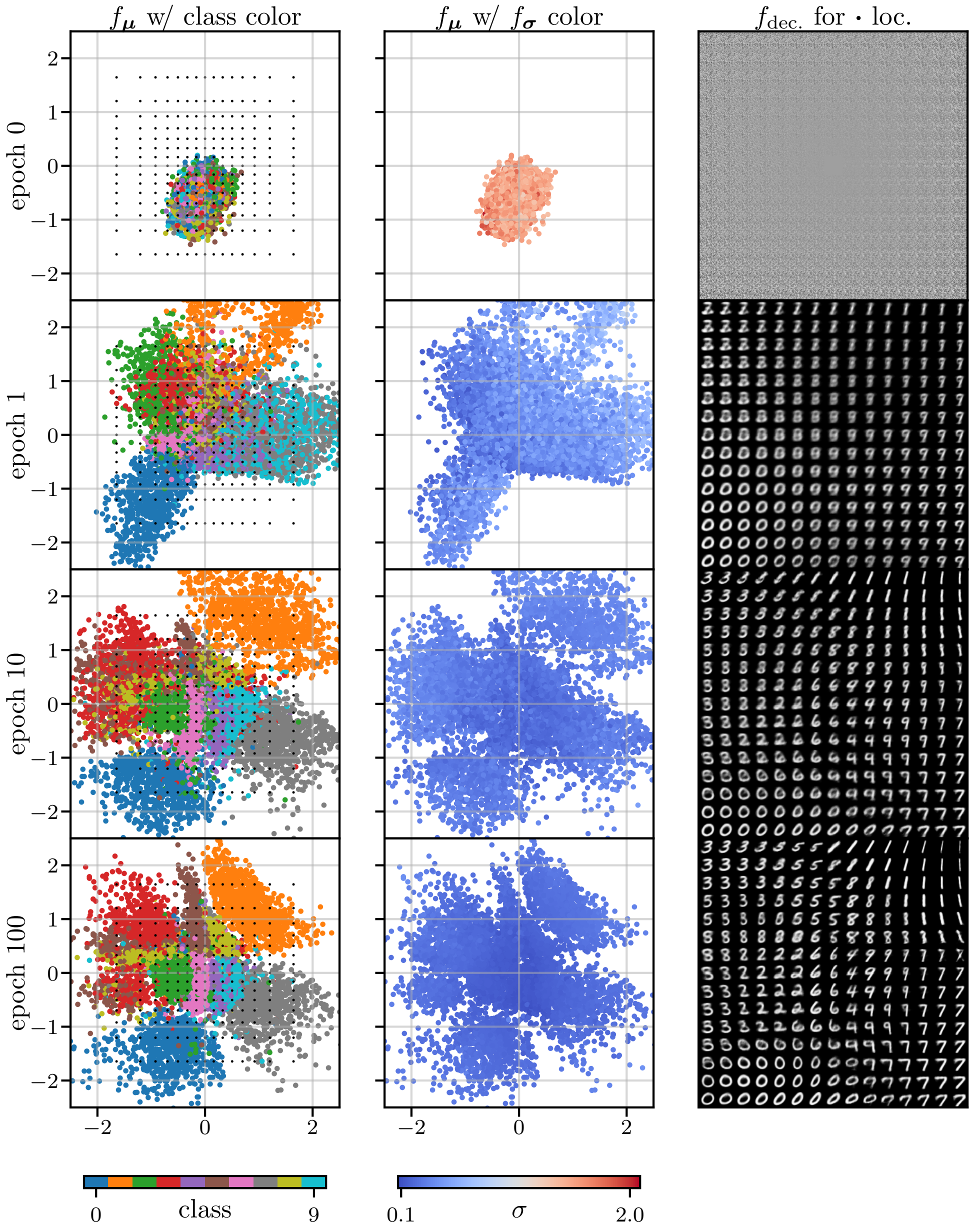


Figure 3.4: Encoding and decoding of the MNIST test data set ($\mathcal{O}_{\text{test}}$) using the trained encoder/decoder network after 0, 1, 10, and 100 epochs of training (top to bottom). Left col.: $f_\mu(f_{\text{enc.}}(\mathcal{O}_{\text{test}}))$ with class coloring (i.e., 0–9). Middle col.: $f_\mu(f_{\text{enc.}}(\mathcal{O}_{\text{test}}))$ with standard deviation coloring (i.e., $\sigma^2 = \sum_{D_z} \exp f_\sigma(f_{\text{enc.}}(\mathcal{O}_{\text{test}})))$. Right col.: Decoding of certain $\mathbf{z}$ values according to dot pattern (denoted by •) in the left column. See Appendix B.2.1 for training setup and evolution of losses.

encoding and decoding, which is introduced by the sampling in the latent space. The learned per-sample distribution, $q_\phi(\mathbf{z}|\mathbf{x}_o)$, can also be considered as the "uncertainty" or the "information" of the sample $\mathbf{x}_o$, which is learned during training by means of lowering the standard deviation[10] (see Fig. 3.4 (mid.)).

However, because the VAE can directly train high-dimensional data, such as images, it can also create new images. None of the images in the right column of Fig. 3.4 are in the MNIST dataset, but they were generated by the VAE from scratch.

3.3.3 Conditional Variational Autoencoder (CVAE)

The process of generating the observed variable $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z}, c)$ with $\hat{\mathbf{z}} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and c being another observable stochastic variable is considered in this section. Then the conditional likelihood for $\mathbf{x}$ under a given c is $p_\theta(\mathbf{x}|c) = \int_\mathcal{Z} p_\theta(\mathbf{x}|\mathbf{z}, c)p(\mathbf{z})$. Figure 3.5 is the GM of the conditional variational autoencoder (CVAE). The CVAE is an extension to the VAE, which handles the previously mentioned GM [Soh+15a]. The observation variable c is added to the model in Fig. 3.3.

In a CVAE, the approximate distribution (encoder) is given as $q_\phi(\mathbf{z}|\mathbf{x}, c)$. The ELBO of the CVAE becomes

$$\mathcal{L}(\mathbf{x}, c; \phi, \theta) = \overset{\text{Eq. (3.27)}}{\ldots} = -\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, c) \| p_\theta(\mathbf{z})) + \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x},c)} \log(p_\theta(\mathbf{x}|\mathbf{z}, c)), \quad (3.38)$$

from which the objective function can be derived in a similar way as in Section 3.3.2. In contrast to the VAE model from Fig. 3.3, an additional input for c is concatenated with the input space $\mathbf{x}$, as well as with the latent space $\mathbf{z}$. The resulting model can be trained in the same way as the VAE.

The observed variable c can be viewed from two different perspectives. The first one is that of a target label corresponding to $\mathbf{x}$ (c.f. $\mathbf{y}$ from Section 3.1). The second one is that of another information source or modality supporting $\mathbf{x}$.

While the VAE is learning in an unsupervised fashion and only $\mathbf{x}$ is given as the training set, the CVAE can be regarded as undergoing supervised learning where both $\mathbf{x}$ and c are given. For example, $\mathbf{x}$ is a handwritten numerical image and c is the corresponding numerical class label. Because the generation model of CVAE is $p_\theta(\mathbf{x}|\mathbf{z}, c)$, the corresponding handwritten numeric images can be generated by changing the value of the numeric label c and the latent variable $\mathbf{z}$ after training. $\mathbf{z}$ is independent of c, so information such as "handwriting", for which $\mathbf{z}$ is independent of the numeric label, is obtained. Figure 3.6 shows the results of training a CVAE with MNIST and then generating various handwritten images for each digit. The learned

[10] The standard deviation for the depicted example is the square root of the total variance with $\sigma^2 = \sum_{D_\mathbf{z}} \exp f_{\boldsymbol{\sigma}}(f_{\text{enc.}}(\cdot))$ (see Andres [And13]).

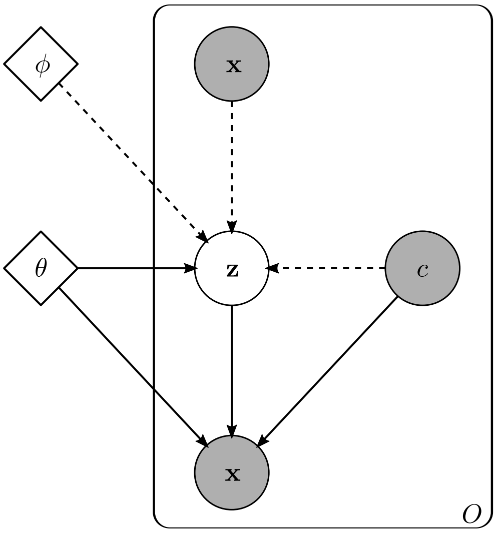


Figure 3.5: The CVAE graphical model.

embedding for each numeric label shares a similar area and shape in the latent space. The decoded images along the vertical axis reveal, that the encoder network has learned a correlation of styles and their variations among the handwritten numerical images.

Another perspective is that c is another modality that differs from $\mathbf{x}$. From this perspective, CVAE learns a stochastic transformation model from c to $\mathbf{x}$. As mentioned earlier, it is also possible to consider that $\mathbf{z}$ represents the uncertainty of the transformation. Therefore, CVAE can learn the probabilistic correspondence, even if there is no one-to-one correspondence between $\mathbf{x}$ and c.

In this thesis, to distinguish between $\mathbf{x}$ and c, the label information corresponding to $\mathbf{x}$ is denoted as c, while the label c in the target space (e.g., as one-hot encoded vector) is denoted as $\mathbf{y}$ (see Section 3.1). In recent studies on multi-modal learning, the other modality is often denoted as $\mathbf{y}$ (see Vedantam et al. [Ved+17] or Higgins et al. [Hig+17c]) or $\mathbf{w}$ (see Suzuki et al. [Suz+17]). In this thesis, the alphabet is used in a consecutive order $(a, b, \dots)$ to distinguish the modalities, which will be introduced in the following chapters.

3.3.4 Further Remarks on VAEs

Further essential remarks and concepts of the VAE framework are comprised in this section.

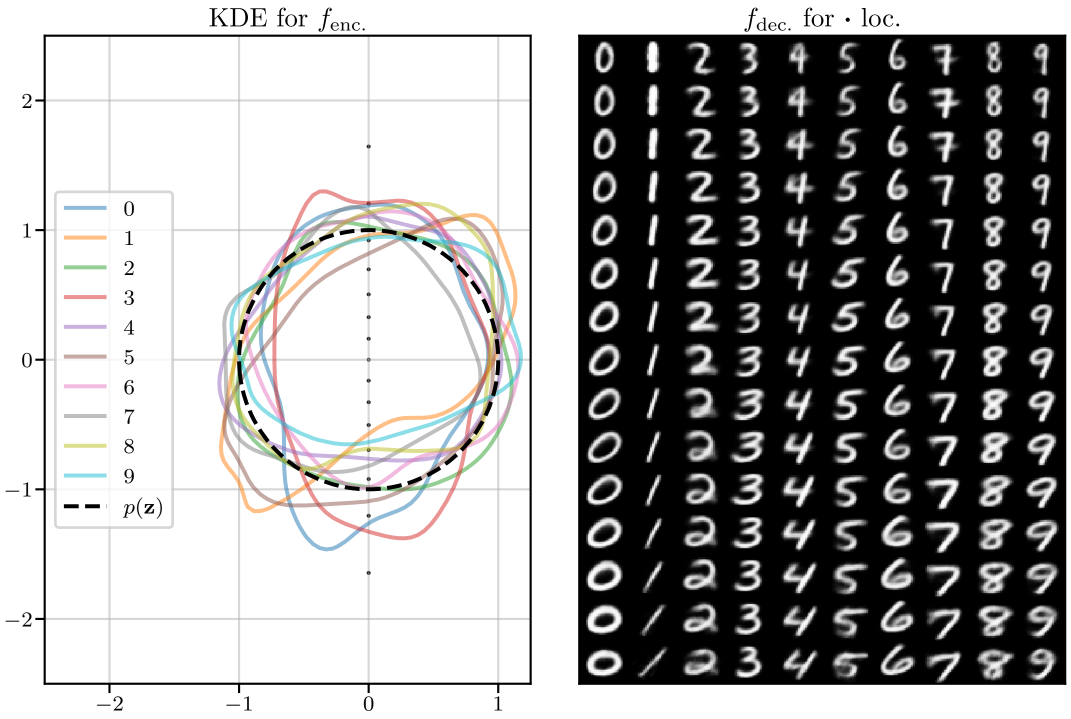


Figure 3.6: Encoding and decoding of the MNIST test data set ($\mathcal{O}_{\text{test}}$) using the trained encoder/decoder network after 100 epochs of training. Left: KDE of $f_{\boldsymbol{\mu}}(f_{\text{enc.}}(\mathcal{O}_{\text{test}}, c_{\text{test}}))$ at 2σ of the prior for each numeric label c. Right: Decoding of certain $\mathbf{z}$ values according to dot pattern (denoted by •) in the left figure. The horizontal column corresponds to each numeric label c. The vertical column corresponds to various brushstrokes or styles that were found by the CVAE during training, which are shared among the numeric labels. See Appendix B.2.2 for the training setup.

3.3.4.1 Amortized VAE

The concept of VI is not to learn the distribution of the whole data set, but only a per-sample distribution where each input sample has its own parameter set. Compared to the mean-field VI approach, which learns a dedicated parameters θ set for each sample, the VAE makes use of a DNN that learns a shared parametrization for the whole input data set. The DNN finds similarities in the input data like recurrent patterns or styles and naturally clusters these features to efficiently store and share the information in the network's weights. This concept is called amortized inference [Zha+19].[11]

3.3.4.2 Blurry Reconstruction

The clustering of similar samples in the latent space happens naturally because the VAE tries to reduce the confusion between encoding and decoding that is introduced by the sampling in the latent space. The sampling process in the latent spaces causes confusion in the decoder's output, which lets the decoder put learning pressure on only the most dominant features in the data. These dominant features are commonly represented by the low frequencies[12] that mainly describe the data set and its samples. Information that is stored in relatively high frequencies, like information about serifs, background, or noise, is only decisive for single samples. This causes the decoder network to reproduce an average reconstruction value over multiple samples in which these artifacts occur. That lets the reproduced sample appear blurred (see Fig. 3.7).

The high frequency information is pruned by the VAE because of the small dimensions of the latent space, which limits the amount of information that can be stored. Furthermore, the prior, $p(\mathbf{z})$, acts as an additional low-pass because it regularizes/-confines where and how information may be embedded. It is worth noticing that the distribution of the data set in the latent space does follow the prior distribution only qualitatively.

Further enhancements to the VAE, like combinations with a GAN to learn sharper reconstructions, were made by [Lar+16], but commonly, techniques used to enhance the reconstruction causes the loss of the expressiveness of the latent embedding in general (see "VAEs and GANs" talk by Rosca [Zhu+18]). However, the increase in latent dimensions already relaxes the low-pass behavior and enables the network to pass additional information through the bottleneck, which results in sharper reconstructions. High dimensionality also causes the VAE to embed individual, so-called, generative factors along single dimensions, which is explained in Section 3.3.4.3.

[11] A data set that only contains samples that share no information with each other would result in a non-amortized embedding that also follows the prior.

[12] by means of a Fourier analysis of the data

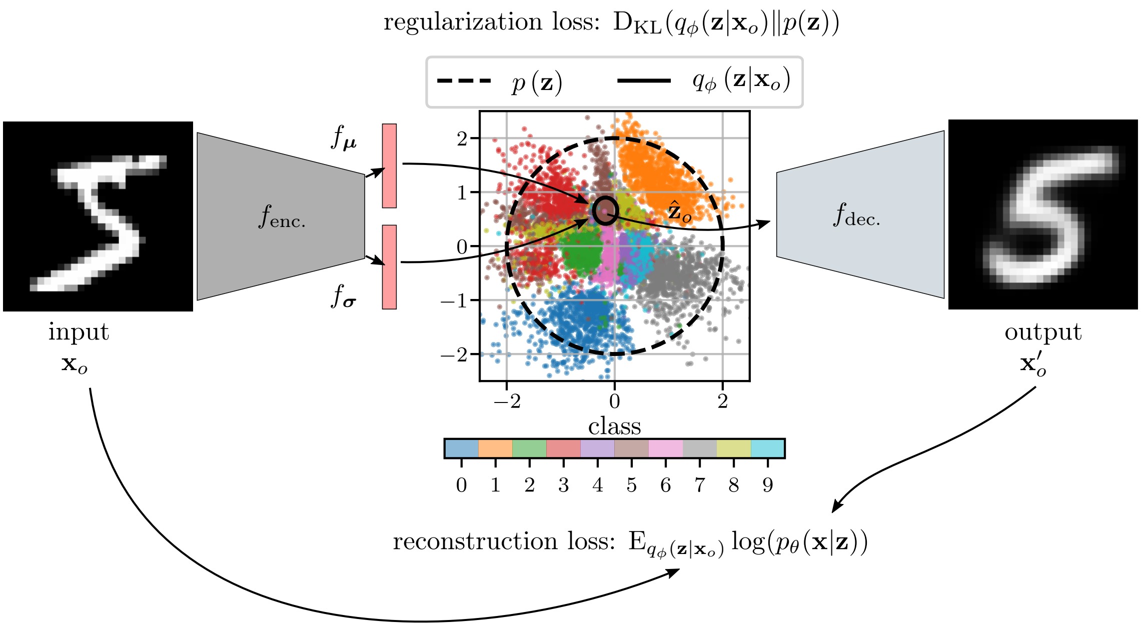

regularization loss: $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_o)\|p(\mathbf{z}))$
$p(\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x}_o)$
f_μ
$f_{\text{enc.}}$
f_σ
$\hat{\mathbf{z}}_o$
$f_{\text{dec.}}$
input
$\mathbf{x}_o$
output
$\mathbf{x}_o'$
2
1
0
−1
−2
−2
0
2
class
0
1
2
3
4
5
6
7
8
9
reconstruction loss: $E_{q_\phi(\mathbf{z}|\mathbf{x}_o)}\log(p_\theta(\mathbf{x}|\mathbf{z}))$

Figure 3.7: VAE during training. The input $\mathbf{x}_o$ parametrizes a Gaussian $q_\phi(\mathbf{z}|\mathbf{x}_o)$ using the encoder network in the latent space, which is regularized by the prior $p(\mathbf{z})$. A sample $\hat{\mathbf{z}}_o \sim q_\phi(\mathbf{z}|\mathbf{x}_o)$ is fed into the decoder network, which outputs a $\mathbf{x}_o'$ as a reconstruction of $\mathbf{x}_o$. The sampling process and limited confinement in the latent space causes the reconstruction to blur and generalize over a set of similar inputs. This apparent drawback causes the VAE to cluster similar inputs and work out decisive and striking features that describe the data set the most. The contours of $p(\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x}_o)$ are drawn with two times their standard deviation. Scatter plot shows the encoding of $f_\mu(f_{\text{enc.}}(\mathcal{O}_{\text{test}}))$ using the MNIST test data set ($\mathcal{O}_{\text{test}}$) after 100 epochs of training concerning Fig. 3.4.

3.3.4.3 Disentangled Latent Space and Generative Factors

The feature of the VAE used to disentangle the latent space and assign generative factors of the data set to axes was first observed by Higgins et al. [Hig+17a], who introduced the concept of a beta variational autoencoder (βVAE). This disentangling effect can already be seen in Fig. 3.6, where one numeric label can facilitate the whole 2D latent space. Generative factors that represent the stroke width or skewness, for example, are placed along the vertical axis. This illustrates a more general

finding, namely that one can generalize that a VAE is capable of projecting complex data into a linear separable latent space.

In contrast, a low dimensionality in the data set causes the VAE to entangle the data set's embedding. Figure 3.4 reveals that the embedding of the whole MNIST data set in two dimensions causes the VAE to separate individual classes and distribute them in some arbitrary area. This causes the generative factors to be entangled among the two dimensions, which can be seen in the reconstructed images in Fig. 3.4 because they change styles and numeric labels by sweeping along a single latent dimension.

However, just increasing the number of latent dimensions is not sufficient. Higgins et al. [Hig+17a] observed, that the disentangling effect can be controlled by a single scalar β, which leverages the regularizer against the reconstruction loss. Commonly, β is chosen so the regularizer has more influence on the loss term. One side effect is that the reconstruction becomes worse (see Section 3.3.4.2), and a high β causes the VAE to not learn anything from the data. A technique called warm-up by Sønderby et al. [Søn+16], which involves linearly increasing β over the first epochs, circumvents this issue. Thus, the VAE learns a proper reconstruction in the early epochs and the gradually increasing β causes the regularizer to disentangle the latent space in the later epochs.

Various enhancements were achieved after the introduction of the βVAE. Some contributions worth mentioning include semi-supervised βVAE by Li et al. [Li+17], SCAN by [Hig+17c], further attempted explanations by [Bur+18b], one-shot-learning by Higgins et al. [Hig+17b], InfoGAN by [Che+16], disentangling by factorizing by Kim et al. [Kim+18], β-TCVAE by [Che+18], and spatial broadcast decoder by Watters et al. [Wat+19b].

3.3.4.4 VAE for Semi-Supervised Learning

VAE can also be used as a model for semi-supervised learning [Kin+14b; Maa+16]. In a semi-supervised learning setup, in general, a small labeled set $\mathcal{O}_{\mathrm{L}} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_O, \mathbf{y}_O))$ and a larger unlabeled data set $\mathcal{O}_{\mathrm{U}} = (\mathbf{x}_1, \ldots, \mathbf{x}_{O'})$ are used to train a discriminative model $p(\mathbf{y}|\mathbf{x})$. Although $p(\mathbf{y}|\mathbf{x})$ is generally trained on the labeled set $\mathcal{O}_{\mathrm{L}}$, in the framework of semi-supervised learning, the unlabeled set $\mathcal{O}_{\mathrm{U}}$ is also used for learning, which is further explained in Appendix B.4.

The semi-supervised VAE consists of a generative and discriminative model that can be trained in an end-to-end fashion, similarly to VAE and CVAE.[13] The advantage of the M2 model by Kingma et al. [Kin+14b] is that it can treat supervised and unsupervised learning in a unified way. In the DGM, the difference between supervised and unsupervised models only depends on whether the random variables that correspond to the labels are observables or latent variables. The discriminative model can also be integrated into the generative model as a probability distribution

[13] See Fig. B.9 for the GM of the semi-supervised VAE.

to generate labels given the input. The objective function in Appendix B.4 includes all of these frameworks and can be trained in a unified manner by optimizing the objective function. Other semi-supervised models based on VAE include ADGM and SDGM by Maaløe et al. [Maa+16], which enhance the M2 model.

4 Multi-Modal Perception

The definition plus taxonomy of multi-modal information and the related research to build a foundation of multi-modal perception are discussed in this chapter. Furthermore, the objectives and challenges are derived for later investigation in the subsequent chapters. First, the various multi-modal ML topics that emerged over the past decades and their taxonomies which relate to this work are recaptured in Section 4.1. Second, Section 4.2 contains a discussion of the fundamental properties of multi-modal data, such as heterogeneity, correlation, and ambiguity. Finally, Section 4.3 contains an overview of related problem settings of multi-modal learning and previous studies relevant to this thesis.

4.1 Multi-Modal Machine Learning – Definition and Taxonomy

The motivation for using more than one modality in an observation arises from three main benefits. First, having access to multiple modalities that observe the same phenomenon may allow for highly accurate, consistent, and dependable predictions. Second, having access to multiple views of a phenomenon by similar or heterogeneous modalities enables the retrieval of complementary information, which would not be visible using a single modality. Third, a system gains robustness when it has multiple sensors because it can still operate when one of the modalities is missing.

All these properties are biologically inspired from neuroscience by the properties of the multi-sensory learning pressures that have been suggested to act in the perirhinal cortex of the ventral stream in the human brain [She+16]. Therefore, researchers of multi-modal ML pursue all these desirable benefits, which resulted in various branches over the past decades. It is worth mentioning that in the ordinary language multi-modal ML is commonly referred as the topics of sensor fusion, which is diversified in the following paragraphs.[1]

This work is grounded in the multi-modal ML domain, which is a vibrant field of research with huge area of overlapping and interdisciplinary scientific topics. The goal of multi-modal ML is, generally speaking, to build algorithms that can process

[1] The history plus various definitions of sensor fusion do exists and are collected by Koch [Koc20].

and relate information from multiple data sources which lead to a better outcome compared to algorithms which only use a single data source. Therefore, multi-modal ML is a meta-research field which approaches and methodologies can be applied to virtually every other research domain.

Baltrušaitis et al. [Bal+19] recently developed a taxonomy of multi-modal ML that is shown and extended in Fig. 4.1. The approaches of this work in the taxonomy diagram, which is visualized via the shaded nodes, are grounded in the following paragraph.

Representation involves learning how to represent or summarize the complementarity and supplementary features of multi-modal data. One major issue is the challenge of how to construct representations for heterogeneous modalities. However, it is often beneficial to transfer varying modalities into a common representation with similar statistical quantities across modalities compared to the original data, to become more efficient and suitable for downstream applications [Ngi+11]. For instance, a video is often a composition of visual and auditory streams that substantially differ in their data rate per dimension and overall dimensionality, which makes it tedious to combine the raw data for any application. However, deriving an intermediate representation in which all signals are aligned eases their application to further processing steps.

Baltrušaitis et al. [Bal+19] proposes distinguishing between two categories of multi-modal representation: joint and coordinated. A function f obtains a joint representation if it combines all uni-modal signals $a, b, \ldots$ and projects them into the same representation z: $z = f(a, b, \ldots)$. Furthermore, the functions $f_1, f_2, \ldots$ obtain coordinated representations by processing each uni-modal signal separately while ensuring similarity constraints between all projections: $f_1(a) \approx f_2(b) \approx \ldots$.

Joint representation is commonly used when multi-modal data is available during training and inference. One way to build a supervisely trained DNN, for instance, is to concatenate all input data and feed it into the DNN. The DNN then learns to combine different inputs and correlate them to the desired output. This is commonly referred to as early fusion, which is applicable if the nature of the respected modalities allows it. In the case of varying modality natures, like auditory or visual data, where modality-specific DNN architectures (RNN or CNN respectively) are necessary to gain the best performance. Mid or late fusion are considered in this case, which is the technique of merging and concatenating the output of hidden layers as described by Ramachandram et al. [Ram+17a].

Other approaches are probabilistic graphical models (PGMs). These are generative approaches to emitting the observed data through the adaptation of latent random variables. The advantage of a PGM is its explicitness of relations and conditions between variables, which can be dependent on the prior knowledge or complexity of the issue, either given by an expert or automatically retrieved from data. An

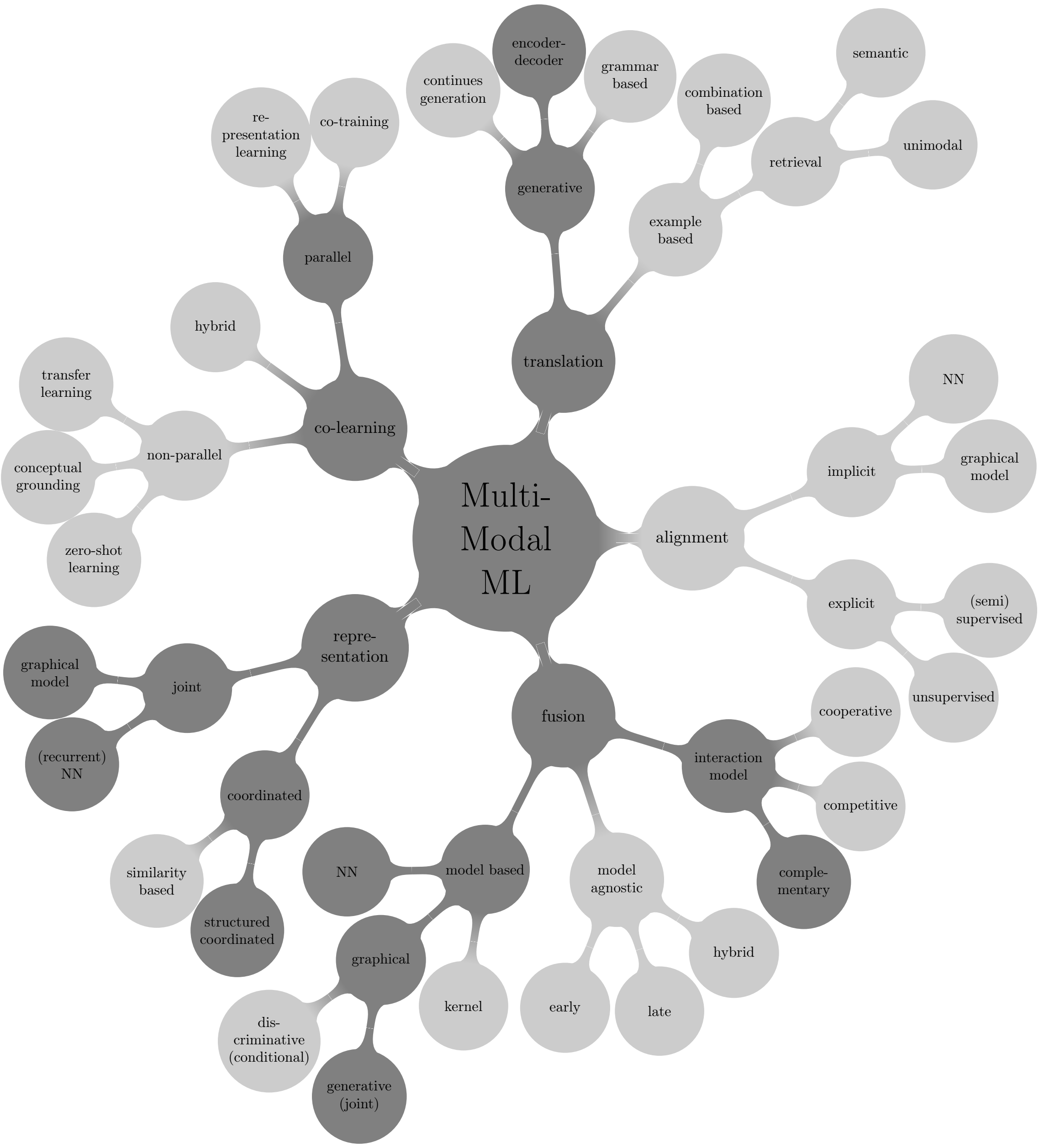


Figure 4.1: Embedding of this work's topics in the taxonomy of multi-modal machine learning.

extensive review of PGMs and its variants and techniques was provided by Koller et al. [Kol+09].

As discussed in Chapter 3, the VAE combines two approaches, PGM and DNN. All variables are comprised by a feed-forward DNN that can be efficiently trained using the backpropagation (BP) algorithm. Therefore, DNNs gain the power of Bayesian inference, which can be explicitly formulated into the network's architecture. This ability is facilitated in this work by formulating the ELBO for a multi-modal observation.

Coordinated representation enforces similarity between uni-modal representations, which can be distinguished into two categories. Similarity models minimize the distance between modalities in the coordinated space. A well-recognized example is DeepFace by Taigman et al. [Tai+14] who formulated a triplet-loss to train a Siamese DNN that minimizes the Euclidean distances of face images from the same person while maximizing the distance between different people.

Structured coordinated space extends the former approach by formulating additional constraints between the modalities' representations. Ramachandram et al. [Ram+17a] listed canonical correlation analysis (CCA), a technique for computing a linear projection that maximizes the correlation between two modalities, as a model under the structured coordinated space taxonomy. Thus, the non-linear VAE variant by Wang et al. [Wan+16], VCCA, implicitly falls under the same taxonomy. The author of this work formulates the relationships between uni-modal and multi-modal observations in the later chapters, which therefore, shares the same taxonomy of coordinated representation.

Translation is utilized to address the mapping from one modality to another. The most widely known applications are speech-to-text and text-to-speech, which either generate written text from dictated words or read out written words, which is available on most web sites and smartphones nowadays. Generally, every classifier or feature extractor can be seen as a translation application because they transform the input data into another representation with similar or equal information content. However, Ramachandram et al. [Ram+17a] differentiated two categories of translation: generative and example-based. While an example-based method can be used to retrieve the best translation from some dictionary, a generative approach can be used to derive a model from a set of translations. The latter taxonomy fits the definition of the proposed approach in Chapter 5 and is, therefore, discussed further.

Ramachandram et al. [Ram+17a] defined generative approaches as techniques used to construct models that can perform multi-modal translation given a uni-modal source instance. This taxonomy is limited in that only uni-modal to uni-modal (1-to-1) translations are defined. This work broadens the definition because translating multi-modal to multi-modal data (m-to-n), multi-modal to uni-modal data (m-to-1) and vice-versa (1-to-n) are reasonable options. However, generative models are further categorized into grammar-based, continuous generation, and encoder-decoder models. Approaches based on grammar are restricting by the target domain's grammar definition, which has to satisfy templates that consists of a subject, object, and verb in a sentences-based translation task, for instance. Continuous generation models extend the former approaches by handling stream-like input and output modalities, which are common for translating between temporal sequences (e.g., text-to-speech). Finally, the most suitable definition for this work's approach are encoder-decoder models, which first encode the source modality to an intermediate representation, as discussed in the former section, which is then interpreted by a decoder to generate the target modality.

Encoder-decoder models are based on end-to-end trained DNNs with a fan-in to fan-out (i.e., bottleneck or hourglass) architecture. This way, the network is forced to compress the data into a vectorial representation in the bottleneck to its most representative information. The most desirable feature of vectorial representation is that the DNN possibly learns to reflect the underlying generative factors of the data (see [Loc+18]).

Alignment involves elaborating on relationships and correspondences between subsections of multi-modal observations. It is categorized into the two types of implicit and explicit alignment. Explicit alignment occurs when the main modeling objective is to align instances of two or more modalities. In contrast to explicit alignment, implicit alignment approaches involve learning how to latently align data during training so that the learned representation can be used for downstream applications. DTW or HMM are well known algorithms that do perform unsupervised alignment of speech utterances. However, the approach proposed in this work does not align observations as defined by Ramachandram et al. [Ram+17a]. Rather it aligns representations between sets of observations. This approach is deeply related to manifold alignment as comprised by Wang et al. [Wan+11].[2] For instance, assuming a bi-modal observation (a, b) is represented by its latent sample $z_{\mathrm{a,b}}$, then the unimodal representations z_{a} and z_{b} are explicitly aligned via the KLD, such that $z_{\mathrm{a}} \approx z_{\mathrm{a,b}}$ and $z_{\mathrm{b}} \approx z_{\mathrm{a,b}}$. This approach falls under the taxonomy of representation rather than alignment.

Co-learning involves cases in which one modality lacks resources (like samples or labels), while another modality that observes the same phenomena does not. Therefore, the resource rich modality can be used to aid in the modeling of the resource poor modality. Co-learning methods can be categorized into three cases based on the configuration of the data set: parallel, non-parallel, and hybrid. If the data set consists of parallel observations, then all observations are directly linked to each

[2] See Wang et al. [Wan+11] for an overview of manifold alignment and Wang et al. [Wan+09] for unsupervised approaches in particular.

other as if all recordings where synchronized. Non-parallel data sets do not require the modalities to be directly linked to each other. Rather, they only have to share categories or concepts. In the hybrid data setting, non-parallel modalities are bridged by some shared modality or a dataset that serves as a proxy.

The author of this work does not explicitly deal with co-learning as defined by Ramachandram et al. [Ram+17a], but some similarities can be drawn to the parallel case as follows: It is assumed, that the data sets used for training consist of observations that were performed simultaneously or that differing data sets are at least semantically aligned. In a bi-modal case, a sample consisting of both modalities (a, b) represents the resource rich observation because it comprises all available information. The uni-modal observations a and b represent the resource poor observations because they might be missing some information that is only available in the other modality. However, as discussed in the former paragraph, all sets of modalities are linked to each other to enable the learning of a coherent representation by guiding the resource poor observations to have similar embeddings as the resource rich one.

Fusion is presumably one of the most studied topics in multi-modal machine learning, which is the concept of integrating information from multiple modalities to improve results in all branches of machine learning: classification, regression, clustering, or dimensionality reduction. The benefits are twofold: First, having multiple and redundant modalities that observe the same phenomenon enable the making of failsafe and robust predictions. Second, by combining interacting modalities, one can observe more states and features together than with a single modality. Fusion approaches can be categorized into two domains [Ram+17a]: model-agnostic approaches that do not directly depend on a specific machine learning method and model-based approaches that explicitly address fusion in their formulation and construction. One advantage of model agnostic approaches is that they can be implemented using almost any unimodal classifier or regressor.

However, not all sensor fusion applications are of the same kind or have the same benefits. Elmenreich [Elm02] defined three interaction models of how the data from multiple sensors can be fused: complementary, competitive, and cooperative.

Interaction models are divided into complementary, competitive, and cooperative approaches. They define how the data from multiple sensors can be fused.

A sensor configuration is called complementary if the sensors do not directly depend on each other but can be combined to give a more complete image of the phenomenon under observation. This resolves the problem of the incompleteness of sensor data. An example of a complementary configuration is the employment of multiple cameras that each observe disjunct parts of a room in surveillance application.

Sensors in a competitive configuration have each sensor delivering independent measurements of the same property. Competitive configurations are used on faulttolerant and robust systems. An example would be the reduction of noise by combining two overlaying camera images.

A cooperative sensor network uses the information provided by two independent sensors to derive information that would not be available from the individual sensors. An example of a cooperative sensor configuration is stereoscopic vision: a three-dimensional image of the observed scene can be derived by combining two-dimensional images from two cameras at slightly different viewpoints.

The applied methods in this work build upon a multi-modal generative model, which graphical model assumed independent observations retrieved using complementary modalities (see Fig. 5.1). This suits the context of complementary fusion with the goal of resolving ambiguous observations of single sensors.

Model-agnostic approaches do not follow any specific rule about how to combine the information provided by multiple modalities. They are application and architecturally driven and may be implemented using almost any unimodal preprocessing methods like classification or regression. Therefore, they are easy to implement but suffer from techniques that are not designed for multimodal data. Liggins et al. [Lig+08] generally distinguishes fusion into early (raw-based), mid (feature-based), late (decision-based), and hybrid fusion. Early fusion involves integrating raw signals directly after necessary preprocessing steps like normalization or smoothing. Mid fusion is applied after considerable preprocessing of the raw signal, to extract features like the statistical quantities of the signal, for instance. Late fusion involves performing integration after each of the modalities has made a decision (for instance a classification or regression). Hybrid fusion combines all the above approaches by taking the most profitable representation of each modality for fusion into account.

Model-based approaches follow an explicit technique on how to combine the modalities' data for fusion. They are categorized into three approaches: kernel-based methods, PGM, and DNN. Kernel-based methods, like multiple kernel learning, are extensions of kernel support vector machines in a way because modality-specific kernels exploit similarities between modalities. PGMs can be further split into approaches that models joint (also known as generative) or conditional (also known as discriminative) probability. The benefits of graphical models are twofold: The spatial and temporal structure of the data can be easily exploited, and they allow the building of human expert knowledge about the process in the models that leads to interpretability. DNNs are data-driven approaches that learn the correlations between and beneficial combinations of modalities from the data through non-linear functions. This is a decisive feature compared to other fusion approaches because they do not suffer from over-simplifications or necessary assumptions in the setup. Furthermore, the use of DNNs allows for end-to-end training of both multi-modal representation and fusion, which leads to high performance as well as the high interpretability of the data, for instance [Ram+17a].

The author of this work uses the approaches of Kingma et al. [Kin+13] and Rezende et al. [Rez+14] that combine PGMs and DNNs to learn a representation of the data through a generative processed that is parametrized by a DNN known as VAE. Suzuki et al. [Suz+17] built up on this by exploiting VAEs to perform bi-modal exchange. The author extends the former approaches in this work to model the multi-modal observation in a generative process to identify a coherent representation between all subsets of modalities. The fact that the generative process is modeled by a DNN allows to learn fusion in a data-driven way without labels while representing all information in a single latent vector. This feature can be used in a further downstream application to challenge the model-agnostic fusion approaches.

4.2 Multi-Modal Properties

As explained in the introduction, the author of this work focuses on multi-modal data that allows for heterogeneity, correlations, and ambiguities. However, these fundamental properties of multi-modal data taxonomy tree from Section 4.1 because they are omni-present issues of all multi-modal ML approaches. Heterogeneity is defined in Section 4.2.1 while correlation is captured in Section 4.2.2 and, finally, ambiguities are discussed in Section 4.2.3.

4.2.1 Heterogeneity of Multi-Modal Data

The most affected topics by heterogeneity are co-learning and, in particular, transfer-learning because they deal with the translation between modalities. Therefore, the work on transfer learning by Weiss et al. [Wei+16] is adapted for the taxonomy of heterogeneous multi-modal data in this thesis.

Weiss et al. [Wei+16] restricted themselves to the definition that difference in the observation space between modalities exclusively denote heterogeneity while an equal observational support denotes homogeneity.[3] This statement is generalized in the following section in order to capture the requirements of this work.

[3] Weiss et al. [Wei+16] relates to the “feature space”, while the more general terms “observation space”, “observation support”, or “support of observations” relate to any non-trivial raw, feature, or symbolic set.

4.2.1.1 Domain

A set of observations from one modality is defined as $a = a_1, \dots, a_N \in \mathcal{A}$ while $\mathcal{A}$ denotes the observation space. With $p(a)$ being the marginal likelihood distribution, the domain can be defined as the set $\{\mathcal{A}, p(a)\}$. The following properties of the two domains $\{\mathcal{A}, p(a)\}$ and $\{\mathcal{B}, p(b)\}$ can be identified, which results in heterogeneity:

1. the observation spaces are different: $\mathcal{A} \neq \mathcal{B}$
2. the marginal likelihood distributions are different: $p(a) \neq p(b)$
3. observation spaces and the distributions are different

As an example, in a multi-modal sensor setup, "1" can be a set of different kinds of sensors or similar sensors with different setups while "2" is another environment that is sensed.

4.2.1.2 Task

Given a domain $\{\mathcal{A}, p(a)\}$, a task is constructed as $\{\mathcal{Z}, p(z|a)\}$ by the target support $\mathcal{Z}$, hereinafter referred to as latent space, and the posterior distribution[4] $p(z|a)$. $\mathcal{Z} = \{z_1, \dots, z_N\}$ denotes the set of latent samples pertaining to $\mathcal{A}$.

Depending on its purpose, a realization of the posterior distribution z_n is called a class, label, attribute, target, latent code, or generative sample. Furthermore, as with the discussion of the domain, the heterogeneity properties of Section 4.2.1.1 result in the following cases:

1. corresponding latent spaces are different: $\mathcal{Z}_{\mathcal{A}} \neq \mathcal{Z}_{\mathcal{B}}$
2. the posteriors differ: $p(z|a) \neq p(z|b)$
3. latent spaces and posteriors differ

As an example, in a multi-modal sensor setup, "1" can occur when two observed latent spaces have different numbers of generative factors while "2" relates to the distributions of generative factors.

4.2.1.3 Conclusion

Based on these discussions, it can be seen that the domains and tasks relate to the observations and latent space, respectively. Following the definitions by Weiss et al. [Wei+16], given two domains or tasks, the issue of translating from one domain or task to another is called transfer learning. The issue setting of the transfer leaning can be applied to co-learning and multi-modal ML in general. Heterogeneity relates

[4] Also known as inference, inverse, recognition, or encoder model.

to at least one difference in the domain, which leads to differences in the task, while the common underlying issue in multi-modal ML relates to the identification of correlations between modalities.

4.2.2 Correlations between Modalities, Classes, and Attributes

As stated before, a fundamental constraint for sensor fusion is that two signals are somehow correlated. This means that for a generative process, when a generative factor is varied slightly, observations through multiple modalities of the same phenomenon covary correspondingly. However, this does not mean that all observations change by a constant factor or that the factor is the same for every modality, which makes standard calculus or correlation analysis hardly applicable to complex multi-modal data. Nevertheless, it can be assumed for real-world data that continual changes in the generative factors result in continual changes in observation and that binary switches do not exists in nature (this is further motivated in Section 4.2.3). Furthermore, the generative process of multiple observations also illustrates the causality/correlation issue: correlation supports the notion of causation (dependency), but it does not prove it. It can generally be assumed that two different sensing modalities are correlated by means of the differing physical influences, but any modality does never influences the cause and, therefore, not the outcome of a measurement. This argument can be supported by Bayesian graphical models, whereby, if one wants to model such a generative graph, the dependencies between observations break up because all information is already supported by the latent factor.

An observation sample a_n can be surjectively associated with a latent sample z_n. A latent sample z_n consists of generative factors that can include the class association, class attributes, and the instance attributes. Figure 4.2 illustrates various types of correlations between the attributes and classes of uni- (three top rows) and bi-modal (three bottom rows) observations, and two examples are always given (left and right). The first row shows two examples that do not have any correlation because the attributes' translation (left) and rotation (right) are independently altered. Attribute correlation, on the contrary, is shown in the second row, where the left-to-right translation is correlated to cw rotation (left) or up-to-down translation is correlated to scale (right). The third uni-modal example shows an additional class correlation, where any object shape shows the same attribute correlation as in the before mentioned example.

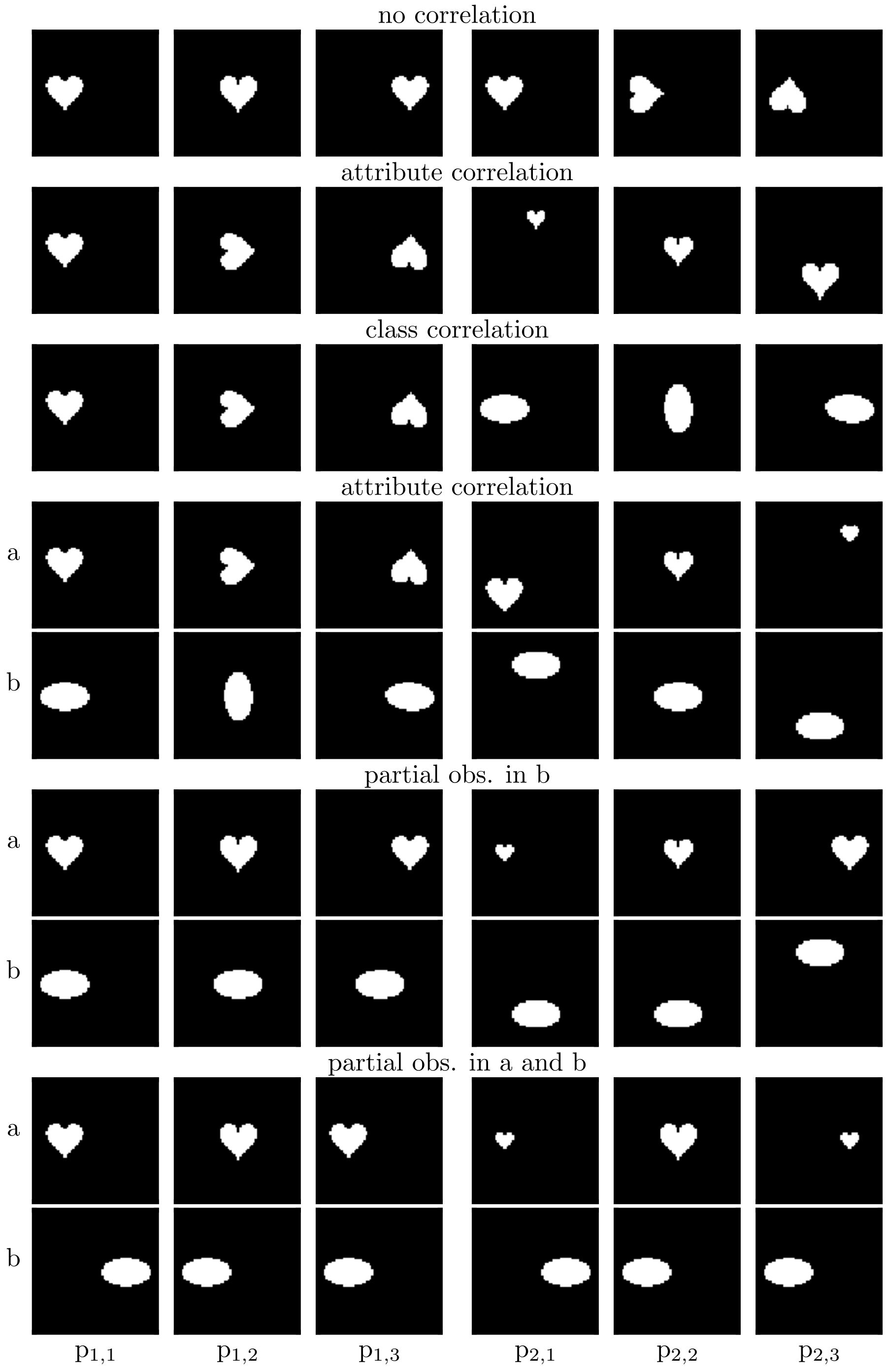


Figure 4.2: Various types of correlation are shown for uni- and multi-modal observations by means of the dSprites data set. The corresponding generative factors are swept for three samples (i.e., $p_{\cdot,1} \rightarrow p_{\cdot,2} \rightarrow p_{\cdot,3}$) in two examples each.

The first bi-modal observation extends the former correlations to two views (i.e., a and b of a phenomenon). The left example shows the linear correlation between the same attributes while the right example shows diverse attribute correlations. Note, that the mentioned example is supplementary and does not show any ambiguity between the modalities.

The second bi-modal observations introduce the partial observability of the generative factors through b. Although a varies for all observations, b stops varying for $(p_{1,2}, p_{1,3})$ and $(p_{2,1}, p_{2,2})$. Thus, b only partially allows the deduction of the generative factors through its observations, which makes b ambiguous concerning a. Furthermore, a enables full observability, which leads to the fact that any observation from b can be deduced through a.

The final example introduces ambiguities in both modalities, which only enables the deduction of the generative factors when both modalities are observed. a varies when b is constant and vice versa, and thus, they are no longer supplementary. This is, however, a very simplified example but demonstrates the necessity of mulit-modal observation and fusion in deducing the full underlying generative factors.

4.2.3 Requirements for Multi-Modal Data and Observation Ambiguities

Perry et al. [Per+10] stated that Hebbian learning relies on the fact that the same objects are continuously transformed to their nearest neighbor in the observable space. Higgins et al. [Hig+16] adopted this approach in their assumptions that this notion can be generalized within the latent manifold (i.e., latent space) learning. However, neither a coherent manifold nor a proper factorization of the latent space can be trained if these assumptions are not fulfilled by the dataset as shown by Higgins et al. [Hig+16]. In summary, this means that observations of natural phenomena must have the property of continuous transformation concerning their properties (e.g., the position and shape of an object) such that a small deviation of the observations results in proportional deviations in the latent space and that switches do not exists.

Adopting this assumption for multi-modal datasets means that observations should correlate, if the same phenomena is observed by each modality. Therefore, a small deviation in the underlying generative factors (i.e., the common latent representation) between all modalities conducts a proportional impact on all observations. This becomes a fundamental requirement for any multi-modal dataset because correlation and coherence are within the context of multi-modal ML.

The previously mentioned assumptions only hold, if and only if (iff) the modalities observe the same phenomena and are able to rectify all its properties. This property of retrieving equivalent information by every modality (i.e., $p(a) = p(b)$) is called supplementary and only offers redundancy, as discussed by Baltrušaitis et al. [Bal+19]. In any other case there exists a factorization of the likelihood, $p(a, b|z)$, for two modalities a and b, which factorization of z might differ. Assume for this particular case that the generative factor z can be bi-parted into $z = (\dot{z}, \tilde{z})$ with $\dot{z} \perp \tilde{z}$. Factorizing the likelihood results in $p(a, b|z = (\dot{z}, \tilde{z})) = p(a|\dot{z}, \tilde{z})p(b|\dot{z}, \tilde{z})$. Now, for example, if b cannot sense a particular property, then it is likely that it becomes independent of one factor and, therefore, marginalizes to $p(b|\dot{z}, \tilde{z}) = p(b|\dot{z})$ (e.g., see second bi-modal example in Fig. 4.2). The described marginalization manifests itself in ambiguous observation and results in the partial observability of the latent factors. In the previously mentioned example, this means, that if b makes ambiguous observations, they can only be resolved if the rectification is done by a or a and b together. Generally, partial observability results in injective mappings, ambiguous observations result in surjective mappings, and full observability without ambiguous observations results in bijective mappings of the posterior distribution between the observable and latent space.

Summarizing the requirements for multi-modal observations results in the fact that observed data must have the property of continuous transformation between the observable and latent space sets. Furthermore, a small change in the manifold of the latent space should at least result in a proportional deviation of the modalities in the observable space. Observations of any modality are called ambiguous when deviations in the latent space do not result in changes in the observable space. Finally, the postulation by Higgins et al. [Hig+16] can be extended to a requirement for multi-modal observations that it is important for the observed multi-modal data to be generated using factors of variation that are densely sampled from their respective continuous distributions.

4.2.4 Conclusion and Challenges faced in this Work

Based on the discussion so far, a multi-modal data set consists out of at least two different observations, a and b. It is called heterogeneous, iff one difference between their domains or tasks can be identified. Otherwise, it is considered to be homogeneous. However, it is worth mentioning that different observation spaces and different likelihoods are not mutually exclusive, and therefore, it is necessary to take the possibility into account that likelihoods, and even tasks, may be different for the same observations. This is the part that has been barely focused on in past multi-modal studies on ML. Therefore, this work focuses on DGMs, which can capture these differences by means of observation ambiguities.

Furthermore, the concept of correlation and ambiguities between observations received from different modalities was introduced. Correlation refers to the mutual equivalence of varying factors between observations while ambiguities introduce the partial observability of the correlation. This issue of ambiguous observations, particularly during modality drop-out, has not been studied yet but is a crucial aspect of any resilient and trustworthy ML. Therefore, the author of this work also focuses on DGMs, which can coherently capture correlations and ambiguities such that drop-out no longer causes catastrophic outcomes for downstream applications, for example.

4.3 Deep Multi-Modal Machine Learning

Beneficially combining multiple modalities poses many difficulties: how to combine the data from heterogeneous sources, how to deal with different levels of noise, and how to handle missing data. Furthermore, the ability to represent data in a meaningful way is crucial to any multi-modal issue and highly depends on the architectural choice and desired task goals. All these topics deeply relate to sensor fusion, which is a major topic in every autonomous system. Generally, any autonomous systems is advised to apply fusion to its multiple sensors and models to provide information to the autonomous system to estimate its own state (proprioceptive sensing) and the environment in which it is operating (exteroceptive sensing). Traditional fusion architectures, as discussed in Section 4.3.1, are built hierarchically from the raw sensor level up to the cognitive behaviors. However, due to the advent of DNNs, these intermediate steps are no longer necessary because end-to-end fusion learning is becoming increasingly feasible (see [Ram+17a]).

The idea of the traditional architecture approaches and their requirements on algorithms are captured in Section 4.3.1. Finally, state-of-the-art (SOTA) approaches in deep multi-modal ML are discussed in Section 4.3.2 while recent developments in multi-modal DGMs are focused in Section 4.3.3.

4.3.1 Fusion in Autonomous Architectures

Figure 4.3 shows an architectural overview of autonomous systems as derivative of a Gajsk–Kuhn chart (i.e., Y diagram). In extension of the Gajsk–Kuhn chart, the nonfunctional properties of the algorithms of autonomous systems are attached by horizontal lines to the chart concerning the hierarchical structure. Four concentric circles characterize the hierarchical levels within an architectural design, with increasing information abstraction from the inner to the outer circle. Each circle characterizes the logic or physical interfaces while they separate the levels of the following domains[5]:

- Cognitive Domain: This domain describes the cognitive capabilities and behaviors of a system.
- Processing Domain: A system is assembled from subsystems that are optimized for their purposes. Different subsystems and their interconnections to each other are contemplated for each level of abstraction.
- Information Representation Domain: This domain describes the information abstraction properties that are handled by the system and its subsystems.

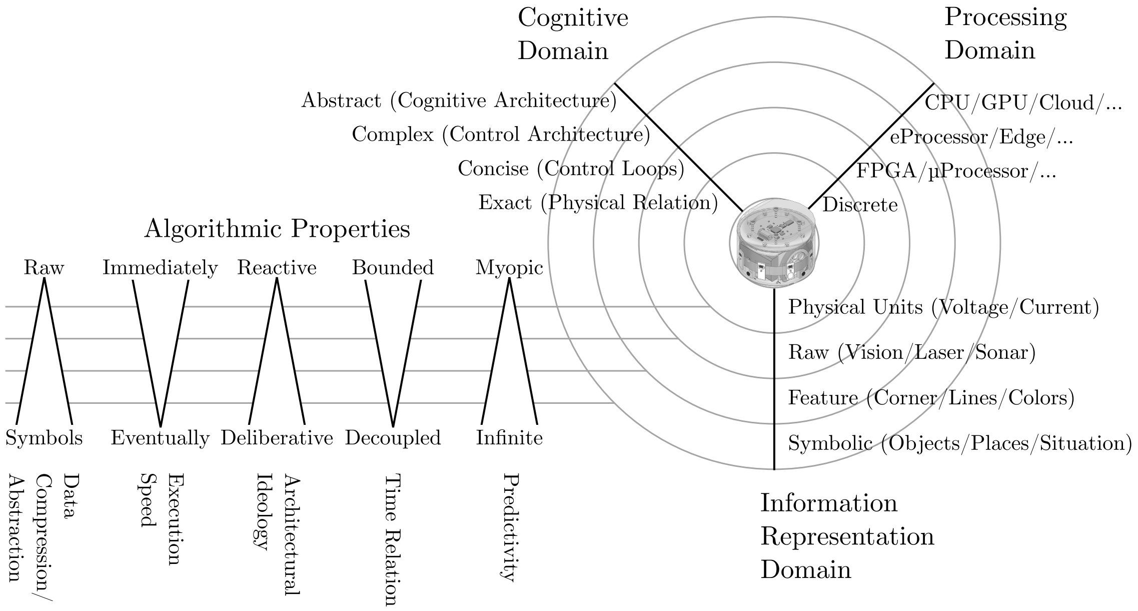


Figure 4.3: Architectural overview of autonomous systems as derivatives of a Gajsk–Kuhn chart (also known as Y diagram) with algorithmic properties attached to the hierarchical levels.

However, following the recommended and classical design of fusion architectures by Liggins et al. [Lig+01, Figure 1.4], ranging from raw- over feature- to symbolic-level, they coincide with the information representation domain. Their non-functional properties can be identified according to algorithmic properties in general. Therefore, complex autonomous systems also demand complex fusion architectures consisting of consecutive layers in which data and information is combined. In the contemporary area of fusion, DNNs offer a powerful framework for shortcutting and

[5] In comparison to the Gajsk–Kuhn chart, the non-relevant geometric properties are not respected and substituted by the information representation domain.

outperforming traditional engineered approaches by processing raw observations to outcomes with high symbolic and predictive power.

4.3.2 Deep Multi-Modal Fusion

To broaden the view of multi-modal DNN fusion, further noteworthy approaches can be found in the following literature: Vielzeuf et al. [Vie+19b] address late versus early fusion in DNNs by forwarding all intermediate hidden activations to the classifier network, which chooses the right level of fusion. Arevalo et al. [Are+17] introduced gated multi-modal units that identify intermediate representations based on a combination of data from different modalities. Kahou et al. [Kah+16], for instance, handled various feature extractors as different modalities that act on the same data to classify emotions in a video stream. Li et al. [Li+16] trained a cross-modal specific weight mask to learn the probabilities of connecting the units belonging to different modalities. A novel direction in neural architecture searching whereby the HPs like the number of layers are learned was approached by Perez-Rua et al. [Per+19]. They introduced a fusion network that can learn the best combinations and activations based on all intermediate activations of pretrained feature extractors for each modality as a succeeding approach to the CentralNet architecture by Vielzeuf et al. [Vie+19b; Vie+19a]. Even earlier attempts in the field of DNN based sensor fusion were composed by Ramachandram et al. [Ram+17a].

One topic that is of particular interest within this thesis is modality drop-out (i.e., when one sensor produces inappropriate or no data for fusion during testing). However, this research is currently just unrolling. Ngiam et al. [Ngi+11] were the first to apply autoencoder structures to unsupervised feature extraction in a bi-sensory setup, investigating classification during drop-out as well as neurocognitive similarities like the McGurk effect. They handled modality drop-out by augmenting the data sets by zeroing modality inputs during training. Inspired by the regularization technique of dropout for DNNs, Neverova et al. [Nev+16] proposed a modality drop-out method that gradually fuses observations involving the random dropping of separate channels for learning crossmodal correlations while preserving the uniqueness of each modality-specific representation. Liu et al. [Liu+17a] introduced the method of sensor drop-out to reinforcement learning and showed that the monolithic network became more resilient by randomly dropping the sensor during training.

4.3.3 Multi-Modal Deep Generative Models

The motivation for using DGM in ML applications comprises reconstruction, missing data generation, domain transfer, anomaly detection, denoising, or feature extraction (see Foster [Fos19] for an exhaustive history overview). Feature extraction is especially facilitated by the dimensionality reduction in the latent space. The latent space captures the unique and shared generative factors of the given modalities. This aspect is important from the perspective of downstream tasks, where well decomposed representations by means of latent factorization are most amenable for use on a wider variety of tasks (see Lipton [Lip16] and Mathieu et al. [Mat+19]). A downstream task is commonly referred to a supervised-learning task that utilizes a pre-trained model or component, which is, in this particular case, the retrained encoder of a DGM.

4.3.3.1 Requirements

Bengio et al. [Ben+12] summarized some requirements for the properties of a learned latent space: smoothness, temporal and spatial coherence, sparsity, and natural clustering, among others. Srivastava et al. [Sri+12] identified additional desirable properties for multi-modal representations: similarity in the representation space should reflect the similarity of the corresponding concepts, the representation should be easy to obtain, even in the absence of some modalities, and finally, it should be possible to fill in missing modalities given the observed ones.

4.3.3.2 Literature Overview

Various SOTA publications on multi-modal DGMs are discussed in the following paragraphs. Table 4.1 shows a brief overview of the publications' evaluated data sets and objectives. However, this table is revisited in the following chapters and, therefore, not discussed in full detail within this section.

Given a set of multiple modalities $\mathcal{M} = \{a, b, c, \dots\}$, multi-modal variants of the VAE have been proposed since its discovery. They have been applied to the training of generative models, mainly for multi-directional reconstruction.

Approaches that model other modalities based on conditions are the CVAE by [Soh+15a] or conditional multi-modal autoencoder (CMMA) by [Pan+17]. These methods can be applied during inference, even without conditional variables if they are trained as semi-supervised models (see Section 3.3.4.4). However, conditional approaches come with the drawback that the conditional variable is not a part of the reconstruction output. Therefore, they cannot be applied to bi-directional modality exchange.

Multi-modal AEs (i.e., the non-VI approach of a VAE) wereobviously proposed before the discovery of the VAE. Multi-modal stacked AEs, as applied by [Lar+07] or Ranzato et al. [Ran+06], are composed of various uni-modal AE architectures that are concatenated in the input, deep, or latent layers of the DNN. A noteworthy approach that can be interpreted as the predecessor to joint variational autoencoder

(JVAE) in Section 5.1.1 is the MV-AE by Ngiam et al. [Ngi+11]. Their purpose is the reconstruction of missing modalities, as demonstrated by Ngiam et al. [Ngi+11] and Cadena et al. [Cad+16]. The approach follows that of stacked AEs, but the training is conducted with an augmented data set that substitutes placeholder values for missing modalities. The substituted values may have zero values for one of the input modalities and original values for the other input modality but still require the network to reconstruct both modalities.

