UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MODELOS GENERATIVOS-INFERENCIALES: TEORÍA Y APLICACIONES

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

NICOLÁS JAVIER ASTORGA ROCHA

PROFESOR GUÍA:
PABLO ESTÉVEZ VALENCIA

PROFESOR CO-GUÍA:
PABLO HUIJSE HEISE

MIEMBROS DE LA COMISIÓN:
GUILLERMO CABRERA-VIVES
JORGE SILVA SÁNCHEZ

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA,
Y AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: NICOLÁS JAVIER ASTORGA ROCHA
FECHA: 2021
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

MODELOS GENERATIVOS-INFERENCIALES: TEORÍA Y APLICACIONES

This thesis studies generative-inference (GI) models *i.e.* generative models based on neural networks that consider an inference model. The inference model is useful to reach the most relevant features of the observed space $\mathcal{X}$ compressed into the latent space $\mathcal{Z}$. We propose two desired properties for them: one associated with the graphical model and other associated to their representation learning capabilities. Having formalized these properties we focus on how to accomplish them under two frameworks: 1) Matching the joint distributions of GI models and 2) using a novel perspective based on the mutual information of GI models's distributions. From these general perspectives we propose new GI models. We also validate the theoretical findings with extensive experimental results.

We found that the prior distribution $p(z)$ of GI models is fundamental for both generation and representation learning. We derive the models theoretically with multi-modal priors instead of uni-modal priors like $\mathcal{N}(0, I)$. Based on this theory we proposed two models. One model is used for image clustering, obtaining state of the art performance. The other model can be used for anomaly detection in lightcurve datasets thanks to a new decoder and anomaly detection score.

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA,
Y AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: NICOLÁS JAVIER ASTORGA ROCHA
FECHA: 2021
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

MODELOS GENERATIVOS-INFERENCIALES: TEORÍA Y APLICACIONES

Esta tesis estudia modelos generativos-inferenciales (GI) *i.e.* modelos generativos basados en redes neuronales que consideran un modelo de inferencia. El modelo de inferencia es útil para alcanzar las características más relevantes del espacio observado $\mathcal{X}$ comprimidas en el espacio latente $\mathcal{Z}$. Proponemos dos propiedades deseadas para estos modelos: una asociada al modelo gráfico y otra asociada a sus capacidades de representación. Habiendo formalizado estas propiedades, nos enfocamos en alcanzarlas bajo dos marcos: 1) Hacer coincidir las distribuciones conjuntas de los modelos GI y 2) utilizando una perspectiva novedosa basada en la información mutua de las distribuciones de los modelos GI. Desde estas perspectivas proponemos nuevos modelos GI. También validamos los hallazgos teóricos con extensos resultados experimentales.

Observamos que la distribución prior $p(z)$ de los modelos GI es fundamental tanto la generación de datos como para su representación. Derivamos teóricamente los modelos con priors multimodales en lugar de unimodales. Basándonos en esta teoría, propusimos dos modelos. Un modelo es usado para realizar *clustering* en imágenes, obteniendo resultados en el estado del arte. El otro modelo puede ser usado para detección de anomalías en curvas de luz gracias a un nuevo decodificador y un nuevo *score* para detectar anomalías.

A mis padres y a todas la personas que me han dado su apoyo, muchas gracias.

Agradecimientos

Quiero agradecer profundamente a mi mamá y papá que me han apoyado en todo. Sin toda la ayuda que me han dado a lo largo de mi vida no estaría aquí terminando de escribir esta tesis. Gracias a su apoyo, he podido dedicarle el mayor tiempo posible a lo que me gusta, investigar, así que por esto no tengo palabras que puedan expresar mi gratitud. Quiero agradecer a mi abuela Yeya y abuelo Lolo por acompañarme en mi niñez, y a mi abuela Kako y mi abuelo Nano por acompañarme en mi edad más juvenil. Quiero agradecer a mi hermana Cata, a mis tíos Luli, Socio y Andrea, y a mis primos por estar siempre presentes y por su compañía.

Quiero agradecer a la Yara, mi polola, por haber hecho este tiempo de pandemia más divertido y movido, ya sea jugando o saliendo. También quiero agradecer a mis amigos del colegio Rocuz, Óscar, Diego, Cristóbal, Fatias, Lolito y Mario por las interminables horas que hemos jugado videojuegos o juegos de mesa. A mis amigos los bellos: Feña, Alonso, Vichotote, Javier, Vicho Vegano, Mauro, Cata y Alfredo por haber hecho esta vida universitaria mucho más alegre y divertida :D. Quiero agradecer a los cuchufletos Maca, Macla, Maos, Mare y Mati por las risas y estupideces que hemos hecho en la universidad. Gracias también a mis amigos que conocí en el extranjero: Belén, Manuel, Mohit e Isa por hacer esta experiencia mucho más divertida y acogedora.

Quiero agradecer al profesor Pablo Estévez que me recibió en su laboratorio cuando apenas sabía que era una red neuronal y por siempre darme consejos desde la sabiduría. También a Francisco Förster y a Guillermo Cabrera por incluirme en sus colaboraciones. Quiero dar gracias a mis amigos del laboratorio Germán, Rodrigo, Esteban, Nico, Rosario, Ignacio, Mau y Óscar por compartir esta única experiencia de aprendizaje, y bueno, por procastinar en el lab cuando era necesario. Quiero agradecer a Pavlos Protopapas por invitarme a realizar una pasantía de investigación y obtener una experiencia inolvidable. Finalmente quiero dar gracias a Pablo Huijse que en términos académicos es la persona que más me ha apoyado y confiado en mí, gracias a él soy el investigador que soy hoy en día.

Aunque no las haya mencionado, muchas gracias a todas las personas con quienes compartí buenos momentos. Saludo especial al Balto, que le gusta acompañarme en las tardes de pandemia.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In probability and statistics, discriminative and generative models are the two main approaches to model data and/or unobserved distributions. Discriminative methods usually model the posterior distribution of the data $p(z|x)$, being $x \in \mathcal{X}$ the observed variable and $z \in \mathcal{Z}$ the target variable. For example x may represent the independent variables or features entering a classifier and z the class predicted by the classifier. On the other hand generative models often takes $\mathcal{Z}$ as a hidden latent space and estimate the joint distribution $p(x, z)$. Usually this joint distribution is obtained by approximating the real data distribution $q(x)$ with a modelled marginal distribution $p(x)$. In general this approximation is obtained by sampling points z_i from a prior distribution $p(z)$ and obtaining, after several transformations, its decodified version called $\tilde{x}_i$. Ideally this decodified version $\tilde{x}_i$ of z_i should have a high likelihood with respect to the real distribution $q(x)$, i.e. we can use the model to create new synthetic data that is similar to our observations.

The data that come from this sampling procedure have different properties depending on the optimization and the transformations applied to z_i. Simpler transformations have been used by classical approaches like Naive Bayes models [65, 82], Hidden Markov Models [89, 90], Gaussian mixture models [76, 93], Latent Dirichlet allocations [7], among others. More recent generative models [29, 53, 94, 110] have used deep neural networks [28] to increase the amount and flexibility of the transformations applied to z_i. Neural networks are universal approximators [72] and thanks to recent advances [34, 43, 78, 24] generative models have scaled to unprecedented high dimensional data. Examples of such models are Generative Adversarial networks [29] (GANs), Variational Autoencoders [53] (VAEs), Normalizing flows [94] and Autoregressive models [110].

When $\mathcal{Z}$ is lower dimension than $\mathcal{X}$ the generative process induce a compression of the real distribution $q(x)$ into the prior distribution $p(z)$, this information could be used if an inference model is available to reach the latent space $\mathcal{Z}$ of the prior. The generative models that consider such inferential distribution will be denominated as generative-inference models for the remainder of this thesis. The study of the generative and representation learning capabilities of generative-inference models is the main focus of this work. In representation learning it is desirable that the inferred space should be useful for tasks like compression or classification, for this reason we will consider models that can be optimized when the

dimension of the latent space $\mathcal{Z}$ is less than the observable space $\mathcal{X}$. Two generative models that follow this optimization are Variational autoencoders [53] (VAEs) and Generative adversarial networks [29] (GANs) which will be considered as the generative process of many generative-inference models.

In VAE an encoder and a decoder network pair is trained to map the data to a low-dimensional latent space, and to reconstruct it back from the latent space, respectively. The encoder is used for inference while the decoder is used for generation. The main limitations of the standard VAE are the restrictive assumptions associated with the explicit distributions of the encoder and decoder outputs. For the latter this translates empirically as loss of detail in the generator output. In GAN a generator network that samples from latent space is trained to mimic the underlying data distribution while a discriminator is trained to detect whether the generated samples are true or synthetic (fake). This adversarial training strategy avoids explicit assumptions on the distribution of the generator, allowing GANs to produce the most realistic synthetic outputs up to date [8, 48, 49, 50]. The weaknesses of the standard GAN are the lack of inference capabilities and the difficulties associated with training (*e.g.* mode collapse). Recent works [66, 19, 23] have extended GANs including classifiers and/or encoders to perform inference. Generative models that include a classifier and/or encoder can be used for additional applications like semi-supervised learning and clustering.

VAEs have been extended by adding associated class information in [54, 74] with a semi-supervised approach. Other unsupervised VAE approaches have chosen to modify the prior distribution [46, 47] to force separability of the latent variables and perform clustering. GANs have also been studied with a similar purpose in [103, 79, 4] for clustering and in [66, 17, 98] for semi-supervised learning. Some models have combined both GANs and VAEs approaches to exploit their advantages [23, 20, 68, 14] but these have not been used largely for clustering or semi-supervised learning. The performance of clustering or semi-supervised learning can be associated with the correlation between the input and the latent variables. In the literature the area of study that looks for useful embeddings of the data is called representation learning and is one of our main concerns in the study of generative-inference models.

Representation learning [6] of data in an unsupervised way is a fundamental problem in machine learning. The objective of representation learning is to train an encoder that transforms the observable space into a generally lower dimensional space and thus obtain a "representative space" of data. The properties required in a "representative space" will depend on the task at hand. For dimensionality reduction tasks a representative space is one where redundancy and noise are reduced, effectively compressing the data to its most relevant features. For classification tasks a "representative space" should be a low-dimensional space whose characteristics are linearly separable. Nowadays finding linearly separable spaces is the main goal of "representation learning" and many unsupervised algorithms evaluate their effectiveness on a supervised task that is trained after an unsupervised training [3, 38, 35]. One way to measure the representation learning capabilities of a model is by using the concept of mutual information from the field of information theory.

Information theory is a broad area initially proposed by Shannon in 1948 [99], that studies the quantification and storage of information. Generative models store the empirical data distribution information in a prior distribution to generate realistic data. The likelihood of

the data is maximized using information-theoretic concepts such as divergences and cross-entropies between distributions of random variables. This optimization gives us a way to approximate an empirical distribution with a model but it does not tell us insights of how correlated is our input with respect its codifications. To measure such dependency information theory gives us another tool called mutual information. Mutual information have been used by discriminative methods obtaining state of the art performance [3, 35] in representation learning tasks. In generative models mutual information has been maximized in [14]. Few works in generative models have chosen a more general approach [121] but they have not explored the trade-offs between generation and representation learning. Our work studies generative-inference models from an information theory viewpoint to understand them from a broader perspective. We study the relation of generative-inference models using information theory measures like mutual information, Kullback Leibler (KL) divergence and cross-entropy. We perform a deep theoretical and empirical analysis of the trade-offs between generation and representation capabilities. Finally, we propose new generative-inference models based on this analysis.

The structure of this thesis is the following. In Chapter 2 we describe the most fundamental previous work required for the understanding of this thesis. In Chapter 3 we enunciate the foundations of generative-inference models. In Chapter 4 we show how to obtain generative-inference models from a joint distribution matching perspective between their generative and inference distributions. In Chapter 5 we associate generative-inference models with the mutual information of the generative and inference distributions. We analyze the trade-offs between generation and representation capabilities of generative-inference models. From this analysis we propose and thoroughly evaluate new generative-inference models. In Chapter 6 we present the empirical results obtained by these new models and we test some observations made in Chapter 5. In Chapter 7 we study models that consider an additional categorical variable, in particular we create a new clustering model called MPCC for which we present extensive experiments in Chapter 8. In Chapter 9 we apply a generative-inference model called VaDE to a practical application in astronomy. Finally in Chapter 10 we discuss the implications of our work and future research.

1.1 Hypothesis

- It is possible to design a framework that generalizes generative-inferential models based on information theory.
- The optimization framework of generative-inference models is relevant for training, and has a big impact on their generative and inference capabilities. In other words, even if the architectures to model the conditional distributions $p(x|z)$ (decoder) and $q(z|x)$ (encoder) are the same they can perform drastically different depending of how the model is trained.
- The selection of the prior for generative-inference models has a big impact on generation and inference. Generative-inference models can be extended with multimodal priors to perform tasks such as clustering and outlier detection, outperforming the traditional unimodal priors. Depending on the task at hand this will be measured using Fréchet inception distance, classification accuracy or cluster purity.

1.2 General Objective

To develop a new theoretical framework to unify and generalize existing generative-inference models. The framework is based on the relation between the distributions that compose generative-inference models. We recognize two general schemes to unify generative-inference models. The first scheme is based on a matching joint distributions perspective. The second scheme is the association of generative and inference distributions under the concept of mutual information. This association allows us to study the impact on the model's loss functions in terms of generation and inference. Based on this analysis and unified framework we propose new generative-inference models.

1.3 Specific Objectives

- Associate the loss function of generative-inference models with the Kullback Leibler divergence of its distributions and identify if a relation between generation and/or inferential capabilities exists.
- Develop a new mathematical perspective that associates the loss function of a generative-inference model with the mutual information of its distributions and identify if a relation between generation and/or inferential capabilities exists.
- Identify metrics computed in an unsupervised way that allows us to identify generative-inference models that perform good in generation, inference or both.
- Propose, analyze and test a generative-inference model for clustering in computer vision benchmark datasets by changing the common unimodal prior distribution $\mathcal{N}(0, I)$ by a mixture distribution prior.
- Propose, analyze and test a generative-inference model for classification and outlier detection applied to astronomical light curves. This includes the design of a new decoder based on Gaussian processes that allows training with irregularly sampled time series.

Chapter 2

Generative-Inference models: Background

In this chapter we study the most relevant methods and evaluation metrics for the understanding of this thesis. Since our work comprehend the study and association of different generative-inference models in a common general mathematical framework, some of the previous work will be explained in following chapters where the context is closer to a more specific contribution. For example, in Chapter 3 we give the foundations of what we called generative-inference models and we formalize some of the notation that is introduced in this chapter. We chose this structure for an easy reading of the thesis.

Two generative models are the base of our study: Generative adversarial networks (GANs) [29] and Variational autoencoders (VAEs) explained in section 2.1.1 and 2.2, respectively. In section 2.3 we explained the most relevant metrics that are used in this thesis, more specific metrics are explained in the corresponding chapters.

2.1 Generative models

2.1.1 Generative adversarial networks

Generative adversarial networks (GANs) [29] is a method that approximates the available samples of the real data distribution $q(x)$ by decoding a prior distribution $p(z)$ with a decoder $p(x|z)$ *i.e* it seeks to match the marginal distributions of the generative model $p(x) = \mathbb{E}_{p(z)}[p(x|z)]$ with the empirical data distribution $q_\delta(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i)$, where δ is dirac delta function. The GAN loss function can be formulated as follows:

$$\max_{D} \ \mathbb{E}_{x\sim q_\delta(x)}[f(D(x))] + \mathbb{E}_{\tilde{x}\sim p(x,z)}[g(D(\tilde{x}))], \tag{2.1}$$

$$\min_{G} \ \mathbb{E}_{\tilde{x}\sim p(x,z)}[h(D(\tilde{x}))]. \tag{2.2}$$

This formulation is generalized in [21] where D is a discriminator network and G is a generator network, tilde is used to denote sampled variables. In the original GAN formulation [29] these functions are defined as $f(y) = -\log(1 + \mathrm{e}^{-y})$, $g(y) = -y - \log(1 + \mathrm{e}^{-y})$ and $h(y) = -y - \log(1 + \mathrm{e}^{-y})$.

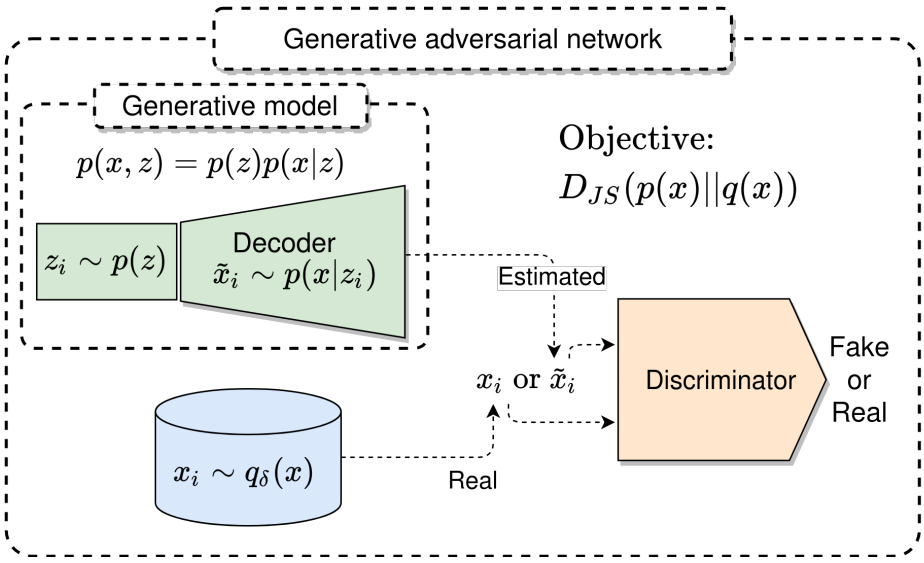


Figure 2.1: Diagram of GAN adversarial training.

The score of the discriminator network is maximized for data that come from the empirical data distribution $q_\delta(x)$ and the score is minimized when the data is received from the generator network as observed in Eq. (2.1). Realistic samples from $p(x)$ are obtained by enforcing the decoder $p(x|z)$ to maximize the score of the discriminator network as shown in Eq. (2.2).

In [29] it is demonstrated that vanilla GAN minimizes the Jensen Shannon divergence $D_{JS}(p(x)||q(x))$ under an optimal discriminator and generator. More recent approaches [85, 1, 108] in the literature have shown that other integral probability metrics like Wasserstein distance [1], Maximum Mean Discrepancy (MMD) [67], or any divergence from the f-divergence family [85] are minimized under different selection of f, g, and h. We will use the notation $\mathcal{D}(p(x)||q(x))$ to refer to any divergence [85] or integral probability metric [1, 67] that can be minimized using a general adversarial training approach like Eq. (2.1) and Eq. (2.2). In general these equations optimize loss functions that induce a matching between $p(x)$ and $q(x)$. In Fig. 2.1 shows a diagram of the vanilla GAN optimization procedure.

2.2 Variational autoencoders

The Variational autoencoder (VAE) [53] is a generative model that maximizes the likelihood of the marginal distribution of the generative model $p(x)$ with respect to the real distribution $q(x)$. In practice it is not feasible to optimize the likelihood $\mathbb{E}_{q(x)}[\log p(x)]$ since it requires to evaluate each sample x_i from the real data distribution $q(x)$ with $p(x)$. To estimate the marginal distribution of the model $p(x) = \mathbb{E}_{p(z)}[p(x|z)]$ a large amount of samples $z_i \sim p(z)$ are required for each x_i. The search space of $p(z)$ can be reduced by introducing an amortized encoder distribution that obtains samples $\tilde{z} \sim q(z|x)$ dependent on real data x_i. The VAE loss function $\mathcal{L}^{\text{VAE}}$ can be obtained by adding this amortized encoder as follows:

$$
\begin{aligned}
\mathbb{E}_{q(x)}[\log p(x)] &= \mathbb{E}_{q(x)}\left[\log \int_z p(x|z)p(z)dz\right] \\
&= \mathbb{E}_{q(x)}\left[\log \int_z \frac{p(x|z)p(z)}{q(z|x)} q(z|x)dz\right] \\
&= \mathbb{E}_{q(x)}\left[\log \mathbb{E}_{q(z|x)}\left[\frac{p(x|z)p(z)}{q(z|x)}\right]\right] \\
&\geq \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)}\left[\log \frac{p(x|z)p(z)}{q(z|x)}\right] \qquad (2.3) \\
&= \mathbb{E}_{q(x)}\left[\mathbb{E}_{q(z|x)} \log p(x|z) - D_{KL}(q(z|x)||p(z))\right] \qquad (2.4) \\
&\equiv \text{ELBO} \equiv -\mathcal{L}^{\text{VAE}},
\end{aligned}
$$

where Jensen's inequality was used in step (2.3). From the previous demonstration we note that the VAE loss function $\mathcal{L}^{\text{VAE}}$ is a lower bound on the marginal likelihood $\mathbb{E}_{q(x)}[\log p(x)]$, which is commonly known as the evidence lower bound (ELBO). For an efficient computation of the loss function in Eq. (2.4), the term $\mathbb{E}_{q(x)q(z|x)}[\log p(x|z)]$ is computed by minimizing the MSE between the input data x_i and its reconstruction. The term $\mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))]$ is computed in closed-form assuming $q(z|x)$ and $p(z)$ as normal distributions with a diagonal covariance matrix $\mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$ and $\mathcal{N}(0, I)$, respectively. The parameters $\tilde{\mu}$ and $\tilde{\sigma}$ are learned through neural networks. Under this assumption the second term in Eq. (2.4) has the following closed-form

$$
D_{KL}(q(z|x)||p(z)) = \sum_{j=1}^{J} \tilde{\mu}^2 + \tilde{\sigma}^2 - \log \tilde{\sigma}^2 + 1, \qquad (2.5)
$$

where J is the dimensionality of the latent space.

Fig 2.2 shows a graphical diagram of the VAE loss function obtained by the previous demonstration (Eq. (2.4)). VAE is one of the most studied generative models of the literature and several approaches have advanced on them, increasing the flexibility of their encoder $q(z|x)$ or prior $p(z)$. Particularly relevant are the methods that modify the restriction $D_{KL}(q(z|x)||p(z))$. β-VAE [37] adds a constant to the divergence and Adversarial variational bayes (AVB) [77] replaces it with adversarial training. AVB matches this distribution similarly to GANs by identifying the pair of points $(x_i, \tilde{z}_i)$ from the pair (x_i, z_i), where $x_i \sim q_\delta(x), \tilde{z}_i \sim q(z|x = x_i)$ and $z_i \sim p(z)$ (for insight about this type of training see section 4.1) . Adversarial autoencoders (AAE) [75] and Wasserstein autoencoders (WAE) [107] replace this divergence with adversarial training in the marginal distributions minimizing $\text{Adv}(q(z)||p(z))$. Finally Info-VAE [120] and also WAE replace this divergence with MMD in the marginal distributions minimizing $\text{MMD}(q(z), p(z))$. In Chapters 4 and 5 we show how these models can be unified from a matching joint distributions perspective and a mutual information perspective, respectively.

2.3 Evaluation metrics

As mentioned in Chapter 1 the main focus of this thesis is the study of generative-inference models *i.e.* models with generative and inferential capabilities. The generative capability

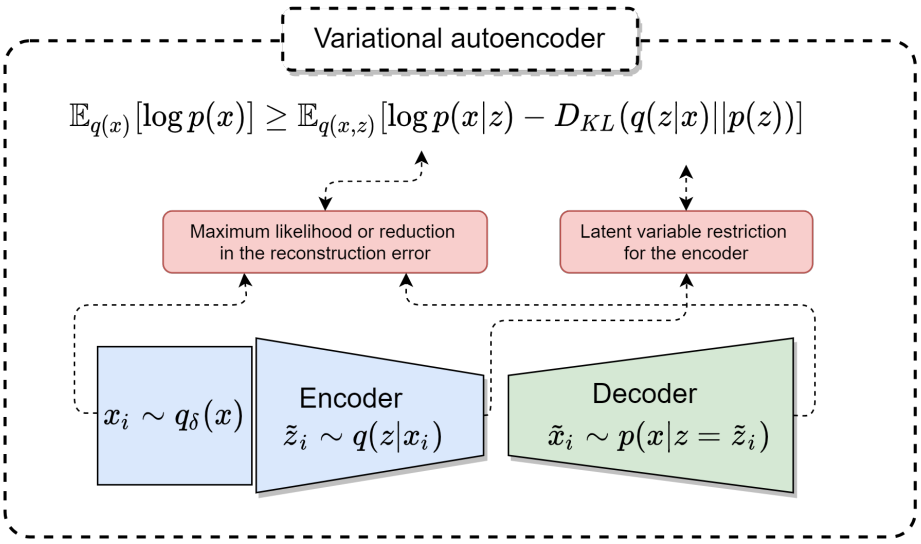


Figure 2.2: Diagram for the optimization of a variational bound of $\mathbb{E}_{q(x)}[\log p(x)]$.

is measured by broadly [8, 50] used metrics, the Inception score (IS) [98] and the Fréchet inception distance (FID) [36], which are explained in sections 2.3.1 and 2.3.2, respectively. The inferential capabilities are measured using a linear predictor on top of the frozen representation as is done in state of the art unsupervised representation learning algorithms [12, 13, 9].

2.3.1 Inception Score (IS)

Inception Score (IS) is a quality metric that measures how realistic an image looks for an human observer. A higher value of this score is better. This score was originally presented in [98] and is the first proposition for an automatic evaluation of the generative capabilities of generative models. The IS metric is given by

$$\begin{aligned} \text{IS} &= \exp\left(\mathbb{E}_{p_\delta(x)}[D_{KL}(p^*(y|x)||p^*(y))]\right) && (2.6) \\ &= \exp\left((\mathbb{E}_{p_\delta(x)p^*(y|x)}[\log p^*(y|x)] - \mathbb{E}_{p^*(y)}[\log p^*(y)]\right), && (2.7) \end{aligned}$$

where $p_\delta(x)$ are samples of the generative model and $p^*(y)$ is the marginal distribution of the data classified using the pretrained classifier $p^*(y|x)$. The meaningful information of a generated image is measured by the first term in Eq. (2.7) that is expected to have low entropy for $p^*(y|x)$. The variability of a generated image is measured by the second term of Eq. (2.7), *i.e.* the distribution $p^*(y)$ should have high entropy.

