Faculty 2
Institute of Mathematics

Master Thesis

«Robustness of Hybrid Discriminative-Generative Models»

by

Berkant Ferhat Turan 353132

Referees:	Prof. Dr. Sebastian Pokutta
	Prof. Dr. Thorsten Koch
Supervisor:	Prof. Dr. Sebastian Pokutta

Technical University Berlin, Faculty 2 – Institute of Mathematics,
Department for Mathematical Optimization
Berlin, June 15, 2022

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Ort, Datum	Berkant Ferhat Turan

Abstract

In machine learning, probabilistic models are usually divided into discriminative and generative models. While the former often achieve higher accuracy on large datasets, the latter are designed to incorporate more extensive information. Moreover, the decision whether to use a discriminative or a generative model is a fundamental problem in large-scale applications due to safety-related aspects. To address this problem, we extend upon the results of [Kuleshov and Ermon (2017)] and instantiate Hybrid Discriminative-Generative Models (HDGMs) with Residual Neural Networks (ResNets) and Variational Autoencoders (VAEs). The resulting Deep Hybrid Models (DHMs) are examined with respect to three robustness metrics relevant to safety, namely the Expected Calibration Error (ECE), the Out-Of-Distribution (OOD) detection, and the adversarial robustness in the context of the Fast Gradient Sign Method (FGSM). We show that DHMs achieve comparable results to supervised ResNets and unsupervised β-VAEs on well-known image recognition benchmark datasets, that is, Street View House Numbers (SVHN) and CIFAR-10. Furthermore, we demonstrate that DHMs are subject to the same problem as deep generative models in near-OOD detection tasks. This suggests that the multi-conditional objective of DHMs does not provide a solution to the problem of assigning higher likelihood values to near-OOD samples associated with deep generative models. Finally, we show that the ability of DHMs to interpolate between discriminative and generative approaches does not lead to significant improvements in ECE and adversarial robustness.

Zusammenfassung

Im Bereich des maschinellen Lernens werden probabilistische Modelle in der Regel in diskriminative und generative Modelle unterteilt. Während erstere oft eine höhere Genauigkeit bei großen Datensätzen erreichen, sind letztere darauf ausgelegt, umfangreichere Informationen zu berücksichtigen. Darüber hinaus stellt die Entscheidung, ob ein diskriminatives oder ein generatives Modell verwendet werden soll, aufgrund von Sicherheitsaspekten bei groß angelegten Anwendungen ein grundlegendes Problem dar. Um dieses Problem anzugehen, bauen wir auf den Ergebnissen von [Kuleshov and Ermon (2017)] auf und realisieren *Hybrid Diskriminative-Generative Models* (HDGMs) mit *Residual Neural Networks* (ResNets) und *Variational Autoencoders* (VAEs). Die resultierenden *Deep Hybrid Models* (DHMs) werden im Hinblick auf drei sicherheitsrelevante Robustheitsmetriken untersucht, nämlich den *Expected Calibration Error* (ECE), die *Out-Of-Distribution* (OOD)-Detektion und die *adversarial robustness* im Kontext der *Fast Gradient Sign Method* (FGSM). Wir zeigen, dass DHMs vergleichbare Ergebnisse wie überwachte ResNets und unüberwachte β-VAEs auf bekannten Benchmark-Datensätzen zur Bilderkennung, nämlich *Street View House Numbers* (SVHN) und *CIFAR-10*, erzielen. Darüber hinaus zeigen wir, dass DHMs bei der Aufgabe der Erkennung von *near*-OOD Bildern demselben Problem unterliegen wie tiefe generative Modelle. Dies deutet darauf hin, dass die multikonditionale Zielfunktion von DHMs keine Lösung für das Problem der Zuweisung höherer Wahrscheinlichkeitswerte für *near*-OOD Bilder bietet, das mit tiefen generativen Modellen verbunden ist. Und schließlich zeigen wir, dass die Fähigkeit von DHMs, zwischen dem diskriminativen und generativen Ansatz zu interpolieren, nicht zu signifikanten Verbesserungen des ECE und der *adversarial robustness* führt.

Contents

Acronyms

AUROC Area Under the Receiver-Operating Curve.

CNN Convolutional Neural Network.

DHM Deep Hybrid Model.

DLVM Deep Latent Variable Model.

DNN Deep Neural Network.

DoSE Density of States Estimator.

ECE Expected Calibration Error.

ELBO Evidence Lower Bound.

FCNN Fully Connected Neural Network.

FGSM Fast Gradient Sign Method.

FN False Negative.

FP False Positive.

GAN Generative Adversarial Net.

GPU Graphics Processing Unit.

HDGM Hybrid Discriminative-Generative Model.

KL divergence Kullback-Leibler divergence.

NN Neural Network.

OOD Out-Of-Distribution.

PAI Problem of Approximate Inference.

ResNet Residual Neural Network.

ROC curve Receiver Operating Characteristic curve.

SGD Stochastic Gradient Descent.

SGVB Stochastic Gradient Variational Bayes.

SVHN Street View House Numbers.

TN True Negative.

TP True Positive.

VAE Variational Autoencoder.

VI Variational Inference.

1 Introduction

In this introductory chapter, the objective of this thesis is motivated and formulated. Subsequently, the content of the individual chapters is summarized to provide an overview of the work.

1.1 Motivation

Over the past decade, machine learning has made tremendous progress, especially with the advancement of deep learning. Its success story spans many areas where other methods have seemingly reached their limits. The resulting widely recognized relevance of deep learning has led to an astonishing increase in knowledge and peer-reviewed publications [Zhang *et al.* (2021)].

One of the best known successes of the last decade is the use of deep learning for computer vision. More specifically, the most widely recognized breakthroughs were initially achieved in image classification or recognition [Krizhevsky, Sutskever, and Hinton (2012)]. The techniques inspired by deep learning not only surpassed state-of-the-art methods, but also proved to surpass human-level on some restricted visual tasks [He, Zhang, Ren, and Sun (2016)]. This has led to the increasing use of deep learning models for computer vision in industry, which has enabled a number of new applications such as self-driving cars, medical image analysis, detecting defective parts in manufacturing, and many more [Zhang *et al.* (2021)].

However, the success story is not limited to computer vision. Important contributions of deep learning can also be found in the area of natural language processing. Natural language processing includes various tasks such as translation, text generation, and semantic or sentiment analysis, to name a few. Especially in recent years, state-of-the-art deep learning models like BERT [Devlin, Chang, Lee, and Toutanova (2019)] or GPT-3 [Brown *et al.* (2020)] could be developed, which incorporate so-called transformers [Vaswani *et al.* (2017)]. According to leading technology companies like Microsoft and Google, these models have been integrated into search engines and are used by billions of people [Zhang *et al.* (2021)]. Other broad areas where deep learning has

led to breakthroughs include speech recognition [Deng and Li (2013)], playing complex games [Berner *et al.* (2019)], and drug development [Senior *et al.* (2020)], among many others.

But despite the astonishing advances in deep learning, major concerns have been raised about artificial intelligence safety in view of its large-scale application [Amodei *et al.* (2016)]. One of the safety-related issues concerns the robustness of Deep Neural Networks (DNNs) employed in mission-critical tasks. In computer vision, which is the focus of this thesis, the term robustness has taken on many definitions. Most commonly, robustness in this particular field is defined as robustness to adversarial examples [Szegedy *et al.* (2013)]. In addition to the above, we follow a broader interpretation and add calibration [Guo, Pleiss, Sun, and Weinberger (2017)] and Out-Of-Distribution (OOD) detection [Bishop (1994)] to the definition of robustness, as suggested by [Grathwohl *et al.* (2019)].

Recent studies have shown that while deep discriminative models have achieved human-level performance on limited image classification tasks, the predicted probability estimates diverged from the true correctness likelihood [Guo, Pleiss, Sun, and Weinberger (2017)]. In medical image analysis, for example, models are supposed to indicate whether the decision associated with a particular image may be incorrect. If a model's decisions can be successfully labeled as "potentially inaccurate", human intervention can be initiated. In addition, state-of-the-art deep generative models tend to fail in detecting images that come from a different distribution than the training data. Consequently, a classifier will silently assign one of the classes from the training data to the image [Nguyen, Yosinski, and Clune (2015); Nalisnick *et al.* (2018)]. Considering that distributional shifts are common in real-world applications, this questions the reliability and safety of such employed models.

In machine learning, models are often divided into discriminative and generative approaches, depending on which probability distribution is being approximated [Ng and Jordan (2001)]. In addition to the two approaches mentioned above, there is also the hybrid approach. Kuleshov and Ermon have developed a flexible framework to combine discriminative and generative models into so-called Hybrid Discriminative-Generative Models (HDGMs) [Kuleshov and Ermon (2017)]. Due to its flexibility, modern deep learning models can be integrated, leading to models, that we refer to as Deep Hybrid Models (DHMs).

1.2 Objective of the thesis

The underlying architecture of DHMs is based on coupling both discriminative and generative models via latent variables. In addition, the optimization requires a multi-conditional objective that may be useful for the robustness considerations mentioned above due to inherent regularization effects. The question considered in this thesis is whether DHMs equipped with Residual Neural Networks (ResNets) [He, Zhang, Ren, and Sun (2016)] and Variational Autoencoders (VAEs) [Kingma and Welling (2014)] lead to better performance metrics for robustness in terms of Expected Calibration Error (ECE), OOD detection and adversarial robustness in the context of the Fast Gradient Sign Method (FGSM).

The results are compared to those obtained with the respective components of the DHMs, namely supervised ResNets and unsupervised VAEs. Moreover, since DHMs are able to interpolate between the discriminative and generative approaches, the results are compared to the purely discriminative setting of DHMs to see if the multi-conditional objective provides advantages over the purely discriminative objective. To this end, all investigated models are trained with the same training configurations on well-known image recognition benchmark datasets and evaluated with respect to the performance metrics. To the best of our knowledge, there is no previous work that considers the analysis of DHMs with respect to any of the above robustness metrics.

1.3 Outline of the thesis

Following this introduction, the outline of this thesis is as follows.

In Chapter 2, the basic concept of probabilistic modeling is introduced, as well as the division between probabilistic discriminative and generative modeling. In addition, probabilistic models parameterized by Neural Networks (NNs) relevant to this work are outlined. Then, the general framework of Deep Latent Variable Models (DLVMs) and the related Problem of Approximate Inference (PAI) is presented. Finally, VAEs , which are commonly used to solve the PAI, and the corresponding objective function are derived as well as the extension to β-VAEs [Higgins *et al.* (2017)].

Chapter 3 presents the underlying framework of HDGMs and the associated multi-conditional objective function. Moreover, two hyperparameters are introduced, which allow interpolation between the discriminative and generative

approaches. The last section of Chapter 3 presents our specific implementation of DHMs with ResNets and β-VAEs.

The robustness metrics regarding ECE, OOD detection and adversarial robustness are presented in Chapter 4. Furthermore, the binary classification problem associated with OOD detection is highlighted, as well as the FGSM commonly used to study the adversarial robustness of discriminative models.

Chapter 5 deals with the numerical experiments conducted on various image recognition datasets. In addition, specific questions are formulated to which our numerical experiments contribute to. Subsequently, the results of all studied models are presented and analyzed in detail.

Finally, in Chapter 6, the thesis closes with a summary of the main findings and a general conclusion regarding the robustness of DHMs. In addition, potential drawbacks of the experimental setup as well as possible future research directions regarding DHMs are pointed out.

2 Probabilistic Modeling

In this chapter, the concept of probabilistic modeling is introduced and categorized into discriminative and generative approaches. Furthermore, the respective strengths and weaknesses of discriminative and generative modeling are highlighted to emphasize the extension to hybrid models, which attempt to combine the advantages of both methods. Additionally, recent advances in Variational Inference (VI) are presented in the context of Variational Autoencoders (VAEs), which are incorporated in the Deep Hybrid Models (DHMs) in Chapter 3.

2.1 Probabilistic Models

To start with, the class of machine learning algorithms, which we will focus on, aims to approximate the probability distributions of natural or artificial phenomena from data. Since modeling intrinsically captures only a small fraction of the underlying processes, probabilistic models formulate the inherent uncertainty via probability distributions. Probabilistic models play an important role in pattern recognition tasks, such as the classification of unknown data or to obtain a better understanding of the phenomena [Bishop (2006)].

One common problem discussed in this chapter is the problem of density estimation. Consider an observed variable $\mathbf{x} \sim p^*(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$ is a random sample from the underlying unknown probability distribution $p^*(\mathbf{x})$. Let all observed samples be collected in the dataset $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^n$ with $n \in \mathbb{N}$. We wish to find an approximation $p_{\boldsymbol{\theta}}(\mathbf{x})$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ such that

$$p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p^*(\mathbf{x})$$

for any observed variable $\mathbf{x} \in \mathcal{X}$. Here, the subscript $\boldsymbol{\theta}$ is used to denote that the approximation $p_{\boldsymbol{\theta}}(\mathbf{x})$ is parametrized by $\boldsymbol{\theta}$. Most commonly, the goal of density estimation is achieved by learning the parameters $\boldsymbol{\theta}$. Once $p_{\boldsymbol{\theta}}(\mathbf{x})$ is learned, it can be used for many purposes, e.g., sampling or outlier detection [Bishop (1994)]. That being said, the fundamental difference in

discriminative and generative modeling lies in the chosen distribution that is being approximated.

2.1.1 Probabilistic Discriminative Models

If we choose to learn the conditional probability $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ over the labels $\mathbf{y} \in \mathcal{Y}$ given the observations $\mathbf{x}$, we obtain a discriminative model. In other words, discriminative models aim to predict the labels $\mathbf{y}$ from observed features $\mathbf{x}$ via a conditional model

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \approx p^*(\mathbf{y}|\mathbf{x}) \ .$$

Given $C \in \mathbb{N}$ different class labels, $\mathbf{y}$ can be considered to be a real-valued vector of size C, where all entries are zero except one. The non-zero entry is set to one and indicates the class of the corresponding observed variable $\mathbf{x}$. In classification or regression problems, such models are learned through supervised learning. Typically, in the paradigm of supervised learning one attempts to find a parametric function, $\Phi_{\boldsymbol{\theta}} \colon \mathcal{X} \to \mathbb{R}^C$, which maps each datapoint to so-called logits. These logits are used to approximate the posterior utilizing the softmax transfer function, i.e.,

$$p_{\boldsymbol{\theta}}(y_i|\mathbf{x}) = \frac{\exp(\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i])}{\sum_i \exp(\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i])} \,, \tag{2.1}$$

where $\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i]$ denotes the logit corresponding to the i^{th} label. The parametric function $\Phi_{\boldsymbol{\theta}}$ is learned by minimizing the negative log-likelihood,

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p^*(\mathbf{x},\mathbf{y})}[\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})] \,, \tag{2.2}$$

which can be derived from the categorical cross-entropy loss [Goodfellow, Bengio, and Courville (2016)]. Minimizing the cross-entropy loss is analogous to the minimization of the Kullback-Leibler divergence (KL divergence) between $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and $p^*(\mathbf{y}|\mathbf{x})$ [Liu and Abbeel (2020)]. Note, that the true posteriors are encoded by one-hot vectors with non-zero entries for the respective class.