One of the most familiar approaches to learning a common latent representation is the CCA because it determines a linear transformation of – even heterogeneous – data sets into a subspace where the samples show correlation.[6] This approach has been extended by Andrew et al. [And+13] to capture even more complex modalities via a DNN, which they called deep CCA. The variational aspect was brought into this framework by Wang et al. [Wan+16], who proposed the variational canonical correlation analysis (VCCA) method. VCCA by [Wan+16] is used to train two VAEs together with interacting inference networks to facilitate two-way reconstruction without the need to directly model the joint distribution by enforcing its correlation via a CCA approach.

Vedantam et al. [Ved+17] (TELBO), Wu et al. [Wu+18] (MVAE), and [Shi+19] (MMVAE) pursue an expert approach that respects multiple modalities via DGMs. Vedantam et al. [Ved+17] introduced a product-of-experts objective for the bi-modal VAE, which they call the triple ELBO (TELBO). First, the full multi-modal VAE is trained, after which the encoder weights are pinned to train the remaining uni-modal networks. Wu et al. [Wu+18] proposed a Product-of-Expert architecture by first training a common VAE for each modality. Afterwards, they combine the variational distribution, which has to be Gaussian, of the set of all uni-modal encoders. Shi et al. [Shi+19] proposed a mixture-of-experts approach that involves learning a set linearly combined VAEs encoder networks.

Tsai et al. [Tsa+18] pursued a factorization of multi-modal embedding by introducing a joint-discriminative approach. They conditioned the latent space of a joint encoder via a discriminator DNN model with the goal of predicting generative factors.

The GAN approach by Liu et al. [Liu+17b] called unsupervised image-to-image translation (UNIT) is also noteworthy. They model the joint distribution between modalities using the marginals and construct a coupled GAN framework with a shared latent space.

Finally, there also exists a VAE approach that was not derived from the marginal log-likelihood but, rather, the variation of information (VoI). This approach by Suzuki et al. [Suz+17] involves modeling the mutual information between the modalities with

[6] See Section 7.1.1 for more details about CCA.

the objective to estimate the joint distribution with the capabilities of bi-directional reconstruction.

Table 4.1: List of current approaches striving against multi-modal DGMs. All data sets are revisited in Section 6.1.2. †: Data sets with GT as additional modality. ‡: Data sets with class correlation.

	data set	metric	approach
MV-AE [Ngi+11]	CUAVE, AVLetters(2)	accuracy (acc.) of linear downstream classification	zeroing modalities
VCCA [Wan+16]	noisy MNIST‡, MIR Flickr†	acc. of linear downstream classification	canonical correlation
TELBO [Ved+17]	MNIST-A†, CelebA†	IS, JSD, attribute accuracy	product-of-experts
JMMVAE [Suz+17]	MNIST†, CelebA†	log-likelihood	variation of information
UNIT [Liu+17b]	MNIST+USPS‡, MNIST+SVHN‡	average pixel accuracy	shared-latent space assumption on coupled GANs
MVAE [Wu+18]	MNIST†, bMNIST†, Multi-MNIST†, FMNIST†, CelebA†	log-likelihood	product-of-experts
MFM [Tsa+18]	MNIST+SVHN‡, POM†, CMU-MOSI†, ICT-MMMO†, YouTube†, MOUD†, IEMOCAP†	multi-class accuracy, F1 score, MAE, Pearson's ρ	factorization of multi-modal discriminative and modality-specific generative factors
MMVAE [Shi+19]	MNIST+SVHN‡, CUB†	correlation by CCA	mixture-of-experts

4.3.4 Conclusion and Challenges for this Work

None of the approaches focused on the derivation of training objective, that is consequently derived from the full marginal joint likelihood. The analytically impeccable derived approaches (like the JMMVAE) show drawbacks in the number of respected modalities, as they only target bi-modal setups. However, while the expert approaches allow an arbitrary number of modalities, they just learn combined representations in the latent space of each one. This raises the question of whether this is possible, and more importantly, whether it is practical to formulate and train a multi-modal VAE using an objective that was derived from the marginal joint likelihood. Suzuki et al. [Suz+17] and Vedantam et al. [Ved+17] argued that the training of the full multi-modal VAE is intractable because of the $2^{|\mathcal{M}|}-1$ modality subsets of inference networks. Therefore, this work derives a training objective from the joint likelihood without introducing any simplifications and brings the objective in line with those of other multi-modal DGMs. Furthermore, the architectural choice of the resulting multi-modal VAE will be introduced to maintain the tractability of even great multi-modal setups.

All the proposed approaches from Section 4.3.3 present their results via bi-modal exchange and downstream application using complex SOTA data sets. However, perusing improved over-all-scores does not mean that the learned latent representations also obey the introduced properties of coherency, especially in the case of ambiguous or modality drop-out, for example. Therefore, the author of this work focuses on an the analytical analysis of whether the proposed approach obeys ambiguities and drop-outs. Furthermore, it is also necessary to identify and introduce comprehensible data sets that reveal the properties of the learned latent space.

5 Multi-Modal Variational Autoencoder

In this chapter, a deep generative model (DGM) that enables the coherent learning of latent space embeddings for multi-modal data is introduced. An approach of using shared encoder networks for each observation set was chosen to facilitate the learning of multi-modal embeddings by means of a DGM.

The most related research, that strives in a similar direction compared to this work, tackles the approach of modality exchange. Modality exchange approaches are used to try to learn shared representations between modalities such that a learned encoder-decoder setup can transform one modality to the other.[1] As shown by Srivastava et al. [Sri+12], if one can obtain an appropriate shared latent representation of multi-modal observations, then one can also transform the modalities back and forth via the shared latent representation. Another technique for modality transformation is to train a DNN to transform in one direction separately. Several DGMs that transform modalities in one direction have been proposed by Kingma et al. [Kin+14a], Sohn et al. [Soh+15b], and Pandey et al. [Pan+17].

However, when multiple modalities are transformed into each other, this approach results in an exponential increase in the number of networks required for the number of modalities. This is visualized in Fig. 5.1, where the number of paths in the inference part of the DGM grows according to the number of modalities. Furthermore, because the networks in each direction are trained independently, the hidden layer is not shared, and differing representations of the modalities are obtained. Therefore, the previously mentioned methods are not very suitable for the goal of learning coherent representations for multiple modalities.

To transform the different modalities, it is important to model the shared representation is a stochastic latent variable. This is because, as described in Sections 1, 4, and 6, different modalities have different distributions, and thus, their relationships are not deterministic.

The DBM is known as a method for achieving the detection of the correlations between different modalities, which leads to a coherent and shared representation

[1] For example, learning the similarity between a light detection and ranging (LiDAR) and camera observation such that one can convert red/green/blue (RGB) images to range scans.

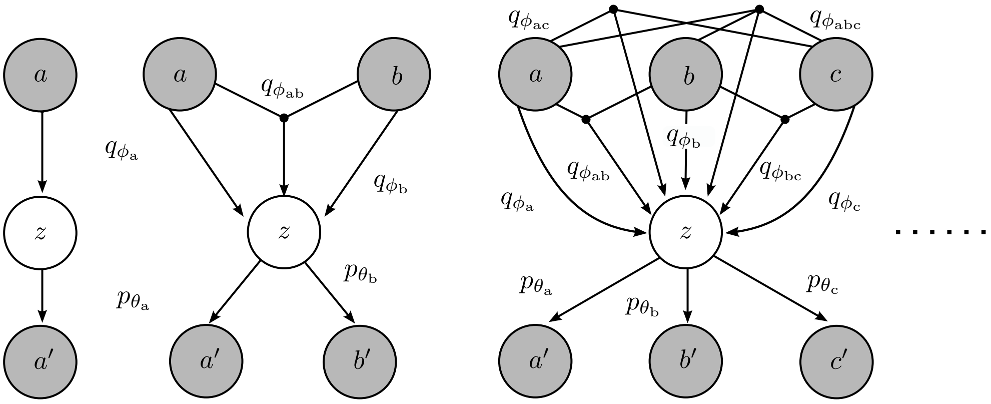


Figure 5.1: DGM concept of multi-modal VAEs with increasing numbers of modalities. The left DGM depicts the common VAE. The middle DGM shows a bi-modal DGM with three paths (i.e., encoder networks) in the inference part. The left DGM has already $2^{\#\text{modalities}} - 1 = 7$ paths. For the sake of brevity and compared to the other plate models from Chapter 3, the observations O are neglected and the parameters are absorbed by the encoders q_{ϕ_*} and decoders p_{θ_*}.

[Sri+12; Soh+15b]. However, the learning rule of a DBM is based on the MCMC method and it is cumbersome to train such a model with large scale and high-dimensional data as input. Recently, the variational auto encoder (VAE) by Kingma et al. [Kin+13] and Rezende et al. [Rez+14] has been proposed for use as a model that can flexibly train a DGM by VI (see Section 3.3). Because this approach can use the backpropagation (BP) method at the time of training as well as common DNNs, which already come with mature leaning techniques, it can handle large and complex data sets more efficiently than the conventional MCMC-based training of a DBM.

Various autoencoder approaches that can handle multimodal data are investigated in the first study in Section 5.1. It can be shown that these models do not fully follow the derivation from the marginal log-likelihood. This may lead to incoherent shared representations when single modalities are dropped, which causes inconsistencies, modality exchange, or downstream applications.[2]

In the second study in Section 5.2, the multi-modal variational autoencoder (M²VAE) is proposed as a method to solve the previously mentioned issues by following the derivation of the full multi-modal marginal log-likelihood.

[2] Autoencoder architectures are often used to encode the high-dimensional input data (e.g., images) to a low-dimensional feature vector, which can be used for further, so called, downstream applications (e.g., classification or regression tasks).

Learning is done by comparing the distances between the learned distribution and the modalities' observations. This method solves the problem of correlating raw observations with ambiguities and enables coherent embedding, even when modalities drop out. Figure 5.2 qualitatively shows the overall concept of a multi-modal VAE with observation ambiguities in the data set.

In the third study in Section 5.3, the behavior of the regularizer term is investigated. As mentioned in Section 3.3.2, the learned per-sample distribution q_ϕ can also be considered as the "uncertainty" or the "information" of the sample observation. Consequently, q_ϕ should behave accordingly by means of increasing variances when modalities drop out during training. Therefore, the validity of the ELBO term of the VAE, and particularly the M²VAE, is questioned.

In the final study in Section 5.4, the capability of the M²VAE to consecutively fuse multiple modalities in the latent space is investigated. This particular application is important for real-life observations where observations by multiple sensors are not done synchronously but, rather, one after another or in a distributed fashion. Therefore, the latent embeddings of the single observation need to be fused by the multi-modal encoder DNN in the latent space instead. Because the latent space does not follow statistical properties which are sufficient for fusion, the proposed approach follows a re-encoding scheme. First, the latent embeddings are decoded into observations, after that they are re-encoded altogether. However, as mentioned in Section 3.3.4.2 and shown in Fig. 3.7, the reconstructed observation only qualitatively follows the real observation. Even worse, the information decays or changes when re-encoding happens too often (i.e., observation → encoding → decoding → encoding → ...), as shown by Dosovitskiy et al. [Dos+16] as iterative re-encoding. To circumvent this issue, an "auto re-encoding" approach that stabilizes re-encoding and facilitates the consecutive fusion during training is introduced in this section.

For the derivation of the proposed approach, various multi-modal VAE approaches are revisited in Section 5.1. This reveals, that the M²VAE is a logical enhancement of prior approaches:

$$\text{M}^2\text{VAE (sec. 5.2)} \supset \text{JMMVAE (sec. 5.1.2)} \supset \text{JVAE (sec. 5.1.1)} \supset \text{VAE (sec. 3.3.2)}$$

Furthermore, the nomenclature slightly changes in comparison to that in Chapter 3. First, the alphabet is used in a consecutive order $(a, b, \ldots)$ to distinguish the modalities instead of $\mathbf{x}$ or $\mathbf{y}$. Second, the parameters θ and ϕ of the encoder q_ϕ and decoder p_θ are broadly neglected for the sake of brevity.[3] However, they are written explicitly when formulating the training objectives of each approach.

[3] This leads to a slightly alternative derivation of the common VAE, which is, therefore, re-derived in Appendix B.6 using the mentioned nomenclature.

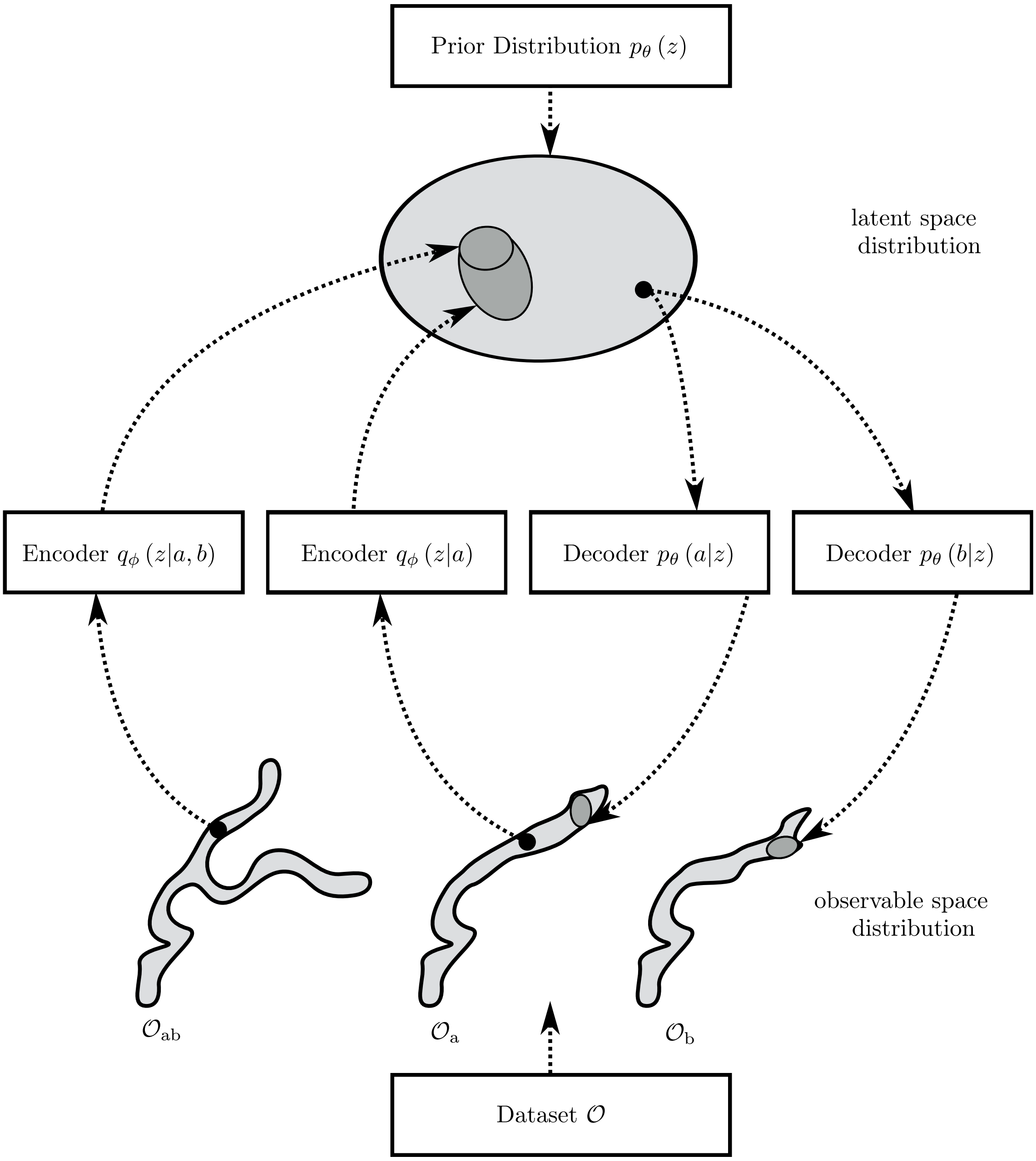

Prior Distribution $p_\theta(z)$
latent space distribution
Encoder $q_\phi(z|a,b)$
Encoder $q_\phi(z|a)$
Decoder $p_\theta(a|z)$
Decoder $p_\theta(b|z)$
observable space distribution
$\mathcal{O}_{\mathrm{ab}}$
$\mathcal{O}_{\mathrm{a}}$
$\mathcal{O}_{\mathrm{b}}$
Dataset $\mathcal{O}$

Figure 5.2: Latent space concept of a bi-modal data-set ($\mathcal{O}_{\mathrm{ab}}$) and its corresponding uni-modal data-sets ($\mathcal{O}_{\mathrm{a}}$ and $\mathcal{O}_{\mathrm{b}}$). The observable distributions have regions where they are coherent and share the same information (lower part). In the regions where the information of $\mathcal{O}_{\mathrm{a}}$ and $\mathcal{O}_{\mathrm{b}}$ diverges, the observations can be collaboratively fused to resolve ambiguities (upper part). The bi-modal encoder can embed observations in the latent space with lower uncertainty than its uni-model counterpart. The decoder for each modality reconstructs a latent sample into the observable space. This figure is an extension of the uni-modal latent space concept by Kingma [Kin17].

The main contributions of this study are the following:

- A DGM approach called M^{2}VAE is presented in Section 5.2. It is trainable on arbitrary large modality sets in an end-to-end fashion and handles modality drop outs during testing while maintaining the latent embeddings.
- A mathematical proof is derived in Section 5.3: in the case of a sensor drop-out that introduces ambiguities, the statistical properties of per-sample distributions capture the observation uncertainty.
- An auto re-encoding stabilization technique that facilitates a multi-modal, in-place posterior fusion is introduced in Section 5.4.

5.1 Preliminary Approaches

This section includes the mathematical foundations for the derivation of the proposed M^{2}VAE. First, the JVAE is derived in Section 5.1.1 based on the joint marginal log-likelihood. Second, the joint multi-modal variational autoencoder (JMMVAE) is derived Section 5.1.2 based on the variation of information (VoI). The evolution of these models is visualized in Fig. 5.3.

5.1.1 Joint Variational Autoencoder (JVAE)

When more than one modality is available (e.g., a and b) as shown in Fig. 5.1, the derivation of the ELBO $\mathcal{L}_\mathrm{J}$ for a marginal joint log-likelihood $\log p(a) := \mathrm{L_J}$ is analog to the uni-modal VAE.

5.1.1.1 Derivation of the Training Objective

The common VAE approach from Section 3.3.2 can be extended to the marginal joint log-likelihood, from which the variational bound can be derived as follows:

$$\begin{aligned}
\mathrm{L_J} &= \log(p(a,b)) && && (5.1)\\
&= \sum_z q(z|a,b)\log(p(a,b)) && \text{Eq. (C.6) w/o cond.} && (5.2)\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a,b)}{p(z|a,b)}\right) && \text{Eq. (C.2)} && (5.3)\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a,b)}{p(z|a,b)}\frac{q(z|a,b)}{q(z|a,b)}\right) && \text{mul. by 1} && (5.4)\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a,b)}{q(z|a,b)}\frac{q(z|a,b)}{p(z|a,b)}\right) && \text{reo.} && (5.5)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_z q(z|a,b) \log\left(\frac{p(z,a,b)}{q(z|a,b)}\right) && && (5.6)\\
&\quad + \sum_z q(z|a,b) \log\left(\frac{q(z|a,b)}{p(z|a,b)}\right) && \text{Eq. (C.3)} && (5.7)\\
&= \mathcal{L}_\mathrm{J} + \mathrm{D}_\mathrm{KL}(q(z|a,b)\|p(z|a,b)) && \text{C.2 \& (C.5)} && (5.8)\\
&\geq \mathcal{L}_\mathrm{J} && && (5.9)
\end{aligned}
$$

Now, the ELBO $\mathcal{L}_\mathrm{J}$ can be rewritten to facilitate approximate inference:

$$
\begin{aligned}
\mathcal{L}_\mathrm{J} &= \sum_z q(z|a,b) \log\left(\frac{p(z,a,b)}{q(z|a,b)}\right) && && (5.10)\\
&= \sum_z q(z|a,b) \log\left(\frac{p(a,b|z)p(z)}{q(z|a,b)}\right) && \text{Eq. (C.1)} && (5.11)\\
&= \sum_z q(z|a,b) \log\left(\frac{p(z)}{q(z|a,b)}\right) + \sum_z q(z|a,b) \log(p(a,b|z)) && \text{Eq. (C.3)} && (5.12)\\
&= \sum_z q(z|a,b) \log\left(\frac{p(z)}{q(z|a,b)}\right) && && (5.13)\\
&\quad + \sum_z q(z|a,b) \log(p(a|z)) + \sum_z q(z|a,b) \log(p(b|z)) && \text{Eq. (C.7)} && (5.14)\\
&= -\mathrm{D}_\mathrm{KL}(q(z|a,b)\|p(z)) && \text{C.2} && (5.15)\\
&\quad + \mathrm{E}_{q(z|a,b)} \log(p(a|z)) + \mathrm{E}_{q(z|a,b)} \log(p(b|z)) && \text{C.1} && (5.16)
\end{aligned}
$$

Three different terms can be identified from Eq. (5.16) to maintain the training objective:

$$
\begin{aligned}
\mathcal{L}_\mathrm{J} = &\underbrace{-\mathrm{D}_\mathrm{KL}(q_{\phi_\mathrm{ab}}(z|a,b)\|p(z))}_{\text{Regularization}} && (5.17)\\
&+ \underbrace{\mathrm{E}_{q_{\phi_\mathrm{ab}}(z|a,b)} \log(p_{\theta_\mathrm{a}}(a|z))}_{\text{Reconstruction wrt. } a} + \underbrace{\mathrm{E}_{q_{\phi_\mathrm{ab}}(z|a,b)} \log(p_{\theta_\mathrm{b}}(b|z))}_{\text{Reconstruction wrt. } b} && (5.18)
\end{aligned}
$$

Equation (5.18) shows the JVAE's objective. It is built up based on one bi-modal encoder-network q_{ϕ_ab} and two decoder-networks for each modality, p_{θ_a} and p_{θ_b}. The KLD regularizes the joint encoder q_{ϕ_ab} concerning the prior, while the two reconstruction terms optimize each decoder p_{θ_a} and p_{θ_b} concerning the bi-modal input.

5.1.1.2 Discussion

Figure 5.3 shows the JVAE as a DGM. However, given Eq. (5.18) it is not clear how to perform inference if the dataset consists of samples lacking from modalities (e.g., for samples i and k: $(a_i, \varnothing)$ and $(\varnothing, b_k)$).

Ngiam et al. [Ngi+11] propose the training of a bi-modal deep AE using an augmented dataset with additional samples that experience modality drop-out. Therefore, one could triple the dataset in a bi-modal case with ⅓ of the complete bi-modal observation, ⅓ of observations with one modality present and the other set to zero, and ⅓ vice versa. This is, however, a very questionable approach because zeroing inputs may cause a DNN to split internal representations and, therefore, lose vital capacity as well as the ability of crossmodal representation.

5.1.2 Joint Multi-Modal Variational Autoencoder (JMMVAE)

While the JVAE approach cannot directly be applied to missing modalities, Suzuki et al. [Suz+17] proposed the use of a JMMVAE that is trained via two uni-modal encoder-networks and a bi-modal en-/decoder-network with one objective function. This objective is derived from the VoI[4] of the marginal conditional log-likelihoods $\log p(a|b)p(b|a) =: \mathrm{L}_{\mathrm{M}}$ by optimizing the ELBO $\mathcal{L}_{\mathrm{M}}$.

5.1.2.1 Derivation of the Training Objective

First, the conditional probability is investigated as follows:

$$\begin{aligned}
p(a|b) &= \frac{p(z,a|b)}{p(z|a,b)} && \text{Eq. (C.2)} \quad (5.19)\\
&= \frac{1}{p(z|a,b)}\frac{p(z,a,b)}{p(b)} && \text{Eq. (C.1)} \quad (5.20)\\
&= \frac{1}{p(z|a,b)}\frac{p(a,b|z)p(z)}{p(b)} && \text{Eq. (C.1)} \quad (5.21)\\
&= \frac{1}{p(z|a,b)}\frac{p(a|z)p(b|z)p(z)}{p(b)} && \text{Eq. (C.7)} \quad (5.22)\\
&= \frac{1}{p(z|a,b)}\frac{p(a|z)p(z|b)\frac{p(b)}{p(z)}p(z)}{p(b)} && \text{Eq. (C.1)} \quad (5.23)\\
&= \frac{p(a|z)p(z|b)}{p(z|a,b)} && (5.24)
\end{aligned}$$

Furthermore, the marginal log-likelihood of a conditional distribution can be rewritten as follows to obtain its ELBO:

$$\begin{aligned}
\mathrm{L}_{\mathrm{M_a}} &= \log(p(a|b)) && (5.25)\\
&= \sum_z q(z|a,b)\log(p(a|b)) && \text{Eq. (C.6) w/o cond.} \quad (5.26)
\end{aligned}$$

[4] A graphical representation of the VoI for two and three modalities can be found in Appendix B.9.

$$\begin{align}
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a|b)}{p(z|a,b)}\right) && \text{Eq. (C.2)} \tag{5.27}\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a|b)}{p(z|a,b)}\frac{q(z|a,b)}{q(z|a,b)}\right) && \text{mul. by 1} \tag{5.28}\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a|b)}{q(z|a,b)}\frac{q(z|a,b)}{p(z|a,b)}\right) && \text{reo.} \tag{5.29}\\
&= \sum_z q(z|a,b)\log\left(\frac{p(z,a|b)}{q(z|a,b)}\right) && \tag{5.30}\\
&\quad + \sum_z q(z|a,b)\log\left(\frac{q(z|a,b)}{p(z|a,b)}\right) && \text{Eq. (C.3)} \tag{5.31}\\
&= \mathcal{L}_{\mathrm{M_a}} + \mathrm{D_{KL}}(q(z|a,b)\|p(z|a,b)) && \text{C.2 \& (C.5)} \tag{5.32}\\
&\geq \mathcal{L}_{\mathrm{M_a}} && \tag{5.33}
\end{align}$$

The result from Eq. (5.33) can now be applied to the log-likelihood of the VoI:

$$\begin{align}
\mathrm{L_M} &= \mathrm{L_{M_a}} + \mathrm{L_{M_b}} && \tag{5.34}\\
&= \log(p(a|b)) + \log(p(b|a)) && \tag{5.35}\\
&= \mathcal{L}_{\mathrm{M_a}} + \mathcal{L}_{\mathrm{M_b}} + 2\,\mathrm{D_{KL}}(q(z|a,b)\|p(z|a,b)) && \text{Eq. (5.32)} \tag{5.36}\\
&\geq \mathcal{L}_{\mathrm{M_a}} + \mathcal{L}_{\mathrm{M_b}}. && \tag{5.37}
\end{align}$$

The sum of ELBOs is used to compare the distributions against each other but lacks a prior distribution, which is needed to regularize the latent space. However, one can subtract any value from the ELBO, and it still remains an ELBO to the marginal log-likelihood. Therefore, Suzuki et al. [Suz+17] introduced the KLD between the bi-modal approximator and the prior as follows:

$$\begin{align}
\mathcal{L}_{\mathrm{M_a}}+\mathcal{L}_{\mathrm{M_b}} &= \dots && \text{B.7} \tag{5.38}\\
&= \mathrm{E}_{q(z|a,b)}\log(p(a|z)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|b)) && \tag{5.39}\\
&\quad + \mathrm{E}_{q(z|a,b)}\log(p(b|z)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|a)) && \text{C.2} \tag{5.40}\\
&= \mathrm{E}_{q(z|a,b)}\log(p(a|z)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|b)) && \tag{5.41}\\
&\quad + \mathrm{E}_{q(z|a,b)}\log(p(b|z)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|a)) && \tag{5.42}\\
&\quad + \mathrm{D_{KL}}(q(z|a,b)\|p(z)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z)) && \text{add 0} \tag{5.43}\\
&= \mathcal{L}_{\mathrm{J}} - \mathrm{D_{KL}}(q(z|a,b)\|p(z|b)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|a)) && \tag{5.44}\\
&\quad + \mathrm{D_{KL}}(q(z|a,b)\|p(z)) && \text{(5.16)} \tag{5.45}\\
&\geq \mathcal{L}_{\mathrm{J}} - \mathrm{D_{KL}}(q(z|a,b)\|p(z|b)) - \mathrm{D_{KL}}(q(z|a,b)\|p(z|a)) && \tag{5.46}\\
&=: \mathcal{L}_{\mathrm{M}}. && \tag{5.47}
\end{align}$$

Concerning Eq. (5.17), the following objective can be identified:

$$\mathcal{L}_{\mathrm{M}} = \mathcal{L}_{\mathrm{J}} - \underbrace{\mathrm{D_{KL}}(q_{\phi_{\mathrm{ab}}}(z|a,b)\|q_{\phi_{\mathrm{b}}}(z|b))}_{\text{uni-modal PDF fitting of encoder b}} - \underbrace{\mathrm{D_{KL}}(q_{\phi_{\mathrm{ab}}}(z|a,b)\|q_{\phi_{\mathrm{a}}}(z|a))}_{\text{uni-modal PDF fitting of encoder a}}. \tag{5.48}$$

Equation (5.48) consists of the ELBO for the JMMVAE and two regularization terms, one for each encoder $q_{\phi_{\mathrm{a}}}$ and $q_{\phi_{\mathrm{b}}}$, that match the distributions between the uni- and bi-modal encoder.

5.1.2.2 Discussion

Figure 5.3 shows the JMMVAE as a DGM approach. The introduced regularization terms in Eq. (5.48) by Suzuki et al. [Suz+17] try to learn a representation for the uni-modal encoders q_{ϕ_*} by covering the representation of the bi-modal encoder. This is a valid approach for training the uni-modal encoder networks, but it comes with a decisive drawback. Because the uni-modal encoders q_{ϕ_*} learn to match distributions, their embeddings are not compared against a reconstruction. This makes even the bi-modal exchange questionable because it is undefined during training if the locations of the uni-modal encoders' embeddings already describe some other embeddings from the bi-modal encoder.

A decoding of the uni-modal encoders' embeddings may lead to artifacts because the uni-modal encoders q_{ϕ_*} blindly adopt their parametrization during training from $q_{\phi_{\mathrm{ab}}}$. This is not an issue if the information content of each modality's sample is congruent. However, it becomes an issue when observation ambiguities occur, which is explained in additional detail in Section 5.3 and recapped in Section 5.3.2.3.

5.2 M^{2}VAE

The author introduces the multi-modal variational autoencoder (M^{2}VAE) in this section. By successively applying logarithmic and Bayes rules, the ELBO for the multi-modal variational autoencoder (M^{2}VAE) is derived as follows: First, given the independent set of observable modalities $\mathcal{M} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots\}$, its marginal log-likelihood $\log p(\mathcal{M}) =: \mathrm{L}_{\mathrm{M}^2}$ is multiplied by the cardinality of the set as the neutral element $1 = {}^{|\mathcal{M}|}/_{|\mathcal{M}|}$. Second, by applying the logarithmic multiplication rule, the nominator is written as the argument's exponent. Third, the Bayes rule is applied to each term concerning the remaining observable modalities to derive their conditionals. Furthermore, the derivation technique is demonstrated in a bi-modal (see Section 5.2.1) and tri-modal (see Section 5.2.2) case to compare the M^{2}VAE against the JMMVAE and illustrate its advantages. Section 5.2.3 is used to extend this derivation to an arbitrary large set of modalities $\mathcal{M}$. Finally, a way to implement the M^{2}VAE as a DNN is proposed in Section 5.2.4 by the author.

5.2.1 Derivation of the Bi-Modal M²VAE

The M²VAE training objective in the bi-modal case is derived in this section.

5.2.1.1 The Variational Lower Bound

Excessively applying the scheme until the convergence of the mathematical expression leads to the following expression for the bi-modal set $\mathcal{M} = \{\mathrm{a}, \mathrm{b}\}$:

$$\begin{aligned}
\mathrm{L}_{\mathrm{M}^2} &= \log(p(a,b)) && && (5.49)\\
&= {}^2\!/\!_2 \log(p(a,b)) && \text{mul. by 1} && (5.50)\\
&= {}^1\!/\!_2 \log\left(p(a,b)^2\right) && \text{Eq. (C.4)} && (5.51)\\
&= {}^1\!/\!_2 \log(p(a,b)p(a,b)) && && (5.52)\\
&= {}^1\!/\!_2 \log(p(b)p(a|b)p(b|a)p(a)) && \text{Eq. (C.1)} && (5.53)\\
&= {}^1\!/\!_2 (\log(p(a)) + \log(p(b|a)) + \log(p(a|b)) + \log(p(b))) && \text{Eq. (C.3)} && (5.54)\\
&= {}^1\!/\!_2 (\mathrm{L}_\mathrm{a} + \underbrace{\mathrm{L}_{\mathrm{M}_\mathrm{a}} + \mathrm{L}_{\mathrm{M}_\mathrm{b}}}_{\mathrm{VoI}} + \mathrm{L}_\mathrm{b}) && && (5.55)
\end{aligned}$$

Equation (5.55) describes the M²VAE's log-likelihood $\mathrm{L}_{\mathrm{M}^2}$ as a weighted sum of log-likelihoods. L_a and L_b express the log-likelihoods of the common VAE for each modality. $\mathrm{L}_{\mathrm{M}_\mathrm{a}} + \mathrm{L}_{\mathrm{M}_\mathrm{b}}$ describe the variation of information (VoI) between the modalities, which is also the starting point of the JMMVAE's training objective. Therefore, the weighted sum from Eq. (5.55) can be rewritten as the sum of ELBO inequalities of each marginal L_a, L_b, and the conditionals $\mathrm{L}_\mathrm{M} = \mathrm{L}_{\mathrm{M}_\mathrm{a}} + \mathrm{L}_{\mathrm{M}_\mathrm{b}}$:

$$\begin{aligned}
\mathrm{L}_{\mathrm{M}^2} &\geq {}^1\!/\!_2 (\mathcal{L}_\mathrm{a} + \mathcal{L}_{\mathrm{M}_\mathrm{a}} + \mathcal{L}_{\mathrm{M}_\mathrm{b}} + \mathcal{L}_\mathrm{b}) && \text{(B.12) \& (5.37)} && (5.56)\\
&\geq {}^1\!/\!_2 (\mathcal{L}_\mathrm{a} + \mathcal{L}_\mathrm{M} + \mathcal{L}_\mathrm{b}) && \text{Eq. (5.48)} && (5.57)\\
&:= \mathcal{L}_{\mathrm{M}^2} && && (5.58)
\end{aligned}$$

Equation (5.57) describes the M²VAE's ELBO $\mathcal{L}_{\mathrm{M}^2}$ as a weighted sum of ELBOs. $\mathcal{L}_\mathrm{a}$ and $\mathcal{L}_\mathrm{b}$ express the ELBO of the common VAE for each modality. $\mathcal{L}_\mathrm{M}$ is the JMMVAE's ELBO that captures both modalities together.

5.2.1.2 Approximating Inference (i.e., rewriting $\mathcal{L}_{\mathrm{M}^2}$)

The ELBOs from Eq. (5.57) can now substituted with the corresponding expressions from the former sections:

$$\begin{aligned}
2\mathcal{L}_{\mathrm{M}^2} &= \mathcal{L}_\mathrm{a} + \mathcal{L}_\mathrm{M} + \mathcal{L}_\mathrm{b} && && (5.59)\\
&= -\mathrm{D}_{\mathrm{KL}}(q(z|a)\|p(z)) + \mathrm{E}_{q(z|a)} \log(p(a|z)) && \text{Eq. (B.16)} && (5.60)
\end{aligned}$$

$$
\begin{aligned}
&- \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z)) && \text{Eq. (5.16)} && (5.61)\\
&+ \mathrm{E}_{q(z|a,b)} \log(p(a|z)) + \mathrm{E}_{q(z|a,b)} \log(p(b|z)) && \text{Eq. (5.16)} && (5.62)\\
&- \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|b)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a)) && \text{Eq. (5.46)} && (5.63)\\
&- \mathrm{D}_{\mathrm{KL}}(q(z|b)\|p(z)) + \mathrm{E}_{q(z|b)} \log(p(b|z)) && \text{Eq. (B.16)} && (5.64)
\end{aligned}
$$

After applying the corresponding encoder networks, the formula becomes an objective for a DNN, that can be written as follows:

$$
\begin{aligned}
2\mathcal{L}_{\mathrm{M}^2} &= \mathcal{L}_{\mathrm{a}} + \mathcal{L}_{\mathrm{M}} + \mathcal{L}_{\mathrm{b}} && && (5.65)\\
&= - \mathrm{D}_{\mathrm{KL}}(q_{\phi_{\mathrm{a}}}(z|a)\|p(z)) + \mathrm{E}_{q_{\phi_{\mathrm{a}}}(z|a)} \log(p_{\theta_{\mathrm{a}}}(a|z)) && \text{Eq. (B.17)} && (5.66)\\
&\quad - \mathrm{D}_{\mathrm{KL}}(q_{\phi_{\mathrm{ab}}}(z|a,b)\|p(z)) && \text{Eq. (5.17)} && (5.67)\\
&\quad + \mathrm{E}_{q_{\phi_{\mathrm{ab}}}(z|a,b)} \log(p_{\theta_{\mathrm{a}}}(a|z)) + \mathrm{E}_{q_{\phi_{\mathrm{ab}}}(z|a,b)} \log(p_{\theta_{\mathrm{b}}}(b|z)) && \text{Eq. (5.17)} && (5.68)\\
&\quad - \mathrm{D}_{\mathrm{KL}}(q_{\phi_{\mathrm{ab}}}(z|a,b)\|q_{\phi_b}(z|b)) - \mathrm{D}_{\mathrm{KL}}(q_{\phi_{\mathrm{ab}}}(z|a,b)\|q_{\phi_{\mathrm{a}}}(z|a)) && \text{Eq. (5.48)} && (5.69)\\
&\quad - \mathrm{D}_{\mathrm{KL}}(q_{\phi_{\mathrm{b}}}(z|b)\|p(z)) + \mathrm{E}_{q_{\phi_b}(z|b)} \log(p_{\theta_{\mathrm{b}}}(b|z)) && \text{Eq. (B.17)} && (5.70)
\end{aligned}
$$

5.2.1.3 Discussion

By investigating every line of the formula, the following properties can be identified: Equations (5.67) and (5.68) represent the JVAE loss derived from the joint probability. Equation (5.69) adds the KLDs losses introduced by Suzuki et al. [Suz+17]. It introduces the KLD regularization that brings the posterior distribution of an uni-modal encoder close to the distribution of the bi-modal case. The drawback of this approach was discussed in Section 5.1.2.2. However, the additional lines in the M²VAE (i.e., Eq. (5.66) and (5.70)) introduce the regularization of the uni-modal encoders concerning the common prior plus the reconstruction loss. The support that the uni-modal distribution does not deviate to much from the common prior is given by the regularizer and the remaining statistics in the latent space are shaped by the reconstruction term. The additional reconstruction loss is very important because it supports the uni-modal encoder networks with feedback about their embeddings. Figure 5.3 comprises the evolution and differences of the bi-modal approach for a JVAE, JMMVAE, and the proposed M²VAE.

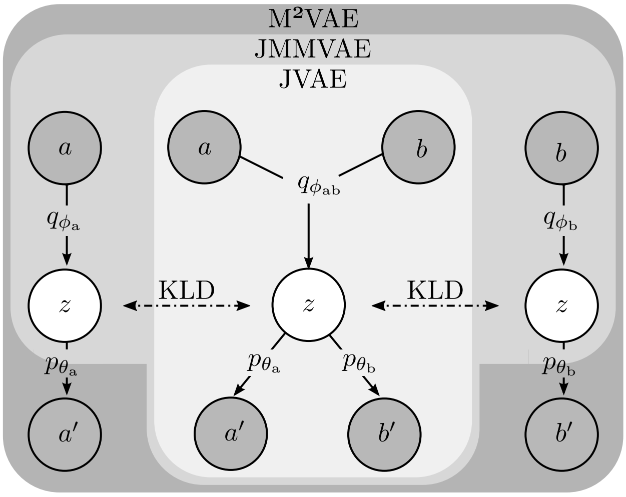


Figure 5.3: Evolution from the bi-modal JVAE over the JMMVAE to the proposed M²VAE. This is not an architectural choice but evolves naturally from the marginal joint log-likelihood.

5.2.2 Extension to three Modalities

It should be clear that both approaches, the proposed M²VAE and the JMMVAE by Suzuki et al. [Suz+17], can be extended to multiple modalities. In the following, an example of three modalities $\mathcal{M} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ is given.

First, the conditional likelihood of one modality is investigated, which shortens the derivation in the remaining sections:

$$
\begin{aligned}
p(a|b,c) &= \frac{p(a,b,c,z)}{p(a,b,c,z)} \frac{p(a,b,c)}{p(b,c)} && \text{mul. by 1 \& Eq. (C.2)} \qquad (5.71)\\
&= \frac{p(z,a|b,c)}{p(z|a,b,c)} && \text{Eq. (C.2)} \qquad (5.72)\\
&= \frac{1}{p(z|a,b,c)} p(z,a|b,c) && \text{reo.} \qquad (5.73)\\
&= \frac{1}{p(z|a,b,c)} \frac{p(z,a,b,c)}{p(b,c)} && \text{Eq. (C.2)} \qquad (5.74)\\
&= \frac{1}{p(z|a,b,c)} \frac{p(a,b,c|z)p(z)}{p(b,c)} && \text{Eq. (C.7)} \qquad (5.75)\\
&= \frac{1}{p(z|a,b,c)} \frac{p(a|z)p(b,c|z)p(z)}{p(b,c)} && \text{Eq. (C.2)} \qquad (5.76)\\
&= \frac{1}{p(z|a,b,c)} \frac{p(a|z)p(z|c,b)\frac{p(c,b)}{p(z)}p(z)}{p(b,c)} && \text{Eq. (C.2)} \qquad (5.77)
\end{aligned}
$$

$$= \frac{p(a|z)p(z|c,b)}{p(z|a,b,c)} \tag{5.78}$$

The log-likelihood can then be rewritten to maintain the ELBO as:

$$\begin{aligned}
\log(p(a|b,c)) &= \sum_z q(z|a,b,c)\log\left(\frac{p(z,a|b,c)}{q(z|a,b,c)}\right) && (5.79)\\
&\quad + \sum_z q(z|a,b,c)\log\left(\frac{q(z|a,b,c)}{p(z|a,b,c)}\right) && (5.80)\\
&= \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}} + \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|a,b,c)) && (5.81)\\
&\geq \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}}, && (5.82)
\end{aligned}$$

while $\mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}}$ represents the ELBO of the conditional log-likelihood.

5.2.2.1 JMMVAE for Three Modalities

The log-likelihood of the VoI between three distributions can be written as follows:

$$\begin{aligned}
\mathrm{L}_{3\mathrm{M}} &= \log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|b,c)) && (5.83)\\
&= \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{b}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{c}} + 3\,\mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|a,b,c)) \qquad \text{Eq. (5.81)} && (5.84)\\
&\geq \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{b}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{c}} && (5.85)
\end{aligned}$$

Following the derivation scheme by Suzuki et al. [Suz+17], the combined ELBO from Eq. (5.85) can be rewritten as

$$\begin{aligned}
\mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{a}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{b}} + \mathcal{L}_{\widetilde{\mathrm{M}}_\mathrm{c}} &= \mathrm{E}_{q(z|a,b,c)}\log(p(a|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|c,b)) && (5.86)\\
&\quad + \mathrm{E}_{q(z|a,b,c)}\log(p(b|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|a,c)) && (5.87)\\
&\quad + \mathrm{E}_{q(z|a,b,c)}\log(p(c|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|a,b)) && (5.88)\\
&\geq \mathcal{L}_{\widetilde{\mathrm{J}}} - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|b,c)) && (5.89)\\
&\quad - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|a,c)) && (5.90)\\
&\quad - \mathrm{D}_{\mathrm{KL}}(q(z|a,b,c)\|p(z|b,c)) && (5.91)\\
&:= \mathcal{L}_{\widetilde{\mathrm{M}}} && (5.92)
\end{aligned}$$

$\mathcal{L}_{\widetilde{\mathrm{J}}}$ is the joint ELBO of a joint log-likelihood distribution with three modalities (i.e., a, b, and c). The derivation was analog to that in subsection 5.1.1. A further step would be the application of encoder and decoder networks to form the training objective, which was neglected for the sake of brevity.

The following properties can be identified by investigating the previous equations: There are common reconstruction terms (E) for each decoder $p(\cdot|z)$ concerning the full multi-modal encoder $q(z|a,b,c)$. The KLD terms show an additional drawback of the VoI approach. As before, these regularizers tend to make the encoders' distributions match each other, but now, only pairwise encoders (e.g., $q(z|a,b)$) remain, and thus, uni-modal encoders are neglected. In conclusion, this means that if one has M modalities in a setup, then only the multi-modal encoders that cover either all M or $M-1$ modalities can be trained.

5.2.2.2 Proposed M²VAE for three Modalities

The derivation from the joint log-likelihood can be written analogously:

$$\begin{aligned}
\log(p(a,b,c)) &= {}^3\!/\!_3 \log(p(a,b,c)) && (5.93)\\
&= {}^1\!/\!_3 \log\Big(p(a,b,c)^3\Big) && (5.94)\\
&= {}^1\!/\!_3 \log(p(a,b,c)p(a,b,c)p(a,b,c)) && (5.95)\\
&= {}^1\!/\!_3 \log(p(a,b)p(b,c)p(a,c)p(a|b,c)p(b|a,c)p(c|a,b)) && (5.96)\\
&= {}^1\!/\!_3 (\log(p(a,b)) + \log(p(b,c)) + \log(p(a,c)) && (5.97)\\
&\qquad + \log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|a,b))) && (5.98)\\
&= {}^1\!/\!_3 ({}^2\!/\!_2 (\log(p(a,b)) + \log(p(b,c)) + \log(p(a,c))) && (5.99)\\
&\qquad\qquad + \log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|a,b))) && (5.100)\\
&= {}^1\!/\!_6 \Big(\log\Big(p(a,b)^2\Big) + \log\Big(p(b,c)^2\Big) + \log\Big(p(a,c)^2\Big)\Big) && (5.101)\\
&\quad + {}^1\!/\!_3 (\log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|a,b))) && (5.102)\\
&= {}^1\!/\!_6 (\log(p(a)p(b)p(a|b)p(b|a)) + \log(p(c)p(b)p(c|b)p(b|c)) && (5.103)\\
&\qquad + \log(p(a)p(c)p(a|c)p(c|a))) && (5.104)\\
&\quad + {}^1\!/\!_3 (\log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|a,b))) && (5.105)\\
&= {}^1\!/\!_6 (\log(p(a|b)) + \log(p(b|a)) + \log(p(c|b)) && (5.106)\\
&\qquad + \log(p(b|c)) + \log(p(a|c)) + \log(p(c|a))) && (5.107)\\
&\quad + {}^1\!/\!_3 (\log(p(a)) + \log(p(b)) + \log(p(c)) && (5.108)\\
&\qquad + \log(p(a|b,c)) + \log(p(b|a,c)) + \log(p(c|a,b))) && (5.109)
\end{aligned}$$

All previously mentioned equations can now be applied in a straight forward fashion to derive the ELBO for the tri-modal marginal log-likelihood. As one can imagine, the above equation results in a highly complex objective but with the big advantage of respecting all permutations of modalities. The further derivation of the training objective is neglected for the sake of brevity, and it is covered by a general expression in the next Section 5.2.3. In conclusion, and unlike the JMMVAE, one can now train encoder networks in a multi-modal setup that covers all 1 to M modality combinations.

5.2.3 Derivation of a General Expression for an Arbitrary Number of Modalities

If the derivation is applied to the log-likelihood $\mathrm{L}_{\mathrm{M}^2}$ of an arbitrary multi-modal set $\mathcal{M}$, then it can be shown that it results in a recursive form consisting of JMMVAE and M^{2}VAE log-likelihood terms. By comprising the applied steps to derive $\mathcal{L}_{\mathrm{M}^2}$ from the former section and successively applying logarithmic and Bayes rules, the ELBO can be derived as follows: First, given the independent set of observable modalities $\mathcal{M} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots\}$, its marginal log-likelihood $\log p(\mathcal{M}) =: \mathrm{L}_{\mathrm{M}^2{}_{\mathcal{M}}}$ is multiplied by the cardinality of the set as the neutral element $1 = {}^{|\mathcal{M}|}/_{|\mathcal{M}|}$. Second, when applying logarithm multiplication rule, the nominator is written as the argument's exponent. Third, the Bayes rule is applied to each term concerning the remaining observable modalities to derive their conditionals. This procedure results in the following expression:

$$\mathrm{L}_{\mathrm{M}^2{}_{\mathcal{M}}} = \log p(\mathcal{M}) \overset{\text{mul. 1}}{=} \frac{|\mathcal{M}|}{|\mathcal{M}|} \log p(\mathcal{M}) \overset{\text{log.\_mul.}}{=} \frac{1}{|\mathcal{M}|} \log p(\mathcal{M})^{|\mathcal{M}|} \tag{5.110}$$

$$\overset{\text{Bayes}}{=} \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \log p(\mathcal{M} \setminus m) p(m|\mathcal{M} \setminus m) \tag{5.111}$$

$$\overset{\text{log.\_add}}{=} \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \log p(\mathcal{M} \setminus m) + \log p(m|\mathcal{M} \setminus m). \tag{5.112}$$

The expression $\sum_{m \in \mathcal{M}} \log p(m|\mathcal{M} \setminus m)$ is the general form of the marginal log-likelihood for the variation of information (VoI), as introduced by Suzuki et al. [Suz+17] for the JMMVAE, for any set $\mathcal{M}$. Thus, it can be directly substituted with $\mathrm{L}_{\mathrm{M}_{\mathcal{M}}}$. The expression $\sum_{m \in \mathcal{M}} \log p(\mathcal{M} \setminus m)$ is the combination of all joint log-likelihoods of the subsets of $\mathcal{M}$ that have one element less than their superset. Therefore, this term can be rewritten as follows:

$$\sum_{m \in \mathcal{M}} \log p(\mathcal{M} \setminus m) = \sum_{\widetilde{m} \in \widetilde{\mathcal{M}}} \log p(\widetilde{m}) \tag{5.113}$$

with $\widetilde{\mathcal{M}} = \{m | m \in \mathcal{P}(\mathcal{M}), |m| = |\mathcal{M}| - 1\}$ Finally, $\log p(\widetilde{m})$ can be substituted by $\mathrm{L}_{\mathrm{M}^2{}_{\widetilde{m}}}$ sacrificing generality. However, it is worth noticing that substitution stops at the end of recursion, and therefore, all final expressions $\log p(\widetilde{m})\ \forall\ |\widetilde{m}| \equiv 1$ remains.

This results in the following final recursive log-likelihood expression from which the ELBO can be directly derived:

$$\mathrm{L}_{\mathrm{M}^2{}_{\mathcal{M}}} = \frac{1}{|\mathcal{M}|}\left(\mathrm{L}_{\mathrm{M}_{\mathcal{M}}} + \sum_{\widetilde{m}\in\widetilde{\mathcal{M}}} \mathrm{L}_{\mathrm{M}^2{}_{\widetilde{m}}}\right) \geq \frac{1}{|\mathcal{M}|}\left(\mathcal{L}_{\mathrm{M}_{\mathcal{M}}} + \sum_{\widetilde{m}\in\widetilde{\mathcal{M}}} \mathcal{L}_{\mathrm{M}^2{}_{\widetilde{m}}}\right) =: \mathcal{L}_{\mathrm{M}^2{}_{\mathcal{M}}}. \quad (5.114)$$

5.2.4 Realization as Deep Neural Network

The implementation of the objective from Eq. (5.114) as a DNN is proposed in this section. First, it is worth noticing that Suzuki et al. [Suz+17] and Vedantam et al. [Ved+17] argue that the training of a full multi-modal VAE, such as the M²VAE, is intractable because of the exponentially growing number of modality combinations. Investigating Eq. (5.114) reveals that one has to train $2^{|\mathcal{M}|}-1$ modality subsets of encoder networks (i.e., $2^{|\mathcal{M}|}-1$ times $f_{\text{enc.}}$, $f_{\boldsymbol{\mu}}$, $f_{\boldsymbol{\sigma}}$) plus $|\mathcal{M}|$ decoder networks (i.e., $|\mathcal{M}|$ times $f_{\text{dec.}}$) for a set $\mathcal{M}$ of modalities.

By using a DNN training technique called weight-sharing, one can link and reuse layers inside a DNN. From another DNN technique, called transfer-learning (see Goodfellow et al. [Goo+16]), it is known that all layers but the ultimate layer L act as feature extractors. Only the last layer performs the desired classification or regression task. These approaches can be applied to the common VAE framework because the encoder network $f_{\text{enc.}}$ can be seen as a feature extractor while $f_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\sigma}}$ perform regression in the latent space.

Adopting these techniques, the burden of training $2^{|\mathcal{M}|}-1$ different encoder networks can be reduced to train $|\mathcal{M}|$ encoder networks $f_{\text{enc.}}$. There are still $2^{|\mathcal{M}|}-1$ different regression networks $f_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\sigma}}$, that are, however, quite tractable because they are commonly realized as single linear layers of low dimensionality.

The previously mentioned architecture approach sounds reasonable in the first place, but one might question whether the linear layers used for the fusion are sufficient. Firstly, VAEs perform a regression task in the latent space, which justifies a linear activation function in general. Secondly, VAEs project generative factors into a linear separable latent space (see Section 3.3.4.3). Therefore, all combinations of these factors can be expressed via linear combinations. This makes linear layers, in the particular case of the M²VAE, sufficient for sensor fusion in the latent space.

Although the linear behavior of the latent space does not always hold (see Fig. 3.6 vs. 3.7) such that one might introduce further non-linear networks for sensor fusion. However, these networks can be very shallow because they only have to learn a non-linear combination of the extracted features. The results of later evaluations will show whether these additional networks improve the M²VAE's performance.

Additional weightings are introduced by the author following the concept of βVAE by [Hig+16]) and JVAE by Suzuki et al. [Suz+17]. These additional weightings are scalar and multiplicative factors that are introduced to the prior and mutual losses in the latent space's objective during training. The βVAE concept is applied to the prior losses as follows:

$$\mathrm{D}_{\mathrm{KL}}\Big(p\big(\widetilde{\mathcal{M}}\big)\|p(\mathbf{z})\Big) \rightarrow \beta\,\mathrm{D}_{\mathrm{KL}}\Big(p\big(\widetilde{\mathcal{M}}\big)\|p(\mathbf{z})\Big)\forall\widetilde{\mathcal{M}} \subseteq \mathcal{M}. \tag{5.115}$$

Higgins et al. [Hig+16] recommended to use a normalized β that depends on the data and latent dimensionality: $\beta_{\mathrm{norm}} = \beta\,{}^{D_m}\!/_{D_z} \approx 10^{-2} \dots 10^{-3}$. Furthermore, the JVAE concept introduces a weighting of the mutual losses, which can be generalized for the M²VAE as follows:

$$\mathrm{D}_{\mathrm{KL}}\Big(p\big(\widetilde{\mathcal{M}}\big)\|p\big(\widetilde{\mathcal{M}}\setminus m\big)\Big) \rightarrow \beta_{\mathrm{M}}\,\mathrm{D}_{\mathrm{KL}}\Big(p\big(\widetilde{\mathcal{M}}\big)\|p\big(\widetilde{\mathcal{M}}\setminus m\big)\Big)\forall\widetilde{\mathcal{M}} \subseteq \mathcal{M}, \tag{5.116}$$

while β_{M} denotes that the mutual β. β_{norm} has the ability to disentangle the latent space, β_{M} balances the impact of the mutual losses on the latent space embeddings. However, while the value of choice for β_{norm} is already determined, later evaluations will reveal the impact of β_{M} on the M²VAE's performance.

Figure 5.4 shows the proposed architecture approach for two modalities, as derived in Section 5.2.1.2. The network configuration comprises the three encoder and two decoder networks from equations (5.65) to (5.70).

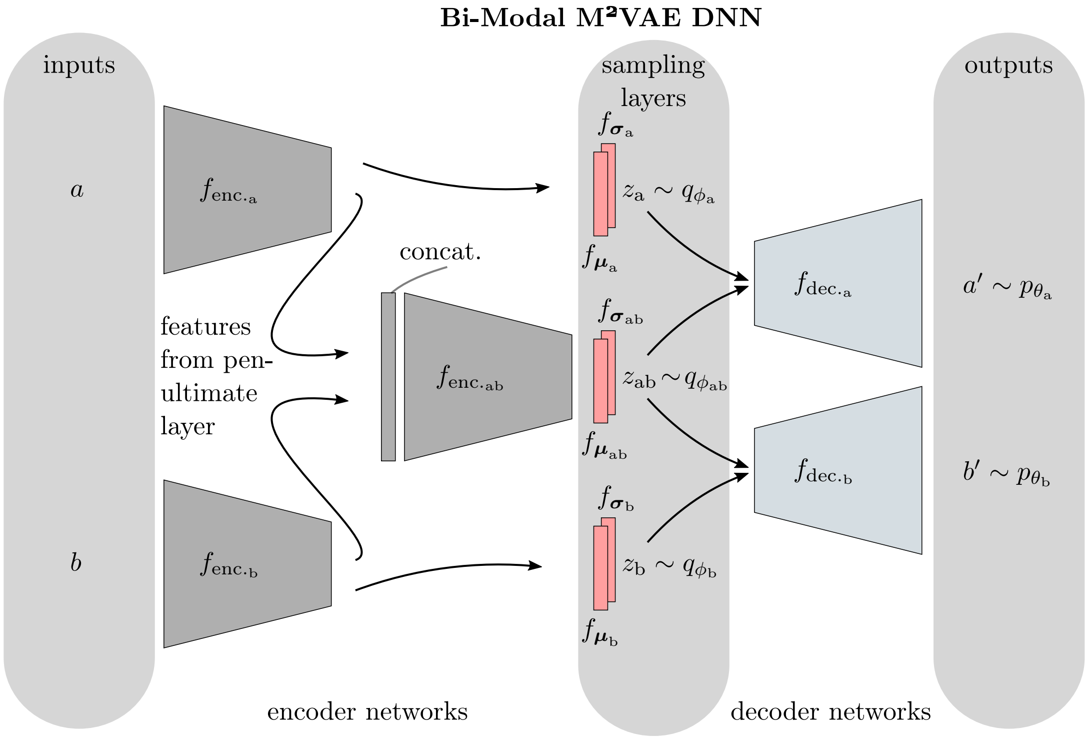
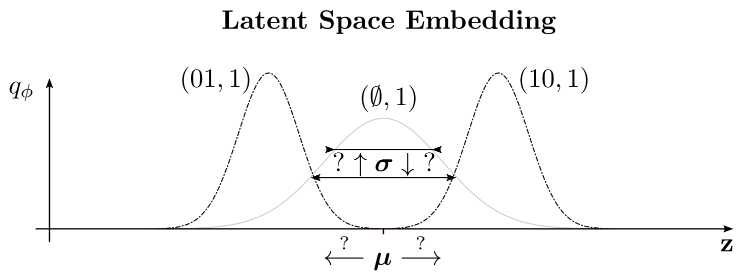

Bi-Modal M²VAE DNN
inputs
a
b
$f_{\mathrm{enc.a}}$
$f_{\mathrm{enc.b}}$
features from penultimate layer
concat.
$f_{\mathrm{enc.ab}}$
sampling layers
$f_{\boldsymbol{\sigma}_{\mathrm{a}}}$
$z_{\mathrm{a}} \sim q_{\phi_{\mathrm{a}}}$
$f_{\boldsymbol{\mu}_{\mathrm{a}}}$
$f_{\boldsymbol{\sigma}_{\mathrm{ab}}}$
$z_{\mathrm{ab}} \sim q_{\phi_{\mathrm{ab}}}$
$f_{\boldsymbol{\mu}_{\mathrm{ab}}}$
$f_{\boldsymbol{\sigma}_{\mathrm{b}}}$
$z_{\mathrm{b}} \sim q_{\phi_{\mathrm{b}}}$
$f_{\boldsymbol{\mu}_{\mathrm{b}}}$
$f_{\mathrm{dec.a}}$
$f_{\mathrm{dec.b}}$
outputs
$a' \sim p_{\theta_{\mathrm{a}}}$
$b' \sim p_{\theta_{\mathrm{b}}}$
encoder networks
decoder networks

Figure 5.4: Realization of a bi-modal DNN following the proposed scheme.



Latent Space Embedding

q_ϕ
(01, 1)
(∅, 1)
(10, 1)
? ↑ σ ↓ ?
z
$\overset{?}{\longleftarrow} \mu \overset{?}{\longrightarrow}$

Figure 5.5: Qualitative depiction of a 1D latent space with three different embeddings. The example shows the embedding of a subset of the XOR gate logic with its input and output as two different modalities (i.e., $a \in (10, 01, \emptyset)$, $b \in (1, 1, 1)$, $a \times b = \{(10, 1), (01, 1), (\emptyset, 1)\}$, with $\emptyset$ denoting a modality drop-out (see Section 6.2.1 for the XOR data set).

5.3 Conscious vs. Unconscious M^{2}VAE

A part of the multi-modal VAE's objective is to minimize the KLD in the latent space between sets of modalities (e.g., (a, b) and (a)). Therefore, what the KLD actually calculates is of special interest when the VAE experiences ambiguities or mode-collapse[5] between two sets. As mentioned earlier in Section 3.3.2, the VAE's objective is to only compare sample-based distributions during the calculation of a loss over the mini-batch. It can never compare richer projections than these of sample-based mini-batches because this would require supervised guidance for samples that represent the same phenomenon.[6] Furthermore, for this group of samples that represent the same phenomena, one could then calculate the exact KLD. However, this endeavor is cumbersome because it would require a numerical calculation of the KLD via Monte Carlo sampling, which makes BP slow.

In the case of the VAE's objective, the expected value over all sample-based distributions is naïvely calculated (see Eq. (3.29)) without any consideration of the input modalities. For further demonstration, it is now assumed that two modalities are observed over a set of samples while one modality observers a constant phenomenon and the other observes a varying phenomenon. This is defined as observation am-

[5] Mode-collapse addresses the behavior of the modes of the sample-based posterior distribution, which may collapse to the same values. This varies from complete collapse (i.e., when all encoded and decoded samples are identical) to partial collapse (i.e., when most of the samples share some common properties). It depends on various factors like the architecture, training approach, or the complexity of the data set.

[6] This statement holds true, even in comparison to (semi-)supervised VAE (see Section 3.3.4.4) because these still train a per-sample distribution.

biguity and is depicted in Fig. 5.5 with part of the XOR gate logic as a bi-modal observation. Figure 5.5 depicts the central question of this section:

- How do the statistics of the latent embedding of a partial observation behave?
- Does the embedding of a partial observation render the embeddings of full observations statistically correct (e.g., by averaging the mean values and increasing the variance)?
- Does the ELBO of the partial observation respect the full observation during training?