The IS metric has been widely used in the literature [8, 114, 20] but it has a severe limitation. The variability is measured by the entropy of $p^*(y)$, but it only measures variability in terms of different classes. For example the entropy of $p^*(y)$ could be high but the generated images of a given index $y = c$ can be the same generated image repeated several times.

2.3.2 Fréchet inception distance (FID)

The Fréchet inception distance (FID) is a generation quality score that also measures how the generated images of the model look to a human observer. This score improves over

the IS as it includes the variability of the generated images over multiple features. For its computation a pretrained classifier $p^*(y|x)$ is required. Instead of using the labels as IS, FID uses a pretrained classifier to obtain an embedding z in an earlier layer of the network. This embedding is fitted with a multivariate Gaussian distribution with statistics μ and Σ. When the samples of the generated model p_δ are used as input we obtain the mean and covariance μ_1 and Σ_1, respectively. When the samples from the real data are used as input we obtain the mean and covariance μ_2 and Σ_2, respectively. The FID score compares these distributions as follows:

$$\text{FID} = ||\mu_1 - \mu_2||^2 + \text{Tr}(\Sigma_1 + \ \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}), \tag{2.8}$$

where lower values are associated with more similar distributions. We refer the reader to [36] for complete derivation of Eq. (2.8).

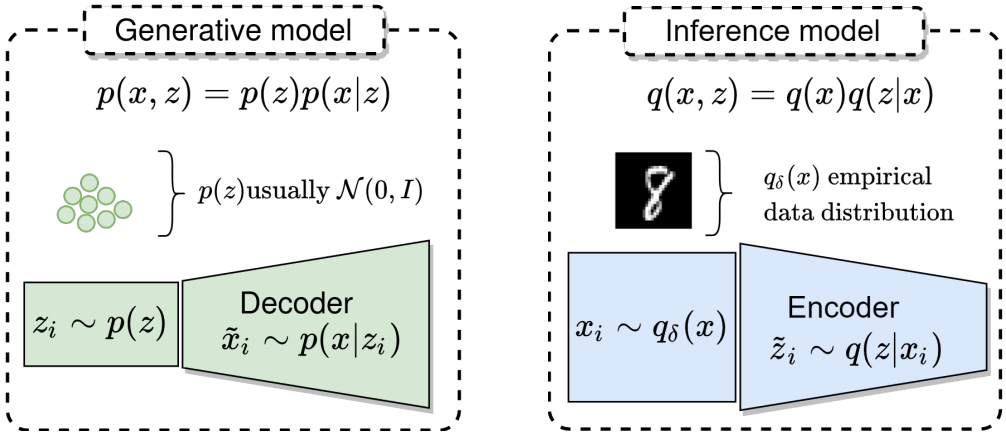
In the original paper [36] they used an inception network pretrained on Imagenet [18] as a feature extractor with an embedding of 2048 dimensions. For both IS and FID we will specify how the pretrained classifier is obtained.

Chapter 3

Generative-Inference models: Foundations

We will refer as generative-inference models to every model that considers a generative model p and an inference model q. For these models, in general we will consider two variables: $x \in \mathcal{X}$ and $z \in \mathcal{Z}$, $\mathcal{X}$ being the observable space and $\mathcal{Z}$ being a latent low-dimensional manifold space. When two variables are considered the joint distributions are $p(x, z)$ and $q(x, z)$. The generative model $p(x, z) = p(x|z)p(z)$ is decomposed in a prior distribution $p(z)$ and a decoder $p(x|z)$ usually modeled as a neural network (NN), see Fig. 3.1a for a graphical insight and Fig. 3.2a for the graphical model. The inference model $q(x, z) = q(x)q(z|x)$ is decomposed in the true underlying distribution of the data $q(x)$ and an encoder $q(z|x)$ usually modeled as a NN that codifies the data, see Fig. 3.1b for a graphical insight and Fig. 3.2b for the graphical model. Note that we refer to $q(x)$ as the real data distribution and not to the empirical data distribution as previous work [23, 2]. We will use δ as sub-index to emphasize that we are using the empirical distribution, i.e. a counting distribution based on a finite sample from a theoretical distribution. For example, in this case $q(x)$ is the real data distribution and $q_\delta(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i)$ is the empirical data distribution of the samples generated by $q(x)$, where δ is dirac delta function. For the rest of this thesis every term that requires an expectation $\mathbb{E}_{q(x)}$ will be evaluated in practice by $\mathbb{E}_{q_\delta(x)}$ since we don't have access to $q(x)$.

Matching the empirical data distribution $q_\delta(x)$ with the marginal distribution of the generative model $p(x)$ is one of the main goals of generative-inference models. Moreover, the objective of these models is to have the ability to generalize *i.e.* approximate $q(x)$ having only access to $q_\delta(x)$ (see Fig. 3.3). In Chapter 4 we will observe how $p(x)$ matches $q(x)$ for different generative-inference models. We argue that correctly approximating the real distribution depends on the architecture and the optimization framework. In this work we will explore these capabilities for generative-inference models based on two dependencies: the optimization framework and the selection of the prior distribution. The prior distribution has a big impact in the approximation of the real distribution, which is observed in generation quality performances [49, 2]. This impact can be explained by the fact that disconnected features in the observed space should have also disconnected features in the low-dimensional manifold space (see Fig. 3.5). We can manipulate the low-dimensional space through the use



(a) Generative model (b) Inference model

Figure 3.1: (a) Generative model (coloured green) composed by a prior distribution $p(z)$ usually modeled as a standard normal distribution $\mathcal{N}(0, I)$ and a decoder $p(x|z)$ usually modeled as a neural network. (b) Inference model (coloured blue) composed by the real data distribution $q(x)$ and a encoder $q(z|x)$ usually modeled as a neural network.

(a) Generative model (b) Inference model

Figure 3.2: Graphical model of (a) generative model and (b) inference model.

of different priors.

Another goal of generative-inference models is to match $q(z) = \mathbb{E}_{q(x)}[q(z|x)]$ with the selected prior distribution $p(z)$. The selection of the prior distribution has important effects in the inferred variables $q(z)$ of these models and its influence is thoroughly studied in Chapters 7 and 8. A restrictive prior $p(z)$ can hinder the ability to properly compress the data of the empirical data distribution $q_\delta(x)$ into $p(z)$ (see Fig. 3.3). For example, different classes shouldn't share certain features in the low-dimensional manifold space (See Fig. 3.5). The most broadly used prior in the literature is the very restrictive $p(z) = \mathcal{N}(0, I)$, which is commonly used in GANs [29] and in VAEs [53]. This prior doesn't give the possibility for data that is separated in the observable space to be separated in the latent space. One option to increase the flexibility of the prior distribution is to model it as a mixture of distributions

$$p(z) = \sum_{i=1}^{K} p(z|y = c_i)p(y = c_i) \text{ with } y \in \mathcal{Y},$$

where $\mathcal{Y}$ is the space of categorical variables. The models that consider mixture of distributions in their prior are studied in Chapter 7. In this thesis we address representation learning by studying the class separation in the manifold space and its relation with the prior distribution $p(z)$. In Chapter 5 we observe that the inductive-bias of generative-inference models is given by the maximum likelihood of the decoder or encoder, or more intuitively by the

reconstruction error reduction in $\mathcal{X}$ or $\mathcal{Z}$, respectively (see Fig. 3.4). This reconstruction error is one way to measure the dependency of $x_i \in \mathcal{X}$ with its corresponding variable in the latent space $z(x_i) \in \mathcal{Z}$ and can be associated with the concept of mutual information and representation learning.

Note that representation learning is not a primary objective in GANs [29] since they don't even have an inference model. We establish two desired properties of generative-inference models that will be thoroughly studied in Chapters 4 and 5 (see Fig. 3.3 and Fig. 3.5). These properties are:

- **Desired property 1**. *The marginal distributions of generative-inference models should match i.e.* $q(z), p(z)$ *should be equal and* $p(x), q(x)$ *should be equal.*
- **Desired property 2**. *Data* $x_i \in \mathcal{X}$ *sampled from the empirical data distribution* $q_\delta(x)$ *should have a high correlation with its codified version* $z(x_i) \in \mathcal{Z}$ *in the latent space.*

Desired property 1 is a restriction given by the graphical models p and q. The inferred marginal distribution $q(z)$ should follow the prior distribution $p(z)$ and the generated marginal distribution $p(x)$ should follow the real distribution $q(x)$. We study this property from a joint matching distribution perspective in Chapter 4 and from a mutual information perspective in Chapter 5. Note that matching marginal distributions doesn't give us information about the relation between an observation $x_i \in \mathcal{X}$ and its latent codification $z(x_i) \in \mathcal{Z}$.

Desired property 2 tells us about the representation learning capabilities of the model. If the codified version $z(x_i)$ of the data $x_i \sim q_\delta(x)$ is well represented in the latent space this would mean that a simpler classifier can be trained in the lower dimensional space $\mathcal{Z}$. The desired property 2 is studied in Chapter 5 by deriving the loss function of various generative-inference models from a mutual information perspective and associating them with their representation learning objective. We study the mutual information perspective by bounding it with likelihoods $\mathbb{E}_{q(x)}[\log p(x|z)]$ or $\mathbb{E}_{p(z)}[\log q(z|x)]$, which is different from the approaches studied in [88] that train a critic to differentiate $z_i \sim p^*(z)$, $x_i \sim p^*(x)$ from $x_i, z_i \sim p^*(x, z)$.

At the end of Chapter 5 we compare the mutual information perspective and the joint matching distribution perspective observing the trade-offs between generation/compression capabilities and representation learning. Finally we create new models based on these analyses.

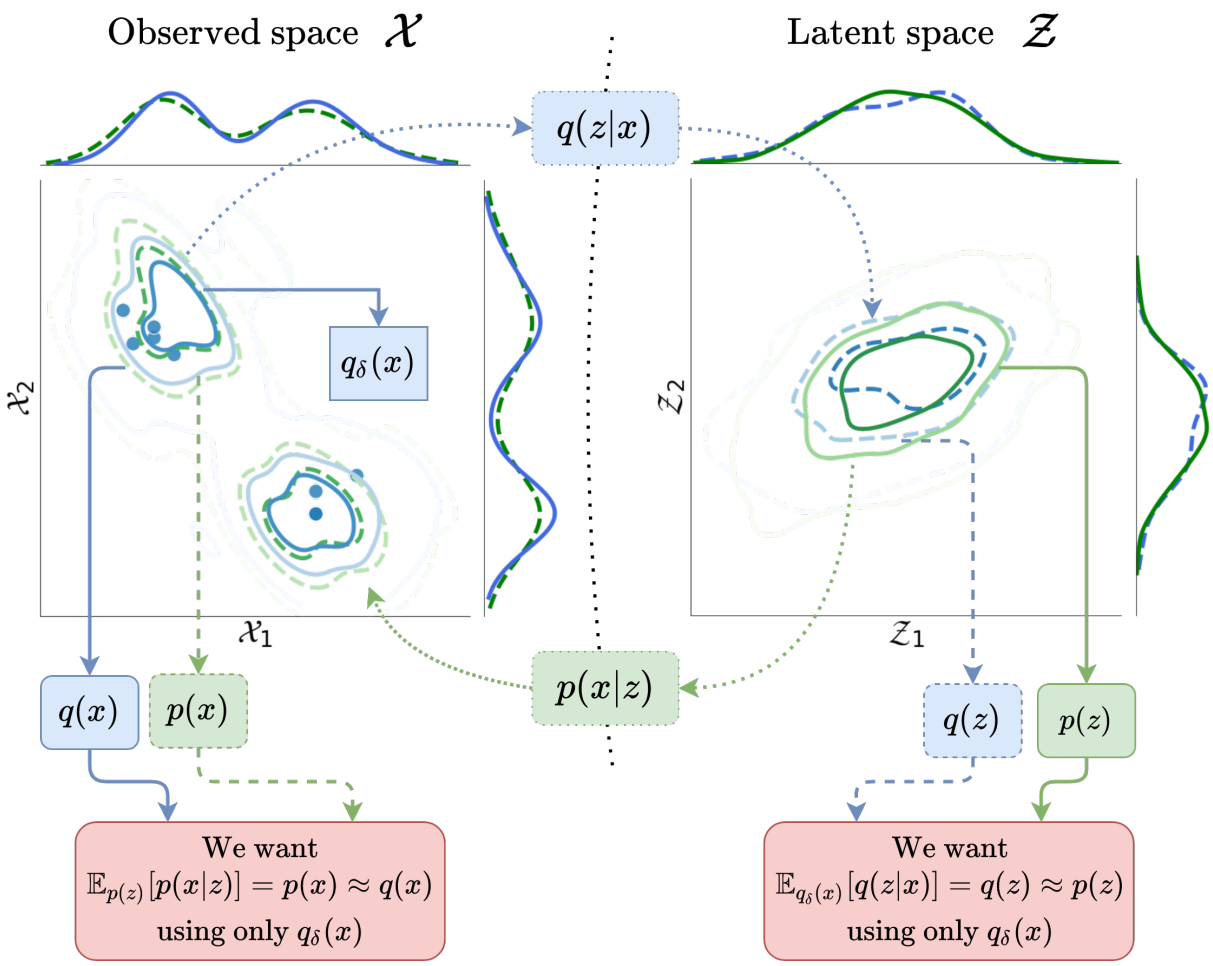


Figure 3.3: Diagram of the first objective of generative-inference models. The first objective of these models is that their marginals distributions should match, i.e. $p(x)$ matches $q(x)$ and $q(z)$ matches $p(z)$. The red rectangle makes reference to this first objective. The blue and green colors are associated with components of the inference and generative model, respectively. Solid lines correspond to the true underlying distribution, the circles correspond to samples from this distribution, dashed lines correspond to the approximation of the marginal distribution obtained by the model and dotted arrows correspond to transformations, *i.e.* conditional distributions of the model; encoder or decoder.

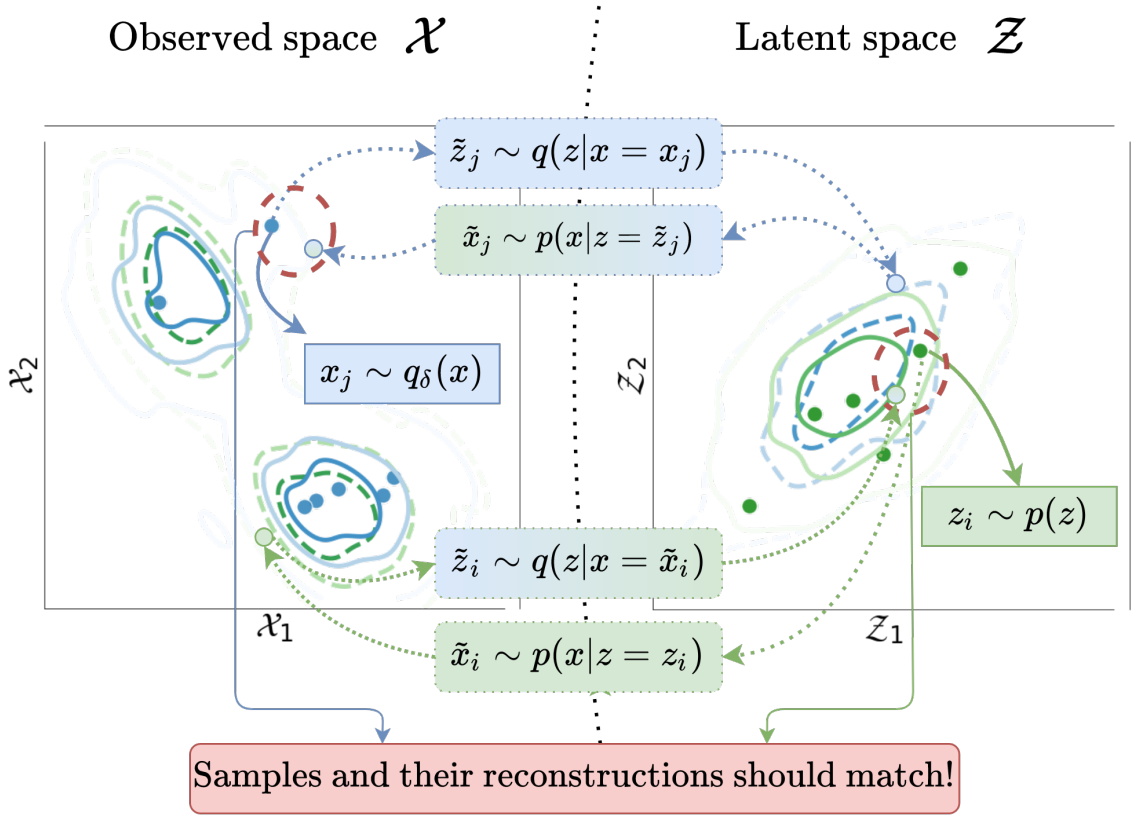


Figure 3.4: Diagram of the second objective of generative-inference models. The second objective of these models is that the samples from the underlying distributions of the data and its reconstruction should match, *i.e.* $\mathbb{E}_{q_\delta(x)q(z|x)}[\log p(x|z)]$ and $\mathbb{E}_{p(z)p(x|z)}[\log q(z|x)]$ should be high. The red rectangles makes reference to this second objective. The blue and green colors are associated with components of the inference and generative model, respectively. Rectangles and data with both colors correspond to transformations or samples, respectively, that involve the two model distributions. Solid lines correspond to the true underlying distribution, the circles correspond to samples from this distribution, dashed lines correspond to the approximation of the marginal distribution obtained by the model and dotted arrows correspond to transformations *i.e.* conditional distributions of the model; encoder or decoder.

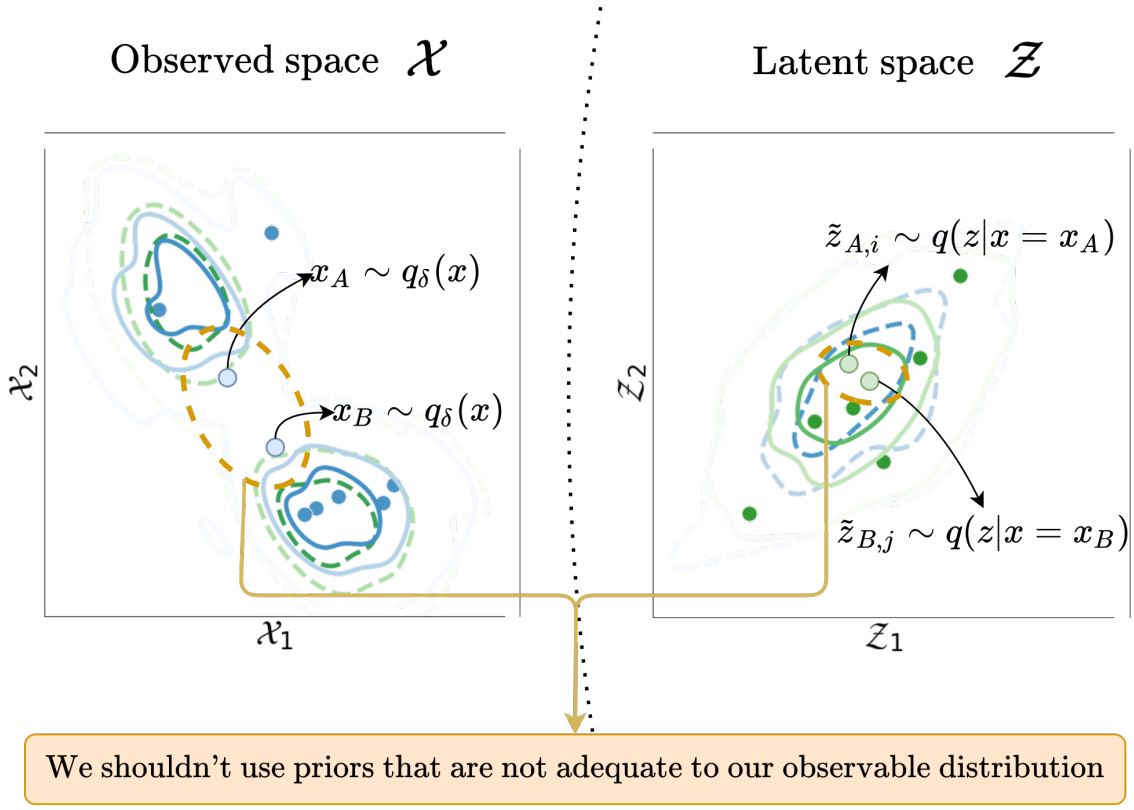


Figure 3.5: Example that shows the impact of a badly selected prior for the observable distribution. We should not set an unimodal distribution as a prior if the distribution of the observable space is a mixture of distributions. The points x_A and x_B are separated despite that their respective codifications $\tilde{z}_{A,i}$ and $\tilde{z}_{B,j}$ are close in the latent space. This will happen without loss of generality to some of the pairs that comes from different modes in the observable space and have an unimodal distribution as prior.

Chapter 4

Generative-Inference models: A matching joint distributions perspective

Desired properties 1 and 2 presented in Chapter 3 can be achieved by matching the joint distributions $p(x, z)$ and $q(x, z)$. If the joint distributions are equal then the marginal distributions $p(x)$, $q(x)$ and $q(z)$, $p(z)$ are equal too (desired property 1). This is easy to observe by integrating one variable *i.e.* $\int_z p(x, z)dz = \int_z q(x, z)dz \equiv p(x) = q(x)$ and $\int_x p(x, z)dx = \int_x q(x, z)dx \equiv p(z) = q(z)$. Additionally if the joint distributions are equal the conditionals distributions are also equivalent as we can observe mathematically $q(x, z)/q(z) = p(x, z)/p(z) \equiv q(x|z) = p(x|z)$ and $q(x, z)/q(x) = p(x, z)/p(x) \equiv q(z|x) = p(z|x)$. If this occurs using deterministic conditional distributions we would obtain perfect reconstructions [19] (desired property 2). This means that the encoder distribution $q(z|x)$ would maintain the most relevant and representative features of the observed data. As we will observe mathematically this dependency of the variables in the observable and latent space will present itself as a maximization in the likelihood of the decoder or encoder. We will analyze this correlation from an information theoretic perspective in the next chapter. In this chapter we focus on the matching of joint distributions perspective.

The literature shows two different ways of matching the joint distributions. The first way is by matching the joint distributions adversarially [23, 19, 68, 20]. The second way is based on decomposing the matching of the joint distributions in simpler terms. In general this is done using the Kullback-Leibler (KL) divergence in which the order of the joint distributions matters. Decomposing $D_{KL}(q(x, z)||p(x, z))$ or $D_{KL}(p(x, z)||q(x, z))$ yields different models with different characteristics. In what follows these models are studied in detail.

4.1 Matching $q(x, z)$ and $p(x, z)$ adversarially

Matching $q(x, z)$ and $p(x, z)$ adversarially [19, 23] is based on the same objective of Generative adversarial networks (GANs) proposed in [29] (Chapter 2). This objective can be extended to match the joint distribution $q(x, z)$ and $p(x, z)$ as shown in Eq. (4.1). If the same $f(y)$, $g(y)$ and $h(y)$ of the vanilla GAN are used it can be proved [19, 23] that Eq. (4.1) minimizes the Jensen Shannon divergence of the joint distributions *i.e.* minimizes $D_{JS}(q(x, z)||p(x, z))$.

$$\begin{aligned} &\max_{D} \ \mathbb{E}_{x,\tilde{z}\sim q(x,z)}[f(D(x,\tilde{z}))] + \mathbb{E}_{\tilde{x},z\sim p(x,z)}[g(D(\tilde{x},z))], \\ &\min_{G} \ -\mathbb{E}_{x,\tilde{z}\sim q(x,z)}[h(D(x,\tilde{z}))] + \mathbb{E}_{\tilde{x},z\sim p(x,z)}[h(D(\tilde{x},z))]. \end{aligned} \tag{4.1}$$

More recently [20] combined adversarial matching of the joint distributions with adversarial matching of the marginals resulting in an overall better performance. In [68] the authors noted that matching the joint distributions adversarially have identifiability problems *i.e.* the inference and the generative model don't have the capability to perform good reconstructions. To solve the drawback of joint distribution adversarial approaches they also maximize the likelihood of the decoder and encoder in the observable space and the latent space, respectively. Veegan in [104] proposed a similar approach maximizing only of the encoder in the latent space. In Fig. 4.1 we can observe a diagram of adversarial training for joint distributions matching.

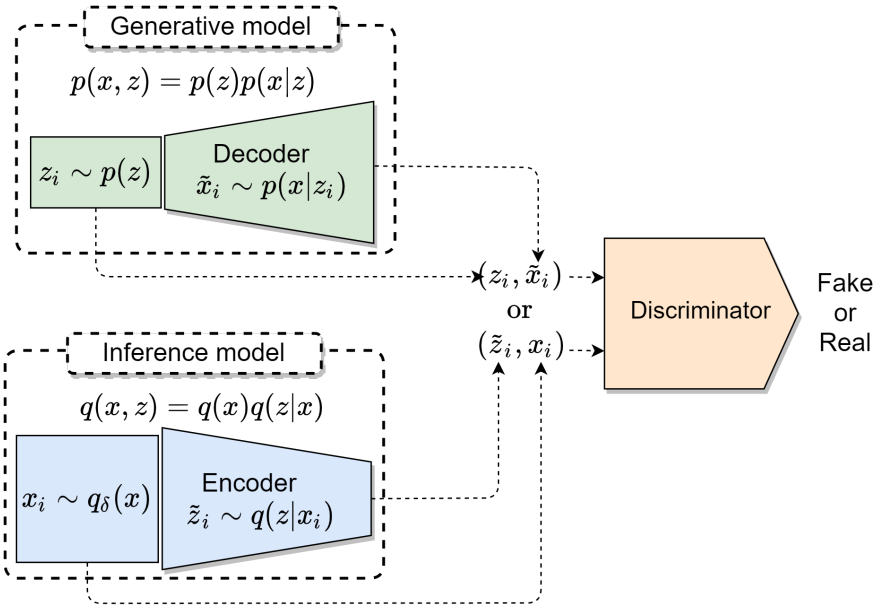


Figure 4.1: Diagram of adversarial training to match joint distributions.