Common examples of probability models relevant to this work are parametrized by differentiable feed-forward Neural Networks (NNs). For example, we use Fully Connected Neural Networks (FCNNs), see Definition 2.2 based on [Berner, Grohs, Kutyniok, and Petersen (2021)].

Definition 2.1 (Neuron)**.** *Let $d \in \mathbb{N}$ be the input dimension. An (artificial) neuron is defined as the mapping $f \colon \mathbb{R}^d \to \mathbb{R}$, $f(\mathbf{x}) = \varrho\left(\sum_{i=1}^{d} w_i x_i - b\right)$, where*

$\boldsymbol{w} = (w_1, \dots, w_d) \in \mathbb{R}^d$ and $b \in \mathbb{R}$ denote the weights and bias, respectively. Moreover, we refer to $\varrho\colon \mathbb{R} \to \mathbb{R}$ as the (non-linear) activation function.

Definition 2.2 (Fully Connected Neural Network)**.** *Let $L \in \mathbb{N}$ be the number of layers and $\boldsymbol{N} = (N_0, \dots, N_L) \in \mathbb{N}^{L+1}$ denote the number of neurons in each layer. The underlying architecture of a FCNN is described by $\boldsymbol{a} = (\boldsymbol{N}, \varrho)$. For $l = 1, \dots, L$, the weight matrix and bias vector of the l-th layer of the directed acyclic graph are denoted by $\boldsymbol{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\boldsymbol{b}^l \in \mathbb{R}^{N_l}$, respectively. Let $P(\boldsymbol{N}) \coloneqq \sum_{l=1}^{L} N_l N_{l-1} + N_l$ be defined as the total number of parameters. In correspondence to the underlying architecture $\boldsymbol{a}$, the realization of the FCNN is given by the mapping*

$$\Phi_a\colon \mathbb{R}^{N_0} \times \mathbb{R}^{P(\boldsymbol{N})} \to \mathbb{R}^{N_L}\,,$$

satisfying $\Phi_a(\mathbf{x}, \boldsymbol{\theta}) = \Phi^{(L)}(\mathbf{x}, \boldsymbol{\theta})$ and

$$\begin{aligned}
\Phi^{(1)}(\mathbf{x}, \boldsymbol{\theta}) &= \boldsymbol{W}^{(1)}\mathbf{x} + \boldsymbol{b}^{(1)} \\
\bar{\Phi}^{(l)}(\mathbf{x}, \boldsymbol{\theta}) &= \varrho\left(\Phi^{(l)}(\mathbf{x}, \boldsymbol{\theta})\right), \quad l \in \{1, \dots, L-1\} \quad \text{and} \\
\Phi^{(l+1)}(\mathbf{x}, \boldsymbol{\theta}) &= \boldsymbol{W}^{(l+1)}\bar{\Phi}^{(l)}(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{b}^{(l+1)}\,.
\end{aligned}$$

Note, that the activation function ϱ is applied element-wise. The dependence of $\boldsymbol{\theta}$ is also given by the subscript as in Eq. (2.1) and is used interchangeably.

However, many other architectures have been established in the field of probabilistic modeling. Particularly, modern Deep Neural Networks (DNNs) using architectures other than FCNNs as in Definition 2.2 have proven to surpass human-level performance on various challenges and thus have become relevant for mission-critical tasks, e.g., self-driving cars or medical image analysis [He, Zhang, Ren, and Sun (2015); Fauw *et al.* (2018); Levinson *et al.* (2011)]. By DNNs, we refer to NNs with $L > 2$, i.e., multiple hidden layers. One major contribution of DNNs is given by the award-winning Residual Neural Networks (ResNets) [He, Zhang, Ren, and Sun (2015)].

Residual Neural Networks. ResNets are typically based on convolutional operators with compactly supported filters [LeCun, Bottou, Bengio, and Haffner (1998)]. The filters aim to extract features of the input, which incorporate affine transformations and element-wise nonlinearities. The affine transformations depend on parameters $\boldsymbol{\theta}$, also called weights, as in Definition 2.2. Learning weights in the context of supervised learning is formulated as an optimization problem, e.g., given in Eq. (2.2).

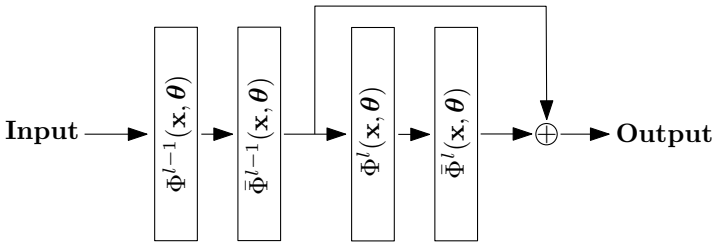


Fig. 2.1: Illustrative example of a residual block with the identity layer added to the l-th layer. The mappings Φ and $\bar{\Phi}$ are according to Definition 2.2.

Leaving aside the convolution operator, the main difference between a FCNN and a ResNet is the additional identity layer that can be added to the l-th layer by the following redefinition,

$$\bar{\Phi}^{(l)}(\mathbf{x}, \boldsymbol{\theta}) = \varrho\left(\Phi^{(l)}(\mathbf{x}, \boldsymbol{\theta})\right) + \bar{\Phi}^{(l-1)}(\mathbf{x}, \boldsymbol{\theta}),$$

for the simplified case of $N_l = N_{l-1}$ [Berner, Grohs, Kutyniok, and Petersen (2021)]. This constitutes a so-called residual block. A sketch of such a residual block can be seen in Figure 2.1. In this illustrative example, the identity layer is added to the l-th layer.

Typically, the number of learnable layers within a ResNet is referenced. For instance, a ResNet with 18 learnable layers is called ResNet-18. Static layers such as max-pooling layers are usually integrated, but not referenced [Ioffe and Szegedy (2015)]. Note, that the ResNet architecture can be utilized for both supervised and unsupervised learning and thus provides a versatile tool for modeling probabilistic models.

Challenges of Discriminative Deep Neural Networks. As mentioned above, discriminative modeling can have disadvantages compared to other modeling approaches. Some issues, which will be discussed in more detail in Chapter 4, are as follows:

- The requirement of large labeled datasets poses a limitation on the adoption of discriminative models parametrized by DNNs. The annotation of large datasets by humans, which are obligatory for DNNs to achieve high accuracy and generalization, can be very expensive and thus infeasible. Generative models have been shown to be competitive with discriminative models in terms of accuracy on small datasets [Ng and Jordan (2001)].
- Another problem associated with discriminative DNNs is the poor calibration of modern architectures. By calibration, we refer to confidence

calibration, which is a measure of how accurately the predicted probability estimates represent the true correctness likelihood [Guo, Pleiss, Sun, and Weinberger (2017)]. Although the accuracy of recent classification networks has improved drastically, the ability to indicate whether the model might be incorrect has suffered [Lakshminarayanan, Pritzel, and Blundell (2016)].

- Moreover, a serious issue concerning discriminative DNNs is their vulnerability to so-called adversarial attacks [Szegedy *et al.* (2013)]. Adversarial attacks are inputs to discriminative models that are deliberately intended to negatively influence the classification.

2.1.2 Probabilistic Generative Models

Probabilistic generative modeling, on the other side, aims to learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ that approximates the true joint distribution $p^*(\mathbf{x}, \mathbf{y})$ over $\mathbf{x}, \mathbf{y} \in \mathcal{X} \times \mathcal{Y}$. Thus, more complex information is inherent in the generative model, which enables the model to answer more general queries such as imputing or denoising $\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$ as well as predicting unknown labels by utilizing Bayes' rule [Kuleshov and Ermon (2017)]. As mentioned above, one prominent domain of generative models is semi-supervised learning. Since the annotation of data by humans can be very expensive, many problems do not provide sufficient labeled data for supervised methods to approach their asymptotic error effectively [Ng and Jordan (2001)]. Given the circumstances of a larger, yet unlabeled dataset, generative models can leverage unlabeled data to improve its generalization [Zhu and Goldberg (2009)]. Moreover, generative models approach their asymptotic error in terms of accuracy much faster in comparison to discriminative models [Ng and Jordan (2001)]. Further recent and impactful advances in generative modeling can be found in the field of language modeling, image generation, language pre-training, and vision pre-training [Goodfellow *et al.* (2014); Radford *et al.* (2019); Zhou *et al.* (2020)].

All of these contributions have incorporated generative models parametrized by DNNs. However, if predicting unknown labels is the only objective, discriminative models tend to obtain higher accuracies for large labeled datasets since the model parameters are used more efficiently [Kuleshov and Ermon (2017)].

2.2 Latent Variable Modeling

The problem of density estimation can be approached by latent variable models. Latent variable models are an extension to fully-observed models discussed in the previous section. The motivation for latent variable models originates from the idea that all datapoints from a given dataset may lie close to a manifold of lower dimensionality compared to the original high dimensional space [Bishop (2006)]. Many of the aforementioned advances in generative modeling make use of this idea.

Let $\mathbf{x} \sim p_\theta(\mathbf{x})$ be the observed variable and let $\mathbf{z} \sim p_{\boldsymbol{\theta}}(\mathbf{z})$ be the latent variable. The marginal distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$, also called marginal likelihood or model evidence, over the observed variables, is given by

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) \, \mathrm{d}\mathbf{z} \,, \tag{2.3}$$

where $p_\theta(\mathbf{x}, \mathbf{z})$ is the latent variable model. Such compound probability distributions provide great flexibility for modeling. In the case of discrete variables $\mathbf{z}$ and a Gaussian distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, one would obtain a mixture-of-Gaussian model. If we choose $\mathbf{z}$ to be continuous, $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be considered to be an infinite mixture [Kingma and Welling (2019)].

The goal of density estimation is approached by maximizing the log-likelihood of the given dataset. We assume that the datapoints are independently and identically distributed and hence, the log-likelihood of the dataset $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ can be simplified to the sum of each log-likelihood, respectively, i.e,

$$\log p_{\boldsymbol{\theta}}(\mathcal{X}) = \sum_{\mathbf{x} \in \mathcal{X}} \log p_\theta(\mathbf{x}) \,. \tag{2.4}$$

The objective is to find the parameters $\boldsymbol{\theta}$ such that Eq. (2.4) is maximized. Since DNNs have proven to be well suited for probabilistic modeling, we will consider latent variable models $p_\theta(\mathbf{x}, \mathbf{z})$ whose underlying distributions are parametrized by DNNs. Such models will be referred to as Deep Latent Variable Models (DLVMs) [Kingma and Welling (2019)].

2.2.1 Problem of Approximate Inference

The optimization of the objective function Eq. (2.4) is typically done via Stochastic Gradient Descent (SGD). Modern DNNs are often trained on large datasets to achieve high accuracy and better generalization. However, optimization concerning all datapoints simultaneously is inefficient. Therefore, only small

batches of the dataset, which are randomly selected in each step, will be used for the computation of the gradient [Bottou (2004)]. To optimize the objective function in Eq. (2.4), one needs to compute the marginal likelihood efficiently. Unfortunately, Eq. (2.3) is typically intractable, due to the required integration, which does not provide a closed-form analytical solution. Moreover, numerical integration techniques are not applicable, because of the high dimensionality and the complexity of the integrand [Bishop (2006)]. Thus, a central task of latent variable models is to solve the Problem of Approximate Inference (PAI).

The Problem of Approximate Inference. Let $\mathbf{x} \in \mathcal{X}$ be observed variables and $\mathbf{z} \in \mathcal{Z}$ latent variables. The joint probability over $\mathbf{x}$ and $\mathbf{z}$ is given by $p_\theta(\mathbf{x}, \mathbf{z})$. Given the joint probability, Bayes' theorem states that the conditional probability $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ can be formulated as follows,

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})} \,. \tag{2.5}$$

According to Eq. (2.5), the intractability of the denominator $p_{\boldsymbol{\theta}}(\mathbf{x})$ is related to the intractability of $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. If one can compute either $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ or $p_{\boldsymbol{\theta}}(\mathbf{x})$, the other probability can be evaluated as well. Hence, the PAI consists of finding an approximation of the conditional probability $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, which is tractable. Note, that the joint probability can be factorized by

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \,, \tag{2.6}$$

where each factor is specified and thus the evaluation of $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ is tractable.

Various methods have been developed to solve the PAI associated with latent variable models. A common family of techniques, which attempts to approximate the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ by formulating the problem as an optimization problem, are known as Variational Inference (VI) methods. An impactful contribution in the field of VI, which we will focus on, are so-called VAEs [Kingma and Welling (2014); Rezende, Mohamed, and Wierstra (2014)]. In the following, the idea of VI methods is presented using VAEs as an example.

2.3 Variational Autoencoder

The previous section dealt with DLVMs and the PAI. Recent advances in VI allow computationally efficient approximations of the posterior distribution

$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ and thus constitute an important building block of Deep Hybrid Models (DHMs) discussed further down in Chapter 3.

Let the prior over the latent variables of the VAE be given by the multivariate isotropic Gaussian

$$p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{I}) \,,$$

where $\boldsymbol{I} \in \mathbb{R}^{m \times m}$ denotes the identity matrix. The intractable posterior in Eq. (2.5) is approximated by VAEs through a parametric inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, also called encoder or recognition model [Kingma and Welling (2019)]. Accordingly, the posterior $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ given in Eq. (2.6) is called a decoder. A schematic representation of the underlying architecture of VAEs is shown in Figure 2.2. Notice, that the variational parameters of the encoder are denoted by $\boldsymbol{\phi} \in \mathbb{R}^p$. Employing SGD, the variational parameters $\boldsymbol{\phi}$ are optimized such that

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \,.$$

Similar to DLVMs, the encoder and decoder can be parametrized by DNNs. Let $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}}^2 \boldsymbol{I})$ be a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_{\mathbf{z}} \in \mathbb{R}^m$ and standard deviation $\boldsymbol{\sigma}_{\mathbf{z}} \in \mathbb{R}^m$. A common choice is to assume that the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is parametrized by the multivariate Gaussian distribution

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}}^2 \boldsymbol{I}) \,,$$

cf. [Kingma and Welling (2014)].

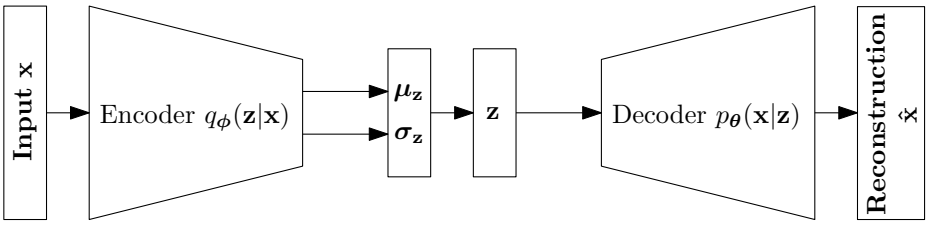


Fig. 2.2: Schematic representation of the underlying architecture of VAEs. The input $\mathbf{x}$ is encoded into latent variables $\mathbf{z}$ by sampling from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}}^2 \boldsymbol{I})$. These variables are then decoded to yield the reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^d$.