Note to the reader: *The reader should be aware of the ambiguous nomenclature regarding the word "modal,", which might cause confusion in the next chapters. For the sake of consistency, however, the definition is adopted from related literature: The author of this thesis investigates multi-modal observations, while the term "modal" refers to "modality". In functional analysis, the terms "uni-modal," "bi-modal," ..., "multi-modal" refer to the number of "modes" (i.e., the number of maximums) of a function. The term "multi-variate," however, refers to the dimensional properties of a function.*

5.3.1 Comparison of Uni-Modal and Mixture Distribution

A M^{2}VAE experiences varying and a constant input in the particular case of Fig. 5.5. Any set of varying inputs $a = \{a_1, a_2, \ldots, a_K\}$ will be projected onto the latent space $z_{\mathrm{a}} = \{z_{\mathrm{a}_1}, z_{\mathrm{a}_2}, \ldots, z_{\mathrm{a}_K}\}$. For the constant inputs, the set of inputs $b = \{b_1, b_2, \ldots, b_K\}$ remains unchanged, which therefore, always leads to the same projection in the latent space $z_{\mathrm{b}} = \{z_{\mathrm{b}_1}, z_{\mathrm{b}_2}, \ldots, z_{\mathrm{b}_K}\}$ with $z_{\mathrm{b}_k} \equiv z_{\mathrm{b}_i} \forall i, k$. The bi-modal projection z_{ab} in the latent space will be congruent with z_{a} because $(a \times b)$ and a hold the same statistics.[7][8]

For further analysis, the conscious versus the unconscious case of calculating the multi-modality objective is discussed as follows: In the conscious or supervised case, the relations between all samples is known. Therefore, one can compare the full mixture density of a single phenomenon of z_{ab} with z_{a} to minimize their exact divergence. The VAE, however, is completely unconscious because it cannot derive any information about the membership or clustering of particular input samples. It can only calculate the loss per single sample and BP the error. Therefore, the mixture density becomes the equally weighted sum of all sample-based distributions.

In the following sections, the relation of the conscious case versus the unconscious batch case used in the VAE's objective in two scenarios is derived as follows: First,

[7] with × being the Cartesian product

[8] It is worth noting, that the observation of the constant b in $(a \times b)$ is not non-informative. In the depicted example, it serves as a correlator between all varying inputs of $\mathcal{A}$.

$\mathrm{D_{KL}}(p\|p_\mathrm{M})$ versus $\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p\|p_{\mathrm{M}_k})$ is compared in Section 5.3.1.1. Second, $\mathrm{D_{KL}}(p\|p_\mathrm{M})$ versus $\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p\|p_{\mathrm{M}_k})$ is compared in Section 5.3.1.2. Finally, the discussion and summary of the results is performed in Section 5.3.1.3. While the VAEs uses Gaussian distributions for the projection of inputs in the latent space, an MoG p_M is used to represent z_ab (i.e., conscious case), and a Gaussian p represents z_b (i.e., unconscious case) in the discussion.

5.3.1.1 Uni-Modal versus Mixture Distribution

First, the unconscious loss is written as part of the negative entropy over the ambiguous observation p and the mean over cross-entropies between the inputs:

$$\begin{aligned}\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p\|p_{\mathrm{M}_k}) &= \frac{1}{K}\sum_k^K \int p\log\frac{p}{p_{\mathrm{M}_k}} && (5.117)\\ &= \frac{1}{K}\sum_k^K \int p\log p - \frac{1}{K}\sum_k^K\int p\log p_{\mathrm{M}_k} && (5.118)\\ &= -\mathrm{H}[p] + \frac{1}{K}\sum_k^K \mathrm{H}[p,p_{\mathrm{M}_k}] && (5.119)\end{aligned}$$

While minimizing the unconscious objective, the entropy of the ambiguous input is encouraged to grow (e.g., the variance of a Gaussian distribution increases) due to the negation while the cross-entropy between the ambiguous input and all mixture components is decreased until both reach an equilibrium with $p \equiv q$. The same decomposition into entropy and cross-entropy is performed in the conscious case as follows:

$$\begin{aligned}\mathrm{D_{KL}}(p\|p_\mathrm{M}) &= \mathrm{D_{KL}}\left(p\middle\|\sum_k^K \lambda_k p_{\mathrm{M}_k}\right) && (5.120)\\ &= \int p\log\frac{p}{\sum_k^K \lambda_k p_{\mathrm{M}_k}} && (5.121)\\ &= \int p\log p - \int p\log\sum_k^K \lambda_k p_{\mathrm{M}_k} && (5.122)\\ &= -\mathrm{H}[p] - \int p\log\sum_k^K \lambda_k p_{\mathrm{M}_k} && (5.123)\\ &\le -\mathrm{H}[p] - \int p\sum_k^K \lambda_k \log p_{\mathrm{M}_k} \qquad \text{C.41} && (5.124)\\ &= -\mathrm{H}[p] - \sum_k^K \lambda_k\int p\log p_{\mathrm{M}_k} && (5.125)\end{aligned}$$

$$= -\,\mathrm{H}[p] + \frac{1}{K}\sum_{k}^{K}\mathrm{H}[p, p_{\mathrm{M}_k}] \qquad\qquad \lambda_k = \frac{1}{K}\forall k \tag{5.126}$$

Jensen's inequality is applied in Eq. (5.124) to narrow the mixture density by pulling the logarithm in the finite sum. Finally, the finite sum can be exchanged with the integral in Eq. (5.125) and written as cross-entropy in Eq. (5.126) It can be assumed, that all observations are equally distributed and weighted such that λ_k is no longer a categorical distribution but, rather, a constant with $\lambda_k = 1/K$. Thus, their associated sample-based distribution, as part of the mixture distribution, have a uniform weight, as applied in Eq. (5.126).

5.3.1.2 Mixture versus Uni-Modal Distribution

First, the unconscious loss is written as part of the mean over all negative entropies of the mixture distribution plus the mean over cross-entropies between the inputs:

$$\frac{1}{K}\sum_{k}^{K}\mathrm{D_{KL}}(p_{\mathrm{M}_k}\|p) = \frac{1}{K}\sum_{k}^{K}\int p_{\mathrm{M}_k}\log\frac{p_{\mathrm{M}_k}}{p} \tag{5.127}$$

$$= \frac{1}{K}\sum_{k}^{K}\int p_{\mathrm{M}_k}\log p_{\mathrm{M}_k} - \frac{1}{K}\sum_{k}^{K}\int p_{\mathrm{M}_k}\log p \tag{5.128}$$

$$= -\frac{1}{K}\sum_{k}^{K}\mathrm{H}[p_{\mathrm{M}_k}] + \frac{1}{K}\sum_{k}^{K}\mathrm{H}[p_{\mathrm{M}_k}, p] \qquad \text{C.19} \tag{5.129}$$

Similarly, as in the previous Section 5.3.1.1, the unconscious objective is given by the mean entropy over all mixture components. The mean entropy increases during minimization due to negation, while the cross-entropy between all mixture components and the ambiguous input is decreased. The conscious case uses the same decomposition into entropy and cross-entropy as follows:

$$\mathrm{D_{KL}}(p_{\mathrm{M}}\|p) = \mathrm{D_{KL}}\left(\sum_{k}^{K}\lambda_k p_{\mathrm{M}_k}\middle\| p\right) \tag{5.130}$$

$$= \int\sum_{k}^{K}\lambda_k p_{\mathrm{M}_k}\log\frac{p_{\mathrm{M}}}{p} \tag{5.131}$$

$$= \int\sum_{k}^{K}\lambda_k p_{\mathrm{M}_k}\log p_{\mathrm{M}} - \int\sum_{k}^{K}\lambda_k p_{\mathrm{M}_k}\log p \tag{5.132}$$

$$= \sum_{k}^{K}\lambda_k\int p_{\mathrm{M}_k}\log p_{\mathrm{M}} - \sum_{k}^{K}\lambda_k\int p_{\mathrm{M}_k}\log p \tag{5.133}$$

$$= -\sum_{k}^{K}\lambda_k\,\mathrm{H}[p_{\mathrm{M}_k}, p_{\mathrm{M}}] + \sum_{k}^{K}\lambda_k\,\mathrm{H}[p_{\mathrm{M}_k}, p] \qquad C.2.2 \tag{5.134}$$

$$= -\sum_k^K \lambda_k (\mathrm{H}[p_{\mathrm{M}_k}] + \mathrm{D}_{\mathrm{KL}}(p_{\mathrm{M}_k} \| p_{\mathrm{M}})) + \dots \qquad \text{C.19} \tag{5.135}$$

$$= -\sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}] - \sum_k^K \lambda_k \, \mathrm{D}_{\mathrm{KL}}(p_{\mathrm{M}_k} \| p_{\mathrm{M}}) + \dots \tag{5.136}$$

$$\leq -\sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}] + \sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}, p] \tag{5.137}$$

$$= -\frac{1}{K} \sum_k^K \mathrm{H}[p_{\mathrm{M}_k}] + \frac{1}{K} \sum_k^K \mathrm{H}[p_{\mathrm{M}_k}, p] \qquad \lambda_k = \frac{1}{K} \forall k \tag{5.138}$$

Equations (5.131) to (5.134) demonstrate the decomposition of the entropy $\mathrm{H}[p_{\mathrm{M}}]$ into the mean over all cross-entropies over its mixture components: $-\sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}]$. Furthermore, the cross-entropy is written as the sum of its entropy and KLD, from which the KLD is dropped in Eq. (5.137), because $-\mathrm{D}_{\mathrm{KL}} \leq 0$. Again, it is assumed in Eq. (5.138) that all observations are equally distributed and weighted.[9]

5.3.1.3 Discussion

The former sections reveal the inequalities summarized in Table 5.1 for any uni-modal and mixture density.

Table 5.1: Derived inequalities between the conscious and unconscious cases

Uni-Modal vs. Mixture (Section 5.3.1.1):	$\mathrm{D}_{\mathrm{KL}}(p \| p_{\mathrm{M}}) \leq \frac{1}{K} \sum_k^K \mathrm{D}_{\mathrm{KL}}(p \| p_{\mathrm{M}_k})$
Mixture vs. Uni-Modal (Section 5.3.1.2):	$\mathrm{D}_{\mathrm{KL}}(p_{\mathrm{M}} \| p) \leq \frac{1}{K} \sum_k^K \mathrm{D}_{\mathrm{KL}}(p_{\mathrm{M}_k} \| p)$

This is an intriguing observation because it reveals that any unconscious case is lower-bounded by the conscious case. That fact makes gradient-descent using BP, as it is applied in NNs and VAEs, suitable for multi-modal observations with ambiguities. Figure 5.6 illustrates and empirically proves that the former derivation holds for a mixture of Gaussians and a Gaussian distribution with one mode and variable standard deviation.

[9] It is worth noting that by applying Jensen's inequality to Eq. (5.132), another interesting fact can be derived: $\sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}, p_{\mathrm{M}}] \leq \sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}}, p_{\mathrm{M}_k}]$ (see Appendix C.5).

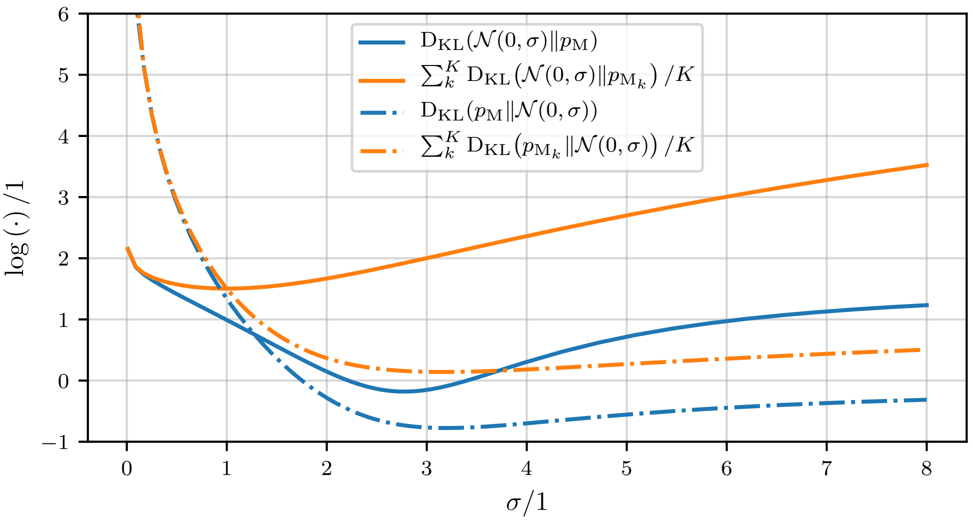


Figure 5.6: Kullback–Leibler divergences (KLDs) between a uni-modal Gaussian and a MoG distribution $p_{\mathrm{M}} = \frac{1}{2}p_{\mathrm{M}_1} + \frac{1}{2}p_{\mathrm{M}_2}$ with $p_{\mathrm{M}_k} = \mathcal{N}(\mu_k, \sigma_k)$ and $\mu_1 = -\mu_2 = 3., \sigma_1 = \sigma_2 = 1, K = 2$.

It can be seen, that the naïve calculation of the KLD performed by the VAE always lies above the true KLD. The figure also reveals that the calculations do not share the same minimum, which is further discussed in Section 5.3.2.

It is worth mentioning that the case from Section 5.3.1.2 can be generally applied to the M²VAE's optimization objective. Equation (5.69) shows that the KLD tends to have the form of $\mathrm{D_{KL}}(\text{proxy of the higher modality-set}\|\text{proxy of the lower modality-set})$. One can undoubtedly argue that many modalities always observe equally or more information than few modalities.[10] Although the "equal" case is of no interest because it only serves redundancy, the "more" case serves the investigations concerning ambiguities. The modes of ambiguous observations collapse in a lower modality-set while the higher modality-set can still explain the information by pushing the modes of the posterior apart (therefore, forming a mixture of Gaussians (MoG)).

[10] It is assumed that the system is causal and that no systematic errors exist.

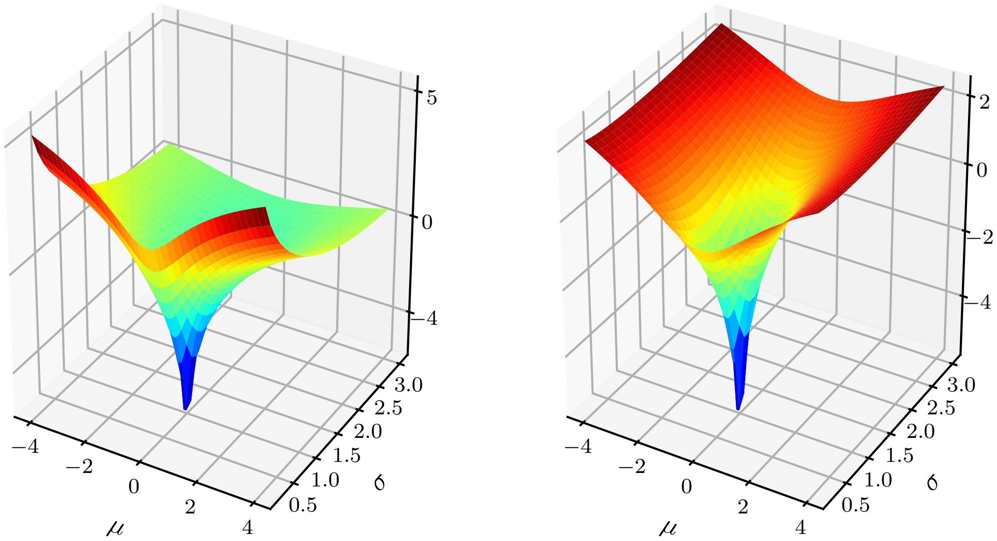


Figure 5.7: Comparison of the convex and asymmetric (which shows that KLD is no metric) error surfaces of the KL-divergence: $\ln(\mathrm{D}_{\mathrm{KL}}(\mathcal{N}(0,1)\|\mathcal{N}(\mu,\sigma)))$ (left), $\ln(\mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\mu,\sigma))\|\mathcal{N}(0,1))$ (right).

5.3.2 Evaluation of Convexity for Optimization

Another feature worth investigating is the convexity of the optimization. The fact that the conscious KLD is always a lower bound of the unconscious KLD was shown in the former section. Although this justifies minimization of the unconscious KLD, it is of high interest for optimization if the functions are convex and, if so, the convex points are congruent. Furthermore, if it can be shown that the arguments for the minimum values coincide, then optimizing the unconscious term leads to the same results as the conscious case.

For the sake of simplicity, it can be assumed that only the uni-modal function p is influenced by the KLD because the function p_{M} is held in place. It is worth noting that this assumption only holds true for the investigated use-case of multi-sensory VAEs that comprise a KLD loss on the latent distribution and a reconstruction loss between the input and the decoded output resulting from the latent embedding. The reasoning why the multi-modal function is pinned in the latent space is that there can be high reconstruction loss when the latent multi-modal distribution is moved. On the contrary the uni-modal function p can move freely in the area that is allocated by p_{M} because every latent embedding of p concerning p_{M} results in the same reconstruction as depicted in Fig. 5.8.

The figure shows a bi-sensory embedding of samples $\{a_1, a_2, a_3, a_4\}$ of modality a and $\{b_1, b_2, b_3\}$ of modality b with the observation tuples $\{(a_1, b_1), (a_2, b_2), (a_3, b_2)(a_4, b_3)\}$. The observations $\{a_1, a_2, a_3, a_4, b_1, b_3\}$ are encoded unambiguously to their embeddings, due to the fact that the observations may come from different modalities but are congruent concerning their information. The observation b_2 is ambiguous concerning a_2 and a_3. $a_1 - a_4$ cannot move because reconstruction loss would increase if the embeddings start to overlap, and it cannot become tighter because of the regularization. b_2 is constant for observations a_2 and a_3 and, therefore, has the possibility to instantiate into various possible latent embeddings.

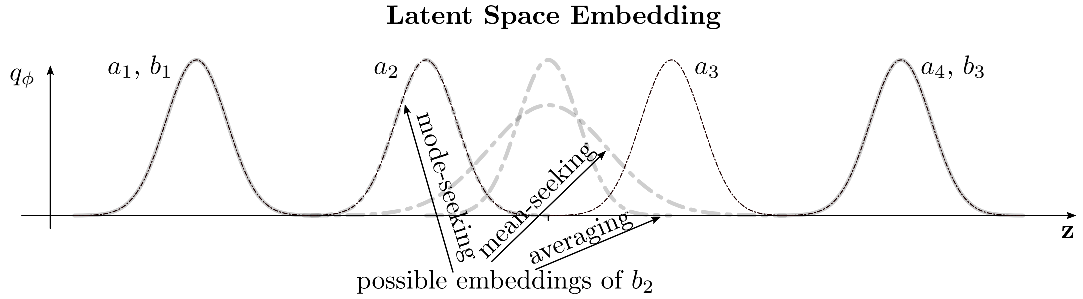


Figure 5.8: Possible latent embedding that questions the parametrization of an ambiguous observation.

In conclusion, only the two derivatives

$$\nabla_\theta \, \mathrm{D_{KL}}(p_\theta \| p_\mathrm{M}) \text{ and } \nabla_\theta \, \mathrm{D_{KL}}(p_\mathrm{M} \| p_\theta) \tag{5.139}$$

need to be investigated.

5.3.2.1 Investigation of Derivatives

First, the derivatives concerning to the two distributions in Section 5.3.1.1 are compared. The optima are only the same if the lhs. term is equal to the rhs. term. Thus, the author questions if

$$\nabla_\theta \frac{1}{K} \sum_k^K \mathrm{D_{KL}}(p_\theta \| p_{\mathrm{M}_k}) \overset{?}{=} \nabla_\theta \, \mathrm{D_{KL}}(p_\theta \| p_\mathrm{M}). \tag{5.140}$$

in this paragraph. Because the entropy terms are the same for both expressions, only the cross-entropy needs to be considered. Furthermore, the derivative can be exchanged with the integral using Leibniz's integral rule. This leads to

$$\sum_k^K \int \nabla_\theta p \log p_{\mathrm{M}_k} \neq \int \nabla_\theta p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{5.141}$$

(see Appendix B.8). From this equation, one cannot determine whether the gradients vanish for the same θ in general. Particularly the rhs. term has been studied in literature by Goodfellow et al. [Goo+16, p. 71], for instance. Because the term cannot be solved analytically, experiments show that the optimization of $\mathrm{D_{KL}}(p_\theta \| p_\mathrm{M})$ is mode-seeking if the modes of p_M are far apart and mean-seeking and, therefore, a convex optimization if the modes of p_M are close together. Intriguingly, the lhs. term is convex because of the following arguments: p_θ and p_{M_k} are particularly Gaussian in the VAE-case, which makes optimization in $\mathrm{D_{KL}}(p_\theta \| p_{\mathrm{M}_k})$ convex.[11] Furthermore, the sum of convex functions is, again, convex (see Appendix C.4 for convex definition). Therefore, $\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p_\theta \| p_{\mathrm{M}_k})$ is convex. □

Second, the derivatives concerning to the two distributions in Section 5.3.1.2 are compared. Again, only if the lhs. term is equal to the rhs. are the optima the same. Thus, the author questions if

$$\nabla_\theta \frac{1}{K} \sum_k^K \mathrm{D_{KL}}(p_{\mathrm{M}_k} \| p_\theta) \overset{?}{=} \nabla_\theta \, \mathrm{D_{KL}}(p_\mathrm{M} \| p_\theta). \tag{5.142}$$

in this paragraph. Luckily, only the KLD in Eq. (5.136), which does not depend on θ anyway, needs to be dropped to show that both terms share the same gradient:

$$-\frac{1}{K}\sum_k^K \mathrm{H}[p_{\mathrm{M}_k}] + \frac{1}{K}\sum_k^K \mathrm{H}[p_{\mathrm{M}_k}, p] = -\sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}] + \sum_k^K \lambda_k \, \mathrm{H}[p_{\mathrm{M}_k}, p] \tag{5.143}$$

for $\lambda_k = \frac{1}{K}$. As before, $\mathrm{D_{KL}}(p_{\mathrm{M}_k} \| p_\theta)$ is convex. It is worth noticing that the KLD in this manner is convex in general when both distributions are in the exponential family because it can be shown that the KLD becomes the Bregmann divergence with $\mathrm{D_{KL}}(p \| q_\theta) = \mathrm{D_B}(q_\theta \| p)$ [Bau+01].

Expressed differently, the fact that the sum of convex functions is again convex leads to the conclusion that $\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p_{\mathrm{M}_k} \| p_\theta)$ is convex. Because Eq. (5.143) reveals that the derivatives are the same as each other, $\mathrm{D_{KL}}(p_\mathrm{M} \| p_\theta)$ must be convex as well with the same optima.

Finally, the following conclusions can be drawn:

[11] The optimization of the KLD in this manner is not generally convex, even if both distributions are in the exponential family.

Table 5.2: Conclusion about the various forms of KLD that can occur in multisensory VAEs.

conscious KLD	
$\mathrm{D_{KL}}(p_\theta \| p_\mathrm{M})$	not convex in general
$\mathrm{D_{KL}}(p_\mathrm{M} \| p_\theta)$	convex in θ

unconscious KLD	
$\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p_\theta \| p_{\mathrm{M}_k})$	convex in θ
$\frac{1}{K}\sum_k^K \mathrm{D_{KL}}(p_{\mathrm{M}_k} \| p_\theta)$	convex in θ and share the same θ for the minimum value of $\mathrm{D_{KL}}(p_\mathrm{M} \| p_\theta)$

5.3.2.2 Exemplary Demonstration

Finally, the findings from Section 5.3.2.1 will be visualized via the KLD between a fixed MoG p_{M_*} and a variable Gaussian distribution p_θ. The distributions are as follows:

$$
\begin{aligned}
p_\theta &= \mathcal{N}(\mu, \sigma) && (5.144)\\
p_{\mathrm{M}_2} &= \mathcal{N}(-2., .3) + \mathcal{N}(2., .3) && (5.145)\\
p_{\mathrm{M}_1} &= \mathcal{N}(-1., .3) + \mathcal{N}(1., .3) && (5.146)\\
p_{\mathrm{M}_{05}} &= \mathcal{N}(-.5, .3) + \mathcal{N}(.5, .3) && (5.147)
\end{aligned}
$$

The choice of the values for the MoG demonstrates the findings and was found empirically. Figure 5.9 and 5.10 show the error surface that visualizes the KLD between the MoG variable Gaussian distributions in various cases.

Figure 5.9 shows the surfaces with the variable Gaussian distribution on the left side of the $\mathrm{D_{KL}}$ argument. The conscious case (upper row) reveals the two possible minima at each mode of the MoG. However, as soon as the modes of the MoG come closer together, the two minima become one. Thus, p_θ becomes mode-seeking (see Fig. 5.8), as it converges in one minimum (see Table 5.4).

In the unconscious case (i.e., lower row), the error surface only has one minimum, which lies exactly between in the center of the MoG. In this particular case, the $\mathrm{D_{KL}}$ performs an averaging (see Fig. 5.8) over the single parameters of the MoG. This can also be seen in Table 5.4, which shows constant μ and σ in all cases. These parameters are, in fact, the average of the MoG parameters.

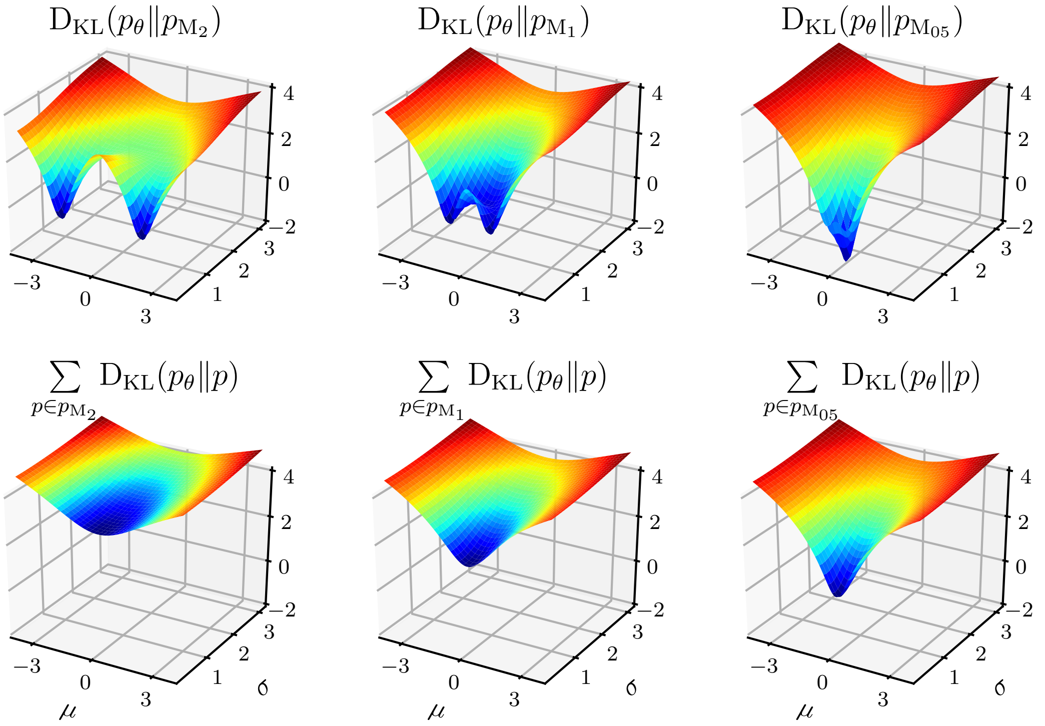


Figure 5.9: Visualization of the KLD between the variable Gaussian distribution (i.e., left side of the argument) and the MoG (i.e., right side of the argument) in various cases. The top row shows the conscious case that would result if data about the GT labels is available. The bottom row shows the unconscious case, which matches the calculation in the objective of the M^{2}VAE.

Figure 5.10 shows the surfaces with the variable Gaussian distribution on the right side of the $\mathrm{D_{KL}}$ argument. It shows the exact behavior that was derived in Section 5.3.2.1. The convex function loss surface shows that there was similar behavior in the conscious and unconscious cases. Furthermore, the minima coincide with each other (see Table 5.4). The optimal parameters in the minimum for p_θ try to cover the whole MoG. Therefore, μ becomes the average of the MoG, while σ increases to cover all modes. This behavior is called mean-seeking (see Fig. 5.8).

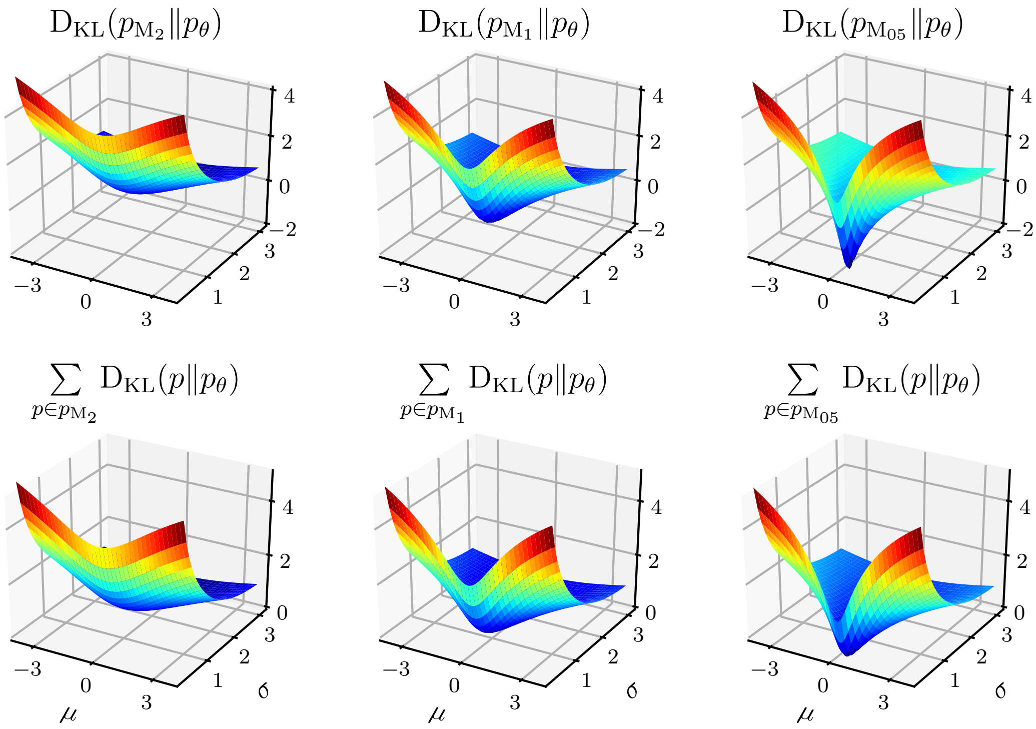


Figure 5.10: Visualization of the KLD between the MoG (i.e., left side of the argument) and the variable Gaussian distribution (i.e., right side of the argument) in various cases. The top row shows the conscious case that would result if data about the GT labels is available. The bottom row shows the unconscious case, which matches the calculation in the objective of the M^2VAE.

Finally, the Jensen–Shannon divergence (JSD) is calculated and shown in Table 5.3 for all depicted cases between the MoG and Gaussian distributions. The JSD, compared to KLD, is a true metric and enables the comparison between different results. The JSD of the conscious case is always less than or equal to the unconscious case. The unconscious case in Fig. 5.9 shows the worst results because an averaging behavior does not describe a MoG well in the cases in which modes are far apart or collide. The conscious case in Fig. 5.9 shows desirable behavior (mode-seeking) when the modes of the MoG are far apart, but it cannot describe the variance well if the modes collide. The conscious and unconscious case in Fig. 5.10, with their mean-seeking behaviors, show desirable performance when the modes of the MoG are far apart and performs best if the modes collide.

Table 5.3: Jensen–Shannon divergence (JSD) for the optimal values μ and σ (see Table 5.4) that minimize the KLD between p_θ and the MoG (lower is better).

	wrt. Fig. 5.9			wrt. Fig. 5.10		
	$(p_{\mathrm{M}_2}\|p_{\theta^*})$	$(p_{\mathrm{M}_1}\|p_{\theta^*})$	$(p_{\mathrm{M}_{05}}\|p_{\theta^*})$	$(p_{\mathrm{M}_2}\|p_{\theta^*})$	$(p_{\mathrm{M}_1}\|p_{\theta^*})$	$(p_{\mathrm{M}_{05}}\|p_{\theta^*})$
conscious	0.465	0.464	0.175	0.566	0.394	0.159
unconscious	0.832	0.723	0.383	0.566	0.394	0.159

Table 5.4: Minima values for all depicted cases. The results show the different behaviors beh.: mode-seeking (1), mean-seeking (2), and averaging (3) (see Fig. 5.8).

		wrt. Fig. 5.9	
conscious $(\mu, \sigma, \text{beh.})$	± 2.0, 0.3, (1)	± 1.0, 0.3, (1)	.0, .5, (1)
unconscious $(\mu, \sigma, \text{beh.})$	.0, .3, (3)	.0, .3, (3)	.0, .3, (3)
		wrt. Fig. 5.10	
conscious $(\mu, \sigma, \text{beh.})$	.0, 2., (2)	.0, 1., (2)	.0, .6, (2)
unconscious $(\mu, \sigma, \text{beh.})$	.0, 2., (2)	.0, 1., (2)	.0, .6, (2)

5.3.2.3 Discussion

The former results cover all behaviors of the $\mathrm{D_{KL}}$ term in the objective of the proposed M^{2}VAE. However, the most important investigation comes with the unconscious case in Fig. 5.10 because this describes the learning behavior between a multi-modal observation without ambiguity and a single modality (i.e., modality drop-out) with ambiguity. The mean-seeking behavior ensures these three essential facts: First, the variance grows as soon as ambiguity occurs, which facilitates the use of the M^{2}VAE's variances as a measure of information content. Second, the mean-seeking behavior captures all unambiguous embeddings with one ambiguous embedding, which lets the M^{2}VAE construct a consistent latent space. Third, putting extra effort into the labeling of observations to improve the latent embeddings between different sets of modalities is futile because the optimal values remain the same.

Recapping the JMMVAE and M^{2}VAE properties under these findings leads to the following conclusions (see Fig. 5.11). The JVAE part of the JMMVAE learns the bi-modal embeddings at unique locations in the latent space. The uni-modal encoders q_{ϕ_*} of the JMMVAE adopt the latent embeddings just from the bi-modal encoder. This naïve training objective for q_{ϕ_*} may lead to confusion because the embeddings are blindly adopted for the corresponding inputs. When observation ambiguities occur, q_{ϕ_*} learns to describe various bi-modal embeddings via mean-seeking behavior. This happens without checking whether this ambiguous embedding reconstructs into a meaningful sample. $q_{\phi_{\mathrm{ab}}}$ may have already encoded some complete but different information in the ambiguous location,[12] which is also reconstructed into that different sample. The mean-seeking behavior may have put the ambiguous encoding into that same spot as well, which then leads to completely different reconstructions. This fact makes modality-exchange, which is the original purpose of the JMMVAE, questionable, if the data set contains ambiguous observations.

The M^{2}VAE, on the contrary, has additional reconstruction terms for the uni-modal encoders in the objective. These terms prevent the possibly faulty behavior of the JMMVAE.

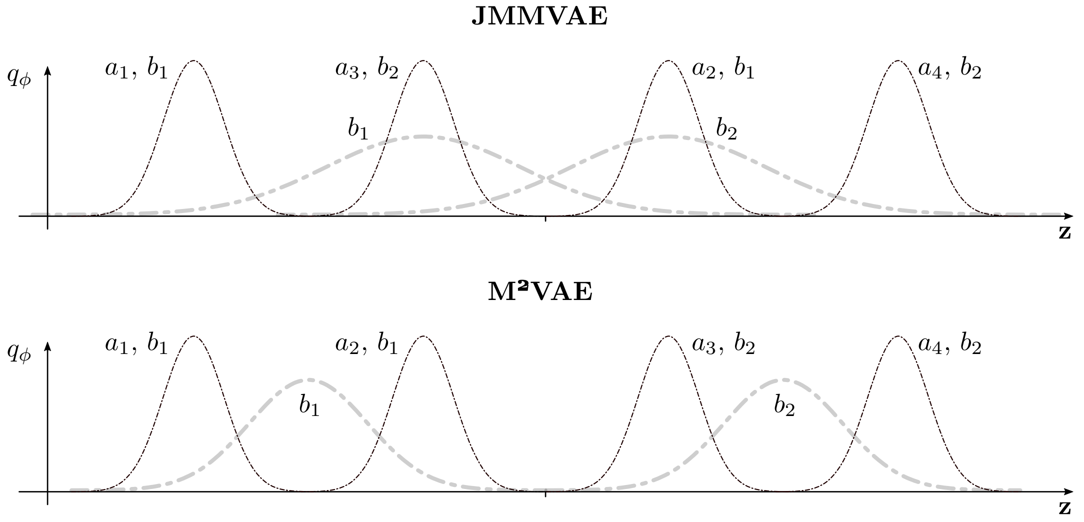
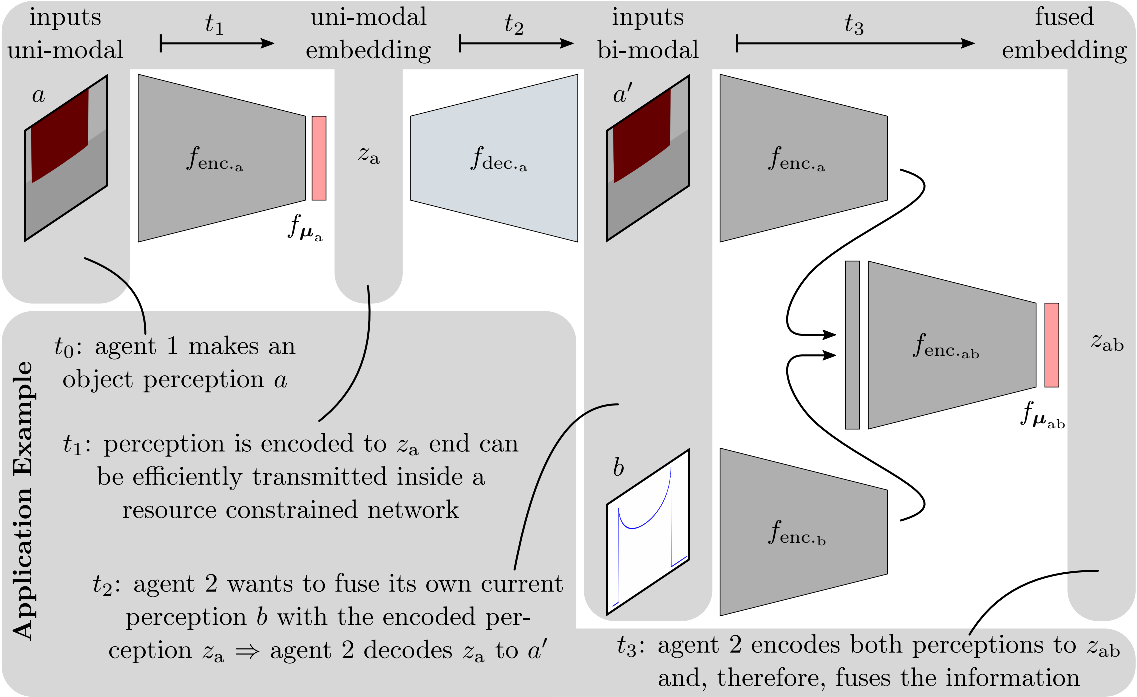


Figure 5.11: The XOR data set $((a_1, a_2, a_3, a_4) = (01, 10, 00, 11)$ and $(b_1, b_2) = (1, 0))$ is adopted in this figure. It depicts likely embeddings of the JMMVAE and the M^{2}VAE for XOR data set. Both approaches have the KLDs that enable the uni-modal encoders to match the latent embeddings of the bi-modal encoder. However, only the M^{2}VAE has additional reconstruction losses for the uni-modal encoders. The JMMVAE embeds the bi-modal observations arbitrarily, which may cause the depicted confusion of b_1, for example, representing (a_3, b_2). The M^{2}VAE can detect these confusions and reorder all embeddings accordingly.

[12] The location is only ambiguous for the uni-modal encoders because it only experiences partial observations. The bi-modal encoder always experiences the full observation, and every sample encodes to a unique location in the latent space.

5.4 Auto Re-Encoding

A deep generative model (DGM) enables the combining of multi-modal data by means of the joint posterior and likelihood models to embed various modalities into the same latent space. This facilitates sensor fusion via neuronal networks that obey variational inference (VI) methods. The M²VAE framework that learns coherent posterior models between all modality subsets in a sensor setup was introduced in Section 5.2. This approach may give an end-to-end solution to learn inverse sensor models (ISMs) by relying on binary Bayes filters [Kor+18b; Wes+18] and may overcome their limitations regarding multi-modal fusion [Elf92; Kor+18b].



t_0: agent 1 makes an object perception a

t_1: perception is encoded to z_a end can be efficiently transmitted inside a resource constrained network

t_2: agent 2 wants to fuse its own current perception b with the encoded perception z_a ⇒ agent 2 decodes z_a to a'

t_3: agent 2 encodes both perceptions to z_ab and, therefore, fuses the information

Figure 5.12: Proposed in-place fusion of a simulated, distributed camera and LiDAR perception from the AMiRo [Her+16]. A red cylinder that is fused to the latent embedding z_ab was sensed. The interaction between the multi-modal encoder (right) facilitating sensor fusion of the formerly encoded latent embedding z_a (left) and the new observation b is shown. Sampling is not applied during inference, and thus, only the mean-layer $f_{\text{enc.}*}$ is used for encoding.

DGMs may circumvent the simplifying assumption of conditionally-independent measurements for distributed estimation to facilitate fusion (c.f. [Lig+01]) by following a data-driven approach of the M²VAE, which models the full posterior distribution in a multi-modal setup. To achieve this goal, the author proposes a multi-modal, in-place, posterior fusion approach based on M^{2}VAE in this section. This approach is applicable in distributed sensing and fusion tasks as proposed in Korthals et al. [Kor+19d].

Compressed representations (i.e., the latent space's embedding, z) of an object's observations $o_{\mathcal{M}'}$ by the modality-set $\mathcal{M}'$ are shared between all sensing agents and updated as follows: As depicted in Fig. 5.12, $z_\text{a} \in \mathcal{Z}$ can be unfolded to the original observation using the M^{2}VAE's decoder networks in combination with any new observation b to update the information in-place $z_\text{a} \xrightarrow{b} z_\text{ab} \in \mathcal{Z}$. To facilitate this fusion approach, a novel training objective is necessary to maintain the re-encoding of the embeddings (i.e., observation → encoding → decoding → encoding → ...).

5.4.1 In-Place Sensor Fusion

The following is the proposed concept of in-place sensor fusion that updates an existing embedding z to z^* using a M^{2}VAE:

$$q_{\phi_{m\sqcup\mathcal{M}'}}(z^*|o_m, f_{\mathcal{M}'}(z)) \quad \text{with} \quad f_{\mathcal{M}'}(z) = \bigcup_{m'\in\mathcal{M}'} p_{\theta_{m'}}(o_{m'}|z). \tag{5.148}$$

$f_{\mathcal{M}'}(z)$ decodes the former z to all observations that contributed to its embedding via $q_{\phi_{\mathcal{M}'}}$. Then $q_{\phi_{m\sqcup\mathcal{M}'}}$ is applied to the greater set of observations, including the new observation o_m

A necessary requirement of Eq. (5.148) is that auto re-encoding (i.e., $z \to z$ via $q_{\phi_{\mathcal{M}'}}(z|\mathcal{M}')$) does not manipulate the information represented by z in an unrecoverable way (e.g., by perturbating the data in a way makes that the class label switch). One may assume that a VAE tends to have a natural denoising characteristic (despite the explicit denoising autoencoder (DAE) by Im et al. [Im+15]), as explained in Section 3.3.4.2. This should re-encode any z into a smoothed version of itself by means of the reconstruction loss concerning the observation. Surprisingly, this behavior only occurs for clearly-separable observations, as discussed in later sections. For non-separable data, the common VAE tends to re-encode any observation to the prior's mean and, thus, fundamentally change the initial information. Similar observations were already made by Dosovitskiy et al. [Dos+16], whose findings impede the requirements of in-place sensor fusion. Therefore, an additional training objective is necessary to circumvent changes in the output path.

5.4.2 Training with Re-Encoding

To maintain the stability and immutability of the encoding during re-encoding, a new training objective is proposed by adding a re-encoding loss to the common objective (see Fig. 5.13):

$$\mathcal{L}_{\text{reenc.}} = \mathcal{L} - \underbrace{\alpha \, \mathrm{D}(q_\phi(z|a) \| q_\phi(z|a'))}_{\text{re-encoding loss}} \,. \tag{5.149}$$

$\mathcal{L}$ is the ELBO (i.e., training objective) of any desired VAE, that includes familiar regularization and reconstruction losses. D compares the current encoding and its re-encoded sample using any divergence function or loss. The parameter $\alpha \in \mathbb{R}^+$ scales the re-encoding loss to leverage its influence against to the original training objective. $\mathcal{L}_{\text{reenc.}}$ is the final objective, which again, is an ELBO, if $\mathrm{D} \geq 0$.

An architecture setup for use during training is proposed in Fig. 5.13. The same encoder-network can be placed in the first place, as well as behind the decoder-network by using the weight-sharing technique. Thus, the encoder-network weights are influenced by all losses during BP. To effectively train this setup, two essential hints need to be considered: First, the re-encoding loss cannot be applied right from the beginning of the training. Otherwise, the encoder-network just learns to generate constant values as a trivial solution to minimizing the re-encoding loss. Therefore, techniques like warm-up by Sønderby et al. [Søn+16], which gradually increases α over the epochs, are highly recommended. Second, the decoder-network weights need to be pinned when applying the re-encoding loss.[13] Otherwise, the networks learn to cheat on the loss by embedding particular information about z in the reconstruction. This needs to be prohibited because the reconstructed observation must be the only available information that should be considered for re-encoding. A proven method is to train the common VAE/JMMVAE/M^{2}VAE/… first, and then pin the decoder-network weights, and finally, learn the whole architecture in Fig. 5.13 with the proposed re-encoding loss.

[13] Weight-pinning a layer means that its weights remain unchanged during BP.

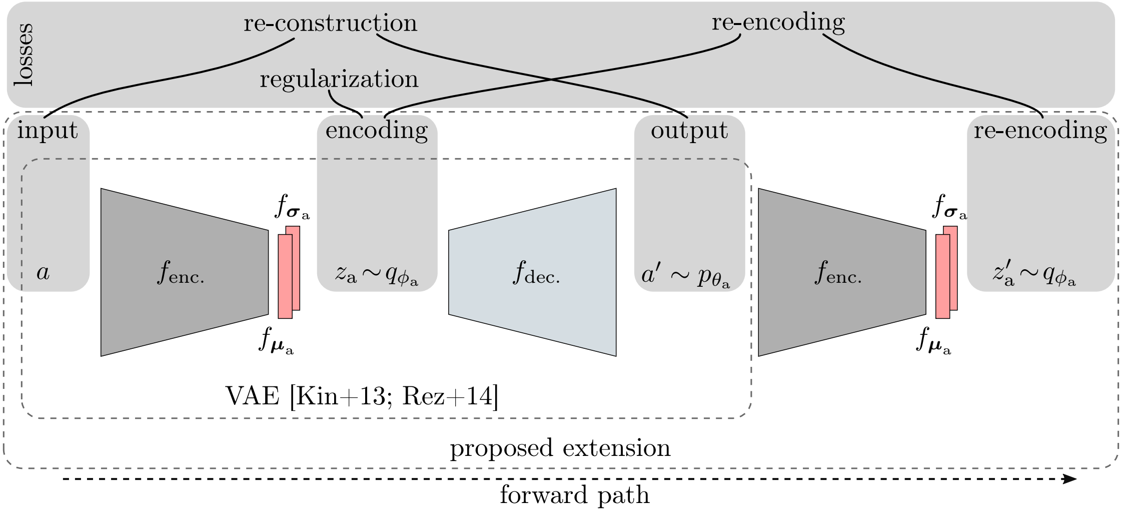

losses
re-construction
re-encoding
regularization
input
encoding
output
re-encoding
a
$f_{\text{enc.}}$
$f_{\boldsymbol{\sigma}_\text{a}}$
$f_{\boldsymbol{\mu}_\text{a}}$
$z_\text{a} \sim q_{\phi_\text{a}}$
$f_{\text{dec.}}$
$a' \sim p_{\theta_\text{a}}$
$f_{\text{enc.}}$
$f_{\boldsymbol{\sigma}_\text{a}}$
$f_{\boldsymbol{\mu}_\text{a}}$
$z'_\text{a} \sim q_{\phi_\text{a}}$
VAE [Kin+13; Rez+14]
proposed extension
forward path

Figure 5.13: Uni-modal VAE architecture setup that is trainable via the proposed re-encoding loss.

5.4.3 Re-Encoding Demonstration

The benefits of training a VAE via the proposed re-encoding loss approach are twofold: First, the re-encodings become nearly immutable and label switching can be suppressed (see Fig. 5.14). Second, the latent spaces' statistics about the encoder networks can be visualized by traversing the latent space $\mathcal{Z}$ while obtaining the output parameters of the encoder via re-encoding (see Fig. 5.16).

The immutability of single $z_{\text{init.}} \in \mathcal{Z}$ are visualized as colorized perturbations by calculating the Euclidean distance z_{diff} between the encoding before and after (i.e., $z_{\text{reenc.}}$) re-encoding:

$$z_{\text{diff}} = \|z_{\text{init.}} - z_{\text{reenc.}}\|_2 \quad \text{with} \quad z_{\text{reenc.}} = f_{\boldsymbol{\mu}}(f_{\text{enc.}}(f_{\text{dec.}}(z_{\text{init.}}))). \tag{5.150}$$

Note that Eq. (5.150) does not contain any sampling like $z_{\text{reenc.}} \sim q_\phi(z|p_\theta(z_{\text{init.}}))$ because the networks are now applied during inference (i.e., after the training that depends on the stochastic process).

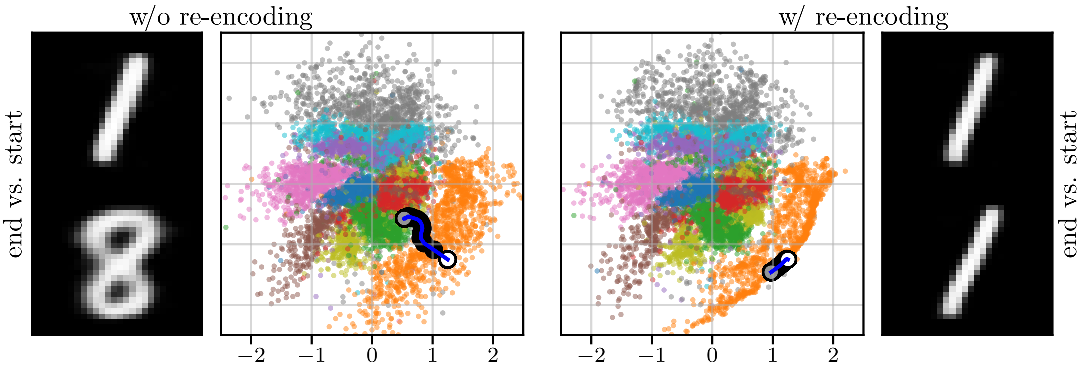


Figure 5.14: Embeddings of the MNIST test set using a VAE's encoder q_ϕ. Left: trained via common loss (see Table B.4). Right: trained via proposed loss (see Table B.5). Black trajectories indicate the initial (○) encoding of a '1' and the terminal (●) encoding after 200 steps of auto re-encoding. Without the proposed loss, the re-encoding traverses the latent space and label-switching occurs ('1' → '8'). The color coding is the same as that in Fig. 3.7.

The results in this section show a uni-modal VAE that was trained on the MNIST data set. A VAE that was trained via the common loss (see Table B.4) and the proposed loss (see Table B.5) is considered. Figure 5.15 shows z_{diff} for every point in the latent space. The lower the value, the better the re-encoding property is. The common VAE shows very high perturbation all over the latent space. A value of $z_{\text{diff}} = 1.$, for example, means that the position of the re-encoded z has changed by a magnitude of 1., which is quite a lot in comparison to the whole confinement of the demonstrated data set's embedding (see Fig. 5.14). The proposed VAE, as trained and shown in Fig. 5.13, shows almost no perturbation for the confinement of the demonstrated data set embedding. However, one might mention the increased perturbation at the figure's boundaries of Fig. 5.15. These increased perturbations lie outside of any embedding when Fig. 5.14 is taken into account. Therefore, they can be neglected. It is also worth mentioning that for the demonstrated MNIST data-set, the median perturbation, after one step of auto re-encoding, of the common VAE is $\bar{z}_{\text{diff}} = .0651$, and this value was reduced by the proposed approach by one magnitude to $\bar{z}_{\text{diff}} = .00902$.[14]

[14] Further values about this finding can be found in Table B.26.

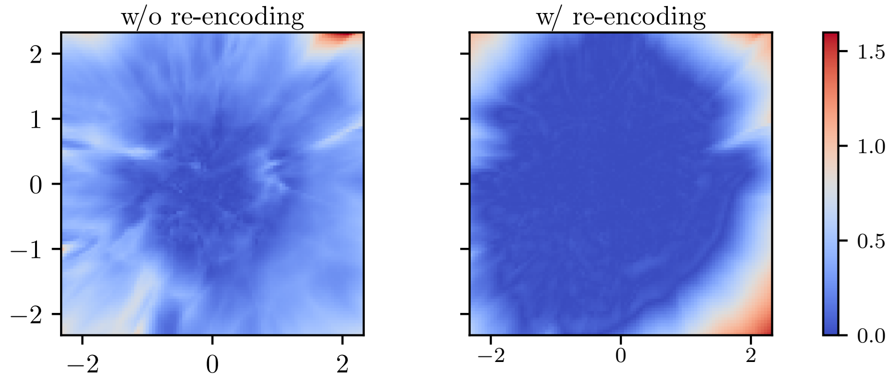


Figure 5.15: Quantitative difference z_{diff} between the initial and re-encoded mean value of z after one step of auto re-encoding (lower is better).

Although an arbitrary z can be chosen for decoding in the latent space, for example, its variance at this particular position z remains unknown. However, if one could regenerate the missing parameters for a z, then one could visualize the losses for every position in the latent space, for instance. The proposed training approach facilitates this visualization of the latent space behavior, which would not be possible otherwise. It enables insight to the latent space and reveals the intriguing behaviors of VAEs.

This technique works similar to the sampling from the latent space, which produces the lattice of output data, as visualized in the right column of Fig. 3.4. However, instead of visualizing the reconstruction for $a'_{i,j} = f_{\text{dec}}(z_{i,j})$ for a specific $z_{i,j}$, now $a'_{i,j}$ are re-encoded to get the missing parameters for $z_{i,j}$. Re-generating the missing parameters, which are the variances $\sigma_{i,j} = \exp f_{\boldsymbol{\sigma}}(f_{\text{enc.}}(f_{\text{dec.}}(z)))$, enables the plotting of the standard deviation itself and all parts of the training losses: the regularizer (i.e., the KLD concerning the prior), reconstruction loss, and ELBO. Thus, instead of visualizing these statistics for every sample in the data set, as shown for the standard deviation in the middle column of Fig. 3.4, continuous figures can be plotted over the support of $\mathcal{Z}$.

As shown in Fig. 5.16, the encoder network tends to tie up the standard deviation[15] σ and, therefore, deviates from the prior, as indicated by the KLD, where the encoder embeds observations into $\mathcal{Z}$. Furthermore, the reconstruction loss becomes higher at the vicinity of cluster boarders, where the encoder embeds poor or ambiguous observations. The ELBO demonstrates the combined loss, which shows virtually the same behavior as the reconstruction loss because of its dominance that is attributable to the high input dimensionality (i.e., $D_{\text{a}} = 784$ vs. $D_{\text{z}} = 2$ in this particular case).

[15] The standard deviation for the depicted example is the square root of the total variance with $\sigma^2 = \sum_{D_{\text{z}}} \exp f_{\boldsymbol{\sigma}}(f_{\text{enc.}}(\cdot))$ (see Andres [And13]).

Applying this kind of visualization to the common VAE would result in misleading results because the re-encoded sample would not be the same as the originally encoded one. Only the proposed training approach preserves re-encoded samples, which leads to interpretable figures. However, for the sake of completeness, the analog figure of the common VAE in comparison to Fig. 5.16 can be found in Appendix B.3, Fig. B.7. Furthermore, Fig. B.8 shows the embedding of the MNIST test data set, using the encoder-networks of both approaches. It can be seen that for the confinement of the data set, the proposed approach follows the true trend more precisely than the commonly trained VAE does.

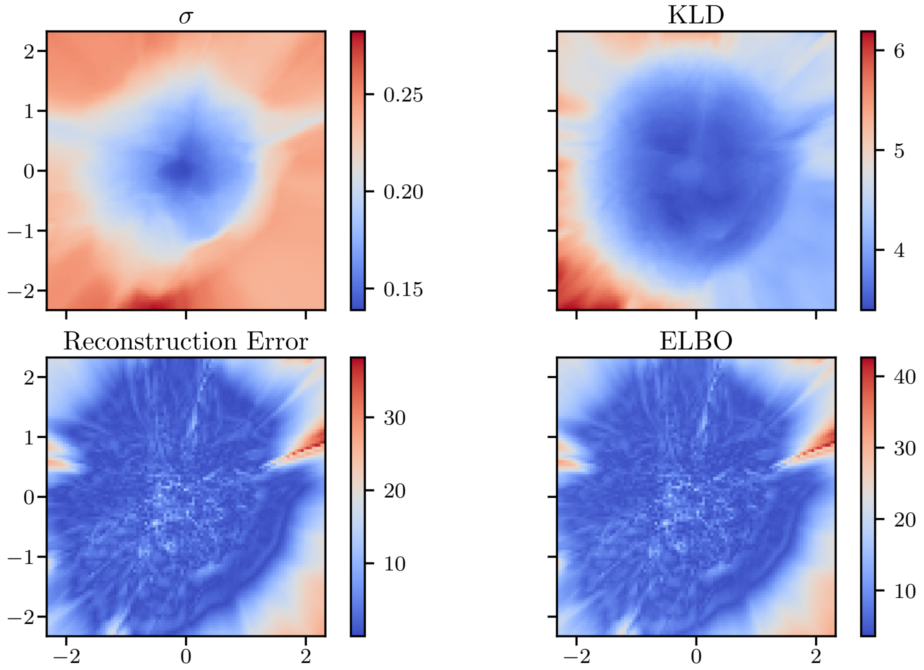


Figure 5.16: Quantitative latent space statistics for the proposed approach.

6 Multi-Modal Data Sets and their Properties

Many data set collections exist and are publicly available via the WWW, such as robotics (Radish, MRPT, IJRR), computer vision (CVonline, CVPapers, YACVID, Computer Vision Online), and general data set collections and search engines for machine learning (Kaggle, UCI, CMU, VisualData, Google's Dataset Search), to name just a few besides the many widespread single-hosted, undocumented, or unpublished data sets.[1] Despite the availability of all these sources, a categorization of multi-modal data set properties is missing in the literature.

To find a proper categorization, a sufficient definition of properties is presented in Section 6.1, which can be facilitated to match available data sets to any experiment. Furthermore, Section 6.1 contains insight into a multitude of available data sets, and the author highlights the most commonly-used data sets.[2] Although the available data sets truly reveal the capabilities and possibilities of DGMs, they are by no means comprehensible. It will be revealed by the author that most available mulit-modal data sets do not exhibit drop-out or ambiguities.

The available data sets are not truly multi-modal nature in, as discussed in Section 4.2.3. New and comprehensible multi-modal data sets need to be developed that help to reveal the true capabilities of the proposed M^{2}VAE. Therefore, in Section 6.2, particular data sets and a generative technique for multi-modal data sets are proposed and introduced. Finally, Section 6.3 contains a discussion of the choice of available and proposed data sets for further experiments.

6.1 Review of Available Data Sets

Data sets can be categorized as uni- or multi-modal with class or attribute presence. However, every labeled uni-modal data set can be considered to be a multi-modal data set because the labels can be used as their own modality. Therefore, data sets

[1] See Table B.27 for a link-list to the corresponding websites.

[2] Deeper insights and more comprehensible overviews of other data sets regarding the corresponding fields of research can be commonly found in the referenced articles.

that are promoted as multi-modal because of their variety of GT labels are listed as uni-modal data sets.[3]

6.1.1 Data Set Properties

The data sets are categorized according to Table 6.1 into natural source (src.), natural correlation (corr.), stream-like (S), multi-modality, and continues (cont.) or discrete (disc.) nature of phenomenons. Furthermore, the purpose of attributes and classes is categorized into regression (regr.) or classification (class.) tasks and if the correlation (corr.) in multi-modal data sets occurs concerning the attribute (attr.) or the classes.[4]

Natural refers to two different features of the data set. The natural source (src.) column refers to the source of data if it comes from a real-world recording (✓) or a simulator (•). The natural correlation (corr.) column refers to how the correlation between the modalities was ensured. It was either done by collating or merging two independent data sets (•) (e.g., artificial correlation as in the MNIST+SVHN data set) or if the correlation naturally occurs between the modalities (✓) (e.g., natural correlation as in the RGB-D data set). If the data set is multi-modal, then the number of modalities that have the corresponding natural feature is given in parenthesis if it is different from the number of absolute modalities in the data set (✓(#)).

Stream-like (S) refers to the time-association of the data set. For instance, a data set can be continuous in nature, like the shape positions in the dSprites data set, but independent in time (•). Auditory or visual modalities, like those in the RGB-D data set, have a consecutive and time-dependent nature (✓). If the data set is multi-modal, then the number of modalities that are stream-like is given in parenthesis if it is different from the number of absolute modalities in the data set (✓(#)).

Multi-modal refers to the number of modalities. Uni-modal data sets are always multi-modal if one choses to take the labels as additional modalities (•). This technique is commonly performed in multi-modal learning but it does not relate to the nature of true multi-modal observations. Multi-modal data sets have an additional number in parenthesis, which stands for how many different modalities were recorded (✓(#)).

Nature refers to the discrete or continuous nature of the observed phenomenon. A modality can be discrete in nature. For instance, static images of an object like a

[3] MNIST-A and dSprite are promoted as multi-modal data sets, but the phenomenons they record are only observed via one modality (i.e., images), while the other modality describes the sprite in the image based on its GT information (e.g., pose, shape, etc.)

[4] For this thesis, classes and attributes are particularly differentiated, despite the fact, that a class association is, of course, an attribute of the phenomenon.

car in the CIFAR-10 data set (✓). On the contrary, in the RGB-D data set, objects are recorded in a continuous video by sweeping around them (✓). Multi-modal data sets can have multiple modalities of different natures. Furthermore, the number of modalities with the corresponding nature (i.e., discrete or continuous) is given in parenthesis if it is different from the number of absolute modalities in the data set (✓(#)).

Purpose refers to the intentional design of the data set and how it was labeled. This is important to the later downstream evaluation of the DGMs. If continuous attributes are labeled, like the shape position in the dSprites data set, then one can perform a regression on that feature (✓). If the data set is labeled concerning its discrete features, like the numeric class association in the MNIST data set, then one can only perform a classification (✓). Both purposes can be checked or unchecked if the corresponding labels exists or are missing.