In the literature this type of training has been used to match other joint distributions. In AVB [77] (section 2.2) the pairs $x_i \sim q(x), \tilde{z}_i \sim q(z|x = x_i)$ and the pairs $x_i \sim q(x), z_i \sim p(z)$ are distinguished by a discriminator, loss function that we will refer as $\mathcal{D}(q(x)q(z|x)||q(x)p(z))$. In ALICE [68], in domain adaptation experiments, this training objective is used to maximize the likelihood, distinguishing the pairs (x_i, x_i) and the pairs $(x_i, \tilde{x}_i)$, with $x_i \sim q(x), \tilde{x}_i \sim p(x|z = z_i \sim q(z|x = x_i))$. When one domain is low dimensional $\mathcal{Z}$ this type of training has no been explored, yet is still an open problem if this approach could result useful for representation learning tasks.

4.2 Minimizing $D_{KL}(q(x,z)||p(x,z))$ by decomposing it

We start by decomposing the KL divergence of the joints $q(x,z)$ and $p(x,z)$ as follows:

$$
\begin{aligned}
& D_{KL}(q(x,z)||p(x,z)) \\
&= D_{KL}(q(x)||p(x)) + \mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z|x))] && (4.2)\\
&= \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)}[-\log p(x|z)] - h_q(x) + \mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))] && (4.3)\\
&= \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)}[-\log p(x|z)] - h_q(x) - h_q(z|x) - \mathbb{E}_{q(z)}[\log p(z)] && (4.4)\\
&= \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)}[-\log p(x|z)] - h_q(x) + \mathcal{I}_q(x,z) + D_{KL}(q(z)||p(z)), && (4.5)
\end{aligned}
$$

where $h_q(x)$ is the differential entropy of the data, which we can not be optimized. $\mathcal{I}_q(x,z)$ is the mutual information of the inference model q. Eq. (4.5) is obtained by adding and subtracting the differential entropy $h_q(z)$ in Eq. (4.4), since $\mathcal{I}_q(x,z) = h_q(z) - h_q(z|x)$.

This divergence can be decomposed as in Eq. (4.3) or Eq. (4.5). Eq. (4.3) is optimized by Variational Autoencoders (VAEs) with the exception of the entropy term $h_q(x)$. β-VAE [37] and Adversarial Variational Bayes (AVB) [77] also follow Eq. (4.3) modifying the optimization of $D_{KL}(q(z|x)||p(z))$, in particular β-VAE adds a constant to the KL-divergence term and AVB replaces it with adversarial training. We note that this view has been previously discussed for VAE models [121, 69] but we expand it including models that match the marginal distributions $q(z)$, $p(z)$ as follows.

Other methods can be associated with Eq. (4.5) by replacing $D_{KL}(q(z)||p(z))$ with similar objectives. Info-VAE [120], Adversarial Autoencoders (AAEs) [75], Wasserstein Autoencoders (WAEs) [107] have followed this approach using adversarial training or maximum mean discrepancy between the marginal distributions. They don't explicitly optimize $\mathcal{I}_q(x,z)$ and thus the entropy term $h_q(z|x)$, which is included in $\mathcal{I}_q(x,z)$. The relevance of this term and its relation with the generative capabilities are discussed at the end of Chapter 5.

The first term of the $D_{KL}(q(x,z)||p(x,z))$ decomposition is $\mathbb{E}_{q_\delta(x)}\mathbb{E}_{q(z|x)}[-\log p(x|z)]$. This cross-entropy term is optimized using MSE for most models. In [68] they replace this cross-entropy using a more flexible approach like adversarial training [68]. In Fig. 4.2 we show graphically and intuitively how the decomposition of $D_{KL}(q(x,z)||p(x,z))$ is optimized.

4.3 Minimizing $D_{KL}(p(x,z)||q(x,z))$ by decomposing it

We start by decomposing the KL divergence of the joints $p(x,z)$ and $q(x,z)$ as

$$
\begin{aligned}
& D_{KL}(p(x,z)||q(x,z)) \\
&= D_{KL}(p(z)||q(z)) + \mathbb{E}_{p(z)}[D_{KL}(p(x|z)||q(x|z))] && (4.6)\\
&= \mathbb{E}_{p(z)}\mathbb{E}_{p(x|z)}[-\log q(z|x)] - h_p(z) + \mathbb{E}_{p(z)}[D_{KL}(p(x|z)||q(x))] && (4.7)\\
&= \mathbb{E}_{p(z)}\mathbb{E}_{p(x|z)}[-\log q(z|x)] - h_p(z) - h_p(x|z) - \mathbb{E}_{p(x)}[\log q(x)]. && (4.8)\\
&= \mathbb{E}_{p(z)}\mathbb{E}_{p(x|z)}[-\log q(z|x)] - h_p(z) + \mathcal{I}_p(x,z) + D_{KL}(p(x)||q(x)). && (4.9)
\end{aligned}
$$

The divergence $D_{KL}(p(x,z)||q(x,z))$ can be decomposed either as Eq. (4.7) or Eq. (4.9). Note that it is not possible to minimize $\mathbb{E}_{p(z)}[D_{KL}(p(x|z)||q(x))]$, the third term of Eq. (4.7),

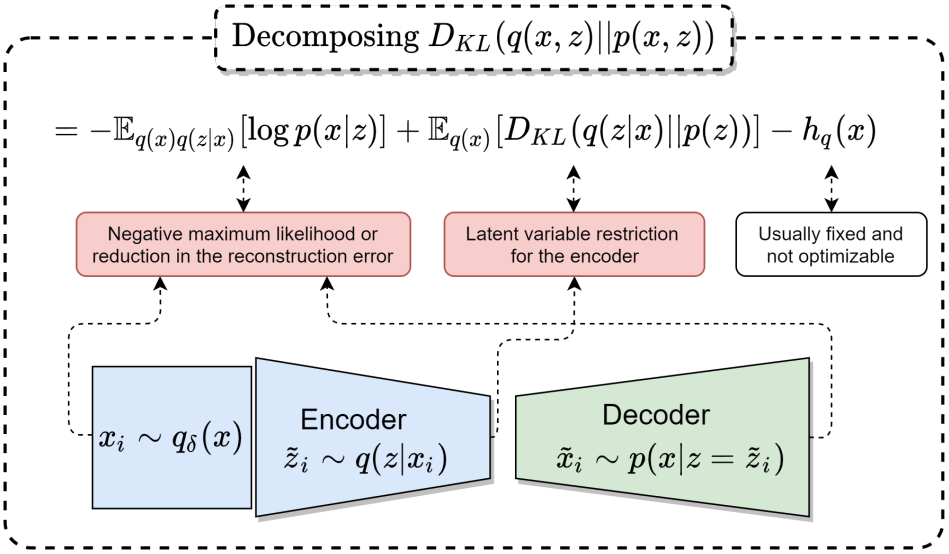


Figure 4.2: Diagram for the optimization of $D_{KL}(q(x,z)||p(x,z))$ decomposition.

using a closed form solution since we don't have access to the $q(x)$ distribution. This could be optimized using adversarial training like AVB, but to the best of our knowledge this hasn't been explored in the literature. We note that some models follow this perspective by matching the marginal distributions $p(x)$, $q(x)$ instead of all the terms of $D_{KL}(p(x,z)||q(x,z))$ as follows.

Minimizing $D_{KL}(p(x,z)||q(x,z))$ has been explored in Adversarial Inference by Matching Priors and Conditionals (AIM) [69] and in Info-GAN [14] using some components of Eq. (4.9). The first term of Eq. (4.9), $\mathbb{E}_{p_\delta(z)}\mathbb{E}_{p(x|z)}[-\log q(z|x)]$, is optimized with the MSE for the Gaussian case or the negative loglikelihood for more general cases. The second term $h_p(z)$ is the entropy of the prior, which can be fixed or optimized. The third term $\mathcal{I}_p(x,z)$ is the mutual information of the generative model that is not optimized explicitly. The last term is the Kullback-Leibler divergence $D_{KL}(p(x)||q(x))$, matching these marginal distributions is replaced by adversarial training in [69] and [14]. In Fig. 4.3 we show graphically and intuitively how the decomposition of $D_{KL}(p(x,z)||q(x,z))$ is optimized.

4.4 Which method should I use to match $q(x,z)$ and $p(x,z)$?

In theory minimizing $D_{KL}(q(x,z)||p(x,z))$, $D_{KL}(p(x,z)||q(x,z))$ or $\mathcal{D}(q(x,z)||p(x,z))$ should be equivalent in the optimal case of $q(x,z) = p(x,z)$. But in practice this is not the case because of the training procedure or the model capacity of either $q(z|x)$ or $p(x|z)$. In Chapter 6 we perform an empirical study of many generative-inference models with different training objectives. In the following we give general insights about the advantages and disadvantages of minimizing $D_{KL}(q(x,z)||p(x,z))$, $D_{KL}(p(x,z)||q(x,z))$ or $\mathcal{D}(q(x,z)||p(x,z))$.

In $D_{KL}(q(x,z)||p(x,z))$ the expected negative log-likelihood $\mathbb{E}_{q_\delta(x)}\mathbb{E}_{q(z|x)}[-\log p(x|z)]$ occurs in the observed space $\mathcal{X}$ (Eq. (4.4)), which can be difficult to minimize using MSE in high dimensional $\mathcal{X}$. Additionally, from Eq. (4.4) the term $\mathbb{E}_{q(z)}[\log p(z)]$ show that samples from $q(z)$ should have high likelihood with respect to $p(z)$. The support of $q(z)$ is increased

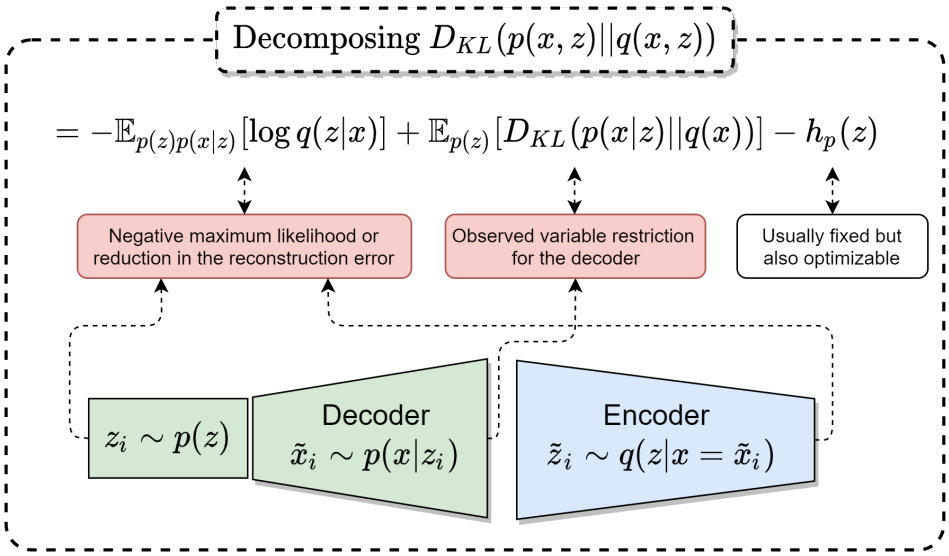


Figure 4.3: Diagram for the optimization of $D_{KL}(p(x,z)||q(x,z))$ decomposition.

by maximizing the entropy $h_q(z|x)$ from the same equation. Since the support of $q(z)$ doesn't cover all the support of $p(z)$ models that minimize $D_{KL}(q(x,z)||p(x,z))$ have better representation learning capabilities than models that minimize $D_{KL}(p(x,z)||q(x,z))$. In Chapter 5 we explain in detail this observation by associating the loss function of several models with the mutual information of their distribution.

Models that minimize $D_{KL}(q(x,z)||p(x,z))$ instead should have better generative capabilities since they match $p(x)$ and $q(x)$ directly, minimizing $D_{KL}(p(x)||q(x))$ from Eq. (4.8). In practice adversarial training is used to match $p(x)$ and $q(x)$ and can scale to more complex datasets than models that minimize $D_{KL}(q(x,z)||p(x,z))$. In conclusion models that minimize $D_{KL}(p(x,z)||q(x,z))$ are more suitable for generation. Instead, models that minimize $D_{KL}(q(x,z)||p(x,z))$ are more useful when the complexity of $\mathcal{X}$ is low (when using MSE for reconstruction error in the observed space) and the main task of interest is representation learning instead of generation.

Models that minimize $\mathcal{D}(q(x,z)||p(x,z))$ use adversarial training so these are expected to have high generative capabilities. On the other hand, previous works [19, 20, 23] have shown poor reconstruction performance for these models. In [69] they showed that their model AIM, obtained by decomposing $D_{KL}(p(x,z)||q(x,z))$, outperforms models that minimize $\mathcal{D}(q(x,z)||p(x,z))$. It is unknown if the representation learning capabilities of the models that minimize $\mathcal{D}(q(x,z)||p(x,z))$ are better than the models obtained from $D_{KL}(p(x,z)||q(x,z))$ or $D_{KL}(q(x,z)||p(x,z))$. In Chapter 6 we show empirical results comparing the generative and representation learning capabilities of generative-inference models with different training objectives.

Chapter 5

Generative-Inference models: Representation learning perspective through mutual information.

In this chapter we present a general formulation for generative-inference models from the perspective of representation learning. This formulation is obtained using the information theoretic concept of mutual information (MI). Shannon's MI between the observable variables $x \in \mathcal{X}$ and latent variables $z \in \mathcal{Z}$ is defined as

$$\begin{aligned}\mathcal{I}_r(x,z) &= \int_{\mathcal{X}}\int_{\mathcal{Z}} r(x,z)\log\frac{r(x,z)}{r(x)r(z)}\,dx\,dz \\ &= D_{KL}(r(x,z)||r(x)r(z)) \\ &= \mathbb{E}_{r(x,z)}\left[\log r(x|z)\right] - \mathbb{E}_{r(x)}\left[\log r(x)\right] \\ &= \mathbb{E}_{r(x,z)}\left[\log r(z|x)\right] - \mathbb{E}_{r(z)}\left[\log r(z)\right],\end{aligned} \tag{5.1}$$

where $r(x,z)$ is the joint distribution of x and z. The definition of MI depends on the choice of the joint distribution. In this case we have two definitions of MI, one for the joint distribution of the inference model $q(x,z)$ and one for the generative model $p(x,z)$. In what follows we will use the sub-indexes q and p to refer to quantities associated with the inference and generative distributions, respectively. The sub-index r is used to refer to quantities associated with either one of the distributions.

We start by decomposing the MI of the inference model $q(x,z) = q(x|z)q(z)$ as follows

$$\begin{aligned}\mathcal{I}_q(x,z) &= \mathbb{E}_{q(x,z)}\left[\log q(x|z)\right] - \mathbb{E}_{q(x)}\left[\log q(x)\right] \\ &= \mathbb{E}_{q(x,z)}\left[\log p(x|z)\right] - \mathbb{E}_{q(x)}[\log q(x)] + \mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))] \qquad (5.2)\\ &= \mathbb{E}_{q(x,z)}[\log p(x|z)] - \boldsymbol{\mathcal{R}}^{\text{model}}_{\boldsymbol{q}}\,(\boldsymbol{x},\boldsymbol{z}) + \boldsymbol{\mathcal{R}}^{\text{model}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z}) \\ &\quad + h_q(x) + \mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))] \\ &= \mathbb{E}_{q(x,z)}[\log p(x|z)] - \boldsymbol{\mathcal{R}}^{\text{model}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z}) + \Delta\mathcal{I}^{\text{model}}_q \qquad (5.3)\\ &= -\mathcal{L}^{\text{model}}_q + \Delta\mathcal{I}^{\text{model}}_q \qquad (5.4)\end{aligned}$$

where the loss function of the model is given by

$$\mathcal{L}_q^{\text{model}} = \mathbb{E}_{q(x,z)}[-\log p(x|z)] + \boldsymbol{\mathcal{R}_q^{\text{model}}(x, z)}, \tag{5.5}$$

and the mutual information gap of the generative-inference model loss function and the MI of the inference model is given by

$$\Delta\mathcal{I}_q^{\text{model}} = \boldsymbol{\mathcal{R}_q^{\text{model}}(x, z)} + h_q(x) + \mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))]. \tag{5.6}$$

The terms collected in $\mathcal{L}_q^{\text{model}}$ correspond to the loss function's components of the generative-inference model. The loss function is composed by the likelihood and the term $\mathcal{R}_q^{\text{model}}$ which represents a set of restrictions (regularization) for the distributions of the model. This restriction vary depending on the model and is also part of the MI gap $\Delta\mathcal{I}_q^{\text{model}}$. Later we will recognize this restriction term in generative-inference models from the literature to show that their loss functions comply with this formulation.

The MI of the generative model $p(x, z) = p(z|x)p(x)$ can be decomposed in a similar way to Eq. (5.3) as follows:

$$\begin{aligned}
\mathcal{I}_p(x, z) &= \mathbb{E}_{p(x,z)}[\log p(z|x)] - \mathbb{E}_{p(z)}[\log p(z)] \\
&= \mathbb{E}_{p(x,z)}[\log q(z|x)] - \mathbb{E}_{p(z)}[\log p(z)] + \mathbb{E}_{p(x)}[D_{KL}(p(z|x)||q(z|x))] \qquad (5.7) \\
&= \mathbb{E}_{p(x,z)}[\log q(z|x)] - \boldsymbol{\mathcal{R}_p^{\text{model}}(x, z)} + \boldsymbol{\mathcal{R}_p^{\text{model}}(x, z)} \\
&\quad + h_p(z) + \mathbb{E}_{p(x)}[D_{KL}(p(z|x)||q(z|x))] \\
&= \mathbb{E}_{p(x,z)}[\log q(z|x)] - \boldsymbol{\mathcal{R}_p^{\text{model}}(x, z)} + \Delta\mathcal{I}_p^{\text{model}} \qquad (5.8) \\
&= -\mathcal{L}_p^{\text{model}} + \Delta\mathcal{I}_p^{\text{model}}, \qquad (5.9)
\end{aligned}$$

where $\mathcal{L}_p^{\text{model}}$ and $\Delta\mathcal{I}_p^{\text{model}}$ are the generative distribution counterparts of the aforementioned terms. The decompositions in Eq. (5.4) and Eq. (5.9) associate the loss function of the model $\mathcal{L}_r^{\text{model}}$ with the MI $\mathcal{I}_r(x, z)$. In both decompositions the term $\Delta\mathcal{I}_r^{\text{model}}$ represents the gap between the loss function of the model and the MI. For a generative-inference model to learn faithful representations this gap should be small. The term $\boldsymbol{\mathcal{R}_r^{\text{model}}(x, z)} \geq 0$ is a restriction over the distributions of the model and is included in both $\mathcal{L}_r^{\text{model}}$ and $\Delta\mathcal{I}_r^{\text{model}}$. When $\mathcal{L}_r^{\text{model}}$ is minimized the restriction is also minimized. A small value for the restriction is desirable as it reduces the gap.

In the literature $\boldsymbol{\mathcal{R}_r^{\text{model}}(x, z)}$ has been utilized to match some of the distributions of the inference model and/or generative model in addition to the decoder likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ or encoder likelihood $\mathbb{E}_{p(x,z)}[\log q(z|x)]$. Models loss function that maximize $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ are bounds of $\mathcal{I}_q(x, z)$ (see Eq. (5.3)) and models loss function that maximize $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ are bounds of $\mathcal{I}_p(x, z)$ (see Eq. (5.8)). In Table 5.1 we collected all generative-inference models that are bounds of $\mathcal{I}_q(x, z)$, $\mathcal{I}_p(x, z)$ or both and present their corresponding restrictions $\boldsymbol{\mathcal{R}_r^{\text{model}}(x, z)}$. We note that models loss function that are bounds of $\mathcal{I}_q(x, z)$ tend to apply restrictions on their latent variable distributions in order to match $p(z)$ with $q(z|x)$ or $q(z)$. On the other hand models that are bounds of $\mathcal{I}_p(x, z)$ try to match their distributions in the observed space *e.g.* $q(x)$ with $p(x)$. Table 5.1 give us an idea of the current model's restrictions and the ones that are not yet explored. Note that the models that match

	Restrictions for models that bound $\mathcal{I}_q^{model}$
Models that match $q(z\|x)$, $p(z)$	$\boldsymbol{\mathcal{R}_q^{\text{VAE}}(x,z)} = \mathbb{E}_{q(x)}[D_{KL}(q(z\|x)\|\|p(z))]$ [53] $\boldsymbol{\mathcal{R}_q^{\beta-\text{VAE}}(x,z)} = \beta \cdot \mathbb{E}_{q(x)}[D_{KL}(q(z\|x)\|\|p(z))]$ [37] $\boldsymbol{\mathcal{R}_q^{\text{AVB}}(x,z)} = \mathcal{D}(q(z\|x)q(x)\|\|p(z)q(x))$ [77]
Models that match $q(z)$, $p(z)$	$\boldsymbol{\mathcal{R}_q^{\text{AAE/WAE}}(x,z)} = \mathcal{D}(q(z)\|\|p(z))$ [75, 107] $\boldsymbol{\mathcal{R}_q^{\text{Info-VAE/WAE}}(x,z)} = \text{MMD}(q(z),p(z))$ [120, 107]
Models with other restrictions	$\boldsymbol{\mathcal{R}_q^{\text{VAE-GAN}}(x,z)} = D_{KL}(q(z\|x)\|\|p(z)) + \mathcal{D}(p^+(x)\|\|q(x))$ [61] $\boldsymbol{\mathcal{R}_q^{\text{Cycle-GAN}}(x,x')} = \mathcal{D}(p(x')\|\|q(x'))$ [122]
	Restrictions for models that bound $\mathcal{I}_p^{model}$
Models that match $p(x)$, $q(x)$	$\boldsymbol{\mathcal{R}_p^{\text{AIM/InfoGAN}}(x,z)} = \mathcal{D}(p(x)\|\|q(x))$ [69, 14]
Models with other restrictions	$\boldsymbol{\mathcal{R}_p^{\text{Veegan}}(x,z)} = \mathcal{D}(p(x,z)\|\|q(x,z))$ [104]
	Restrictions for models that bound $\mathcal{I}_q^{model} + \mathcal{I}_p^{model}$
Models with other restrictions	$\boldsymbol{\mathcal{R}_q^{\text{ALICE}}(x,z)} = \mathcal{D}(p(x,z)\|\|q(x,z))$ [68] $\boldsymbol{\mathcal{R}_q^{\text{DiscoGAN}}(x,x')} = \mathcal{D}(p(x)\|\|q(x)) + \mathcal{D}(p(x')\|\|q(x'))$ [51]

Table 5.1: Restrictions of different generative-inference models. We separate models that are bounds of $\mathcal{I}_q(x,z)$, $\mathcal{I}_p(x,z)$ or both. The $\mathcal{D}$ can be replaced for any adversarial training. The variable $x' \in \mathcal{X}'$ appeared in Cycle-GAN and DiscoGAN refers to data that belongs to a high dimensional space (similar to $\mathcal{X}$). $\mathcal{D}(p^+(x)||q(x))$ refers to an adversarial training optimization with samples from the posterior $q(z|x)$ and the prior $p(z)$ [61].

joint distribution [23, 19] solely do not maximize likelihood and can not be analyzed by the proposed perspective.

As shown before several generative-inference models can be formulated as optimizing a bound on the MI $\mathcal{I}_r(x,z)$. Note that this quantity is not fixed and depends on the joint distribution that we are measuring *i.e.* $p(x,z) = p(x|z)p(z)$ or $q(x,z) = q(z|x)q(x)$. These models are learned through training but we can still obtain a measurable quantity that depends on the distributions p and q. For example, we can compute the MI of the inference model q as

$$\begin{aligned}\mathcal{I}_q(x,z) &= \mathbb{E}_{q(x,z)}\left[\log q(z|x) - \log q(z)\right] &(5.10)\\ &= \mathbb{E}_{q(x,z)}\left[\log \frac{q(z|x)p(z)}{p(z)q(z)}\right]\\ &= \mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))] - D_{KL}(q(z)||p(z)), &(5.11)\end{aligned}$$

using either Eq. (5.10) or Eq. (5.11). Eq. (5.11) allows us to know if the model's restriction augment or reduce the maximum reachable MI. Low/High MI implies less/more representation learning capabilities. Under certain assumptions it is possible to use Eq. (5.10) or Eq. (5.11) to measure the maximum inference model MI. For example, in Eq. (5.10) we can compute both entropy terms, $E_{q(x,z)}[\log q(z|x)] = -h_q(z|x)$ is the encoder entropy (commonly Gaussian) and $E_{q(z)}[\log q(z)] = -h_q(z)$ can be computed approximating $q(z)$ as a multivariate Gaussian. In Eq. (5.11) $\mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))]$ can be computed similarly to VAEs [53] and $D_{KL}(q(z)||p(z))$ can be computed by approximating $q(z)$ as a multivariate Gaussian and using the close form solution of the KL divergence between Gaussian distributions (Appendix A).

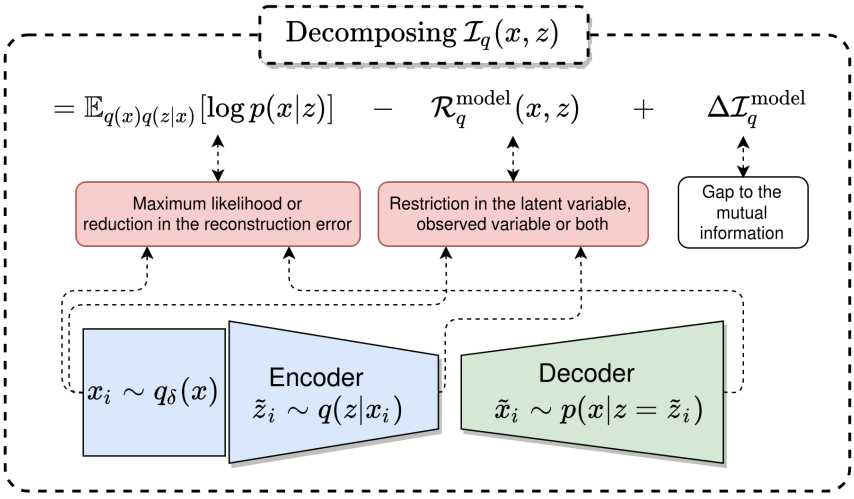


Figure 5.1: Diagram for the optimization of a variational bound of $\mathcal{I}_q(x,z)$.