The weights and biases of the encoding NN, indicated by $\boldsymbol{\phi}$, are optimized to fit the mean and standard deviation such that the true posterior is approximated.

Furthermore, the variational parameters of VAEs are optimized with respect to all datapoints, i.e., the variational parameters are shared across all datapoints, which is known as amortized inference [Kingma and Welling (2019)]. Therefore, optimization methods such as SGD can be utilized to handle large datasets efficiently.

2.3.1 Evidence Lower Bound

The objective function of VAEs is given by the Evidence Lower Bound (ELBO). The derivation of the ELBO can be done in various ways, most commonly using Jensen's inequality. However, we follow the derivation provided in [Kingma and Welling (2019)], which does not make use of the Jensen inequality.

The log-likelihood of a given observation $\mathbf{x}$ can be rewritten as follows,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \tag{2.7}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \tag{2.8}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \tag{2.9}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]. \tag{2.10}$$

We obtain Eq. (2.7), because the log-likelihood can be considered to be constant in the expectation with respect to the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. Moreover, Eq. (2.8) is obtained due to Bayes' rule given in Eq. (2.5). The second term on the right-hand side in Eq. (2.10) is the KL divergence of the approximate from the true posterior, i.e.,

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] = D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))\,.$$

By definition the KL divergence is non-negative,

$$D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) \geq 0\,,$$

and zero if and only if, the two probability distributions are identical. Due to the non-negativity of the KL divergence, the first term on the right-hand side of Eq. (2.10) is referred to as individual-datapoint ELBO $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$, i.e.,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\right]. \tag{2.11}$$

The above individual-datapoint ELBO is equivalent to the more common formulation

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}))\,, \tag{2.12}$$

where $-\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right]$ is known as the expected reconstruction loss [Kingma and Welling (2014)].

The optimization of the lower bound is done with respect to both $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. Similar to Eq. (2.4), the ELBO of the dataset can be written as the sum of the individual-datapoint ELBOs of each observed variable, i.e.,

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{X}) = \sum_{\mathbf{x}\in\mathcal{X}} \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\,.$$

In contrast to other traditional VI methods, the joint optimization of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be achieved by utilizing the SGD [Kingma and Welling (2019)]. However, since the required differentiation of the objective function with respect to the model parameters includes expectations, an approximation of the gradient $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ is required.

The gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ can be estimated directly by a Monte Carlo estimator. However, the gradient regarding the variational parameters $\boldsymbol{\phi}$ is more complicated, due to the expectation, which is taken with respect to the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. In this non-trivial case, the approximation of $\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{X})$, obtained by a simply Monte Carlo estimator, yields a high variance, which is inefficient and thus impractical [Paisley, Blei, and Jordan (2012)]. A better approach to obtain an estimation of the required gradient with lower variance is called the Stochastic Gradient Variational Bayes (SGVB) estimator proposed by [Kingma and Welling (2014)]. The SGVB estimator incorporates a so-called reparametrization trick, which allows us to efficiently approximate the gradient with respect to $\boldsymbol{\phi}$ such that the SGD algorithm can be utilized.

2.3.2 Reparametrization Trick

In order to introduce the SGVB estimator, we first need to introduce the reparametrization trick. The reparametrization trick provides a way to back-propagate the gradient through Gaussian distributions and can be considered to be a technique to obtain a change of variables. Let $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ be a random auxiliary variable sampled from $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon}; \mathbf{0}, \boldsymbol{I})$, which is independent of $\mathbf{x}$ and $\boldsymbol{\phi}$. The auxiliary variable $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$ is used to avoid the non-deterministic sampling process of $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ by expressing the continuous latent variable via the differentiable transformation

$$\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\varepsilon}, \mathbf{x})\,.$$

Note, that the differentiable transformation $g_{\boldsymbol{\phi}}$ is parametrized by the variational parameters $\boldsymbol{\phi}$. Also recall, that the posterior is approximated by

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}^2_{\mathbf{z}}\boldsymbol{I})\,,$$

where the mean and standard deviation $(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}})$ are the outputs of the encoding NN. Given the mean and standard deviation, we obtain a random sample $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ by the following equation:

$$\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\varepsilon}, \mathbf{x}) = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\sigma}_{\mathbf{z}} \odot \boldsymbol{\varepsilon}\,,$$

where $\odot$ denotes the element-wise product of two vectors. Meanwhile, we avoid the sampling process of the latent variable from the underlying distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$.

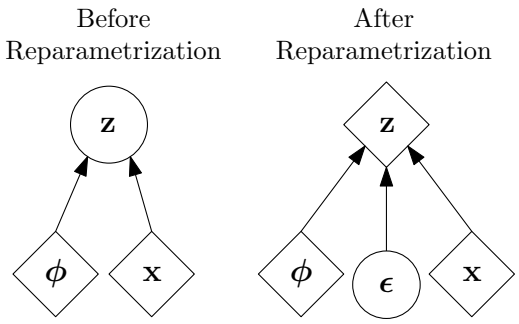


Fig. 2.3: Illustrative sketch of the reparametrization trick. Rectangles represent deterministic nodes, while circles represent random nodes. On the left side, latent variables are randomly sampled from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. On the right-hand side, the reparametrization trick was applied, resulting in a deterministic node for $\mathbf{z}$.

With the above reparametrization trick, there are two possibilities to obtain the SGVB estimate. Firstly, for $M \geq 1$, the expectations in Eq. (2.11) for a single datapoint can be estimated by

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \simeq \tilde{\mathcal{L}}^{\mathrm{A}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \frac{1}{M}\sum_{l=1}^{M} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}^{(l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)}|\mathbf{x})\,,$$

where $\mathbf{z}^{(l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\varepsilon}^{(l)}, \mathbf{x})$ and $\boldsymbol{\varepsilon}^{(l)} \sim p(\boldsymbol{\varepsilon})$. The symbol $\simeq$ denotes unbiasedness of the estimation.

The alternative estimate $\tilde{\mathcal{L}}^{\mathrm{B}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ can be derived from the expression of the ELBO given in Eq. (2.12). The KL divergence in Eq. (2.12) can be calculated analytically, if the underlying distributions are parametrized by Gaussian distributions [Kingma and Welling (2014)]. Thus, an estimation of the KL divergence $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}))$ is not necessary in our case. Given the analytical integration of the KL divergence, the SGVB estimate of the ELBO

$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \simeq \tilde{\mathcal{L}}^{\mathrm{B}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ can be written as follows,

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \simeq \frac{1}{2}\sum_{i=1}^{m}\left(1+\log(\sigma^2_{\mathbf{z},i}) - \mu^2_{\mathbf{z},i} - \sigma^2_{\mathbf{z},i}\right) + \frac{1}{M}\sum_{l=1}^{M}\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(l)})\,,$$

where m denotes the dimensionality of $\mathbf{z}$.

2.3.3 β-VAE

Another desired property of DLVMs or VAEs is to generate latent variables $\boldsymbol{z}$, where each variable z_i is disentangled or factorized, i.e., the latent variable is sensitive to only one single factor and relatively insensitive to others. An advantage often associated with a disentangled representation is that it is easy to interpret and generalize to a variety of tasks.

Following the framework of VAEs, β-VAEs was proposed as a modification to improve the ability of VAEs to obtain disentangled latent representations [Higgins *et al.* (2017)]. The modification is done by introducing an adjustable hyperparameter $\beta \in \mathbb{R}$ to the original VAE objective in Eq. (2.12) as follows,

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x},\beta) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}))\,.$$

The original objective is obtained for $\beta = 1$. Higher values for β encourage the model to learn more efficient representation and promote disentanglement. However, this may also lead to a trade-off between the quality of the reconstructions and the degree of disentanglement. For an in-depth study of the disentangling properties of β-VAEs, we refer to [Burgess *et al.* (2018)].

3 Hybrid Discriminative-Generative Modeling

The choice of whether to use a discriminative or generative model is a fundamental problem in machine learning. To address this problem, Kuleshov and Ermon have come up with a novel framework for Hybrid Discriminative-Generative Models (HDGMs) that can interpolate between a purely discriminative and a purely generative approach [Kuleshov and Ermon (2017)]. In this chapter, we introduce the general concept of HDGMs as well as the architecture of our Deep Hybrid Models (DHMs).

The general idea of HDGMs as proposed in [Kuleshov and Ermon (2017)] is related to a common approach of specifying a joint probability model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ and assigning different weights to the posterior $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and the marginal probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ during the training [McCallum, Pal, Druck, and Wang (2006); Lasserre, Bishop, and Minka (2006)]. However, these models are limited due to the two following requirements.

- Firstly, the tractability of the computation and optimization of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ has to be provided.
- Secondly, the model parameters of both the discriminative and generative model have to be shared, which limits the flexibility.

These requirements make it difficult to incorporate complex models, including Deep Neural Networks (DNNs), into HDGMs.

3.1 Framework of Hybrid Discriminative-Generative Models

The approach proposed in [Kuleshov and Ermon (2017)] does avoid the latter requirement of sharing the model parameters. Instead of sharing the parameters $\boldsymbol{\theta}$, the coupling of both models is done by sharing latent variables $\mathbf{z} \in \mathcal{Z}$ that are introduced into the joint probability model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The latent variables $\mathbf{z}$ can be considered as a high-level representation, and sharing $\mathbf{z}$ across both

the discriminative and generative model can lead to improvements in accuracy [Kuleshov and Ermon (2017)].

Consider the joint probability model

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \approx p^*(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

over variables $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. The joint probability model can be factorized by

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \ ,$$

where $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})$ is the discriminative and $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ the generative component. Note that $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ is not only a generative model, but a latent variable model, as described in Section 2.2. We employ multi-conditional learning to minimize the objective function

$$\alpha L_{\mathrm{D}} \left[p^*(\mathbf{x}, \mathbf{y}, \mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z}) \right] + \beta_{\mathrm{G}} L_{\mathrm{G}} \left[p^*(\mathbf{x}, \mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right] \ , \tag{3.1}$$

with $\alpha, \beta_{\mathrm{G}} > 0$. The two functionals L_{D} and L_{G} constitute the losses of both the discriminative and generative component and are weighted by α and β_{G}, respectively. Shifting weights between α and β_{G} yields an interpolation between the discriminative and generative approach, and thus HDGMs cannot be categorized by either approach. Notice, however, that in the special case of $\alpha = \beta_{\mathrm{G}}$, we obtain a purely generative model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

In the following, we take a closer look at the minimization of the multi-conditional objective in Eq. (3.1) and specify the functionals L_{D} and L_{G}. Starting with L_{G}, our goal is to obtain a model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ that approximates the true underlying, yet unknown distribution $p^*(\mathbf{x}, \mathbf{z})$. For that purpose, any approximation of the f-divergence [Nowozin, Cseke, and Tomioka (2016)] between $p^*(\mathbf{x}, \mathbf{z})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ can be employed, i.e.,

$$L_{\mathrm{G}} \left[p^*(\mathbf{x}, \mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right] = D_f(p^*(\mathbf{x}, \mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})) \, ,$$

see [Kuleshov and Ermon (2017)]. Concerning the discriminative component L_{D}, a suitable classification loss function $\ell_{\boldsymbol{\theta}} \colon \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$ can be chosen, such that

$$L_{\mathrm{D}} = \mathbb{E}_{p^*(\mathbf{x}, \mathbf{y})} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\ell_{\boldsymbol{\theta}}(\mathbf{y}, p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})) \right]$$

is minimized. Note, that the expectation over the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is incorporated, since the classification loss depends on the latent variables $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$.

Next, we further specify the model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and motivate the link between the losses L_{D} and L_{G} and the common learning approach of maximizing the

marginal log-likelihood

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \, \mathrm{d}\mathbf{z}$$

as derived in Eq. (2.3). According to [McCallum, Pal, Druck, and Wang (2006)], the interpolation between the discriminative and generative component is achieved by optimizing the multi-conditional log-likelihood

$$\log \int p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})^{\gamma} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \, \mathrm{d}\mathbf{z} \tag{3.2}$$

with $\gamma > 0$. Due to the intractability of Eq. (3.2), we utilize the Variational Inference (VI) method introduced in Section 2.3 to obtain a variational bound as follows,

$$\log \int p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})^{\gamma} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \, \mathrm{d}\mathbf{z} = \log \int \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})^{\gamma} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \, \mathrm{d}\mathbf{z} \tag{3.3}$$

$$\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\gamma \, \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \right] , \tag{3.4}$$

which we aim to maximize, see [Kuleshov and Ermon (2017)].

The maximization of the lower bound in inequality (3.4) can be considered to be a special case of the previously presented objective in Eq. (3.1). Suppose that the evaluation of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is given in closed form and the gradients are tractable. If we choose $\alpha = \gamma$, $\beta_{\mathrm{G}} = 1$ and utilize the negative log-likelihood as well as the Kullback-Leibler divergence (KL divergence) to approximate the loss functionals

$$L_{\mathrm{D}} = -\mathbb{E}_{p^*(\mathbf{x},\mathbf{y})} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z}) \right] \tag{3.5}$$

$$\approx -\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[p_{\boldsymbol{\theta}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{z}) \right] \tag{3.6}$$

and

$$L_{\mathrm{G}} = D_{\mathrm{KL}} \left(p^*(\mathbf{x}, \mathbf{z}) || p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right) \tag{3.7}$$

$$\approx -\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}) - q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) \right] , \tag{3.8}$$

we obtain the multi-conditional learning framework presented in inequality (3.4). Minimizing L_{D} and L_{G} encourages the model to achieve a high classification accuracy as well as to learn a proper latent space, which can be interpreted as the maximization of the variational bound in inequality (3.4). Additionally, the

joint optimization of both losses can be considered as a type of regularization, which has positive effects on the classification accuracy [Kuleshov and Ermon (2017)].

3.2 Deep Hybrid Models

Following the above framework of HDGMs, we instantiate the discriminative and generative components with a Residual Neural Network (ResNet) and a β-Variational Autoencoder (VAE), respectively. The architecture of ResNets also constitutes the backbone of our generative component [He, Zhang, Ren, and Sun (2016)].

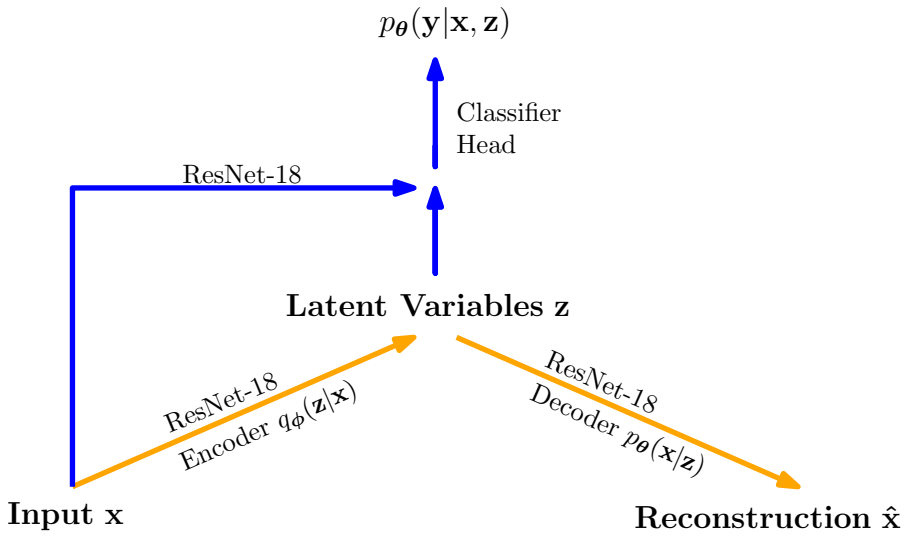


Fig. 3.1: A schematic view of the underlying architecture of the Deep Hybrid Model (DHM) inspired by [Kuleshov and Ermon (2017)]. The discriminative and generative components are highlighted in blue and orange, respectively. The latent variables $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are used to couple both the discriminative and generative model.