Correlation (corr.) categorizes correlation types as attribute (attr.) or class correlations. In various cases, like the MNIST+SVHN or MNIST+USPS data set, two uni-modal data sets are collated by their numeric labels (class) to create a new multi-modal data set. The collation is done despite their different attributes (e.g., skewness or stroke width), which lets the attributes become a noise signal in the data set without any information. True multi-modal data sets that are built by observing natural processes, on the contrary, are commonly correlated by their perceivable attributes and class associations (both are checked).[5] Attribute attr. correlation refers to anything but classes (e.g., skewness or stroke width) that is are correlated between two modalities (✓). Class correlation (class) refers to the collating of data sets based on their labels (✓) alone. Both properties can be checked or unchecked when the corresponding correlation exists or is missing, respectively.

6.1.2 Multi-Modal Data Sets in the Wild

This section contains an exhaustive overview of the most common data sets used in the publications referenced within this thesis.[6] Table 6.1 contains a summary of these data sets and the important features for a multi-modal analysis. Although many uni-modal data sets are listed, they are still used for multi-modal analysis. A separation of data sets into six different categories is shown in Table 6.1: 2D Object, 3D Object, 2D & 3D Faces, human activity recognition (HAR), autonomous ground vehicle (AGV), and Simulator. All data sets within these categories are restricted to

[5] CUAVE is one example with true multi-modal observations but w/o GT concerning the attributes. Therefore, only classification tasks are of interest in this data set.

[6] The task of multi-modal speaker traits recognition (POM), multi-modal sentiment analysis (CMU-MOSI, ICT-MMMO, YouTube, MOUD), and multi-modal emotion recognition (IEMOCAP) are beyond the scope of this thesis and, therefore, the corresponding data sets from Table 4.1 will not be revisited.

unstructured data because this kind of data shows the greatest potential of DGMs, which is learning from data. A detailed explanation of the data sets that were also investigated by the multi-modal DGM publication from Table 4.1 is given as well. Further categories like speech and text are also neglected because these domains are only merrily investigated by other competitive publications on multi-modal DGMs and out of scope of this work.

Table 6.1: Overview of multi-modal data sets that were used in the publications cited in this thesis. Multiple sensors with the same modality are counted as one sensor. ✓: true, •: false, –: not applicable

	natural			mulit-	nature		purpose		corr.	
data set	src.	corr.	S	modal	cont.	disc.	regr.	class.	attr.	class
				2D Object						
(E)MNIST[LeC+98; Coh+17]	✓	–	•	•	•	✓	•	✓	–	–
bMNIST[Sal+08]	✓	–	•	•	•	✓	•	✓	–	–
noisy MNIST[And+13]	✓	•	•	✓(2)	•	✓	•	✓	•	✓
multi-MNIST[Esl+16]	✓	–	•	•	•	✓	•	✓	–	–
MM-MNIST[Vie+19a]	✓	✓	•	✓(2)	•	✓	•	✓	✓	✓
AV-MNIST[Vie+19a]	✓	•	✓(1)	✓(2)	✓(1)	✓(1)	•	✓	•	✓
MNIST-A[Ved+17]	✓	–	•	•	✓	✓	✓	✓	–	–
divided MNIST	✓	✓	•	✓(x)	•	✓	•	✓	✓	✓
FMNIST[Xia+17]	✓	–	•	•	•	✓	•	✓	–	–
SVHN[Net+11]	✓	–	•	•	•	✓	•	✓	–	–
USPS[Hul94]	✓	–	•	•	•	✓	•	✓	–	–
MNIST+SVHN[Liu+17b]	✓	•	•	✓(2)	•	✓	•	✓	•	✓
MNIST+USPS[Liu+17b]	✓	•	•	✓(2)	•	✓	•	✓	•	✓
CUB[Wel+10; Wah+11]	✓	–	•	•	•	✓	•	✓	–	–
multi-MNIST[Esl+16]	✓	–	•	•	•	✓	•	✓	–	–
UT-Zap50K[Yu+14a; Yu+17]	✓	–	•	•	•	✓	•	✓	–	–
dSprites[Mat+17; Bur+18b]	•	–	•	•	✓	✓	✓	✓	–	–
Omniglot[Lak+15]	✓	✓	✓(1)	✓(2)	✓(1)	✓(1)	✓	✓	✓	✓
CalTech 101[Li +06; Mar+10]	✓	–	•	•	•	✓	•	✓	–	–
CIFAR-10/100[Kri09]	✓	–	•	•	•	✓	•	✓	–	–
MIR Flickr[Hui+08; MJH+10]	✓	–	•	•	•	✓	•	✓	–	–
ImageNet[Hui+08; MJH+10]	✓	–	•	•	•	✓	•	✓	–	–
SVHN[Net+11]	✓	–	•	•	•	✓	•	✓	–	–
USPS[Hul94]	✓	–	•	•	•	✓	•	✓	–	–
3D Shapes[Bur+18a]	•	–	•	•	✓	✓	✓	✓	–	–
				3D Object						
Chairs[Aub+14]	–	–	•	•	•	✓	•	✓	–	–
ShapeNet[Cha+15]	–	–	•	•	•	✓	•	✓	–	–
PASCAL3D+[Xia+14]	✓(1)	•	•	✓(2)	•	✓	•	✓	✓	✓
RGB-D[Lai+11]	✓	✓	✓	✓(2)	✓	•	•	✓	✓	✓
				2D & 3D Faces						
Frey Face[Row+00]	✓	–	•	•	✓	•	•	•	–	–

Celeb(A)[Sun+14; Liu+15]	✓	–	·	·	·	✓	·	✓	–	–
LFW(A)[Hua+07; Liu+15]	✓	–	·	·	·	✓	·	✓	–	–
3D Face Model[Pay+09]	·	–	·	·	✓	·	✓	·	–	–
CUAVE[Pat+02]	✓	✓	✓	✓(2)	✓	·	·	✓	✓	✓
AVLetters(2)[Mat+02; Cox+08]	✓	✓	✓	✓(2)	✓	·	·	✓	✓	✓
HAR										
WESAD[Sch+18]	✓	✓	✓	✓(7)	✓	·	·	✓	✓	✓
HAR[Ang+13]	✓	✓	✓	✓(2)	✓	·	·	✓	✓	✓
MHAD[Ofl+13]	✓	✓	✓	✓(5)	✓	·	·	✓	✓	✓
AGV										
KITTI[Gei+13]	✓	✓	✓	✓(10)	✓	·	·	✓	✓	✓
Drive&Act[Mar+19]	✓	✓	✓	✓(18)	✓	·	·	✓	✓	✓
nuScenes[Cae+19]	✓	✓	✓	✓(14)	✓	·	·	✓	✓	✓
FieldSAFE[Kra+17]	✓	✓	✓	✓(7)	✓	·	·	✓	✓	✓
RAGE[Ric+16]	·	–	✓	·	✓	·	·	✓	✓	✓
Robot@Home[RS+17]	✓	✓	✓	✓(2)	✓	·	·	✓	–	–
CREATE[Rou18]	✓	✓	✓	✓(17)	✓	·	·	·	✓	✓
Simulator										
Spriteworld[Wat+19c]	·	✓	✓	·	✓	✓	✓	✓	–	–
Gazebo[Koe+04]	·	✓	✓	✓	✓	✓	✓	✓	✓	✓
MuJoCo[Tod+12]	·	✓	✓	✓	✓	✓	✓	✓	✓	✓
DeepMind Lab[Bea+16]	·	✓	✓	·	✓	✓	✓	✓	–	–
Atari[Bel+13]	·	✓	✓	✓	✓	✓	✓	✓	✓	✓
VizDoom[Kem+16]	·	✓	✓	·	✓	✓	✓	✓	–	–
data set	src.	corr.	S	multi-	cont.	disc.	regr.	class.	attr.	class
	natural			modal	nature		purpose		corr.	

6.1.2.1 2D Object

There are numerous artificial and real-life image data sets for classification because of the lively and huge CV community. The most prominent sets are MNIST, CIFAR-10/100, and ImageNet, but a huge CV database exists for every field of research in which imaging techniques are applied (medicine, astronomic, deep see, animal, and so on), and more can be found in the aforementioned data set search engines

MNIST is data set of handwritten digits by LeCun et al. [LeC+98]. It has 60,000 training and 10,000 testing images of the numeric class labels $0-9$ with 28×28 pixel each. It is a subset of a larger set from NIST [Gro95] which consists of separate digit, upper and lower case, and free text fields from 3,600 writers. A MNIST extension worth mentioning is called **EMNIST** by Cohen et al. [Coh+17]. It contains 240,000 training images and 40,000 testing images of handwritten digits and characters from NIST in 62 classes. The data sets are particularly suited for generative models like VAE, despite their discrete nature and labels because of the high number of samples per class.

bMNIST by [Sal+08] is a binarized version of the original MNIST data set. Each pixel value is stochastically set to true in proportion to its pixel intensity.

MM-MNIST and AV-MNIST are both multi-modal datasets proposed for use in the evaluation of deep fusion architecture searching by Vielzeuf et al. [Vie+19a]. The multi-modal MNIST (MM-MNIST) data set contains pairs of images computed from the original MNIST data set for sue in the evaluation of deep fusion architecture searching. Both images are supposed to be two views of the same MNIST image but from different modalities that were artificially generated via principal component analyses (PCA). First, a PCA of the original MNIST data set was computed, which resulted in a set of singular vectors. Second, the set was separated into two subsets, which were used to re-generate each modality. This technique enables to control the amount of energy (i.e., explained variance) provided to each modality, which is the sum of the energy contained in the chosen vectors. Furthermore, one can also choose a share ratio of the singular vectors, to define the shared energy between modalities. The number and dimensions of samples are identical to those in LeCun et al. [LeC+98].

The MM-MNIST data set is artificially created, but it is worth mentioning that the derived modalities show natural correlation based on their classes and attributes. This is due to the fact that every modality holds various (orthogonal) components of the original information.

The audiovisual MNIST (AV-MNIST) data set was created from one modality of MM-MNIST, which only had 25 percent of energy, plus the Free Spoken Digits Database by Jackson [Jac17]. Both modalities were collated based on their numeric class labels.

MNIST-with-attributes (MNIST-A) by Vedantam et al. [Ved+17] is a modified version of the original MNIST data set. A sampled image was put on a 64 × 64 pixel canvas with a sampled numeric class association (10 values), location (4 values), orientation (3 values), and size (2 values). 290 samples of each possible combination were drawn and rendered, resulting in the new MNIST-A data set.

multi-MNIST by Eslami et al. [Esl+16] is a modified multi-class version of the original MNIST data set. Instead of a single digit per image, multi-MNIST contains zero, one, or two non-overlapping randomly sampled MNIST digits at varying locations in a 50 × 50 pixel image.

noisy MNIST by Andrew et al. [And+13] is a bi-modal collation of the MNIST data set with a different noise approach for each modality. To create the first modality, a random image is sampled from the MNIST data set. It is then randomly scaled and rotated. For the second modality, an image of the same numeric class is sampled and random pixel noise is applied. This data generation process ensures that the numeric class label is the only common variable underlying both modalities.

divided MNIST refers to the technique whereby each image of the MNIST data set is divided into groups of two, four, or more smaller images. The number of modalities into which a sample is split is denoted as follows: “divided MNIST (2)” for a bi-modal and “divided MNIST (4)” for a tetra-modal data set. Dividing or splitting samples of data sets is the de facto baseline standard to generate multiple modalities of the same nature because they also resemble the features of true multi-modal data in class and attribute correlation (see [And+13; Nev+16; Cha+16; Li+16]).

Fashion-MNIST by Xiao et al. [Xia+17] comprises 60,000 images, and the test set has 10,000 images of fashion products from 10 categories like shoes, bags, and sweaters. Fashion-MNIST (FMNIST) serves as a direct analogy for the original MNIST data set for benchmarking ML algorithms because it shares the same image size, data format, and structure of training and testing splits.

UT-Zappos50K (UT-Zap50K) by Yu et al. [Yu+14b; Yu+14a; Yu+17] is a shoe data set consisting of 50,025 catalog images collected from Zappos.com. The images are divided into 4 major classes (shoes, sandals, slippers, boots) followed by attributes of functional types and individual brands. The shoes are centered on a white background and pictured in the same orientation. This data set was created in the context of an online shopping task, where users pay special attention to fine-grained visual differences.

CIFAR-10 and CIFAR-100 by Krizhevsky [Kri09] are subsets of the unlabeled 80 Million Tiny Images data set [Tor+08] that are reliably labeled into 10 and 100 classes. CIFAR-10 consists of 50,000 training images and 10,000 color images with 6,000 images per class, which can be airplane, bird, ship, etc. CIFAR-100 is just like the CIFAR-10, except it has 100 classes containing 600 images each. The 100 classes are grouped into 20 super-classes where each image comes with a super-class (e.g., trees or vehicles) and sub-class (i.e., maple, oak, palm, pine, willow or bicycle, bus, motorcycle, pickup truck, train) label.

Omniglot by Lake et al. [Lak+15] contains 1,623 different handwritten characters from 50 different alphabets written by 20 individuals. Interestingly, each image is paired with stroke data, which are trajectory-like sequences of spatial coordinates and timings for how each character was written. Therefore, Omniglot is a bi-modal data set with the image and label of the character itself and the sequence of how the character was written.

MIR Flickr (MIRFLICKR) comprises either 25,000 [Hui+08] (MIRFLICKR-25K) or 1 million [MJH+10] (MIRFLICKR-1M) real-life images plus 1,386 possible labels per image that describe the scene in it. The labels are designed for visual concept learning and, therefore, comprise information like night, sunset, bird, and sign.

ImageNet by Deng et al. [Den+09; Rus+15] is the largest annotated image data set available. It was designed for use in visual object recognition research. It consists of over 14 million images that were hand-annotated according to the WordNet[7] hierarchy. The ImageNet website[8] provides access to the images through links and various kinds of annotations that categorize the images into 20,000 categories similarly to MIRFLICKR.

CalTech 101 Silhouettes [Mar+10] data set is a binarized image data set of object silhouettes derived from CalTech 101 [Li +06]. Each silhouette is rendered as a filled black polygon on a white background. This data set consists of 4,100 examples in the training set, 2,264 examples in the validation set, and 2,307 examples in the test set, and every image belongs to one of the 101 categories.

MNIST+SVHN and MNIST+USPS are collated bi-modal data sets that were introduced by Tsai et al.; Liu et al. [Tsa+18; Liu+17b] and Liu et al. [Liu+17b], respectively. **The United States Postal Service (USPS)** data set is an image database for handwritten text obtained from letters. It comprises the digital images of approximately 5,000 city names, 5,000 state names, 10,000 ZIP codes, and 50,000 alphanumeric characters with their corresponding labels. **The Street View House Numbers (SVHN)** data set by Netzer et al. [Net+11] was obtained from a large number of street view images. In each image, the single digits of each house number were localized and transcribed, resulting in 600,000 labeled characters in varying formats.

The corresponding bi-modal data sets were retrieved by sampling the numeric class association (0 – 9) and then sampling an image from each of the two data sets. This data generation process ensures that the numeric class label is the only common variable underlying both modalities.

Caltech-UCSD Birds 200 (CUB-200) and CUB-200-2011 by Welinder et al. [Wel+10] and Wah et al. [Wah+11] are data sets of 6,033 and 11,788 annotated images of birds belonging to 200 bird species, respectively. The images were distilled from the MIRFLICKR data set, and each image was annotated with a bounding box, a bird segmentation, and a set of attribute labels with the goal of multi-class categorization and part localization.

dSprites by Matthey et al. [Mat+17] is a data set that was specifically developed to demonstrate the disentanglement feature in VAEs (see [Kim+18; Hig+16; Che+18]). This data set was published as a colored version by Burgess et al. [Bur+18b]. The advantage of dSprites is its known factors of variation: vertical and horizontal position, size, shape (discrete feature), angle, and RGB color.

3dshapes published by Burgess et al. [Bur+18a], and used by Kim et al. [Kim+18] and Watters et al. [Wat+19b], is a data set derived from the GQN's Rooms data set. The GQN data set [Vio+18] published by Eslami et al. [Esl+18] is a composition of

[7] `https://wordnet.princeton.edu/`

[8] `http://www.image-net.org/`

four sets: Rooms, Shepard-Metzler Objects, Jaco Arm (generated by the MuJoCo [Tod+12] simulator), and Mazes (generated by DeepMind Lab [Bea+16]). It is a data set of 3D shapes procedurally generated from six ground truth independent latent factors: floor, wall, object color, object scale, object shape (discrete feature), and camera orientation. All possible combinations of these latent factors are present exactly once, generating 480,000 total RGB images.

6.1.2.2 3D Object

Chairs [Aub+14] used by Higgins et al. [Hig+16], Watters et al. [Wat+19b] and [Kim+18] is a collection of 3D CAD chair models. The data set originates from Google's publicly-available 3D Warehouse on-line repository of publicly available, user-contributed 3D graphics content created using Google SketchUp. The 1,393 available chairs were manually culled from the repository to represent a variety of chair styles. Each 3D chair model was rendered on a white background from 62 different viewpoints, which results in 86,366 samples.

ShapeNet [Cha+15] used by Mescheder et al. [Mes+19] and [Cho+16] is a collection of 3D CAD models that are organized according to the WordNet hierarchy, analog to ImageNet. It has indexed more than three million models in total, from which 220,000 models are classified into 3,135 categories.

PASCAL3D+ by Xiang et al. [Xia+14] is based on PASCAL 2012 (real) detection images [Eve+] augmented by (artificial) 3D CAD model alignments. It comprises 12 categories with an average of more than 3,000 object instances per category.

RGB-D by Lai et al. [Lai+11] is an objects and scenes data set using Kinect style 3D camera that records synchronized and aligned RGB and depth images. 300 objects are organized into 51 categories arranged using WordNet hierarchy. Each object was isolated and placed on a turntable, and video sequences were captured for one whole rotation from three different viewpoints. Aside from isolated views of the objects, the data set also includes 22 annotated video sequences of natural scenes (e.g., office workspaces, meeting rooms, kitchen) containing objects from the data set.

6.1.2.3 2D & 3D Faces

Facial data sets are commonly used in data science because they facilitate facial attribute estimation (FAE) and facial attribute manipulation (FAM) applications. Common data sets with attributes are Frey Face by Roweis et al. [Row+00], 3D Face Model by Paysan et al. [Pay+09], the CelebFaces Attributes Dataset (CelebA), and Labeled Faces in the Wild Attributes (LFWA) by Liu et al. [Liu+15]. To date, the most popular and commonly-used data sets in both FAE and FAM are CelebA and LFWA, and a comprehensive overview of facial attribute data sets was given in Zheng et al. [Zhe+18].

Frey Face consists of around 2,000 images of Brendan Frey's face taken as sequential frames from a piece of video footage. This data set introduced in the Local Linear Embedding article by Roweis et al. [Row+00] has a time ordering of the frames and, therefore, is expected to show a high correlation between frames close in time that constitute a known structure that one hopes to recover through dimension reduction.

The 3D Face Model data set by Paysan et al. [Pay+09] is a generative 3D shape and texture model that was developed to benchmark pose and illumination invariant face recognition tasks. The data set can be fit to 2D or 3D images acquired under varying situations and with different sensors, which makes it a potential multi-modal data set. The model parameters separate pose, lighting, imaging, and identity parameters, which facilitate continuous and discrete attributes.

Labeled Faces in the Wild (LFW) by Huang et al. [Hua+07] (ca. 13,000 images) and the Celeb-Faces (Celeb) [Sun+14] (ca. 200,000 images) data set are both large-scale face attribute data sets made of publicly-available images of celebrities. They come with the celebrity's name as a per-image label, which makes it suitable for classification.

CelebA and LFWA are both data sets that were extended by Liu et al. [Liu+15] using their face attributes. Each has 40 binary attribute annotations like mustache, hair, and sunglass. The images cover large pose variations and background clutter, which makes the data sets particularly interesting for data-driven approaches.

Clemson University Audio Visual Experiments (CUAVE) and AVLetters(2) by Patterson et al. [Pat+02], Matthews et al. [Mat+02], and Cox et al. [Cox+08] are video and audio recordings of speakers who read numerical digits and letters. CUAVE comprises 36 speakers saying the digits $0-9$ while AVLetters(2) contains 10 to 5 speakers saying the letters $A - Z$ various times. Besides the audio recording, visual recordings of the faces or lip regions were gathered to mimic a listener's situation.

6.1.2.4 Human Activity Recognition (HAR)

HAR aims to recognize the actions of an individual from an either proprioceptive (e.g., inertial measurement unit (IMU)) or exteroceptive (i.e., MoCap) sensors. The goal of this field of research is to provide personalized support for applications like medicine, human machine interaction (HMI), or sociology. HAR data sets are commonly labeled in classes, such as when individual changes its gate or position, while the nature of the data is obviously continuous because the type of gate gradually changes with the walking speed, for instance. Exemplary records are UCIHAR by

Anguita et al. [Ang+13], MHAD by Ofli et al. [Ofl+13], and WESAD by [Sch+18], and an exhaustive overview of multi-modal data sets and their applications was given by Vrigkas et al. [Vri+15].

UCIHAR comprises 30 individuals of different ages who performed six activities (walking, walking upstairs/downstairs, sitting, standing, laying) using a smartphone's IMU on the waist for recording. Therefore, the data set contains linear acceleration and angular velocity data at a constant rate of 50 Hz.

WESAD is a multi-modal data set for stress and affect detection in humans comprising time series data utilizing electrocardiogram, respiration, accelerometer, blood volume pulse, electrodermal activity, and skin temperature with different update rates for each sensor. The data set contains the activities baseline, amusement, stress, and rest for 15 individuals.

The Berkeley Multimodal Human Action Database (MHAD) consists of temporally synchronized data from an optical motion capture (MoCap) system, multi baseline stereo cameras from multiple views, depth sensors, accelerometers, and microphones to analyze human poses and motions. The data set contains recordings of 12 individuals who performed 11 different actions like jumping, bending, sitting, and waving.

6.1.2.5 Autonomous Ground Vehicle (AGV)

AGV data sets exists with single and multiple sensing modalities that are provided by the robotics and autonomous systems community, but none of them were found to be sufficiently labeled. Most data set annotations are suited for object recognition and manipulation, scene understanding, or SLAM, which would be sufficient for downstream tasks but not acceptable for this work.

Autonomous driving is the most vibrant application of AGVs, and the most comprehensive data sets are described in this section. Noticeable autonomous driving data sets with numerous cameras and LiDARs that also include a RADAR modality are the **KITTI** data set by Geiger et al. [Gei+13] and **nuScenes** by Caesar et al. [Cae+19]. The **Drive&Act** data set by Martin et al. [Mar+19] is a multi-modal benchmark for action recognition in automated vehicles with videos (NIR, depth and color) and motion capturing of the interior. **FieldSAFE** by Kragh et al. [Kra+17] is the first multi modal data set for obstacle detection in agriculture that comprises color, thermal, and stereo cameras plus LiDAR and RADAR. Another noteworthy highly diverse data set is produced using the game engine **Rockstar Advanced Game Engine (RAGE)** by Richter et al. [Ric+16], which unfortunately, only supports RGB camera data due to the game engines limitations.

Robotic data sets for domestic scenarios are comprised in the **Robot@Home** data set article by Ruiz-Sarmiento et al. [RS+17]. This article describes a data set for benchmarking semantic mapping algorithms through the categorization of objects and rooms. The data set time-stamped observations are gathered by a mobile robot endowed with a rig of four RGBD cameras and a 2D LiDAR. Another data set worth mentioning is **CREATE** by Rouat [Rou18], which contains a multi-modal data set particularly designed for unsupervised learning and generative modeling of sensory data from a mobile robot. It comprises exteroceptive (e.g., stereo cameras and microphones) as well as proprioceptive data (i.e., inertia measurements and battery state) with the goal of identifying statistical regularities and structures in the sensor inputs per modality and across modalities. Because this data set is promoted to be relevant for unsupervised training, it lacks labels with which to interpret the results.

6.1.2.6 Simulator

As highlighted by Beattie et al. [Bea+16], simulators have superior functionality over manually crafted data sets concerning the creation of data sets. Simulators enable the fast generation of large quantities of data as well as error-free annotations and GT data (see Mahendran et al. [Mah+16] or Richter et al. [Ric+16]).[9]

Many simulators originate either from the robotics or the RL community. Nowadays, a variety of simulators are available for RL applications through the standardized OpenAI Gym toolkit by Brockman et al. [Bro+16], which makes data set generation even very feasible. The simulators mentioned in this work that have first-party support are Atari [Bel+13] and MuJoCo [Tod+12], while third-party support exists for ViZDoom [Kem+16] by Savinov et al. [Sav+18] and Gazebo [Koe+04] by Nuin et al. [Nui+19].[10]

Whenever AE-like architectures are introduced prior to downstream applications, they are applied to project the high dimensional input space onto some feature space that gives properties to the application like resilience (see Amini et al. [Ami+18]), explainability (see Yang et al. [Yan+19]), and domain transfer (see Higgins et al. [Hig+17b]). β-VAE approaches are especially evaluated regarding their disentangling nature on simulated data to ascertain whether the generative factors of the simulation can be trained (see Higgins et al. [Hig+17a], Burgess et al. [Bur+18b], or Watters et al. [Wat+19b]). The simulators most commonly used in recent publications are presented in the following paragraphs.

The Arcade Learning Environment (Atari) by Bellemare et al. [Bel+13] is the simulator with the broadest visibility in media. It became famous through the first

[9] Usually, all natures of phenomenon (cont./disc.) can be described. Furthermore, because one has usually access to the whole simulation state, all attributes (regr./class.) of the phenomenon are available at every time.

[10] ViZDoom originally comes with its own RL API, while Gazebo can be interfaced through the OpenAI Gym-like openai_ros framework by Ezquerro et al. [Ezq+].

successful end-to-end RL learning approach by Mnih et al. [Mni+15], who learned actions from raw pixel data. Atari offers either GT states of the game or the visible pixel screen. Higgins et al. [Hig+16] use games like Breakout, Frostbite, or Pole Position, which comprise challenging[11] continuous and discrete state changes, to show certain effects like disentanglement and representation learning.

DeepMind Lab [Bea+16] is a first-person 3D game platform designed for the research and development of artificial general intelligence and machine learning systems. The simulator provides access to the raw pixels and depth values, analog to a RGBD camera, as rendered by the game engine. The simulator's behavior is intended to only return a reward signal when an agent finishes a certain task. However, the simulator has been exploited for continuously moving objects in an agent's view through level generation, to show the domain transfer and disentangled learning of a VAE by Higgins et al. [Hig+17a] as well as in GQN[12] by [Esl+18].

Spriteworld [Wat+19c] is a RL environment that consists of a 2D arena with simple geometric shapes that can be freely moved. This environment was recently developed for (and introduced in) the COBRA agent publication by Watters et al. [Wat+19a] and used by Watters et al. [Wat+19b] for the disentangled feature learning of a VAE. The motivation of the authors, in contrast to the dSprites data set by Matthey et al. [Mat+17], was intended to provide as much flexibility as possible for procedurally generating multi-object scenes while retaining a simple interface on which to generate new data sets. The simulators' sprites come in a variety of shapes that can continuously vary and be discrete in position, size, color, angle, and velocity. The simulator does not handle the physical properties (e.g., collision or interaction) of the sprites but lets them pass above or beneath each other.

VizDoom [Kem+16] is a test platform for RL research from raw pixel data that employs the first-person perspective in a semi-realistic 3D world. The simulator is based on the first-person shooter video game Doom. The experiments were constrained to two scenarios in the original publication by Kempka et al. [Kem+16]: a basic move-and-shoot scenario and a more complex maze-navigation task. However, Ha et al. [Ha+18] use the move-and-shoot task to demonstrate the role of their vision model, as realized by a VAE, which learns an abstract, compressed representation of each observed input frame.

Robotic simulators help robotic engineers rapidly prototype controllers, simulate virtual sensors, evaluate robot designs, supply an architecture for real robot control, and much more.[13] The main difference between this simulator and the aforementioned simulators is the physics engines that try to mimic real world conditions by

[11] Learning the underlying state of the game Pong might be challenging because of the reconstruction issue of small pixelized objects like the ball (c.f. Goodfellow et al. [Goo+16, p. 542]).

[12] The GQN data set was partially produced by the DeepMind Lab simulator [Vio+18].

[13] See Ivaldi et al. [Iva+14] for an overview of robotic simulators and their applications.

means of forces and sensor readings as accurately as possible to tighten the gap to later deployment in real robots.[14] The huge variety of sensing modalities, like cameras, distance, or contact sensors, make robotic simulators an ideal experimental platform for this work. Two established simulators that were recently used to produce data sets are Gazebo [Koe+04] and MuJoCo [Tod+12]. **MuJoCo**, which is shipped with its own physics engine, was designed for the fast and accurate 3D simulation of multi-joint dynamics with contact forces in robotic applications. While recent data sets were produced using this simulator, namely the 3D Shapes data set [Bur+18a] used by [Kim+18] and [Wat+19b], it is neither open source nor free of charge. **Gazebo** is a multi-purpose 3D simulator that offers the ability to simulate populations of robots in complex indoor and outdoor environments. It supports various physics engines (ODE[15], Bullet[16], Simbody[17], and DART[18]), programmatic and graphical interfaces, and is free of charge as well as open source, which makes it suitable for this work.

6.2 Proposed Data Sets

As shown in Section 6.1, in the multi-modal community, it is quite common to model a bi-modal dataset as follows (see [Wan+16; Ngi+11; Suz+17; Ved+17]): The first modality a denotes the raw data, and b denotes the class labels or attributes (e.g., the digits' images and labels as one-hot vector concerning the MNIST dataset). This is a rather artificial assumption and only sufficient when the objective is within a semi-supervised training framework. Real multi-modal data does not have such structure because there are commonly multiple raw data inputs. Unfortunately, only complex multi-modal datasets of heterogeneous sensor setups exist (e.g., [Ofl+13; Uda16; Kra+17]), which makes a comprehensive evaluation of the proposed M^{2}VAE futile. While these data sets truly reveal the capabilities and possibilities of DGMs, they are by no means comprehensible. Furthermore, they only enable the assessment of DGMs by scalar measures, as introduced in Section 7.1. The direct introspection or geometrical interpretation of the learned latent representation is hardly possible because complex data sets require high-dimensional D_{z} (see the discussion in Section 3.3.4.3). Applying a low-dimensional $D_{\mathrm{z}} = 2$ on complex data sets, as done in the examples on the MNIST data set in Chapter 3 is only justifiable when one introduces the concept of DGMs. Low-dimensional latent spaces hardly reflect the true behavior of DGMs in complex data sets. Therefore, two comprehensible data sets that comprise ambiguity, drop-out, and the necessity of fusion (only 6.2.2) but

[14] Closing the reality gap, referred as Sim2Real transfer, is one major challenge (see Weng [Wen19]).

[15] `http://www.ode.org/`

[16] `http://bulletphysics.org/`

[17] `https://simtk.org/home/simbody/`

[18] `http://dartsim.github.io/`

enable the investigation of the latent spaces are introduced in Sections 6.2.1 and 6.2.2.

The creation of a new complex data set involves the subsequent constraints:

- The observations of the phenomena enable the learning of a coherent generative factors. Thus, data sets must not include disjunct subsets, which would lead to torn latent spaces. In other words, there needs to be at least a small mutual information between at least two samples (see Higgins et al. [Hig+16]).
- There needs to be multiple samples for each class, and the samples need to gradually change their attributes.
- The data sets need to be labeled based on their attributes and classes.

An artificial bi-modal data-set with fundamentally differing modalities, which is quite common in robotic scenarios, is introduced in Section 6.2.3: camera and LiDAR. A bi-modal data set consisting of perceptions from a RGB camera and the actions involved in moving this camera around a Rubiks cube is introduced in Section 6.2.4.

Although the naïve consolidation of non-coherent datasets does not meet the conditions for data continuity, as discussed in Section 6.2.5, a consolidation technique is proposed by sampling from the superimposed latent spaces of various uni-modal trained CVAEs in Section 6.2.5. This approach enables the generation of multi-modal datasets from distinct and disconnected uni-modal sets. Furthermore, Section 6.2.5 is used to introduce the entangled-MNIST (eMNIST) data set and proposes a data set alignment technique based on the superimposed CVAEs, which includes the correlation of attributes in addition to the classes (c.f. MNIST+SVHN or MNIST+USPS from Section 6.1).[19]

6.2.1 Exclusive OR (XOR)

The XOR gate-logic as a bi-modal data set is introduced in this section. XOR gate-logic constitutes the virtually most basic and fundamental example for motivating the necessity of non-linear activation functions and hidden layers in NNs (see Goodfellow et al. [Goo+16, Chapter 6.1]). It represents a binary gate-logic with a surjective non-linear functional mapping $f : a \to b$, as shown in Table 6.2. The input a is a binary tuple while b is the binary outcome of the XOR gate. When exactly one of the binary values of a is equal to 1, then the XOR function maps b to 1. Otherwise, b is mapped to 0. The XOR example is commonly introduced as a discriminative model $p(b|a)$ in which a function approximator with θ parameters must train the model $b = f(a; \theta)$ such that it resembles the XOR gate-logic.

[19] All data sets are available via the vae_tools library.

Table 6.2: Truth tables of the exclusive or (XOR) gate logic. Left: Original truth table with input a and ouput b. Right: Truth table with drop-out, with $\emptyset$ representing an unknown value.

		drop-out			
a	b	$\cancel{a}$	b	a	$\cancel{b}$
00	0	$\emptyset$	0	00	$\emptyset$
01	1	$\emptyset$	1	01	$\emptyset$
10	1	$\emptyset$	1	10	$\emptyset$
11	0	$\emptyset$	0	11	$\emptyset$

The XOR gate-logic can also be interpreted by the joint likelihood $p(a, b)$. Therefore, it forms a bi-modal model that can unite the two requested features ambiguity and drop-out in a comprehensible fashion. The features can be derived from the question: Can a be inferred from b, if a experiences a drop-out and vice-versa? Obviously, b can be inevitably inferred from a because this represents the original gate-logic function. However, unambiguously inferring a from b is impossible because $f : a \to b$ is a surjective mapping. For example, when just observing $b = 0$, a could have been 00 or 11.[20]

6.2.2 Mixture of Gaussians (MoG)

The MoG data set, as proposed in this thesis and shown in Fig. 6.1, represents a comprehensible artificial 2D bi-modal data observation. It focuses on ambiguity resolving properties through fusion.

The two modalities (a and b) of MoGs have ten classes $(0, \ldots, 9)$ each. a's observations are organized on a grid where class $(5, 6, 7)$ and $(0, 8)$ result in ambiguous observations by sharing the same mean value. b's observations are organized on a circle where class $(0, 9)$ has the same mean values and, therefore, lead to ambiguous observations. The classes are sampled from a multinomial distribution with equal probability while the bi-variate Gaussian distributions are scaled by a constant variance and translated by their associated mean values.[21]

This rather artificial data set has the purpose of depicting and evaluating the ambiguity resolving properties of the VAEs.[22] However, data about complex multi-modal sensor setups for complementary fusion show similar behaviors because various modalities are rectified to achieve a complete view of the scene (e.g., vision and grope to rectify objects). In that case, various dependencies of the generative pro-

[20] See vae_tools.loader.xor for additional details about the data set.

[21] See vae_tools.loader.didactical_set for additional details about the data set.

[22] It is worth mentioning that this data set and its distribution are very reasonable in the case of feature extractor outcomes like t-SNE by van der Maaten et al. [van+08].

cess, given the class labels as factorized latent state representation $z = (z_0, \dots, z_9)$, are possible. This is mimicked by the MoG data set in a simplified assumption as follows: $p(a, b|z)$, $p(a|z_0 \equiv z_8, z_1, \dots, z_4, z_5 \equiv z_6 \equiv z_7, z_9)$, and $p(b|z_0 \equiv z_9, z_1, \dots, z_8)$. Therefore, this data set has the following features for the classes $(0, \dots, 9)$:

- 0: can only be unambiguously detected if it is observed by a and b while any uni-modal observation results in ambiguous observations with 8 (a) or 9 (b)
- $1, \dots, 4$: can be unambiguously detected by any uni-modal observation
- $5, \dots, 7$: can be unambiguous detected by b but collates if observed by a
- $8, 9$: see 0

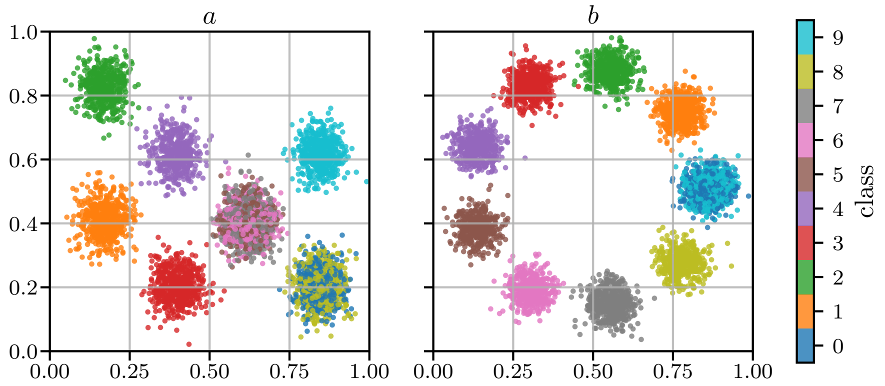


Figure 6.1: MoG input signals for the modalities a and b. The depicted observations are sampled for the corresponding modality for each class.

6.2.3 Camera+LiDAR

Camera+LiDAR is a bi-modal data set collected via the AMiRo [Her+16], which was simulated in the GazeboSim environment.[23] The data set recording involves the following simulated sensors: The simulated camera was configured like the AMiRo's OmniVision® OV5647 camera module with a horizontal FOV of 53.5° at a frame size of 640×480 pixels. The LiDAR was configured like the Hokuyo URG-04LX [Chr00].

Three different object classes that differ in the class attributes color ∈ (red, green) and shape ∈ (cylinder, box) were placed in front of the sensor setup. The object combinations ((box, red), (cylinder, red), (cylinder, green)) exists with ten varying

[23] See Appendix B.2.11 for additional details about the evaluation platform.

diameters $[.1, .2, \ldots, 1.]$ m. Every object combination with every diameter was sampled 300 times from the area of overlapping frustra of both sensors and the rotation of the object, resulting in 10·3·300=9,000 samples.[24]

The classes of objects were chosen with the following motivation (see Fig. 6.2 to follow the argumentation):

- (box, red): is unambiguously detectable by the LiDAR because it is the only box shaped object. It is ambiguous concerning the camera because there are two red objects with different shapes.
- (cylinder, green): is unambiguously detectable by the camera, because it is the only green object. It is ambiguous concerning LiDAR because there are two cylindric shaped objects
- (cylinder, red): is ambiguous concerning both sensors alone but can be unambiguously detected if the camera and LiDAR perceptions are fused.

Therefore, the collection of observations represents a complex, but still comprehensible, data set that addresses ambiguities and the necessity of fusion.

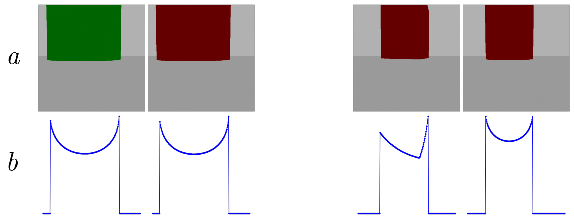


Figure 6.2: Observations of objects via camera and LiDAR in the GazeboSim environment with ambiguous observations concerning shape (left) and color (right).

6.2.4 Rubiks

The rubiks data set is intentionally designed to be a minimalistic real-world simulator for curiosity-driven reinforcement learning (RL) tasks. However, it can be used to sample a bi-modal data set consisting of an RGB-frame → action → RGB-frame → . . . sequence because of its closed-world assumption. The RGB-frame shows an unscrambled Rubiks cube, which can be positioned in 25 different poses. The cube can be inspected from three different viewpoints, which can be approached by 6 different action commands.[25] Two viewpoints are on opposing sites (left/right VP).

[24] See vae_tools.loader.camera_lidar for additional details about the data set.

[25] See `https://github.com/tik0/rubiks-dataset` for the real-world simulator documentation and demonstration videos.

They are redundant and do not add extra information about the state of the cube when they are consecutively observed. One viewpoint (mid. VP) is perpendicular, which reveals the state of the cube if it is observed with one of the other viewpoints.

To become a real-world simulator, 25 RGB frames with a resolution of 40×30 were recorded using an Intel® RealSense™ D435 mounted on a Franka Emika Panda seven axis robot arm from every Rubiks cube's view-point and pose. The positions of the cube were slightly changed to introduce pose-noise while every viewpoint was accurately set up by the robot arm and, therefore, showed a constant background image. As depicted in Fig. 6.3 by choosing a Rubiks cube pose and an action, the resulting frames are sampled from the recorded images. The data-set is constructed by sampling 10,000 pose-action tuples from the simulator.[26]

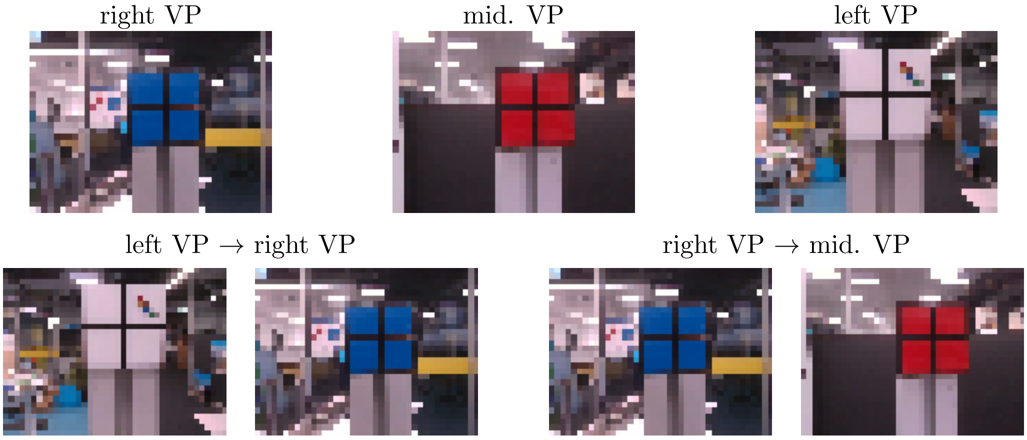


Figure 6.3: The Rubiks data set. Top: All viewpoints (VPs) from one state of a 4×4×4 Rubiks cube. Bottom: Two movements and sampled VP images concerning the according state.

6.2.5 eMNIST

As discussed in Section 6.1, various approaches to creating or consolidating new multi-modal data sets exist. This ranges from naïve class alignment (e.g., MNIST+SVHN) and splitting individual samples (e.g., divided MNIST) to PCA based energy conservation (i.e., MM-MNIST), which are related to the topic of manifold alignment in general.[27] However, none of these techniques involve the

[26] See vae_tools.loader.rubiks for additional details about the data set.

[27] See Wang et al. [Wan+11] for an overview of manifold alignment and Wang et al. [Wan+09] for unsupervised approaches in particular.

consideration of the nature of true multi-modal observations, which motivates the data set generation by DGMs.

Section 4.2.3 contains an explanation of the chain of reasoning that led to requirements for true multi-modal data sets about observable real-world phenomena. Furthermore, a consolidation technique that obeys the requirements of Section 4.2.3 using CVAEs for disjunct data sets is proposed in Section 6.2.5.1 by the author. Finally, Section 6.2.5.2 contains an introduction to one data set called eMNIST that was built from MNIST and FMNIST

6.2.5.1 Generating true Multi-Modal data sets by Collation

In the following section, a generative technique that generates new multi-modal datasets given different uni-modal data sets is proposed. A valuable property of the VAE's learned posterior distribution is that it matches the chosen prior distribution quite sufficiently if the observations were drawn from a similar distribution (see Fig. 3.6). This characteristic can be particularly found in the conditional variational autoencoder (CVAE) [Soh+15a] because its training is supported by the GT labels of the observations. Thus, the CVAE builds a non-related posterior distribution for each class label, and every class distribution closely matches a chosen prior distribution. Furthermore, the idea of β-VAE [Hig+17b] is adopted because it learns disentangled and factorized latent representations. Combining the properties of both approaches allows the superimposing of latent manifolds from various uni-modal encoders that were trained by a β-CVAE. This approach closely follows the idea of manifold alignment in a semi-supervised fashion but without the expense of manifold registration. Instead, every single data set is projected onto a similar factorized distribution. Finally, correlated multi-modal samples can be drawn by sampling from the prior, which operates all CVAE decoders.

6.2.5.2 eMNIST

To test the approach, MNIST by LeCun et al. [LeC+98] and FMNIST by Xiao et al. [Xia+17] are consolidated to an entangled-MNIST (eMNIST) data set as follows: First, β-CVAEs were trained for each data set (see Appendix B.2.4 for the training setup). Then 60,000 samples were drawn from the prior distribution (i.e., $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$) and from the equally distributed multinomial distribution over all classes C. These samples were then fed into the β-CVAEs decoder networks to generate the observation tuples $a \sim p_{\theta_a}(a|\mathbf{z}, C)$ and $b \sim p_{\theta_b}(b|\mathbf{z}, C)$. To avoid artifacts in the generation process, latent samples were only obtained from within 2σ of the prior distribution.[28]

[28] See vae_tools.loader.emnist for additional details about the data set.

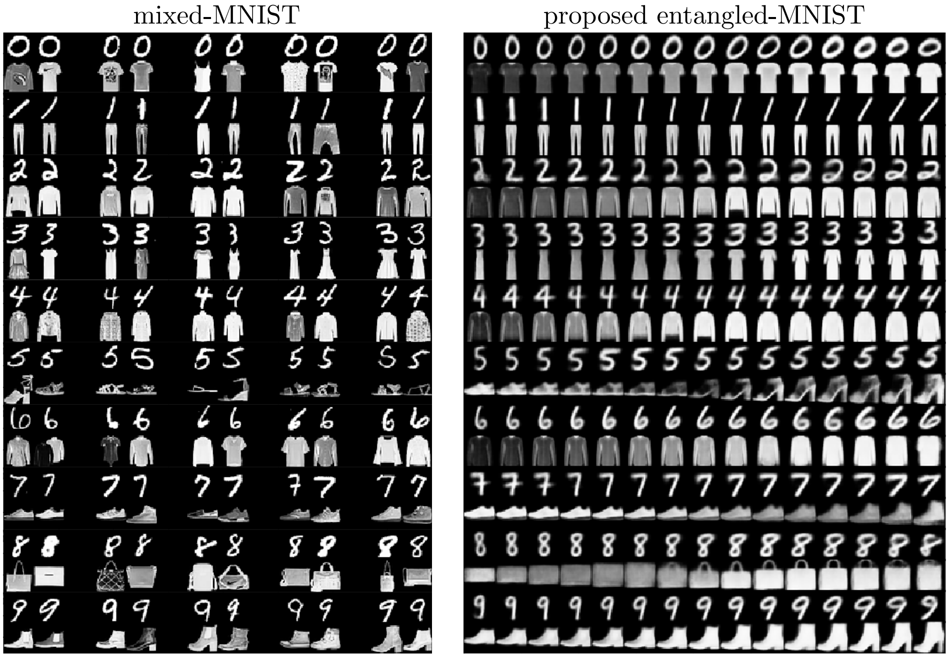


Figure 6.4: Depiction of naïve mixed-MNIST (left) vs. proposed entangled-MNIST (eMNIST) (right). mixed-MNIST is pairwise plotted with the closest match of MNIST digits according to the mean-squared-error. The corresponding fashion-MNIST samples show no continuity or correlation, despite the intended class correlation. eMNIST shows the desired entanglement while sweeping along a single latent space dimension.

Figure 6.4 contains a comparison of the naïve class-wise consolidation results, called mixed-MNIST, as applied in virtually all other publications, to those of the proposed technique. Although mixed-MNIST shows no correlation between the attributes of digits and fashion pieces, eMNIST shows a coherent correlation between the stroke width of MNIST with the brightness of FMNIST (e.g., first row of 0 and t-shirt) or skewness with style (e.g., the last row of 9 and shoe).

These style variations look similar to the approach by Yu et al. [Yu+17], who generated new fashion styles by stratified-like sampling between two points in the latent space, which were learned by the VAE. However, the proposed approach is fundamentally different because it tackles the generation of attribute-correlated multimodal data sets by consolidating disjunct data sets by CVAEs.

It is worth mentioning that instead of choosing the decoder networks to generate new but likely blurry images, one could also perform a 1-NN search of the prior samples z to the actual training set embeddings. Then one could find the associated original training set samples to build up a new correlated data set consisting of crisp samples.[29] However, this approach might not be sufficient because virtually no real-world data set contains samples that gradually change their attributes.[30] Therefore, using a 1-NN search to sort the original data set becomes futile.

6.3 Discussion and Choice of Suitable Data Sets

The goal of multi-modal learning is usually to improve discrimination or regression accuracy by obtaining complementary information from multiple modalities. The particular purposes of DGMs is to learn a refined representation of raw and complex observations that suits downstream applications.[31] The scientific focus of multi-modal DGMs, on the contrary, is mainly dedicated to modality exchange. These objectives are different from the goal of this study. Furthermore, publications that particularly investigate the learning of coherent latent spaces that handle modality dropout remain unknown to the author.

Real world data sets like MIRFLICKR, CelebA, and ImageNet may have differing dimensions or structures and include very generic object images. The generation of the generic object images is still a difficult task and demands the application of further techniques like GANs. However, the latent representation suffers under these techniques, which is why complex data sets do not serve the research of this thesis. Therefore, only comprehensibly multi-modal data sets are used in this section. Thus, the datasets from Section 6.1, divided MNIST (2), divided MNIST (4), and MNIST+SVHN, are applied in the following experiments to conduct comparable and competitive evaluations.

All proposed data sets from Section 6.2, on the contrary, will be applied to evaluate the M^{2}VAE because they serve the purpose of this thesis (i.e., fusion during observation ambiguities and drop-out). Finally, the data sets are comprised as shown in Table 6.1 in Table 6.3.

[29] The 1-NN search technique is commonly used for visualization when DGMs are trained on feature embeddings of an original data set.

[30] A similar argumentation was performed by Yu et al. [Yu+17] to motivate the necessity of augmenting a data set with generated samples from a VAE's decoder.

[31] Although another rising objective of DGMs is the disentanglement learning of the underlying generative factors, its application in multi-modal data remains unknown to the author.

Table 6.3: Overview of proposed multi-modal data sets in Table 6.1. eMNIST has the tick for src. in brackets because the samples were artificially generated. ✓: true, ·: false, –: not applicable

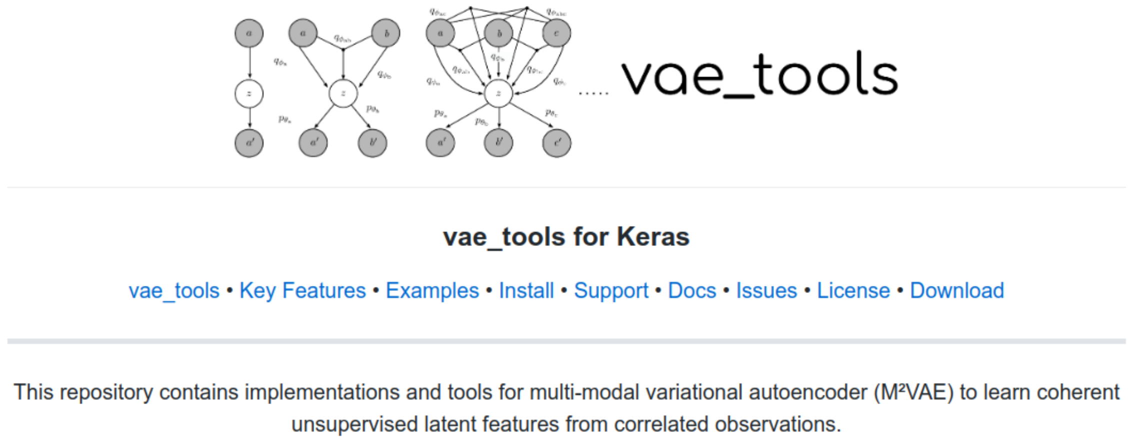
	natural			multi-	nature		purpose		corr.	
data set	src.	corr.	S	modal	cont.	disc.	regr.	class.	attr.	class
XOR	·	✓	·	✓(2)	·	✓	✓	·	✓	·
MoG	·	✓	·	✓(2)	✓	·	·	✓	·	✓
Camera+LiDAR	·	✓	·	✓(2)	✓	✓	✓	✓	✓	✓
eMNIST	(✓)	✓	·	✓(2)	✓	✓	✓	✓	✓	✓
Rubiks	✓	✓	✓(1)	✓(2)	✓(1)	✓(1)	·	✓	✓	·

7 Metrics, Evaluations, and Results

The purpose of this chapter is to confirm that

1. the ambiguous modality problem, indeed, occurs and is remedied by the proposed method,
2. a shared representation of different modalities and an implicit manifold alignment is obtained, and
3. the shared representation is coherent during modality dropout and ambiguous observations.

All experiments and results were conducted using a python library called vae_tools that was written by the author. It uses the TF2 library by Abadi et al. [Aba+15] with the functional Keras API as the backend.[1] It also enables the automated construction of the M²VAE's DNN architecture as proposed in Section 5.2.4, with the objective from Section 5.2.3 by simply specifying the desired encoder and decoder networks for each modality. The vae_tools library also features a high degree of customizability to cover any architectural choices besides the M²VAE. Figure 7.1 shows the logo and landing page of the library, as well as the introduction structure to the library.



vae_tools for Keras

vae_tools • Key Features • Examples • Install • Support • Docs • Issues • License • Download

This repository contains implementations and tools for multi-modal variational autoencoder (M²VAE) to learn coherent unsupervised latent features from correlated observations.

Figure 7.1: Landing page of the vae_tools library.

[1] See Table B.1 for the whole software suite and corresponding versions.

First, scores and metrics to ascertain high fidelity measurement techniques that capture the desired features are revisited and evaluated in Section 7.1. Finally, an evaluating about the impact of the hyperparameters on the M^{2}VAE, an ablation study, and an analysis against competitive multi-modal DGMs are presented in Section 7.2.

7.1 Scores and Metrics

This section contains an overview of the metrics applied in multi-modal learning and a discussion of their proper applications to the M^{2}VAE. The applied metrics from the publications in Table 4.1 are revisited and extended in Section 7.1.1 via additional noteworthy metrics. The application and selection of proper metrics is discussed in Section 7.1.2.

7.1.1 Scores and Metrics in the Wild

The log-likelihood is pursued as an objective by means of the ELBO using the approaches within this thesis. However, one can estimate the true bound of the model using Monte Carlo (MC) sample approximation (see Burda et al. [Bur+15]). The generalized conditional log-likelihood $\log p\Big(\hat{\mathcal{M}}|\mathcal{M},\tilde{\mathcal{M}}\Big)$ is introduced to evaluate all variations of observations. It can be approximated as follows:

$$\log p\Big(\hat{\mathcal{M}}|\mathcal{M},\tilde{\mathcal{M}}\Big) \overset{\text{C.4}}{\geq} \int q\Big(z|\mathcal{M},\tilde{\mathcal{M}}\Big) \log p\Big(\hat{\mathcal{M}}|z\Big)\,\mathrm{d}z \overset{\text{MC}}{\simeq} \frac{1}{N}\sum_{n=1}^{N} \log p\Big(\hat{\mathcal{M}}|z_n\Big) \tag{7.1}$$

with $z_n \sim q\Big(z|m,\tilde{\mathcal{M}}\Big)$.[2] According to Burda et al. [Bur+15], the sample approximation of the log-likelihood approaches the true log-likelihood if the number of samples is sufficiently large. The lower bound, on the contrary, is an unbiased estimator and is, therefore, not biased concerning the number of samples.

The inception score (IS) proposed by Salimans et al. [Sal+16] offers a way to quantitatively evaluate the quality of generated image samples. Although the log-likelihood is often used as the de facto standard for DGMs, it is very susceptible to noise and does not consider the semantic information in the generated samples. IS was motivated by two considerations: First, the conditional label distribution of the samples containing meaningful objects should have low entropy, and second, the variability of these samples should have entropy. The score can be calculated as follows:

$$\mathrm{IS}(p_\theta) = \exp\Big(E_{\hat{\mathcal{M}}\sim p_\theta}\,\mathrm{D}_{\mathrm{KL}}\Big(p\Big(\mathbf{y}|\hat{\mathcal{M}}\Big), p(\mathbf{y})\Big)\Big), \tag{7.2}$$

[2] For the common uni-modal log-likelihood, choose $\mathcal{M} = a$, $\hat{\mathcal{M}} = \hat{a}$, and $\tilde{\mathcal{M}} = \emptyset$, for example. This can be extended to a conditional log-likelihood by choosing $\tilde{\mathcal{M}} = b$.

with $p(\mathbf{y}|\cdot)$ being the Inception Net classifier by Szegedy et al. [Sze+16] trained on the ImageNet data set (see Section 6.1.2.1).

The Fréchet inception distance (FID) by Heusel et al. [Heu+17] provides an alternative approach to IS that involves quantifying the quality of generated image samples. First, the true and generated samples are embedded into a feature space given by the penultimate layer of Inception Net. Then a multivariate Gaussian distribution is estimated for both the generated data and the true data. The Fréchet distance between these two Gaussians distributions is calculated as follows:

$$\mathrm{FID}(\mathcal{M}, p_\theta) = \|\boldsymbol{\mu}_{\mathcal{M}} - \boldsymbol{\mu}_{p_\theta}\|_2^2 + \mathrm{Tr}\Big(\Sigma_{\mathcal{M}} - \Sigma_{p_\theta} - 2(\Sigma_{\mathcal{M}}\Sigma_{p_\theta})^{1/2}\Big), \tag{7.3}$$

with $\boldsymbol{\mu}_{\mathcal{M}}, \boldsymbol{\mu}_{p_\theta}, \Sigma_{\mathcal{M}}, \Sigma_{p_\theta}$ being the mean and covariances of the estimated multivariate Gaussians, respectfully.

Divergencies are a family f of functions $\mathrm{D}_f(p\|q)$ that measure the distance between two probability density functions (PDFs) p (i.e., the ground truth) and q (i.e., the approximator). The most prominent divergence function in this thesis is the KLD, which was used to compare the latent distributions during the training of the DGMs (see chapters 3 and 5). However, the JSD[3] was chosen over KLD[4] for three reasons: First, KLD is defined on $\mathbb{R}^+$ and, therefore, unbounded, while JSD is defined on the open interval $[0, 1] \in \mathbb{R}$. Second, JSD can be used to compare the embedding between encoders that were trained in different ways. Third, KLD is no true metric $(\mathrm{D}_{\mathrm{KL}}(p\|q) \neq \mathrm{D}_{\mathrm{KL}}(q\|p))$ while JSD is at least symmetric $(\mathrm{D}_{\mathrm{JS}}(p\|q) = \mathrm{D}_{\mathrm{JS}}(q\|p))$.

The computation of a divergence is only analytically possible for particular PDFs (see C.2.5), but can be approximated for any distribution using MC as follows:

$$\mathrm{D}_{\mathrm{JS}}(p\|q) = {}^1\!/\!_2(\mathrm{D}_{\mathrm{KL}}(p\|p_m) + \mathrm{D}_{\mathrm{KL}}(q\|p_m)) \text{ with } p_m = {}^1\!/\!_2(p+q) \tag{7.4}$$

$$\mathrm{D}_{\mathrm{KL}}(p\|p_m) \overset{\mathrm{MC}}{\simeq} \frac{1}{N}\sum_{n=1}^{N} \log\frac{p(z_n)}{p_m(z_n)} \text{ with } z_n \sim p. \tag{7.5}$$

p and q can be two different encoder networks of different modality sets for which the embedding is compared by the JSD.

Correlation measures are essential statistical tools used to describe the relationships between two variables as descriptive statistics [Che+02]. However, the quality of the relationship description is highly susceptible to the data's representation, its nature, and moreover, the correlation measures itself. Many different parametric and non-parametric techniques for investigating the correlations between two variables concerning their linear (e.g., Pearson's ρ correlation), monotonous (e.g., Kendall's τ rank correlation), distance (e.g., overview by [Yar+14]), information-based (e.g., MIC by [Res+11]), and canonical (CCA by Hotelling [Hot36]) relations, for example.[5]

[3] i.e., a symmetrized and smoothed version of the KLD (see Lin [Lin91])

[4] See appendix C.2.3 for further information about KLD

[5] see de Siqueira Santos et al. [de +14] for an exhaustive list about correlation measures

Since the advent of DNNs, neural networks have found their way into the correlation analysis of raw and complex data. In fact, a DNN's purpose is nothing but seeking and learning the correlation between the input and output. In particular, the non-linear activation functions in the hidden layers enable a DNN to correlate data of any nature, and its capabilities are only limited by the width and depth of the layers (i.e., the capacity of the DNN). Therefore, so called bottleneck or hourglass DNN architectures, like AE or VAE, are commonly applied to create simplified representations of data, which can then be analyzed using known correlation measures. The Correlational Neural Network (CorrNet) by Chandar et al. [Cha+16], for instance, explicitly formalizes a training objective that maximizes correlation in the represented space.

The canonical correlation analysis (CCA) by Hotelling [Hot36] is an extension of the common correlation measure that uses a representation learning technique for finding correlations in multi-variate variables, even those of different dimensionality. Ordinary correlation analysis depends on the coordinate system in which the observations are described. This means that there may be a very strong correlation between two multidimensional signals, even if this relationship may not be visible through ordinary correlation analysis. The CCA puts the linear projection of the observations into a coordinate system that is optimal for correlation analysis. CCA, as described by Hotelling [Hot36], identifies the projection vectors w_1 and w_2 that maximize the Person's ρ correlation between the projected pair $\left(w_1^\intercal o_1, w_2^\intercal o_2\right)$ of the observation vectors (o_1, o_2).

$$\rho_{\mathrm{CCA}}(o_1, o_2) = \frac{\mathrm{E}\left[w_1^\intercal o_1 \; o_2^\intercal w_2\right]}{\sqrt{\mathrm{E}\left[w_1^\intercal o_1 \; o_1^\intercal w_1\right] \mathrm{E}\left[w_2^\intercal o_2 \; o_2^\intercal w_2\right]}} \tag{7.6}$$

The maximum of ρ_{CCA} concerning w_1 and w_2 is the maximum canonical correlation.[6]

The classical CCA only facilitates linear projections, which are not commonly applicable to raw high-dimensional data like images. Therefore, non-parametric extensions were developed, as described by Michaeli et al. [Mic+15]. Further extensions to CCA methods that are augmented by DNNs are as follows: Andrew et al. [And+13] presented a deep canonically correlated autoencoders (DCCAE) that learns a nonlinear transformation of two views of data such that the resulting representations are linearly correlated by regularizing the total correlation. Furthermore, they extended their approach to a VCCA that derives an ELBO from the log-likelihood that suits the CCA framework [Wan+16].

Downstream scores refers to application of regression or classification tasks to the learned representation as input. As briefly explained in Section 6.1.2.6, using a

[6] Further information of finding subsequent canonical correlations can be found in Borga [Bor01] and Uurtio et al. [Uur+18].