The MI of the generative model p can be decomposed in a similar way to Eq. (5.11) as

$$\mathcal{I}_p(x,z) = \mathbb{E}_{p(x,z)}[\log p(x|z) - \log p(x)] \tag{5.12}$$
$$= \mathbb{E}_{p(z)}\left[D_{KL}(p(x|z)||q(x))\right] - D_{KL}(p(x)||q(x)) \tag{5.13}$$

but in this case the terms are more difficult to compute. In Eq. (5.12) we require the entropy of the marginal and conditional distribution, which are hard to compute in high dimensions. In Eq. (5.13) both $\mathbb{E}_{p(z)}[D_{KL}(p(x|z)||q(x))]$ and $D_{KL}(p(x)||q(x))$ can be estimated using adversarial training but they will be biased by the discriminator used. This bias makes the MI of the generative model unfeasible to use in practice.

Note that Eq. (5.10) or Eq. (5.12) measures the maximum reachable MI but the likelihood decreases the gap of an arbitrary model to this measure. Ignoring $\boldsymbol{\mathcal{R}_r^{\text{model}}(x,z)}$, in the case of $q = p$ we have equal conditional distributions and the gap between the likelihood and the MI will be given by the differential entropy $h_q(x) = -\mathbb{E}_{q(x)}[\log q(x)]$ for $\mathcal{I}_q(x,z)$ (see Eq. (5.2)) and $h_p(z) = -\mathbb{E}_{p(z)}[\log p(z)]$ for $\mathcal{I}_p(x,z)$ (see Eq. (5.7)). In the ideal case of $\boldsymbol{\mathcal{R}_r^{\text{model}}(x,z) = 0}$ and $p = q$ the MI of the generative and the inference model are equal and thus, in terms of representation learning, it is preferable to optimize a bound of $\mathcal{I}_p(x,z)$ since generally $h_p(z) < h_q(x)$. Given the previous theoretical analysis it is important to force the joint distributions to be equal $p = q$ and use a restriction $\boldsymbol{\mathcal{R}_r^{\text{model}}}$ that can be minimized to zero. Restrictions $\boldsymbol{\mathcal{R}_r^{\text{model}}} > 0$ will increase the gap $\Delta\mathcal{I}_r^{\text{model}}$ of the model loss function $\mathcal{L}_r^{\text{model}}$ to the mutual information $\mathcal{I}_r(x,z)$.

We expect models that optimize the bound of $\mathcal{I}_q(x,z)$ to have better representation learning capabilities than models that optimize the bound of $\mathcal{I}_p(x,z)$. Generally, models that optimize $\mathcal{I}_q(x,z)$ use restrictions of the form $\mathbb{E}_{q(x)}\left[D_{KL}(q(z|x)||p(z))\right]$ or $D_{KL}(q(z)||p(z))$ that directly affect $\mathcal{I}_q(x,z)$ (see Eq. (5.11)) *i.e.* their representation learning capability. Their generative performance is optimized indirectly by the likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ and will also depend on how well the model $q(z)$ fits $p(z)$. Symmetrically we expect models that optimize the bound of $\mathcal{I}_p(x,z)$ to have better generative capabilities than models that optimize the bound of $\mathcal{I}_q(x,z)$. Generally, models that optimize $\mathcal{I}_p(x,z)$ use restrictions

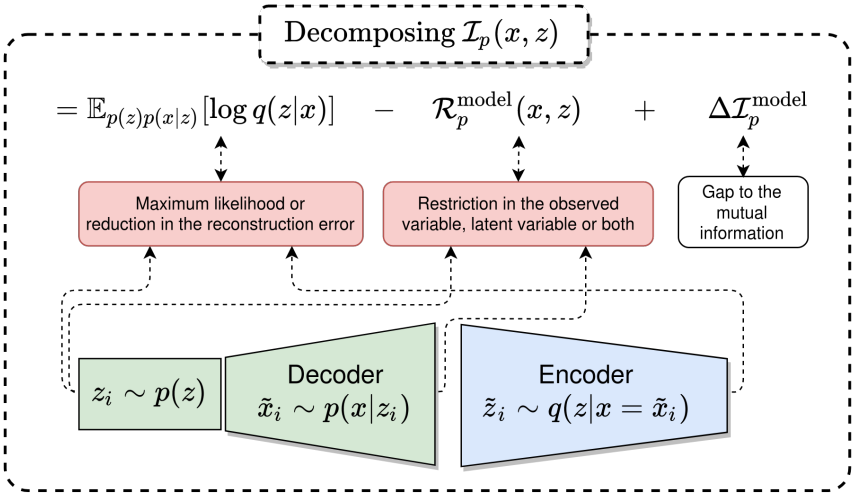


Figure 5.2: Diagram for the optimization of a variational bound of $\mathcal{I}_p(x, z)$.

of the form $\mathbb{E}_{p(z)}\left[D_{KL}(p(x|z)||q(x))\right]$ or $D_{KL}(p(x)||q(x))$ that directly optimize their generative performance. Their performance on inference is optimized indirectly by the likelihood $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ and will also depend on how well the model $p(x)$ approximates $q(x)$. Models with combined restrictions could result in better representation learning and generation capabilities. The representation learning or the generative performance in any case will depend on the limitations of the modeled $q(z|x)$, $p(x|z)$ and the restrictions imposed.

5.1 Detailed Analysis

VAEs are bad for representation learning. From Eq. (5.11) we note when $q(z|x) = p(z)$ the inference model becomes uninformative *i.e.* $\mathcal{I}_q(x, z) = 0$ since all the conditional distribution $q(z|x)$ collapse to the prior $p(z)$. The latter observation is critical for models that try to match the distributions $q(z|x)$ and $p(z)$ such as VAE, β-VAE and AVB. From a theoretical point of view these models will have a limited capacity for representation learning since they reduce the maximum MI reachable by the inference model (Eq. (5.11)). Moreover their restriction $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{model}}$ cannot reach the minimum zero value, if this were the case $q(z|x)$ should be equal to $p(z)$ obtaining an uninformative $q(z|x)$. In consequence the restriction $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{model}}$ should be positive increasing the gap $\Delta\mathcal{I}_q^{\text{model}}$ (Eq. (5.6)). From Eq. (5.11) we can note that models that match the marginal distributions $q(z)$ and $p(z)$ would likely perform better for representation learning since they increment $\mathcal{I}_q(x, z)$ by reducing $D_{KL}(q(z)||p(z))$ (Eq. (5.11)). The models AAE, WAE and Info-VAE enter in this category by bounding the inference model MI. The models AIM and InfoGAN also belong to this category as they bound the generative model MI. Based on these observations we develop new methods that match the marginals $q(z)$ and $p(z)$, which methods are presented in Section 6.2.

Matching conditional distributions by optimizing the likelihood. Without loss of generality, given a fixed mutual information $\mathcal{I}_q(x, z)$, optimizing the corresponding expected log-likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ decreases the divergence between conditional distribution $\mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))]$. This can be observed in Eq. (5.10) since $\mathbb{E}_{q(x)}[\log q(x)]$

cannot be optimized. The minimization of $\mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))]$ occurs when optimizing the decoder $p(x|z)$ under the support of $q(z)$. In practice $\mathcal{I}_q(x,z)$ is fixed in two situations: when we train a model iteratively and when q is fixed because is learned before p. In an iterative training we have a certain q and p in an arbitrary iteration and the gradients are computed under those distributions at that iteration validating our analysis in this case. When q is learned before p, $p(x|z)$ will be optimized under a fixed $q(z|x)$ and $q(z)$. In practice q is easier to learn than p since encoding a lower dimension is simpler than decoding a higher dimension. Finally, note that the symmetrical analysis of the mutual information $\mathcal{I}_p(x,z)$ is valid, the likelihood optimized in this case is $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ and the divergence minimized is $\mathbb{E}_{p(x)}D_{KL}(p(z|x)||q(z|x))$ (see Eq. (5.7)). In practice the assumption of a fixed p in this is case is more accurate since $q(z|x)$ is learned by fixing p in $\mathbb{E}_{p(x,z)}[\log q(z|x)$ [69] to avoid unstable training. In $\mathcal{I}_q(x,z)$ the backpropagation in $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ occurs in both q and p [53].

Matching joint distributions by optimizing likelihood and matching marginals. From the previous analysis maximizing the likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ is equivalent to minimizing $\mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))]$ under the support of $q(z)$. If we additionally match the marginals $q(z)$ with $p(z)$ we will be matching the conditionals and the marginals, which is equivalent to matching the joint distributions. We can also observe this from a KL Divergence perspective as shown in Eq. (4.6) or Eq. (4.8). We will later discuss about the empirical restrictions to match $p(z)$ with $q(z)$. Note that the symmetrical case is also valid, *i.e.* maximizing $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ and matching $p(x)$ with $q(x)$ enforces matching the joints.

5.1.1 Generative capabilities for models that bound $\mathcal{I}_q(x,z)$

The generative capabilities of a model are measured by how well $p(x) = \mathbb{E}_{p(z)}[p(x|z)]$ approximates $q(x)$ having only access to $q_\delta(x)$. Note that in $p(x) = \mathbb{E}_{p(z)}[p(x|z)]$ the expectation is over all the prior $p(z)$ *i.e.* we expect that each sample from the prior after decoding looks realistic. As we previously discussed, models loss function that bound $\mathcal{I}_p(x,z)$ usually optimize directly $\mathcal{D}(p(x)||q(x))$ and thus their generative performance. The generative capabilities of models that bound $\mathcal{I}_q(x,z)$ will depend on how well they fit $q(z)$ to $p(z)$ and how well they optimize the likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$. In the literature these models usually use restrictions of the form $\boldsymbol{\mathcal{R}_q^{\text{model}}(x,z)} = \mathbb{E}_{q(x)}[D_{dist}(q(z|x), p(z))]$ or $\boldsymbol{\mathcal{R}_q^{\text{model}}(x,z)} = D_{dist}(q(z), p(z))$, *dist* in this case refers to a divergence or distance between distribution. Theoretically, we will see that using one restriction or the other can have drastic effects over the generative performance for models that bound $\mathcal{I}_q(x,z)$.

Let's consider a model that bounds $\mathcal{I}_q$ and have infinite capacity. Under these assumptions $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ is maximum and the generative capabilities of the model will depend only on how well $q(z)$ fits $p(z)$. One could think that we should prefer models that match $q(z)$ directly to $p(z)$, unfortunately in practice obtaining a good approximation for $q(z) = \mathbb{E}_{q(x)}[q(z|x)]$ is hard since we only have $q_\delta(x)$ available instead of $q(x)$. In this empirical case the marginal distribution will be given by $q(z) = \mathbb{E}_{q(x)}[q(z|x)] \approx \mathbb{E}_{q_\delta(x)}[q(z|x)] = \frac{1}{N}\sum_{i=1} q(z|x = x_i)$. From the last equality we note that the model should approximate $p(z)$ relying on finite data from $q_\delta(x)$ and the stochasticity of the encoder distribution $q(z|x)$. If the entropy of $q(z|x)$ is low the model won't achieve this objective since it won't be considering all the values that $p(z)$ could take. Models that match the marginals like AAE, WAE, or Info-VAE don't consider

this entropy term $h_q(z|x)$ and thus their optimization to match $p(z)$ with $q(z)$ reduces to estimating $q(z)$ using only the first statistical moment of $q(z|x)$. From this observation, methods like VAE and β-VAE will have better generation capabilities since they enforce the maximization of $h_q(z|x)$.

Although the increment of $h_q(z|x)$ enforces a bigger support of $q(z)$ it also has negative consequences in representation learning. We can observe this drawback in two manners. The first and simplest is by observing the definition of the mutual information $\mathcal{I}_q(x,z) = D_{KL}(q(x,z)||q(x)q(z))$. The mutual information is the KL divergence between $q(x,z)$ and $q(x)q(z)$. Informally the MI is trying to distinguish the samples $x_i, \tilde{z}_i \sim q(x,z)$ from the samples of $x_i, \tilde{z}_i \sim q(x)q(z)$. If we increase the entropy $h_q(z|x)$ then it is going to be harder to distinguish samples from both distributions lowering the mutual information. The second way to note the drawback of adding the entropy term is by observing Eq. (5.11). We can note that if we choose a new restriction that considers to maximize an additional entropy term $h_q(z|x)$ *i.e.* $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{new-model}} = \boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{model}} - h_q(z|x)$ the gap $\Delta\mathcal{I}_q^{\text{model}}$ to the mutual information will increase. This is because in general the differential entropy $h_q(z|x)$ is negative (the entropy of the conditional distribution is low) and $-h_q(z|x)$ increments the gap to the mutual information $\Delta\mathcal{I}_q^{\text{model}}$. Moreover it also minimizes the maximum MI reachable since $\mathcal{I}_q = h_q(z) - h_q(z|x)$ (Eq. (5.11)).

We observe that maximizing $h_q(z|x)$ affects the MI of the inference model and also increases the gap of the model loss function $\Delta\mathcal{I}_q^{\text{model}}$. This in general should affect the representation learning capabilities of the model *i.e.* its ability to separate classes in the latent space. However, representation learning is only going to be affected if the codified data from different classes are close in the latent space. This occurs if we use an unimodal prior such as $\mathcal{N}(0, I)$ since in some frontier of the latent space the data from different classes would be close. On the contrary we would like that data from the same class to be close together in the latent space so $p(x)$ properly generalizes $q(x)$ having only $q_\delta(x)$ available. We could use a multimodal prior (studied in detail in Chapter 7) to model and enforce similar features to be close in the latent space and dissimilar features to be far. In a multimodal prior the increment of $h_q(z|x)$ shouldn't affect the representation learning capabilities since it would be enforcing the smoothness of data with similar features.

5.2 Related methods

The most similar work in the literature is [121] where they express the loss function of different generative-inference models with the MI and consistency terms of their distributions. In comparison we treat the loss functions of many generative-inference models as a bound of the MI, which is fundamentally different to [121]. This different view allows us to extract additional conclusions about the behaviour of the generative-inference models. Thanks to our interpretations we can expand the generative-inference models of the literature as shown in Chapter 6.

Another interpretation from [121] that differs from our work, is treating β-VAE [37] as a model that minimizes the mutual information between the observed variables $x \in \mathcal{X}$ and the latent variables $z \in \mathcal{Z}$. In [121] the loss function of β-VAE is expressed as $\mathcal{L}_{\beta-\text{VAE}} = (\beta-1)\mathcal{I}_q(x,z) + \beta D_{KL}(q(z)||p(z)) + \mathbb{E}_{q(z)}[D_{KL}(q(x|z)||p(x|z))]$, being the divergences terms

consistency constrains. Following this it is affirmed that the minimization of $\mathcal{L}_{\beta-\text{VAE}}$ with $\beta > 1$ implies a minimization on $\mathcal{I}_q(x, z)$. From the perspective of our work this is not necessarily true. As previously discussed, under reasonable empirical assumptions the optimization of $D_{KL}(q(x|z)||p(x|z)$ is equivalent to the maximization of $\mathbb{E}_{q(x)q(z|x)}[\log p(x|z)]$ and thus it affects the optimization of $q(z|x)$. The encoder $q(z|x)$ is the only optimizable component of $q(x, z)$, which defines $\mathcal{I}_q(x, z)$. Therefore if $\mathcal{I}_q(x, z)$ is actually minimized will depend on how $\mathbb{E}_{q(x)q(z|x)}[\log p(x|z)]$ scales with respect to $(\beta - 1)\mathcal{I}_q(x, z)$.

5.3 Discussion

Matching the joint or marginal distributions together with the optimization of the generative model MI bound seems promising for models with good representation learning performance. Models like AIM and Veegan follow this optimization procedure so it is interesting if their generative and inference capabilities translate according to this observation in practice. Note that when we maximize the likelihood with the same marginals (or models that enforce this condition) we are also matching the joint distribution. Enforcing the joint distributions to be equal using different approaches seems promising in terms of representation learning and generation. The most similar methods in the literature that uses this kind of approach are ALICE [68] and Veegan [104]. We will test these and other generative-inference models in the next Chapter. Finally it is worth to mention that all the analysis made in this chapter comes from a theoretical point of view. In practice it is relevant to explore if the model can actually minimize the expected log-likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$, $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ or KL divergence of the marginals $D_{KL}(q(z)||p(z))$, $D_{KL}(p(x)||q(x))$.

Chapter 6

Generative-Inference models: Empirical analysis and proposed models

In Chapter 5 we studied from a theoretical point of view the generative inference models loss functions and their relationship with the mutual information of their distributions. In this chapter we observe empirically how this theory translates into practice. For this purpose we implement various generative-inference models using the same graphical model for the distributions $p(x|z)$ and $q(z|x)$ for fair comparison.

In section 6.1 we enunciate the possible limitations of generative-inference models in practice. Motivated by the relevance of matching marginal distributions, studied in the previous chapter, in section 6.2 we propose new generative inference models. In section 6.3 we test various generative-inference models with a focus on the trade-off between generation and representation learning capabilities

6.1 Practical considerations of generative-inference's loss functions

The generative-inference models loss function that can appear when we decompose the matching of their joint distributions or when we associate them with the mutual information of their distributions are $\mathbb{E}_{q(x,z)}[\log p(x|z)]$, $\mathbb{E}_{p(x,z)}[\log q(z|x)]$, $D_{KL}(p(x|z)||q(x))$, $D_{KL}(p(x)||q(x))$, $D_{KL}(q(z|x)||p(z))$ or $D_{KL}(q(z)||p(z))$. In the following we will study the assumption made to compute these loss functions.

In practice $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ usually is computed using l_2 norm [53] (MSE) or l_1 norm [68]. When optimizing in this way it is assumed that each of the input neurons are independent of each other. This assumption can drastically affect the optimization of $p(x|z)$ for input data x with correlated neurons (for example, images). Some previous work [22] has assumed a covariance for the distribution $p(x|z)$, however this estimation can be difficult to estimate in high dimensional data. This limitation is present in all models that optimize $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ *i.e.* models that bound $\mathcal{I}_q(x,z)$. In practice we will use the MSE in the observed space as an estimation of $\mathbb{E}_{q(x,z)}[\log p(x|z)]$.

The likelihood $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ in practice has been computed using l_1 norm [68], l_2 norm [104] or Gaussian likelihood [69] (Eq. (6.1), for diagonal Gaussian encoder $\mathcal{N}(\tilde{u}, \tilde{\sigma}^2) = q(z|x)$). Each of these loss functions assume neuron independence across the J dimensions of the latent space, similarly for MSE in the observed space. Although this is a strong assumption in the observed space, for latent variable is not restrictive since the model learn how to codify the data x in the latent dimension $\mathcal{Z}$ using $q(z|x)$. In practice we use Gaussian likelihood (Eq. (6.1)) to estimate the likelihood as follows

$$\mathbb{E}_{p(z)p(x|z)}[-\log q(z|x)] = \mathbb{E}_{z_i \sim p(z), \tilde{x}_i \sim p(x|z=z_i)} \left[\sum_{j=1}^{J} \frac{1}{2} \log(2\pi\tilde{\sigma}_{ij}^2) + \frac{(z_{ij} - \tilde{\mu}_{ij})^2}{2\tilde{\sigma}_{ij}^2} \right], \tag{6.1}$$

The term $D_{KL}(p(x|z)||q(x))$ has not been optimized in the literature but this can be done in a similar way as AVB [77], using adversarial training. In practice adversarial training has been used in GAN models to optimize the term $D_{KL}(p(x)||q(x))$ minimizing not necessarily the KL divergence, using implicit training and not assuming a distribution in $p(x|z)$. Given the current success of GAN models in complex datasets we hypothesize that GANs will have better generation and representation learning capabilities than autoencoding models that maximize $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ and assuming a distribution in $p(x|z)$. Note we don't consider models that maximize the likelihood in the observed space hierarchically since they correspond to a different graphical model, moreover the architecture of these models are drastically different to GAN models.

Recent contributions [15] have shown improvement in the generation capabilities of VAE models using such hierarchically optimization.

The term $D_{KL}(q(z|x)||p(z))$ usually is optimized in VAE models, in [53] it is assumed that both distributions are diagonal. Previous contributions [94, 55] have improved the flexibility of the posterior $q(z|x)$. For simplicity we will consider $q(z|x)$ as diagonal Gaussian for all experiments.

Finally the term $D_{KL}(q(z)||p(z))$ has been optimized using adversarial training [75, 107] or using Maximum mean discrepancy [120, 107]. Minimizing $D_{KL}(q(z)||p(z))$ is limited by the estimation of $q(z)$ that is computed using a finite amount of $x \sim q_\delta(x)$ as we discussed in the previous chapter. We note that closed form solution models to minimize $D_{KL}(q(z)||p(z))$ has not been explored in the literature so we propose them in the next section.

6.2 Wasserstein Variational Autoencoders and others

As we have discussed in the previous chapter, matching the marginals distributions is relevant to maximize the mutual information of the inference or generative model. Although several models in the literature match the marginals adversarially like AAE [75], WAE [107] or AIM [69] none of these models use closed form solutions. We note that matching the marginals in the observable space is not feasible since $q(x)$ is not known. In contrast this is possible for the latent variable by making small assumptions. Motivated by [123] we can obtain the first

two statistics of $q(z)$ given a certain data batch as

$$\mu(z) \;=\; \mathbb{E}_{q(z)}[z] \;=\; \frac{1}{N}\sum_{i=1}^{N} z_i \tag{6.2}$$

$$\mathrm{Cov}(z,z) \;=\; \mathbb{E}_{q(z)}[(z-\mu(z))(z-\mu(z))^T] \;=\; \frac{1}{N-1}\sum_{i=1}^{N}[(z_i-\mu(z))(z_i-\mu(z))^T] \tag{6.3}$$

where Eqs. 6.2 and 6.3 correspond to the mean and the covariance of $q(z)$, respectively. In these equations z_i is a vector sampled from $q(z) = \mathbb{E}_{q_\delta(x)}[(q(z|x)]$ and the sub-index i represents a particular data sample in the batch. In practice we can use the μ's from the posterior distribution $q(z|x = x_i)$ for each data $x_i \sim q_\delta(x)$ in the batch. One option to match these first two statistics to the prior distribution $p(z) = \mathcal{N}(0, I)$ is to minimize the Kullback Leibler divergence $D_{KL}(q(z)||p(z))$, $D_{KL}(p(z)||q(z))$ or even $D_{JS}(q(z)||p(z))$. The KL between two Gaussian distributions has the following closed form

$$D_{KL}(\mathcal{N}(\mu_1,\Sigma_1)||\mathcal{N}(\mu_2,\Sigma_2)) = \frac{1}{2}\bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - D + (\mu_1-\mu_2)\Sigma_2^{-1}(\mu_1-\mu_2)^T + \mathrm{Tr}\{\Sigma_2^{-1}\Sigma_1\}\bigg]. \tag{6.4}$$

Another option is to minimize the Wasserstein distance $\mathcal{W}_2$ [112] between $q(z)$ and $p(z)$. The Wasserstein distance between two Gaussian distributions has the following closed form

$$\mathcal{W}_2(\mathcal{N}(\mu_1,\Sigma_1),\mathcal{N}(\mu_2,\Sigma_2)) = ||\mu_1-\mu_2||^2 + \mathrm{Tr}(\Sigma_1 + \;\Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}). \tag{6.5}$$

We call Wasserstein Variational Autoencoder (WVAE) to the model that use the restriction $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{WVAE}} = \mathcal{W}_2(q(z),p(z))$ and maximizes the likelihood in the observable space $\mathcal{X}$ *i.e.* its loss function is given by $\mathcal{L}^{\text{WVAE}} = \mathbb{E}_{q(x,z)}[\log p(x|z)] + \mathcal{W}_2(q(z),p(z))$. Since this model optimizes $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ it is a bound of $\mathcal{I}(x,z)$. Note that WVAE uses a closed-form solution to match the marginal distributions $q(z)$ and $p(z)$ that differs from WAE [107] that uses adversarial training or MMD.

We can also obtain other models following a similar approach: Jensen Shannon Variational Autoencoder (JSVAE) that minimizes $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{JS-VAE}} = D_{JS}(q(z)||p(z))$, inverse Kullback Leibler Variational Autoencoder (IKL-VAE) that minimizes $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{I-VAE}} = D_{KL}(q(z)||p(z))$ and forward Kullback Leibler Variational Autoencoder (FKL-VAE) that minimizes $\boldsymbol{\mathcal{R}}_{\boldsymbol{q}}^{\text{F-VAE}} = D_{KL}(p(z)||q(z))$. We will compare WVAE and JS-VAE against other generative-inference models with a focus on the trade-off of generation and inference. We don't test IKL-VAE or FKL-VAE since their loss function are collected by JS-VAE.

6.3 Experiments

6.3.1 Dataset

We evaluate various generative-inference models in three benchmarks datasets: a handwritten digit dataset (MNIST, [63]), one color image dataset (CIFAR-10 [58]) and a fashion products image dataset (Fashion-MNIST, [115]).

6.3.2 Evaluation metrics

We measure the quality of the generated samples of all generative-inference models using the Fréchet inception distance (FID) [36] and the inception score (IS) [98]. For representation learning we train a linear classifier on top of the frozen representation of the encoded variables. For the computation of these metrics the labels y are required. Only these supervised metrics will be given as an oracle for the generative-inference model performance. In practice however these metrics are not useful for choosing a model if we are training in an unsupervised way.

In Chapter 5 we observed many unsupervised loss functions that appear by associating generative-inference models with the mutual information of their distributions. The mutual information can be computed assuming $q(z)$ as Gaussian as discussed in the previous chapter. We called this estimation as $\tilde{\mathcal{I}}_q(x, z)$, note $\tilde{\mathcal{I}}_q(x, z)$ should be ≥ 0 but in practice this could be negative when the Gaussian assumption of $q(z)$ is no longer valid. Other metrics that appears to be useful are $D_{KL}(q(z)||p(z))$ and $h_q(z|x)$ since they are obtained by decomposing $\tilde{\mathcal{I}}_q(x, z)$. The term $D_{KL}(q(z)||p(z))$ can be computed assuming $q(z)$ Gaussian similarly to $\tilde{\mathcal{I}}_q(x, z)$.The entropy $h_q(z|x)$ is easier to compute since $q(z|x)$ is assumed Gaussian generally. Finally the likelihoods $\mathbb{E}_{q(x,z)}[\log p(x|z)]$, $\mathbb{E}_{p(x,z)}[\log q(z|x)]$ give us an insight of the codification and decodification capabilities of $q(z|x)$ and $p(x|z)$.