Figure 3.1 shows that both the encoder and the decoder of the β-VAE, highlighted in orange, are equipped with a ResNet-18. Utilizing ResNets is an extension to the model given in [Kuleshov and Ermon (2017)], which is modeled with simple Convolutional Neural Networks (CNNs). Furthermore, the discriminative component $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x},\mathbf{z})$, highlighted in blue, is modeled by a ResNet-18 and a fully connected classifier head. Note, that the input of the classifier head is a concatenation of the latent variables $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the output of the discriminative ResNet-18. Thus, the latent variables contribute to both model

components, leading to a hybrid approach. Differentiating the classification loss in Eq. (3.6) with respect to the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ leads to backpropagation through the sampling process of the latent variables. As previously discussed, the reparametrization trick must be utilized here.

Interestingly, the number of parameters that contribute to the classification is independent of β_{G}. Since the true underlying distributions are unknown, we approximate the loss functionals of the DHM by

$$\begin{aligned}
&\alpha L_{\mathrm{D}}\left[p^*(\mathbf{x},\mathbf{y},\mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x},\mathbf{z})\right] + \beta_{\mathrm{G}} L_{\mathrm{G}}\left[p^*(\mathbf{x},\mathbf{z}), p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\right] \\
&\quad \approx -\frac{\alpha}{n}\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[p_{\boldsymbol{\theta}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)},\mathbf{z})\right] \\
&\qquad -\frac{\beta_{\mathrm{G}}}{n}\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)},\mathbf{z}) - q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})\right] ,
\end{aligned}$$

where $\mathbf{x}^{(i)} \in \mathcal{X}$ and $\mathbf{y}^{(i)} \in \mathcal{Y}$. If we choose $\beta_{\mathrm{G}} = 0$, we obtain a purely discriminative model. However, the encoder of the β-VAE is incorporated in the discriminative model regardless of β_{G}. Thus, the number of parameters that contribute to the classification is constant for all $\beta_{\mathrm{G}} \geq 0$. This encourages us to choose the purely discriminative model with $\beta_{\mathrm{G}} = 0$ as one baseline among others for numerical experiments.

4 Robustness Metrics

This chapter focuses on metrics that we used to examine our models with respect to robustness. In addition to test accuracy, three other metrics are considered that contribute to the evaluation of robustness. First, the Expected Calibration Error (ECE) is discussed, followed by the binary classification problem associated with Out-Of-Distribution (OOD) detection. And finally, the adversarial robustness in the context of the Fast Gradient Sign Method (FGSM) is introduced and discussed.

4.1 Expected Calibration Error

In the paradigm of supervised learning, Neural Networks (NNs) provide predicted probability estimates associated with their decision. These probability estimates indicate the confidence of the model's decision. Although the accuracy of NNs has improved over the years, the predicted probability estimates diverged from the true correctness likelihood [Guo, Pleiss, Sun, and Weinberger (2017)]. Modern NN architectures tend to assign higher probability estimates, which results in overconfident models. The ability to provide approximately correct probability estimates is necessary in cases such as medical diagnosis and self-driving cars, to name just a few examples.

A measure to analyze a discriminative model with respect to this ability is given by the Expected Calibration Error (ECE) [Naeini, Cooper, and Hauskrecht (2015)]. Predicted probability estimates, that are close to the true correctness likelihood, result in a lower ECE. Thus, models with a lower ECE are considered to be better calibrated. On the other hand, if the model shows a higher ECE, the model is considered to be worse calibrated.

Again, let $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$ denote the dataset and let the corresponding classes be denoted by $\mathcal{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^n\}$. Consider the approximated posterior

$$p_{\boldsymbol{\theta}}(y_i|\mathbf{x}) = \frac{\exp(\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i])}{\sum_i \exp(\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i])},$$

of a datapoint $\mathbf{x} \in \mathcal{X}$, where $\Phi_{\boldsymbol{\theta}} \colon \mathcal{X} \to \mathbb{R}^C$ is parametrized by a NN. Here, $\Phi_{\boldsymbol{\theta}}(\mathbf{x})[y_i]$ for $i \in \{1, \ldots, C\}$ denotes the logit corresponding to the i^{th} class. Among all classes, the entry with the highest estimated probability results in the predicted class $\hat{y}_i$, whereas the value itself is the predicted probability estimate. Hence, the classification provides two entities, namely the predicted class and the predicted probability estimate.

Let $K > 0$ denote the number of equally-spaced interval bins. Moreover, for $k \in \mathbb{N}$ and $1 \leq k \leq K$, each interval bin is given by

$$\mathcal{I}_k = \left(\frac{k-1}{K}, \frac{k}{K} \right] .$$

The set of indices of the datapoints of which the predicted probability estimates fall into the k^{th} interval bin is denoted by B_k. Now, the model's accuracy can be calculated for each bin B_k by

$$\text{acc}(B_k) = \frac{1}{|B_k|} \sum_{i \in B_k} \mathbf{1}(y_i = \hat{y}_i) ,$$

where y_i and $\hat{y}_i$ denote the true and predicted class for the i^{th} datapoint [Guo, Pleiss, Sun, and Weinberger (2017)]. Let the predicted probability estimate of the i^{th} datapoint be denoted by $\hat{p}_i$. Similarly, the average confidence of each bin is given by

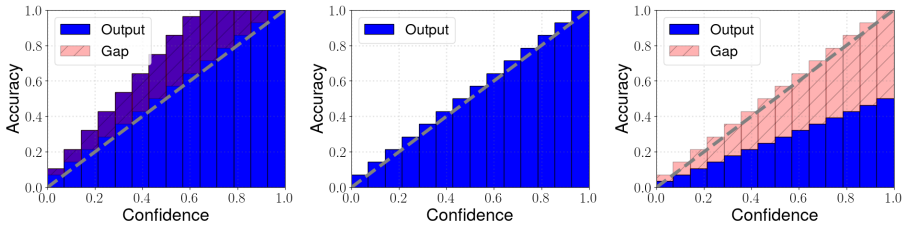
$$\text{conf}(B_k) = \frac{1}{|B_k|} \sum_{i \in B_k} \hat{p}_i .$$

The model's ECE is then defined by

$$\text{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{n} \left| \text{acc}(B_k) - \text{conf}(B_k) \right| , \tag{4.1}$$

where n denotes the cardinality of $\mathcal{X}$. As one can see in Eq. (4.1), the calibration gaps $|\text{acc}(B_k) - \text{conf}(B_k)|$ of each bin determine the model's ECE. In our numerical experiments, we chose $K = 15$.

A perfectly calibrated model predicts probability estimates that are consistent with the actual measured accuracy of the model, i.e., all gaps are equal to zero. For illustration purposes, take a look at Figure 4.1. In Figure 4.1c, an example of an overconfident model is given whose confidence values are higher than the actual accuracy, resulting in confidence gaps. In contrast, the perfectly calibrated model in Figure 4.1b shows alignment with the dashed diagonal line. And in Figure 4.1a, the model predicted lower probability estimates than the actual accuracy achieved and is therefore categorized as underconfident.



(a) Underconfident Model **(b)** Perfectly Calibrated Model **(c)** Overconfident Model

Fig. 4.1: Examples of histograms of an underconfident (a), a perfectly calibrated (b), and an overconfident (c) model.

As mentioned earlier, most modern architectures in deep learning seem to be overconfident, like the illustrated example in Figure 4.1c.

4.2 Out-of-Distribution Detection

Many machine learning models rely on the assumption, that the test data is sampled from the same distribution as the training data. Based on this assumption, these models are not capable of detecting whether a datapoint might be sampled from another probability distribution. In the following, we refer to datapoints drawn from a different probability distribution than the training data as OOD data.

For example, a Deep Neural Network (DNN) might be trained to distinguish images of two different classes. However, if an image $\bar{\mathbf{x}} \sim p_{\text{ood}}(\bar{\mathbf{x}})$ does not belong to either class, the DNN will silently assign one of the two known classes to $\bar{\mathbf{x}}$. The model will not only predict incorrect classes, but it often does so with high confidence [Nguyen, Yosinski, and Clune (2015)]. This particular behavior can be harmful, when employed in real-world tasks. Thus, estimating whether a datapoint is OOD is considered to be of great concern for artificial intelligence safety [Amodei *et al.* (2016)].

OOD detection is a binary classification problem, in which the model must provide a score

$$s_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}\,.$$

Our goal is to obtain higher scores for in-distribution samples than for OOD samples. If in-distribution samples achieve higher scores, the model is able to

distinguish the two classes by choosing a threshold that marks the decision boundary. As a rule, the two classes are categorized as positive and negative. In our case, the in-distribution samples constitute the set of positive examples, whereas the OOD samples constitute the set of negative examples. To provide an overview of the model's decisions, the decisions are often represented in a confusion matrix.

A confusion matrix comprises four categories, namely True Positives (TPs) that were correctly classified as positive, False Positives (FPs) that were incorrectly classified as positive, while True Negatives (TNs) refer to data correctly classified as negative and False Negatives (FNs) refer to incorrectly classified as negative. Confusion matrices can be used to construct Receiver Operating Characteristic curves (ROC curves). The ROC curve illustrates the ability of a binary classifier to distinguish the two classes for different thresholds. To obtain a threshold-independent performance metric for OOD detection, the Area Under the Receiver-Operating Curve (AUROC)[1] is usually used as a score to compare different models [Davis and Goadrich (2006); Hendrycks and Gimpel (2016)].

OOD Detection in Supervised Learning. In supervised learning, discriminative models are trained to maximize the conditional log-likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ of the given training data $\mathbf{x} \in \mathcal{X}$. On this basis, it is reasonable to assume that the conditional log-likelihood for in-distribution samples are higher than for OOD samples [Hendrycks and Gimpel (2016)]. Therefore, we employ the predicted conditional probability estimates as our scores $s_{\boldsymbol{\theta}}(\mathbf{x})$ to distinguish between in-distribution and OOD samples.

OOD Detection in Unsupervised Learning. In unsupervised learning, on the other side, generative models are often trained to explicitly maximize the log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. Similar to the above case, it can be assumed that the log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ is higher for in-distribution samples than for OOD samples [Bishop (1994)]. Note, that there are generative models that do not have explicit access to the log-likelihood, but implicit access, e.g., Generative Adversarial Nets (GANs) [Goodfellow *et al.* (2014)] or Markov chain models [Goodfellow (2016)]. However, we have included Variational Autoencoders (VAEs) and thus have access to the approximation of $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. Consequently, the approximation of $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ is used to distinguish between in-distribution and OOD samples.

[1] [Hendrycks and Gimpel (2016)] proposed a "debatable and imprecise" interpretation of the AUROC score as follows: '90%–100%: Excellent, 80%–90%: Good, 70%–80%: Fair, 60%–70%: Poor, 50%–60%: Fail.'

As stated above, intuitively, models should assign low likelihood values to OOD samples. However, Nalisnick *et al.* first showed that that deep generative models such as VAEs can assign higher probability values to OOD data [Nalisnick *et al.* (2018)], especially at near-OOD detection tasks. Near-OOD detection is more challenging because the classes of the OOD dataset are semantically closer to the classes of the in-distribution dataset than those of far-OOD datasets. This phenomenon was subsequently confirmed and further investigated in [Choi, Jang, and Alemi (2018); Nalisnick, Matsukawa, Teh, and Lakshminarayanan (2019)].

4.3 Adversarial Robustness

Machine learning models have shown vulnerability to adversarial attacks that are barely visible to the human eye [Szegedy *et al.* (2013)]. Adversarial attacks are inputs to discriminative models that are deliberately intended to negatively influence the classification. While there are various approaches investigating the adversarial robustness, we focus on a common approach called Fast Gradient Sign Method (FGSM) [Goodfellow, Shlens, and Szegedy (2015)].

In the context of image recognition tasks, let $\mathbf{x}^{(i)} \in \mathcal{X}$ denote the image and $\mathbf{y}^{(i)} \in \mathcal{Y}$ the corresponding true label. Furthermore, let the loss of the classification, used to train the network, be denoted by $\ell_{\boldsymbol{\theta}} \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Utilizing FGSM, adversarial examples can be generated by a small modification to the original image $\mathbf{x}^{(i)}$ as follows:

1. We first calculate the gradient of the loss with respect to the input image $\nabla_{\mathbf{x}}(\ell_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}))$.

2. Next, we apply a small modification into the direction of the gradient to obtain an adversarial example

$$\mathbf{x}^{\text{adv}} = \mathbf{x}^{(i)} + \epsilon \operatorname{sign}(\nabla_{\mathbf{x}}(\ell_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}))\,,$$

   with $\epsilon > 0$, but small.

The choice of ϵ determines the magnitude of the modification. Often, images are encoded in 8 bits per pixel, such that all information below the precision of 8 bits is discarded. Thus, an ϵ below that precision can be considered to be a small modification to the original input image [Goodfellow, Shlens, and Szegedy (2015)]. Due to the access to the parameters of the model and the gradients, FGSM is considered a white-box attack.

Recall, that supervised NNs are optimized to minimize the negative log-likelihood, i.e.,

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p^*(\mathbf{x},\mathbf{y})}[\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})]\,. \tag{4.2}$$

From an optimization perspective, discriminative models can be trained to be more robust against adversarial attack such as FGSM. Replacing the negative log-likelihood by the corresponding loss function and adding an inner maximization to Eq. (4.2), we obtain

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{p^*(\mathbf{x},\mathbf{y})} \left[\max_{\boldsymbol{\delta}\in\mathcal{S}} \ell_{\boldsymbol{\theta}}(\mathbf{x}+\boldsymbol{\delta},\mathbf{y})\right]\,, \tag{4.3}$$

where $\boldsymbol{\delta} \in \mathcal{S}$ describes the perturbation that comes from a set of perturbations $\mathcal{S} \subset \mathbb{R}^d$ determined by the adversarial attack method used [Madry *et al.* (2017)]. Such strategies are primarily used to improve adversarial robustness. However, in this thesis, we do not consider robust optimization strategies as in Eq. (4.3), but instead investigate the adversarial robustness of our models optimized with respect to the losses presented in Chapter 2 and Chapter 3.