DNN as feature extractor facilitates even comprehensible approaches to perform well on complex data. Because VAEs tend to create linearizable representations of the data (see Section 3.3.4.3), one can apply deterministic linear classifiers and measure statistics that are comparable among different approaches, like precision, recall, or F1 score.[7]

7.1.2 Discussion and Choice of Suitable Metrics

The calculation of the log-likelihood is the de facto standard if one wants to evaluate a VAE because it renders the actual DGM's objective. However, the value of the log-likelihood is not comparable between architectures or HP sets, and it does not give any clue about the practicability of the latent space embeddings or generated samples.

For the IS, Salimans et al. [Sal+16] found that this score does correlate well with scores from human annotators. However, as mentioned by Lucic et al. [Luc+18], the drawbacks of IS include insensitivity to the prior distribution over labels and not being a proper distance measurement. Furthermore, the FID is more robust to noise than IS.

The divergence and correlation measures can be directly applied to the latent space embeddings and render their coherency during modality drop-out. However, the correlation scores always assume some consecutiveness between samples, which make them hardly applicable when no GT is available or the data set, like MNIST, does not have this feature at all. Divergencies, on the contrary, quantitatively evaluate the coherence of the latent space embeddings per sample. However, it needs to be considered that divergencies require an additional sanity check because they always evaluate perfecter coherence, if the model suffers from mode-collapse.

Downstream scores render an actual application purpose of the DGM's learned latent representation. Although there are virtually infinite approaches to further processing the embeddings, it is reasonable to apply a deterministic baseline model. Therefore, the downstream performance is evaluated by means of the accuracy of a Gaussian naïve Bayes classifier.

7.2 Results

The proposed M^{2}VAE is evaluated in this section in various manners to understand its architectures' characteristics and discuss beneficial and disadvantageous

[7] See `https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics` by Pedregosa et al. [Ped+11] for an exhaustive overview of classification metrics.

features. First, the HP dependencies of the proposed DNN architecture from Section 5.2.4 is investigated in Section 7.2.1. Second, the results of an ablation study are presented in Section 7.2.2 using comprehensible data sets while revisiting preceding DGM approaches. Finally, a summarizing evaluation on various data sets and DGM approaches is performed in Section 7.2.3.

7.2.1 Hyperparameter Analysis

The HP dependencies of the M^{2}VAE architecture, as proposed in Section 5.2.4 concerning the training objectives, coherency, and downstream performance are analyzed in this section. The analysis mentioned here was performed on the divided MNIST (2) data set with two modalities because it serves as a baseline data set in most multi-modal DL publications. Section 7.2.1.1 contains an investigation of the learning rate behavior of the M^{2}VAE while Section 7.2.1.2 contains an analysis of the weighting of the mutual losses, dimensionality of the latent space, and the necessity of an additional latent encoder. Every hyperparameter set was trained on five different seeds to maintain a sufficient statistic for evaluation.

7.2.1.1 Learning Rate Analysis

The learning rate (LR) is one of the most important HPs to tune when training a DNN whenever one is facing a new architecture or dataset. Smith [Smi17, Chapter 3.3] described the “LR range test“ technique for estimating reasonable bounds. It is performed by linearly increasing the LR of the optimization algorithm over a few epochs and investigating the changes in the optimizers objective. The steepest change is the base LR, which can be chosen to train the network.

This evaluation is exemplified for the bi-modal M^{2}VAE and the divided MNIST data set in Fig. 7.2. Various M^{2}VAEs with different D_{z} were trained over 20 epochs using the Adam optimizer. Every colored plot shows the average ELBO over five training runs with different seeds and their corresponding standard deviations as error bars.

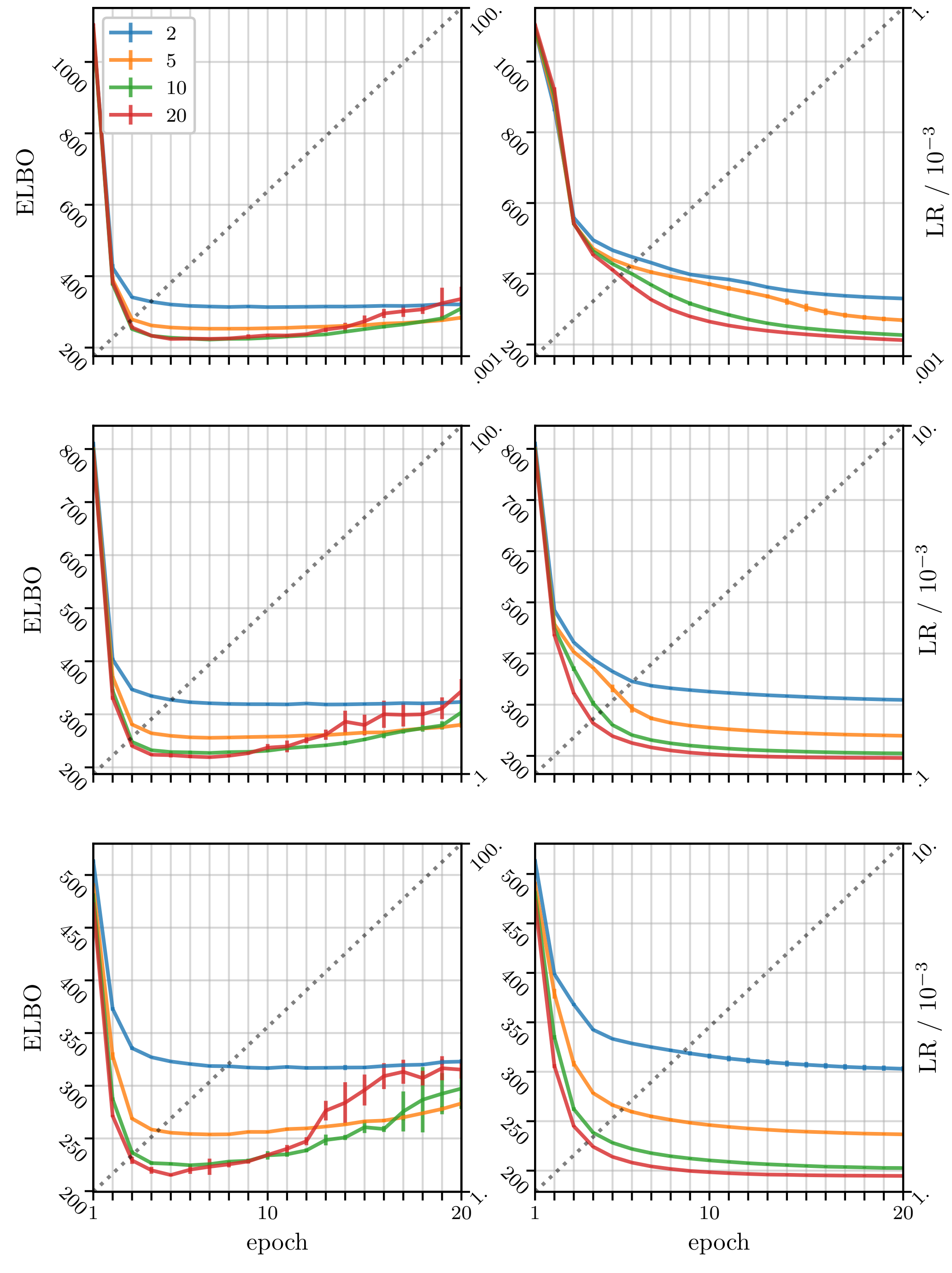


Figure 7.2: Learning rate (LR) evaluation of the bi-modal M²VAE trained on the divided MNIST data set, as proposed by [Smi17] for various $D_{\text{z}} = \{2, 5, 10, 20\}$. See Table B.7 for the M²VAE setup.

The left column shows a broad range sweep for $[10^{-5}, 10^{-1}]$ (top), $[10^{-4}, 10^{-1}]$ (mid.), and $[10^{-3}, 10^{-1}]$ (bot.). All three plots show the characteristic U-curve, as predicted by Smith [Smi17]. First, the optimization performs well in the first epochs when the LR is low. Then it flattens out while the LR increases until it starts to diverge for a too large LR. The mid.- and bot.-figures magnify the divergence phase for the evaluation of another LR range. They reveal a consistent behavior: the higher the latent space dimension D_{z} is, the earlier the divergence toward greater LRs starts.

A closer range sweep for $[10^{-5}, 10^{-3}]$ (top), $[10^{-4}, 10^{-2}]$ (mid.), $[10^{-3}, 10^{-2}]$ (bot.) is performed in the right column. In contrast to the left column, the right column magnifies the convergence-phase.

Mentioning the different ordinates reveals the following: The top figure shows the slowest convergence of the ELBO, starting from 10^{-5}. The bot.-figure shows a reasonable fast convergence for 10^{-3}, which is also the LR parameter of choice for the Adam optimizer and, therefore, chosen for further HP optimization within this thesis. Figure B.2 confirms those findings for the eMNIST data set by showing a similar trend, but with an earlier convergence due to the higher input dimensionality.

The optimization of the ELBO performs best concerning increasing D_{z} on all 120 runs,[8] besides the divergence artifacts. The next sections reveal whether this trend also holds true for the other metrics that are not directly optimized.

7.2.1.2 Correlation Analysis

The following are further important HPs to tune for training the M²VAE:

- D_{z}: the latent dimensionality in the bottleneck
- β_{m}: the mutual loss weighting between the encoder networks
- $D_{\text{enc.}_{\text{ab}}}$: size of the latent bi-modal encoder network $f_{\text{enc.}_{\text{ab}}}$

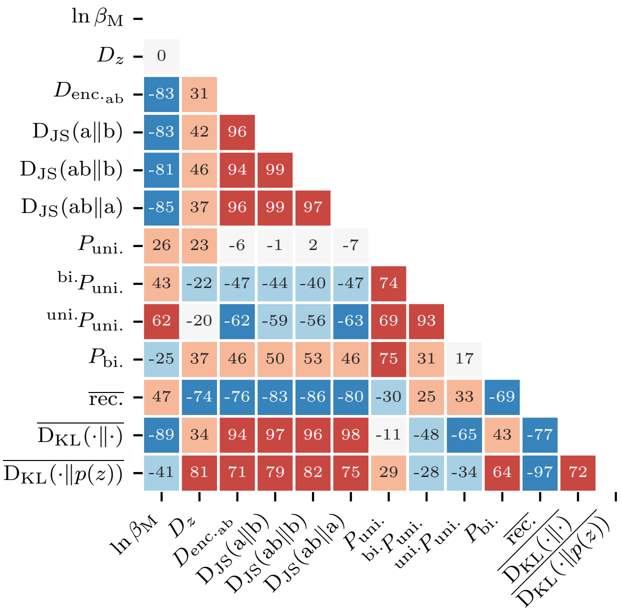
Other architectural choices are listed in Table B.9. The HPs' impacts on the JSD and downstream performance is analyzed in this section by means of a linear naïve Bayesian classifier. To perform a sufficient analysis of the effects, parallel coordinates (pc) are plotted in addition to the Pearson's ρ correlation.[9]

The following statistics were analyzed: D_{JS} denotes the JSDs of the trained representation in the latent space for the corresponding modalities. P denotes the downstream performances of the Gaussian naïve Bayes classifiers that were trained on the projected training sets and evaluated on the corresponding test sets.

[8] Figure 7.2 consists of 5 seeds/plot · 4 plots/sub-figure · 6 sub-figures.

[9] Parallel coordinates are an important tool to augment and substantiate the findings in a multivariate analysis because the Pearson's ρ correlation can only capture linear dependencies. Additional information on pcs was summarized by Inselberg [Ins97].

- $\mathrm{D_{JS}}(\mathrm{a}\|\mathrm{b})$: JSD between modality a and b
- $\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{a})$: JSD between modality ab and a
- $\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{b})$: JSD between modality ab and b
- $P_{\mathrm{uni.}}$: uni-modal downstream performance of a classifier that was trained on the same modality
- ${}^{\mathrm{bi.}}P_{\mathrm{uni.}}$: uni-modal downstream performance of a classifier that was trained on the bi-modal embedding (i.e., dropout performance)
- ${}^{\mathrm{uni.}}P_{\mathrm{uni.}}$: uni-modal downstream performance of a classifier that was trained on each others modality (i.e., exchange performance)
- $P_{\mathrm{bi.}}$: bi-modal downstream performance of a classifier that was trained on the same modalities
- $\overline{\mathrm{rec.}}$: avg. reconstruction objective
- $\overline{\mathrm{D_{KL}}(\cdot\|\cdot)}$: avg. mutual KLD objective, normalized by β_{M}
- $\overline{\mathrm{D_{KL}}(\cdot\|p(z))}$: avg. regularization objective



	$\ln\beta_{\mathrm{M}}$	D_z	$D_{\mathrm{enc.ab}}$	$\mathrm{D_{JS}}(\mathrm{a}\|\mathrm{b})$	$\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{b})$	$\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{a})$	$P_{\mathrm{uni.}}$	${}^{\mathrm{bi.}}P_{\mathrm{uni.}}$	${}^{\mathrm{uni.}}P_{\mathrm{uni.}}$	$P_{\mathrm{bi.}}$	$\overline{\mathrm{rec.}}$	$\overline{\mathrm{D_{KL}}(\cdot\|\cdot)}$	$\overline{\mathrm{D_{KL}}(\cdot\|p(z))}$
$\ln\beta_{\mathrm{M}}$													
D_z	0												
$D_{\mathrm{enc.ab}}$	-83	31											
$\mathrm{D_{JS}}(\mathrm{a}\|\mathrm{b})$	-83	42	96										
$\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{b})$	-81	46	94	99									
$\mathrm{D_{JS}}(\mathrm{ab}\|\mathrm{a})$	-85	37	96	99	97								
$P_{\mathrm{uni.}}$	26	23	-6	-1	2	-7							
${}^{\mathrm{bi.}}P_{\mathrm{uni.}}$	43	-22	-47	-44	-40	-47	74						
${}^{\mathrm{uni.}}P_{\mathrm{uni.}}$	62	-20	-62	-59	-56	-63	69	93					
$P_{\mathrm{bi.}}$	-25	37	46	50	53	46	75	31	17				
$\overline{\mathrm{rec.}}$	47	-74	-76	-83	-86	-80	-30	25	33	-69			
$\overline{\mathrm{D_{KL}}(\cdot\|\cdot)}$	-89	34	94	97	96	98	-11	-48	-65	43	-77		
$\overline{\mathrm{D_{KL}}(\cdot\|p(z))}$	-41	81	71	79	82	75	29	-28	-34	64	-97	72	

Figure 7.3: Correlation heatmap visualizing Pearson's ρ of varied HPs (i.e., β_{m}, D_{z}, $D_{\mathrm{enc._{ab}}}$) and the resulting statistics. Saturated colors indicate significant correlation ($|\rho| > 60\,\%$).

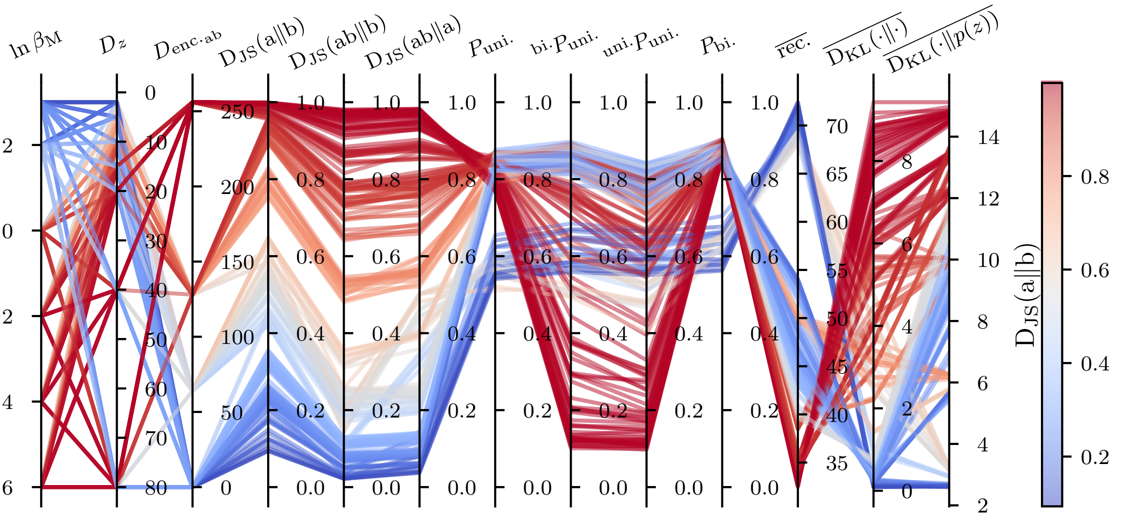


Figure 7.4: Parallel coordinates with varying HPs (i.e., β_m, D_z, $D_{\mathrm{enc}\cdot_\mathrm{ab}}$). The architecture setup is listed in Table B.9. Saturated colors result from overlaying transparent lines.

The JSDs all behave very similarly to each other, which can be seen by the strong correlations in Fig. 7.3 and parallel lines in Fig. 7.4. They show significant negative correlations with the mutual β_M, which is an essential feature of the M²VAE.[10] This means that the coherence between the uni- and bi-modal embeddings can be controlled with a single scalar HP.

D_z has a non-significant, but still noticeable, impact on the JSD. This does not necessarily mean that a low D_z performs better than a high D_z. In fact, a further dependency on $D_{\mathrm{enc}\cdot_\mathrm{ab}}$ can be suggested by analyzing Fig. 7.4: Although the blue lines, which indicate a good JSD, show many crossovers for D_z, they assemble according to the dimensionality of the latent encoder network $D_{\mathrm{enc}\cdot_\mathrm{ab}}$. Furthermore, the JSD naturally degenerates for high dimensions because small deviations in the distributions have increasingly high negative impacts on similarity scores as the dimensionality grows.[11]

The strong correlation of the JSD with $D_{\mathrm{enc}\cdot_\mathrm{ab}}$ is very intriguing because one might think that an additional non-linear network should help boost the performance. A reasonable explanation is that the additional non-linearities might further entangle the latent space of the bi-modal embedding. However, if one leaves them out, then only the linear regression layers f_μ and f_σ dominate the embedding.

[10] $\mathrm{D_{JS}} \approx 0$ denotes a strong overlap of the corresponding distributions, and $\mathrm{D_{JS}} \approx 1$ denotes a poor overlap of the corresponding distributions.

[11] i.e., the curse of dimensionality

The downstream performance shows a contradicting behavior between the pure multi-modal embedding $P_{\mathrm{bi.}}$ and all other uni-modal affected performances. Although a high mutual β_{M} is also positively correlated with uni-modal affected performances, it shows an negative correlation with the bi-modal embedding. In some ways, this can be attributed to the no-free-lunch theorem as follows: A high mutual β_{M} demands the embedding with the most information (i.e., the bi-modal one) to sacrifice itself for the sake of the uni-modal embeddings and to obey the mutual losses.

D_{z} has almost no significant impact on performance. However, it tends to correlate with the pure uni- and bi-modal embeddings, and it slightly anti-correlates with the exchange performance.

$D_{\mathrm{enc.}_{\mathrm{ab}}}$ behaves reciprocally to β_{M}, which can be explained in an analog fashion. A high $D_{\mathrm{enc.}_{\mathrm{ab}}}$ increases the bi-modal encoder's capacity to embed information. Although the uni-modal encoders are not equipped with further non-linear layers that might adapt the more complex bi-modal embedding, the performance degenerates for them.

The JSD and downstream performance was almost done in an analog fashion compared to the training objectives. This is an important investigation because only the training objectives are accessible during the unsupervised training. The JSD and downstream performances can only be determined if and only if (iff) the GT labels are accessible. Furthermore, the reconstruction loss behaves in contradiction to the mutual and regularization losses. This means that one must sacrifice some reconstruction performance for the sake of obtaining a coherent embedding, which is a known behavior of VAEs in general (see "VAEs and GANs" talk by Rosca [Zhu+18]).

The HP recommendations for the proposed M^{2}VAE are as follows: The additional encoder network should be completely omitted ($D_{\mathrm{enc.}_{\mathrm{ab}}}$ = None) because it shows a strong degeneracy of all measures with increasing dimensionality. The mutual β_{M} should be set to be as high as possible because it strongly controls the coherency in the latent space embeddings between all modality subsets. This fact is supported by the left plot in Fig. 7.5, which shows that there is a monotonic correlation between the JSD and β_{M}. A β_{M} that is too high, on the contrary, let the latent space collapse such that the downstream performance degenerated as well. This fact is supported by the three right plots in Fig. 7.5.[12] Therefore, one should choose a β_{M} such that the JSD vanishes to maintain good downstream performance. This recommendation contradicts the findings by Suzuki et al. [Suz+17]. Suzuki et al. [Suz+17] choose very low mutual weightings α, which corresponds to β_{M}, to pursue high likelihood scores. However, the likelihood score is mainly driven by reconstruction loss, which behaves

[12] Note that $\beta_{\mathrm{M}} < 1$ is neglected in this plots because the previous sections already revealed a negative dropout performance.

reciprocal to the coherency, as in the previously mentioned paragraph. Finally, the dimensionality D_z of the latent space should be chosen according to the data set in general. Although Fig. 7.5 shows a drastic drop in downstream performance, this is outside the β_M recommendation.

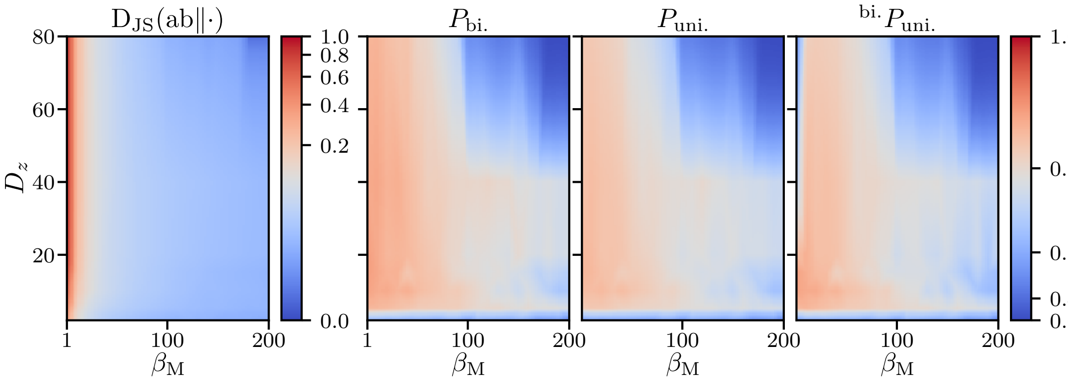


Figure 7.5: Surface plots of the two HPs β_M and D_z.

7.2.2 Ablation Study

In the context of ML, and especially DNNs, ablation studies have been adopted to describe procedures whereby certain parts of a network are removed, to improve the understanding of the network's behavior.[13] Chollet, the inventor of the Keras framework, highlighted ablation studies as crucial to analysis because they serve as way to look into a network's causality.[14]

Section 7.2.2.1 contains an analysis about the M^{2}VAE by systematically removing parts of the objective function.[15] Furthermore, probabilistic PCA (pPCA) is introduced as the linear baseline model of the M^{2}VAE, which demonstrates the necessity of non-linearities in the network's architecture. Section 7.2.2.2 highlights the behaviors of the M^{2}VAE in the case of complementary fusion and the absence of information that is necessary for fusion. Analog to Section 7.2.1, Section 7.2.2.3 contains an analysis on the necessity of the weight sharing approach in the encoder networks, as well as the absence of the mutual KLD loss in the M^{2}VAE's objective function.[16]

[13] The term ablation originates from the experimentally and surgical removal of body tissue to neuropsychological study it effects on a subject.

[14] `https://twitter.com/fchollet/status/1012721582148550662`

[15] This approach results in the M^{2}VAE's predecessor approaches JMMVAE and JVAE.

[16] All architectures in this section were trained w/o an additional latent encoder network (i.e., $D_{\mathrm{enc._{ab}}}$ = None), as recommended in Section 7.2.1.2.

7.2.2.1 XOR Evaluation

This section highlights the advantages of the M^{2}VAE in comparison to its predecessors JMMVAE and JVAE in the most elementary way. It reflects the discussions from Section 5.1 to 5.2 and empirically shows the validity of the proof in Section 5.3. Therefore, the XOR fits into the ablation study framework because the architectures only vary by separate features (see Fig. 5.3). Moreover, the M^{2}VAE is trained with linear activation functions that mimics the probabilistic PCA to respect the motivation behind the XOR history.[17] All architectures for the following results can be found in Appendix B.2.6.

Figure 7.6 depicts the most commonly learned distributions using the various approaches.[18] The proposed M^{2}VAE and the JMMVAE both show mean-seeking behaviors, while only the M^{2}VAE shows the desired property of arranging the embeddings in a sufficient fashion without any confusion. The JMMVAE collates the ambiguous observations from b because it has no feedback about its embedding. Although $q_{\phi_{\mathrm{ab}}}$ arranges all embeddings symmetrically around zero, $q_{\phi_{\mathrm{b}}}$ mean-seekingly puts all its observations to zero on the abscissa as well. This is called mode-collapse and causes confusion in any observation made by $q_{\phi_{\mathrm{b}}}$ because they are reconstructed to 0.

The JVAE was adapted from Ngiam et al. [Ngi+11], while the $\emptyset$ from Table 6.2 was substituted with a placeholder in the case of a drop out. Because zeroing $\emptyset$ would lead to catastrophic results because 0 is part of the observation space, it was reasonably substituted with .5, as the mean value of observations. This led to totally bisected embeddings, as shown in Fig. 7.6, where each single modality no longer shares any information with its bi-modal equivalent. That also demonstrates that NNs are not effective at handling placeholder values in general because these placeholders span up sub-spaces in the NNs' weight distributions, which also occupies unnecessarily additional network capacity. The M^{2}VAE, on the contrary, can handle multi-modal observations, and the absence of modalities by design, which leads to coherent latent space embeddings.

Finally, the pPCA shows an almost complete mode collapse in the latent space. This is due to the linear activation functions, which are not capable of finding the nonlinear relationship between a and b. Therefore, the decoder networks also collate the sigmoidal output function to average the desired reconstructions.

[17] As stated by Lucas et al. [Luc+19], the pPCA by Tipping et al. [Tip+99] has a direct correspondence to a VAE with linear activation functions.

[18] Because the learned representation highly depends on the initial conditions of the DNNs, the scales, offsets, and even arrangements of the modes may vary.

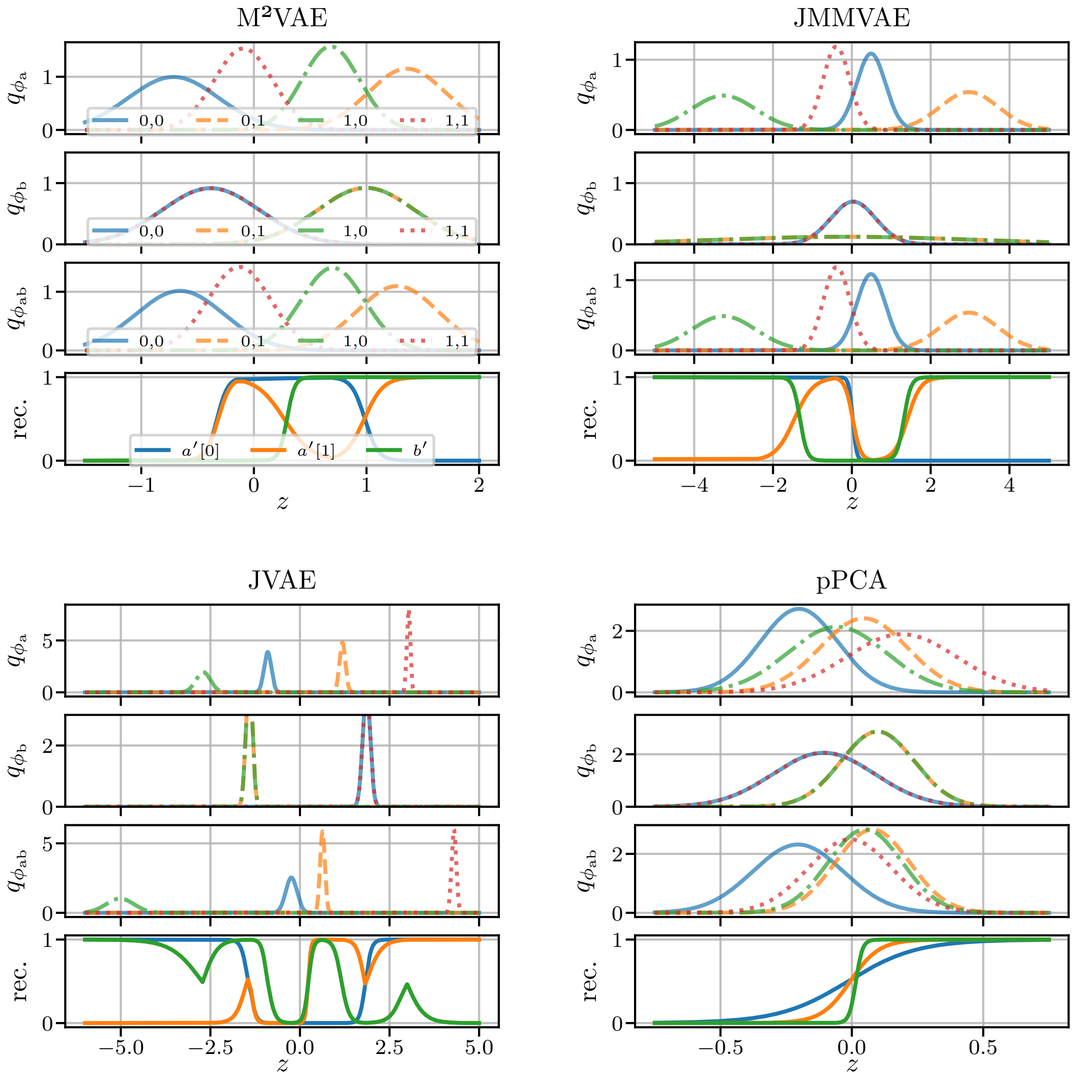


Figure 7.6: Representative embeddings and reconstructions over the latent space z for M²VAE, JMMVAE, JVAE, and pPCA (i.e., linear M²VAE). The legends of the M²VAE plot are valid for all other models' plots.

7.2.2.2 MoG Evaluation

The behaviors of the M²VAE in the case of complementary fusion and the absence of information that is necessary for fusion is highlighted in this section. Therefore, the MoG data set fits into the ablation study framework because the necessary information to form the latent space is ablated during inference. The architectures' HPs can be found in appendix B.2.7.

The M^{2}VAE inherently enforces its encoder networks q_{ϕ_*} to approximate the same posterior distribution, which can be seen by the strong coherence between all embeddings in Fig. 7.7. In the depicted case coherence means that the same observations lead to the same latent embedding: $q_{\phi_{\mathrm{ab}}}(a, b) \approx q_{\phi_{\mathrm{a}}}(a) \approx q_{\phi_{\mathrm{b}}}(b)$. However, this property only holds for non-ambiguous observations. Observations made from classes that are not separable collapse to a common mean in the latent space, which is denoted for the uni-modal cases by (+) and (-). Furthermore, the embeddings show an interesting behavior for samples from class 0: because this class is only ambiguously detectable in the uni-modal case, the encoder networks learn a separable and, therefore, unambiguous embedding if both modalities are present (denoted by (-)).

The depicted behaviors are also rendered by the ELBO, which is the objective for training the M^{2}VAE. This is an intriguing observation because the samples are no longer separable (not even non-linearly) in latent space. The ELBO for the observation goes down (see ($*$) and (/)) and, therefore, gives evidence about the embedding quality and information content. This insight might connect VAEs to the free-energy principle introduced by Friston [Fri10] and might be fruitful in terms of epistemic (ambiguity resolving) tasks, where for instance an unsupervised learning approach could involve the use of the ELBO as a signal for learning epistemic action selection. However, although the ELBO is not accessible during inference, the accessible KLD was plotted as well (i.e., the prior loss $\mathrm{D_{KL}}(q_{\phi_*} \| p(z))$). Friston [Fri10] stated, that the KLD is a value that can be interpreted as the learned complexity of an observation. This quantity, behaves inversely to the ELBO as postulated by Friston, and will be investigated in later studies.

The last row of Fig. 7.7 shows the interaction between the latent embeddings and a single naïve Bayes classifier that was trained on these embeddings. Because one needs three classifiers to classify all permutations of observations ((a, b), (a), (b)), the M^{2}VAE projects all permutations such that only one naïve classifier is necessary. This is an interesting insight because this single classifier reaches the same classification rate (see Table 7.1) as three exclusive classifiers trained on the raw data. Furthermore, the ambiguous observations lie mainly on the decision boundaries of the classifier. This behavior can be attributed to the fact that VAEs naturally project observations onto the prior distribution by maintaining the sampling distribution. In the depicted case, both are Gaussian and, therefore, seem to interact seamlessly with a Gaussian naïve Bayes classifier. However, other multi-modal VAE approaches tend to learn non-coherent latent spaces, which is recognizable by their relatively low classification scores.

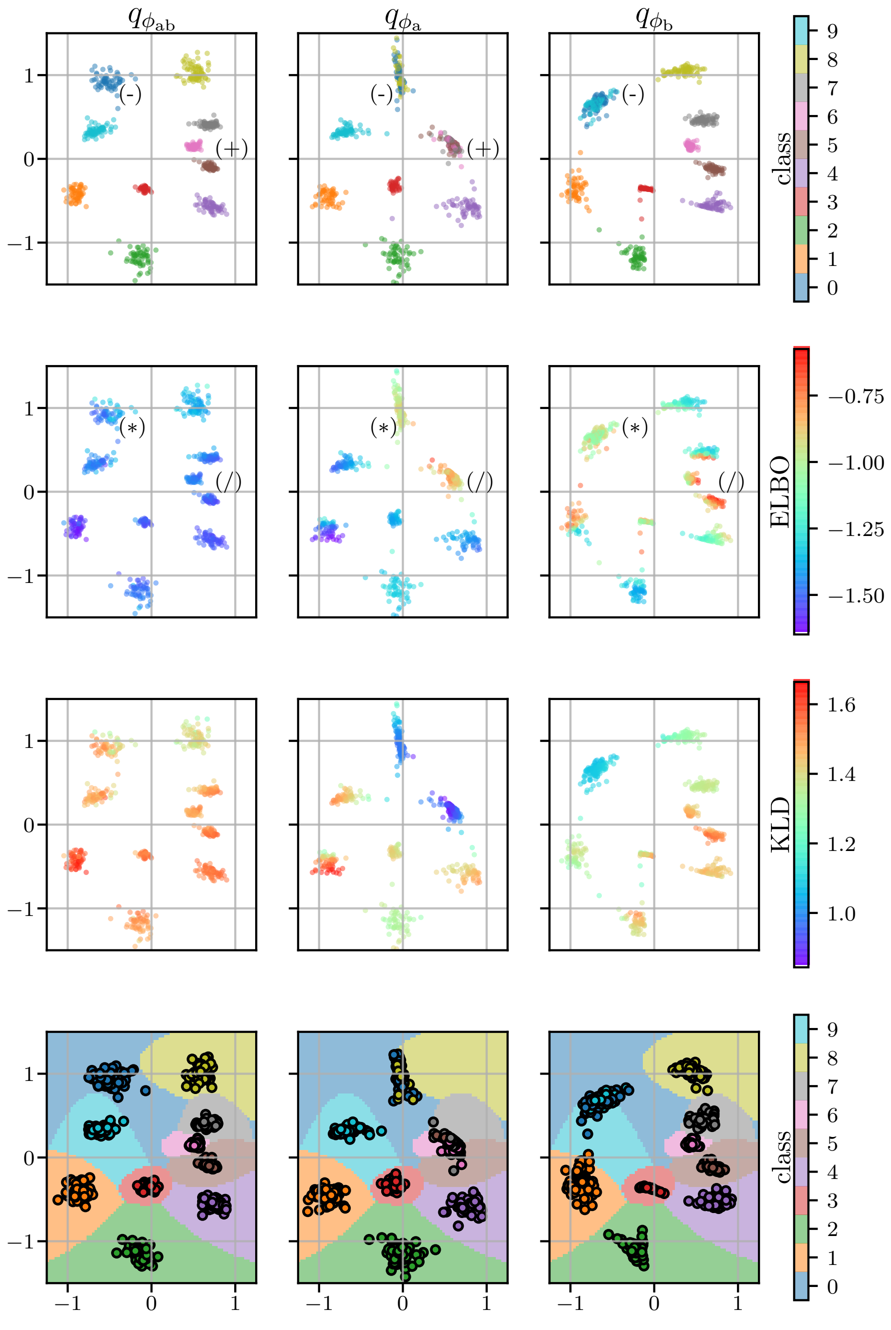


Figure 7.7: 2-dimensional latent space embeddings of the bi-modal MoG test set. Plots from left to right show the embeddings of the encoder networks $q_{\phi_{\mathrm{ab}}}$, $q_{\phi_{\mathrm{a}}}$, $q_{\phi_{\mathrm{b}}}$ and their corresponding observations. Plots from top to bottom show different colorizations: classes, ELBO, KLD, and the decision boundaries of a single naïve Bayes classifier.

The possibility of using just a single classifier on a multi-modal sensor setup that is susceptible to sensory dropout is an outstanding feature of the M²VAE. This could stabilize and optimize future classifying and reinforcement learning approaches, which commonly learn dropout during training such that they learn from the common and coherent latent space.

Table 7.1: Classification score (i.e., the ratio of correctly classified samples to the total number of samples) for the naïve Bayes classification on the raw and encoded data.

input	raw	embedding		
		M²VAE	JVAE	JMMVAE
a, b	.99	**.99**	.99	.99
a	.71	**.71**	.63	.63
b	.90	**.90**	.09	.28

The architecture and training setup of the M²VAE for Fig. 7.7 can be found in Appendix B.2.7. It is worth noticing that the VAEs do not learn the identity function, regardless of their high encoder fan-out ($D_{\mathrm{a}} = 2$ vs. $D = 128$ of the first hidden layer), which can be attributed to the sampling layers and the prior loss in the VAE's bottleneck.

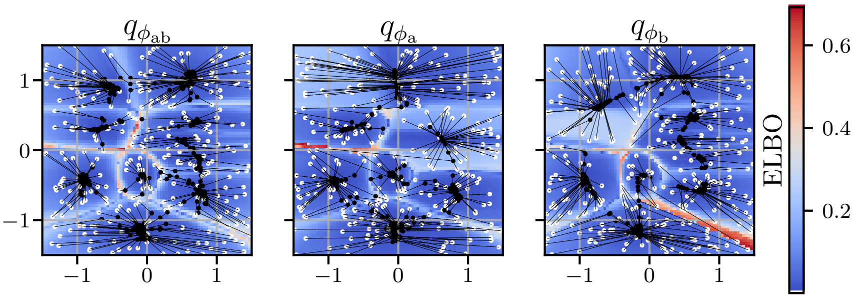


Figure 7.8: Trajectory visualization of re-encoding using the jointly trained bi- and unimodal encoders on the MoG data set with reconstruction loss underlay. White markers denote the initial encoding (looks best on screen).

As mentioned in Section 5.4.1, for the linear separable MoG data set, the M²VAE without the proposed re-encoding loss tends to have a denoising characteristic because it re-encodes any z as a refined version of its own by means of reconstruction loss. This behavior is shown in Fig. 7.8 where the re-encoding trajectories are plotted on the reconstruction loss. One can see naturally learned discrimination boarders of the latent space indicated by high losses that separate clusters' vicinities.[19] Furthermore, initial z values are auto re-encoded and draw the trajectories along their path in latent space. The properties of the various VAE encoders $q_{\phi *}$ during re-encoding show that every observation converges to a fixed-point (i.e., the corresponding clusters' mean values while performing descending steps on the loss manifold).

7.2.2.3 Weight-Sharing and Mutual KLD

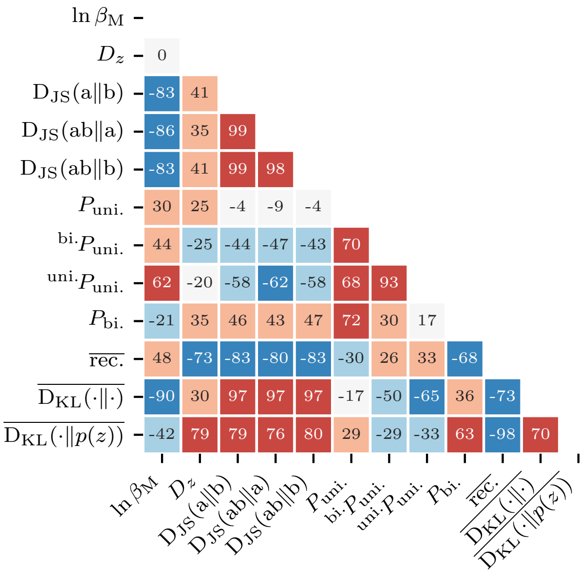
The necessity of weight-sharing, as introduced in Section 5.2.4, to facilitate the training of the full multi-modal subset by one DNN is investigated in this section.[20] The ablation in this analysis removes the weight-sharing such that unique encoder networks are trained for each multi-modal subset. Furthermore, the mutual KLD losses in the M^{2}VAE's objective function, which connects the latent embeddings of all multi-modal subsets, is ablated to analyze its effect. The mutual KLD is of high interest and analyzed in contrast to the weight-sharing to investigate their relationships. It is worth mentioning that the decoder networks are not altered and, therefore, all non-shared-weight encoders use the same decoder network.

The weight-sharing is ablated for the correlation heatmap in Fig. 7.9.[21] The heatmap and the parallel coordinates plot show almost the exact same behavior for the non-shared variant as for the shared variant in Section 7.2.1.2. This is a beneficial coincidence because it means the network complexity can be drastically reduced without any loss in overall performance. Furthermore, Fig. 7.10 shows the validation error over the first 100 epochs for both approaches, taking five different seeds into account. The proposed shared-weights approach also significantly improves in the convergence rate, which can be contributed to the reduced network complexity. Therefore, the shared-weights approach is the method of choice for building the M^{2}VAE.

[19] The reconstruction loss plot w/o trajectories can be found in Appendix B.2.7 for improved traceability.

[20] The network architecture is analog to that in Section 7.2.1.2.

[21] See Fig. B.4 for the corresponding parallel coordinates plot.



	$\ln\beta_\mathrm{M}$	D_z	$\mathrm{D_{JS}}(\mathrm{a}\Vert\mathrm{b})$	$\mathrm{D_{JS}}(\mathrm{ab}\Vert\mathrm{a})$	$\mathrm{D_{JS}}(\mathrm{ab}\Vert\mathrm{b})$	$P_\mathrm{uni.}$	${}^\mathrm{bi.}P_\mathrm{uni.}$	${}^\mathrm{uni.}P_\mathrm{uni.}$	$P_\mathrm{bi.}$	$\overline{\mathrm{rec.}}$	$\overline{\mathrm{D_{KL}}(\cdot\Vert\cdot)}$	$\overline{\mathrm{D_{KL}}(\cdot\Vert p(z))}$
$\ln\beta_\mathrm{M}$												
D_z	0											
$\mathrm{D_{JS}}(\mathrm{a}\Vert\mathrm{b})$	-83	41										
$\mathrm{D_{JS}}(\mathrm{ab}\Vert\mathrm{a})$	-86	35	99									
$\mathrm{D_{JS}}(\mathrm{ab}\Vert\mathrm{b})$	-83	41	99	98								
$P_\mathrm{uni.}$	30	25	-4	-9	-4							
${}^\mathrm{bi.}P_\mathrm{uni.}$	44	-25	-44	-47	-43	70						
${}^\mathrm{uni.}P_\mathrm{uni.}$	62	-20	-58	-62	-58	68	93					
$P_\mathrm{bi.}$	-21	35	46	43	47	72	30	17				
$\overline{\mathrm{rec.}}$	48	-73	-83	-80	-83	-30	26	33	-68			
$\overline{\mathrm{D_{KL}}(\cdot\Vert\cdot)}$	-90	30	97	97	97	-17	-50	-65	36	-73		
$\overline{\mathrm{D_{KL}}(\cdot\Vert p(z))}$	-42	79	79	76	80	29	-29	-33	63	-98	70	

Figure 7.9: Correlation heatmap visualizing Pearson's ρ of non-shared weights, analog to Fig. 7.3. Saturated colors indicate significant correlation ($|\rho| > 60\,\%$).

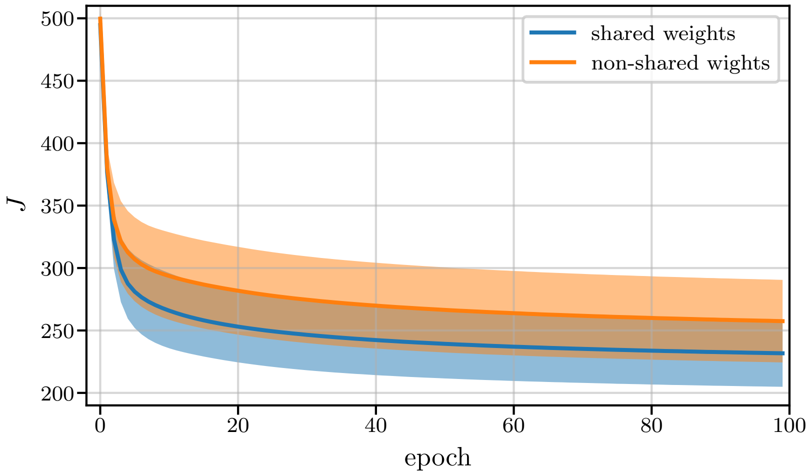


Figure 7.10: Convergence of the validation loss during the first 100 training epochs between the shared and non-shared weights approach. Shaded region highlights the standard deviation σ off all seeds.

The mutual KLD was ablated for the shared and non-shared approach to gather the correlation heatmap in Fig. 7.11.[22] Both approaches show the same behavior, that is, the complete degeneracy of the latent space coherency between the multi-modal subsets. Even the shared weights approach has no impact on the performance, because the networks f_μ and f_σ are not shared in any of the cases. This is an important outcome because it reveals that the mutual loss is the driving feature of the coherency between the multi-modal subsets' embeddings

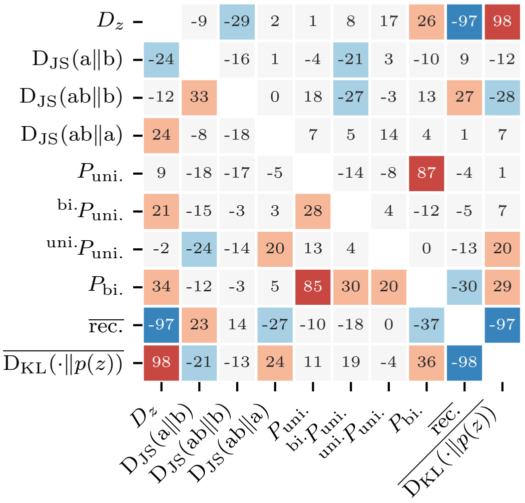


Figure 7.11: Correlation heatmap visualizing Pearson's ρ of shared (triu.) vs. non-shared weights (tril.) with $\beta_{\mathrm{M}} = .0$. Saturated colors indicate significant correlation ($|\rho| > 60\,\%$).

7.2.3 Competitive Evaluation and Other Data Sets

In contrast to the previous sections, this section contains the results of an evaluation of the relatively complex datasets. The bar charts in Fig. 7.12 summarize the results for all full- and lesser-modal[23] evaluations.[24] The architecture setups for the M^{2}VAE are summarized in appendix B.2.9, while the HPs for the other architec-

[22] See Fig. B.6 and Fig. B.4 for the corresponding parallel coordinates plot.

[23] The term "lesser-modal" refers to setups, where less modalities were used during inference than in the full multi-modal observation.

[24] Because the introspection of high dimensional latent spaces does not add any value to this evaluation, bar charts were chosen for the sake of comprehensibility.

tures were adopted from the corresponding publications.[25] The VCCA by Wang et al. [Wan+16] was added as one competitor of the DGM approaches on multi-modal data because the authors also explicitly studied the coherence of the latent embedding based on correlation.

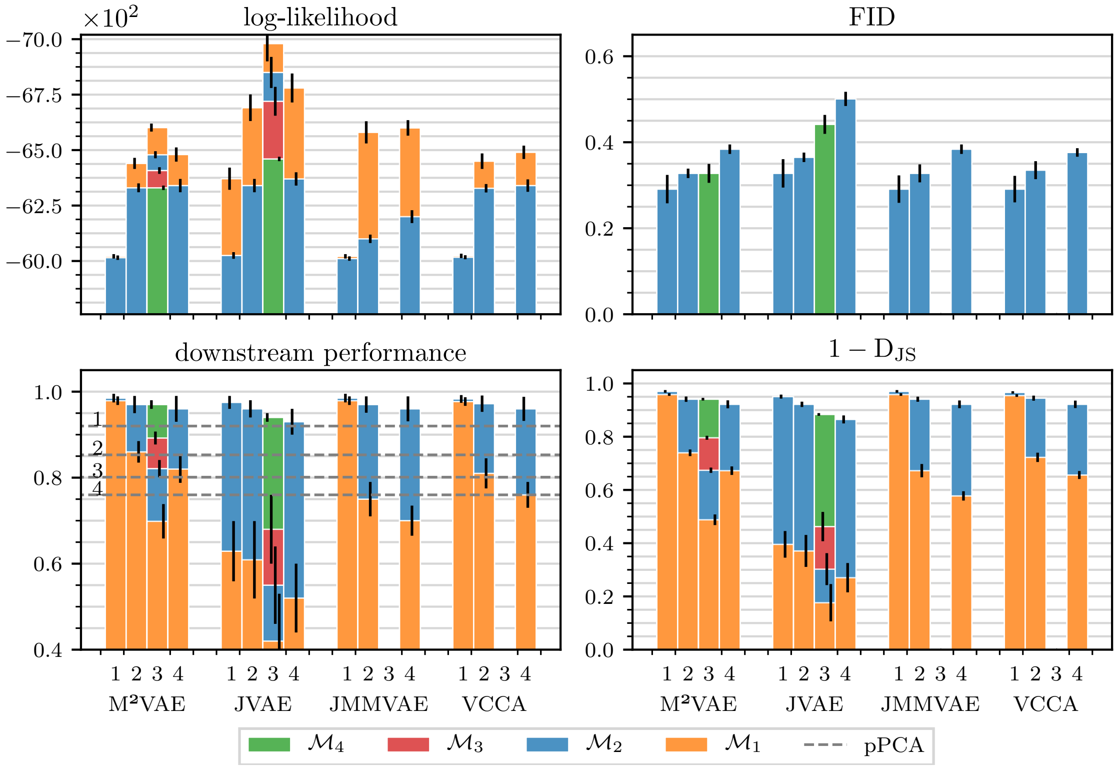


Figure 7.12: Competitive evaluation between five approaches on the data sets. The legend has to be considered as follows: For the bi-modal data sets (i.e., **1**, **2**, and **4**), $\mathcal{M}_2$ denotes the full bi-modal observation, while $\mathcal{M}_1$ denotes the uni-modal scenario during modality drop-out w/o the questioned modality. For the tetra-modal data sets (i.e., **3**), $\mathcal{M}_4$ denotes the full tetra-modal observation while $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ denote the corresponding lesser-modal scenarios during modality drop-out w/o the questioned modality. pPCA denotes the baseline downstream performance concerning the data sets **1** – **4** with full observation.

[25] Further tweaks to improve the visual appearance of the decoded samples, like Suzuki et al. [Suz+17] applied the VEAGAN by Larsen et al. [Lar+16], were not applied as these techniques aggravate the latent space (see “VAEs and GANs” talk by Rosca [Zhu+18]). Therefore, the different architectures can be compared “as is”.

Every evaluation was performed on five different seeds. The following data sets were used:

- **1**: eMNIST
- **2**: divided MNIST (2)
- **3**: divided MNIST (4)
- **4**: MNIST+SVHN

Only the M^{2}VAE and JVAE approaches enable the training of more than two modalities while also respecting all modality subsets during drop-out. However, their two different objectives result in slightly different architectures (see Fig. 5.3), which results in a necessary data set augmentation for the JVAE. Although M^{2}VAE naturally considers all permutations and drop-outs, the drop-outs for the JVAE need to be rendered by zeroing the corresponding modalities. This results in exponentially increasing data set size and complexity. Note that divided MNIST (2) for $\mathcal{M}_1$ competes with divided MNIST (4) for $\mathcal{M}_2$ (i.e., both are half of the MNIST image). Furthermore, divided MNIST (2) for $\mathcal{M}_2$ competes with divided MNIST (4) for $\mathcal{M}_4$ (i.e., both are the full MNIST image).

7.2.3.1 Log-Likelihood and FID

The two scores, log-likelihood and FID, are applied to the approaches in this section because both manly render the reconstruction of the DGMs.[26] The log-likelihood was calculated with $\tilde{\mathcal{M}}$ being one of $\{\mathcal{M}_1, \ldots, \mathcal{M}_4\}$ (see Section 7.1.1) while the dropped modalities were permuted and averaged. Note that the ordinate is inverted and that high values (i.e., smaller bars) denote good results.

The FID, on the contrary, can only be calculated on the full multi-modal reconstruction because of the subsequent feature calculation by the Inception Net. Note that small values (i.e., smaller bars) denote good results.

M^{2}VAE vs. JVAE vs. JMMVAE vs. VCCA on data set 1: Among all approaches, the eMNIST data set performed tremendously and equally well, at least for the bi-modal observation. This can be attributed to the data set creation approach which involves sampling from the latent spaces where the generative factors were aligned. Thus, the two modalities show almost perfect correlation (see Section 6.2.5.2), which is resembled by the almost similar results for the bi-modal and drop-out scenario. However, the JVAE's log-likelihood drops noticeably, which can be attributed to the lack of capacity of the DNN for the additional zeroed samples.

M^{2}VAE vs. JVAE vs. JMMVAE vs. VCCA on data sets 2 & 4: The JMMVAE performs best for bi-modal observations on the bi-modal data sets divided MNIST

[26] Although the log-likelihood value also renders the mutual and regularization effects, these do not have a great and only reciprocal influence on the result.

(2) and MNIST+SVHN while M^{2}VAE, JVAE, and VCCA perform equally well on the log-likelihood. This behavior was already explained by the no-free-lunch effect in the previous sections. Although the uni-modal encoder networks of JMMVAE do not have an additional decoder network, they also do not regularize the bi-modal encoder in the latent space. Therefore, the bi-modal encoder network has improved capabilities in embedding more information into the latent space than the uni-modal encoder networks. However, this initial advantage turns out to be a disadvantage in the case of drop-out (i.e., $\mathcal{M}_1$) because the JMMVAE has never explicitly learned the uni-modal decoding. The decrease of the log-likelihood in the case of drop-out for the M^{2}VAE, JVAE, and VCCA can be attributed to the remaining ambiguities in the data set.

The FID score, on the contrary, is not that susceptible to small errors in the reconstruction. It displays equally performing scores for all approaches, besides the JVAE, which leads to the conclusion that the reconstructions are basically equivalent.

M^{2}VAE vs. JVAE on data sets 2 & 3: In the case of the divided MNIST (4) data set, M^{2}VAE reaches an almost equal performance to the divided MNIST (2) data set case. Furthermore, the drop-out of an increasing number of modalities is monotonically rendered by the log-likelihood, which validates a reasonable training approach. Finally, the JVAE performs comparably poorly, which can be attributed to the exponentially growing data set complexity and size due to the zeroed samples.

7.2.3.2 Downstream Performance and JSD

The downstream performance was calculated based on the accuracy of a linear Gaussian naïve Bayesian classifier. The classifier was trained on the full multi-modal embedding and then evaluated on the lesser-modal test sets.

The dashed lines denote the baseline evaluations on the corresponding data sets while using the pPCA for embedding the data. It is worth mentioning that every non-linear architecture performs better than the linear pPCA, which legitimize the effort of using DGMs overall.

The JSD renders similarly to the downstream performance. This effect was already discussed in the previous sections and was validated in Fig. 7.12 on a large variety of data sets.

M^{2}VAE vs. JVAE vs. JMMVAE vs. VCCA on data set 1: Among all approaches, the eMNIST data set performed equally well, at least for bi-modal observation. As mentioned before, the data set creation enforces a strong correlation between the generated samples. This results in very coherent latent spaces when the samples are used for training. However, the JVAE's performance noticeably drops, which can be attributed to the tearing of the embeddings due to the zeroed data sets. This commonly results in different separated embedding regions that are rendered by the downstream performance and JSD.

M^{2}VAE vs. JVAE vs. JMMVAE vs. VCCA on data sets 2 & 4: Although all approaches perform equally well for the bi-modal embedding, the proposed M^{2}VAE performs significantly better during drop out. This behavior can be attributed to the holistic objective function of the M^{2}VAE, which considers all permutations of embeddings and reconstruction loss at once.

M^{2}VAE vs. JVAE on data sets 2 & 3: In the case of the data set divided MNIST (4), M^{2}VAE achieved almost equal performance as the divided MNIST (2) data set. Furthermore, the drop-out of an increasing number of modalities is monotonically rendered by the downstream performance as well as the JSD, which correlates with the results of the log-likelihood. Finally, the JVAE performs comparably poorly, and this performance can be attributed to the teared embedding due to the zeroed samples.

7.3 Discussion

In this chapter, the author demonstrated the benefits of the proposed M^{2}VAE over its preceding and competing architectures on various data sets. The most beneficial feature of the M^{2}VAE is that it considers more than two modalities. Although the JVAE also supports this feature by augmenting the data set, its performance drops with increasing data set complexity and size. The M^{2}VAE is not affected by this drawback because it inherently learns all data set permutations and mutual correlations together.

Compared to the JMMVAE and VCCA approaches, the M^{2}VAE performs equally well for full multi-modal observations. However, the M^{2}VAE truly shines during modality drop-out because by that stage, it has learned how to embed the data in the latent space while maintaining strong coherency to all other observations. This was proven not only mathematically in Section 5.3 but also practically in the ablation studies.

The correlation analysis between HPs, training objectives, coherence, and downstream performance revealed the satisfactory behaviors of the M^{2}VAE. The introduced mutual β can be facilitated to directly control latent space's coherency between the modality subsets' embeddings. Furthermore, the coherence, by means of the JSD, correlates with downstream performances that are affected by modality dropout.

In the following statements, the author concludes and summarizes this chapter: The true nature of the facilitated multi-modal data set needs to be considered at the beginning of the design of any multi-modal experiment. Natural phenomena that are observed by multiple modalities inevitably cause ambiguities in the samples. Observation ambiguities are not particular or rare corner cases but, rather, ordinary cases, and approaches that considering these effects, like the proposed M^{2}VAE, perform well in general. The M^{2}VAE also performs the most coherent, but still implicit, manifold alignment between all modality subsets, including drop-out and observation ambiguities. This coherency was shown in various manners through the ablation studies and JSD or downstream evaluations.

8 Applications

The investigations in the last chapters of this work concerning the proposed M^{2}VAE strove to resolve ambiguity, but how further applications can be derived from the M^{2}VAE's embeddings if ambiguous measurements were made is also of particular interest. A qualitative motivation can be seen in the XOR example from Section 7.2.2.1. Particularly, the results from the M^{2}VAE in Fig. 7.6 indicate that the variances of ambiguous embeddings are higher than these of unambiguous embeddings. This is due to the mean-seeking property that was investigated by the author in Section 5.3.2, which always causes the ambiguous embeddings to encapsulate all related embeddings.

This behavior will be investigated throughout this chapter in detail according to the following questions:

- How does the proposed concept of ambiguity in the M^{2}VAE relate to active sensing?
- How can one retrieve the M^{2}VAE latent space properties to derive an applicable behavior?
- How does the M^{2}VAE approach compare to the intrinsic curiosity module (ICM), and how does it perform in a real-world scenario?

Section 8.1 contains a discussion of active sensing (AS) and how the M^{2}VAE's objective could relate to this. A deep reinforcement learning (DRL) framework is proposed by the author in Section 8.2, which the M^{2}VAE enables to learn and perform active sensing tasks. Finally, Section 8.3 contains a comparison of the M^{2}VAE against the ICM and demonstrates the interplay by means of the distributed sensing setup.

8.1 Active Sensing through Ambiguity

Active sensing (AS) is one part of the most fundamental problems in navigation and exploration according to a survey conducted by The Robot Report in 2018 about the 10 biggest challenges in robotics that may have breakthroughs in 5 – 10 years.[1] Briefly AS is used to maximize the efficiency of an estimation task by

[1] https://www.therobotreport.com/10-biggest-challenges-in-robotics/

actively controlling the sensing parameters. It can roughly be divided into two subtask (see [Baj+18; Yu+09; Mih+02; Chu+04; Kre+05]): First, the identification of an objective to achieve (e.g., an object to sense, estimation of vantage points, or selection of sensing modality) and second, the navigation of a robot through an environment to reach a certain goal location without colliding with any obstacles. The autonomous identification of a point of interest (PoI) (i.e., object or place) for exploration or to answer the question "where to go next" is no less interesting than the identification and navigation tasks. Combining navigation and exploration by means of information retrieval remains a challenging field in research for tasks such as map building and surveillance operations. This field can be extended to cases in which multiple robots (i.e., "multi-agent" or "multi-robot") are used to speed up a task, or achieve reliability by distributing task specific equipment.

Heterogeneous robot teams that use various sensing modalities are of further interest in this thesis because particular sensors may only enable the perception of few object properties, which therefore, become ambiguous observations.[2] For instance if two agents, one with a shape detector and the other with a color detector, have to find all red boxes in an environment, then they must face the following issue: not all boxy shapes are red and not all red patches are boxes. Thus, the agents have to resolve the ambiguities of each other's observations.

In summary, the goal of AS can be formulated such that an agent would only take the effort to approach a particular sensor configuration if, and only if, it helps to resolve an observation ambiguity. Such a behavior can by facilitated through the correlation between the maximization of the trainable ELBO, by means of a M^{2}VAE, and the minimization of variational free energy, resulting in active-sensing with epistemic behaviors. This correlation exists between the actions and resulting observations and is exploited by embedding it in a RL framework in the upcoming section.

8.2 Embedding M^{2}VAE in a Learning Framework

The embedding of the M^{2}VAE into an DRL framework to enable epistemic (ambiguity resolving) goal-directed behavior in an AS application is proposed by the author in this section. AS can be formulated as a Markov decision process (MDP) to estimate the sensing actions of an agent with different sensor modalities. An MDP is a framework that can be seen as a stochastic extension of finite automata and as Markov processes augmented with actions and rewards.

[2] The term "partial observation" is not used in order to not confuse this expression with the POMDP framework. Partial observable MDP (POMDP): Partial observation concerning the agent's state. Mentioned case: partial observation (i.e., ambiguity) concerning environmental/object states.

The MDP consists of states $\mathcal{S}$, actions $\mathcal{A}$, transitions $\mathbf{T}$ between states, and a reward function definition r. Therefore, an MDP M can be seen as a tuple $M = (\mathcal{S}, \mathcal{A}, \mathbf{T}, \mathrm{r})$. If the model (r, T) of the MDP exist, then Bellman's equation can be applied to calculate an optimal policy π that maximizes the expected reward. The RL framework is a model-free solution that can be applied when $\mathbf{T}$ and r – and, therefore, the model of the problem – are not given. RL enables to learn the correlation between actions and observations by interacting with the environment while having the reward r as the only feedback signal. However, observations and actions can be complex, non-linear, and high-dimensional, which is why researches celebrated the advent of DNNs as powerful function approximators that determine actions directly from raw observation sequences, which was first published by Mnih et al. [Mni+15]. RL and DRL have been widely studied, and readers are encouraged to look at the comprehensive overviews by Sutton et al. [Sut+18] (RL), Tai et al. [Tai+16] (DRL for robotic control), or Bus et al. [Bus+10] (MARL), for example, to delve into this chapter. Therefore, an agent could be trained by applying a DRL approach utilizing deep Q-learning, for example, such that the agent learns the interplay between the current state of observation, the outcome of its action, and the resulting reward.