As we will observe the relevance of these unsupervised terms will give us a close insight of how well our model perform in generation or representation learning empirically. We want to emphasize the relevance of the unsupervised metrics obtained in the training set and the metrics obtained in test set. The test metrics measure the ability of generalization of the models but they are not usually available, unless we separate data for this purpose. We found the unsupervised metrics obtained in the training are equally informative and can be used in practice.

6.3.3 Architecture details

We compare various generative-inference models using the same architecture for the estimation of $p(x|z)$ and $q(z|x)$ for fair comparison. We test various models autoencoding models like VAE [53], Info-VAE [120] and the proposed models from section 6.2. We also include GAN models: ALI [23], ALICE [68], Veegan [104] and AIM [69]. In the following we described the general architecture for these models, additionally for GAN models we include the discriminator architecture.

We use the BigGAN model techniques [8] as a base for all our experiments. We employ ResNet [34] architectures and Spectral Normalization [78]. The residual block components of the generator and discriminator/encoder are shown in Fig. 6.1 (a) and (b), respectively. We only use discriminator architectures for GANs based models, we called this discriminator as D_x. All the 3×3 Conv use a padding equal to one, while 1×1 Conv have no padding. The upsampling operation of the generator residual block is done using bilinear interpolation ("Resblock up" in Table 6.1). The downsampling operation of the encoder residual block is done using average pooling with kernel size two ("Resblock down" in Table 6.2). A general scheme of the generator, discriminator and encoder architecture is shown in Fig. 6.2. The first residual block of the discriminator/encoder inverts the order of the 1×1 Conv and the average pooling and omits the first ReLU activation. Residual blocks with an asterisk

correspond to the ones that do not perform average pooling and as a consequence they do not use 1×1 Conv. We can write the architectures for all datasets in a general way as in Fig. 6.2 or more specific as in Tables 6.1, 6.2 and 6.3, where C, J and D change between datasets. For MNIST and FMNIST $C = 32$, $J = 10$ and $D = 1$. For CIFAR10 $C = 96$, $J = 128$ and $D = 3$.

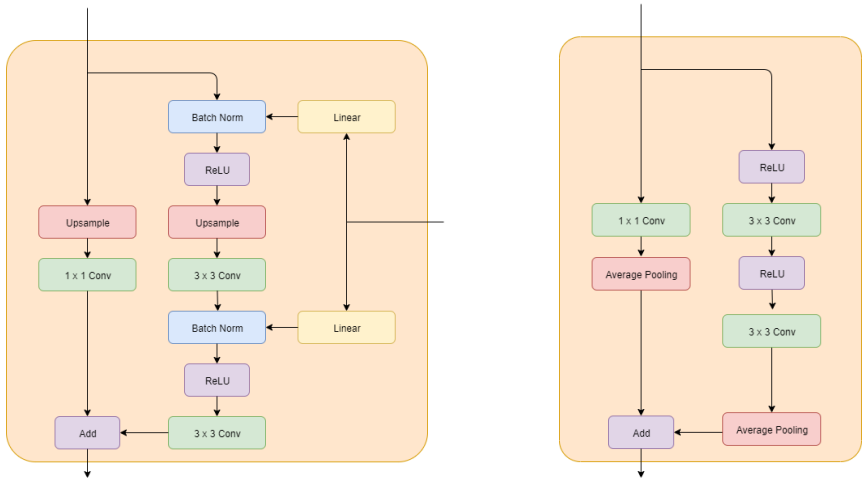
For models that estimate joint distribution like ALI, ALICE and Veegan we also considered two additional discriminators with the same architectures. These discriminators consist of one residual block with two NN with $8C$ hidden units and a last hidden layer with one neuron as output, we additionally use skip connection. One discriminator codify z and the other discriminate the pair of point $(\tilde{x}, z) \sim p(x, z)$ from the pair $(x, \tilde{z}) \sim q(x, z)$, let's call these discriminators as D_z and D_{xz} respectively. The input of D_{xz} is the concatenation of the last features of D_x and D_z.

We use the Adam optimizer [52] with its default parameters $\beta_1 = 0$ and $\beta_2 = 0.999$. We use a learning rate of $2e-4$ for all networks and experiments. All generator, discriminator and encoder parameters use Spectral Normalization and are initialized with $\mathcal{N}(0, 0.02I)$. For evaluation we use standing statistics [8] *i.e.* in evaluation mode we run many times (in our case 16) the forward propagation of the generator model $\tilde{x} \sim p(x, z)$ storing the means and variances aggregated across all forward.

The latent z is used to estimate the parameters of batch normalization layers and can be associated as form of conditional batch normalization [24]. We use the same z in each generator block (see Fig. 6.1) and different linear transformations represented in the yellow boxes of Fig. 6.1.

We develop a Pytorch implementation based on the implementation of Big-GAN [1]. The IS and FID scores are calculated as explained in section 2.3 and training a classifier with the train set of the corresponding datasets (MNIST and FMNIST). The architecture of the classifier is the same that the encoder except that the last linear transformation correspond linear with Softmax activation and the number of dimensions in the output is equal to the number of classes instead of J. To obtain the statistics of FID the test set is used. The architecture for the CIFAR10 dataset is the same inception network trained in imagenet [18] but for pytorch implementation.

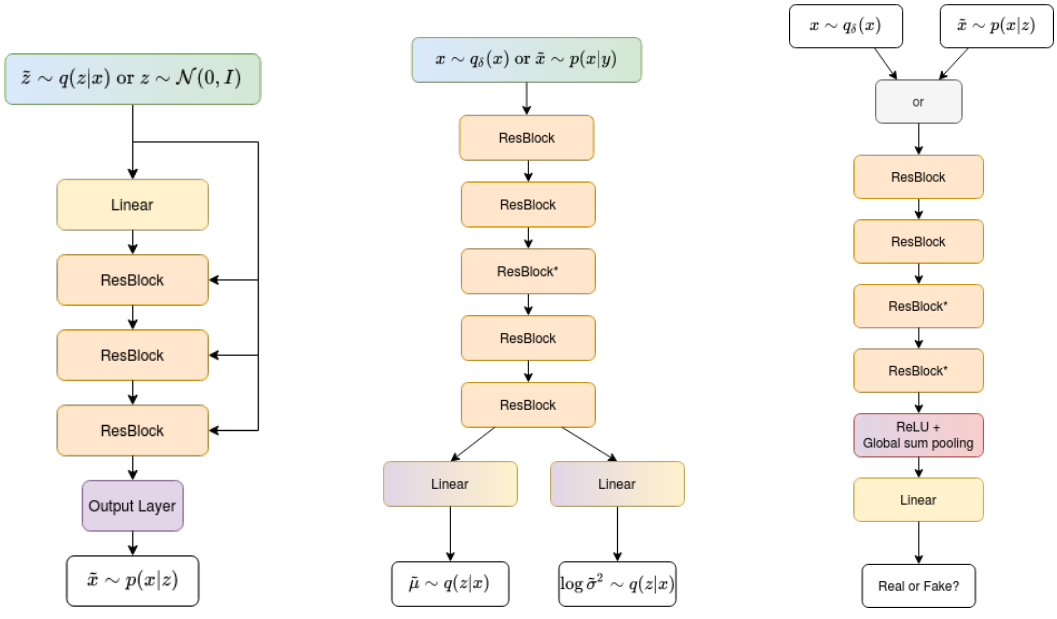
[1] `https://github.com/ajbrock/BigGAN-PyTorch`



(a) Generator block

(b) Discriminator/Encoder block

Figure 6.1: Residual blocks used in generator, discriminator and encoder networks.



(a) Generator

(b) Encoder

(c) Discriminator

Figure 6.2: Architectures for the generator, encoder and discriminator networks, respectively. The input of the generator and encoder may vary depending on the model *i.e.* GAN based or not. Note that discriminator network is only used in GAN based models.

$\tilde{z}_i \in \mathbb{R}^J \sim \mathcal{N}(\tilde{\mu}_i, \tilde{\sigma}_i^2)$ or $z_i \in \mathbb{R}^J \sim \mathcal{N}(0, I)$
Linear$(J) \to 4 \times 4 \times 4C$
Resblock up $4C \to 4C$
Resblock up $4C \to 4C$
Resblock up $4C \to 4C$
Output Layer: BN, ReLU, 3×3 Conv $C \to D$ Tanh

Table 6.1: Generator

$x \in \mathbb{R}^{32 \times 32 \times D}$
Resblock down $D \to 4C$
Resblock down $4C \to 4C$
Resblock $4C \to 4C$
Resblock down $4C \to 4C$
Resblock down $4C \to 4C$
Flatten
$\times 2$: Linear $(32//2^4 \times 4C) \to J$

Table 6.2: Encoder

$x \in \mathbb{R}^{32 \times 32 \times D}$
Resblock down $D \to 4C$
Resblock down $4C \to 4C$
Resblock $4C \to 4C$
Resblock $4C \to 4C$
ReLU, Global sum pooling
(linear $\to 1$)

Table 6.3: Discriminator

6.4 Results

The estimation of the distributions $q(z|x)$ and $p(x|z)$ depends drastically on the complexity of the dataset (see sections 6.4.1 and 6.4.3), the number of dimensions of $\mathcal{Z}$ (see section 6.4.2) and the architecture of $q(z|x)$ and $p(x|z)$. We choose a state of the art architecture (see section 6.3.3) for both VAE and GAN models that has a sufficiently high capacity for the estimation of $q(x)$.

6.4.1 Datasets with low complexity

Table 6.4 shows the most relevant metrics for the MNIST dataset computed in the training set. For this simple dataset we assume that the distribution $q(z|x)$ and $p(x|z)$ have enough capacity to maximize the likelihood $\mathbb{E}_{q(x,z)}[\log q(z|x)]$, $\mathbb{E}_{p(x,z)}[\log p(x|z)]$ or for the estimation $\mathcal{D}(q(x)||p(x))$ (or any GAN model).

From Table 6.4, as we discussed in the previous chapter, models that optimize the bound $\mathcal{I}_q(x, z)$ tend to have better representation learning capabilities and models that bound $\mathcal{I}_p(x, z)$ tend to have better generative capabilities. We note from Table 6.4 that any model

with better generative capabilities perform worse in representation learning and any model with better representation learning tend to have worse generative capabilities. More generally for any model that bounds $\mathcal{I}_q(x,z)$, $\mathcal{I}_p(x,z)$ or both a higher entropy $h_q(z|x)$ translates in better generative capabilities. We note that this also occurs for GAN models that do not maximize necessarily $h_q(z|x)$. We hypothesize that for GAN models $h_q(z|x)$ tells us about how the encoder $q(z|x)$ compress the data in the prior distribution. Autoencoding models that do not maximize $h_q(z|x)$ have lower $h_q(z|x)$ (see Table 6.4) which translates in better representation learning capabilities. VAE models minimize $D_{KL}(q(z|x)||p(z))$ augmenting $h_q(z|x)$ and improving generative capabilities despite their worse representation learning capabilities.

Theoretically we seek for a high $\mathcal{I}_q(x,z)$ and a model with high $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ so the gap is closer (see Eq. (5.3)) to the mutual information $\mathcal{I}_q(x,z)$. We observe the generative and representation learning capabilities of some generative-inference models in Figures 6.3 and 6.4, respectively. Note that ALI is the only method that can not be considered in the analysis made in Chapter 5 and is the only with $\mathcal{I}_q(x,z) < 0$. We can observe the general tendencies of all generative inference models in Figures 6.5 and 6.6 for representation learning and generative capabilities, respectively. From these results we can have a close idea of the generative and representation learning capabilities of a model using unsupervised metrics in the training set when simpler datasets are used. This general tendency is observed in FMNIST (Appendix B) although the supposition of the adequate behaviour of $q(z|x)$ and $p(x|z)$ tend to vanish, *i.e* more models have $\mathcal{I}_q(x,z) < 0$ or worse reconstructions. These tendencies can happen when we augment z, since the estimation of $\mathcal{I}_q(x,z)$ is worse, or when more complex datasets are considered since it is harder for $q(z|x)$, $p(x|z)$ to encode or decode these distributions.

Finally we observe that the proposed WVAE have a similar behaviour than VAE and Info-VAE (MMD), where the latter is a direct competitor of WVAE since both match the marginal distributions. Table 6.4 shows that both WVAE and Info-VAE (MMD) have similar test accuracies but WVAE have a much better trade-off in terms of generation. It is worth mentioning that VAE has better generative performance and worse representation learning capabilities as the theory developed in Chapter 5 indicates.

Model	**LL in $\mathcal{X}$**	LL in $\mathcal{Z}$	$D_{KL}^{\mathcal{Z}}$	D_{KL}^{VAE}	$\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$	$h_q(z\|x)$	IS	FID	Acc%
VAE ($\beta = 0.33$)	-4.57	-3.73	0.10	27.76	27.68	-13.44	8.95	223.76	91.37
	± 0.02	± 0.12	± 0.03	± 0.18	± 0.15	± 0.07	± 0.01	± 8.52	± 0.73
VAE ($\beta = 1.00$)	-4.43	-3.62	0.19	20.94	20.72	-6.89	9.36	105.89	87.86
	± 0.02	± 0.08	± 0.09	± 0.21	± 0.21	± 0.14	± 0.02	± 2.75	± 0.60
VAE ($\beta = 30.00$)	-1.90	-0.34	34.52	1.18	-33.32	12.98	3.03	4714.07	56.06
	± 0.01	± 0.01	± 1.15	± 0.05	± 1.17	± 0.01	± 0.07	± 75.28	± 1.50
WVAE ($\beta = 0.33$)	-4.66	-3.51	0.03	62.26	62.28	-48.14	8.59	351.22	93.63
	± 0.02	± 0.06	± 0.00	± 0.52	± 0.52	± 0.52	± 0.04	± 5.04	± 0.47
WVAE ($\beta = 1.00$)	-4.64	-3.50	0.16	62.69	62.72	-48.88	8.62	342.92	94.00
	± 0.03	± 0.03	± 0.06	± 0.26	± 0.25	± 0.32	± 0.03	± 13.82	± 0.13
WVAE ($\beta = 30.00$)	-4.61	-3.09	0.14	62.70	62.71	-48.86	8.19	518.53	93.50
	± 0.01	± 0.01	± 0.01	± 0.33	± 0.33	± 0.34	± 0.13	± 30.97	± 0.28
MMD ($\beta = 0.33$)	-4.71	-0.74	2.01	85.04	65.82	-43.40	6.45	1483.71	94.86
	± 0.04	± 0.02	± 0.02	± 1.21	± 0.35	± 0.33	± 0.14	± 103.37	± 0.02
MMD ($\beta = 1.00$)	-4.68	-1.28	1.73	74.43	64.97	-44.60	6.76	1151.08	94.45
	± 0.01	± 0.07	± 0.02	± 0.74	± 0.29	± 0.40	± 0.33	± 135.75	± 0.24
MMD ($\beta = 30.00$)	-4.67	-3.50	0.37	63.80	63.44	-48.89	8.27	461.61	93.86
	± 0.01	± 0.12	± 0.03	± 0.50	± 0.53	± 0.40	± 0.08	± 20.17	± 0.57
JS-VAE ($\beta = 0.33$)	-4.59	-3.31	0.12	61.58	61.58	-47.09	8.43	436.72	93.86
	± 0.02	± 0.18	± 0.02	± 0.47	± 0.46	± 0.43	± 0.02	± 5.33	± 0.04
JS-VAE ($\beta = 1.00$)	-4.56	-3.27	0.01	61.64	61.67	-47.45	8.49	403.08	92.68
	± 0.02	± 0.07	± 0.00	± 0.56	± 0.56	± 0.58	± 0.09	± 18.21	± 0.41
JS-VAE ($\beta = 30.00$)	-3.49	-1.89	0.07	67.25	67.26	-52.89	8.07	815.19	80.32
	± 0.64	± 0.71	± 0.02	± 6.78	± 6.78	± 6.73	± 0.03	± 213.64	± 8.62
AIM	-3.05	-2.48	1.36	10.58	9.70	3.06	9.79	25.92	88.83
	± 0.01	± 0.01	± 0.19	± 0.09	± 0.06	± 0.23	± 0.01	± 1.62	± 0.79
Veegan	-2.54	-1.73	1.09	9.94	8.91	4.09	9.72	38.63	81.35
	± 0.07	± 0.11	± 0.15	± 0.18	± 0.25	± 0.22	± 0.02	± 6.12	± 6.30
ALI	-1.54	-0.33	27.63	21.92	-5.58	-7.65	8.82	388.02	51.73
	± 0.37	± 0.17	± 38.14	± 16.17	± 53.86	± 15.94	± 1.24	± 465.66	± 28.48
ALICE	-3.04	-1.84	0.50	15.00	14.42	-0.89	9.69	46.62	89.13
	± 0.14	± 0.20	± 0.07	± 0.62	± 0.69	± 0.76	± 0.02	± 2.70	± 0.54

Table 6.4: Relevant metrics for the generative-inference models considered in MNIST dataset with $J = 10$. LL in $\mathcal{X}$ and LL in $\mathcal{Z}$ refer to the MSE in the observed and latent space respectively. $D_{KL}^{\mathcal{Z}} = D_{KL}(q(z)||p(z))$ is computed in closed form assuming $q(z)$ as multivariate Gaussian. $D_{KL}^{\text{VAE}} = D_{KL}(q(z|x)||p(z))$. In **bold** we represent the main unsupervised learning metrics of our study; $\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$ measures the MI of the inference model q and **LL in $\mathcal{X}$** measures the likelihood in $\mathcal{X}$.

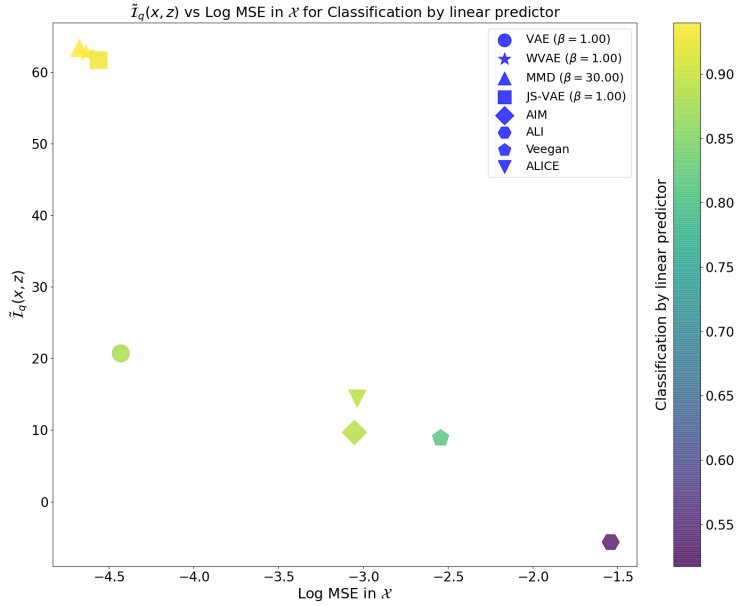


Figure 6.3: $\tilde{I}_q(x, z)$ vs Log MSE in the observed space. The color corresponds to the classification accuracy of a linear predictor trained on the latent space learned from MNIST with $J = 10$. The scatters show the mean of the three runs for each model. For VAE, WVAE, JS-VAE, MMD we chose the β that obtained the best result according to FID.

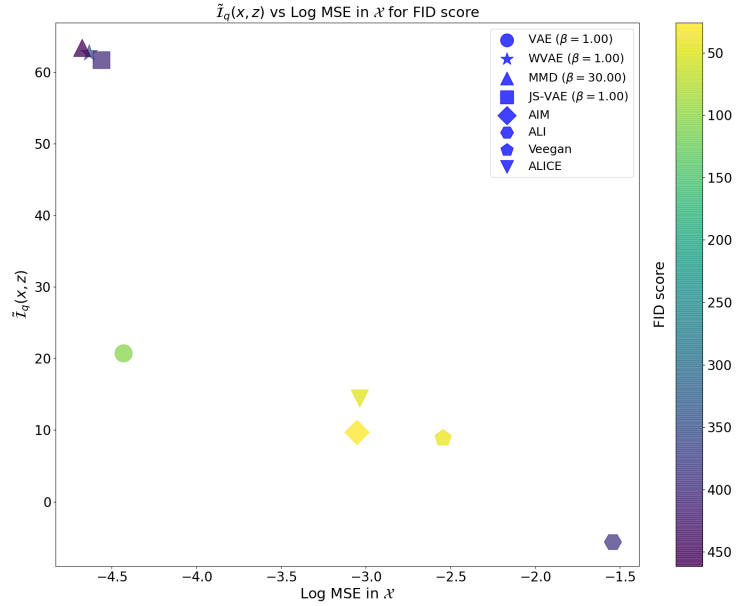


Figure 6.4: $\tilde{I}_q(x, z)$ vs Log MSE in the observed space. The color corresponds to the FID score in MNIST with $J = 10$. The scatter shows the mean of the three runs for each model. For VAE, WVAE, JS-VAE, MMD we chose the β that obtained the best result according to FID.

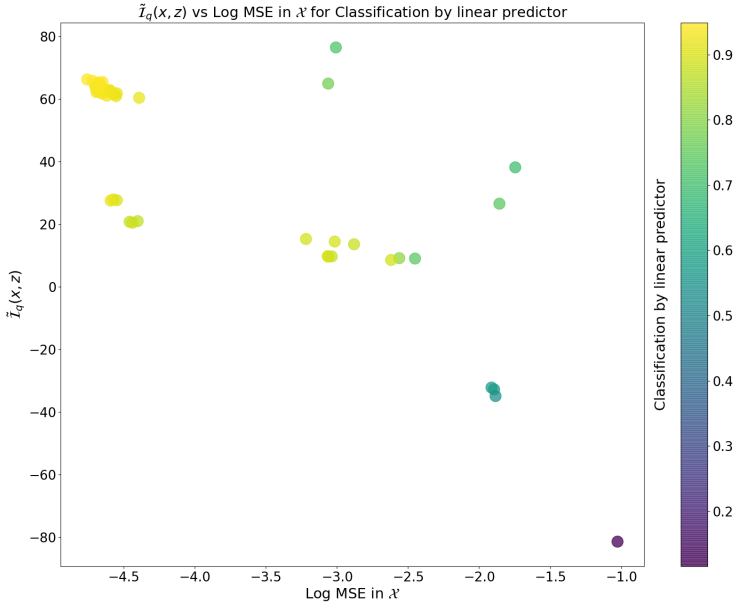


Figure 6.5: Tendency of all generative-inference models considered using $\tilde{I}_q(x, z)$ vs Log MSE in the observed space for classification accuracy by a linear predictor on the latent space in MNIST with $J = 10$.

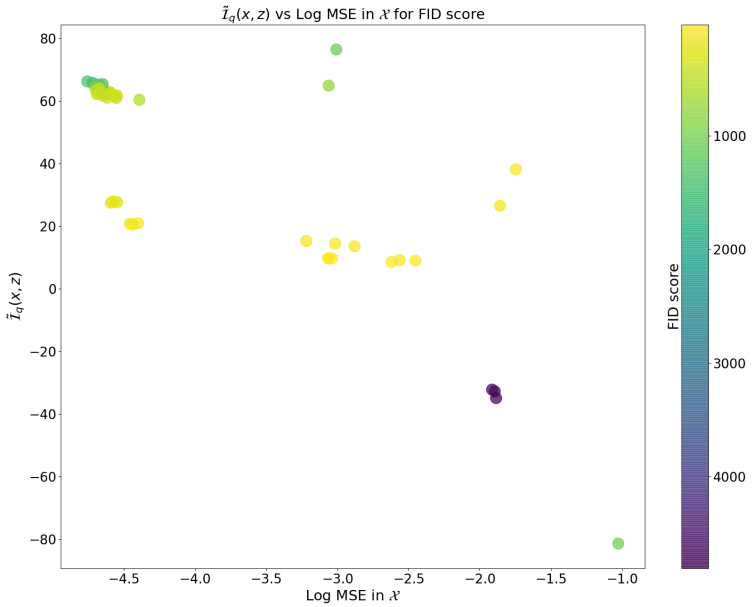


Figure 6.6: Tendency of all generative-inference models considered using $\tilde{I}_q(x, z)$ vs Log MSE in the observed space for FID in MNIST with $J = 10$.

6.4.2 Changing the number of latent dimensions

We explore varying the latent dimensions from $J = 10$ to $J = 20$ in Table 6.5. We also show the same general tendencies of all runs in figures 6.7 and 6.8 for classification and generation, respectively. We observe that in general the accuracy for $J = 20$ is higher than for $J = 10$, in contrast we observe worse generative performance. We hypothesize that when J is higher there is more space in the prior $p(z)$ that the decoder should decode well, however since the space is bigger the task of classification is easier. This is not always the case, for example more complex datasets will need more space (J dimensions) to compress the data in the prior, see Appendix B for FMNIST results.