5 Numerical Experiments

This section deals with numerical experiments conducted on various image recognition datasets to analyze the models with respect to Expected Calibration Error (ECE), Out-Of-Distribution (OOD) detection, and adversarial robustness. More precisely, the experiments contribute to the evaluation of the following questions:

- Does the interpolation between discriminative and generative approaches of Deep Hybrid Models (DHMs) yield any increase in accuracy or robustness in terms of ECE, OOD detection, and adversarial accuracy?
- Do DHMs yield higher classification accuracies over state-of-the-art supervised deep learning models such as Residual Neural Networks (ResNets)?
- Do DHMs yield higher OOD detection scores via $\log p(\mathbf{y}|\mathbf{x})$ compared to supervised ResNets?
- Do DHMs yield higher OOD detection scores via $\log p(\mathbf{x})$ compared to state-of-the-art generative models such as unsupervised β-Variational Autoencoders (VAEs)?
- Do DHMs yield a lower ECE compared to supervised ResNets?
- Do DHMs yield higher adversarial accuracies compared to supervised ResNets?

The open-source machine learning framework PyTorch has been utilized to perform the numerical experiments [Paszke *et al.* (2019)]. Due to the size of the models, the numerical experiments were required to be conducted on Graphics Processing Units (GPUs). To accelerate the execution, the Fraunhofer Heinrich-Hertz-Institute[1] kindly provided their GPU-Cluster, enabling us to conduct multiple experiments at the same time on Nvidia's Tesla V100 GPUs. In addition, to realize better reproducibility and limit the nondeterministic behavior, all experiments were repeated 10 times and averaged with respect to the performance metrics. Thus, in the following, we always refer to the average and standard deviation if not other stated. Moreover, models with the

[1] https://www.hhi.fraunhofer.de/en/index.html

exact same setting have been set to the same random seed to further reduce the randomness.

Model Selection. Principally, the evaluation of Neural Networks (NNs) includes training and validation of the models. The entire learning process is split into multiple epochs. An epoch refers to one cycle through the entire dataset. Our experiments were conducted, such that the models' parameters were saved separately after each epoch. Consequently, for each experiment, we had access to a set of candidates to choose from for further analysis. To allow a fair comparison, we selected the candidate with the highest accuracy with respect to the validation datasets, as we consider accuracy to be decisive. The parameters of this candidate were used for a final evaluation and comparison of the performance metrics on the test datasets.

5.1 Experimental Setup

In the following, we describe the basic design of the numerical experiments. Firstly, we present the datasets and their division into training, validation, and testing dataset. In addition, the standard data augmentation scheme applied to the datasets is demonstrated. Finally, this section gives an overview of the models and the configurations of the optimizer used to perform the experiments.

5.1.1 Datasets

The numerical experiments were conducted on three widely used and publicly available datasets for image recognition tasks, namely Street View House Numbers (SVHN), CIFAR-10, and CIFAR-100 [Netzer *et al.* (2011); Krizhevsky (2009)]. Note, however, that SVHN and CIFAR-10 constitute the in-distribution datasets, while the OOD samples are taken from CIFAR-100, see Table 5.1. In other words, we train our models on SVHN and CIFAR-10 and evaluate, in the context of OOD detection, whether the model is capable of distinguishing the trained datasets from the samples of CIFAR-100. The exact configurations of the datasets can be taken from Table 5.1.

The SVHN dataset contains colored real-world images sized at 32×32 pixels. The images consist of digits from house numbers that are taken from *Google Street View*. The dataset incorporates ten classes, one for each digit, and is originally separated into two subsets, namely 73257 images for training and

Dataset	#Classes	#Training samples	#Validation samples	#Test samples	In-Distribution/ Out-of-Distribution
SVHN	10	58 605	14 652	26 032	In-Distribution
CIFAR-10	10	42 000	8000	10 000	In-Distribution
CIFAR-100	100	0	0	60 000	Out-of-Distribution

Tab. 5.1: Datasets used to perform the numerical experiments.

26032 images for testing. We further divide the training dataset into 58605 images for training and 14652 images for validation. The validation dataset is used for model selection, i.e., the parameters with the highest accuracy among all epochs concerning the validation dataset are selected to conduct a final test on the test dataset. Examples of the dataset can be seen in Figure 5.1.

Fig. 5.1: Examples of the SVHN dataset taken from [Netzer *et al.* (2011)].

The CIFAR-10 dataset consists of 60000 colored images, each with 32×32 pixels. For each of the ten mutually exclusive classes, 6000 images are provided. We have split the training dataset such that 42000 images are for training and 8000 images for validation and model selection. The remaining 10000 images are used for testing.

The CIFAR-100 dataset, as the name suggests, consists of real images of 100 classes and is commonly used for image recognition benchmark tests. Each class contains 600 images sized at 32×32 pixels. We employ the CIFAR-100 dataset to obtain OOD samples for the OOD detection task. Note, that there are no overlapping classes with the CIFAR-10 dataset. Nevertheless, the task to distinguish the two non-overlapping datasets is considered to be difficult [Grathwohl *et al.* (2019)]. Surprisingly, recent studies show that deep generative models can assign higher likelihood values to OOD data, albeit being optimized

to maximize the log probability of in-distribution data [Nalisnick *et al.* (2018); Choi, Jang, and Alemi (2018)].

Data Augmentation. Typically, data augmentation is applied to datasets to achieve generalization. The above training datasets were augmented by the same standard scheme, in which the images are normalized, zero-padded with 4 pixels on each side, randomly mirrored with a probability of 0.5 and randomly cropped to obtain 32×32 images [Romero *et al.* (2014); Lee *et al.* (2015); Huang *et al.* (2017)].

5.1.2 Network Architectures and Training Configuration

We divide our models into three categories, namely discriminative and generative models, and DHMs. Since DHMs can handle both discriminative and generative tasks, we choose the discriminative ResNet-18 and generative β-VAE in Table 5.2 as the basis for the comparison with DHMs. Moreover, the purely discriminative setting of DHMs ($\alpha = 1$ and $\beta_{\mathrm{G}} = 0$), which is listed in Table 5.2 as discriminative DHM, forms another baseline. In this way, we can assess whether the regularization effect of the joint optimization ($\alpha > 0$, $\beta_{\mathrm{G}} > 0$) is an advantage over purely discriminative optimizations ($\alpha = 1$ and $\beta_{\mathrm{G}} = 0$) for DHMs.

Model	Approach	#Parameters	Epochs	Batch size	Learning rate (t $\widehat{=}$ training epoch)
ResNet-18	Discriminative	11.4M	200	512	$\eta_t = \begin{cases} 10^{-3}, & t \in [0, 50]\,, \\ 10^{-4}, & t \in [51, 100]\,, \\ 10^{-5}, & t \in [101, 150] \\ 10^{-6}, & t \in [151, 200] \end{cases}$
Disc. DHM	Discriminative	22.6M	200	512	
β-VAE	Generative	20.6M	200	512	
DHM	Hybrid	31M	200	512	

Tab. 5.2: An overview of the models and their configuration for training and optimization.

As mentioned before, the ResNet-18 forms the backbone of our DHMs. This is also true for the discriminative and generative models in Table 5.2. The architecture of DHMs can be considered as a combination of the discriminative ResNet-18 and the generative β-VAE from Table 5.2. More specifically, the latent variables of the β-VAE are concatenated with the output of the discriminative ResNet-18 and fed into a classifier header. The classifier head consists of two fully connected layers with 200k learnable parameters and a dropout probability of 0.5 (second to last layer) and 0.25 (last layer) [Srivastava *et al.* (2014)]. For the generative and hybrid approaches, we set $\beta = 0.1$ as this

gave us the best results, albeit often $\beta > 1$ is the typical choice. It should be emphasized that although the models have the same basic framework, the parameters are optimized based on different loss functions.

For the optimization, the Adam algorithm was used, which has become the standard optimization method for deep learning [Kingma and Ba (2015); Ruder (2016); Gregor *et al.* (2015); Xu *et al.* (2015)]. The initial value of the learning rate was set to 10^{-3} and decreased by a factor of 0.1 every 50 epochs. In addition, the exponential decay rate for the first and second momentum estimates were set to 0.9 and 0.999, respectively, as suggested by [Kingma and Ba (2015)]. More information about the configurations can be seen in Table 5.2.

To see to what extent the interpolation between discriminative and generative approaches offer advantages, we examine the effect of the hyperparameters α and β_{G}. More precisely, we set $\beta_{\mathrm{G}} = 1$ and $\alpha \in \{10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$ and train each configuration 10 times for 200 epochs on SVHN and CIFAR-10 from scratch. In the following, we use the labels DHM-0 to DHM-5 to highlight the hyperparameters that were used to train the hybrid models. For example, DHM-0 represents a DHM trained with $\alpha = 10^0$, while DHM-5 represents a DHM trained with $\alpha = 10^5$. Higher values for α mean that more emphasis was placed on the discriminative approach during training. Given the architecture of DHMs, we have access to both likelihoods, $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and $\log p_{\boldsymbol{\theta}}(\mathbf{x})$, and compare the accuracy and resulting robustness metrics to their respective baseline.

5.2 Results

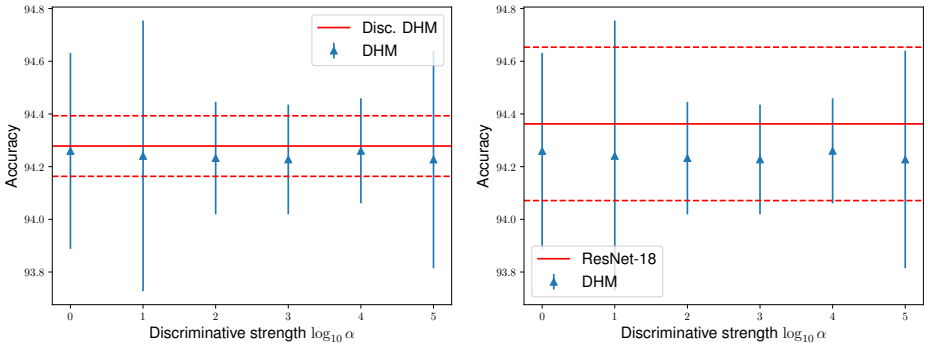
This section presents the results of our numerical experiments conducted on SVHN and CIFAR-10, and is divided into subsections for each of the metrics studied, namely classification accuracy, OOD detection, ECE, and robustness to adversarial attacks. Note, that all results are averaged over 10 runs, with error bars corresponding to $\pm 2\sigma$, where σ denotes the standard deviation, if not stated otherwise. Moreover, a conclusion for each experiment is provided at the end of each subsection.

5.2.1 Classification Accuracy

In this subsection, we report the test accuracy obtained through supervised learning with our discriminative baselines, i.e., discriminative DHM and ResNet-18, and our DHMs optimized using the multi-conditional objective. It should be emphasized that the number of parameters of the DHMs that contribute to the classification task are constant for all $\beta_{\mathrm{G}} \geq 0$, although the total number of parameters varies according to Table 5.2. This is due to the underlying architecture of DHMs. More precisely, the additional parameters of the DHMs represent the parameters of the decoder that are not used for classification. Nevertheless, the parameters that are used for classification are optimized based on different loss functions and therefore result in different solutions.

Results on SVHN. First, we examine the results in Figure 5.2 obtained by the experiments with the SVHN dataset. In Figure 5.2a, the average accuracy achieved by the purely discriminative setting of the DHM, the solid red line (baseline), is higher than that of the DHMs optimized according to the multi-conditional objective. This is in contrast to the results reported in [Kuleshov and Ermon (2017)], where a small gain in accuracy was obtained. However, the differences in the average accuracy of our DHMs from our discriminant baseline are so small that they are within one standard deviation of the discriminant baseline. Moreover, the standard deviation of the average accuracy of our DHMs is two to four times greater than that of the discriminative DHM, showing that some runs DHMs achieved higher accuracies compared to the baseline. Nevertheless, we consider the average accuracy, which is slightly below the baseline, to be decisive.

Next, we compare the results of our DHMs with the ResNet-18 baseline in Figure 5.2b. The ResNet-18 was trained for 200 epochs each on the SVHN and



(a) Test accuracy on SVHN with discriminative DHM as the baseline.

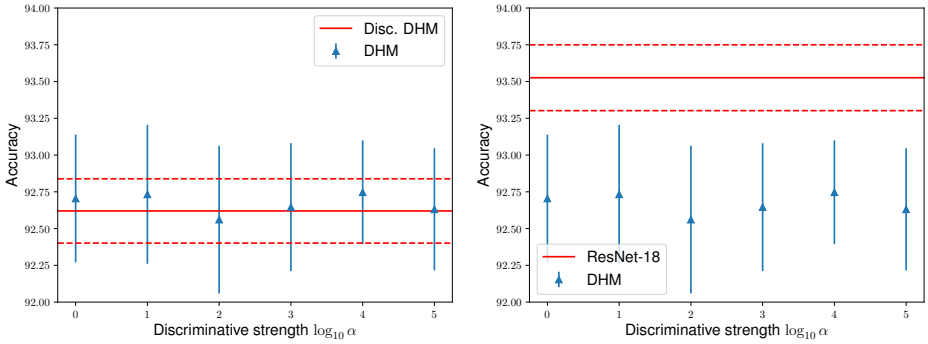
(b) Test accuracy on SVHN with discriminative ResNet-18 as the baseline.

Fig. 5.2: Mean and two times standard deviation of test accuracy values of DHMs with different configurations of weights α and β_G on SVHN compared to the discriminative baselines, i.e., the discriminative DHM (left) and ResNet-18 (right). The solid red lines, which serve as the baselines, show the averaged results of the discriminative DHM (i.e. $\alpha = 1$, $\beta_G = 0$) and the ResNet-18. The corresponding error bars of the baselines are shown in dashed lines.

CIFAR-10 datasets with the configurations listed in Table 5.2. In Figure 5.2b, the results for the SVHN dataset are shown. Again, the DHMs achieved lower average test accuracy values on SVHN compared to the ResNet-18 baseline, regardless of the discriminative strength $\log_{10}\alpha$. In addition, the ResNet-18 baseline provides the highest test accuracy with an average accuracy of $94.36\% \pm 0.14$ among all models studied on SVHN, see Table 5.3. This means that neither the DHM with purely discriminative settings nor the DHMs with the multi-conditional objective could achieve higher test accuracies, even though the DHM has almost twice as many parameters as the ResNet-18. However, it should be noted that the differences in the average test accuracy of our DHMs compared to the ResNet-18 are less than 0.2 percentage points.

Results on CIFAR-10. The results with respect to another well-studied benchmark dataset, namely CIFAR-10, can be seen in Figure 5.3. To start with, we notice a difference compared to the experiments conducted with the SVHN dataset. In Figure 5.3a, for example, one can see, that the average test accuracy of five of our six DHMs exceeds the average test accuracy of the purely discriminative setting. The highest accuracy among these models was achieved by the DHM-4, which is 0.13 percentage points above baseline, see Tab. 5.3 on the facing page. Only the DHM-2 trained with $\alpha = 10^2$ achieved a lower accuracy value than the baseline model. Again, the average accuracy values shown in

Figure 5.3a are within one standard deviation of each other, illustrating the small differences.



(a) Test accuracy on CIFAR-10 with discriminative DHM as the baseline.

(b) Test accuracy on CIFAR-10 with discriminative ResNet-18 as the baseline.

Fig. 5.3: Mean and two times standard deviation of test accuracies of DHMs on CIFAR-10 compared to discriminative DHM (left) and ResNet-18 (right).