A detailed analysis of the observation–action process and a DRL framework, that unifies Q-learning and M^{2}VAE, is performed in the following sections. A formulation of the observation–action process as DGM is performed in Section 8.2.1. Finally, a unification architecture to embed the M^{2}VAE into a RL framework is proposed in Section 8.2.2.

8.2.1 Analysis of the Observation–Action Process

The observation–action process based on the intrinsic curiosity module (ICM) approach by Pathak et al. [Pat+17] is analyzed in this section, but in contrast to the process used by [Pat+17], the process here will be rendered as a probabilistic graphical model (PGM).

The Rubiks data set demonstrates that a static state of environment m exists and is intentionally learned by the ICM. Moreover, an object state s exists and remains constant during one epoch. That was exclusively learned by the M^{2}VAE in addition to the static environmental state m. Furthermore, an initial pose p with observation o exists, which is followed by a subsequent pose p' with observation o' after performing an action a.

Neglecting s, this setup looks very similar to one step of graph-based SLAM as described by Grisetti et al. [Gri+10]. Thus, the generative model (GM) without any object s can be directly adapted, as shown in Fig. 8.1 (left).

Next, the interplay between the environment and the RL framework is analyzed in this paragraph. Sun et al. [Sun+19] proposed a GM perspective of RL that can be adapted to the former graph-based SLAM GM. However, RL consists of two alternating phases, namely exploration and exploitation, which are crucial to learning (see Sutton et al. [Sut+18]). The learning phases are differentiated by the dashed arrow in Fig. 8.1 (mid.), where the action a is drawn randomly from a distribution of any choice during exploration and chosen by a DNN given the observation during exploitation. The process in the Rubiks data set can be formulated with just one action, while a bag of A actions is drawn.

Finally, the interplay between the object and the RL framework is analyzed in this paragraph. Grisetti et al. [Gri+10] and Sun et al. [Sun+19] assume a static map, which needs to be extended by a variable object. As shown in Fig. 8.1 (right), a bag of objects S is drawn, and it can be analyzed by A actions while the environment m remains constant. Because the object, environment, and the outcome of actions are represented by means of observations, a DNN can learn their correlated outcomes. To learn and differentiate good from bad actions, it is necessary to include an additional reward signal r that guides the DNN. However, it is worth mentioning that reward r is omitted because it is intrinsically calculated and deterministically depends on the observation, which will be further discussed in Section 8.2.2.

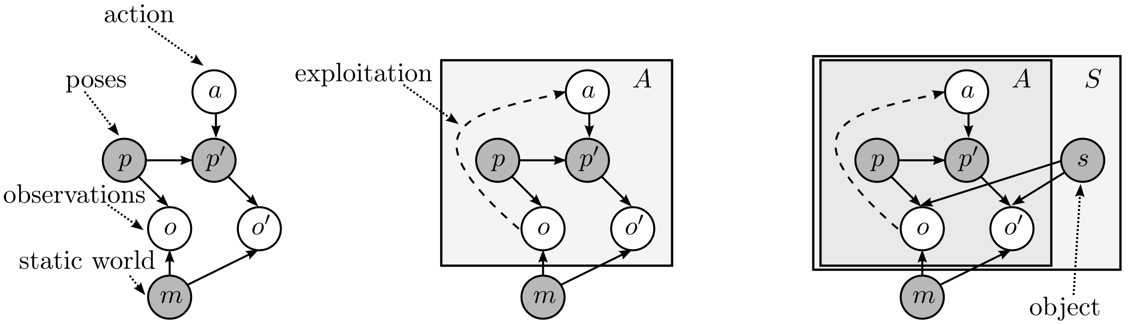


Figure 8.1: Plate models of the perception–action process. Left: Analog depiction of the first graph-based SLAM step by Grisetti et al. [Gri+10, Figure 4] plus the GM's nomenclature. Mid.: GM for exploration (w/o dashed arrow) and exploitation (w/ dashed arrow). Right: GM with additional objects with hidden states.

8.2.2 Perceived Environment

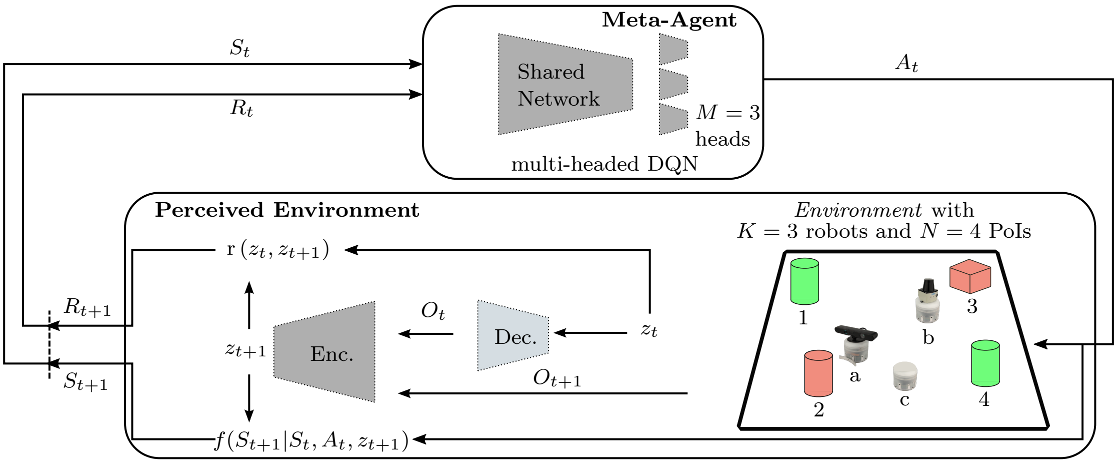


Figure 8.2: Overview of the meta-agent–environment interaction. The *perceived environment* consists of the simulator that executes actions and returns sensor readings and a M²VAE structure for calculating the reward and observation embedding. The meta-agent is requested K times for each robot, with the corresponding modality dependent head network to build the joint action $A \in \mathcal{A}^K$.

The perceived environment is introduced in a comprehensible fashion in this section. The perceived environment is an environment with observation post-processing that consists of four parts: the original environment (simulator or real-world), the state transition function f, the M²VAE, and the reward function r. The original environment manages K agents equipped with one of M sensing modalities. It executes control action $A_t \in \mathcal{A}$ for one agent k, and generates its sensor observations O_{t+1}. An action A for some chosen agent k enables it to drive and observe PoI n in the environment.

To facilitate the M²VAE for RL, it needs to be pre-trained on M modalities. The encoder networks of the M²VAE are used to encode a current observation O_{t+1}, while its decoder networks are used to decode a former embedding z_t to O_t that can be fused via re-encoding with the current observations. The state transition function $f(S_{t+1} \mid S_t, A_t, z_{t+1})$ produces the new state S_{t+1} based on the taken action A_t, the former state S_t, and the new embedding z_{t+1}. The reward function r is used to calculate the reward R_{t+1} based on the shift between the observations' embeddings, z_t and z_{t+1}.

The reward function can be based on the *epistemic value* defined by Friston et al. [Fri+17], which is the mutual information between hidden states: $r\,(z_t, z_{t+1}) \propto$

$D_{\mathrm{KL}}\left(z_t \| z_{t+1}\right)$. The *epistemic value* enables the calculation of the reduction in uncertainty about hidden states afforded by new observations by calculating the KLD. Because the KLD (i.e., the information gain) cannot be less than zero, it disappears when the former embeddings are not informed by new observations.

The meta-agent (see Fig. 8.2) comprises a policy network that maps a state to action. A single deep Q-network (DQN) can be applied as a meta-agent with M heads for each sensor modality.

Next, the internal state of the perceived environment can be defined as $S = (Z, V, T_1, \dots, T_K)$. The agent-independent environmental state of the world holds the M^{2}VAE's embeddings $Z = (\mu_1, \dots, \mu_N, \sigma_1, \dots, \sigma_N)$, and visits $V = (v_1, \dots, v_N)$ for every PoI n. A visit $v_n \in \{0, 1\}^M$ indicates which modalities have already observed this particular spot. PoIs can be stored by an environmental representation like a grid-cell or topological map where every cell or node n holds the information (μ_n, σ_n, v_n). An agent-dependent known pose T_k is given by the simulator or any localization system.

The state representation S_{t+1} passed to the policy network is constructed for every agent k by f as follows: The policy network had a fixed input size and, therefore, focused only on I PoIs in the vicinity of an agent. However, this approach is just quasi-myopic because only PoIs that have not been perceived by the same modality, indicated by V, were considered. Therefore, the states' respected surroundings can greedily grow to an arbitrary size. Thus, the agent specific state of the policy network comprises the world's state of I out of N PoI-embeddings and its path distances D for the requested agent k to every PoI: $\mathcal{S}_{t+1}|_k = (\mu_1, \dots, \mu_I, D_1, \dots, D_I)$.

For the agent's network output space $\mathcal{A}$, it can be assumed that each PoI can be observed by taking one action a_i, or the episode could be terminated before observing all PoIs through the selection of *no-operation* (NOP): $\mathcal{A} = (a_1, \dots, a_I, \mathrm{NOP})$. The agent's policy π_m was calculated based on the shared network and the modality dependent head, which was always chosen concerning the currently-controlled agent. PoIs were marked as visited if the policy samples NOP and the task was done if no more PoIs could be visited.

Finally, an average reward over all agents $\bar{r} = \sum_K r_k$ was calculated as the team reward for all agents to encourage cooperative behavior. Thus, every agent followed the policy, which maximized the future expected reward for the team.

8.3 Evaluation

The ICM vs. M^{2}VAE behavior on the Rubiks data set is analyzed in this section in addition to an AS task in a live scenario.

8.3.1 Rubiks

Despite the usual way of using complex data sets to show the capabilities of an approach, this evaluation is used to reduce the state space of observations to a comprehensible minimum while facilitating expressiveness for the interpretation of the results.

8.3.1.1 The Rubiks data set

The Rubiks data set is very comprehensible because it only consists of three poses p, six actions a, and 24 object states s (see Fig. 8.1). Therefore, 432 different observation-action-observation combinations are possible. Furthermore, the data set has some very unique properties and they are decisive for this analysis, as shown in Fig. 8.3:

- Only three poses and, therefore, three VP on the background m are possible $\Rightarrow$ the action deterministically determines some part of the observation o.
- The left and right VP are redundant for any object state s_i.
- Performing action left→right and vice versa is deterministic.
- Any other action determines the state of the object.

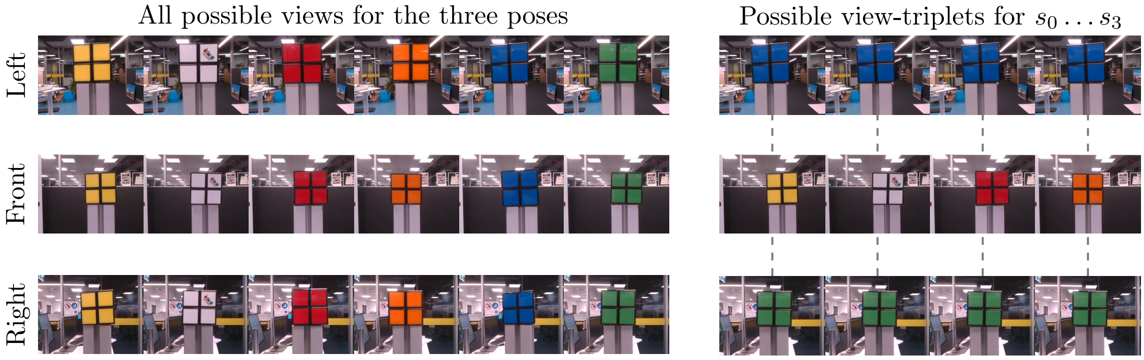

All possible views for the three poses
Possible view-triplets for $s_0 \dots s_3$
Left
Front
Right

Figure 8.3: Visualization of the Rubiks data set. Left: All possible background/object combinations demonstrating that the background is static concerning the viewpoint (VP) of the agent. Right: Four different states of the Rubiks cube and the corresponding observations, which demonstrate that left and right VP are redundant.

8.3.1.2 Intrinsic Curiosity Module (ICM) vs. Multi-Modal Variational Autoencoder (M²VAE)

The common approaches in RL demand a well-engineered extrinsic reward function r, which results in many pitfalls, such as incorrect specifications, reward sparsity, or huge parameter space.[3] In contrast to the extrinsic reward, one can facilitate curiosity as an intrinsic reward signal to enable an agent to learn how to explore its environment on its own. While not following any specific objective, the intrinsic reward is often derived from densely available proprioceptive and exteroceptive signals, which enable an agent to bootstrap its skills that might be useful in later tasks. In particular, the objective of the ICM by Pathak et al. [Pat+17] is to provide a reward signal just from the consecutive observations that are caused by its own action.

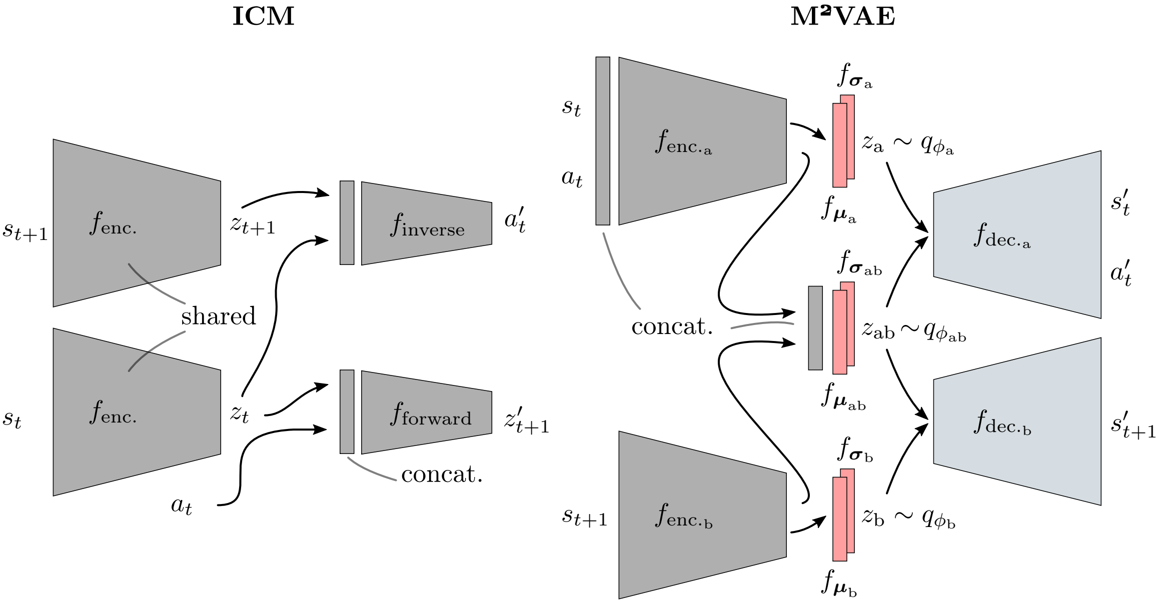

ICM
M²VAE
s_{t+1}
$f_{\text{enc.}}$
z_{t+1}
f_{inverse}
a'_t
shared
s_t
$f_{\text{enc.}}$
z_t
f_{forward}
z'_{t+1}
concat.
a_t
s_t
a_t
$f_{\text{enc.a}}$
$f_{\boldsymbol{\sigma}_{\text{a}}}$
$z_{\text{a}} \sim q_{\phi_{\text{a}}}$
$f_{\boldsymbol{\mu}_{\text{a}}}$
$f_{\text{dec.a}}$
s'_t
a'_t
concat.
$f_{\boldsymbol{\sigma}_{\text{ab}}}$
$z_{\text{ab}} \sim q_{\phi_{\text{ab}}}$
$f_{\boldsymbol{\mu}_{\text{ab}}}$
$f_{\text{dec.b}}$
s'_{t+1}
$f_{\boldsymbol{\sigma}_{\text{b}}}$
s_{t+1}
$f_{\text{enc.b}}$
$z_{\text{b}} \sim q_{\phi_{\text{b}}}$
$f_{\boldsymbol{\mu}_{\text{b}}}$

Figure 8.4: ICM based on Pathak et al. [Pat+17, Figure 2] vs. M²VAE architecture for the Rubiks data set.

The ICM is depicted in Fig. 8.4 (left) and consists of an encoder, forward model, and inverse model that are realized as DNNs. By comparing the actual next state z_{t+1} to the predicted next state z'_{t+1}, Pathak et al. [Pat+17] derived the reward signal as $\mathrm{r}_{\mathrm{ICM}} = \|z_{t+1} - z'_{t+1}\|_2^2$. The learning objective of the ICM itself is the prediction of the actions given the observations. Therefore, the ICM has no incentive to learn

[3] The blog post "Faulty Reward Functions in the Wild" by OpenAI gives an example of this issue: `https://openai.com/blog/faulty-reward-functions/`.

and represent factors of variation in the environment that do not affect the agent itself. Pathak et al. [Pat+17] mention a noticeably issue with the ICM that it can only be facilitated during bootstrapping because the reward signal degenerates as the DNN learns how to predict the environment. This bring the overall approach into question because the ICM loses its curiosity property over time. It is also worth mentioning that the reward signal itself can only be calculated a-posteriori, which means after the action was performed and the consecutive observation was obtained.

In contrast to the ICM, any VAE tries to learn and represent any information in general. Figure 8.4 (right) shows an analog M^{2}VAE architecture for providing an intrinsic reward signal from observations and actions. The incentive for the design is as follows: The encoder $f_{\mathrm{enc}\cdot\mathrm{a}}$ handles the joint information (s_t, a_t) and is analog to the forward model of the ICM. The encoder $f_{\mathrm{enc}\cdot\mathrm{ab}}$ embeds the consecutive observation s_t into the same latent space. By means of ambiguity, one only has to investigate whether the embeddings z_{a} and z_{ab} are the same (i.e., a non-informative and non-curious action was performed) or different (i.e., an informative and curious action was performed): $\mathrm{r}_{\mathrm{M^2VAE}} = \|f_{\boldsymbol{\mu}_a} - f_{\boldsymbol{\mu}_{ab}}\|_2^2$. Other than the ICM, this definition of the reward signal should not degenerate over time because it is a property of the latent space.

8.3.1.3 Results

The results of the ICM were confirmed concerning the predictions by Pathak et al. [Pat+17] because the architecture's objective is to predict the agent's movement using the embeddings z. As discussed before, every observation consists of the static background and a variable object. Pathak et al. [Pat+17] stated that the ICM has no incentive to represent factors of variation that are independent of the agent's movement (i.e., the Rubiks cube). However, the static background correlates very well with the agent's movement and is, therefore, represented by the ICM in its latent space. The three states are clearly separated in Fig. 8.5 and show no variation concerning the object. Furthermore, Fig. 8.6 shows that the intrinsic reward signal degenerates as the ICM learns the embedding of the background.

The M^{2}VAE, on the contrary, shows a very rich embedding in its latent space, representing all the different states of the Rubiks cube and the background (see Fig. 8.5 (right)). It is worth mentioning that the 2D embedding is not actually applicable and was just chosen for demonstration because the M^{2}VAE highly entangles the information in the latent space. However, choosing $D_{\mathrm{z}} = 64$ let the M^{2}VAE embed the observations such that the reward signal correlates with the observations of the object. Figure 8.6 (right) depicts two different reward signals that distinguish the outcome for deterministic and non-informative left to right movements (i.e., $\mathrm{r}_{\mathrm{M^2VAE}}\,|_{\mathrm{left/right}}$) and other informative movements (i.e., $\mathrm{r}_{\mathrm{M^2VAE}}\,|_{\mathrm{else}}$). Unexpectedly, do the left to right movements still cause a reward signal (i.e., a shift in the latent space) that can be attributed to the object's pose, which can slightly vary. However, the other informative movements always cause a high reward, which facilitates the M²VAE to build proper latent space statistics for AS because RL only pursues the maximization of future expected rewards.

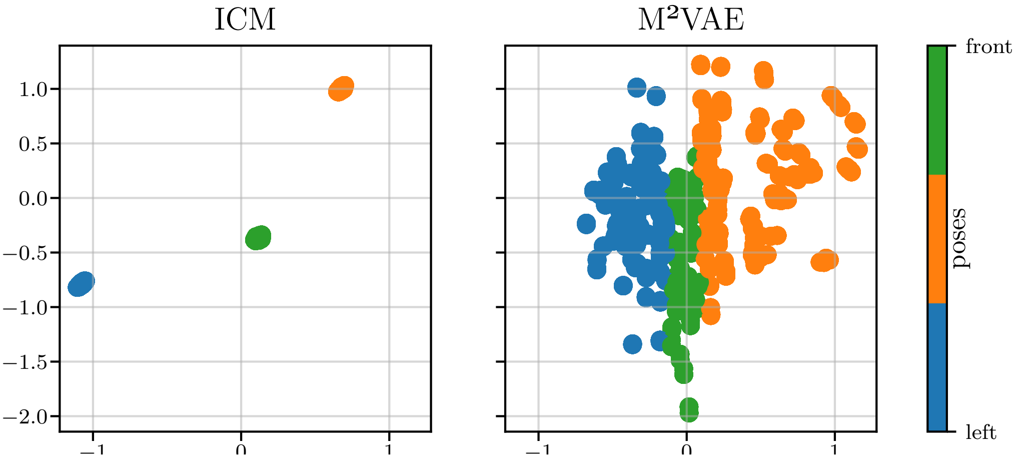


Figure 8.5: Latent space of ICM vs. M²VAE for $D_\mathrm{z} = 2$.

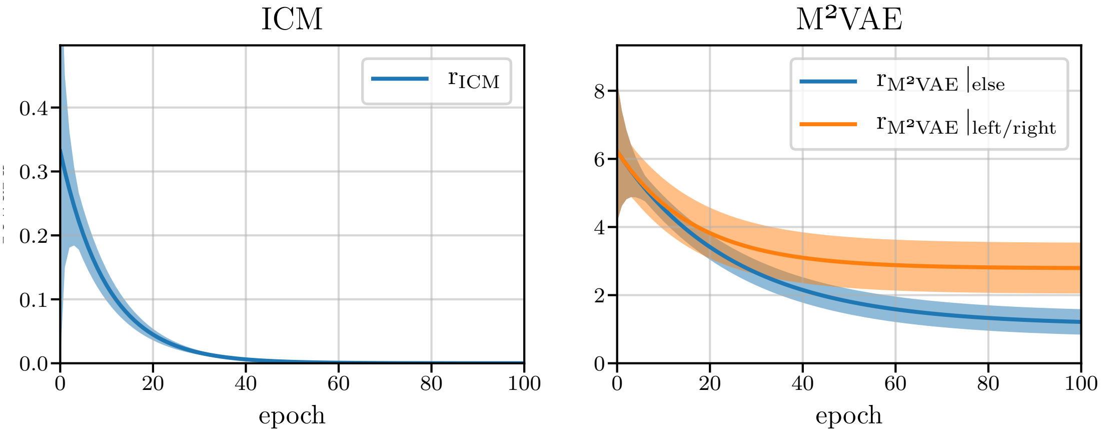


Figure 8.6: Evolution of the reward signal over the first 100 epochs ICM vs. M²VAE. In comparison to Fig. 8.5 is $D_\mathrm{z}|_\mathrm{ICM} = 2$ and $D_\mathrm{z}|_\mathrm{M^2VAE} = 64$, as described in Appendix B.2.10.

Both DNN architectures are summarized in Appendix B.2.10. The findings of the author about these architectures refute the statement by Pathak et al. [Pat+17] that VAEs are not suitable for deriving intrinsic motivation signals from observations.

Figures 8.5 and 8.6 clearly show the superiority of the M^{2}VAE as follows: First, the M^{2}VAE's reward signal does not degenerate during training and, therefore, any agent can learn a stable behavior based on this. Second, instead of static parts in the observations (i.e., the background), the M^{2}VAE learns the variational parts. This aligns with the idea of active sensing (AS) with the goal of learning behavior to resolve observation ambiguities.

8.3.2 Active Sensing with Distributed and Heterogeneous Robots

An evaluation based on the physically available CITrack and three AMiRos with different sensor configurations (see Fig. 8.7), RGB camera (a), LiDAR (b), proximity sensor (c) is performed in this section.[4] As previously motivated, AS reduces ambiguities of observations intrinsically through epistemic (ambiguity resolving) actions. Friston [Fri10] stated that actions enable the realization of preferred outcomes based on the assumption that both action and perception are used to maximize the evidence or marginal likelihood of a generative model, as scored by variational free energy. Following this principle, if one could directly obtain an estimation of free energy through the current observation, then this would enable the intrinsically motivated training of autonomous agents to gather information about their environment. Moreover, the agent would learn an epistemic goal-directed behavior because it would only take the effort of driving to a particular vantage point if and only if (iff) its sensor modality helps to resolve ambiguity.

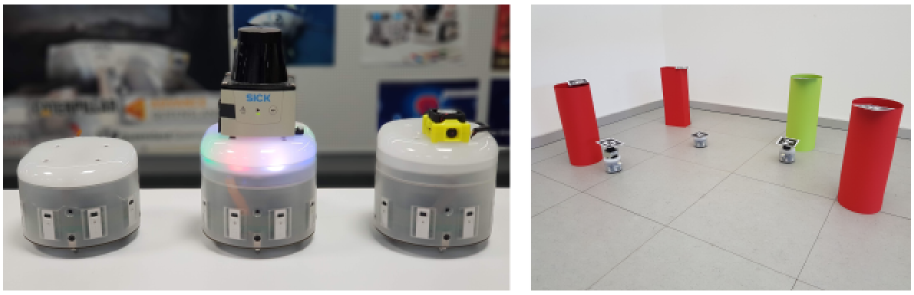


Figure 8.7: Three different AMiRo sensor setups (left) and exemplary CITrack setup (right).

Higgins et al. [Hig+17b] proposed a valid approach to train DRL approaches based on the representations of a VAE, first learn-how-to-see and then learn-how-to-act. Following this principle, uni-modal VAEs were bootstrapped on the Camera+LiDAR

[4] Both platforms are introduced in Appendix B.11.

data set from Section 6.2.3 on each modality. This stabilizes later multi-modal training by projecting the observations to a latent space. Second, a tri-modal M²VAE was applied to retrieve a common latent embedding of the observations and estimate the ELBO as a quantity of free energy, as shown in Fig. 8.8 and Fig. 8.9. Third, a multi-headed DQN was trained, as described in Section 8.2.2, on the latent embedding of the M²VAE with the M²VAE KLD estimations as reward signal to perform epistemic actions concerning its modality.

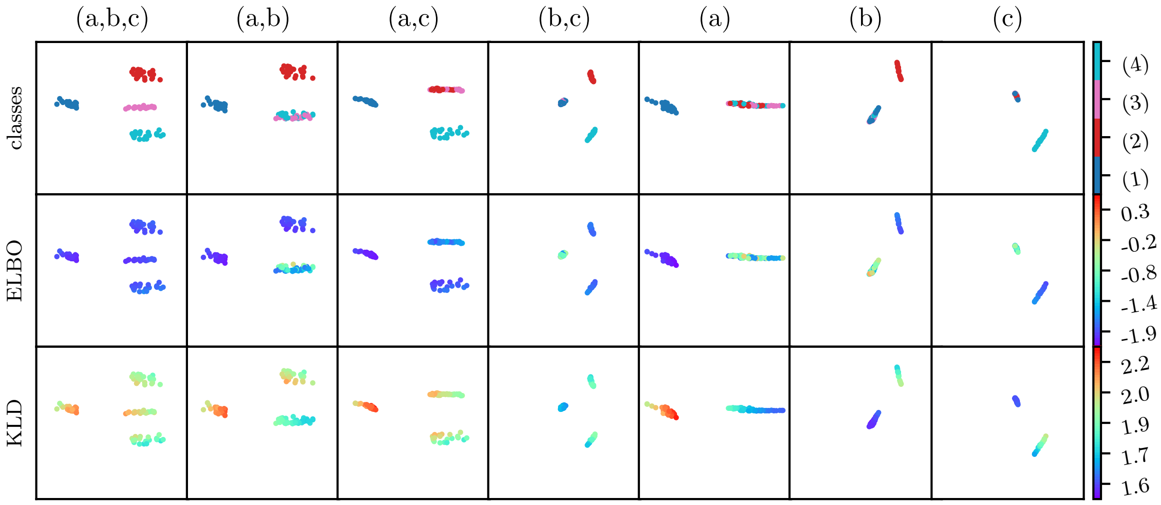


Figure 8.8: Visualization of jointly trained latent space embeddings z for all seven encoders q_{ϕ_*} of the subsets $\mathcal{P}(\mathcal{M})\backslash\emptyset$ with $\mathcal{M} = \{a, b, c\}$.

The modality combinations required to achieve the unambiguously classification of an object is shown in Table B.21. However, Fig. 8.8 shows that the M²VAE is able to detect ambiguous classifications and develop coherent relations by means of distribution and log-likelihood.

The upper row of Fig. 8.8 shows that the different embeddings for each modality subset clearly separate unambiguous observations from each other. Objects that are unambiguously detectable by a subset share similar distributions among all latent spaces, which demonstrates the coherence between all encoder networks. One example of this case is object (1), which shows a pure scatter for the modality-subsets (a), (a, c), (a, b), and (a, b, c). This is obviously caused by modality a, which determines the object's encoding on its own, but it is an important fact that other information sources do not corrupt or alter the embedding. Ambiguous detections from any modality subset collapse to the mean value of the separable classes, which was already observed in earlier experiments. Compared to the previous example, the collapse can be noticed in any subset concerning object (1) where a is missing.

The middle row of Fig. 8.8 shows the latent embeddings using the log-likelihood value for coloring. It can be observed that ambiguous embeddings show high log-likelihood, which is also rendered in Fig. 8.9. This shows a desirable reward signal, but it is worth mentioning that the log-likelihood, as calculated in Section 7.1, can only be retrieved if all modalities are available. Although each agent only carries one sensor, the log-likelihood cannot be calculated in the later RL scenario, which legitimates the loss-definition purely on the latent embeddings, as introduced in the former section. However, Fig. 8.9 shows the decreases with every modality that joins the subset (e.g., (c) vs. (3) $\xrightarrow{\text{add obs. of (a)}}$ (a,c) vs. (3) $\xrightarrow{\text{add obs. of (b)}}$ (a,b,c) vs. (3)), whereas a subset that unambiguously detects a class has already the lowest log-likelihood value (see (b) vs. (2) $\xrightarrow{\text{add obs. of (a)}}$ (a,b) vs. (2) $\xrightarrow{\text{add obs. of (c)}}$ (a,b,c) vs. (2)).

Finally, and for the sake of completeness, the bottom row of Fig. 8.8 shows the KLD concerning the regularizer. It behaves reciprocally to the log-likelihood because it has to tighten its shape against the prior distribution to find a specific encoding that enables good reconstruction. Although this KLD is always retrievable and shows correct behavior, its use is not recommended, because it highly depends on the VAE's choice of how it embedded the signal. It does not necessarily have to show any correlation with epistemic observations.

Figure 8.9 (mid.) shows the reward signals between epistemic observations that render a similar behavior to the log-likelihood. Ambiguity-resolving observations always cause a higher reward signal then the other cases, which leads to a sufficient reward definition for any RL task. The evolution of the collected reward by the agent at the end of every epoch is depicted in Fig. 8.9. This leads to the conclusion that the agent learns how to perform epistemic actions that lead to AS behavior just by performing observation facilitated through the proposed M^{2}VAE.

8.4 Discussion

This chapter was used to demonstrate the applicability of the M^{2}VAE to active sensing (AS) tasks via its epistemic, ambiguity resolving capabilities. First, AS was introduced and the analogy to epistemic sensing was discussed. Following the statements by Friston [Fri10] and his introduction to epistemic sensing, a one-to-one analogy to the introduced properties of ambiguity representation by the M^{2}VAE in the latent space, as introduced in Section 5.3, and epistemic sensing was argued by the author.

In the first experiment, the AS property was applied by the author in a curiosity learning task comparing the results to the SOTA ICM approach by Pathak et al. [Pat+17] on the Rubiks data set. All features and drawbacks of the ICM were

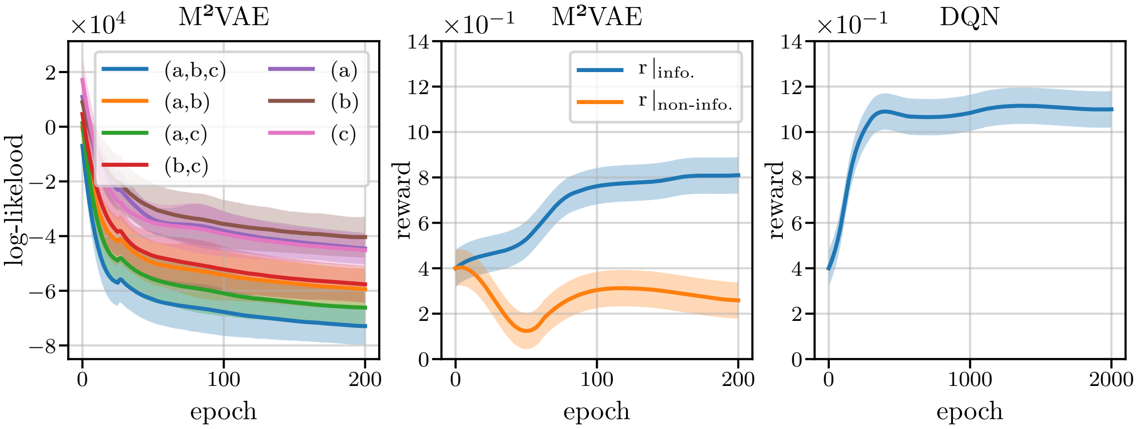


Figure 8.9: Evolution of the log-likelihood (left) and reward (mid.) over the training epochs of the M²VAE. The right figure shows the reward of the DQN training using the intrinsic reward signal of the M²VAE.

confirmed, and the M²VAE showed major superiority in the curiosity task by first maintaining a reward signal after convergence and distinguishing informative versus non-informative observation–action sequences. Furthermore, the inspection of the actual architectures and hyperparameters (HPs) in Appendix B.2.10 concerning the ICM reveals the demand of plentiful regularization techniques to facilitate stable learning results. This was not the case with the M²VAE, demonstrating, again, its superior applicability to such tasks. However, the Rubiks data set was chosen because of its comprehensibility and high data dimensionality but limited state space complexity. Highly complex data sets could also demand great efforts in finding suitable M²VAE architectures for AS, but the experiment already demonstrated a promising and plausible results.

The second and final experiment extended the formerly revealed features to an actual use-case of distributed sensing via three heterogeneous robots. First, the perceived environment was introduced. It combined a multi-headed meta-agent DQN with a tri-modal M²VAE for camera, LiDAR, and proximity sensors. Second, the statistics about the learned sensor observations in the latent space were demonstrated and revealed promising results based on the ELBO for each modality sub-set by following the paradigm of epistemic sensing. Finally, the coupling of all components led to the coordinated and optimized sensing behavior of the whole fleet that was enabled by first learning-how-to-fuse and then learning-how-to-act.

9 Conclusion and Outlook

In this thesis, a new approach to derive a deep generative model for multi-modal machine learning was introduced. Several topics related to multi-modal problem settings were addressed and discussed regarding the formal definition of multi-modality, ambiguities, modality drop-out, fusion, and the nature of multi-modal observations. The formal derivations and conclusions about the models' behaviors, especially those related to observation ambiguities and drop-out, demonstrated that the models exhibit superior performance compared to earlier approaches.

Chapter 1 contains a description of the background and objectives of this research. The evaluation of the multi-modal and deep generative topics from the most influential conferences revealed that there is a young but prospering community surrounding multi-modal deep generative models, which also emphasizes the importance of this thesis. In Chapter 2, the author delved into the addressed topics and points out the related contributions of the author that led to this thesis.

Chapter 3 contains an introduction to the mathematical foundations and frameworks around deep learning and generative models. Special attention was drawn to the unification of both techniques that ultimately became the unsupervised trainable variational auto encoder (VAE) by Kingma et al. [Kin+13] and Rezende et al. [Rez+14] to give the reader a deep and concise background for the later derivations and conclusions.

In Chapter 4, the author demonstrated the necessity of multi-modality and the challenges that had to be addressed in this thesis. First, a formal definition of multi-modal perception was derived from Weiss et al. [Wei+16], distinguishing domains, tasks, correlation, ambiguity, and heterogeneity in multi-modal setups. Second, the multi-modal deep generative model approach was drawn into the extended taxonomy canvas by Baltrušaitis et al. [Bal+19]. A discussion of ambiguous observations particularly during modality dropout revealed that ambiguity has not been studied in deep generative models yet. However, this may become a crucial aspect in future resilient and trustworthy machine learning and artificial intelligence applications, which demonstrates the necessity of the conducted research.

Chapter 5 contains the proposal of the multi-modal variational autoencoder (M^{2}VAE) as a multi-modal deep generation model that was derived from the full marginal joint log-likelihood. The author showed that the multi-modal variational autoencoder (M^{2}VAE) is a general evolution from the joint multi-modal variational autoencoder (JMMVAE) by Suzuki et al. [Suz+17], who derived their approach from the variation of information to address bi-modal exchange. Compared to the joint multi-modal variational autoencoder (JMMVAE), the multi-modal variational autoencoder (M^{2}VAE) not only introduces additional reconstruction paths but also enables generalization to arbitrary modality subsets that handle modality-drop-out. Special attention was placed on the mathematical derivations for observation ambiguities and modality drop-out, which were introduced by the conscious and unconscious objectives of the VAE. The approach revealed two tremendous results for the applicability and behavior of the latent space during training using unsupervised multi-modal deep generative models. First, ambiguous observations are represented by mean-seeking behavior in the latent space, such that all related observations are consolidated but still distinguishable through their latent statistics. Second, knowing the ground truth labels for ambiguous observations would not improve the results because the unsupervised and hypothetically supervised objectives are both concave and share the same optima. Furthermore, an architectural choice for training the M^{2}VAE based on weight-sharing was proposed to handle the exponentially-increasing number of network weights, depending on the utilized number of modalities. Finally, a new training approach based on re-encoding was introduced. It was not found to improve the VAE in general, but enabled the visualization of the latent statistics and consecutive fusion of observations in the latent space.

In Chapter 6, the author bootstrapped the evaluation by investigating available multi-modal data sets and proposing new ones that comprehensibly demonstrate the results of the M^{2}VAE. A proper categorization of multi-modal data sets was developed with sufficient definitions of properties, which can be used to match available data sets to any experiment. The author provided an exhaustive insight into a multitude of data sets and pointed out that state-of-the-art multi-modal data sets are by no means comprehensible because of their high complexity or lack of true multi-modal nature. This ultimately led to the introduction of the proposed multi-modal data sets XOR (which should be the gold-standard baseline anyway), Mixture of Gaussians (MoG), Camera+LiDAR, Rubiks, and eMNIST. The last data set also comes with a multi-modal data set generating technique based on the latent space consolidation of uni-modal CVAEs.

In Chapter 7, the author demonstrated the benefits of the proposed M^{2}VAE over its preceding and competing architectures on various data sets. First, the behavior based on the M^{2}VAE's hyperparameter was analyzed, revealing that the coherence of the latent space embeddings between modality subsets can be well controlled by introducing a single scalar as mutual beta. The second experiment was an exhaustive ablation study, which revealed that the M^{2}VAE truly shines during modality drop-out because it learns how to embed the data in the latent space while maintaining strong coherency to all other observations. This was not only proven mathematically in earlier chapters but also practically demonstrated within this chapter. Finally, the experiments on the complex visual data showed that the proposed model is more effective in the case of latent space coherence than competitive models. All experiments were conducted using the vae_tools library, which was developed by the author to facilitate the M^{2}VAE. However, it became a powerful and convenient tool that is easily adaptable to new VAE architectures and enables the conduction of experiments with deep generative models in just a few lines of code.

In Chapter 8, the author looked beyond the horizon of the common deep generative evaluations and applied the M^{2}VAE in two curiosity driven active sensing tasks. Curiosity, in this case, is the derivation of an intrinsic reward signal, which was achieved by analyzing the alternations of the latent space statistics in the case of ambiguity-resolving observations. This enabled the researcher to draw analogies to epistemic sensing and, therefore, extended this thesis by relating the M^{2}VAE to the work by Pathak et al. [Pat+17] on intrinsic curiosity and the free energy principle, including epistemic sensing by Friston et al. [Fri+16; Fri+17].

VAE approaches already play a fundamental role in anomaly detection, feature extraction, modality-exchange, and much more because of their robust, unsupervised, and data-driven architectures. In the field of deep generative models, the author of this thesis fundamentally extended state-of-the-art multi-modal deep generative models by analytically and practically investigating observation ambiguities during during training. Furthermore, the author proposed the M^{2}VAE that handles dropout and ambiguities better than any comparable approach. These findings may play a significant role in upcoming multi-sensory architectures that pursue resilient artificial intelligence by facilitating the informative fusion outcomes of observations, even when sensors break.

Handling very complex or high dimensional data is still a big challenge and often results in mode collapses. However, research is already underway and actively drives related conferences and news around the world. Considering the various fields of deep generative models, this work contributes new mathematical frameworks, fundamental findings, and promising results that will actively shape the current and future discussions on multi-modal machine learning.

Acronyms and Abbreviations

acc.	accuracy
approx.	approximately
attr.	attribute
art.	artificial
ADGM	auxiliary deep generative models
AE	auto encoder
AGI	artificial general intelligence
AGV	autonomous ground vehicle
AI	artificial intelligence
aka	also known as
ANN	artificial neural network
AMiRo	Autonomous Mini-Robot
API	application programming interface
AR	augmented reality
AS	active sensing
avg.	average
AVSR	audio-visual speech recognition
β**VAE**	beta variational autoencoder
BA	bundle adjustment
BCE	binary cross-entropy
beh.	behavior
bMNIST	binarized MNIST
bot.	bottom
BP	backpropagation
brv.	brevity

ca.	circa
CAD	computer aided design
CCA	canonical correlation analysis
ccw	counter-clockwise
CE	conditional entropy
c.f.	*confer*, compare
CITrack	cognitive interaction tracking
class.	classification
CMMA	conditional multi-modal autoencoder
CNN	convolutional neural network
col.	column
concat.	concatenation
cond.	condition/conditional
cont.	continues
conv	convolutional
corr.	correlation
CorrNet	Correlational Neural Network
CT	coordinate transformation
CUAVE	Clemson University Audio Visual Experiments
CV	computer vision
CVAE	conditional variational autoencoder
cw	clockwise
DAE	denoising autoencoder
DBM	deep Boltzmann machine
DC-IGN	deep convolution inverse graphics network
DCCAE	deep canonically correlated autoencoders
dec.	decoder/decoding
deconv	deconvolutional
DGM	deep generative model
disc.	discrete

DL	deep learning
DNN	deep neural network
DoF	degree of freedom
DQN	deep Q-network
DRL	deep reinforcement learning
DTW	dynamic time warping
DVAE	denoising variational autoencoder
e.g.	*exempli gratia*, for example
ELBO	evidence lower bound
ELU	exponential linear unit
EM	expectation–maximization
eMNIST	entangled-MNIST
enc.	encoder/encoding
env.	environment
et al.	*et alii*, and collegues
etc.	*et cetera*, and so on
FAE	facial attribute estimation
FAM	facial attribute manipulation
FFNN	feedforward neural network
FID	Fréchet inception distance
FM	fiducial marker
FMNIST	Fashion-MNIST
FOV	field of view
GAN	generative adversarial network
GM	generative model
GN	Gauss–Newton
GNB	Gaussian naïve Bayes
GNSS	global navigation satellite system
GQN	generative query network
GT	ground truth

HAR	human activity recognition
HMI	human machine interaction
HMM	hidden Markov models
HP	hyperparameter
ICM	intrinsic curiosity module
i.e.	*id est*, that is
iff	if and only if
IMU	inertial measurement unit
IoU	intersection over union
IS	inception score
ISM	inverse sensor model
JE	joint entropy
JVAE	joint variational autoencoder
JMMVAE	joint multi-modal variational autoencoder
JSD	Jensen–Shannon divergence
KDE	kernel density estimation
KLD	Kullback–Leibler divergence
lhs.	left hand side
lin	linear
LiDAR	light detection and ranging
LM	Levenberg–Marquardt
LR	learning rate
M3L	multi-modal machine learning
MAE	mean absolute error
MAP	maximum a posteriori probability
MARL	multi-agent RL
max.	maximum/maximization/maximal
MC	Monte Carlo
MCI	mutual conditional information
MCMC	Markov chain Monte Carlo

MDP	Markov decision process
mid.	middle
min.	minimum/minimization/minimal
M^{2}VAE	multi-modal variational autoencoder
MNIST	Modified National Institute of Standards and Technology
MNIST-A	MNIST-with-attributes
ML	machine learning
MLP	multi-layer perceptron
MoG	mixture of Gaussians
MoCap	motion capture
MSE	mean squared error
mul.	multiply/multiplied
MI	mutual information
MIC	maximal information coefficient
MV-AE	multi-view autoencoder
nat.	nature
NN	nearest-neighbor
NIR	near infrared
NN	neural network
norm.	normalization
NTP	Network Time Protocol
O	orthogonal group
obj.	object
p.	page
pc	parallel coordinates
PC	point cloud
PCA	principal component analyses
PDF	probability density function
PGA	perceptual generative autoencoder
PGM	probabilistic graphical model

proxy	approximator
PoI	point of interest
POMDP	partial observable MDP
pp.	pages
pPCA	probabilistic PCA
PTP	Precision Time Protocol
R	translatory group
RADAR	radio detection and ranging
rec.	reconstruction
reg.	regularization
regr.	regression
ReLU	rectified linear unit
reo.	reorder/reordering/reordered
RGB	red/green/blue
RGBD	RGB/depth
rhs.	right hand side
RL	reinforcement learning
ROS	Robot Operating System
RMSE	root mean squared error
RNN	recurrent neural network
RSB	Robotics Service Bus
SBA	sparse bundle adjustment
SDGM	skip deep generative model
SE	special Euclidean group
sec.	section
SGD	stochastic gradient decent
sig	sigmoid
SO	special orthogonal group
SOTA	state-of-the-art
src.	source

s.t.	such that
sub.	substitute
SLAM	simultaneous localization and mapping
SVM	support vector machine
tanh	hyperbolic tangent
TCVAE	total correlation variational autoencoder
TELBO	triple ELBO
TF	TensorFlow
triu.	upper triangle
tril.	lower triangle
t-SNE	t-distributed stochastic neighbor embedding
UNIT	unsupervised image-to-image translation
VAE	variational auto encoder
VCCA	variational canonical correlation analysis
VI	variational inference
VP	viewpoint
VoI	variation of information
vs.	*versus*, against
w/	with
w/o	without
wrt.	with respect to
WWW	world wide web
XOR	exclusive or
1D	one dimensional
2D	two dimensional
3D	three dimensional

List of Figures

List of Tables

Bibliography

[Aba+15] Martín Abadi, Ashish Agarwal, Paul Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* 2015. URL: `https://www.tensorflow.org/`.

[Ami+18] Alexander Amini, Wilko Schwarting, Guy Rosman, et al. "Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing". In: (2018).

[And+13] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. "Deep Canonical Correlation Analysis". In: ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1247–1255. URL: `http://proceedings.mlr.press/v28/andrew13.html`.

[And13] Johannes Andres. *Multivariate Statistik.* 2013. URL: `http://www.uni-kiel.de/psychologie/andres/multi13/M_S_13.pdf`.

[Ang+13] Davide Anguita, Alessandro Ghio, Luca Oneto Xavier Parra, and Jorge L. Reyes-Ortiz. "Human Activity Recognition Using Smartphones Data Set". In: *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN).* 2013. URL: `https://www.driveandact.com/`.

[Are+17] John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. *Gated Multimodal Units for Information Fusion.* 2017. arXiv: `1702.01992 [stat.ML]`.

[Aub+14] Mathieu Aubry, Daniel Maturana, Alexei Efros, et al. "Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models". In: *CVPR.* 2014. URL: `https://www.di.ens.fr/willow/research/seeing3Dchairs/`.

[Baj+18] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. "Revisiting active perception". In: *Autonomous Robots* 42.2 (2018), pp. 177–196. ISSN: 15737527. DOI: `10.1007/s10514-017-9615-3`. arXiv: `1603.02729`.

[Bau+01] Heinz H. Bauschke and Jonathan M. Borwein. "Joint and separate convexity of the bregman distance". In: *Studies in Computational Mathematics.* 2001. DOI: `10.1016/S1570-579X(01)80004-5`.

[Bea+16] Charles Beattie, Joel Z. Leibo, Denis Teplyashin, et al. *DeepMind Lab*. 2016. arXiv: `1612.03801 [cs.AI]`.

[Bel+13] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. "The Arcade Learning Environment: An Evaluation Platform for General Agents". In: *Journal of Artificial Intelligence Research* 47 (June 2013), 253–279. ISSN: 1076-9757. DOI: `10.1613/jair.3912`. URL: `http://dx.doi.org/10.1613/jair.3912`.

[Ben+12] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: 1993 (2012), pp. 1–30. ISSN: 15324435. DOI: `10.1145/1756006.1756025`. arXiv: `1206.5538`. URL: `http://arxiv.org/abs/1206.5538`.

[Bis07a] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Vol. 16. 4. 2007, p. 049901. ISBN: 978-0387310732. DOI: `10.1117/1.2819119`. arXiv: `arXiv:1011.1669v3`. URL: `http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.2819119`.

[Bis07b] Christopher M. Bishop. *Pattern Recognition and Machine Learning - Solutions to Exercises*. 1. 2007, p. 101. ISBN: 9780874216561. DOI: `10.1007/s13398-014-0173-7.2`. arXiv: `arXiv:1011.1669v3`.

[Bor+17] Sebastian Meyer zu Borgsen, Timo Korthals, Florian Lier, and Sven Wachsmuth. "ToBI – Team of Bielefeld: Enhancing Robot Behaviors and the Role of Multi-robotics in RoboCup@Home". In: *RoboCup 2016: Robot World Cup XX*. Ed. by Sven Behnke, Raymond Sheh, Sanem Sarıel, and Daniel D. Lee. Cham: Springer International Publishing, 2017, pp. 577–588. ISBN: 978-3-319-68792-6.

[Bor+95] J. Borenstein and Liqiang Feng Liqiang Feng. "Correction of systematic odometry errors in mobile robots". In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* 3 (1995), pp. 569–574. ISSN: 0277786X. DOI: `10.1109/IROS.1995.525942`.

[Bor01] Magnus Borga. *Canonical Correlation a Tutorial*. 2001. URL: `https://www.cs.cmu.edu/$\sim$tom/10701_sp11/slides/CCA_tutorial.pdf`.

[Bro+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al. *OpenAI Gym*. 2016. URL: `http://arxiv.org/abs/1606.01540`.

[Bur+15] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. 2015. arXiv: `1509.00519 [cs.LG]`.

[Bur+18a] Chris Burgess and Hyunjik Kim. *3D Shapes Dataset*. 2018. (Visited on Jan. 30, 2020).

[Bur+18b] Christopher P. Burgess, Irina Higgins, Arka Pal, et al. "Understanding disentangling in β-VAE". In: Nips (2018). arXiv: 1804.03599. URL: http://arxiv.org/abs/1804.03599.

[Bus+10] Lucian Bus and Bart De Schutter. "Multi-Agent Reinforcement Learning: An Overview". In: 38.2 (2010), pp. 156–172.

[Cad+16] Cesar Cadena, Anthony Dick, and Ian D Reid. "Multi-modal Auto-Encoders as Joint Estimators for Robotics Scene Understanding". In: *Robotics: Science and System XIII*. Ed. by Nancy Amato, Siddhartha Srinivasa, Nora Ayanian, and Scott Kiundersma. Cambridge: MIT Press, 2016. DOI: 10.15607/RSS.2016.XII.041.

[Cae+19] Holger Caesar, Varun Bankiti, Alex H. Lang, et al. *nuScenes: A multimodal dataset for autonomous driving*. 2019. arXiv: 1903.11027 [cs.LG]. URL: https://www.nuscenes.org/.

[Cha+15] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, et al. *ShapeNet: An Information-Rich 3D Model Repository*. 2015. arXiv: 1512.03012 [cs.GR]. URL: https://www.shapenet.org/.

[Cha+16] Sarath Chandar, Mitesh M. Khapra, Hugo Larochelle, and Balaraman Ravindran. "Correlational Neural Networks". In: *Neural Computation* 28.2 (Feb. 2016), 257–285. ISSN: 1530-888X. DOI: 10.1162/neco_a_00801. URL: http://dx.doi.org/10.1162/NECO_a_00801.

[Che+02] P.Y. Chen and P.M. Popovich. *Correlation: parametric and nonparametric measures*. Sage university papers series. no. 07-139 Nr. 137-139. Sage Publications, 2002. ISBN: 9780761922285. URL: https://books.google.de/books?id=UN4nAQAAIAAJ.

[Che+16] Xi Chen, Yan Duan, Rein Houthooft, et al. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: (2016). ISSN: 978-3-319-16807-4. DOI: 10.1007/978-3-319-16817-3. arXiv: 1606.03657. URL: http://arxiv.org/abs/1606.03657.

[Che+18] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. "Isolating sources of disentanglement in variational autoencoders". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2610–2620.

[Cho+16] Christopher B. Choy, Danfei Xu, JunYoung Gwak, et al. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". In: *Lecture Notes in Computer Science* (2016), 628–644. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46484-8_38. URL: http://dx.doi.org/10.1007/978-3-319-46484-8_38.

[Chu+04] T.H. Chung, V. Gupta, J.W. Burdick, and R.M. Murray. "On a decentralized active sensing strategy using mobile sensor platforms in a network". In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)* (2004), 1914–1919 Vol.2. ISSN: 0191-2216. DOI: `10.1109/CDC.2004.1430327`. URL: `http://ieeexplore.ieee.org/document/1430327/`.

[Coh+17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. *EMNIST: an extension of MNIST to handwritten letters.* 2017. arXiv: `1702.05373 [cs.CV]`.

[Cox+08] S. Cox, R. Harvey, Yuxuan Lan, et al. "The challenge of multispeaker lip-reading". In: *AVSP*. 2008.

[Csi67] Imre Csiszár. "Information-type measures of difference of probability distributions and indirect observations". In: 1967.

[Dos+16] Alexey Dosovitskiy and Thomas Brox. "Generating Images with Perceptual Similarity Metrics based on Deep Networks". In: (2016). arXiv: `arXiv:1602.02644v2`.

[Elf92] Alberto Elfes. "Dynamic control of robot perception using multi-property inference grids". In: 1992. DOI: `10.1109/ROBOT.1992.220056`.

[Elm02] Wilfried Elmenreich. "Sensor Fusion in Time-Triggered Systems". PhD thesis. Treitlstr. 3/3/182-1, 1040 Vienna, Austria: Technische Universität Wien, Institut für Technische Informatik, 2002.

[Esl+16] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, et al. *Attend, Infer, Repeat: Fast Scene Understanding with Generative Models.* 2016. arXiv: `1603.08575 [cs.CV]`.

[Esl+18] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, et al. "Neural scene representation and rendering". In: *Science* 360.6394 (2018), pp. 1204–1210. ISSN: 0036-8075. DOI: `10.1126/science.aar6170`. eprint: `https://science.sciencemag.org/content/360/6394/1204.full.pdf`. URL: `https://science.sciencemag.org/content/360/6394/1204`.

[Eve+] M. Everingham, L. Van Gool, C. K. I. Williams, et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.* (Visited on Jan. 30, 2020).

[Ezq+] Alberto Ezquerro, Miguel Angel Rodriguez, and Ricardo Tellez. *openai_ros*. URL: `http://wiki.ros.org/openai_ros` (visited on Jan. 30, 2020).

[Fos19] D. Foster. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*. O'Reilly Media, 2019. ISBN: 9781492041917. URL: https://github.com/tik0/GenerativeDeepLearningCode.

[Fri+16] Karl Friston, Thomas FitzGerald, Francesco Rigoli, et al. "Active inference and learning". In: *Neuroscience and Biobehavioral Reviews* 68 (2016), pp. 862–879. ISSN: 18737528. DOI: 10.1016/j.neubiorev.2016.06.022. URL: http://dx.doi.org/10.1016/j.neubiorev.2016.06.022.

[Fri+17] Karl Friston, Thomas FitzGerald, Francesco Rigoli, et al. "Active inference: A process theory". In: *Neural Computation* (2017). ISSN: 1530888X. DOI: 10.1162/NECO_a_00912. arXiv: 1309.2848v1.

[Fri10] Karl Friston. "The free-energy principle: A unified brain theory?" In: *Nature Reviews Neuroscience* 11.2 (2010), pp. 127–138. ISSN: 1471003X. DOI: 10.1038/nrn2787. arXiv: arXiv:1507.02142v2. URL: http://dx.doi.org/10.1038/nrn2787.

[Gei+13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[Glo+10] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: JMLR Workshop and Conference Proceedings, 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[Goo+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[Goo+16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016. URL: http://www.deeplearningbook.org/.

[Gri+10] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. "A Tutorial on Graph-Based SLAM". In: (2010), pp. 1–11. ISSN: 1939-1390. DOI: 10.1109/MITS.2010.939925. URL: http://www2.informatik.uni-freiburg.de/$\sim$stachnis/pdf/grisetti10titsmag.pdf.

[Gro95] Patrick J Grother. *NIST Special Database 19 - Handprinted Forms and Characters Database*. Tech. rep. 1995.

[Ha+18] David Ha and Jürgen Schmidhuber. “Recurrent World Models Facilitate Policy Evolution”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, et al. Curran Associates, Inc., 2018, pp. 2450–2462. URL: https://worldmodels.github.io/.

[Har+20] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, 2015, 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123. arXiv: 1502.01852. URL: https://doi.org/10.1109/ICCV.2015.123.

[Her+16] Stefan Herbrechtsmeier, Timo Korthals, Thomas Schöpping, and Ulrich Rückert. “AMiRo: A Modular & Customizable Open-Source Mini Robot Platform”. In: *ICSTCC*. 2016. ISBN: 9781509027200. DOI: 10.1109/ICSTCC.2016.7790746.

[Heu+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2017. arXiv: 1706.08500 [cs.LG].

[Hig+16] Irina Higgins, Loic Matthey, Xavier Glorot, et al. “Early Visual Concept Learning with Unsupervised Deep Learning”. In: (2016). arXiv: 1606.05579. URL: http://arxiv.org/abs/1606.05579.

[Hig+17a] Irina Higgins, Loic Matthey, Arka Pal, et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *Iclr* 7 (2017), pp. 1–13. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[Hig+17b] Irina Higgins, Arka Pal, Andrei A. Rusu, et al. “DARLA: Improving Zero-Shot Transfer in Reinforcement Learning”. In: (2017). ISSN: 1938-7228. arXiv: 1707.08475. URL: http://arxiv.org/abs/1707.08475.

[Hig+17c] Irina Higgins, Nicolas Sonnerat, Loic Matthey, et al. “SCAN: Learning Abstract Hierarchical Compositional Visual Concepts”. In: (2017). arXiv: 1707.03389 [stat.ML].

[Hil+11] J. Hilgert and K.H. Neeb. *Structure and Geometry of Lie Groups.* Springer Monographs in Mathematics. Springer New York, 2011. ISBN: 9780387847948. URL: `https://books.google.de/books?id=PYWoqskGw1YC`.

[Hin+94] Geoffrey E. Hinton and Richard S. Zemel. "Autoencoders, Minimum Description Length and Helmholtz free Energy". In: *Advances in Neural Information Processing Systems* 3.3 (1994), pp. –. ISSN: 15205207. DOI: `10.1021/jp906511z`. URL: `https://www.cs.toronto.edu/$\sim$hinton/absps/cvq.pdf`.

[Hot36] Harold Hotelling. "Relations Between Two Sets of Variates". In: *Biometrika* 28.3/4 (1936), pp. 321–377. ISSN: 00063444. URL: `http://www.jstor.org/stable/2333955`.

[Hua+07] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007. URL: `http://vis-www.cs.umass.edu/lfw/`.

[Hui+08] Mark J. Huiskes and Michael S. Lew. "The MIR Flickr Retrieval Evaluation". In: *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval.* Vancouver, Canada: ACM, 2008. URL: `https://press.liacs.nl/mirflickr/`.

[Hul94] J. J. Hull. "A Database for Handwritten Text Recognition Research". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 16.5 (May 1994), 550–554. ISSN: 0162-8828. DOI: `10.1109/34.291440`. URL: `https://doi.org/10.1109/34.291440`.

[Hun07] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: `10.1109/MCSE.2007.55`.

[Im+15] Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. *Denoising Criterion for Variational Auto-Encoding Framework.* 2015. arXiv: `1511.06406 [cs.LG]`.

[Iof+15] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* 2015. arXiv: `1502.03167 [cs.LG]`.

[Iva+14] Serena Ivaldi, Vincent Padois, and Francesco Nori. *Tools for dynamics simulation of robots: a survey based on user feedback.* 2014. arXiv: `1402.7050 [cs.RO]`.

[Jac17] Z. Jackson. *free-spoken-digit-dataset.* 2017. URL: `https://github.com/Jakobovski/decoupled-multimodal-learning` (visited on Apr. 1, 2020).

[Jen06] J. L W V Jensen. “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”. In: *Acta Mathematica* (1906). ISSN: 00015962. DOI: `10.1007/BF02418571`.

[Kah+16] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, et al. “EmoNets: Multimodal deep learning approaches for emotion recognition in video”. In: *Journal on Multimodal User Interfaces* (2016). ISSN: 17838738. DOI: `10.1007/s12193-015-0195-2`. arXiv: `arXiv:1503.01800v2`.

[Kem+16] Michal Kempka, Marek Wydmuch, Grzegorz Runc, et al. “ViZDoom: A Doom-based AI research platform for visual reinforcement learning”. In: *2016 IEEE Conference on Computational Intelligence and Games (CIG)* (Sept. 2016). DOI: `10.1109/cig.2016.7860433`. URL: `http://dx.doi.org/10.1109/CIG.2016.7860433`.

[Kim+18] Hyunjik Kim and Andriy Mnih. “Disentangling by Factorising”. In: (2018). arXiv: `arXiv:1802.05983v2`.

[Kin+13] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *CoRR* abs/1312.6 (2013). arXiv: `1312.6114`. URL: `http://arxiv.org/abs/1312.6114`.

[Kin+14a] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* 2014. arXiv: `1412.6980 [cs.LG]`.

[Kin+14b] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. “Semi-Supervised Learning with Deep Generative Models”. In: (2014), pp. 1–9. ISSN: 10495258. arXiv: `1406.5298`. URL: `http://arxiv.org/abs/1406.5298`.