Model	**LL in $\mathcal{X}$**	LL in $\mathcal{Z}$	$D_{KL}^{\mathcal{Z}}$	D_{KL}^{VAE}	$\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$	$h_q(z\|x)$	IS	FID	Acc%
VAE ($\beta = 0.33$)	-5.23	-2.29	2.02	36.23	35.08	-8.76	8.82	259.46	91.67
	± 0.03	± 0.03	± 0.12	± 0.16	± 0.23	± 0.17	± 0.02	± 11.28	± 0.56
VAE ($\beta = 1.00$)	-4.73	-1.18	22.52	23.51	1.47	4.39	9.35	107.46	88.49
	± 0.02	± 0.02	± 2.00	± 0.13	± 2.10	± 0.18	± 0.02	± 12.03	± 0.19
VAE ($\beta = 30.00$)	-1.90	-0.16	86.56	1.18	-85.35	27.17	3.11	4694.91	60.45
	± 0.00	± 0.00	± 1.05	± 0.01	± 1.06	± 0.02	± 0.14	± 131.56	± 0.41
WVAE ($\beta = 0.33$)	-5.52	-2.64	1.39	118.30	118.27	-91.35	7.70	736.55	95.06
	± 0.02	± 0.01	± 0.01	± 0.19	± 0.19	± 0.19	± 0.03	± 8.62	± 0.37
WVAE ($\beta = 1.00$)	-5.54	-2.57	1.25	118.73	118.74	-91.70	7.65	767.54	94.88
	± 0.03	± 0.05	± 0.05	± 0.54	± 0.53	± 0.56	± 0.02	± 12.19	± 0.13
WVAE ($\beta = 30.00$)	-5.36	-2.09	0.70	130.26	130.28	-102.76	7.31	961.73	92.89
	± 0.02	± 0.02	± 0.20	± 0.36	± 0.38	± 0.23	± 0.07	± 37.96	± 0.45
MMD ($\beta = 0.33$)	-5.50	-1.01	2.01	142.58	124.57	-86.64	6.07	1492.64	95.81
	± 0.07	± 0.18	± 0.03	± 1.27	± 0.16	± 0.59	± 0.32	± 70.36	± 0.34
MMD ($\beta = 1.00$)	-5.54	-1.42	1.75	132.05	122.43	-87.54	6.64	1185.70	95.88
	± 0.05	± 0.09	± 0.03	± 0.57	± 0.12	± 0.46	± 0.11	± 116.86	± 0.19
MMD ($\beta = 30.00$)	-5.51	-2.93	2.22	120.86	118.76	-92.60	6.94	1017.10	95.40
	± 0.06	± 0.03	± 0.02	± 0.31	± 0.34	± 0.35	± 0.02	± 21.88	± 0.24
JS-VAE ($\beta = 0.33$)	-4.80	-1.98	1.02	131.52	130.29	-102.62	7.14	933.93	91.45
	± 0.77	± 0.73	± 1.25	± 22.33	± 20.50	± 21.97	± 0.39	± 153.30	± 2.85
JS-VAE ($\beta = 1.00$)	-5.33	-2.41	0.16	116.82	116.91	-88.08	7.39	865.89	92.84
	± 0.04	± 0.04	± 0.06	± 0.77	± 0.78	± 0.90	± 0.03	± 15.22	± 0.39
JS-VAE ($\beta = 30.00$)	-5.03	-1.88	0.31	111.87	111.84	-82.70	7.06	948.96	90.30
	± 0.09	± 0.07	± 0.13	± 5.04	± 5.01	± 4.84	± 0.03	± 40.82	± 0.18
AIM	-3.18	-0.89	12.02	7.49	-3.92	20.29	9.72	40.89	90.86
	± 0.01	± 0.01	± 0.14	± 0.18	± 0.34	± 0.20	± 0.02	± 8.69	± 0.40
Veegan	-2.13	-0.53	9.82	9.13	-0.25	18.81	9.25	259.20	82.45
	± 0.57	± 0.14	± 0.30	± 2.00	± 1.55	± 1.85	± 0.72	± 313.76	± 8.20
ALI	-1.76	0.17	4.62	43.25	38.68	-14.92	9.75	38.33	86.90
	± 0.12	± 0.03	± 1.41	± 5.45	± 7.05	± 5.71	± 0.02	± 2.56	± 1.93
ALICE	-3.21	-0.60	8.61	15.66	7.26	12.51	9.71	43.43	88.94
	± 0.07	± 0.03	± 0.68	± 0.79	± 1.03	± 0.73	± 0.06	± 14.84	± 2.02

Table 6.5: Relevant metrics for the generative-inference models considered in MNIST dataset with $J = 20$. LL in $\mathcal{X}$ and LL in $\mathcal{Z}$ refer to the MSE in the observed and latent space respectively. $D_{KL}^{\mathcal{Z}} = D_{KL}(q(z)||p(z))$ is computed in closed form assuming $q(z)$ as multivariate Gaussian. $D_{KL}^{\text{VAE}} = D_{KL}(q(z|x)||p(z))$. In **bold** we represent the main unsupervised learning metrics of our study; $\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$ measures the MI of the inference model q and **LL in $\mathcal{X}$** measures the likelihood in $\mathcal{X}$.

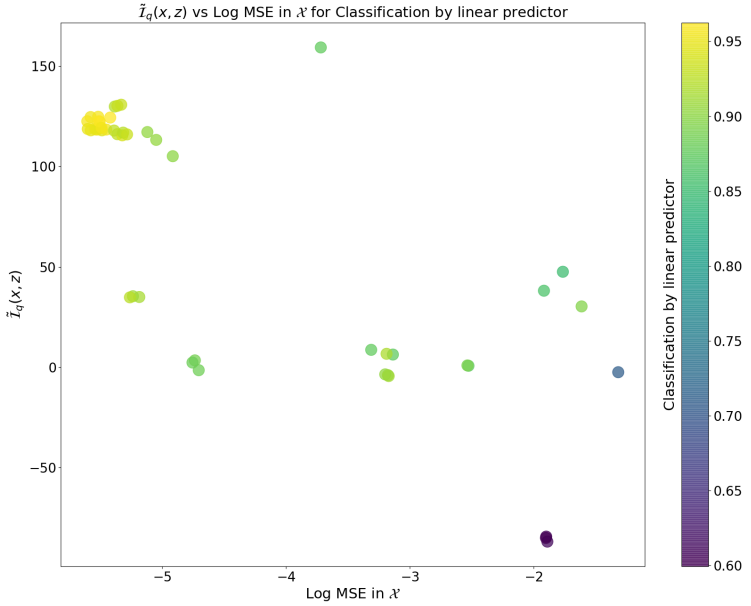


Figure 6.7: Tendency of all generative-inference models considered using $\tilde{I}_q(x, z)$ vs Log MSE in the observed space for classification accuracy by a linear predictor on the latent space in FMNIST with $J = 10$.

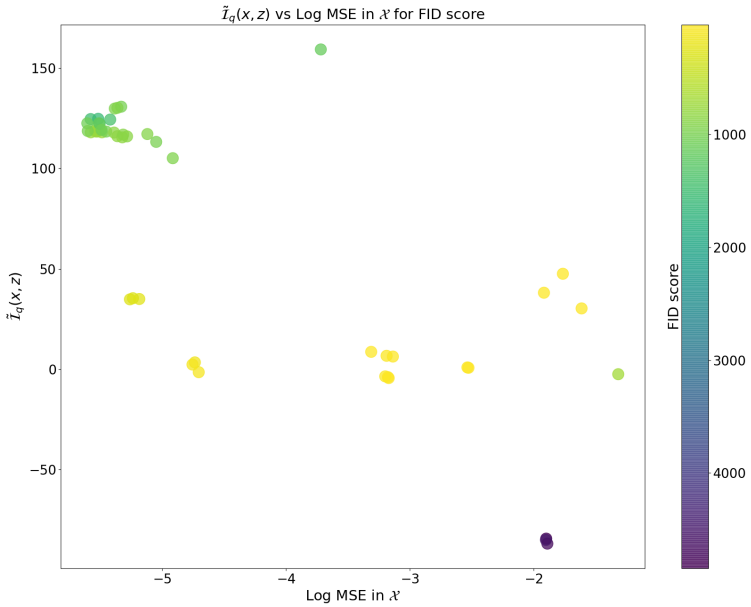


Figure 6.8: Tendency of all generative-inference models considered using $\tilde{I}_q(x, z)$ vs Log MSE in the observed space for classification accuracy by a linear predictor on the latent space in FMNIST with $J = 20$.

6.4.3 Datasets with higher complexity

In simpler datasets the optimization for any generative-inference model is not a limitation. In more complex datasets like CIFAR10 the models tend to have worse generation and/or reconstruction than in simpler datasets. This implies that the suppositions to compute $\tilde{\mathcal{I}}(q(x,z))$ are no longer valid. The likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ computed by the MSE between the input neurons and their reconstruction can also be a bad estimation. As a consequence the generative and representation learning capabilities of generative-inference models are reduced to observing their empirical results.

Table 6.4.3 shows the main results of many generative-inference models in order to give an idea of the models that scale better for more complex datasets. We observe that AIM is the model with the best results overall. In figures 6.9, 6.10 and 6.11 we observe the reconstruction ability of VAE, ALI and AIM models, respectively. To the best of our knowledge no current work in the literature has considered and compared as many generative-inference models under similar optimization conditions.

Model	Acc%	FID	IS
VAE	38.02 ± 2.61	100.05 ± 15.61	3.01 ± 0.21
AIM	76.54 ± 0.43	11.80 ± 0.19	7.08 ± 0.06
ALI	58.94 ± 4.38	20.08 ± 1.89	6.28 ± 0.20
Veegan	70.85 ± 1.43	17.70 ± 1.44	6.57 ± 0.24
ALICE	67.80 ± 1.28	17.75 ± 0.93	6.47 ± 0.12

Table 6.6: Accuracy on test by linear predictor and generative scores for CIFAR10 dataset.

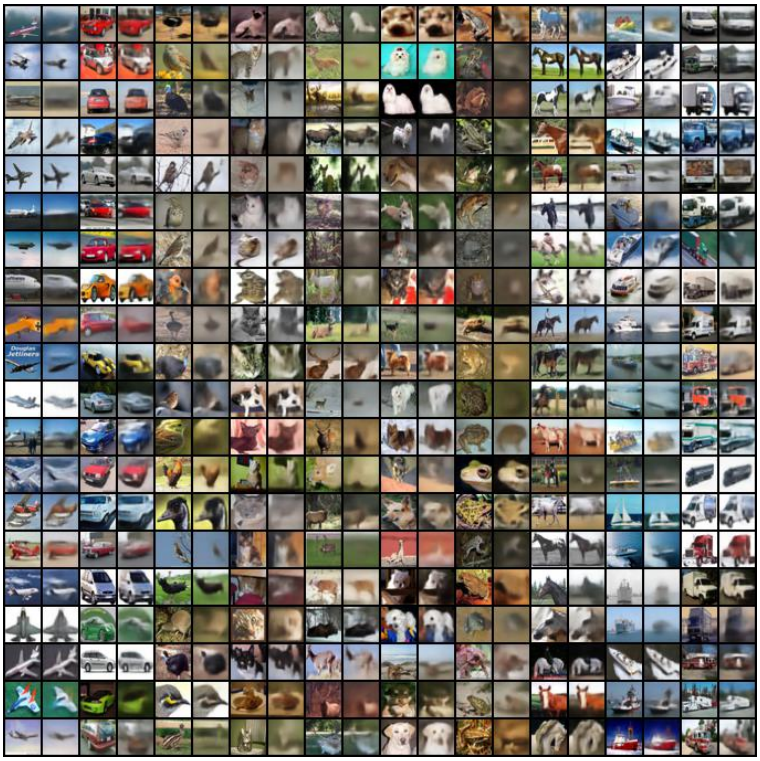


Figure 6.9: Reconstructions for VAE model by classes. Odd columns represent real data and even columns correspond to their reconstructions.

Figure 6.10: Reconstructions for ALI model by classes. Odd columns represent real data and even columns correspond to their reconstructions.

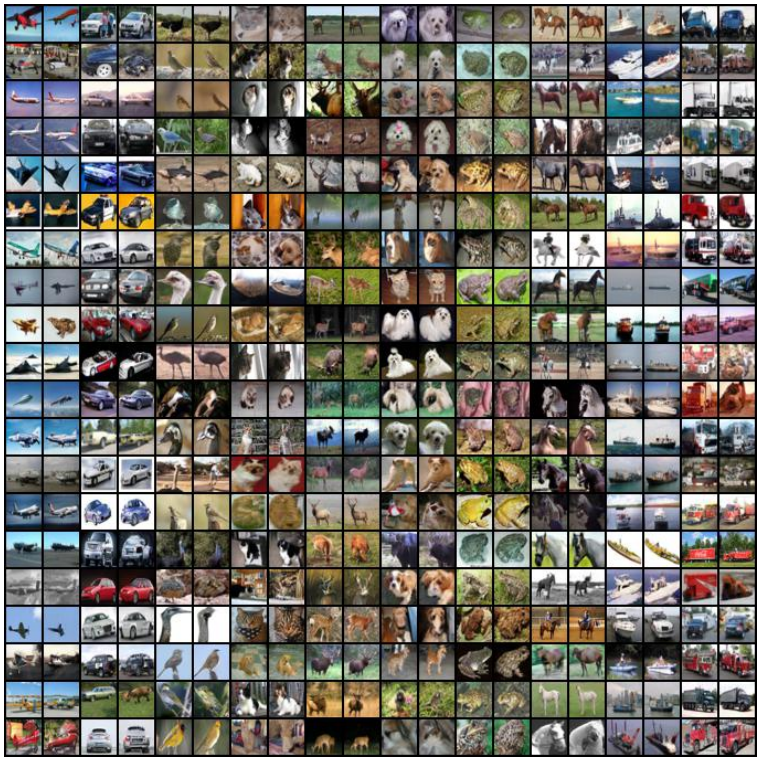


Figure 6.11: Reconstructions for AIM model by classes. Odd columns represent real data and even columns correspond to their reconstructions.

6.4.4 Discussion

In this chapter we computed the empirical results of many generative-inference models taking in consideration the theoretical analysis developed in Chapter 5. We note that for low complexity datasets and small latent dimensions J we can use $\mathcal{I}_q(x, z)$ and the likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ to obtain a close approximation of how a generative-inference model behaves in comparison to others in an unsupervised way. When using more complex datasets it is not possible to use these statistics since each one of them can be wrongly estimated. As a consequence, to know what models behave better we have to rely on empirical results. When simpler datasets are used and VAE based models are optimized correctly ($q(z|x)$ and $p(x|z)$ don't suffer limitations to estimate distributions) there exists a strong trade-off between generation and representation learning for any generative-inference model considered. When more complex datasets are used the representation learning and generation are dominated by the models that scale better. When using the same BigGAN architecture for all models we found that AIM is more consistent and have better inference and generative capabilities than the other models. In simpler datasets it is possible to use either WVAE or VAE for representation learning depending on the generative needs of the particular application.

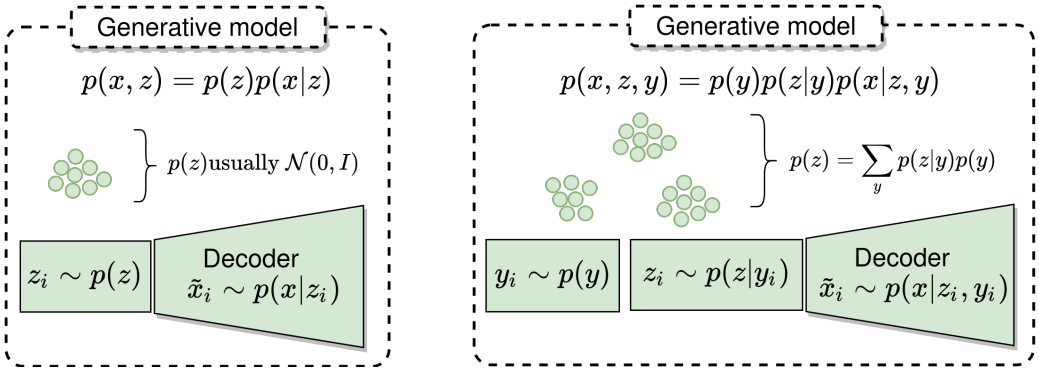
Chapter 7

Generative-Inference models: Extending the graphical model to three variables

In Chapter 6 we showed that generative-inference models that consider the variables $x \in \mathcal{X}$ and $z \in \mathcal{Z}$ have a trade-off between its generation and inference capabilities and that this trade-off could be alleviated by choosing more flexible priors. In this chapter, we review the existing models that add a categorical variable $y \in \mathcal{Y}$ to their graphical model. This additional variable is used to condition the prior $p(z)$ to various conditional priors $p(z|y)$ that are dependent of the categorical variable y. This modification allows having a flexible multimodal prior formed by a mixture of distributions $p(z) = \sum_y p(y)p(z|y)$, which can also be used for clustering applications. Figures 7.1a and 7.1b show how the structure of the generative and inference model change when this new categorical variable y is added. The general three-variable graphical model is given by

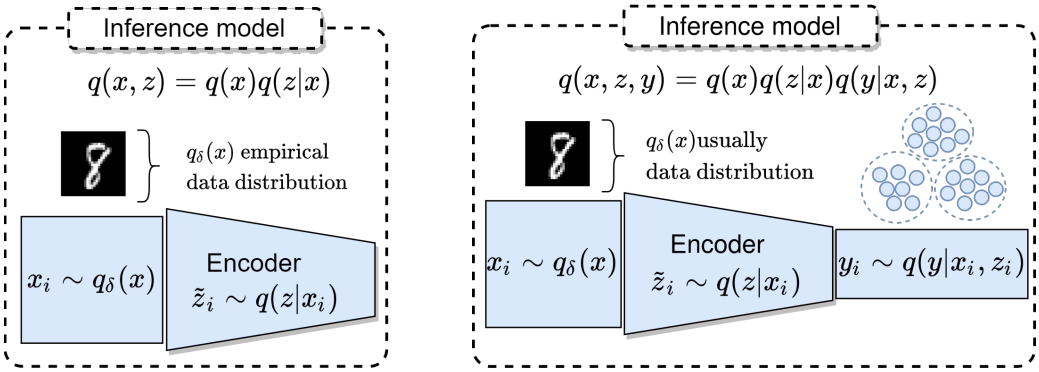
- $p(x, z, y) = p(y)p(z|y)p(x|z, y)$,
- $q(x, z, y) = q(y|x, z)q(z|x)q(x)$.

We can observe the graphical models diagrams of the generative and inference model in Fig. 7.3a and 7.3b, respectively. In the following sections we review the models that consider variables $x \in \mathcal{X}$, $z \in \mathcal{Z}$ and $y \in \mathcal{Y}$, and follow a similar decomposition to those of the previous chapters in terms of the matching joint distribution perspective and the mutual information perspective. In Section 7.1 we prove that Variational Deep Embedding [46] can be obtained by decomposing $D_{KL}(q(x, z, y)||p(x, z, y))$. Noting that the decomposition of $D_{KL}(p(x, z, y)||q(x, z, y))$ has not been explored in the current literature, we propose a method called Matching priors and conditional for clustering [2] which is thoroughly explained in Section 7.2.



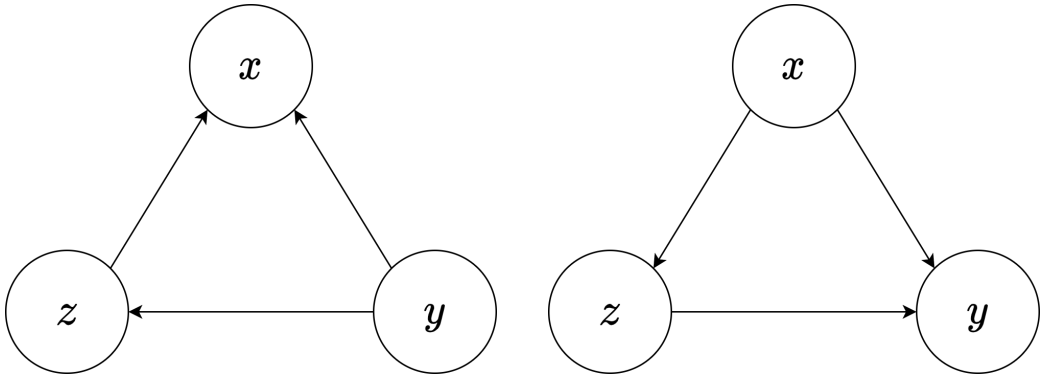
(a) Two variables generative model (b) Three variables generative model

Figure 7.1: Generative models of (a) two variables and (b) three variables.



(a) Two variables inference model (b) Three variables inference model

Figure 7.2: Inference models of (a) two variables and (b) three variables.



(a) Generative model (b) Inference model

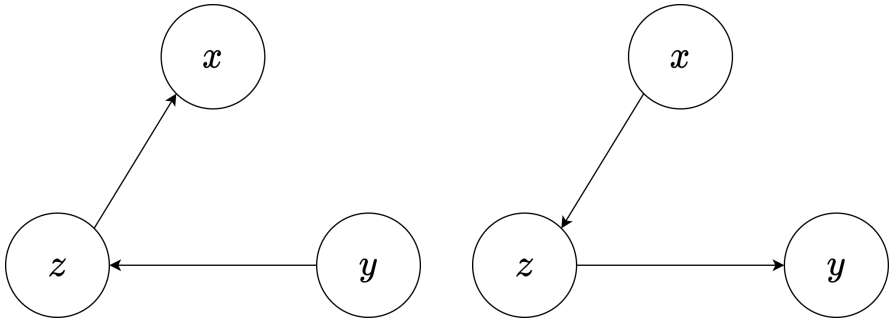
Figure 7.3: Three variable graphical model of (a) generative model and (b) inference model.

7.1 Variational Deep Embedding (VaDE)

Variational Deep Embedding [46] (VaDE) is a three variable generative-inference model for clustering that considers the following graphical model for *generation* and *inference*

- $p(x, z, y) = p(y)p(z|y)p(x|y)$,
- $q(x, z, y) = q(y|z)q(z|x)q(x)$,

respectively. The assumptions made in the graphical model are i) $q(y|z) = q(y|z, x)$, *i.e.* z contains all the necessary information from x to estimate y and ii) $p(x|z, y) = p(x|z)$ *i.e.* z contains all the necessary information from y to estimate x. Figures 7.4a and 7.4b show a graphical model diagram of the generative and inference model, respectively. In section 7.1.1 we present the loss function of VaDE with a small modification in the notation with respect to the original demonstration [46]. In section 7.1.2 we demonstrate that in fact VaDE matches the joint distributions between the inference and the generative model. Finally in section 7.1.3, we show the corresponding decomposition of the VaDE loss function associating it with the mutual information of the inference model.



(a) Generative model (b) Inference model

Figure 7.4: VaDE graphical model of (a) generative model and (b) inference model.

7.1.1 Deriving the loss function of VaDE using Jensen's inequality

In this section we show how the original loss function of VaDE was obtained in [46]. The demonstration starts from maximizing the marginal likelihood between the real data distri-

bution $q(x)$ and the model data distribution $p(x)$ as is shown in Eq. (7.1).

$$\begin{aligned}
\mathbb{E}_{q(x)}\left[\log p(x)\right] &= \mathbb{E}_{q(x)}\left[\log \int_z \sum_y p(x,z,y)\Delta y dz\right] \\
&= \mathbb{E}_{q(x)}\left[\log \int_z \sum_y q(y,z|x)\frac{p(x,z,y)}{q(y,z|x)})\Delta y dz\right] \\
&= \mathbb{E}_{q(x)}\left[\log \mathbb{E}_{q(z|x)q(y|x,z)}\left[\frac{p(x,z,y)}{q(y,z|x)}\right]\right] \\
&\geq \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)q(y|x,z)}\left[\log \frac{p(x,z,y)}{q(y,z|x)}\right] \\
&= \mathbb{E}_{q(x)}\mathbb{E}_{q(z|x)q(y|x,z)}\left[\log \frac{p(x|z,y)p(z|y)p(y)}{q(y|z,x)q(z|x)}\right] \\
&= -\mathcal{L}^{\text{VaDE}}(x)
\end{aligned} \tag{7.1}$$

To optimize VaDE in [46] it was assumed that $p(x|z,y) = p(x|z)$ and $q(y|x,z) = q(y|z)$. It addition a Gaussian distribution for $q(z|x)$ and $p(z|y)$ to obtain closed-form solutions was used.

7.1.2 Loss function of VaDE from a matching joint distributions perspective

Here we show that VaDE is in fact matching the joint distributions of the encoder and decoder by matching posteriors and marginals in the space of the observed variable x (data). We start by expanding the divergence between the joint distributions of the encoder and decoder as

$$\begin{aligned}
D_{\text{KL}}\left(q(x,z,y)||p(x,z,y)\right) = &\int_x\int_y\int_z q(x,z,y)\log\frac{q(x,z,y)}{p(x,z,y)}\,dx\,dy\,dz \\
= &\int_x\int_z q(x,z)\int_z q(y|x,z)\log\frac{q(y|x,z)}{p(y|x,z)}\,dx\,dy\,dz \\
+ &\int_x q(x)\int_z q(z|x)\log\frac{q(z|x)}{p(z|x)}\,dz\,dx + \int_x q(x)\log\frac{q(x)}{p(x)}\,dx \\
= &\,\mathbb{E}_{z,x\sim q(z,x)}\left[D_{\text{KL}}\left(q(y|z,x)||p(y|z,x)\right)\right] \\
&+ \mathbb{E}_{x\sim q(x)}\left[D_{\text{KL}}\left(q(z|x)||p(z|x)\right)\right] + D_{\text{KL}}\left(q(x)||p(x)\right).
\end{aligned} \tag{7.2}$$

The first divergence on the right hand side of Eq. (7.2) is

$$D_{\text{KL}}\left(q(y|z,x)||p(y|z,x)\right) = \mathbb{E}_{y\sim q(y|z,x)}\left[\log\frac{q(y|z)}{p(z|y)p(y)}\right] + \log p(z), \tag{7.3}$$

where we used the replacements $q(y|z,x) = q(y|z)$ and $p(y|z,x) = \frac{p(x|z)p(z|y)p(y)}{p(x|z)p(z)}$, which come from the graphical model assumptions considered in [46].

The second divergence on the right hand side of Eq. (7.2) is

$$D_{\mathrm{KL}}\left(q(z|x)||p(z|x)\right) = \mathbb{E}_{z\sim q(z|x)}\left[\log\frac{q(z|x)}{p(x|z)} - \log p(z)\right] + \log p(x), \tag{7.4}$$

and the third divergence on the right hand side of Eq. (7.2) is

$$D_{\mathrm{KL}}\left(q(x)||p(x)\right) = \mathbb{E}_{x\sim q(x)}\left[\log q(x) - \log p(x)\right]. \tag{7.5}$$

If we add the expectations over $q(z,x) = q(z|x)q(x)$ of Eq. (7.3), with $q(x)$ of Eq. (7.4) and Eq. (7.5) we obtain:

$$\begin{aligned}
&\mathbb{E}_{z,x\sim q(z,x)}\left[D_{\mathrm{KL}}\left(q(y|z,x)||p(y|z,x)\right)\right] + \mathbb{E}_{x\sim q(x)}\left[D_{\mathrm{KL}}\left(q(z|x)||p(z|x)\right)\right] + D_{\mathrm{KL}}\left(q(x)||p(x)\right)\\
&= \mathbb{E}_{q(x)}[\log q(x) - \mathbb{E}_{z,y\sim q(z,y|x)}[\log p(x|z) - \log q(z|x) - \log q(y|z) + \log p(z|y) + \log p(y)]]\\
&= \mathbb{E}_{q(x)}\left[\log q(x) + \mathcal{L}_q^{\mathrm{VaDE}}(x)\right],
\end{aligned}$$

where $\mathcal{L}^{\mathrm{VaDE}}(x)$ corresponds to Eq. 9 in [46]. This means that by maximizing VaDE's loss function one is matching the conditionals and marginals between encoder and decoder in data space. Note that the entropy of the data distribution $\mathbb{E}_{q(x)}\left[\log q(x)\right]$ is constant during optimization.