In contrast to the small differences seen so far, there is a fairly large difference between the DHMs and the ResNet-18 baseline in Figure 5.3b. The ResNet-18 outperforms all other examined models, with an average accuracy of 93.52% on CIFAR-10. In addition, the average accuracy values of the DHMs do not overlap with the error bars of the ResNet-18, indicating a significant difference. The exact test accuracies of all models can be seen in Table 5.3.

	Accuracy in %	
Model	SVHN	CIFAR-10
ResNet-18	$\mathbf{94.36 \pm 0.14}$	$\mathbf{93.52 \pm 0.11}$
Disc. DHM	94.29 ± 0.06	92.62 ± 0.11
DHM-0	94.26 ± 0.37	92.71 ± 0.43
DHM-1	94.24 ± 0.51	92.73 ± 0.47
DHM-2	94.23 ± 0.21	92.56 ± 0.50
DHM-3	94.23 ± 0.21	92.65 ± 0.43
DHM-4	94.26 ± 0.20	92.75 ± 0.35
DHM-5	94.22 ± 0.41	92.63 ± 0.41

Tab. 5.3: Test accuracy values on SVHN and CIFAR-10.

Conclusion. Overall, the ResNet-18 achieved the highest accuracy for both datasets. Comparing the DHMs against each other, we find that the results are quite robust with respect to the hyperparameter α, suggesting that tuning of α is less necessary. Nevertheless, the results are very close to each other, especially

compared to the discriminative DHM baseline, since the error bars strongly overlap. Moreover, it is known that NNs are sensitive to hyperparameters of the optimizer, such as the learning rate. It is likely that numerical experiments with configurations different from ours could lead to different findings. In our experimental setup, we limited the number of runs for each model to ten, selected the models with the highest validation accuracies, and performed final experiments on the test datasets. Considering the standard deviation of the results, the limit of ten runs might be too inaccurate. However, the experiments are computationally expensive due to the size of the models. On the other side, the significant difference with respect to the ResNet-18 baseline shows that DHMs do not provide an accuracy advantage over modern NN architectures.

One final note: Given our inconsistent results comparing DHMs and their purely discriminative setting, it is difficult to draw a conclusion from our experimental setup about whether hybrid modeling via latent variable coupling has an advantage in terms of accuracy.

5.2.2 Out-of-Distribution Detection

This subsection presents the results obtained by the above discriminative, generative and hybrid models concerning the OOD detection task. First, we study the models with respect to the obtained Area Under the Receiver-Operating Curve (AUROC) scores. For this purpose, the CIFAR-100 dataset mentioned above represents the OOD dataset, while SVHN and CIFAR-10 constitute the in-distribution datasets, respectively. To be precise, the models were trained to learn either the unconditional or the conditional probability distribution, $p_{\boldsymbol{\theta}}(\mathbf{x})$ or $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, from the SVHN or CIFAR-10 dataset, depending on whether supervised or self-supervised learning was used. The models were then tested to see if they can distinguish the in-distribution samples from the OOD samples based on the probability distributions.

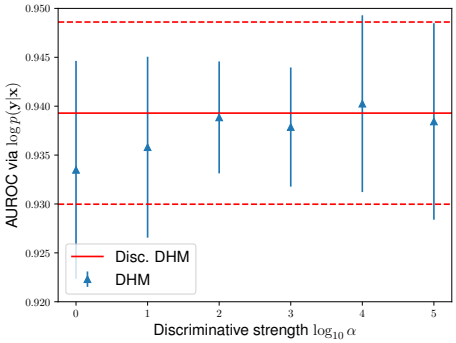
In addition, the normalized density histograms are plotted to examine whether the learned probability distributions are overlapping. If a model assigns higher probability values for in-distribution samples than for OOD samples, one could set a threshold and easily solve the binary classification problem. Intuitively, models should assign low likelihood values to OOD samples. However, Nalisnick *et al.* first showed that that deep generative models such as VAEs can assign higher probability values to OOD data [Nalisnick *et al.* (2018)]. This phenomenon was subsequently confirmed and further investigated in [Choi, Jang, and Alemi (2018); Nalisnick, Matsukawa, Teh, and Lakshminarayanan (2019)].

Results for SVHN vs. CIFAR-100. First, we examine the results in Figure 5.4 obtained with respect to the *far*-OOD detection task, namely SVHN vs. CIFAR-100. This task is considered less difficult than the *near*-OOD detection task. All reported values are AUROC values calculated using the conditional likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ in Figure 5.4a and Figure 5.4b or unconditional likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ in Figure 5.4c. To start with, we examine the results in Figure 5.4a, where the DHM is compared to its purely discriminative setting. We notice that the results are again very close to each other, i.e., the average values are within the standard deviation of the baseline. The model that performed best based on the average values is the DHM-4 with an AUROC value of 0.94 ± 0.01. At the same time, this is the only configuration of the DHM that could exceed the baseline. However, the baseline achieved 0.93 ± 0.01, which is only 0.01 percentage points below the DHM-4. Moreover, the differences between the DHMs trained with different discriminative strengths are rather small, suggesting that α does not need to be tuned much.

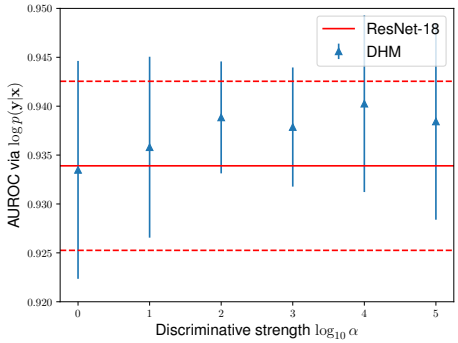
In Figure 5.4b the comparison to the ResNet-18 baseline is shown. Five out of six DHMs achieved higher average values via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ than the ResNet-18 baseline. It should be emphasized, though, that the error bars are overlapping, which highlights the inaccuracy of the experimental setup. The only configuration of the DHM that did not exceed the ResNet-18 baseline is the DHM-0, which can be considered a purely generative model due to the equal weighting $\alpha = \beta_{\mathrm{G}} = 1$ mentioned above.

In Figure 5.4c, we compare the AUROC values obtained by the DHMs and the β-VAE baseline via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. Note that our DHMs have both $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ and $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ available, and thus are the only models in our experimental setup that provide values for both comparisons. Interestingly, the AUROC values via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ are generally much higher than via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, i.e., suggesting that detecting OOD samples via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ works better for our datasets. Some models, such as the β-VAE, DHM-0, DHM-1, DHM-2, approximately achieved an AUROC value greater than 0.996 ± 0.00. This shows that the models were able to perfectly distinguish almost all OOD samples, considering that an AUROC value of one is the maximum achievable value.

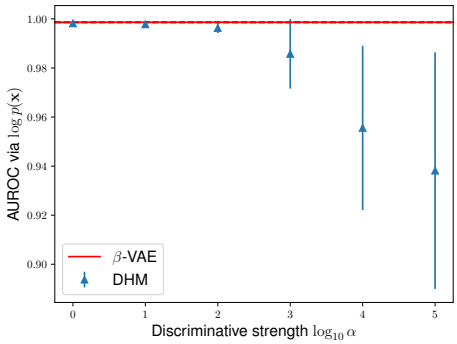
The most striking phenomenon we noticed is that discriminative strength has a great influence here. In general, higher values of α resulted in less successful OOD detection via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ according to AUROC values of the DHMs. The models with the highest α not only perform the worst in terms of AUROC values, but also have a higher standard deviation. This behavior is not surprising, since learning $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ is weighted less heavily when α is increased. Finally, the models with the highest AUROC value are both the DHM-0 and



(a) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ with discriminative DHM as the baseline.



(b) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ with ResNet-18 as the baseline.



(c) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ with generative β-VAE as the baseline.

Fig. 5.4: Results regarding the SVHN vs. CIFAR-100 OOD detection task. DHMs are compared with baselines, i.e., discriminative DHM, ResNet-18, and β-VAE. All values are AUROC values averaged over ten runs. The error bars correspond to two times the standard deviation. In (a) and (b), the scores were calculated using the conditional log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, whereas in (c) $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ was used.

the β-VAE with a value of 0.998 ± 0.00. In Table 5.4, column 2 and 3, one can see an overview of all AUROC values achieved in the experiments.

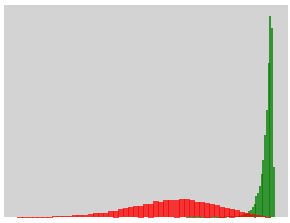
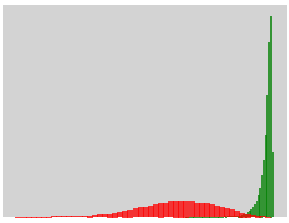
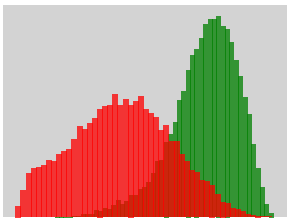
Next, we plot the normalized histograms of the $p_{\boldsymbol{\theta}}(\mathbf{x})$ values obtained by the β-VAE, DHM-0 and DHM-5. We chose DHM-0 and DHM-5 because they have the largest difference in weights with respect to the discriminative strength. In addition, the former scored the best and the latter the worst according to the AUROC values. As mentioned earlier, this is not surprising since the classification task is weighted more heavily as α increases, while the goal of the generative approach is neglected.

In Table 5.5 one can see the histograms for the OOD detection. While the β-

Model	SVHN vs. CIFAR-100		CIFAR-10 vs. CIFAR-100	
	$\log p_{\boldsymbol{\theta}}(\mathbf{y}\|\mathbf{x})$	$\log p_{\boldsymbol{\theta}}(\mathbf{x})$	$\log p_{\boldsymbol{\theta}}(\mathbf{y}\|\mathbf{x})$	$\log p_{\boldsymbol{\theta}}(\mathbf{x})$
ResNet-18	$.934 \pm .01$	n.a.	$.841 \pm .01$	n.a.
Disc. DHM	$.939 \pm .01$	n.a.	$.835 \pm .01$	n.a.
β-VAE	n.a.	$.998 \pm .00$	n.a.	$.272 \pm .00$
DHM-0	$.933 \pm .01$	$\mathbf{.998 \pm .00}$	$.834 \pm .01$	$.267 \pm .01$
DHM-1	$.936 \pm .01$	$.997 \pm .00$	$\mathbf{.843 \pm .01}$	$.257 \pm .00$
DHM-2	$.939 \pm .01$	$.996 \pm .00$	$.840 \pm .01$	$.259 \pm .02$
DHM-3	$.938 \pm .01$	$.985 \pm .01$	$.839 \pm .01$	$.287 \pm .01$
DHM-4	$\mathbf{.940 \pm .01}$	$.955 \pm .03$	$.837 \pm .01$	$.308 \pm .05$
DHM-5	$.938 \pm .01$	$.938 \pm .05$	$.835 \pm .01$	$\mathbf{.317 \pm .04}$

Tab. 5.4: AUROC values of the OOD detection tasks. Models trained on SVHN.

VAE and the DHM-0 clearly assigned higher $p_{\boldsymbol{\theta}}(\mathbf{x})$ values for the in-distribution samples, the DHM-5 shows weaknesses in the density estimation. Moreover, the overlapping histograms explain the poorer AUROC result of the DHM-5.

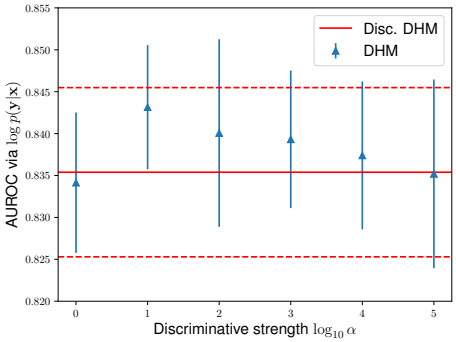
β-VAE	DHM-0	DHM-5
		

Tab. 5.5: Histograms for OOD detection. All models are trained on SVHN. Green corresponds to the score obtained with the in-distribution dataset SVHN and red corresponds to the score for the OOD dataset CIFAR-100.

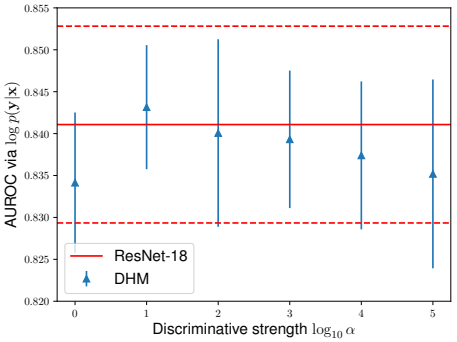
Results for CIFAR-10 vs. CIFAR-100. In the following, we present the results for the CIFAR-10 vs. CIFAR-100 OOD detection task, which is considered more difficult because the images are semantically closer than the previous datasets. Following the order above, we start with the AUROC values in Figure 5.5a, which are determined via the conditional probability $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$. In contrast to the SVHN vs. CIFAR-100 detection, we find that the DHMs trained with a multi-conditional target achieves slightly higher AUROC values for $10^1 \leq \alpha \leq 10^5$ compared to the purely discriminative baseline. The best model, the DHM-1, obtained a AUROC value of 0.843 ± 0.1, which is 0.08 ± 0.01

higher than the baseline. Nevertheless, the error bars overlap, as has been mentioned several times above.

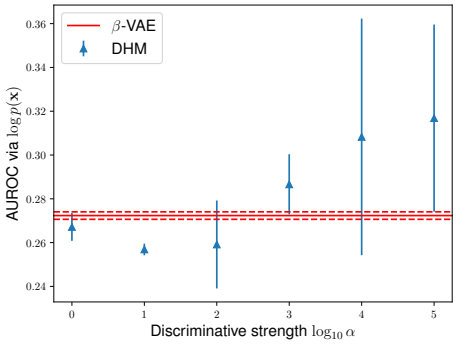
Comparing the AUROC values in Figure 5.5b, we find that the ResNet-18 baseline outperforms almost all DHMs, except for DHM-1, which is our best model. However, if we look at the plotted standard deviation, the differences are very small. Moreover, as expected, the AUROC values are generally smaller than those of the SVHN vs. CIFAR-100 detection, which is due to the more difficult task. The exact values can be taken from Tab. 5.4 on the previous page, column 4 and 5.



(a) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ with discriminative DHM as the baseline.



(b) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ with ResNet-18 as the baseline.



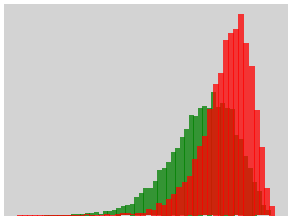
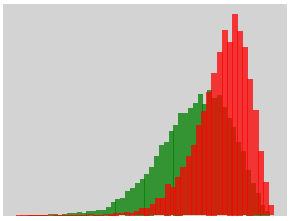
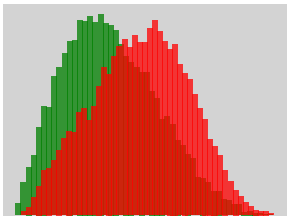
(c) AUROC via $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ with generative β-VAE as the baseline.