[Kin17] Diederik P. Kingma. “Variational inference & deep learning - A new synthesis”. PhD thesis. University of Amsterdam, 2017, pp. 1–162. ISBN: 978-94-6299-745-5. URL: `https://hdl.handle.net/11245.1/8e55e07f-e4be-458f-a929-2f9bc2d169e8`.

[Koc20] Wolfgang Koch. *What is Sensor Data and Information Fusion.* 2020.

[Koe+04] N. Koenig and A. Howard. “Design and use paradigms for Gazebo, an open-source multi-robot simulator”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)* 3 (2004), pp. 2149–2154. DOI: `10.1109/IROS.2004.1389727`.

[Kol+09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press, 2009. ISBN: 9780262013192.

[Kor+15] Timo Korthals, Thilo Krause, and Ulrich Rückert. "Evidence Grid Based Information Fusion for Semantic Classifiers in Dynamic Sensor Networks". In: *Machine Learning for Cyber Physical Systems* 1.1 (2015), p. 6.

[Kor+16a] Timo Korthals, Andreas Skiba, and Thilo Krause. "Einsatz Event-Basierter Systemarchitektur für Erntemaschinen zur Elektronischen Umfelderkennung". In: *74. Tagung LAND.TECHNIK*. VDI e.V., 2016.

[Kor+16b] Timo Korthals, Andreas Skiba, Thilo Krause, and Thorsten Jungeblut. "Evidenzkarten-basierte Sensorfusion zur Umfelderkennung und Interpretation in der Ernte". In: *Informatik in der Land-, Forst und Ernährungswirtschaft*. 2016, pp. 15–18.

[Kor+16c] Timo Korthals, Marvin Barther, T. Schöpping, et al. "Occupancy Grid Mapping with Highly Uncertain Range Sensors based on Inverse Particle Filters". In: *ICINCO 2016 - Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics* 2 (2016).

[Kor+17a] Timo Korthals, Julian Exner, Thomas Schöpping, et al. "Semantical Occupancy Grid Mapping Framework". In: *2017 European Conference on Mobile Robots, ECMR 2017*. IEEE, 2017. ISBN: 9781538610961. DOI: `10.1109/ECMR.2017.8098673`.

[Kor+17b] Timo Korthals, Mikkel Kragh, Peter Christiansen, and Ulrich Rückert. "Towards Inverse Sensor Mapping in Agriculture". In: *IROS 2017 Workshop on Agricultural Robotics: learning from Industry 4.0 and moving into the future*. Vancouver, 2017.

[Kor+18a] Timo Korthals, Jürgen Leitner, and Ulrich Rückert. "Coordinated Heterogeneous Distributed Perception based on Latent Space Representation". In: *IROS 2018 Second Workshop on Multi-robot Perception-Driven Control and Planning*. 2018. arXiv: `arXiv:1809.04558v1`. URL: `https://arxiv.org/abs/1809.04558`.

[Kor+18b] Timo Korthals, Mikkel Kragh, Peter Christiansen, et al. "Obstacle Detection and Mapping in Agriculture for Process Evaluation". In: *Frontiers in Robotics and AI Robotic Control Systems* 1.1 (2018). URL: `https://www.frontiersin.org/research-topics/5597/multi-modal-sensor-fusion`.

[Kor+18c] Timo Korthals, Julian Exner, Thomas Schöpping, and Marc Hesse. "Path Evaluation via HMM on Semantical Occupancy Grid Maps". In: *ArXiv e-prints* (2018). arXiv: `1805.02944 [cs.RO]`.

[Kor+19a] Timo Korthals, Malte Schilling, and Jürgen Leitner. *A Perceived Environment Design using a Multi-Modal Variational Autoencoder for learning Active-Sensing*. 2019. arXiv: `1911.00584 [cs.RO]`. URL: `https://sites.google.com/site/dpgmcar2019/home`.

[Kor+19b] Timo Korthals, Daniel Wolf, Daniel Rudolph, et al. "Fiducial Marker based Extrinsic Camera Calibration for a Robot Benchmarking Platform". In: *European Conference on Mobile Robots, ECMR 2019, Prague, CZ, September 4-6, 2019*. 2019, pp. 1–6.

[Kor+19c] Timo Korthals, Marc Hesse, Jürgen Leitner, et al. "Jointly Trained Variational Autoencoder for Multi-Modal Sensor Fusion". In: *22st International Conference on Information Fusion, FUSION 2019, Ottawa, CA, July 2-5, 2019*. 2019, pp. 1–8.

[Kor+19d] Timo Korthals, Daniel Rudolph, Jürgen Leitner, et al. "Multi-Modal Generative Models for Learning Epistemic Active Sensing". In: *2019 IEEE International Conference on Robotics and Automation, ICRA 2019, Montreal, CA, May 20-25, 2019*. Montreal, Canada, 2019.

[Kor+19e] Timo Korthals, Andrew Melnik, Marc Hesse, and Jürgen Leitner. "Multisensory Assisted In-hand Manipulation of Objects with a Dexterous Hand". In: *2019 IEEE International Conference on Robotics and Automation Workshop on Integrating Vision and Touch for Multimodal and Cross-modal Perception, ViTac 2019, Montreal, CA, May 20-25, 2019*. 2019, pp. 1–2. URL: `http://wordpress.csc.liv.ac.uk/smartlab/icra-2019-vitac-workshop/`.

[Kor19] Timo Korthals. *M^2VAE - Derivation of a Multi-Modal Variational Autoencoder Objective from the Marginal Joint Log-Likelihood*. 2019. arXiv: `arXiv:1903.07303`. URL: `http://arxiv.org/abs/1903.07303`.

[Kou16] A. Koubaa. *Robot Operating System (ROS): The Complete Reference*. Vol. 1. 1. Springer International Publishing, 2016. ISBN: 9783319260549. DOI: `10.1007/978-3-319-26054-9`. URL: `http://courses.csail.mit.edu/6.141/spring2012/pub/lectures/Lec06-ROS.pdf`.

[Kra+16] Mikkel Kragh, Peter Christiansen, Timo Korthals, et al. "Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture". In: *International Conference on Agricultural Engineering.* Aarhus, 2016. URL: `http://conferences.au.dk/uploads/tx_powermail/2016cigr_-_multi-modal_obstacle_detection_and_evaluation_of_evidence_grid_mapping_in_agriculture.pdf`.

[Kra+17] Mikkel Fly Kragh, Peter Christiansen, Morten Stigaard Laursen, et al. "FieldSAFE: Dataset for Obstacle Detection in Agriculture". In: *Sensors* 17.11 (2017).

[Kre+05] Chris Kreucher, Keith Kastella, and Alfred O. Hero. "Sensor management using an active sensing approach". In: *Signal Processing* 85.3 (2005), pp. 607–624. ISSN: 01651684. DOI: `10.1016/j.sigpro.2004.11.004`.

[Kri09] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *... Science Department, University of Toronto, Tech. ...* (2009). ISSN: 1098-6596. DOI: `10.1.1.222.9220`. arXiv: `arXiv:1011.1669v3`. URL: `https://www.cs.toronto.edu/~kriz/cifar.html`.

[Kul+51] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* (1951). ISSN: 0003-4851. DOI: `10.1214/aoms/1177729694`.

[Kul59] Solomon Kullback. *Information Theory and Statistics.* New York: Wiley, 1959.

[Lak+15] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266 (2015), pp. 1332–1338. ISSN: 0036-8075. DOI: `10.1126/science.aab3050`. eprint: `https://science.sciencemag.org/content/350/6266/1332.full.pdf`. URL: `https://science.sciencemag.org/content/350/6266/1332`.

[Lar+07] Hugo Larochelle, Dumitru Erhan, Aaron Courville, et al. "An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation". In: *Proceedings of the 24th International Conference on Machine Learning.* ICML '07. New York, NY, USA: ACM, 2007, pp. 473–480. ISBN: 978-1-59593-793-3. DOI: `10.1145/1273496.1273556`. URL: `http://doi.acm.org/10.1145/1273496.1273556`.

[Lar+16] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48.* New York, NY, USA, 2016, pp. 1558–1566. arXiv: `arXiv:1512.09300v2`.

[LeC+98] Yann LeCun, Corinna Cortes, and Christopher Burges. “THE MNIST DATABASE of handwritten digits”. In: *The Courant Institute of Mathematical Sciences* (1998). URL: `http://yann.lecun.com/exdb/mnist/`.

[Lex04] Michael Lexa. “Useful Facts about the Kullback-Leibler Discrimination Distance”. In: *None* (2004).

[Li+16] Fan Li, Natalia Neverova, Christian Wolf, and Graham Taylor. “Modout: Learning to Fuse Modalities via Stochastic Regularization”. In: 2016.

[Li+17] Yang Li, Quan Pan, Suhang Wang, et al. “Disentangled Variational Auto-Encoder for Semi-supervised Learning”. In: (2017), pp. 1–10. ISSN: 1063-6919. DOI: `10.1109/CVPR.2017.90`. arXiv: `1709.05047`. URL: `http://arxiv.org/abs/1709.05047`.

[Lig+01] Martin E. Liggins, David L. Hall, and David Llinas. *Handbook of multisensor data fusion*. Taylor & Francis Ltd, 2001, p. 537. ISBN: 978-1420053081. URL: `http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Handbook+of+multisensor+data+fusion#2`.

[Lig+08] Martin E. Liggins, David L. Hall, and David Llinas. *Handbook of multisensor data fusion*. Taylor & Francis Ltd, 2008, p. 537. ISBN: 9781420053098. URL: `http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Handbook+of+multisensor+data+fusion#2`.

[Lin91] Jianhua Lin. “Divergence Measures Based on the Shannon Entropy”. In: *IEEE Transactions on Information Theory* (1991). ISSN: 15579654. DOI: `10.1109/18.61115`.

[Lip16] Zachary Chase Lipton. “The Mythos of Model Interpretability”. In: *CoRR* abs/1606.03490 (2016). arXiv: `1606.03490`. URL: `http://arxiv.org/abs/1606.03490`.

[Liu+15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015. URL: `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`.

[Liu+17a] Guan-Horng Liu, Avinash Siravuru, Sai Prabhakar, et al. “Learning End-to-end Multimodal Sensor Policies for Autonomous Navigation”. In: *CoRR* abs/1705.10422 (2017). arXiv: `1705.10422`. URL: `http://arxiv.org/abs/1705.10422`.

[Liu+17b] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. *Unsupervised Image-to-Image Translation Networks*. 2017. arXiv: `1703.00848 [cs.CV]`.

[Loc+18] Francesco Locatello, Stefan Bauer, Mario Lucic, et al. *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations.* 2018. arXiv: `1811.12359 [cs.LG]`.

[Luc+18] Mario Lucic, Karol Kurach, Marcin Michalski, et al. "Are GANs Created Equal? A Large-Scale Study". In: *Advances in Neural Information Processing Systems 31.* Ed. by S. Bengio, H. Wallach, H. Larochelle, et al. Curran Associates, Inc., 2018, pp. 700–709. URL: `http://papers.nips.cc/paper/7350-are-gans-created-equal-a-large-scale-study.pdf`.

[Luc+19] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. *Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse.* 2019. arXiv: `1911.02469 [cs.LG]`.

[Maa+16] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. *Auxiliary Deep Generative Models.* 2016. arXiv: `1602.05473 [stat.ML]`.

[Mah+16] A. Mahendran, H. Bilen, J. F. Henriques, and A. Vedaldi. *ResearchDoom and CocoDoom: Learning Computer Vision with Games.* 2016. arXiv: `1610.02431 [cs.CV]`.

[Mar+10] Benjamin Marlin, Kevin Swersky, Bo Chen, and Nando Freitas. "Inductive Principles for Restricted Boltzmann Machine Learning". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 509–516. URL: `http://proceedings.mlr.press/v9/marlin10a.html`.

[Mar+19] Manuel Martin, Alina Roitberg, Monica Haurilet, et al. "Drive&Act: A Multi-Modal Dataset for Fine-Grained Driver Behavior Recognition in Autonomous Vehicles". In: *The IEEE International Conference on Computer Vision (ICCV).* Oct. 2019. URL: `https://www.driveandact.com/`.

[Mat+17] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. *dSprites: Disentanglement testing Sprites dataset.* 2017. URL: `https://github.com/deepmind/dsprites-dataset/` (visited on Jan. 30, 2020).

[Mat+19] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. "Disentangling Disentanglement in Variational Autoencoders". In: ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 4402–4412. URL: http://proceedings.mlr.press/v97/mathieu19a.html.

[McC+43] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: https://doi.org/10.1007/BF02478259.

[Mes+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, et al. "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). DOI: 10.1109/cvpr.2019.00459. URL: http://dx.doi.org/10.1109/CVPR.2019.00459.

[Mic+15] Tomer Michaeli, Weiran Wang, and Karen Livescu. *Nonparametric Canonical Correlation Analysis.* 2015. arXiv: 1511.04839 [cs.LG].

[Mih+02] L Mihaylova, T Lefebvre, H Bruyninckx, et al. "Active Sensing for Robotics – A Survey". In: *Robotics* (2002), pp. 1–8. DOI: 10.1.1.18.5320. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.5320.

[MJH+10] B. Thomee Mark J. Huiskes and Michael S. Lew. "New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative". In: *MIR '10: Proceedings of the 2010 ACM International Conference on Multimedia Information Retrieval.* Philadelphia, USA: ACM, 2010, pp. 527–536. URL: https://press.liacs.nl/mirflickr/.

[Mni+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. "Playing Atari with Deep Reinforcement Learning". In: (2013), pp. 1–9. ISSN: 0028-0836. DOI: 10.1038/nature14236. arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[Mni+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533. ISSN: 0028-0836. DOI: 10.1038/nature14236. arXiv: 1312.5602. URL: http://www.nature.com/doifinder/10.1038/nature14236.

[Net+11] Yuval Netzer, Tao Wang, Adam Coates, et al. *Reading Digits in Natural Images with Unsupervised Feature Learning.* 2011. URL: http://ufldl.stanford.edu/housenumbers/.

[Nev+16] Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. "ModDrop: Adaptive Multi-Modal Gesture Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.8 (Aug. 2016), 1692–1706. ISSN: 2160-9292. DOI: `10 . 1109 / tpami . 2015 . 2461544`. URL: `http://dx.doi.org/10.1109/TPAMI.2015.2461544`.

[Ngi+11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, et al. "Multimodal Deep Learning". In: *Proceedings of The 28th International Conference on Machine Learning (ICML)* (2011). ISSN: 9781450306195. DOI: `10 . 1145/2647868.2654931`. arXiv: `1502.07209`.

[Nui+19] Yue Leire Erro Nuin, Nestor Gonzalez Lopez, Elias Barba Moral, et al. *ROS2Learn: a reinforcement learning framework for ROS 2*. 2019. eprint: `arXiv:1903.06282`. URL: `https://github.com/AcutronicRobotics/ros2learn`.

[Ofl+13] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, et al. "Berkeley MHAD: A comprehensive Multimodal Human Action Database". In: *Proceedings of IEEE Workshop on Applications of Computer Vision*. 2013. ISBN: 9781467350532. DOI: `10.1109/WACV.2013.6474999`.

[Pan+17] Gaurav Pandey and Ambedkar Dukkipati. "Variational methods for conditional multimodal deep learning". In: *Proceedings of the International Joint Conference on Neural Networks* 2017-May (2017), pp. 308–315. DOI: `10 . 1109 / IJCNN . 2017 . 7965870`. arXiv: `1603.01801`.

[Pat+17] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. "Curiosity-driven Exploration by Self-supervised Prediction". In: *CoRR* abs/1705.05363 (2017). arXiv: `1705 . 05363`. URL: `http : / / arxiv.org/abs/1705.05363`.

[Pay+09] Pascal Paysan, Reinhard Knothe, Brian Amberg, et al. "A 3D Face Model for Pose and Illumination Invariant Face Recognition". In: *AVSS*. Ed. by Stefano Tubaro and Jean-Luc Dugelay. IEEE Computer Society, 2009, pp. 296–301. ISBN: 978-0-7695-3718-4. URL: `http : / / dblp . uni - trier.de/db/conf/avss/avss2009.html#PaysanKARV09`.

[Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Per+10] G. Perry, E. T. Rolls, and S. M. Stringer. “Continuous transformation learning of translation invariant representations”. In: *Experimental Brain Research* 204.2 (2010), pp. 255–270. ISSN: 00144819. DOI: 10.1007/s00221-010-2309-0.

[Ram+17a] Dhanesh Ramachandram and Graham W Taylor. “Deep Multimodal Learning: A Survey on Recent Advances and Trends”. In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017), pp. 96–108. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2738401.

[Ram+17b] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. “Searching for Activation Functions”. In: *CoRR* abs/1710.05941 (2017). arXiv: 1710.05941. URL: http://arxiv.org/abs/1710.05941.

[Ran+06] Marc’Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun. “Efficient Learning of Sparse Representations with an Energy-based Model”. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS’06. Cambridge, MA, USA: MIT Press, 2006, pp. 1137–1144. URL: http://dl.acm.org/citation.cfm?id=2976456.2976599.

[Res+11] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, et al. “Detecting Novel Associations in Large Data Sets”. In: *Science* 334.6062 (2011), pp. 1518–1524. ISSN: 0036-8075. DOI: 10.1126/science.1205438. eprint: https://science.sciencemag.org/content/334/6062/1518.full.pdf. URL: https://science.sciencemag.org/content/334/6062/1518.

[Rez+14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*. 2014. arXiv: 1401.4082 [stat.ML].

[Ric+16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. “Playing for Data: Ground Truth from Computer Games”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 102–118. ISBN: 978-3-319-46475-6.

[Rou18] Simon Brodeur; Simon Carrier; Jean Rouat. *CREATE: Multimodal Dataset for Unsupervised Learning and Generative Modeling of Sensory Data from a Mobile Robot*. 2018. (Visited on Jan. 30, 2020).

[Row+00] Sam T. Roweis and Lawrence K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2323. eprint: https://science.sciencemag.org/content/290/

5500/2323.full.pdf. URL: https://science.sciencemag.org/content/290/5500/2323.

[RS+17] J.R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez. "Robot@Home, a robotic dataset for semantic mapping of home environments". In: *The International Journal of Robotics Research* 36.2 (2017), pp. 131–141. DOI: 10.1177/0278364917695640. eprint: https://doi.org/10.1177/0278364917695640. URL: https://doi.org/10.1177/0278364917695640.

[Rum+86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: https://doi.org/10.1038/323533a0.

[Rus+15] Olga Russakovsky, Jia Deng, Hao Su, et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[Sal+08] Ruslan Salakhutdinov and Iain Murray. "On the Quantitative Analysis of Deep Belief Networks". In: *Proceedings of the 25th International Conference on Machine Learning.* ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, 872–879. ISBN: 9781605582054. DOI: 10.1145/1390156.1390266. URL: https://doi.org/10.1145/1390156.1390266.

[Sal+09] Ruslan Salakhutdinov and Geoffrey Hinton. "Deep Boltzmann Machines". In: ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, Apr. 2009, pp. 448–455. URL: http://proceedings.mlr.press/v5/salakhutdinov09a.html.

[Sal+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, et al. "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems 29.* Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, et al. Curran Associates, Inc., 2016, pp. 2234–2242. URL: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

[Sav+18] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. *OpenAI Gym wrapper for ViZDoom environments.* 2018. (Visited on Jan. 30, 2020).

[Sch+18] Philip Schmidt, Attila Reiss, Robert Duerichen, and Kristof Van Laerhoven. “Introducing WeSAD - a multimodal dataset for wearable stress and affect detection”. In: *ICMI 2018 - Proceedings of the 2018 International Conference on Multimodal Interaction.* 2018. ISBN: 9781450356923. DOI: `10.1145/3242969.3242985`.

[She+16] Bhavin R Sheth and Ryan Young. “Two Visual Pathways in Primates Based on Sampling of Space: Exploitation and Exploration of Visual Information”. In: *Frontiers in Integrative Neuroscience* 10 (2016), p. 37. ISSN: 1662-5145. DOI: `10.3389/fnint.2016.00037`. URL: `https://www.frontiersin.org/article/10.3389/fnint.2016.00037`.

[Shi+19] Yuge Shi, N. Siddharth, Brooks Paige, and Philip H. S. Torr. *Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models.* 2019. arXiv: `1911.03393 [stat.ML]`.

[Smi17] Leslie N. Smith. “Cyclical Learning Rates for Training Neural Networks”. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Mar. 2017). DOI: `10.1109/wacv.2017.58`. URL: `http://dx.doi.org/10.1109/WACV.2017.58`.

[Soh+15a] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: *Advances in Neural Information Processing Systems 28.* Ed. by C Cortes, N D Lawrence, D D Lee, et al. Curran Associates, Inc., 2015, pp. 3483–3491. URL: `http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf`.

[Soh+15b] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: *Advances in Neural Information Processing Systems 28.* Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, et al. Curran Associates, Inc., 2015, pp. 3483–3491. URL: `http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf`.

[Sri+12] Nitish Srivastava and Ruslan Salakhutdinov. “Multimodal Learning with Deep Boltzmann Machines”. In: *Advances in neural information processing systems (NIPS).* 2012. ISBN: 978-0-7695-4989-7. DOI: `10.1109/CVPR.2013.49`.

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. ISSN: 15337928. DOI: `10.1214/12-AOS1000`. arXiv: `1102.4807`.

[Sun+14] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Representation by Joint Identification-Verification". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2.* NIPS'14. Montreal, Canada: MIT Press, 2014, 1988–1996.

[Sun+19] Xudong Sun and Bernd Bischl. *Tutorial and Survey on Probabilistic Graphical Model and Variational Inference in Deep Reinforcement Learning.* 2019. arXiv: `1908.09381 [cs.LG]`.

[Sut+18] Richard S Sutton and Andrew G Barto. *Introduction to Reinforcement Learning.* 2nd ed. Bradford Books, 2018. ISBN: 978-0262039246. URL: `http://incompleteideas.net/book/the-book-2nd.html`.

[Suz+17] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. "Joint multimodal learning with deep generative models". In: (2017), pp. 1–12. arXiv: `arXiv:1611.01891v1`. URL: `https://arxiv.org/abs/1611.01891`.

[Søn+16] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, et al. "Ladder Variational Autoencoders". In: Nips (2016). ISSN: 10495258. arXiv: `1602.02282`. URL: `http://arxiv.org/abs/1602.02282`.

[Tai+16] Lei Tai, Jingwei Zhang, Ming Liu, et al. "A Survey of Deep Network Solutions for Learning Control in Robotics: From Reinforcement to Imitation". In: 14.8 (2016), pp. 1–19. arXiv: `1612.07139`. URL: `http://arxiv.org/abs/1612.07139`.

[Tie+12] T. Tieleman and G. Hinton. *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.* 2012.

[Tip+99] Michael E. Tipping and Chris M. Bishop. "Probabilistic Principal Component Analysis". In: *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 61.3 (1999), pp. 611–622.

[Tsa+18] Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, et al. *Learning Factorized Multimodal Representations.* 2018. arXiv: `1806.06176 [cs.LG]`.

[Uda16] Udacity. *Self-Driving Car: Annotated Driving Dataset.* 2016. URL: `https://github.com/udacity/self-driving-car/tree/master/annotations` (visited on Sept. 21, 2018).

[Uur+18] Viivi Uurtio, João M. Monteiro, Jaz Kandola, et al. "A Tutorial on Canonical Correlation Methods". In: *ACM Computing Surveys* 50.6 (Jan. 2018), 1–33. ISSN: 1557-7341. DOI: `10.1145/3136624`. URL: `http://dx.doi.org/10.1145/3136624`.

[Ved+17] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. "Generative Models of Visually Grounded Imagination". In: (2017), pp. 1–21. arXiv: 1705.10762. URL: http://arxiv.org/abs/1705.10762.

[Vie+19a] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. "CentralNet: A Multilayer Approach for Multimodal Fusion". In: *Computer Vision – ECCV 2018 Workshops* (2019), 575–589. ISSN: 1611-3349. DOI: 10.1007/978-3-030-11024-6_44. URL: http://dx.doi.org/10.1007/978-3-030-11024-6_44.

[Vie+19b] Valentin Vielzeuf, Alexis Lechervy, Stephane Pateux, and Frederic Jurie. "Multilevel Sensor Fusion With Deep Learning". In: *IEEE Sensors Letters* 3.1 (Jan. 2019), 1–4. ISSN: 2475-1472. DOI: 10.1109/lsens.2018.2878908. URL: http://dx.doi.org/10.1109/LSENS.2018.2878908.

[Vio+18] Fabio Viola and Louise Deason. *GQN Dataset.* 2018. (Visited on Jan. 30, 2020).

[Vir+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[Vri+15] Michalis Vrigkas, Christophoros Nikou, and Ioannis A Kakadiaris. "A Review of Human Activity Recognition Methods". In: *Frontiers in Robotics and AI* 2 (2015), p. 28. ISSN: 2296-9144. DOI: 10.3389/frobt.2015.00028. URL: https://www.frontiersin.org/article/10.3389/frobt.2015.00028.

[Wah+11] C. Wah, S. Branson, P. Welinder, et al. *The Caltech-UCSD Birds-200-2011 Dataset.* Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011. URL: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html.

[Wan+09] Chang Wang and Sridhar Mahadevan. "Manifold alignment without correspondence". In: *Twenty-First International Joint Conference on Artificial Intelligence.* 2009.

[Wan+11] Chang Wang, Peter Krafft, and Sridhar Mahadevan. "Chapter 5 - Manifold Alignment". In: *Manifold Learning Theory and Applications.* 1st. USA: CRC Press, Inc., 2011. ISBN: 1439871094.

[Wan+16] Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. "Deep Variational Canonical Correlation Analysis". In: 1 (2016). arXiv: 1610.03454. URL: http://arxiv.org/abs/1610.03454.

[Wat+19a] Nicholas Watters, Loic Matthey, Matko Bosnjak, et al. *COBRA: Data-Efficient Model-Based RL through Unsupervised Object Discovery and Curiosity-Driven Exploration.* 2019. arXiv: 1905.09275 [cs.LG].

[Wat+19b] Nicholas Watters, Loic Matthey, Christopher P. Burgess, and Alexander Lerchner. *Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs.* 2019. arXiv: 1901.07017 [cs.LG].

[Wat+19c] Nicholas Watters, Loic Matthey, Sebastian Borgeaud, et al. *Spriteworld: A Flexible, Configurable Reinforcement Learning Environment.* 2019. URL: https://github.com/deepmind/spriteworld/ (visited on Jan. 30, 2020).

[Wei+16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3.1 (2016), p. 9. ISSN: 2196-1115. DOI: 10.1186/s40537-016-0043-6. URL: https://doi.org/10.1186/s40537-016-0043-6.

[Wel+10] P. Welinder, S. Branson, T. Mita, et al. *Caltech-UCSD Birds 200.* Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010. URL: http://www.vision.caltech.edu/visipedia/CUB-200.html.

[Wen19] Lilian Weng. *Domain Randomization for Sim2Real Transfer.* 2019. (Visited on Jan. 30, 2020).

[Wes+18] Rob Weston, Sarah Cen, Paul Newman, and Ingmar Posner. *Probably Unknown - Deep Inverse Sensor Modelling Radar.* Tech. rep. 2018, p. 6. arXiv: arXiv:1810.08151v1. URL: https://arxiv.org/abs/1810.08151.

[Wu+18] Mike Wu and Noah Goodman. "Multimodal Generative Models for Scalable Weakly-Supervised Learning". In: (2018). arXiv: 1802.05335. URL: http://arxiv.org/abs/1802.05335.

[Xia+14] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. "Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild". In: *IEEE Winter Conference on Applications of Computer Vision (WACV).* 2014. URL: https://cvgl.stanford.edu/projects/pascal3d.html.

[Xia+17] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: (2017), pp. 1–6. arXiv: 1708 . 07747. URL: http://arxiv.org/abs/1708.07747.

[Yan+19] John Yang, Gyuejeong Lee, Simyung Chang, and Nojun Kwak. "Towards Governing Agent's Efficacy: Action-Conditional β-VAE for Deep Transparent Reinforcement Learning". In: *Proceedings of The Eleventh Asian Conference on Machine Learning.* Ed. by Wee Sun Lee and Taiji Suzuki. Vol. 101. Proceedings of Machine Learning Research. Nagoya, Japan: PMLR, Nov. 2019, pp. 32–47. URL: `http://proceedings.mlr.press/v101/yang19a.html`.

[Yar+14] Stuart Yarrow, Khaleel A. Razak, Aaron R. Seitz, and Peggy Seriès. "Detecting and quantifying topography in neural maps". In: *PLoS ONE* (2014). ISSN: 19326203. DOI: `10.1371/journal.pone.0087178`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3914801/`.

[Yu+09] Shipeng Yu, Balaji Krishnapuram, Romer Rosales, and R Bharat Rao. "Active Sensing". In: *Aistats* (2009), pp. 639–646.

[Yu+14a] A. Yu and K. Grauman. "Fine-Grained Visual Comparisons with Local Learning". In: *Computer Vision and Pattern Recognition (CVPR).* June 2014. URL: `http://vision.cs.utexas.edu/projects/finegrained/`.

[Yu+14b] A. Yu and K. Grauman. *UT Zappos50K.* 2014. URL: `http://vision.cs.utexas.edu/projects/finegrained/utzap50k/` (visited on Jan. 30, 2020).

[Yu+17] A. Yu and K. Grauman. "Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images". In: *International Conference on Computer Vision (ICCV).* Oct. 2017. URL: `http://vision.cs.utexas.edu/projects/semjitter/`.

[Zei12] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method.* 2012. arXiv: `1212.5701 [cs.LG]`.

[Zha+19] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. "Advances in Variational Inference". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8 (Aug. 2019), 2008–2026. ISSN: 1939-3539. DOI: `10.1109/tpami.2018.2889774`. URL: `http://dx.doi.org/10.1109/TPAMI.2018.2889774`.

[Zhe+18] Xin Zheng, Yanqing Guo, Huaibo Huang, et al. *A Survey of Deep Facial Attribute Analysis.* 2018. arXiv: `1812.10265 [cs.CV]`.

[Zhu+18] Jun-Yan Zhu, Taesung Park, Mihaela Rosca, et al. *CVPR 2018 Tutorial on GANs.* 2018. URL: `https://sites.google.com/view/cvpr2018tutorialongans/`.

[Bal+19] T. Baltrušaitis, C. Ahuja, and L. Morency. “Multimodal Machine Learning: A Survey and Taxonomy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (Feb. 2019), pp. 423–443. ISSN: 1939-3539. DOI: `10.1109/TPAMI.2018.2798607`.

[Chr00] Christian Delis. “Entwicklung einer Kooperationsplattform für den Hokuyo URG-04LX Laserscanner”. In: *Bachelor-Arbeit* 23.6 (2000), pp. 376–377.

[Den+09] J. Deng, W. Dong, R. Socher, et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* June 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[Ger+99] F. A. Gers, J. Schmidhuber, and F. Cummins. “Learning to forget: continual prediction with LSTM”. In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470).* Vol. 2. 1999, 850–855 vol.2.

[Goo20a] Google LLC. *Google Scholar Top Publications in Engineering and Computer Science on Artificial Intelligence.* 2020. URL: `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_artificialintelligence` (visited on May 25, 2020).

[Goo20b] Google LLC. *Google Scholar Top Publications in Engineering and Computer Science on Computer Vision & Pattern Recognition.* 2020. URL: `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_computervisionpatternrecognition` (visited on May 25, 2020).

[Goo20c] Google LLC. *Google Scholar Top Publications in Engineering and Computer Science on Robotics.* 2020. URL: `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_robotics` (visited on May 25, 2020).

[Ins97] A. Inselberg. “Multidimensional detective”. In: *Proceedings of VIZ ’97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium.* Oct. 1997, pp. 100–107. DOI: `10.1109/INFVIS.1997.636793`.

[Lai+11] K. Lai, L. Bo, X. Ren, and D. Fox. “A large-scale hierarchical multi-view RGB-D object dataset”. In: *2011 IEEE International Conference on Robotics and Automation.* May 2011, pp. 1817–1824. DOI: `10.1109/ICRA.2011.5980382`.

[Lec+98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[Li +06] Li Fei-Fei, R. Fergus, and P. Perona. “One-shot learning of object categories”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (Apr. 2006), pp. 594–611. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2006.79.

[Mat+02] I. Matthews, T. F. Cootes, J. A. Bangham, et al. “Extraction of visual features for lipreading”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (Feb. 2002), pp. 198–213. ISSN: 1939-3539. DOI: 10.1109/34.982900.

[Mey+15] Sebastian Meyer zu Borgsen, Timo Korthals, Leon Ziegler, and Sven Wachsmuth. “ToBI-Team of Bielefeld The Human-Robot Interaction System for RoboCup@Home 2015”. In: 2015.

[Mey+16] Sebastian Meyer zu Borgsen, Timo Korthals, and Sven Wachsmuth. “ToBI-Team of Bielefeld The Human-Robot Interaction System for RoboCup@Home 2016”. In: 2016.

[McK10] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56 –61. DOI: 10.25080/Majora-92bf1922-00a.

[Pan+17] G. Pandey and A. Dukkipati. “Variational methods for conditional multimodal deep learning”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 308–315.

[Pat+02] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. “CUAVE: A new audio-visual database for multimodal human-computer interface research”. In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. May 2002, pp. II–2017–II–2020. DOI: 10.1109/ICASSP.2002.5745028.

[Per+19] J. Perez-Rua, V. Vielzeuf, S. Pateux, et al. “MFAS: Multimodal Fusion Architecture Search”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019, pp. 6959–6968. DOI: 10.1109/CVPR.2019.00713.

[Sze+16] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. “Rethinking the Inception Architecture for Computer Vision”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[Tai+14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1701–1708. DOI: 10.1109/CVPR.2014.220.

[Tod+12] E. Todorov, T. Erez, and Y. Tassa. “MuJoCo: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.

[Tor+08] A. Torralba, R. Fergus, and W. T. Freeman. “80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.11 (Nov. 2008), pp. 1958–1970. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2008.128.

[Van+09] L J P Van Der Maaten, E O Postma, and H J Van Den Herik. “Dimensionality Reduction: A Comparative Review”. In: *Journal of Machine Learning Research* (2009). ISSN: 0169328X. DOI: 10.1080/13506280444000102.

[de +14] Suzana de Siqueira Santos, Daniel Yasumasa Takahashi, Asuka Nakata, and André Fujita. “A comparative study of statistical methods used to identify dependencies between gene expression signals.” eng. In: *Briefings in bioinformatics* 15.6 (Nov. 2014), pp. 906–918. ISSN: 1477-4054 (Electronic). DOI: 10.1093/bib/bbt051.

[van+08] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

A Most Influential Conferences and Journals

The lists represent the 20 most influential engineering and computer science platforms in robotics (see table A.1) and AI (see table A.2) according to gScholar. Ranks are from top to bottom with the highest to the lowest ranked conference/journal with respect to h5-index/median.

Table A.1: 20 most influential engineering and computer science conferences in robotics according to gScholar [Goo20c].

Publication	index	median
IEEE International Conference on Robotics and Automation	82	113
IEEE/ASME Transactions on Mechatronics	64	80
The International Journal of Robotics Research	63	90
IEEE Transactions on Robotics	58	87
IEEE/RSJ International Conference on Intelligent Robots and Systems	58	77
Robotics and Autonomous Systems	49	75
Robotics: Science and Systems	47	80
Journal of Intelligent & Robotic Systems	43	60
Robotics and Computer-Integrated Manufacturing	42	57
Journal of Field Robotics	41	69
ACM/IEEE International Conference on Human Robot Interaction	40	58
Autonomous Robots	38	54
Mechatronics	37	54
Bioinspiration & Biomimetics	36	52
International Journal of Social Robotics	36	50
IEEE Robotics and Automation Letters	36	49
Soft Robotics	34	60
IEEE Robotics & Automation Magazine	34	49
IEEE-RAS International Conference on Humanoid Robots	33	46
International Conference on Unmanned Aircraft Systems	30	40

The Robotics and Autonomous Systems conference was substituted one time in table A.2 by the most influential conference in Engineering and Computer Science on Computer Vision & Pattern Recognition according to gScholar [Goo20b]:

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) with h5-index of 240 and h5-median of 383.

Table A.2: 20 most influential engineering and computer science conferences in AI according to gScholar [Goo20a].

Publication	index	median
Neural Information Processing Systems	169	334
International Conference on Learning Representations	150	276
International Conference on Machine Learning (ICML)	135	254
Expert Systems with Applications	105	139
IEEE Transactions On Systems, Man And Cybernetics Part B, Cybernetics	100	132
IEEE Transactions on Neural Networks and Learning Systems	96	127
AAAI Conference on Artificial Intelligence	95	153
Applied Soft Computing	83	113
Neurocomputing	83	105
The Journal of Machine Learning Research	81	143
IEEE Transactions on Fuzzy Systems	81	130
Knowledge-Based Systems	79	107
International Joint Conference on Artificial Intelligence (IJCAI)	67	100
Neural Computing and Applications	60	87
Neural Networks	57	90
International Conference on Artificial Intelligence and Statistics	52	77
Journal of Intelligent & Fuzzy Systems	51	80
Engineering Applications of Artificial Intelligence	50	68
Robotics and Autonomous Systems	49	75
Conference on Learning Theory (COLT)	48	65

B Supplemental Material

B.1 List of Applied Software

Table B.1: Applied software within this thesis.

Software	Version	Info
TensorFlow [Aba+15]	2.0.3	
NumPy [Har+20]	1.17.4	
vae_tools [Kor+19c]	1.0.0	
Python	3.5.2	shipped with Ubuntu 16.04.5
SciPy [Vir+20]	1.2.0	
scikit-learn [Ped+11]	0.21.3	
Matplotlib [Hun07]	3.0.3	
pandas [McK10]	0.24.1	

B.2 Architecture Setups and Assets

B.2.1 VAE Training Setup

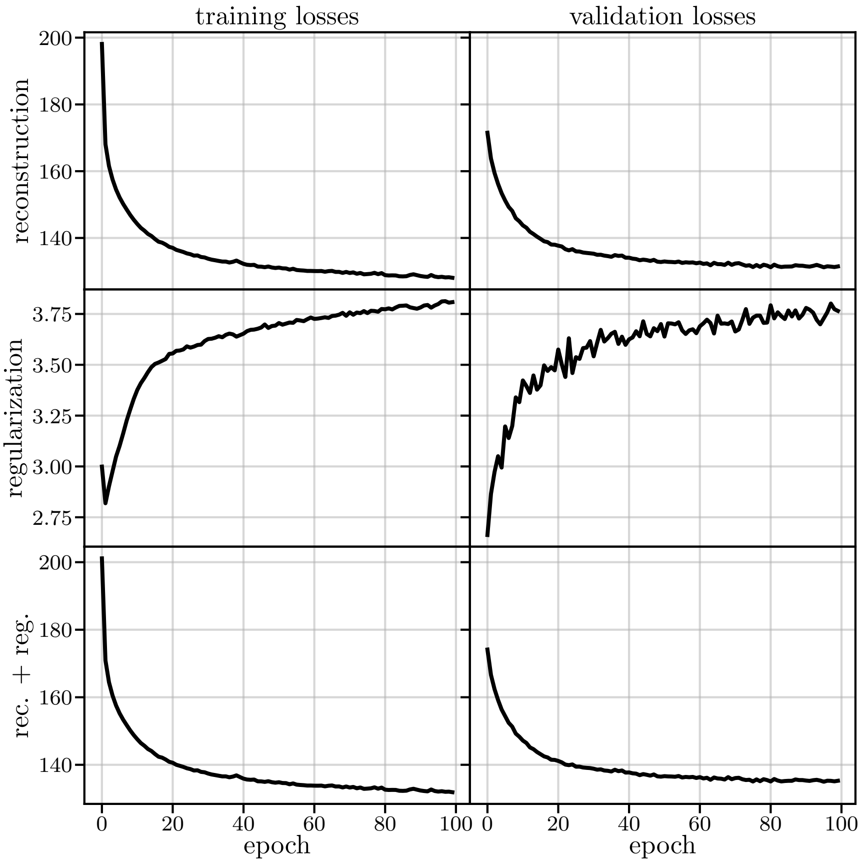


Figure B.1: Training and validation losses for the regularization and reconstruction term. The reconstruction term dominates the overall loss, due to the high dimensionality ($D_\mathrm{x} = 28 \cdot 28 = 784$ vs. $D_\mathrm{z} = 2$). The losses do scale linear with the dimensions, since they are not normalized of the of training a VAE losses MNIST images randomly generated by the VAE decoder.

Table B.2: VAE architecture and training setup.

data set	MNIST by LeCun et al. [LeC+98]
input norm.	min. 0 / max. 1
input format	flattened

D_{z}	2
D_{x}	784
$f_{\mathrm{enc.}}$	input@784 → ReLU@256 → ReLU@128
$f_{\mathrm{dec.}}$	input@2 → ReLU@128 → ReLU@256 → sig@784
$f_{\boldsymbol{\mu}}$	input@128 → lin@2
$f_{\boldsymbol{\sigma}}$	input@128 → lin@2
rec. loss	BCE
β	1.0
epochs	100
batch size	128
optimizer	Adam
LR	0.001

B.2.2 CVAE Training Setup

Table B.3: CVAE architecture and training setup.

data set	MNIST (image, label) by LeCun et al. [LeC+98]
input norm.	(min. 0 / max. 1, one-hot-encoding)
input format	(flattened, flattened)
D_{z}	2
D_{x}	(784, 10)
$f_{\mathrm{enc.}}$	input@(784, 10) → ReLU@256 → ReLU@128
$f_{\mathrm{dec.}}$	input@(2, 10) → ReLU@128 → ReLU@256 → sig@784
$f_{\boldsymbol{\mu}}$	input@128 → lin@2
$f_{\boldsymbol{\sigma}}$	input@128 → lin@2
rec. loss	BCE
β	1.0
epochs	100
batch size	128
optimizer	Adam
LR	0.001

B.2.3 Re-Encoding Training Setup

Table B.4: Common VAE architecture and training setup.

data set	MNIST by LeCun et al. [LeC+98]
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	2
D_{x}	784
$f_{\mathrm{enc.}}$	input@784 → ReLU@256 → ReLU@128
$f_{\mathrm{dec.}}$	input@2 → ReLU@128 → ReLU@256 → sig@784
$f_{\boldsymbol{\mu}}$	input@128 → lin@2
$f_{\boldsymbol{\sigma}}$	input@128 → lin@2

rec. loss	BCE
β	1.0
epochs	400
batch size	1024
optimizer	Adam
LR	0.001

Table B.5: Re-encoding VAE architecture and training setup.

data set	MNIST by LeCun et al. [LeC+98]
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	2
D_{x}	784
$f_{\mathrm{enc.}}$	input@784 → ReLU@256 → ReLU@128
$f_{\mathrm{dec.}}$	input@2 → ReLU@128 → ReLU@256 → sig@784
$f_{\boldsymbol{\mu}}$	input@128 → lin@2
$f_{\boldsymbol{\sigma}}$	input@128 → lin@2
rec. loss	BCE
β	1.0
D	D_{KL}
α	.0 for the first 200 epochs, .01 afterwards with pinned decoder weights
epochs	400
batch size	1024
optimizer	Adam
LR	0.001

B.2.4 eMNIST CVAE Training Setup

Table B.6: CVAE architecture and training setup for MNIST and FMNIST. (de)conv. denotes (de)convoutional layers with the following nomenclature: (#filter)(kernel-size)(stride)(padding)(activation)

data set	MNIST (image, label) [LeC+98] or FMNIST (image, label) [Xia+17]
input norm.	(min. 0 / max. 1, one-hot-encoding)
input format	(original, flattened)
D_{z}	2
D_{x}	$((28, 28)$, 10)
$f_{\mathrm{enc.}}$	input@((28,28)) → conv@(1)(2,2)(1)(same)(ReLU) → conv@(64)(2,2)(2)(same)(ReLU) → conv@(64)(3,3)(1)(same)(ReLU) → conv@(64)(3,3)(1)(same)(ReLU) → flatten & concat. w/ label input@10 → ReLU@128

$f_{\text{dec.}}$	input@(2, 10) → ReLU@128 → ReLU@12544 → reshape to $(64, 14, 14)$ → deconv@(64)(3,3)(1)(same)(ReLU) → deconv@(64)(3,3)(1)(same)(ReLU) → deconv@(64)(3,3)(2)(valid)(ReLU)→ deconv@(1)(2,2)(1)(valid)(sig)
$f_{\boldsymbol{\mu}}$	input@128 → lin@2
$f_{\boldsymbol{\sigma}}$	input@128 → lin@2
rec. loss	BCE
β	4.0
epochs	2500
batch size	1024
optimizer	Adam
LR	0.001

B.2.5 Hyperparameter Dependencies Training Setup

Table B.7: M^{2}VAE architecture and training setup for Fig. 7.2.

data set	divided MNIST (2), vertical split
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	see Fig. 7.2
D_{a}	392
D_{b}	392
$f_{\text{enc.}_{\text{a}}}$	input@392 → ReLU@500
$f_{\text{enc.}_{\text{b}}}$	input@392 → ReLU@500
$f_{\text{enc.}_{\text{ab}}}$	None
$f_{\text{dec.}_{\text{a}}}$	input@<see Fig. 7.2> → ReLU@500 → sig@392
$f_{\text{dec.}_{\text{b}}}$	input@<see Fig. 7.2> → ReLU@500 → sig@392
$f_{\boldsymbol{\mu}_{\text{a}}}$	input@500 → lin@<see Fig. 7.2>
$f_{\boldsymbol{\sigma}_{\text{a}}}$	input@500 → lin@<see Fig. 7.2>
$f_{\boldsymbol{\mu}_{\text{b}}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\sigma}_{\text{b}}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\mu}_{\text{ab}}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\sigma}_{\text{ab}}}$	input@500 → lin@<see Fig. B.2>
rec. loss	BCE
β	1.0
epochs	20
batch size	1024
optimizer	Adam
LR	see Fig. 7.2

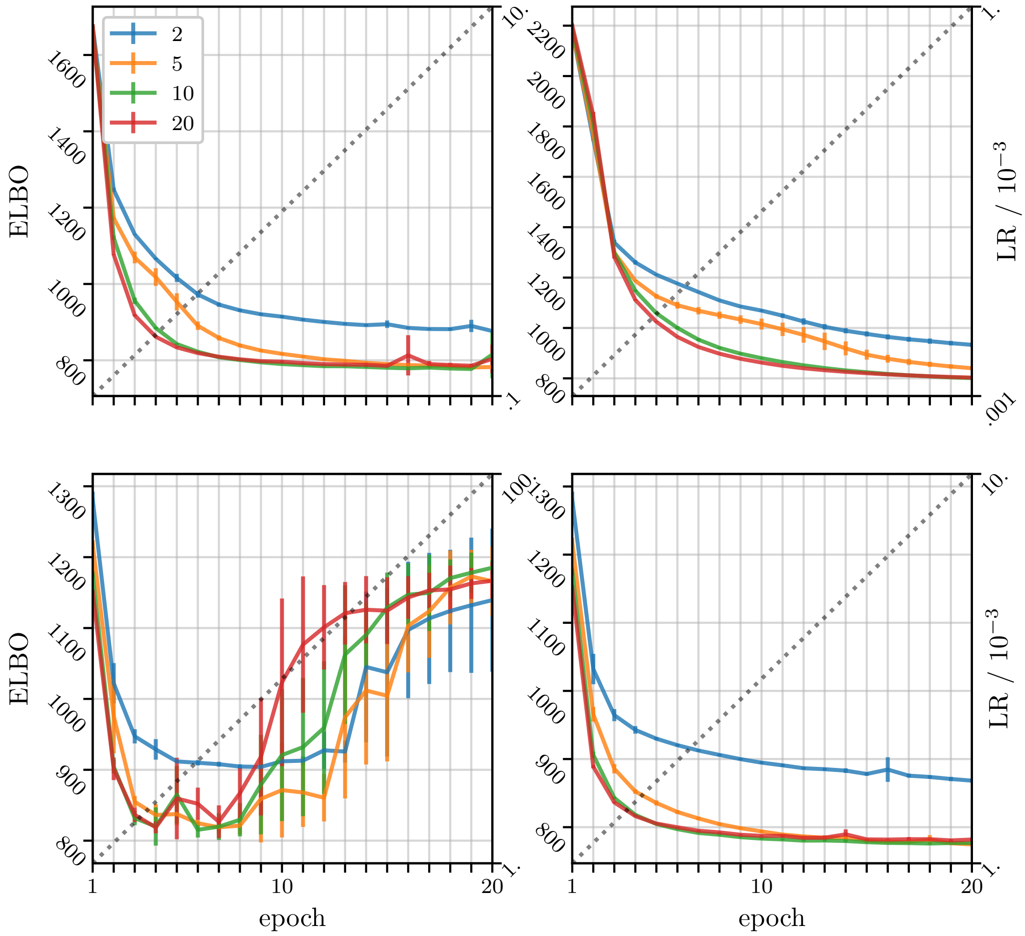


Figure B.2: Learning rate (LR) evaluation of the bi-modal M²VAE trained on the eMNIST data set for various $D_{\mathrm{z}} = \{2, 5, 10, 20\}$. See Table B.8 for the M²VAE setup.

Table B.8: M²VAE architecture and training setup for Fig. B.2.

data set	eMNIST
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	see Fig. B.2
D_{a}	784
D_{b}	784
$f_{\mathrm{enc{\cdot}a}}$	input@784 → ReLU@500
$f_{\mathrm{enc{\cdot}b}}$	input@784 → ReLU@500
$f_{\mathrm{enc{\cdot}ab}}$	None
$f_{\mathrm{dec{\cdot}a}}$	input@<see Fig. B.2> → ReLU@500 → sig@784
$f_{\mathrm{dec{\cdot}b}}$	input@<see Fig. B.2> → ReLU@500 → sig@784

$f_{\boldsymbol{\mu}_\mathrm{a}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\sigma}_\mathrm{a}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\mu}_\mathrm{b}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\sigma}_\mathrm{b}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\mu}_\mathrm{ab}}$	input@500 → lin@<see Fig. B.2>
$f_{\boldsymbol{\sigma}_\mathrm{ab}}$	input@500 → lin@<see Fig. B.2>
rec. loss	BCE
β	1.0
epochs	20
batch size	1024
optimizer	Adam
LR	see Fig. B.2

Table B.9: M²VAE architecture and training setup for Fig. 7.4 and 7.3.

data set	divided MNIST (2)
input norm.	min. 0 / max. 1
input format	flattened
D_z	$(2, 5, 10, 15, 20, 40, 80)$
$D_{\mathrm{enc.}_\mathrm{ab}}$	$(0 := \mathrm{None}, 64, 128, 256)$
D_a	392
D_b	392
$f_{\mathrm{enc.}_\mathrm{a}}$	input@392 → ReLU@500
$f_{\mathrm{enc.}_\mathrm{b}}$	input@392 → ReLU@500
$f_{\mathrm{enc.}_\mathrm{ab}}$	input@500 → ReLU@$<D_{\mathrm{enc.}_\mathrm{ab}}>$
$f_{\mathrm{dec.}_\mathrm{a}}$	input@$<D_\mathrm{z}>$ → ReLU@500 → sig@392
$f_{\mathrm{dec.}_\mathrm{b}}$	input@$<D_\mathrm{z}>$ → ReLU@500 → sig@392
$f_{\boldsymbol{\mu}_\mathrm{a}}$	input@500 → lin@$<D_\mathrm{z}>$
$f_{\boldsymbol{\sigma}_\mathrm{a}}$	input@500 → lin@$<D_\mathrm{z}>$
$f_{\boldsymbol{\mu}_\mathrm{b}}$	input@500 → lin@$<D_\mathrm{z}>$
$f_{\boldsymbol{\sigma}_\mathrm{b}}$	input@500 → lin@$<D_\mathrm{z}>$
$f_{\boldsymbol{\mu}_\mathrm{ab}}$	input@500 → lin@$<D_\mathrm{z}>$
$f_{\boldsymbol{\sigma}_\mathrm{ab}}$	input@500 → lin@$<D_\mathrm{z}>$
rec. loss	BCE
β_norm	1.
β_M	$(0.001, 0.01, 0.1, 1.0, 10, 20, 30)$
epochs	5000
batch size	512
optimizer	Adam
LR	0.001

B.2.6 XOR Training Setup

Table B.10: M²VAE architecture and training setup for Section 7.2.2.1.

data set	XOR

input norm.	None
input format	flattened
D_z	1
D_a	2
D_b	1
$f_{\text{enc-a}}$	input@2 → ReLU@4
$f_{\text{enc-b}}$	input@1 → ReLU@4
$f_{\text{enc-ab}}$	None
$f_{\text{dec-a}}$	input@1 → ReLU@4 → sig@2
$f_{\text{dec-b}}$	input@1 → ReLU@4 → sig@1
$f_{\boldsymbol{\mu}_\text{a}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{a}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\text{b}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{b}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\text{ab}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{ab}}$	input@4 → lin@1
rec. loss	BCE
β_norm	0.001
β_M	0.001
epochs	10,000
batch size	32
optimizer	RMSprop
LR	0.001

Table B.11: JMMVAE architecture and training setup for Section 7.2.2.1.

data set	XOR
input norm.	None
input format	flattened
D_z	1
D_a	2
D_b	1
$f_{\text{enc-a}}$	input@2 → ReLU@4
$f_{\text{enc-b}}$	input@1 → ReLU@4
$f_{\text{enc-ab}}$	None
$f_{\text{dec-a}}$	input@1 → ReLU@4 → sig@2
$f_{\text{dec-b}}$	input@1 → ReLU@4 → sig@1
$f_{\boldsymbol{\mu}_\text{a}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{a}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\text{b}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{b}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\text{ab}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\text{ab}}$	input@4 → lin@1
rec. loss	BCE
β_norm	0.001
α (i.e., β_M)	0.001
epochs	10.000
batch size	32
optimizer	RMSprop

LR	0.001

Table B.12: JVAE architecture and training setup for Section 7.2.2.1.

data set	XOR
input norm.	None
input format	concat. & flattened w/ x=.5
D_z	1
D_ab	3
$f_\mathrm{enc.}$	input@3 → ReLU@8
$f_\mathrm{dec.}$	input@1 → ReLU@8 → sig@3
$f_{\boldsymbol{\mu}_\mathrm{a}}$	input@8 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{a}}$	input@8 → lin@1
$f_{\boldsymbol{\mu}_\mathrm{b}}$	input@8 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{b}}$	input@8 → lin@1
$f_{\boldsymbol{\mu}_\mathrm{ab}}$	input@8 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{ab}}$	input@8 → lin@1
rec. loss	BCE
β_norm	0.001
epochs	10.000
batch size	32
optimizer	RMSprop
LR	0.001

Table B.13: pPCA architecture and training setup for Section 7.2.2.1.

data set	XOR
input norm.	None
input format	flattened
D_z	1
D_a	2
D_b	1
$f_{\mathrm{enc.}_\mathrm{a}}$	input@2 → lin@4
$f_{\mathrm{enc.}_\mathrm{b}}$	input@1 → lin@4
$f_{\mathrm{enc.}_\mathrm{ab}}$	None
$f_{\mathrm{dec.}_\mathrm{a}}$	input@1 → lin@4 → sig@2
$f_{\mathrm{dec.}_\mathrm{b}}$	input@1 → lin@4 → sig@1
$f_{\boldsymbol{\mu}_\mathrm{a}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{a}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\mathrm{b}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{b}}$	input@4 → lin@1
$f_{\boldsymbol{\mu}_\mathrm{ab}}$	input@4 → lin@1
$f_{\boldsymbol{\sigma}_\mathrm{ab}}$	input@4 → lin@1
rec. loss	BCE
β_norm	0.001
epochs	10.000
batch size	32

optimizer	RMSprop
LR	0.001

B.2.7 MoG Training Setup

Table B.14: M²VAE architecture and training setup for Section 7.2.2.2. The other VAEs JMMVAE and JVAE are configured accordingly.

data set	MoG
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	2
D_{a}	2
D_{b}	2
$f_{\mathrm{enc}\text{-}\mathrm{a}}$	input@2 → ReLU@64
$f_{\mathrm{enc}\text{-}\mathrm{b}}$	input@2 → ReLU@64
$f_{\mathrm{enc}\text{-}\mathrm{ab}}$	None
$f_{\mathrm{dec}\text{-}\mathrm{a}}$	input@2 → ReLU@64 → lin@2
$f_{\mathrm{dec}\text{-}\mathrm{b}}$	input@2 → ReLU@64 → lin@2
$f_{\boldsymbol{\mu}_{\mathrm{a}}}$	input@64 → lin@2
$f_{\boldsymbol{\sigma}_{\mathrm{a}}}$	input@64 → lin@2
$f_{\boldsymbol{\mu}_{\mathrm{b}}}$	input@64 → lin@2
$f_{\boldsymbol{\sigma}_{\mathrm{b}}}$	input@64 → lin@2
$f_{\boldsymbol{\mu}_{\mathrm{ab}}}$	input@64 → lin@2
$f_{\boldsymbol{\sigma}_{\mathrm{ab}}}$	input@64 → lin@2
rec. loss	MSE
β_{norm}	0.01
β_{M}	0.01
epochs	400
batch size	128
optimizer	RMSprop
LR	0.001

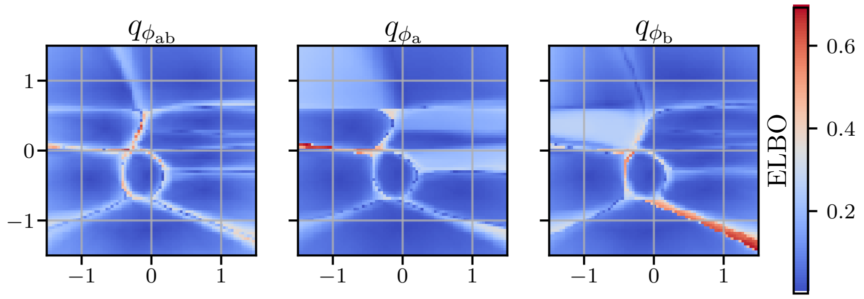


Figure B.3: Plain reconstruction loss w/o trajectories (c.f. Fig. 7.8).

B.2.8 Shared Weights Training Setup

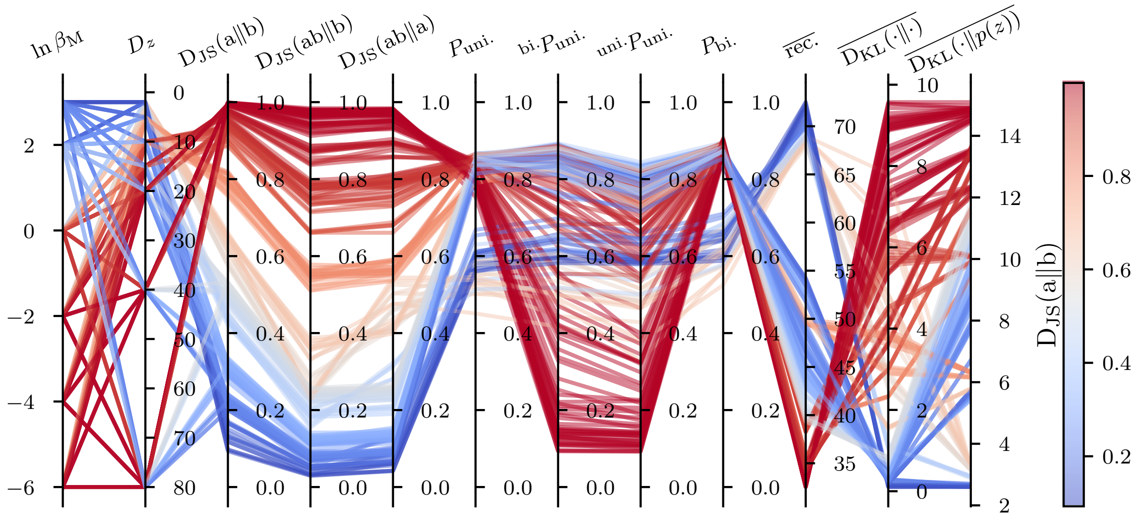


Figure B.4: Parallel coordinates for non-shared weights in Section 7.2.2.3 and analog to Fig. 7.4.

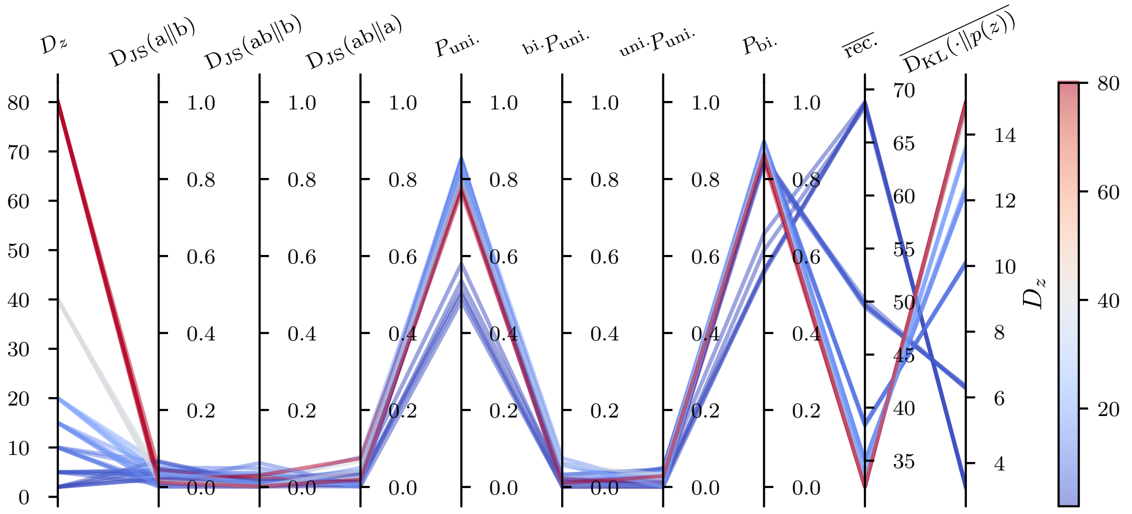


Figure B.5: Parallel coordinates for non-shared weights from Section 7.2.2.3 and analog to Fig. 7.4 with $\beta_{\mathrm{M}} = .0$.

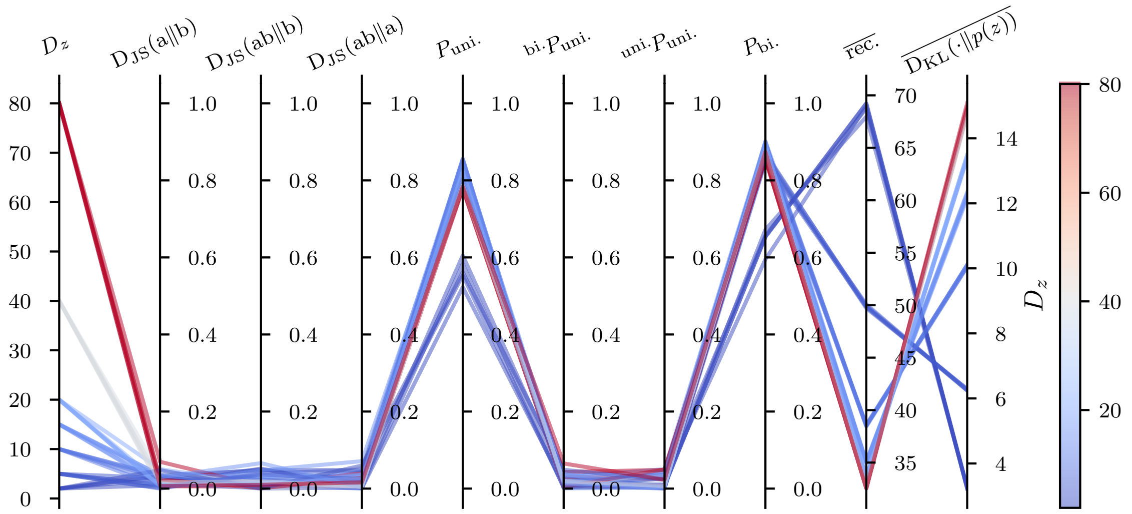


Figure B.6: Parallel coordinates for shared weights from Section 7.2.2.3 and analog to Fig. 7.4 with $\beta_{\mathrm{M}} = .0$.