7.1.3 Loss function of VaDE from a mutual information persepctive

The loss function of VaDE can also be obtained as a variational bound of a sum of mutual information of the inference model given by

$$\mathcal{I}_q(x,z) + \mathbb{E}_{q(x)}[\mathcal{I}_q(z,y|x)] = -\mathcal{L}_q^{\mathrm{VaDE}} + \Delta\mathcal{I}_q^{\mathrm{VaDE}}, \tag{7.6}$$

where $\Delta\mathcal{I}_q^{\mathrm{VaDE}}$ is the gap between the loss function $\mathcal{L}_q^{\mathrm{VaDE}}$ and the sum of the mutual informations presented in Eq. (7.6). To demonstrate this equivalence we begin by expanding the first term on the left as

$$\begin{aligned}
\mathcal{I}_q(x,z) = &\mathbb{E}_{q(x,z)}[\log p(x|z)] - \mathbb{E}_{q(x)}[\log q(x)]\\
&+ \mathbb{E}_{q(x,z)}[D_{KL}(q(x|z)||p(x|z)).
\end{aligned} \tag{7.7}$$

Symmetrically, the second term on the left hand side can be expanded as

$$\begin{aligned}
\mathbb{E}_{q(x)}[\mathcal{I}_q(z,y|x)] = &\mathbb{E}_{q(x)}[\mathbb{E}_{q(z,y|x)}[\log p(z|y)] - \mathbb{E}_{q(z|x)}[\log q(z|x)]\\
&+ \mathbb{E}_{q(z,y|x)}[D_{KL}(q(z|y)||p(z|y))].
\end{aligned} \tag{7.8}$$

Adding equations (7.7) and (7.8), and subtracting the positive restriction $\boldsymbol{\mathcal{R}_q^{\mathrm{VaDE}}(x,z,y)} = \mathbb{E}_{q(x,z,y)}[D_{KL}(q(y|z)||p(y))]$ on the right side, we obtain by the non-negativity of the KL divergence the following:

$$\mathbb{E}_{q(x)}[\mathcal{I}_q(z,y|x)] + \mathcal{I}_q(z,y) \geq -\mathcal{L}_q^{\mathrm{VaDE}},$$

i.e. the loss function of VaDE is a lower bound of two mutual information terms. The mutual information gap in this case collects the following terms

$$\Delta\mathcal{I}_q^{\mathrm{VaDE}} = \mathbb{E}_{q(x,z,y)}[D_{KL}(q(y|z)||p(y))] + \mathbb{E}_{p(x,z,y)}[D_{KL}(p(z|x)||q(z|x)) + D_{KL}(p(y|z)||q(y|z))].$$

7.2 Matching priors and conditional for clustering

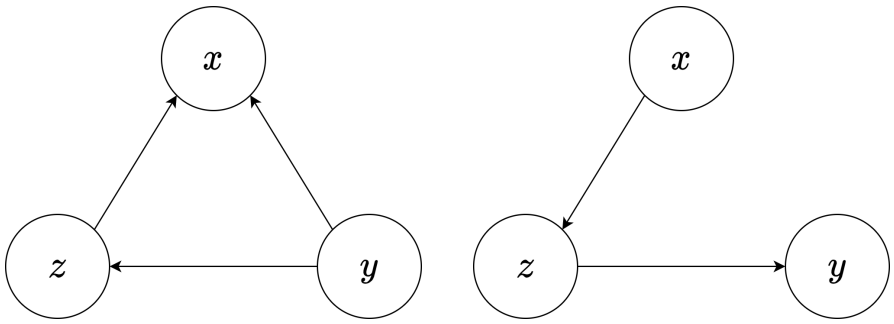
Noting that models obtained by decomposing $D_{KL}(p(x,z,y)||q(x,z,y)$ have not been explored in the literature we propose a new model called Matching Priors and Conditional for Clustering (MPCC). MPCC is a GAN-based model with (a) a learnable mixture of distributions as prior for the generator, (b) an encoder to infer the latent variables from the data and (c) a clustering network to infer the cluster membership from the latent variables. Since this is a new model in the literature the following subsections provide detailed explanations of the optimization process. Sub-section 7.2.1 presents the definition of the model. Sub-section 7.2.2 shows how the loss function of MPCC is obtained by decomposing $D_{KL}(p(x,z,y)||q(x,z,y))$. Finally sub-section 7.2.3 shows how the MPCC loss function can be associated with the mutual information of the generative model. Experiments and results using MPCC are presented in Chapter 8.

7.2.1 Model definition

We specify the graphical models for *generation* and *inference* as follows:

- $p(x,z,y) = p(y)p(z|y)p(x|z,y)$,
- $q(x,z,y) = q(y|z)q(z|x)q(x)$.

The only assumption in the graphical model is that $q(y|z) = q(y|z,x)$, *i.e.* z contains all the necessary information from x to estimate y. Figures 7.5a and 7.5b show graphical model diagram of the generative and inference model, respectively.



(a) Generative model (b) Inference model

Figure 7.5: MPCC graphical model of (a) generative model and (b) inference model.

For generation, we seek to match the decoder $p(x|z,y)$ to the real data distribution $q(x)$. The latent variable is defined by the conditional distributions $p(z|y)$ which can be any distribution under certain restrictions that are presented in Section 7.2.2. The marginal distribution $p(y)$ is defined as a multinomial distribution with weight probabilities ϕ. Note that under this graphical model the latent space becomes multimodal defined by a mixture of distributions $p(z) = \sum_y p(y)p(z|y)$. For inference the latent variables are obtained by the conditional posterior $q(z|x)$ using the empirical data distribution $q(x)$. The distribution $q(y|z)$ is a posterior approximation of the cluster membership of the data.

MPCC is optimized by minimizing the forward Kullback-Leibler divergence of the conditionals and priors between the inference and generative networks as follows:

$$
\begin{aligned}
D_{\mathrm{KL}}\left(p(x,z,y)||q(x,z,y)\right) = & \int_x\int_y\int_z p(x,z,y)\log\frac{p(x,z,y)}{q(x,z,y)}\,dx\,dy\,dz \\
= & \int_y\int_z p(z,y)\int_x p(x|z,y)\log\frac{p(x|z,y)}{q(x|z,y)}\,dx\,dy\,dz \\
+ & \int_y p(y)\int_z p(z|y)\log\frac{p(z|y)}{q(z|y)}\,dz\,dy + \int_y p(y)\log\frac{p(y)}{q(y)}\,dy \\
= & \,\mathbb{E}_{z,y\sim p(z,y)}\left[D_{\mathrm{KL}}\left(p(x|z,y)||q(x|z,y)\right)\right] \\
+ & \,\mathbb{E}_{y\sim p(y)}\left[D_{\mathrm{KL}}\left(p(z|y)||q(z|y)\right)\right] + D_{\mathrm{KL}}\left(p(y)||q(y)\right), \qquad (7.9)
\end{aligned}
$$

In the following sections we derive a tractable expression for Eq. (7.9) and present the MPCC algorithm.

7.2.2 Loss function and optimization of MPCC from a matching joint distributions perspective

Because $q(y)$, $q(z|y)$ and $q(x|z,y)$ are impossible to sample from, we derive a closed-form solution for Eq. (7.9). In particular for any fixed y and z we can decompose $D_{KL}(p(x|z,y)||q(x|z,y))$ as follows:

$$
\begin{aligned}
& D_{KL}(p(x|z,y)||q(x|z,y)) \\
= & \,\mathbb{E}_{p(x|z,y)}\left[\log\frac{p(x|z,y)}{q(x)}\frac{q(z,y)}{q(z,y|x)}\right] \\
= & \,\mathbb{E}_{p(x|z,y)}\Bigg[\log\frac{p(x|z,y)}{q(x)} - \log q(y|z) - \log q(z|x) + \log q(z|y) + \log q(y)\Bigg]. \qquad (7.10)
\end{aligned}
$$

Adding $\log p(z|y) + \log p(y) - \log q(z|y) - \log q(y)$ to both sides of Eq. (7.10) and taking the expectation with respect to $p(z,y)$ Eq. (7.9) is recovered. After adding these terms and taking the expectation we can collect the resulting right hand side of Eq. (7.10) as follows:

$$
\begin{aligned}
& \mathbb{E}_{p(z,y)}[D_{KL}(p(x|z,y)||q(x|z,y)) + D_{KL}(p(z|y)||q(z|y)) + D_{KL}(p(y)||q(y))] \\
= & \underbrace{\mathbb{E}_{p(y)p(z|y)}[D_{KL}(p(x|z,y)||q(x))]}_{\textbf{Loss I}} + \underbrace{\mathbb{E}_{p(y)p(z|y)p(x|z,y)}[-\log q(z|x) - \log q(y|z)]}_{\textbf{Loss II}} \\
+ & \underbrace{\mathbb{E}_{p(z|y)p(y)}[\log p(y) + \log p(z|y)]}_{\textbf{Loss III}}, \qquad (7.11)
\end{aligned}
$$

where **Loss I** seeks to match the true distribution $q(x)$, **Loss II** is related to the variational approximation of the latent variables and **Loss III** is associated with the distribution of the cluster parameters. The right hand term of Eq. (7.11) is a loss function, composed of three terms with distributions that we can sample from. In the next section we explain the strategy to optimize each of the terms of the proposed loss function.

MPCC follows the idea that the data space $\mathcal{X}$ is compressed in the latent space $\mathcal{Z}$ and a separation in this space will likely partition the data in the most representatives clusters

$p(z|y)$. The separability of these conditional distributions will be enforced by $q(y|z)$ which also backpropagates through the parameters of $p(z|y)$. The connection with the data space is through the decoder $p(x|z,y)$ for generation and the encoder $q(z|x)$ for inference.

In what follows we describe the assumptions made in the distributions of the graphical model and how to optimize Eq. (7.11). For simplicity we assume the conditional $p(z|y)$ to be a Gaussian distribution, but other distributions could be used with the only restriction being that their entropy should have a closed-form or at least a bound (second term in **Loss III**). In our experiments the latent variable $z|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ is sampled using the reparameterization trick [53], *i.e.* $z = \mu_y + \sigma_y \odot \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, I)$ and $\odot$ is the Hadamard product. The parameters μ_y, σ_y^2 are learnable and they are conditioned on y. Under Gaussian conditional distribution the latent space becomes a GMM, as we can observe mathematically $p(z) = \sum_y p(y)p(z|y) = \sum_y p(y)\mathcal{N}(\mu_y, \sigma_y^2)$.

The distribution $p(x|z,y)$ is modeled by a neural network and trained via adversarial learning, *i.e.* it does not require parametric assumptions. The inferential distribution $q(z|x)$ is also modeled by a neural network and its distribution is assumed Gaussian for simplicity. The categorical distribution $q(y|z)$ may also be modeled by a neural network but we propose a simpler approach based on the membership from the latent variable z to the Gaussian components. A diagram of the proposed model considering these assumptions is shown in Fig. 7.6. We now expand on this for each of the losses in Eq. (7.11).

Loss I: Instead of minimizing the Kullback-Leibler divergence shown in the first term on the right hand of Eq. (7.11) we choose to match the conditional decoder $p(x|z,y)$ with the empirical data distribution $q(x)$ using a generative adversarial approach. The GAN loss function can be formulated as [21]

$$\begin{aligned} &\max_D \ \mathbb{E}_{x\sim q(x)}[f(D(x))] + \mathbb{E}_{\tilde{x}\sim p(x,z,y)}[g(D(\tilde{x}))], \\ &\min_G \ \mathbb{E}_{\tilde{x}\sim p(x,z,y)}[h(D(\tilde{x}))], \end{aligned} \tag{7.12}$$

where D and G are the discriminator and generator networks, respectively, and tilde is used to denote sampled variables. For all our experiments we use the hinge loss function [70], [108], *i.e.* $f = -\min(0, o-1)$, $g = \min(0, -o-1)$ and $h = -o$, being o the output of the discriminator. The parameters and distribution associated with **Loss I** are colored in blue in Fig.7.6.

Loss II: The first term of this loss is estimated through Monte Carlo sampling as

$$\begin{aligned} &\mathbb{E}_{p(y)p(z|y)p(x|z,y)}[-\log q(z|x)] \\ &= \mathbb{E}_{y_i\sim p(y), z_i\sim p(z|y=y_i), \tilde{x}_i\sim p(x|z=z_i,y=y_i)} \underbrace{\left[\sum_{j=1}^{J} \frac{1}{2}\log(2\pi\tilde{\sigma}_{ij}^2) + \frac{(z_{ij}-\tilde{\mu}_{ij})^2}{2\tilde{\sigma}_{ij}^2}\right]}_{L_q(\tilde{\mu}_i, \tilde{\sigma}_i^2, z_i)}, \end{aligned} \tag{7.13}$$

where J is the dimensionality of the latent variable z. By minimizing Eq. (7.13) we are maximizing the log-likelihood of the encoder $q(z|x)$ with respect to the Gaussian prior $p(z|y)$. This reconstruction error is estimated by matching the samples $z_i \sim p(z|y=y_i)$ with the Gaussian distribution $(\tilde{\mu}_i, \tilde{\sigma}_i^2) \sim q(z|x=\tilde{x}_i)$, where $\tilde{x}_i$ is the decoded representation of z_i.

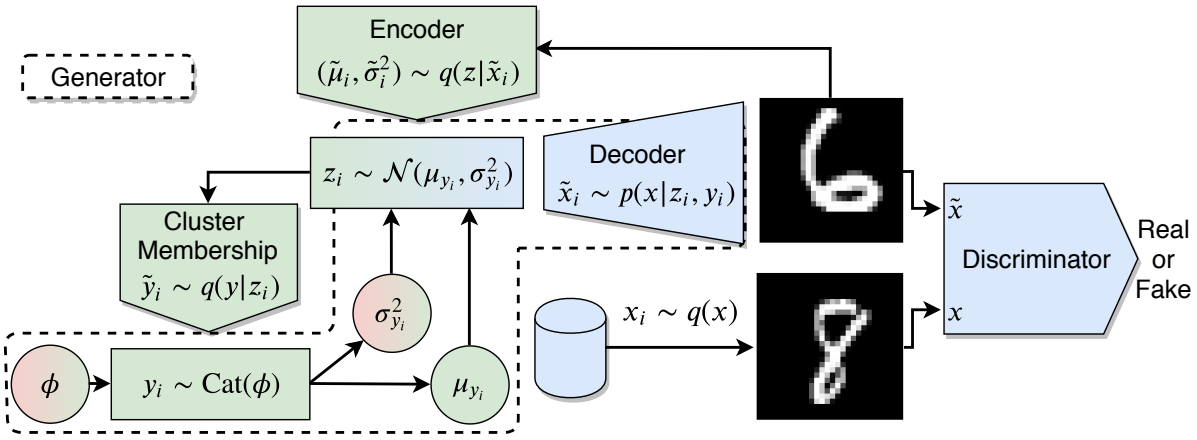


Figure 7.6: Diagram of the MPCC model. The blue colored elements are associated with **Loss I** (Eq. (7.12)). The green colored elements are associated with **Loss II** (Equations 7.13 and 7.14). The red colored elements are associated with **Loss III** (Eq. (7.16)). The dashed line corresponds to the generator (GMM plus decoder).

The second term of **Loss II** is equivalent to the cross-entropy between the sampled label $y_i \sim p(y)$ and the estimated cluster membership $\tilde{y}_i$

$$L_c(y_i, \tilde{y}_i) = -\sum_{k=1}^{K} y_{ik} \log \tilde{y}_{ik}, \tag{7.14}$$

where K is the number of clusters and

$$\tilde{y}_{im} = q(y = m|z = z_i) = \frac{\mathcal{N}(z_i|\mu_m, \sigma_m^2)}{\sum_{k=1}^{K} \mathcal{N}(z_i|\mu_k, \sigma_k^2)}, \tag{7.15}$$

is the membership of z_i to the m-th cluster. The parameters μ_m and σ_m^2 are learnable, and $m \in [1, \ldots, K]$ is the index corresponding to each cluster. The parameters and distribution associated with **Loss II** are colored in green in Fig. 7.6. In practice Eq. (7.15) is estimated using the log-sum-exp trick.

Loss III: This loss is associated with the regularization of the Gaussian mixture model parameters ϕ, μ and σ^2 and has a closed form

$$\begin{aligned} &\mathbb{E}_{p(y)p(z|y)}[\log p(y) + \log p(z|y)] \\ &= \underbrace{\sum_{k=1}^{K} \phi_k \left[\log \phi_k - \sum_{j=1}^{J} \left(\frac{1}{2} + \frac{1}{2} \log(2\pi\sigma_{kj}^2) \right) \right]}_{L_p(\phi, \sigma^2)}, \end{aligned} \tag{7.16}$$

where the first term corresponds to the entropy maximization of the mixture weights, *i.e.* in general every Gaussian will not collapse to less than K modes of the data distribution which is a solution with lower entropy. In our experiments we fix $\phi_k = 1/K$, *i.e.* ϕ is not

learnable. The second term is a regularization for the variance (entropy) of each Gaussian which avoids the collapse of $p(z|y)$. The parameters associated with **Loss III** are shown in red in Fig. 7.6.

Loss I scale differs from that of the terms associated with the latent variables. To balance all terms we multiply Eq. (7.13) by one over the dimensionality of x[1] and the second term of Eq. (7.16) by one over the dimensionality of the latent variables. During training **Loss III** is weighted by a constant factor λ_p. We explain how this constant is set in Section 8.1.3. The full procedure to train the MPCC model is summarized in Algorithm 1. Note that MPCC is scalable in the number of clusters since Eq. (7.13) is a Monte Carlo approximation in y and the cost of Eq. (7.16) is low since J is small in comparison to the data dimensionality.

Algorithm 1 MPCC algorithm

K, $J \leftarrow$ Set number of clusters and latent dimensionality
η, $\eta_p \leftarrow$ Set learning rates
$\theta_g, \theta_d, \theta_e \leftarrow$ Initialize network parameters
$\phi, \mu, \sigma^2 \leftarrow$ Initialize GMM parameters
$\theta_c \leftarrow [\phi, \mu, \sigma^2]$
repeat
  for D_{steps} **do**
    $x_1, \ldots, x_n \sim q(x)$ ▷ Draw n samples from empirical distribution
    $y_1, \ldots, y_n \sim p(y)$ ▷ Draw n samples from categorical prior
    $z_i \sim p(z|y = y_i), \quad i = 1, \ldots, n$ ▷ Draw n samples from Gaussian conditional prior
    $\tilde{x}_i \sim p(x|z = z_i, y = y_i), \quad i = 1, \ldots, n$ ▷ Generate samples using generator network
    $\theta_d \leftarrow \theta_d + \eta \nabla_{\theta_d} \left[\frac{1}{n} \sum_{j=1}^{n} f(D(x_j)) + \frac{1}{n} \sum_{i=1}^{n} g(D(\tilde{x}_i)) \right]$ ▷ Gradient update on discriminator network
  end for
  $y_1, \ldots, y_n \sim p(y)$ ▷ Draw n samples from categorical prior
  $z_i \sim p(z|y = y_i), \quad i = 1, \ldots, n$ ▷ Draw n samples from Gaussian conditional prior
  $\tilde{x}_i \sim p(x|z = z_i, y = y_i), \quad i = 1, \ldots, n$ ▷ Generate samples using generator network
  $(\theta_g, \theta_c) \leftarrow (\theta_g, \theta_c) - \eta \nabla_{(\theta_g, \theta_c)} \frac{1}{n} \sum_{i=1}^{n} h(D(\tilde{x}_i))$ ▷ Gradient update on generator network
  for E_{steps} **do**
    $y_1, \ldots, y_n \sim p(y)$ ▷ Draw n samples from categorical prior
    $z_i \sim p(z|y = y_i), \quad i = 1, \ldots, n$ ▷ Draw n samples from Gaussian conditional prior
    $\tilde{x}_i \sim p(x|z = z_i, y = y_i), \quad i = 1, \ldots, n$ ▷ Generate samples using generator network
    $(\tilde{\mu}_i, \tilde{\sigma}_i^2) \sim q(z|x = \tilde{x}_i), \quad i = 1, \ldots, n$ ▷ Encode $\tilde{x}$ to obtain mean and variance
    $\theta_e \leftarrow \theta_e - \eta \nabla_{\theta_e} \frac{1}{n} \sum_{i=1}^{n} L_q(\tilde{\mu}_i, \tilde{\sigma}_i^2, z_i)$ ▷ Gradient update on encoder network
    if first E_{step} **then**
      $\tilde{y}_i \sim q(y|z = z_i), \quad i = 1, \ldots, n$
      $\theta_c \leftarrow \theta_c - \eta_p \nabla_{\theta_c} \left[\frac{1}{n} \sum_{i=1}^{n} L_c(y_i, \tilde{y}_i) + \lambda_p \cdot L_p(\phi, \sigma^2)) \right]$ ▷ Gradient update on Prior parameters
    end if
  end for
until convergence

7.2.3 Loss function of MPCC from a mutual information perspective

The loss function of MPCC can be obtained as a variational bound of the sum of mutual information of the generative model given by

$$\mathbb{E}_{p(y)}[\mathcal{I}_p(x, z|y)] + \mathcal{I}_p(z, y) = -\mathcal{L}_p^{\text{MPCC}} + \Delta\mathcal{I}_p^{\text{MPCC}}, \tag{7.17}$$

where $\Delta\mathcal{I}_p^{\text{MPCC}}$ is the gap between the loss function $\mathcal{L}_p^{\text{MPCC}}$ and the sum of the mutual informations presented in Eq. (7.17). To demonstrate this equivalence we begin by expanding the first term on the left as

$$\begin{aligned} \mathbb{E}_{p(y)}[\mathcal{I}_p(x, z|y)] = \mathbb{E}_{p(y)}[\mathbb{E}_{p(x,z|y)}[\log q(z|x)] - \mathbb{E}_{p(z|y)}[\log p(z|y)] \\ + \mathbb{E}_{p(x,z|y)}[D_{KL}(p(z|x)||q(z|x))]. \end{aligned} \tag{7.18}$$

[1] If x is an image then its dimensionality would be $channels \times height \times width$

Symmetrically, the second term on the left hand side can be expanded as

$$\begin{aligned}\mathcal{I}_p(z,y) = \mathbb{E}_{p(z,y)}[\log q(y|z)] - \mathbb{E}_{p(y)}[\log p(y)] \\ + \mathbb{E}_{p(z,y)}[D_{KL}(p(y|z)||q(y|z)).\end{aligned} \tag{7.19}$$

Adding equations 7.18 and 7.19 then subtracting the positive restriction $\boldsymbol{\mathcal{R}_p^{\text{MPCC}}(x,z,y)} = \mathcal{D}(p(x)||q(x))$ on the right side, we obtain by the non-negativity of the KL divergence the following:

$$\mathbb{E}_{p(z)}[\mathcal{I}_p(x,z|y)] + \mathcal{I}_p(z,y) \geq -\mathcal{L}_p^{\text{MPCC}},$$

i.e. the loss function of MPCC is a lower bound of two mutual information terms. The mutual information gap in this case collects the following terms

$$\Delta\mathcal{I}_p^{\text{MPCC}} = \mathcal{D}(p(x)||q(x)) + \mathbb{E}_{p(x,z,y)}[D_{KL}(p(z|x)||q(z|x)) + D_{KL}(p(y|z)||q(y|z))].$$

7.2.4 Related methods

In Section 7.2 we showed that the latent space of MPCC is reduced to a GMM under Gaussian conditional distributions. Because all the experiments are performed based on this assumption, in this section we summarize the literature of generative and autoencoding models that consider GMMs. The combination of generative models and GMMs is not new. Several methods have applied GMM in autoencoding [111], [123] or GAN [32], [95] with applications in novelty detection, anomaly detection, training in scarce data regimes or generative capability evaluations. Other approaches have performed clustering using GMM and GANs but are not directly comparable to MPCC because they use mixtures of various generators and discriminators [119] or fixed priors with ad-hoc set parameters [5].

Among the related works on generative models for clustering the closest approaches to MPCC are ClusterGAN [79] and Variational Deep Embedding (VaDE) [46]. ClusterGAN differs from our model in that it sets the dimensions of the latent space as either continuous or categorical while MPCC uses a continuous latent space which is conditioned on the categorical variable y.

On the other hand, VaDE differs greatly in the training procedure, despite its similar theoretical basis. VaDE, as a variational autoencoder model, matches the joint distributions in the inverse KL sense $D_{KL}(q(x,z,y)||p(x,z,y))$ by matching the posteriors and the marginals in data space as demonstrated in section 7.1.2 . MPCC optimizes the forward KL, *i.e.* matching the priors in latent space and conditionals in data space. Optimizing different KLs yield notably different decompositions and thus training procedures. For the inverse KL [46] it is more difficult to generalize the latent space to any multi-modal distribution as we discuss in what follows.

In MPCC the latent space can be naturally extended to any mixture of distributions, the only requirement being that the entropy of each distribution component $p(z|y)$ should have a closed-form or at least a bound. In general any model decomposed by the forward KL enjoy this property.

Inverse KL decomposition's, such as the case of VAE and VaDE, need a closed-form solution for the divergence between the posterior and the prior. In VaDE this term corresponds to

$$\mathbb{E}_{q(x)}\mathbb{E}_{q(z,y|x)}[\log q(z|x) - \log p(z|y)] = \mathbb{E}_{q(x)}\mathbb{E}_{q(y|x)}[D_{KL}(q(z|x)||p(z|y))], \tag{7.20}$$

which has a closed-form since $q(z|x)$ and $p(z|y)$ are Gaussians. Other distributions can be used, however they need to be from the exponential family and to have the same distribution [84], although some exceptions exists [101], [16]. In addition to the exponential family requirement, a reparameterization trick is needed for the posterior distribution further limiting the distributions that can be used and requiring other forms of reparameterization [25], [97], [44].