Fig. 5.5: CIFAR-10 vs. CIFAR-100 OOD detection results. All values are AUROC values averaged over ten runs. The error bars correspond to two times the standard deviation. In (a) and (b), the scores were calculated using the conditional log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, whereas in (c) $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ was used.

Now the results in Figure 5.5c are particularly salient. The AUROC values were calculated using the unconditional log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. As can be seen, all

the models examined achieved AUROC values of less than 0.5. This indicates that the models have assigned higher log-likelihood values to the OOD samples than to the in-distribution samples, as first reported for deep generative models in [Nalisnick *et al.* (2018)]. Considering that a random classifier would achieve an AUROC value of approximately 0.5, this is obviously not applicable in real world tasks. The DHMs were not able to avoid this seemingly paradoxical behavior, although the multi-conditional objective is different from purely generative models. However, we note that there is a clear tendency for DHMs with larger α to provide improvements in terms of AUROC values compared to the purely generative baseline.

To confirm that the models mostly assigned higher log-likelihoods to the OOD samples, take a look at Figure 5.6. There you can see the histograms obtained from β-VAE, DHM-0 and DHM-5. Qualitatively speaking, the histograms of the OOD dataset (in red) are further to the right, i.e., the datapoints were assigned higher log-likelihoods. Moreover, the histogram of DHM-5 suggests that the model performed best not because it was able to better maximize the log-likelihood of the in-distribution samples, but because it maximized the log-likelihood of the OOD samples less.

β-VAE	DHM-0	DHM-5
		

Tab. 5.6: Histograms for OOD detection. All models are trained on CIFAR-10. Green corresponds to the score obtained with the in-distribution dataset CIFAR-10 and red corresponds to the score for the OOD dataset CIFAR-100.

Conclusion. Considering that the DHMs are able to provide both $p_{\boldsymbol{\theta}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ in a single forward pass, we find that the hybrid approach does offer some small advantages for OOD detection over the purely supervised and unsupervised baselines. However, every discriminative model can be supported by a separate generative model and vice versa for the OOD detection. Thus, linking two separate models, one discriminative and one generative, would lead to a similar result.

For the far-OOD detection, the unsupervised β-VAE achieved the highest AUROC value along with the DHM-4 and thus can be prioritized over the OOD detection via $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$. On the other hand, the OOD detection via $p_{\boldsymbol{\theta}}(\mathbf{x})$ is not applicable for the near-OOD detection. Neither the purely generative β-VAE nor the DHMs were able to exceed an AUROC value of 0.5 and thus performed worse than a random classifier. This is particularly interesting since maximizing $p_{\boldsymbol{\theta}}(\mathbf{x})$ with respect to a loss function other than the typical loss function for β-VAEs could have led to different results. However, according to our results, we do not consider the optimization of the multi-conditional objective to be advantageous over purely generative models such as β-VAEs. Additionally, we did not find a significant advantage of DHMs over supervised ResNets, suggesting that linking a supervised ResNet-18 with an unsupervised β-VAE is as good as our hybrid approach.

The reasons why deep generative models tend to assign higher log-likelihoods to some OOD datasets have yet to be discovered. However, we think, the same reasons are most likely responsible for our DHMs behaving similarly.

5.2.3 Expected Calibration Error

In this subsection, we present the results concerning the ECE values obtained by our studied models. First, we examine the results on SVHN, followed by the results on CIFAR-10. Again, all values are averaged over ten runs, and the error bars correspond to two times the standard deviation. We also show the reliability histograms to illustrate the differences in the calibration of some selected models and analyze whether these models are overconfident. And finally, we draw a conclusion on whether hybrid modeling via latent variable coupling yields a better calibration.

Results on SVHN. To start with, we compare the ECE values obtained by our studied models on SVHN. For this purpose, take a look at Figure 5.6. The discriminative DHM and ResNet-18 are shown in red and orange, respectively, and serve as our baselines. Comparing the two baselines, we find that the differences are insignificant. However, for DHMs trained on SVHN, varying the discriminative strength $\log \alpha$ affects the ECE. The DHM-2 achieved the lowest ECE with an error of $.043 \pm .001$, i.e., it is best calibrated among all studied models on SVHN. The exact ECE values are shown in Table 5.7, column 2.

Since ECE is computed according to the confidence gaps in Eq. (4.1), we do not yet know whether the predicted probability estimates of our DHMs are higher or lower than the achieved accuracy. To get a glimpse of this, take a look at

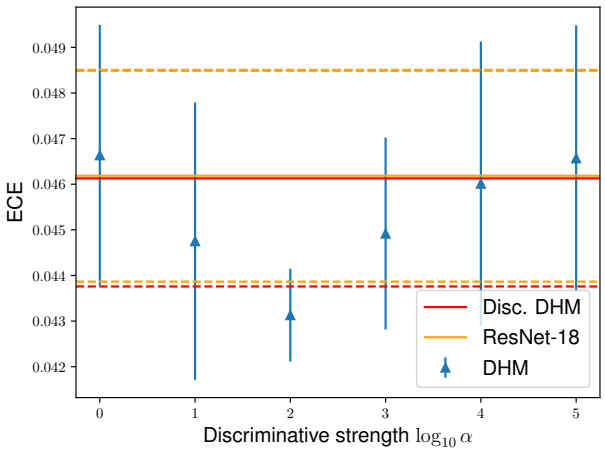


Fig. 5.6: ECE values obtained on SVHN. The discriminative baselines are shown in red and orange, respectively. All values are averaged over ten runs.

	Expected Calibration Error	
Model	SVHN	CIFAR-10
ResNet-18	$.046 \pm .001$	$\mathbf{.057 \pm .001}$
Disc. DHM	$.046 \pm .001$	$.064 \pm .001$
DHM-0	$.047 \pm .001$	$.064 \pm .002$
DHM-1	$.045 \pm .001$	$.063 \pm .002$
DHM-2	$\mathbf{.043 \pm .001}$	$.064 \pm .002$
DHM-3	$.045 \pm .001$	$.065 \pm .002$
DHM-4	$.046 \pm .001$	$.064 \pm .001$
DHM-5	$.047 \pm .001$	$.065 \pm .002$

Tab. 5.7: ECE values on SVHN and CIFAR-10.

Figure 5.7. Examples of the ECE histograms of some selected models are shown there. Note, however, that the values are not averaged over multiple runs, as they are for illustration purposes only. One can see that most of the bars are below the diagonal line, indicating that the model was overconfident and predicted higher probability estimates than it actually achieved in accuracy.

Results on CIFAR-10. Next, we show the results regarding the ECE values obtained on CIFAR-10. In Figure 5.8, the averaged results over ten runs are shown. Note, that all ECE values obtained on CIFAR-10 are higher than those obtained on SVHN, i.e., the models are worse calibrated on CIFAR-10. Also, in contrast to the results obtained on SVHN, the ResNet-18 baseline clearly achieved the lowest ECE. As for the difference between the DHMs

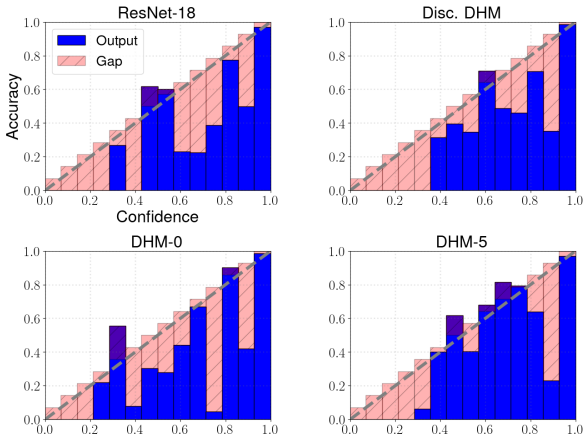


Fig. 5.7: ECE histograms of selected models on SVHN. The gap in each bin contributes to the ECE. The values are not averaged over multiple runs.

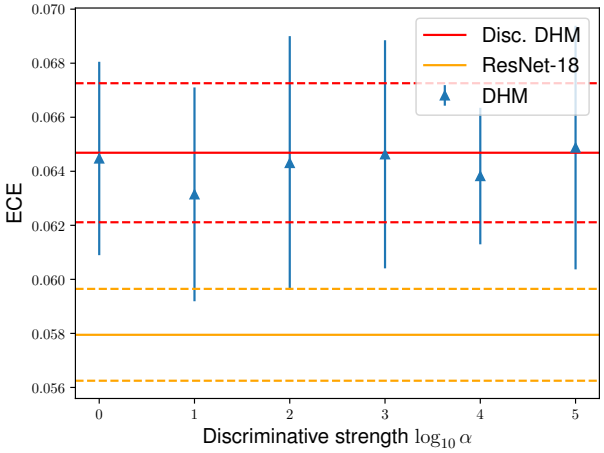


Fig. 5.8: ECE values obtained on CIFAR-10. The discriminative baselines are shown in red and orange, respectively. All values are averaged over ten runs.

and discriminant DHM baseline, we find that the differences are small and insignificant, suggesting that hybrid modeling does not provide an advantage in terms of calibration on CIFAR-10. Also, the clear tendency for DHM-2 to be the best calibrated among all DHMs is not present in this experimental setup. The exact values can be found in Table 5.7, column 3.

Conclusion. Given the results obtained with the SVHN dataset, one might conclude that hybrid modeling via latent variable coupling may lead to a

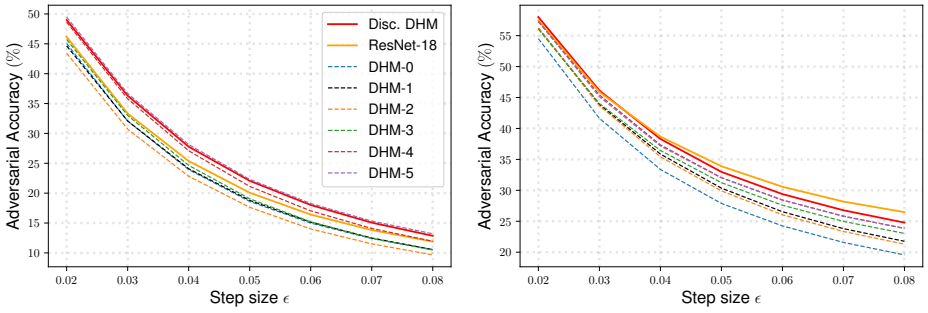
small advantage with respect to ECE. However, the results for CIFAR-10 show that this is not the case. More specifically, since the ResNet-18 performed significantly better on CIFAR-10, we do not consider DHMs to be better suited for tasks where well-calibrated models are required. Since the ECE is only relevant for supervised and semi-supervised learning where classification is required, our DHMs provide reconstructions of input images that are not necessary and contribute to higher computational costs.

5.2.4 Adversarial Accuracy

Our final experiments are dedicated to analyze the models with respect to their adversarial robustness. More specifically, we want to find out whether hybrid modeling via latent variable coupling yields models that are more robust to attacks via the Fast Gradient Sign Method (FGSM). For this purpose, we modified the input images according to the FGSM and chose $\epsilon \in \{0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08\}$ as the step sizes. After modification, the models were tested to classify the manipulated images from the SVHN and CIFAR-10 datasets. Overall, higher adversarial accuracy values with respect to the modified inputs are considered more robust.

Results. The averaged adversarial accuracy values for both SVHN and CIFAR-10 can be seen in Figure 5.9. Starting with SVHN in Figure 5.9a, we find that the FGSM clearly has a very strong influence on the models' accuracy. Even at a step size of $\epsilon = 0.02$, none of our studied models could exceed an adversarial accuracy of 50%. Compared to the accuracy values in Table 5.3, column 2, where FGSM was not applied, this is a notable decrease. Furthermore, as expected, the adversarial accuracy decreased steadily with increasing step size ϵ. For $\epsilon = 0.08$, the studied models achieved adversarial accuracy values from 10% to 14% on SVHN. Considering that SVHN consists of ten mutually exclusive classes, this can be considered random guessing.

As for the comparison between the models, we find that models with higher weights on the classification loss, i.e., DHM-4, DHM-5 and discriminative DHM, perform slightly better. Moreover, compared to the ResNet-18 baseline, hybrid modeling via latent variable coupling does not provide any significant advantages in terms of adversarial accuracy. Nevertheless, the DHM-5 achieved the highest average adversarial values for all step sizes on SVHN, which can be seen in Table 5.8. At the same time, DHM-5 also has the highest standard deviation, which we consider less robust due to reproducibility reasons.



(a) SVHN **(b)** CIFAR-10

Fig. 5.9: Adversarial accuracy results with FGSM for different step sizes on SVHN (left) and CIFAR-10 (right). All values are averaged over ten runs, respectively. The standard deviation is not shown for better readability, but can be seen in Table 5.8.

The tendency of DHMs with higher discriminative strength to yield higher values for adversarial accuracy can be confirmed by our experiments with CIFAR-10 in Figure 5.9b. First, DHM-4 and DHM-5 achieve the highest adversarial accuracy values among all DHMs that were optimized with respect to the multi-conditional objective, as in the case above. Second, the purely discriminative baselines, the discriminative DHM and the ResNet-18, together achieved the highest adversarial accuracy values for all step sizes. However, in contrast to the experiments on SVHN, the adversarial accuracies do not decrease to about 10% at a step size of $\epsilon = 0.08$, but end up at about 20% to 27%. A hypothetical possibility is that the optimization with respect to SVHN is less difficult and therefore the gradients used for FGSM contain more information, which negatively affects the accuracy.