B.2.9 Competitive Evaluation Setup

Table B.15: M²VAE architecture and training setup for Section 7.2.3 on eMNIST.

data set	eMNIST
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	64
D_{a}	784
D_{b}	784
$f_{\mathrm{enc}\cdot\mathrm{a}}$	input@392 → ReLU@500
$f_{\mathrm{enc}\cdot\mathrm{b}}$	input@392 → ReLU@500
$f_{\mathrm{enc}\cdot\mathrm{ab}}$	None
$f_{\mathrm{dec}\cdot\mathrm{a}}$	input@64 → ReLU@500 → lin@784
$f_{\mathrm{dec}\cdot\mathrm{b}}$	input@64 → ReLU@500 → lin@784
$f_{\boldsymbol{\mu}_{\mathrm{a}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{a}}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_{\mathrm{b}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{b}}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_{\mathrm{ab}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{ab}}}$	input@500 → lin@64
rec. loss	BCE
β_{norm}	0.01
β_{M}	10.0
epochs	50,000
batch size	512

optimizer	Adam
LR	0.001

Table B.16: M²VAE architecture and training setup for Section 7.2.3 on divided MNIST (2).

data set	divided MNIST (2), vertical split
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	64
D_{a}	392
D_{b}	392
$f_{\mathrm{enc.}_{\mathrm{a}}}$	input@392 → ReLU@500
$f_{\mathrm{enc.}_{\mathrm{b}}}$	input@392 → ReLU@500
$f_{\mathrm{enc.}_{\mathrm{ab}}}$	None
$f_{\mathrm{dec.}_{\mathrm{a}}}$	input@64 → ReLU@500 → lin@392
$f_{\mathrm{dec.}_{\mathrm{b}}}$	input@64 → ReLU@500 → lin@392
$f_{\boldsymbol{\mu}_{\mathrm{a}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{a}}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_{\mathrm{b}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{b}}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_{\mathrm{ab}}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{ab}}}$	input@500 → lin@64
rec. loss	BCE
β_{norm}	0.01
β_{M}	10.0
epochs	50,000
batch size	512
optimizer	Adam
LR	0.001

Table B.17: M²VAE architecture and training setup for Section 7.2.3 on divided MNIST (4).

data set	divided MNIST (4) vertical and horizontal split
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	64
D_{a}	196
D_{b}	196
D_{c}	196
D_{c}	196
$f_{\mathrm{enc.}_{\mathrm{a}}}$	input@196 → ReLU@500
$f_{\mathrm{enc.}_{\mathrm{b}}}$	input@196 → ReLU@500
$f_{\mathrm{enc.}_{\mathrm{c}}}$	input@196 → ReLU@500
$f_{\mathrm{enc.}_{\mathrm{d}}}$	input@196 → ReLU@500
$f_{\mathrm{enc.}_{\lvert m\rvert>1}}$	None

$f_{\text{dec}._\text{a}}$	input@64 → ReLU@500 → lin@196
$f_{\text{dec}._\text{b}}$	input@64 → ReLU@500 → lin@196
$f_{\text{dec}._\text{c}}$	input@64 → ReLU@500 → lin@196
$f_{\text{dec}._\text{d}}$	input@64 → ReLU@500 → lin@196
$f_{\boldsymbol{\mu}_\text{a}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{a}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_\text{b}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{b}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_\text{c}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{c}}$	input@500 → lin@64
⋮	⋮
$f_{\boldsymbol{\mu}_\text{cd}}$	input@1000 → lin@64
$f_{\boldsymbol{\sigma}_\text{cd}}$	input@1000 → lin@64
⋮	⋮
$f_{\boldsymbol{\mu}_\text{abcd}}$	input@2000 → lin@64
$f_{\boldsymbol{\sigma}_\text{abcd}}$	input@2000 → lin@64
rec. loss	BCE
β_norm	0.01
β_M	10.0
epochs	50,000
batch size	512
optimizer	Adam
LR	0.001

Table B.18: M^2VAE architecture and training setup for Section 7.2.3 on MNIST+SVHN.

data set	MNIST+SVHN
input norm.	min. 0 / max. 1 and RGB to gray conversion for SVHN
input format	flattened
D_z	64
D_a	784
D_b	1025
$f_{\text{enc}._\text{a}}$	input@392 → ReLU@500
$f_{\text{enc}._\text{b}}$	input@392 → ReLU@500
$f_{\text{enc}._\text{ab}}$	None
$f_{\text{dec}._\text{a}}$	input@64 → ReLU@500 → lin@784
$f_{\text{dec}._\text{b}}$	input@64 → ReLU@500 → lin@1025
$f_{\boldsymbol{\mu}_\text{a}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{a}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_\text{b}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{b}}$	input@500 → lin@64
$f_{\boldsymbol{\mu}_\text{ab}}$	input@500 → lin@64
$f_{\boldsymbol{\sigma}_\text{ab}}$	input@500 → lin@64
rec. loss	BCE
β_norm	0.01
β_M	10.0

epochs	50,000
batch size	512
optimizer	Adam
LR	0.001

B.2.10 Rubiks Data Set Evaluation Setup

Table B.19: ICM architecture and training setup for Section 8.3.1 on Rubiks.

data set	Rubiks
input norm.	min. 0 / max. 1 for all channels in RGB
input format	flattened
D_z	2 (i.e., feature)
D_s	$30 \cdot 40 \cdot 3 = 3600$ (i.e., image)
D_a	3 (i.e., action)
$f_\text{enc.}$	input@3600 → ELU@32 → Dropout@0.4 → ELU@16 → Dropout@0.4 → lin@2 → BatchNorm
f_inverse	concat. (input@2, input@2) → ELU@16 → Dropout@0.4 → ELU@16 → Dropout@0.4 → lin@3
f_forward	concat. (input@2, input@3) → ELU@16 → Dropout@0.4 → ELU@16 → Dropout@0.4 → lin@2
β	0.1
inverse rec. loss	$(1. - \beta)$ MSE(a_t, a'_t)
forward rec. loss	β MSE(z_{t+1}, z'_{t+1})
epochs	1000
batch size	128
optimizer	Adam
LR	0.0001

Table B.20: M^{2}VAE architecture and training setup for Section 8.3.1 on Rubiks.

data set	Rubiks
input norm.	min. 0 / max. 1 for all channels in RGB
input format	flattened
D_z	2 (i.e., feature)
D_a	$30 \cdot 40 \cdot 3 + 3 = 3603$ (i.e., image + action)
D_b	$30 \cdot 40 \cdot 3 = 3600$ (i.e., image)
$f_{\text{enc.}_\text{a}}$	concat. (input@3600, input@3) → ReLU@256 → ReLU@128
$f_{\text{enc.}_\text{a}}$	concat. input@3600 → ReLU@256 → ReLU@128
$f_{\text{enc.}_\text{ab}}$	None
$f_{\text{dec.}_\text{a}}$	input@128 → ReLU@256 → lin@3603
$f_{\text{dec.}_\text{b}}$	input@128 → ReLU@256 → lin@3600
$f_{\boldsymbol{\mu}_\text{a}}$	input@128 → lin@64
$f_{\boldsymbol{\sigma}_\text{a}}$	input@128 → lin@64
$f_{\boldsymbol{\mu}_\text{b}}$	input@128 → lin@64
$f_{\boldsymbol{\sigma}_\text{b}}$	input@128 → lin@64

$f_{\boldsymbol{\mu}_{\mathrm{ab}}}$	input@128 → lin@64
$f_{\boldsymbol{\sigma}_{\mathrm{ab}}}$	input@128 → lin@64
rec. loss	MSE
β	1.0
β_{M}	1.0
epochs	1000
batch size	128
optimizer	Adam
LR	0.001

B.2.11 AMiRo-CITrack Evaluation Setup

Table B.21: VAE architecture for downs-sampling the camera data.

class vs. modality	*a*	*b*	*c*
green, mat, cyl. (1)	✓		
red, mat, cube (2)		✓	
red, mat, cyl. (3)	✓	✓	✓
red, shiny, cyl. (4)			✓

Table B.22: VAE architecture for downs-sampling the camera data.

data set	Camera+LiDAR with Camera only for bootstrap
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	64
D_{x}	307,200
$f_{\mathrm{enc.}}$	input@307200 → ReLU@256 → ReLU@128
$f_{\mathrm{dec.}}$	input@64 → ReLU@128 → ReLU@256 → sig@307200
$f_{\boldsymbol{\mu}}$	input@128 → lin@64
$f_{\boldsymbol{\sigma}}$	input@128 → lin@64
rec. loss	BCE
β	1.0
epochs	50000
batch size	128
optimizer	Adam
LR	0.001

Table B.23: VAE architecture for downs-sampling the LiDAR data.

data set	Camera+LiDAR with LiDAR only for bootstrap
input norm.	min. 0 / max. 1
input format	flattened
D_{z}	2
D_{x}	320 (frontal lobe of the LiDAR, i.e., ⅓ of whole scan)

$f_{\text{enc.}}$	input@320 → ReLU@256 → ReLU@128
$f_{\text{dec.}}$	input@64 → ReLU@128 → ReLU@256 → sig@320
$f_{\boldsymbol{\mu}}$	input@128 → lin@64
$f_{\boldsymbol{\sigma}}$	input@128 → lin@64
rec. loss	BCE
β	1.0
epochs	50000
batch size	128
optimizer	Adam
LR	0.001

Table B.24: Tri-modal M²VAE architecture and training setup for Section 8.3.2.

data set	Camera+LiDAR+proximity
input norm.	None
input format	flattened
D_{z}	2
D_{a}	64 (down-sampled camera)
D_{b}	64 (down-sampled LiDAR)
D_{c}	2 (two frontal proximity sensors)
$f_{\text{enc.}_\text{a}}$	input@64 → ReLU@32 → ReLU@16
$f_{\text{enc.}_\text{b}}$	input@64 → ReLU@32 → ReLU@16
$f_{\text{enc.}_\text{b}}$	input@2 → ReLU@8 → ReLU@4
$f_{\text{enc.}_{\lvert m\rvert>1}}$	None
$f_{\text{dec.}_\text{a}}$	input@2 → ReLU@16 → ReLU@32 → lin@64
$f_{\text{dec.}_\text{a}}$	input@2 → ReLU@16 → ReLU@32 → lin@64
$f_{\text{dec.}_\text{a}}$	input@2 → ReLU@4 → ReLU@8 → lin@2
$f_{\boldsymbol{\mu}_\text{a}}$	input@16 → lin@2
$f_{\boldsymbol{\sigma}_\text{a}}$	input@16 → lin@2
$f_{\boldsymbol{\mu}_\text{b}}$	input@16 → lin@2
$f_{\boldsymbol{\sigma}_\text{b}}$	input@16 → lin@2
$f_{\boldsymbol{\mu}_\text{c}}$	input@4 → lin@2
$f_{\boldsymbol{\sigma}_\text{c}}$	input@4 → lin@2
$\vdots$	$\vdots$
$f_{\boldsymbol{\mu}_\text{bc}}$	input@20 → lin@2
$f_{\boldsymbol{\sigma}_\text{bc}}$	input@20 → lin@2
$\vdots$	$\vdots$
$f_{\boldsymbol{\mu}_\text{abc}}$	input@36 → lin@2
$f_{\boldsymbol{\sigma}_\text{abc}}$	input@36 → lin@2
rec. loss	MSE
β_{norm}	0.01
β_{M}	10.0
epochs	50,000
batch size	512
optimizer	Adam
LR	0.001

Table B.25: DQN architecture according to Mnih et al. [Mni+13].

input norm.	min. 0 / max. 1
input format	flattened
γ	.95
ϵ_{start}	1.
$\epsilon_{\text{min.}}$	.01
ϵ_{decay}	.995
f_{shared}	input@21 → ReLU@24 → ReLU@24
f_{head_a}	input@24 → ReLU@24 → lin@5
f_{head_b}	input@24 → ReLU@24 → lin@5
f_{head_c}	input@24 → ReLU@24 → lin@5
rec. loss	see [Mni+13]
epochs	2000
batch size	128
optimizer	Adam
LR	0.001

B.3 Latent Space Statistics

Table B.26: Further statistics about the perturbation z_{diff} wrt. MNIST test and training data set.

	minimum	maximum	mean	median
	MNIST test set			
common	.0002165	.715	.0974	.06512
proposed	.0000866	.397	.0117	.00902
	MNIST training set			
common	.0002615	1.1810	.1076	.07208
proposed	.0000538	0.0942	.0102	.00847

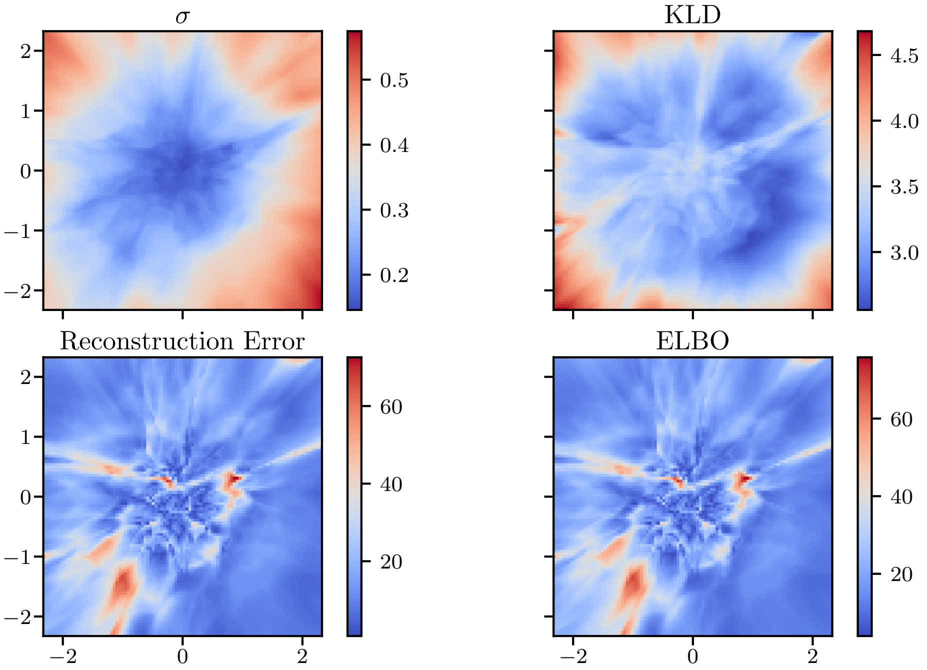


Figure B.7: Quantitative latent space statistics for the common VAE approach

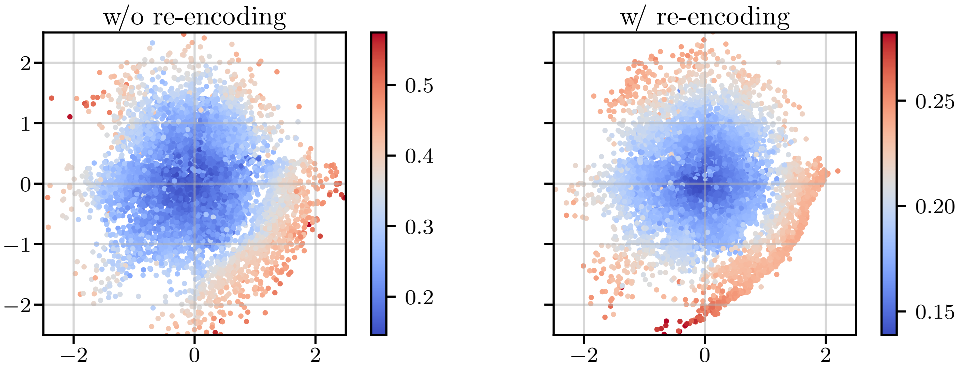


Figure B.8: Encoding of the MNIST test data set ($\mathcal{O}_{\text{test}}$) using the encoder of the commonly trained VAE (left) vs. the proposed approach (right). Both plots show $f_{\boldsymbol{\mu}}(f_{\text{enc.}}(\mathcal{O}_{\text{test}}))$ with standard deviation coloring (i.e., $\sigma^2 = \sum_{D_z} \exp f_{\boldsymbol{\sigma}}(f_{\text{enc.}}(\mathcal{O}_{\text{test}}))$). Coloring is normalized to match the σ-plots in Fig. B.7 and 5.16 respectively.

B.4 Semi-Supervised VAE

In this section, the approach of the generative semi-supervised model M2, proposed by Kingma et al. [Kin+14b], is described. Similar to the CVAE, two observables $\mathbf{x}$ and $\mathbf{y}$ are considered. The only difference in the semi-supervised ELBO is the additional $p(\mathbf{y})$ (compare Eq. (3.38) and Eq. (B.1)).

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \phi, \theta) = \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log\left(\frac{p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})p(\mathbf{z})p(\mathbf{y})}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})}\right), \tag{B.1}$$

However, in the case of K-class classification problem, $p(\mathbf{y})$ becomes a categorical distribution, which is often a constant (i.e., $p(\mathbf{y}) = 1/K$) Thus, the ELBO becomes the same as for the CVAE, so that the lower bounds can be optimized in the same way.

Although Eq. (B.1) is learned in the labeled set $\mathcal{O}_\mathrm{L}$, in the framework of semi-supervised learning, the unlabeled set $\mathcal{O}_\mathrm{U}$ is also used for learning. Therefore, the ELBO of the marginal distribution $p_\theta(\mathbf{x})$ without label information is obtained, which becomes one part of the objective function. To this objective function, the discriminative model $q_\phi(\mathbf{y}|\mathbf{x})$ is introduced:

$$\log p_\theta(\mathbf{x}) \geq \mathrm{E}_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})} \log\left(\frac{p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})p(\mathbf{z})p(\mathbf{y})}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})}\right) := U(\mathbf{x}). \tag{B.2}$$

with $q_\phi(\mathbf{z}, \mathbf{y}|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})q_\phi(\mathbf{y}|\mathbf{x})$.

In order to learn the discriminative model on the labeled set, the log likelihood of the discriminative model in the labeled set is added to Eq. (B.2) to obtain the other part of the objective function as follows:

$$L(\mathbf{x}, \mathbf{y}) = \mathcal{L}(\mathbf{x}, \mathbf{y}; \phi, \theta) + \alpha \log q_\phi(\mathbf{y}|\mathbf{x}), \tag{B.3}$$

where α is a parameter that controls the ratio of the discriminative model to the generative model.

Therefore, the final objective function J in both the labeled and unlabeled unifies to

$$J = \frac{1}{O} \sum_{(\mathbf{x}_o,\mathbf{y}_o)\in\mathcal{O}_\mathrm{L}} L(\mathbf{x}_o, \mathbf{y}_o) + \frac{1}{O'} \sum_{\mathbf{x}_{o'}\in\mathcal{O}_\mathrm{U}} U(\mathbf{x}_{o'}). \tag{B.4}$$

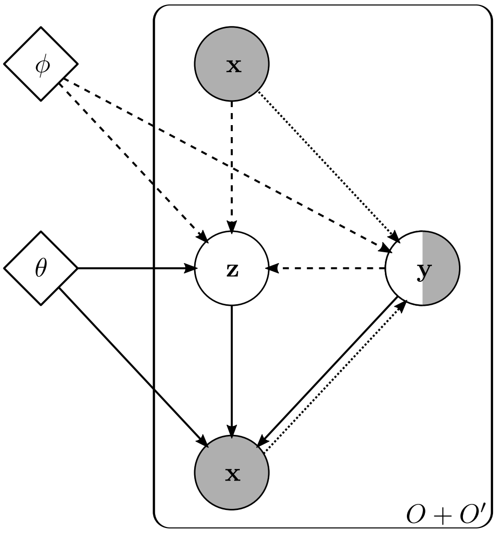


Figure B.9: GM of the M2 model by Kingma et al. [Kin+14b]. Note that the label **y** is half black and half white, in order to consider both labeled and unlabeled variables (i.e., both observed and latent variables). The dotted line represents the discriminative model.

B.5 List of Data Set Websites

Table B.27: Non-exhaustive list of data set hosting and searching websites.

Radish	`http://radish.sourceforge.net/`
MRPT	`https://www.mrpt.org/robotics_datasets`
IJRR	`https://journals.sagepub.com/topic/collections-ijr/ijr-3-datapapers/ijr`
CVonline	`http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm`
CVPapers	`http://www.cvpapers.com/datasets.html`
YACVID	`http://riemenschneider.hayko.at/vision/dataset/`
Computer Vision Online	`https://computervisiononline.com/datasets`
Kaggle	`https://www.kaggle.com/datasets`
UCI	`https://archive.ics.uci.edu/ml/datasets.php`
CMU	`https://guides.library.cmu.edu/machine-learning/datasets`
VisualData	`https://www.visualdata.io/`
Google's Dataset Search	`https://datasetsearch.research.google.com/`

B.6 Alternative nomenclature for the Variational Autoencoder

This section is a one-to-one derivation of the VAE, but with an alternative nomenclature that suits the nomenclature of Chapter 5. First, the derivation of the vanilla *Variational Autoencoder* by [Kin+13] is recaped.

B.6.1 The Variational Bound

$$\begin{aligned}
\mathrm{L} &= \log(p(a)) && && \text{(B.5)}\\
&= \sum_z q(z|a)\log(p(a)) && \text{(C.6) w/o cond.} && \text{(B.6)}\\
&= \sum_z q(z|a)\log\left(\frac{p(z,a)}{p(z|a)}\right) && \text{Eq. (C.1)} && \text{(B.7)}\\
&= \sum_z q(z|a)\log\left(\frac{p(z,a)}{p(z|a)}\frac{q(z|a)}{q(z|a)}\right) && \text{mul. by 1} && \text{(B.8)}\\
&= \sum_z q(z|a)\log\left(\frac{p(z,a)}{q(z|a)}\frac{q(z|a)}{p(z|a)}\right) && \text{reo.} && \text{(B.9)}\\
&= \sum_z q(z|a)\log\left(\frac{p(z,a)}{q(z|a)}\right) + \sum_z q(z|a)\log\left(\frac{q(z|a)}{p(z|a)}\right) && \text{Eq. (C.3)} && \text{(B.10)}\\
&= \mathcal{L} + \mathrm{D_{KL}}(q(z|a)\|p(z|a)) && \text{C.2 \& (C.5)} && \text{(B.11)}\\
&\geq \mathcal{L} && \mathrm{D_{KL}} \geq 0 && \text{(B.12)}
\end{aligned}$$

B.6.2 Approximate Inference (i.e. rewriting $\mathcal{L}$)

$$\begin{aligned}
\mathcal{L} &= \sum_z q(z|a)\log\left(\frac{p(z,a)}{q(z|a)}\right) && && \text{(B.13)}\\
&= \sum_z q(z|a)\log\left(\frac{p(a|z)p(z)}{q(z|a)}\right) && \text{Eq. (C.1)} && \text{(B.14)}\\
&= \sum_z q(z|a)\log\left(\frac{p(z)}{q(z|a)}\right) + \sum_z q(z|a)\log(p(a|z)) && \text{Eq. (C.3)} && \text{(B.15)}\\
&= -\,\mathrm{D_{KL}}(q(z|a)\|p(z)) + \mathrm{E}_{q(z|a)}\log(p(a|z)) && \text{C.2} && \text{(B.16)}
\end{aligned}$$

If the variable a is replaced by some real valued sample $a^{(i)}$ (e.g., image or LiDAR scan), two terms can be identified:

$$\mathcal{L} = \underbrace{-\mathrm{D}_{\mathrm{KL}}\Big(q_\phi\Big(z|a^{(i)}\Big)\|p(z)\Big)}_{\text{Regularization}} + \underbrace{\mathrm{E}_{q_\phi\left(z|a^{(i)}\right)}\log\Big(p_\theta\Big(a^{(i)}|z\Big)\Big)}_{\text{Reconstruction}} \tag{B.17}$$

B.7 Derivation for the Joint Multi-Modal VAE via Variation of Information

$$\begin{aligned}
\mathcal{L}_{\mathrm{M_a}}+\mathcal{L}_{\mathrm{M_b}} &= \sum_z q(z|a,b)\log\left(\frac{p(z,a|b)}{q(z|a,b)}\right) && && \text{(B.18)}\\
&\quad+\sum_z q(z|a,b)\log\left(\frac{p(z,b|a)}{q(z|a,b)}\right) && && \text{(B.19)}\\
&= \sum_z q(z|a,b)\log\left(\frac{p(a|z)p(z|b)}{q(z|a,b)}\right) && && \text{(B.20)}\\
&\quad+\sum_z q(z|a,b)\log\left(\frac{p(b|z)p(z|a)}{q(z|a,b)}\right) && \text{(5.24)} && \text{(B.21)}\\
&= \sum_z q(z|a,b)\log\left(\frac{p(a|z)}{q(z|a,b)}\right) && && \text{(B.22)}\\
&\quad+\sum_z q(z|a,b)\log\left(\frac{p(z|b)}{q(z|a,b)}\right) && && \text{(B.23)}\\
&\quad+\sum_z q(z|a,b)\log\left(\frac{p(b|z)}{q(z|a,b)}\right) && && \text{(B.24)}\\
&\quad+\sum_z q(z|a,b)\log\left(\frac{p(z|a)}{q(z|a,b)}\right) && \text{reo.} && \text{(B.25)}\\
&= \mathrm{E}_{q(z|a,b)}\log(p(a|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|b)) && && \text{(B.26)}\\
&\quad+\mathrm{E}_{q(z|a,b)}\log(p(b|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a)) && \text{C.2} && \text{(B.27)}\\
&= \mathrm{E}_{q(z|a,b)}\log(p(a|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|b)) && && \text{(B.28)}\\
&\quad+\mathrm{E}_{q(z|a,b)}\log(p(b|z)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a)) && && \text{(B.29)}\\
&\quad+\mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a,b)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a,b)) && \text{add 0} && \text{(B.30)}\\
&= \mathcal{L}_{\mathrm{J}} - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|b)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a)) && && \text{(B.31)}\\
&\quad+\mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a,b)) && \text{(5.16)} && \text{(B.32)}\\
&\geq \mathcal{L}_{\mathrm{J}} - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|b)) - \mathrm{D}_{\mathrm{KL}}(q(z|a,b)\|p(z|a)) && && \text{(B.33)}\\
&=: \mathcal{L}_{\mathrm{M}} && && \text{(B.34)}
\end{aligned}$$

B.8 Derivation of KLD-Derivative Inequality

$$\nabla_\theta \frac{1}{K}\sum_k^K \overset{?}{=} \nabla_\theta \,\mathrm{D}_{\mathrm{KL}}(p\|p_{\mathrm{M}}) \tag{B.35}$$

$$\nabla_\theta \left(-\mathrm{H}[p_\theta] - \frac{1}{K}\sum_k^K \int p \log p_{\mathrm{M}_k} \right) \overset{?}{=} \nabla_\theta \left(-\mathrm{H}[p_\theta] - \int p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \right) \tag{B.36}$$

$$\nabla_\theta \frac{1}{K}\sum_k^K \int p \log p_{\mathrm{M}_k} \overset{?}{=} \nabla_\theta \int p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{B.37}$$

$$\frac{1}{K}\sum_k^K \int \nabla_\theta p \log p_{\mathrm{M}_k} \neq \int \nabla_\theta p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{B.38}$$

$$\nabla_\theta \,\mathrm{D}_{\mathrm{KL}}(p\|p_{\mathrm{M}}) = \nabla_\theta - \mathrm{H}[p_\theta] - \int p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{B.39}$$

$$\dots \tag{B.40}$$

$$= \nabla_\theta \int p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{B.41}$$

$$= \int \nabla_\theta p_\theta \log \sum_k^K \lambda_k p_{\mathrm{M}_k} \tag{B.42}$$

B.9 Variation of Information

- $\mathrm{VI}(A, B)$: variation of information (VoI) between some properties A and B
- $\mathrm{I}(A)$: information of A
- $\mathrm{I}(A, B)$: mutual information (MI) of A and B
- $\mathrm{I}(A, B|C)$: mutual conditional information (MCI) of A and B given C
- $\mathrm{H}(A)$: entropy of A
- $\mathrm{H}(A, B)$: joint entropy (JE) of A and B
- $\mathrm{H}(A|B)$: conditional entropy (CE) of A given B

The variation of information (VoI) between some random variables can be written as

$$\mathrm{VI}(A, B) = \mathrm{H}(A) + \mathrm{H}(B) - 2\,\mathrm{I}(A, B) = \mathrm{H}(A|B) + \mathrm{H}(B|A), \tag{B.43}$$

$$\mathrm{VI}(A, B, C) = \mathrm{H}(A) + \mathrm{H}(B) + \mathrm{H}(C) - 3\,\mathrm{I}(A, B) \tag{B.44}$$

$$= \mathrm{H}(A|B,C) + \mathrm{H}(B|A,C) + \mathrm{H}(C|A,B), \tag{B.45}$$

$$\mathrm{VI}(A,B,C,\ldots) = \ldots. \tag{B.46}$$

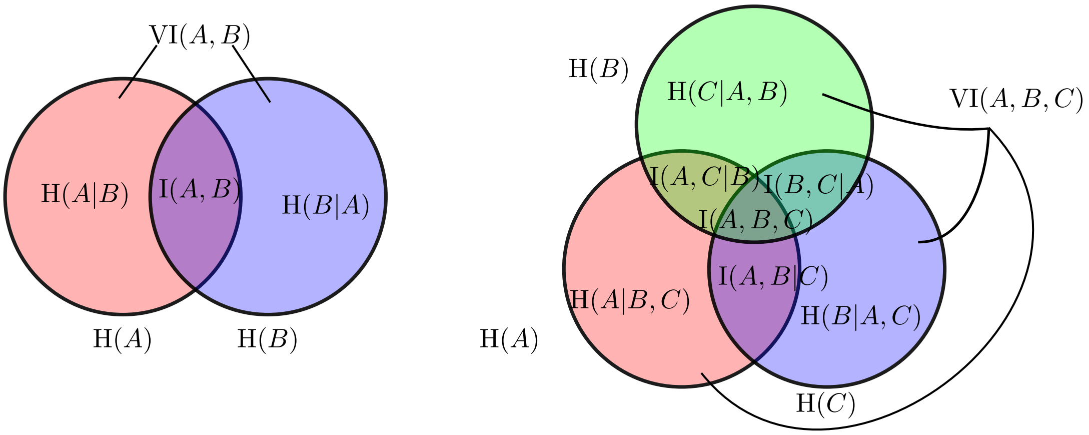


Figure B.10: Visualization of variation of information (VoI) as Venn digram.

B.10 Lie Groups

In general, the notion of Lie groups are applied in modern geometry which are commonly used in robotics to express the degree of freedom (DoF). The reader is advised to the literature by Hilgert and Neeb [Hil+11] for in depth information and notion. However, the most common groups and their meaning within this thesis are summarized in the Table B.28.

Table B.28: Lie groups and their applications within this work.

R^2	translatory group in two dimensions (e.g., translation on a plane)
R^3	translatory group in three dimensions (e.g., translation in a room)
SO(2)	special orthogonal group with $n = 2$ denoting the rotation in two dimensions (i.e., orientation on a plane as yaw angle)
SO(3)	special orthogonal group with $n = 3$ denoting the rotation in three dimensions (i.e., orientation in a room as roll, pitch, and yaw angle)
$\mathrm{R}^3 \times \mathrm{SO}(2)$	denoting the translation in three dimensions with only one possibility for orientation (i.e., translation in a room plus the yaw angle)
SE(2)	special Euclidean group with $n = 2$ denoting the rotation and translation in two dimensions (i.e., combination of R^2 and SO(2))
SE(3)	special Euclidean group with $n = 3$ denoting the rotation and translation in three dimensions (i.e., combination of R^3 and SO(3))

B.11 CITrack & AMiRo

This section gives an overview of the two experimental platforms CITrack and AMiRo. CITrack was developed and published in by the author [Kor+19b].[1] AMiRo was partially developed and published by the author [Her+16].[2]

B.11.1 CITrack

The CITrack comprises a main experiment area of 6 m × 6 m × 1.5 m (width × depth × height) that is perceived by five cameras as depicted in Fig. B.11 (top). The operative hight of 1.5 m is explained by the cameras' overlapping fields of view such that a 10 cm × 10 cm fiducial marker (FM) does never go out of sight. The mentioned volume is covered by all cameras so that experiments can be performed sufficiently among the laboratory (see label-experiments Fig. B.11 (bottom-left) or tracking-experiment[3] (bottom-right)). The area can also be partitioned into four sub-fields running up to four independent experiments in parallel[4]. Robots and objects are attached with FM for position and orientation detection as well as for identification. Four SP-5000M-GE2 grayscale cameras with 8 mm lenses and one SP-5000C-GE2 color camera with 6 mm lens, with a resolution of 2560×2048pixels each, are mounted above the experiment area. Each camera is connected via Ethernet to the university network and is grabbed via GigE-Vision by a common server running Ubuntu 16.04 and ROS Kinetic. Furthermore, all computer based systems are time-synchronized via the Network Time Protocol (NTP) while the cameras are time-synchronized via the Precision Time Protocol (PTP) and synchronously hardware triggered to achieve exact time stamping which is crucial for any later fusion. The server also runs the `multimaster_fkie`[5] [Kou16] to advertise ROS communication in the network. Thus, experiments and recordings can be conducted by any common PC in the network.

[1] The open-source project is comprised in `https://github.com/cognitiveinteractiontracking`.

[2] The open-source project for ROS compatibility is comprised in `https://github.com/autonomoussystemsengineering`.

[3] Video of MIELE RX1 CITrack traversal: `https://youtu.be/6qwv8iizoIU`

[4] VR experience: `https://youtu.be/ezJA2EgBLyk`

[5] `http://wiki.ros.org/multimaster_fkie`

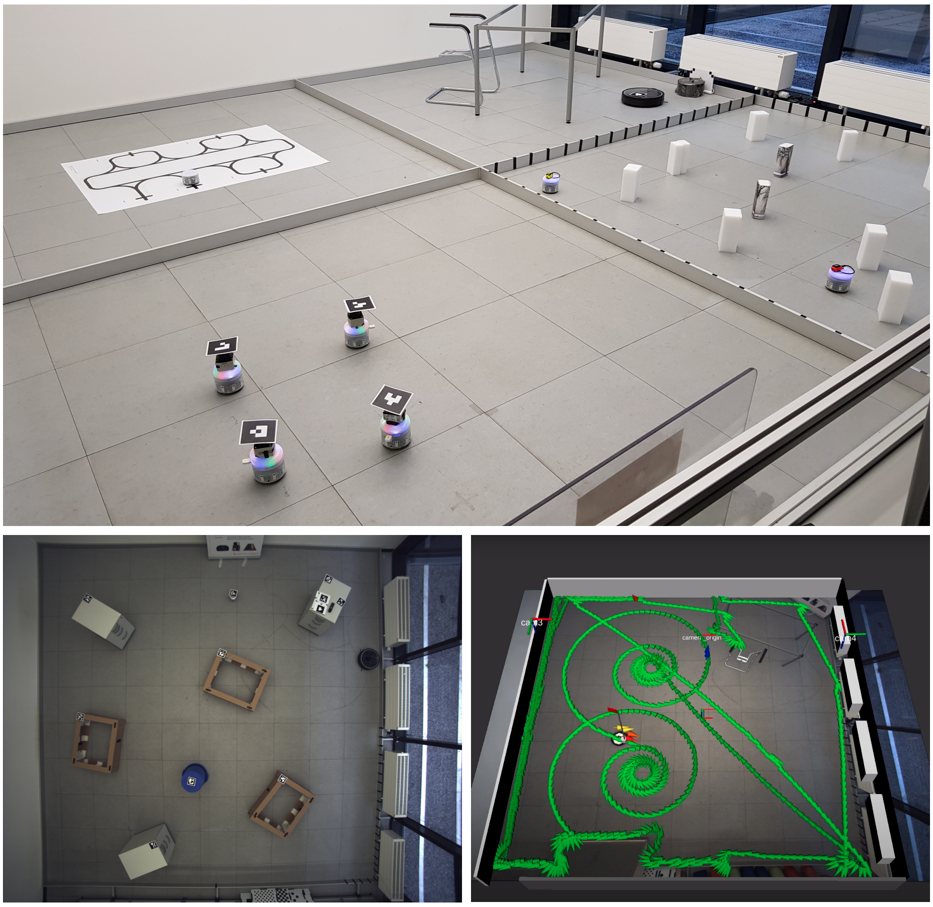


Figure B.11: Exemplary setup of the CITrack. Top: four different experiments, including the AMiRos, running in parallel. Bottom-left: birds-eye view from the centered color camera with labeled objects and robot (bottom-left). Bottom-right: fused tracking (green trajectory) of a MIELE RX1 vacuum cleaning robot transitioning continuously across all cameras (single current and neglected tracks in yellow, red, blue, and orange).

B.11.2 AMiRo Applications in the CITrack

The CITrack has been applied in education and research. While educational purposes concentrate on the trajectory's evaluation and as an indoor localization for control and mapping, it served in various publications for the AMiRo [Kor+18c; Kor+17a; Kor+16c; Kor+17a; Kor+16c; Kor+15]. Therefore, this section is dedicated to the applications which induce the work with the AMiRo. Section B.11.2.2 describes the model identification, which is the calibration of AMiRo's kinematic parameters. Automated data labeling for exteroceptive sensors which is fundamen-

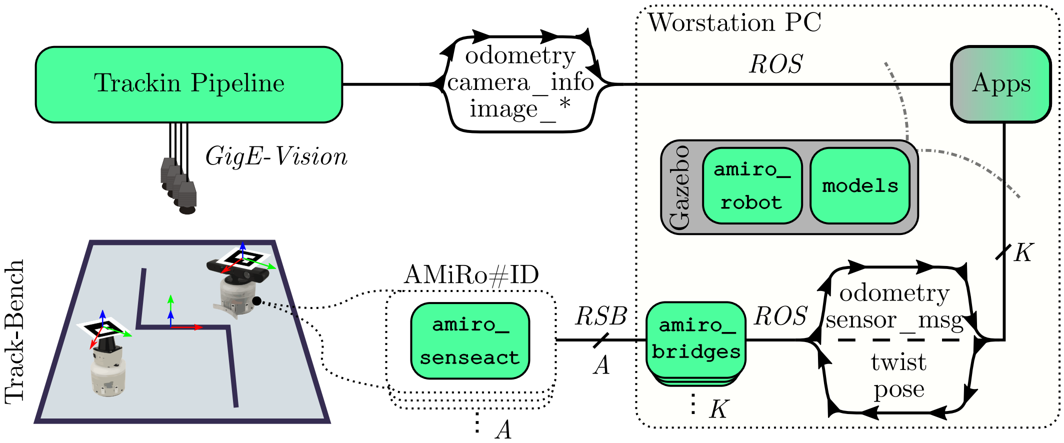


Figure B.12: Setup architecture of CITrack within an AMiRo based multi-robot application. Left: Depiction of the physical CITrack (see B.11.1). Upper structure: Tracking pipeline advertising the absolute poses and images (completely ROS based). Lower structure: RSB interfaces of $A = 2$ AMiRo are advertised by every single robot through the common network. Right: Common workstation PC, running arbitrary applications (Apps), allocating $K \leq A$ robots via `amiro_bridges` that advertise ROS compliant sensor messages and control interfaces. The whole physical setup can be substituted by the Gazebo simulator and the provided models. Worth mentioning, multiple workstations can run the setup in parallel and all ROS topics are automatically namespaced by the robots domain name. Major open-source contributions are highlighted in green, minor contributions and implementations of third parties in partial green, and third party implementation necessary for the setup in gray. Transport types are written in *italic* and package names in `teletype`.

tal for supervised machine learning algorithms and evaluation is explained in section B.11.2.3.

B.11.2.1 AMiRo–CITrack Interaction

The main robotic platform currently used on the CITrack is the Autonomous Mini-Robot (AMiRo) developed by Herbrechtsmeier et al. [Her+16]. To apply AMiRo in multi-robot scenarios with heterogeneous sensor setups, failure-proof, and efficient communication is sufficient. Therefore, the Robotics Service Bus (RSB) runs on all robots. To be further ROS complaint, RSB-ROS bridges translate messages (e.g., sensor percepts or velocity commands) between the two communication frameworks as explained in [Bor+17]. Thus, the CITrack can also directly act as an indoor GNSS-system for the robots. A common setup architecture of the CITrack within a multi-robot setup is illustrated in Fig. B.12. It is worth mentioning that the provided gazebo simulation models introduced in this chapter are able to completely substitute the physical robot and CITrack setup.

B.11.2.2 Model Identification

Model identification concerns the determination of the robots physical parameters, like dimensions as well as static and dynamic behaviors. For applying the AMiRo in multi-robot applications, it is important that all robots have a similar kinematic behavior. Since the fabrication lacks from accuracy and calibration, velocity and turn commands do always have an offset since the essential parameters of a differential kinematic, which are base width b and wheel radius r, differ of each robot. Even the radii of the two wheels on one robot differ, which leads to the well known odometry drift. Therefore, Borenstein and Feng [Bor+95] introduced a calibration method for finding the size factor of the robot's wheels to avoid drift. After this factor has been calibrated, the new absolute wheel radius $\tilde{r}$ and base-width $\tilde{b}$ can be derived from differential kinematics:

$$r_{\text{new}} = \frac{v_{\text{exp.}}}{|\bar{v}_{\text{rec.}}|} r_{\text{old}},\; b_{\text{new}} = \frac{r_{\text{new}}}{r_{\text{old}}} \frac{|\bar{\omega}_{\text{rec.}}|}{\omega_{\text{exp.}}} b_{\text{old}}. \tag{B.47}$$

To accomplish these calibration techniques, the expected linear and angular velocities, $v_{\text{exp.}}$ and $\omega_{\text{exp.}}$, for driving a square clockwise (cw) and counter-clockwise (ccw) are send to the robot. The resulting averaged velocities $|v_{\text{exp.}}|$ and $|\omega_{\text{exp.}}|$ trajectories are recorded by CITrack to set the new parameters of AMiRo. Figure B.13 shows the resulting trajectories and landing-spot-distribution before and after calibration.

B.11.2.3 Data Labeling and Verification

One of the main tasks in autonomous systems is the detection and classification of the environment. Currently, machine learning is a famous technique which allows the parametrization of classification algorithms like a neural network (NN) or support vector machine (SVM). However, all state-of-the-art techniques which learn from data need labeled ground-truth data for parameter optimization (depends on supervised or unsupervised technique), testing, and validation. This still demands exhaustive human labor and is commonly done on single recorded datasets which serve as benchmarks for years. The approach by the CITrack allows an online data labeling in simulation and real-life and thus, allows researchers to design their experiments as intended, and as demanded by any dataset.

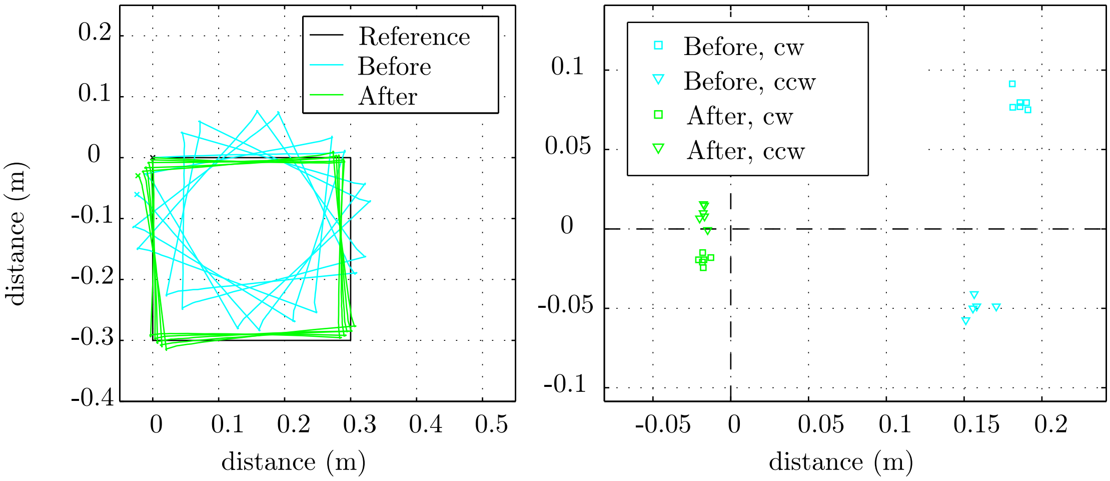


Figure B.13: Trajectory of AMiRo before and after calibration (top) and positioning errors after single rounds (bottom)

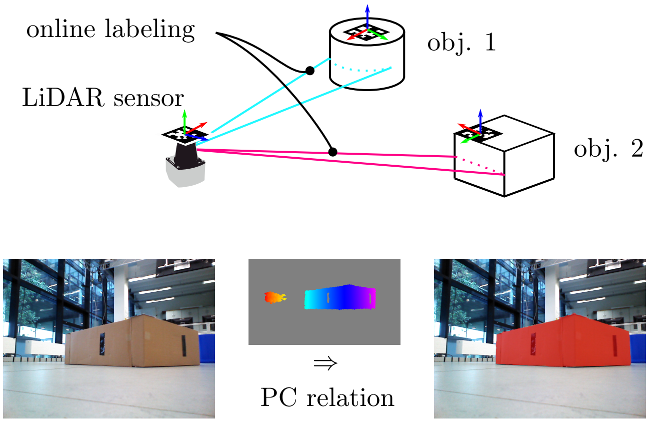


Figure B.14: Schematic depiction of labeling LiDAR data online via known registration between the sensor and an object (obj.) (top). Live online data labeling of RGB data via depth channel (point cloud (PC)) of RGBD camera (bottom). Detections outside of CITrack are neglected and not labeled.

Currently, every exteroceptive sensor which allows a spatial registration of the recorded data can be used in this approach. Further, all objects are attached with FM and the registration between both is known. Since all spacial relations between sensors and objects are observed by CITrack plus the fact the dimensioning of objects is known, the sensor data can be associated to objects as depicted in Fig. B.14. Noise in images and depth data is handled, such that RGBD points are associated to objects in a nearest-neighbor approach. It is worth mentioning that the current approach in simulation is much simpler, as the object ID can be encoded in the reflective channel of each object. Further, the known object poses can also be used for automated simulation building, since the objects can be parametrized by their known poses.

C Mathematical Foundations

Variants of Bayes equation:

$$p(a) = \frac{p(z,a)}{p(z|a)}, \qquad p(z|a) = \frac{p(z,a)}{p(a)}, \qquad p(z,a) = \frac{p(a)}{p(z,a)} \tag{C.1}$$

$$p(a|b,c) \overset{\text{eq. C.1}}{=} \frac{p(a,b|c)}{p(b|c)} \overset{\text{eq. C.1}}{=} \frac{p(a,b,c)}{p(b|c)p(c)} \overset{\text{eq. C.1}}{=} \frac{p(a,b,c)}{p(b,c)} \tag{C.2}$$

Logarithm rules:

$$\log(ab) = \log(a) + \log(b) \tag{C.3}$$

$$a\log(b) = \log(b^a) \tag{C.4}$$

Evidence lower bound:

$$\mathcal{L} = \sum_z q(z|a)\log\left(\frac{p(z,a)}{q(z|a)}\right) \tag{C.5}$$

Marginal likelihood:

$$p(a|b) = \sum_z p(a|z)p(z|b) \tag{C.6}$$

Independent and identically distributed random variables (i.i.d. or iid or IID):

$$p(a,b,c) = p(a)p(b)p(c) \tag{C.7}$$

C.1 Expected Value of a Random Variable

The expected value of a random variable X represents the average of a large number of independent realizations $\mathcal{X} = \{x_1, x_2, \ldots\} \sim X$. If X is discrete, then the expectation of X is defined as

$$\mathrm{E}_P[X] = \sum_{x\in\mathcal{X}} P(x)x, \tag{C.8}$$

where P is the probability mass function of X and $\mathcal{X}$ is the support that is a subset of the domain on which X is defined, of X. If X is continues, then the expectation of X is defined as

$$\mathrm{E}_p[X] = \int_{\mathcal{X}} p(x) x \, \mathrm{d}x =: \mu, \tag{C.9}$$

where p is the probability density function of X. Thus we can interpret $\mathrm{E}[X]$ as a weighted integral of the values x of X, where the weights are the probabilities $p(x)$. Since all properties for continues case hold for the discrete case as well, the explicit staging for discrete variables is neglected in the following for brevity. Furthermore, the braces of the E-operator and the support or range of the integral are neglected:

$$\mathrm{E}\, X := \mathrm{E}[X] := \mathrm{E}_p[X] \quad \text{and} \quad \int p(x) x \, \mathrm{d}x := \int_{\mathcal{X}} p(x) x \, \mathrm{d}x \tag{C.10}$$

If the random variable is perceived through some function g, then, the expected value of a function (aka *Law of the Unconscious Statistician*) of a random variable becomes

$$\mathrm{E}\, g(X) = \int p(x) g(x) \, \mathrm{d}x. \tag{C.11}$$

This leads to the moment, analogue to mechanical systems, as a quantitative measure of the shape of the distribution.

$$\mathrm{E}(X - c)^n = \int p(x)(x - c)^n \, \mathrm{d}x \quad \forall n \in \mathbb{N} \tag{C.12}$$

The value of the integral above is called the n-th moment of the probability distribution p centered about a value c. Known moments are the total probability ($n \equiv 0$) which is 1 by definition, mean ($\mu \in \mathbb{R}$ with $n \equiv 1$ and $c \equiv 0$) denoting the most representative value of a distribution, variance ($\sigma^2 \in \mathbb{R}^+$ with $n \equiv 2$ and $c \equiv \mu$) representing the distribution of probability mass around c, skewness ($\gamma \in \mathbb{R}$ with $n \equiv 3$ and $c \equiv \mu$) revealing if p is more tailed to the left ($\gamma < 0$) or to the right ($\gamma > 0$), and kurtosis ($\kappa \in \mathbb{R}^+$ with $n \equiv 3$ and $c \equiv \mu$) measuring the heaviness of the distribution's tail. The MSEof a distribution can be written as the sum of the variance and the squared mean, providing a useful way to calculate the MSE and implying that in the case of unbiased distribution, the MSE and variance are equivalent:

$$\sigma^2 = \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2 = \mathrm{E}\left[X^2\right] + \mu^2 \Leftrightarrow \mathrm{E}\left[X^2\right] = \sigma^2 + \mu^2 \tag{C.13}$$

For now, the probability density p is supposed to be a distribution over an uni-modal quantity. However, it can be more complex as well by being a joint or conditional density f. Therefore, the expected value is generalized to

$$\mathrm{E}_{f(X)}\, g(X) = \int f(x) g(x) \, \mathrm{d}x. \tag{C.14}$$

This results for example in the expected value, evaluated at y, of the conditional

$$\mathrm{E}_{p(x|y)}[X|Y=\hat{y}] = \int p(x|\hat{y})x\,\mathrm{d}x \underbrace{=:}_{\text{brevity}} \mathrm{E}_{X|Y}[X|Y=\hat{y}]. \tag{C.15}$$

To conclude the definition, the identity of the introduced nomenclature can be used as follows:

$$\mathrm{E}_{p(x)}[X] = \mathrm{E}_X[X] = \mathrm{E}[X] = \mathrm{E}\,X. \tag{C.16}$$

C.2 Further Quantities of a Random Variable

C.2.1 Entropy

The entropyH is the expected surprise of – or average rate at which information is produced by – a random variable X:

$$\mathrm{H}[X] = -\,\mathrm{E}\log_b(p(X)) = -\int p(x)\log_b(p(x))\,\mathrm{d}x \tag{C.17}$$

Since one can find the entropy operator defined as H for discrete – and the differential entropy operator h for continuous – random variable, H is introduced interchangeably. The differences lie in the basis of the logarithm and unit of entropy that is, for example, [H] = bit with $b = 2$ and [h] = nat with $b = \exp$.

C.2.2 Cross Entropy

In information theory, the cross entropy between two probability distributions p and q measures the average amount of information, measured in nats or bits (c.f. section C.2.1), needed to identify an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution q, rather than the true distribution p. The cross entropy is defined as follows:

$$\mathrm{H}[p,q] = -\,\mathrm{E}_p[\log q] = -\int_{\mathcal{X}} p(x)\log q(x)\,\mathrm{d}x \tag{C.18}$$

It relates to the Kullback–Leibler divergence and entropy (c.f. section C.2.3 and C.2.1) as follows:

$$\mathrm{H}[p,q] = \mathrm{D}_{\mathrm{KL}}(p\|q) + \mathrm{H}[p] \tag{C.19}$$

C.2.3 Kullback–Leibler Divergence (KLD)

In statistics, the Kullback–Leibler divergence (also called relative entropy) is a measure of surprise on how one probability distribution q is different from a second, reference probability distribution p. In the simple case, a Kullback–Leibler divergence of 0 indicates that the two distributions are identical. It was introduced by Solomon Kullback and Richard Leibler [Kul+51; Kul59] as the directed divergence between two distributions. In general, if p and q are probability measures over a set $\mathcal{X}$, and p is absolutely continuous with respect to q, then the Kullback–Leibler divergence from q to p is defined as

$$\mathrm{D}_{\mathrm{KL}}(p\|q) = \mathrm{H}[p,q] - \mathrm{H}[p] = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x \geq 0 \tag{C.20}$$

(see [Bis07b, p. 55]).

It is a nonnegative, additive, asymmetric, measure and coincides with the family of f-divergences $\mathrm{D}_f(p\|q) = \int f(\mathrm{d}p/\,\mathrm{d}q)\,\mathrm{d}p$ introduced by Csiszár [Csi67], with all its properties. This integral needs to be solved numerically in general, but there exists a closed-formed expression for Gaussian distributions and others (see Lexa [Lex04]) For two uni-variate Gaussians with $p(x) = \mathcal{N}(x;\mu_1,\sigma_1)$ and $q(x) = \mathcal{N}(x;\mu_2,\sigma_2)$, the KL-divergence can be written as

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(p\|q) &= \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}, && \text{(C.21)}\\
\mathrm{D}_{\mathrm{KL}}(p\|\mathcal{N}(0,1)) &= -\log\sigma_1 + \frac{\sigma_1^2}{2} + \frac{\mu_1^2}{2} - \frac{1}{2}, && \text{(C.22)}\\
\mathrm{D}_{\mathrm{KL}}(\mathcal{N}(0,1)\|q) &= \log\sigma_2 + \frac{1+\mu_2^2}{2\sigma_2^2} - \frac{1}{2} && \text{(C.23)}
\end{aligned}$$

(see section section C.2.5 for full derivation). Interestingly, the equations C.22 and C.23 directly reveal that the KLD is not symmetric and therefore not a metric. Furthermore, pinning the parameters of q in Eq. (C.22) demonstrates that the KLD is convex at least in the parameters of p and q.

C.2.4 Jensen–Shannon Divergence (JSD)

To overcome the issue of asymmetry in the KLD, one can define its symmetrized version $\mathrm{D}_{\mathrm{KL}}(p\|q) + \mathrm{D}_{\mathrm{KL}}(q\|p)$. This comes, however, with the drawback that the calculation of the KLD is unstable and even undefined for $q(x) = 0$ and $p(x) \neq 0$, if it needs to be evaluated numerically. Furthermore, there are certain bounds that the KLD cannot provide for the variational distance and the Bayes probability of error. Therefore, Lin [Lin91] introduced the L divergence as a special case of the JSD for two continues distributions:

$$\mathrm{D_{JS}}(p\|q) = \mathrm{D_{JS}}(q\|p) = \frac{1}{2}\,\mathrm{D_{KL}}\Big(p\Big\|\frac{p+q}{2}\Big) + \frac{1}{2}\,\mathrm{D_{KL}}\Big(q\Big\|\frac{p+q}{2}\Big) \tag{C.24}$$

C.2.5 KLD for two Gaussian distributions

Let $p(x) = \mathcal{N}(x;\mu_1,\sigma_1) \overset{\text{brv.}}{=:} p$ and $q(x) = \mathcal{N}(x;\mu_2,\sigma_2) \overset{\text{brv.}}{=:} q$. Calculating the KL-divergence for two different Gaussian distributions is restricted to solving the two integrals

$$\mathrm{D_{KL}}(p||q) = \int p\log p - \int p \log q \overset{C.18}{=} -\mathrm{H}[p] + \mathrm{H}[p,q]. \tag{C.25}$$

while integration is done over all real values, the solution for $\mathrm{H}[p]$ is given from [Bis07b] with

$$\mathrm{H}[p] = -\int p\log p = -\frac{1}{2}(1+\log 2\pi\sigma_1^2) \tag{C.26}$$

$\mathrm{H}[p,q]$ can be expanded as

$$\mathrm{H}[p,q] = -\int p\log\frac{1}{(2\pi\sigma_2^2)^{(1/2)}}e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \tag{C.27}$$

$$=\frac{1}{2}\log(2\pi\sigma_2^2) - \int p\log e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \tag{C.28}$$

$$=\frac{1}{2}\log(2\pi\sigma_2^2) + \int p\frac{(x-\mu_2)^2}{2\sigma_2^2} \tag{C.29}$$

$$=\frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\int px^2 - 2\mu_2\int px + \mu_2^2\int p}{2\sigma_2^2} \tag{C.30}$$

$$=\frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\mathrm{E}_p[X^2] - 2\,\mathrm{E}_p[X]\mu_2 + \mu_2^2}{2\sigma_2^2}. \tag{C.31}$$

$\mathrm{E}_p[X^2]$ can be substituted via eq. C.13 by $\mathrm{E}_p[X^2] = \sigma_1^2 + \mu_1^2$:

$$\mathrm{H}[p,q] = \frac{1}{2}\log(2\pi\sigma^2) + \frac{\sigma_1^2 + \mu_1^2 - 2\mu_1\mu_2 + \mu_2^2}{2\sigma_2^2} \tag{C.32}$$

$$=\frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2}. \tag{C.33}$$

Putting everything together:

$$\mathrm{D_{KL}}(p||q) = \mathrm{H}[p,q] - \mathrm{H}[p] \tag{C.34}$$

$$\tag{C.35}$$

$$= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + \log 2\pi\sigma_1^2) \tag{C.36}$$

$$\tag{C.37}$$

$$= \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \tag{C.38}$$

C.3 Positiveness of Entropy for a Gaussian Distribution

Putting equation C.20 and equation C.25 together reveals that the entropy of the distribution p is always greater than the cross-entropy between p and the proxy q:

$$\mathrm{D_{KL}}(p||q) = -\,\mathrm{H}[p] + \mathrm{H}[p,q] \geq 0 \qquad \Rightarrow \qquad \mathrm{H}[p,q] \geq \mathrm{H}[p]. \tag{C.39}$$

While this inequality does not allow further deduction of the entropies' sign in general, it is possible to make a statement, if p is Gaussian:

$$\mathrm{H}[p] \overset{C.26}{=} \log 2e\pi\sigma \quad \Rightarrow \quad \mathrm{H}[p] \geq 0 \wedge \mathrm{H}[p,q] \geq 0 \quad \forall\, \sigma \geq \frac{1}{2e\pi} \approx 0.0585 \tag{C.40}$$

This statement is true irregardless of the shape of q.

C.4 Jensen's Inequality

Jensen's inequality, published by Johan Jensen [Jen06], is an elementary inequality for convex and concave functions. Because of its generality, it is the basis of many significant inequalities, especially in analysis and information theory. Jensen's inequality states that the function value $f(x)$ of a convex function at a finite convex combination of its arguments x is always less than or equal to a finite convex combination of the function values of the arguments. Therefore, the inequality can be formulated for any function under the following constraints: a function f needs to be convex with $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \forall t \in [0,1]$ meaning convex, a function g needs to be concave with $-g$ being convex, there exist non-negative and real valued λ_i with $\sum_i \lambda_i = 1$. Given these constraints, the inequality can be formulated as follows:

$$f\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i f(x_i) \quad \text{and} \quad g\left(\sum_i \lambda_i x_i\right) \geq \sum_i \lambda_i g(x_i) \tag{C.41}$$

As depicted in eq. C.41, the difference between convex and concave functions is that Jensen's inequality holds in the opposite direction. These inequalities also hold, if the argument remains constant with $x_i = \text{const.} \forall i$.

C.5 Mixture of Gaussian versus Gaussian - Derivation via Jensen's Inequality

The following derivation shows the application of Jensen's inequality to Eq. (5.130).

$$
\begin{aligned}
\mathrm{H}[p_{\mathrm{M}}] &= \int p_{\mathrm{M}} \log p_{\mathrm{M}} = \int p_{\mathrm{M}} \log \sum_k^K \lambda_k p_{\mathrm{M}_k} && && \text{(C.42)} \\
&\geq \int p_{\mathrm{M}} \sum_k^K \lambda_k \log p_{\mathrm{M}_k} && \text{C.41} && \text{(C.43)} \\
&= \sum_k^K \lambda_k \int p_{\mathrm{M}} \log p_{\mathrm{M}_k} = -\sum_k^K \lambda_k \,\mathrm{H}[p_{\mathrm{M}}, p_{\mathrm{M}_k}] && && \text{(C.44)} \\
&= -\,\mathrm{H}[p_{\mathrm{M}}] - \sum_k^K \lambda_k \,\mathrm{D}_{\mathrm{KL}}(p_{\mathrm{M}} \| p_{\mathrm{M}_k}) && && \text{(C.45)} \\
&\leq -\,\mathrm{H}[p_{\mathrm{M}}] && \lightning && \text{(C.46)}
\end{aligned}
$$

Jensen's inequality is applied in eq. C.42 and further, the sum and integral are swapped. Therefore, the term can be rewritten as:

$$
\mathrm{H}[p_{\mathrm{M}}] \geq -\sum_k^K \lambda_k \,\mathrm{H}[p_{\mathrm{M}}, p_{\mathrm{M}_k}] \quad \overset{5.130}{\Rightarrow} \quad \sum_k^K \lambda_k \,\mathrm{H}[p_{\mathrm{M}_k}, p_{\mathrm{M}}] \leq \sum_k^K \lambda_k \,\mathrm{H}[p_{\mathrm{M}}, p_{\mathrm{M}_k}] \tag{C.47}
$$

Dropping the KLD would lead to an insufficient inequality, since the resulting term would then become bigger as shown in eq. C.46 ($-\,\mathrm{D}_{\mathrm{KL}} \leq 0$). the opposing directed inequalities do then not allow to make any further deduction on the relation ship between $\mathrm{H}[p_{\mathrm{M}}]$ and $-\sum_k^K \lambda_k \,\mathrm{H}[p_{\mathrm{M}}]$

Index