		Adversarial Accuracy in %						
Dataset	Model	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.05$	$\epsilon = 0.06$	$\epsilon = 0.07$	$\epsilon = 0.08$
SVHN	ResNet-18	46.14 ± 2.09	33.33 ± 2.73	25.37 ± 3.10	20.08 ± 3.25	16.42 ± 3.30	13.82 ± 3.27	11.89 ± 3.18
	Disc. DHM	48.99 ± 1.66	36.31 ± 2.17	27.78 ± 2.54	22.07 ± 2.69	18.02 ± 2.78	15.06 ± 2.81	12.86 ± 2.76
	DHM-0	45.05 ± 1.19	32.07 ± 1.36	23.95 ± 1.49	18.62 ± 1.57	15.00 ± 1.60	12.40 ± 1.66	10.56 ± 1.62
	DHM-1	44.66 ± 1.65	32.08 ± 1.55	24.09 ± 1.55	18.79 ± 1.57	15.14 ± 1.55	12.49 ± 1.47	10.55 ± 1.40
	DHM-2	43.43 ± 1.63	30.72 ± 1.86	22.83 ± 1.90	17.64 ± 1.78	14.00 ± 1.65	11.51 ± 1.59	9.67 ± 1.58
	DHM-3	45.73 ± 1.35	33.00 ± 2.13	24.65 ± 2.47	19.06 ± 2.65	15.20 ± 2.65	12.45 ± 2.61	10.47 ± 2.55
	DHM-4	48.62 ± 1.67	35.79 ± 2.35	27.10 ± 2.66	21.12 ± 2.80	17.03 ± 2.83	14.09 ± 2.77	11.94 ± 2.67
	DHM-5	$\mathbf{49.47 \pm 2.11}$	$\mathbf{36.66 \pm 3.19}$	$\mathbf{28.13 \pm 3.84}$	$\mathbf{22.31 \pm 4.35}$	$\mathbf{18.26 \pm 4.65}$	$\mathbf{15.32 \pm 4.78}$	$\mathbf{13.19 \pm 4.87}$
CIFAR-10	ResNet-18	57.45 ± 1.21	45.90 ± 1.73	$\mathbf{38.67 \pm 2.21}$	$\mathbf{33.89 \pm 2.65}$	$\mathbf{30.57 \pm 2.97}$	$\mathbf{28.17 \pm 3.33}$	$\mathbf{26.47 \pm 3.48}$
	Disc. DHM	$\mathbf{58.00 \pm 1.63}$	$\mathbf{46.13 \pm 2.00}$	38.30 ± 2.09	32.97 ± 2.06	29.39 ± 2.11	26.77 ± 2.11	24.78 ± 2.26
	DHM-0	54.51 ± 0.90	41.63 ± 0.75	33.37 ± 1.20	27.88 ± 1.60	24.25 ± 1.80	21.57 ± 1.92	19.56 ± 2.02
	DHM-1	56.21 ± 1.97	43.86 ± 2.26	35.86 ± 2.44	30.35 ± 2.44	26.54 ± 2.48	23.80 ± 2.45	21.76 ± 2.53
	DHM-2	56.22 ± 1.35	43.67 ± 1.71	35.41 ± 1.87	29.90 ± 2.00	26.04 ± 2.01	23.35 ± 2.09	21.30 ± 2.10
	DHM-3	56.04 ± 1.45	44.02 ± 1.70	36.44 ± 1.77	31.23 ± 1.94	27.59 ± 2.01	24.98 ± 2.16	23.03 ± 2.23
	DHM-4	57.30 ± 1.41	45.40 ± 1.66	37.33 ± 1.75	32.05 ± 1.86	28.42 ± 2.00	25.83 ± 2.06	23.84 ± 2.15
	DHM-5	57.29 ± 1.84	45.13 ± 1.79	37.17 ± 1.93	32.00 ± 2.28	28.43 ± 2.60	25.78 ± 2.95	23.87 ± 3.17

Tab. 5.8: Adversarial accuracy values obtained on test set of SVHN and CIFAR-10.

Conclusion. We have shown that adversarial attacks via the FGSM with relatively small step sizes have strong impacts on the accuracy of all investigated models, including DHMs. Interestingly, the regularization effect of the multi-conditional objective of the DHMs does not yield any robustness advantages over purely discriminative models such as the baselines, i.e., discriminative DHM and ResNet-18. On the contrary, considering that the DHMs with higher discriminative strengths led to higher adversarial accuracies, this suggests that interpolation between discriminative and generative approaches does not offer any advantages for DHMs. However, it should be emphasized that tuning hyperparameters of the studied models can have strong effects on performance, and in our experimental setup we used exactly the same training configurations for all models.

Also, note that the models appear to be more robust against FGSM when the dataset is semantically more challenging. Therefore, further analysis could be performed with more challenging datasets such as CIFAR-100 or ImageNet [Deng *et al.* (2009)] to gain better insight into the adversarial robustness of DHMs. However, given our results, we do not consider DHMs to be significantly more robust against adversarial attacks via FGSM or better suited for mission-critical tasks than the baselines.

6 Discussion and Outlook

During the course of this work, several thousand models have been trained to investigate the robustness of Hybrid Discriminative-Generative Models (HDGMs) or, more precisely, Deep Hybrid Models (DHMs). Robustness is often used as a general term, but in this work we have focused on quantitative metrics, i.e., test accuracy, Expected Calibration Error (ECE), Out-Of-Distribution (OOD) detection and adversarial accuracy in the context of the Fast Gradient Sign Method (FGSM). High scores with respect to these metrics are critical and mandatory for the use of probabilistic models in mission-critical tasks. We instantiated HDGMs with Deep Neural Networks (DNNs), resulting in DHMs, which were studied with respect to the above metrics. The analysis was performed numerically using benchmark image recognition datasets, namely Street View House Numbers (SVHN) and CIFAR-10.

We were able to show that the joint optimization of DHMs achieves similar results compared to state-of-the-art DNN architectures, such as supervised Residual Neural Networks (ResNets) and self-supervised β-Variational Autoencoders (VAEs). However, regardless of the discriminative strength, the test accuracy of DHMs did not exceed that of the purely discriminative baselines, namely the discriminative DHM and the supervised ResNet-18. This clearly challenges the results reported in [Kuleshov and Ermon (2017)], where a small gain in accuracy over the purely discriminative setting was shown. Since we consider accuracy to be crucial in our experiments, this is a clear disadvantage of the DHMs.

Moreover, we showed that the problem common to deep generative models of assigning higher log-likelihoods to near-OOD data also applies to DHMs. This is particularly interesting because it shows that the regularization effect of the multi-conditional objective is not sufficient to improve near-OOD detection ability. Overall, we find that the OOD detection did not significantly improve over either the purely generative baseline or the purely discriminative baselines. However, our results do not indicate that our DHMs performed worse either. In the contrary, the ability of DHMs to provide both $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be useful when OOD detection over either quantity fails.

Regarding the ECE, we were able to show that interpolation between discriminative and generative approaches has an impact, albeit small, on the calibration. This resulted in an improvement compared to our baselines for most configurations of the DHMs. Nevertheless, in consideration of the magnitude of the differences, the improvements over the baselines are not remarkable and may differ for other experimental setups. Given our results, we find that instantiating HDGMs with DNNs does not provide a potential solution to obtain non-overconfident probabilistic discriminative models.

Finally, the DHM-5 showed the highest test accuracy in the adversarial robustness framework for all step sizes of the FGSM on SVHN, which is noteworthy. However, this could not be reproduced on CIFAR-10. In addition, the standard deviation of the models that achieved the highest adversarial accuracies were often higher than that of the models that performed slightly worse. It is therefore likely that further experiments with small changes in the configuration of the training could lead to different results.

In general, we find that the results of the DHMs are comparable to those of the baselines in terms of robustness. Basically, DHMs are subject to the same problems as the discriminative and generative baselines, although they have been optimized with respect to a different objective. This is not surprising, since the DHMs are formed on the basis of the discriminative and generative baselines. However, we had hoped that the regularization effect of the multi-conditional objective might have a significant positive impact on the robustness metrics. It should be noted, however, that the DHMs may be useful for special cases where both $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ are needed in a single forward pass.

Now we would like to point out some drawbacks of the experimental setup and provide possible research directions regarding DHMs. First, all numerical experiments were performed with image recognition benchmark datasets. Thus, our experiments represent only one area of many in which probabilistic models are used. In addition, we limited our analysis to the SVHN and CIFAR-10 datasets, except for CIFAR-100, which constituted the OOD dataset. Clearly, more challenging benchmark datasets such as ImageNet are well-studied and could be utilized to perform further experiments. Moreover, analyzing the robustness of probabilistic models is a broad area where many metrics and methods have been developed. We propose to use advanced OOD detection methods such as the Density of States Estimator (DoSE) [Ren *et al.* (2019)] that address the problem of assigning higher probabilities to OOD samples associated with deep generative models. For adversarial accuracy, one could utilize established datasets such as ImageNet-C, ImageNet Adversarial, and ImageNet Rendition, which would allow comparison with other peer-reviewed

articles [Hendrycks and Dietterich (2019); Hendrycks *et al.* (2021a); Hendrycks *et al.* (2021b)].

We also believe that DHMs have potential capabilities in semi-supervised learning as reported in [Kuleshov and Ermon (2017)], which we have not addressed in this work. We therefore propose to further investigate the DHMs in the domain of semi-supervised learning. Moreover, due to the great flexibility of the underlying framework of DHMs, various combinations of discriminative and generative models can be explored. For example, instead of VAEs, one could use normalizing flow models [Rezende and Mohamed (2015)] or Generative Adversarial Nets (GANs) as the generative component. However, we experienced difficulties in the joint optimization of multiple DNNs coupled via latent variables, which may prove to be another drawback of DHMs. Finally, we would like to note that given the speed of progress in deep learning, DHMs provide a versatile tool to integrate upcoming probabilistic models into a single model.

List of Figures

Bibliography

Amodei, Dario *et al.* (2016). "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565*.

Berner, Christopher *et al.* (2019). "Dota 2 with large scale deep reinforcement learning". In: *arXiv preprint arXiv:1912.06680*.

Berner, Julius, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen (2021). "The Modern Mathematics of Deep Learning". In: *CoRR* abs/2105.04026. arXiv: 2105.04026. URL: https://arxiv.org/abs/2105.04026.

Bishop, Christopher M. (1994). "Novelty Detection and Neural Network Validation". In: *IEE Proceedings: Vision, Image, Signal Processing*. Vol. 141.

— (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.

Bottou, Léon (2004). "Stochastic Learning". In: *Advanced Lectures on Machine Learning*. Lecture Notes in Artificial Intelligence, LNAI 3176. Berlin: Springer Verlag, pp. 146–168. ISBN: 978-3-540-23122-6. DOI: 10.1007/978-3-540-28650-9_7.

Brown, Tom *et al.* (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Burgess, Christopher P *et al.* (2018). "Understanding disentangling in β-VAE". In: *arXiv preprint arXiv:1804.03599*.

Choi, Hyunsun, Eric Jang, and Alexander A Alemi (2018). "WAIC, but why? Generative ensembles for robust anomaly detection". In: *arXiv preprint arXiv:1810.01392*.

Davis, Jesse and Mark Goadrich (2006). "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

Deng, Jia *et al.* (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Deng, Li and Xiao Li (2013). "Machine Learning Paradigms for Speech Recognition: An Overview". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.5, pp. 1060–1089. DOI: 10.1109/TASL.2013.2244083.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

Fauw, J. De *et al.* (2018). "Clinically applicable deep learning for diagnosis and referral in retinal disease". In: *Nature Medicine* 24.9, pp. 1342–1350. DOI: 10.1038/s41591-018-0107-6.

Goodfellow, Ian (2016). "Nips 2016 tutorial: Generative adversarial networks". In: *arXiv preprint arXiv:1701.00160*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6572.

Goodfellow, Ian *et al.* (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani *et al.* Vol. 27. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Grathwohl, Will *et al.* (2019). "Your classifier is secretly an energy based model and you should treat it like one". In: *arXiv preprint arXiv:1912.03263*.

Gregor, Karol *et al.* (2015). "Draw: A recurrent neural network for image generation". In: *International Conference on Machine Learning*. PMLR, pp. 1462–1471.

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger (2017). "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 1321–1330.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

— (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Hendrycks, Dan and Thomas Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *Proceedings of the International Conference on Learning Representations*.

Hendrycks, Dan and Kevin Gimpel (2016). "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *arXiv preprint arXiv:1610.02136*.

Hendrycks, Dan *et al.* (2021a). "Natural adversarial examples". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271.

Hendrycks, Dan *et al.* (2021b). "The many faces of robustness: A critical analysis of out-of-distribution generalization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349.

Higgins, Irina *et al.* (2017). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*.

Huang, Gao *et al.* (2017). "Snapshot ensembles: Train 1, get m for free". In: *arXiv preprint arXiv:1704.00109*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations,*

ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6114.

Kingma, Diederik P. and Max Welling (2019). "An Introduction to Variational Autoencoders". In: *Foundations and Trends in Machine Learning* 12.4, pp. 307–392. ISSN: 1935-8237. DOI: 10.1561/2200000056.

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Kuleshov, Volodymyr and S. Ermon (2017). "Deep Hybrid Models: Bridging Discriminative and Generative Approaches". In: *Proceedings of the Conference on Uncertainty in AI (UAI)*.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2016). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *arXiv preprint arXiv:1612.01474*.

Lasserre, J.A., C.M. Bishop, and T.P. Minka (2006). "Principled Hybrids of Generative and Discriminative Models". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1, pp. 87–94. DOI: 10.1109/CVPR.2006.227.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Chen-Yu *et al.* (2015). "Deeply-supervised nets". In: *Artificial intelligence and statistics*. PMLR, pp. 562–570.

Levinson, Jesse *et al.* (2011). "Towards fully autonomous driving: Systems and algorithms". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 163–168. DOI: 10.1109/IVS.2011.5940562.

Liu, Hao and Pieter Abbeel (2020). "Hybrid discriminative-generative training via contrastive learning". In: *arXiv preprint arXiv:2007.09070*.

Madry, Aleksander *et al.* (2017). "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083*.

McCallum, Andrew, Chris Pal, Greg Druck, and Xuerui Wang (2006). "Multi-Conditional Learning: Generative/Discriminative Training for Clustering and Classification". In: *Proceedings of the 21st National Conference on*

Artificial Intelligence - Volume 1 (AAAI'06). Boston, Massachusetts: AAAI Press, 433–439. ISBN: 9781577352815.

Naeini, Mahdi Pakdaman, Gregory F. Cooper, and Milos Hauskrecht (2015). "Obtaining Well Calibrated Probabilities Using Bayesian Binning". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, 2901–2907. ISBN: 0262511290.

Nalisnick, Eric, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan (2019). "Detecting out-of-distribution inputs to deep generative models using a test for typicality". In: *arXiv preprint arXiv:1906.02994* 5, p. 5.

Nalisnick, Eric *et al.* (2018). "Do deep generative models know what they don't know?" In: *arXiv preprint arXiv:1810.09136*.

Netzer, Yuval *et al.* (2011). "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. Accessed: 2022-02-23. URL: http://ufldl.stanford.edu/housenumbers.

Ng, Andrew Y. and Michael I. Jordan (2001). "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes". In: NIPS'01. MIT Press, 841–848.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436. DOI: 10.1109/CVPR.2015.7298640.

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). "f-GAN: Training generative neural samplers using variational divergence minimization". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 271–279.

Paisley, John, David M. Blei, and Michael I. Jordan (2012). "Variational Bayesian inference with stochastic search". In: *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*.

Paszke, Adam *et al.* (2019). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32, pp. 8026–8037.

Radford, Alec *et al.* (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Ren, Jie *et al.* (2019). "Likelihood ratios for out-of-distribution detection". In: *Advances in Neural Information Processing Systems* 32.

Rezende, Danilo and Shakir Mohamed (2015). "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR, pp. 1530–1538.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1278–1286.

Romero, Adriana *et al.* (2014). "Fitnets: Hints for thin deep nets". In: *arXiv preprint arXiv:1412.6550*.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Senior, Andrew W *et al.* (2020). "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792, pp. 706–710.

Srivastava, Nitish *et al.* (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Szegedy, Christian *et al.* (2013). "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199*.

Vaswani, Ashish *et al.* (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Xu, Kelvin *et al.* (2015). "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. PMLR, pp. 2048–2057.

Zhang, Daniel *et al.* (2021). "The AI index 2021 annual report". In: *arXiv preprint arXiv:2103.06312*.

Zhou, Luowei *et al.* (2020). "Unified vision-language pre-training for image captioning and VQA". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07, pp. 13041–13049.

Zhu, Xiaojin and Andrew Goldberg (2009). *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers. ISBN: 1598295470. DOI: 10.2200/S00196ED1V01Y200906AIM006.