Alternatively, adversarial training can be used to match the marginal posterior with more flexible priors. However it has been observed [96] that this kind of optimization [75], [77] underestimates its Kullback-Leibler divergence and it worsens the likelihood of the decoder likely affecting its clustering capabilities.

Chapter 8

Experiments and results for matching priors and conditional for clustering

In this chapter we run experiments and discuss the inference and generative capabilities of MPCC using competitive benchmarks from the literature. Code is available at here.

8.1 Experimental setup

8.1.1 Datasets

In order to evaluate MPCC we performed clustering in five benchmark datasets: a handwritten digit dataset (MNIST, [63]), a handwritten character dataset (Omniglot, [60]), two color image dataset (CIFAR-10 and CIFAR-100 [58]) and a fashion products image dataset (Fashion-MNIST, [115]). For CIFAR-100 we consider the 20 superclasses. Omniglot was created using the procedure described in [41]. Because the task is fully unsupervised we concatenate the training and test sets as frequently done in the area [41], [11], [116]. All datasets have 10 classes except for Omniglot and CIFAR-20 with 100 and 20, respectively. All images were rescaled to 32×32 and reescaled to [-1,1] in order to use similar network architectures. The CIFAR-10 results shown in tables 8.4, 8.5 and 8.6 were trained using only the training set (50,000 examples) for a fair comparison with the literature. For all clustering experiments we use the same number of clusters as classes in the dataset.

8.1.2 Evaluation metrics

Following [116], the performance of MPCC is measured using the clustering accuracy metric in which each cluster is assigned to the most frequent class in the cluster. Formally this is defined as

$$\text{ACC} = \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = m(c_i)\}}{N}, \tag{8.1}$$

where N is the total number of samples, y_i is the ground truth, $c_i = \arg\max_k q(y = k|z = z_i)$ is the predicted cluster and $\mathcal{M}$ is the space of all possible mappings between clusters and labels.

To measure the quality of the samples generated by MPCC we use the inception score (IS) [98] and the Fréchet inception distance (FID) [36].

8.1.3 Empirical details

Our architecture is based on optimization techniques used in the BigGAN [8][1], we found that simpler architectures such as DCGAN [91] were not able to learn complex distributions like CIFAR-10 while optimizing the parameters of the prior. Architecture details are given in section 8.1.4. We consider parameter sharing between the encoder and discriminator, and we test the importance of this in Section 8.2.2. We set $D_{steps} = 4$ (see Algorithm 1). We made small changes in the architecture and optimization parameters depending on the dataset (see section 8.1.4).

We observed the same relation between batch size and (IS, FID) reported in [8]. However we found artifacts that hurt accuracy performance when using batch size larger than 50. For simplicity we used this value in all experiments. We consider a weighting factor λ_p for **Loss III** (Eq. (7.16)). We observed that if $\lambda_p = 1$, the standard deviation of the prior σ would increase monotonically, hindering training. On the other hand if λ_p is too small, σ decreases, collapsing at some point. We found empirically that a value of $\lambda_p = 0.01$ combined with a minimum threshold for σ of 0.5 allow the algorithm to converge to good solutions.

The parameter settings indicated above were fixed for all experiments and didn't show a big effect in accuracy performance. In section 8.2.1 we explore the parameters that most affect training. We trained all experiments for 75,000 iterations, except for MNIST and Omniglot which was 125,000 iterations. For unconditional and conditional training we kept the model of the last iteration.

8.1.4 Architecture details

In MPCC we use the BigGAN model techniques [8] as a base for all our experiments. This architecture employs ResNet [34] and Spectral Normalization [78]. Each Resnet block follows the same configuration described in section 6.3.3 and can be observed in Fig. 8.1. The architectures vary with respect the experiments made in Chapter 6. Fig. 8.2 (a) shows the generator used for the datasets employed. Fig 8.2 (b) shows the unconditional architecture of the discriminator. In the case of the conditional discriminator a term $\text{Embed}(y) \cdot h$ is added, where h is the output of the global sum pooling (see Table 8.2). We can write the architectures for all datasets in a general way as in Fig. 8.2 or more specific as in Tables 8.1, 8.2 and 8.3, where C, J and D change between datasets.

As we observed in the result section (Table 8.5), we found an improvement in terms of sampling quality and reconstruction error when parameters between the discriminator and the encoder are shared. We experimented on the number of residual blocks shared and found that the best performance was obtained when sharing the first three residual blocks.

We use the optimizer configuration and network initialization described in section 6.3.3. We also use a learning rate of $2e-4$ for all networks and experiments but with the exception

[1] `https://github.com/ajbrock/BigGAN-PyTorch`

of the prior parameters that is studied latter. For evaluation we use standing statistics [8] *i.e.* in evaluation mode we run many times (in our case 16) the forward propagation of the generator model $\tilde{x} \sim p(x, z, y)$ storing the means and variances aggregated across all forward passes.

We use three techniques depending on the dataset to deliver the latent information z and y into the decoder distribution $p(x|z, y)$. The first two correspond to a hierarchical latent space architecture [8], which concatenates Embed(y) with a subset of z, followed by a linear transformation to estimate the statistics of the batch norm layers followed by (see Fig. 8.1). The first method is shown in Fig. 8.2, which splits the latent variable z into equal chunks, delivering each one to a different part of the network. In this case we have four chunks (1 entry + 3 residual blocks). The second method is similar to the first one, the only difference being that all z are shared and no split operation is done. The schematic of this generator is equivalent to Fig. 8.2 (a) except that the purple box performs a copy instead of split operation. The third method passes all the latent z as usual [91] and uses conditional batch normalization [24]. This method learns embeddings conditioned on y, which are different for each layer, *i.e.* the linear transformation in the yellow boxes of Fig. 8.1 correspond to an embedding, and the shared embedding should be ignored.

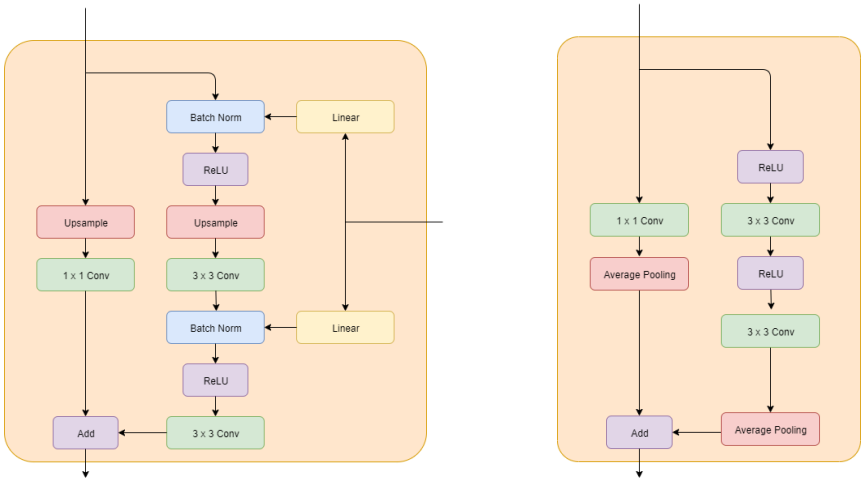
- For CIFAR10 and CIFAR20 we use the first method since this is the default architecture used in BigGAN. For simplicity we kept this configuration for all ablation and clustering experiments with these datasets. We found that mode collapse problems would appear if the third method is used in these datasets. The configuration of the parameters for these datasets is $C = 96$, $D = 3$ (RGB), $J = 128$ and $\eta_p = 6 \cdot 10^{-4}$.

- For datasets with simpler distributions such as MNIST and Omniglot, the third method is more stable and yields the best results. We found that if we use hierarchical latent space architectures poor results were obtained. In particular we observed that the chunks in the first method are decorrelated, which is particularly bad for simpler datasets such as MNIST and Omniglot because the network gains lot of capacity ignoring the embedding y and learning the full real distribution in all the clusters. The configuration of the parameters for these datasets is $D = 1$ (grayscale), $J = 24$ and $\eta_p = 1.6 \cdot 10^{-3}$. For MNIST $C = 12$ and for Omniglot $C = 16$.

- For FMNIST we observed that a poor performance was obtained with both the first and third method. The best results for this dataset were obtained using the second method. The configuration of the parameters for this dataset is $C = 24$, $D = 1$ (gray), $J = 16$ and $\eta_p = 1.6 \cdot 10^{-3}$.

We developed a Pytorch implementation for MPCC based on the implementation of BigGAN[2]. The IS and FID scores are calculated using the official implementations[3]. We run each model in a GeForce RTX 2080 Ti, the amount of time that MPCC iterates depends on the dataset but it is within the range of 12-24 hours.

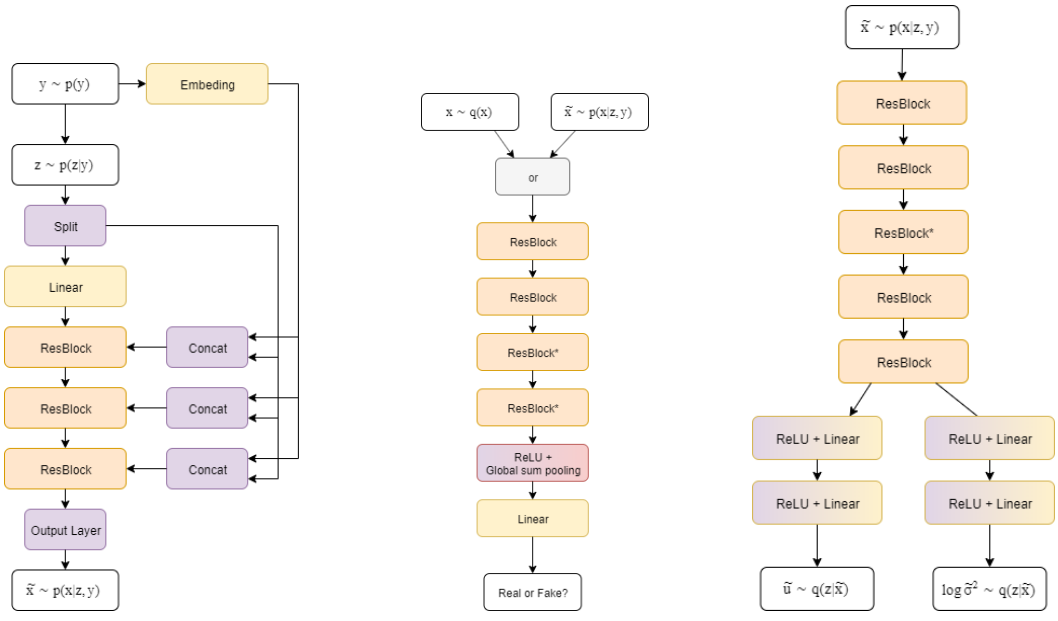
[2] `https://github.com/ajbrock/BigGAN-PyTorch`
[3] `https://github.com/bioinf-jku/TTUR`



(a) Generator block

(b) Discriminator/Encoder block

Figure 8.1: Residual blocks used for MPCC generator, discriminator and encoder networks.



(a) Generator

(b) Discriminator

(c) Encoder

Figure 8.2: Architectures of MPCC generator, discriminator and encoder networks, respectively

$y_i \in \{0, \dots, K-1\} \sim Cat(\phi)$ $z_i \in \mathbb{R}^J \sim \mathcal{N}(\mu_{y_i}, \sigma^2_{y_i})$ Share Embed$(y) \in \mathbb{R}^J$
Linear$(J) \to 4 \times 4 \times 4C$
Resblock up $4C \to 4C$
Resblock up $4C \to 4C$
Resblock up $4C \to 4C$
Output Layer: BN, ReLU, 3×3 Conv $C \to D$ Tanh

Table 8.1: Generator

$x \in \mathbb{R}^{32 \times 32 \times D}$
Resblock down $D \to 4C$
Resblock down $4C \to 4C$
Resblock $4C \to 4C$
Resblock $4C \to 4C$
ReLU, Global sum pooling
(linear $\to 1$) if conditional :+ Embed$(y) \cdot h$

Table 8.2: Discriminator

$x \in \mathbb{R}^{32 \times 32 \times D}$
Resblock down $D \to 4C$
Resblock down $4C \to 4C$
Resblock $4C \to 4C$
Resblock down $4C \to 4C$
Resblock down $4C \to 4C$
Flatten
$\times 2$: Linear $(32//2^4 \times 4C) \to (32//2^4 \times 4C)//2$
$\times 2$: Linear $((32//2^4 \times 4C)//2) \to J$

Table 8.3: Encoder

8.2 Results

8.2.1 Ablation study

We found that E_{steps}, the number of encoder updates per epoch, and η_p, the learning rate of the prior parameters, are the most relevant hyperparameters to obtain high accuracy and generation quality. Increasing E_{steps} improves the estimation of $q(z|x)$ since the prior and generator parameters are changing constantly. Rows 1-3 of Table 8.4 show that the reconstruction error (MSE) decreases with E_{steps}. Generation quality metrics (IS, FID) also improve with larger values of E_{steps} due to the shared parameters between encoder and discriminator.

At initialization the GMM components might not be separated. We observed that the clustering accuracy drops when the generators learns a good approximation of the real distribution before the clusters are separated. To avoid this we use a larger learning rate for the

Table 8.4: E_{steps} stands for the encoder updates and η_p for the learning rate of the prior parameters. The scale of MSE is in 10^{-3}. The statistics were obtained for at least three runs.

E_{steps}	η_p	Acc %	IS	FID	MSE
1	2e-4	41.31	8.82	11.38	1.34
		±5.74	±0.07	±0.23	±0.96
2	2e-4	38.67	9.02	9.66	1.01
		±3.52	±0.05	±3.98	±1.11
4	2e-4	38.27	9.25	7.50	0.331
		±2.46	±0.09	±0.43	±0.09
4	4e-4	52.58	9.44	6.55	0.48
		±5.30	±0.06	±0.33	±0.22
4	6e-4	61.99	9.49	6.59	1.04
		±4.96	±0.15	±0.45	±1.03

Table 8.5: Comparison of MPCC and AIM-MPCC methods with sharing parameters (S) and without sharing (NS) on the CIFAR-10 dataset. The scale of MSE is in 10^{-3}. The statistics were obtained for five runs.

Model	Acc %	IS	FID	MSE
AIM-MPCC (NS)	-	8.24	21.55	1.52
		±0.07	±1.47	±0.84
AIM-MPCC (S)	-	9.09	10.42	1.64
		±0.04	±0.36	±1.42
MPCC (S)	61.99	9.49	6.59	1.04
	±4.96	±0.15	±0.45	±1.03

parameters of the GMM prior with respect to the parameters of the generator, encoder and discriminator. Rows 4-6 of Table 8.4 show that the clustering accuracy increases for larger values of η_p.

8.2.2 Comparison between GMM Prior and Normal Prior

Using the best configuration found in the ablation study, we performed a comparison with AIM [69], whose results are shown in Table 8.5. We can consider AIM as a particular case of MPCC where a standard Normal prior is used instead of the GMM prior. AIM does not perform clustering therefore we compare it with MPCC in terms of reconstruction and generation quality. We use the same architecture and parameter settings of MPCC for AIM and we denote this model as AIM-MPCC. To extend our analysis further, Table 8.5 includes the results of using parameter sharing between the encoder and the discriminator (section 8.1.4), an idea that was considered but not fully explored in [69].

Note that AIM-MPCC (NS) is considered a baseline because the prior is Gaussian and the encoder doesn't share parameters with the discriminator thus the existence of the encoder doesn't affect the generation quality. In Table 8.5 we can observe the relevance of parameter sharing, with this configuration ($E_{steps} = 4$) the baseline improves by 0.85 (IS) and 11.13 (FID) points. Adding the GMM in the prior improves an additional 0.4 (IS) and 3.93 (FID) points. In total when using the GMM Prior and parameter sharing with additional encoder updates we improve the baseline from 21.55 to 6.59 (69.4% improvement) in terms of FID score and 1.25 points (15.2% improvement) in terms of IS. It is important to notice that these techniques are general and can be easily applied to any GAN scheme.

8.2.3 Generation quality of MPCC

Using the configuration of row five from Table 8.4 we compare MPCC with nine state of the art methods (as of July 2020), surpassing them in terms of IS and FID scores in both the unsupervised and supervised setting, as shown in Table 8.6. The unconditional generation

Table 8.6: Inception and FID scores for CIFAR-10, in unconditional and conditional training. Higher IS is better. Lower FID is better. †: Average of 10 runs. ‡: Best of many runs. ††: Average of 5 runs. Results without symbols are not specified.

Model	IS	FID
DCGAN [91]	6.64 ± 0.14	–
SN-GAN† [78]	8.22 ± 0.05	21.7 ± 2.1
AutoGAN [27]	8.55 ± 0.10	12.42
PG-GAN ‡ [48]	8.80 ± 0.05	–
NCSN [102]	8.91	25.32
MPCC††	9.49 ± 0.15	6.59 ± 0.45

(a) Unconditional (unsupervised) generation

Model	IS	FID
WGAN-GP [31]	8.42 ± 0.10	–
SN-GAN † [78]	8.60 ± 0.08	17.5
Splitting GAN ‡ [30]	8.87 ± 0.09	–
CA-GAN † [83]	9.17 ± 0.13	–
BigGAN [8]	9.22	–
MPCC††	9.55 ± 0.08	5.69 ± 0.17

(b) Conditional (supervised) generation

is the most significant with an improvement of 46.9% (FID) over state-of-the-art (SOTA), AutoGAN [27]. Most notably its performance is better than the current best conditional method (BigGAN).

8.2.4 Clustering results

Table 8.7 shows the clustering results for the selected benchmarks. We observe that in all the available benchmarks MPCC outperform the related methods, VADE [46] and ClusterGAN [79]. In more complex datasets such as CIFAR10, MPCC notably surpass discriminative based models (*e.g.* [41], [45]) which are the most competitive methods in the current literature. For benchmarks with more classes the margin is even larger obtaining improvements over the SOTA of $\sim 42\%$ and $\sim 9.7\%$ points in Omniglot and CIFAR-20 respectively, demonstrating empirically the scalability of MPCC when using a high number of clusters.

It can be observed that for all datasets our proposed method either achieves or surpasses the SOTA in terms of clustering. Figures 8.3 and 8.4 show examples of generated and reconstructed images, respectively, using the MPCC model with the highest accuracy in the MNIST and CIFAR-10 datasets.

8.3 Discussion

Our results show that MPCC achieves a superior performance with respect to the SOTA on both clustering and generation quality. We note that the current SOTA on unsupervised and semisupervised learning relies on consistency training [117] and/or data augmentation [45], *i.e.* techniques that are complementary to MPCC and could be used to further improve our results.

To the best of our knowledge MPCC is the first deep generative clustering model capable of dealing with more complex distributions such as CIFAR-10/20 and the first to report clustering accuracy on these datasets. Additionally, we empirically prove the scalability of MPCC showing significant improvements in datasets with a larger number of classes, 20 in case of CIFAR-20 and 100 in case of Omniglot, such scalability has not been proven for the current literature on generative models [46], [79], [103], [118].

Table 8.7: Clustering accuracies for several methods and datasets. All the results of CIFAR-20 dataset were extracted from [45], the results of IMSAT and DEC from [41], the results of InfoGAN and ClusterGAN from [79] and the remaining from their respective papers. †: average of 5 or more runs. ‡: best of 5 runs. §: best of 10 or more runs. $^{\|}$: best of 3 runs. Results without symbols are not specified.

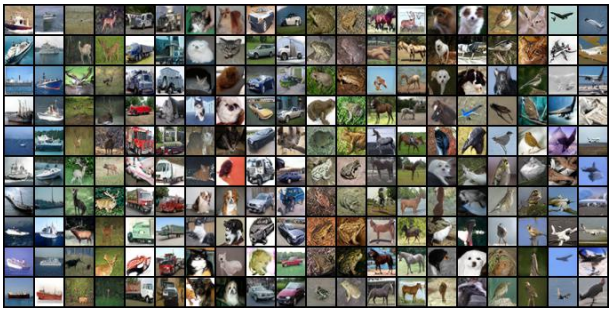
	Datasets				
Methods	MNIST	Onmiglot	FMNIST	CIFAR-10	CIFAR-20
DEC [116]	$84.3^{\S}$	$5.3 \pm 0.3^{\dagger}$	–	$46.9 \pm 0.9^{\dagger}$	18.5
VADE [46]	$94.46^{\S}$	–	–	–	-
InfoGAN [14]	$89.0^{\ddagger}$	–	$61.0^{\ddagger}$	–	-
ClusterGAN [79]	$95.0^{\ddagger}$	–	$63.0^{\ddagger}$	–	-
DAC [11]	$97.75^{\|}$	–	–	$52.18^{\|}$	23.8
IMSAT (VAT) [41]	$98.4 \pm 0.4^{\dagger}$	$24.0 \pm 0.9^{\dagger}$	–	$45.6 \pm 0.8^{\dagger}$	-
ADC [33]	$98.7 \pm 0.6^{\dagger}$	-	-	$29.3 \pm 1.5^{\dagger}$	16.0
SCAE [57]	$98.5 \pm 0.10^{\dagger}$	-	-	$33.48 \pm 0.3^{\dagger}$	-
IIC [45]	$98.4 \pm 0.65^{\dagger}$	-	-	$57.6 \pm 0.3^{\dagger}$	$25.5 \pm 0.46^{\dagger}$
MPCC (Five runs)	98.48 ± 0.52	65.87 ± 1.46	62.56 ± 4.16	64.25 ± 5.31	35.21 ± 1.69
MPCC (Best three runs)	98.76 ± 0.03	66.95 ± 0.62	64.99 ± 2.22	67.73 ± 2.50	36.51 ± 0.71

Our experiments show that MPCC's key innovations: GMM prior, loss function and optimization scheme (*e.g.* extra encoder updates with parameter sharing) are not only relevant to achieve a good clustering accuracy but also allows us to obtain unprecedented results in terms of generation quality (Table 8.6). Which translates in improvements of 69.4% over the baseline (Table 8.5) and 46.9% over the SOTA (Table 8.6) in terms of FID score. We think that the exceptional generation capabilities of MPCC are related to the support that each cluster covers of the real domain. Since each cluster learns a subset of the real distribution the interpolation between two points within a cluster is smoother compared to the case where no latent separation exists. The latter is explained by the learnable shared features which exploit the similarities existing in a cluster and are not present in a fixed global prior (*i.e.* ALI, AIM).

The high generation quality can be appreciated in Fig. 8.3 (more samples in Appendix D), where many clusters sample consistently different classes. However we can still see some classes mixed in some clusters, for example in columns 7-8 with cats and dogs. MPCC also presents a competitive performance in terms of conditional distribution matching (Fig. 8.4). The errors observed in reconstruction are semantic and similar to those observed in [20].

MPCC opens the possibility of future research in many relevant topics which are out of the scope of this thesis. Based on our experiments the most important extensions are: 1) Experiment with other conditional distributions $p(z|y)$, *e.g.* other exponential-family distributions or other flexible distributions by bounding their entropy (Section 7.2.2). This can be suitable for more expressive priors as it's shown in recent work [114]. 2) Experiment with imbalanced distribution of classes by changing ϕ accordingly, we consider this to be a relevant problem in the unsupervised setting which only a few works have addressed [105]. 3) Experiment with higher resolution datasets such as ImageNet [18] or CelebA [71]. Current works on clustering have not focus their attention to higher-resolution due to its complexity,

(a) MNIST samples



(b) CIFAR-10 samples

Figure 8.3: Generated images for a) MNIST and b) CIFAR-10 datasets, respectively. Every two columns we set a different value for the categorical latent variable y. *i.e.* the samples shown correspond to a different conditional latent space $z \sim p(z|y)$.

(a) MNIST reconstruction

(b) CIFAR-10 reconstruction

Figure 8.4: Reconstructions for a) MNIST, and b) CIFAR-10 datasets, respectively. Odd columns represent real data and even columns correspond to their reconstructions.

MPCC is a promising approach to tackle this task from a semantic perspective [20].

Chapter 9

An astronomical application with Variational Deep Embedding

In this chapter we validate the practical application of generative-inference models on real data from astronomical light curve datasets. A light curve is a time series of the luminosity of an astronomical object. Astronomers analyze the light curves to obtain insight about the underlying physical processes of the objects [87]. Light curve analysis is particularly important to study transient and variable astronomical objects. Light curves captured by earth-based telescopes are characterized by their irregular sampling and non-constant errors (heterocedasticity), which hinders the application of classical methods for time series analysis.

In recent years, astronomy has seen an exponential growth in data volume, speed and complexity due to the installation of large panoramic telescopes [42]. An emblematic example is the Vera Rubin Observatory with its main program the Legacy Survey of Space and Time (LSST) [106], which will produce 20TB of data every night. The application of machine learning in these cases is necessary for the automatic analysis of the data. Moreover models and techniques that can exploit the supervised and the unsupervised information will be key. As seen in this thesis generative-inference models can be trained without labels by compressing the real data into a lower dimensional latent space for further applications.

In this chapter we tackle the problem of anomaly detection [10] in astronomical datasets. This problem is formulated as separating data $\bar{x}$ that didn't come from the real distribution $q(x)$ where the model was trained. Solving this problem is key to discover new objects never observed before that may contribute to the literature as new astrophysical theories. We propose to select anomalous data using a new unsupervised information score and a model called LC-VaDE which adapts Variational Deep Embedding (VaDE) for astronomical lightcurve datasets. This model is trained using unsupervised and supervised information available in the dataset.

The use of VaDE in this application is justified based on the following observations

1. VaDE is a model that bounds $\mathcal{I}_q(x, z) + \mathbb{E}_{q(x)}\mathcal{I}_q(z, y|x)$ so it explicitly optimizes the mutual information of the inference model, which is directly associated with its repre-

sentation learning capabilities.

2. VaDE, as a model that bounds $\mathcal{I}_q(x, z)$, optimizes the likelihood $\mathbb{E}_{q(x,z)}[\log p(x|z)]$. In what follows we will show a closed-form solution to estimate this quantity for astronomical lighcurve datasets. In these datasets a simple optimization of $\mathbb{E}_{q(x,z)}[\log p(x|z)]$ is preferred over an implicit adversarial training like MPCC since their dimensional complexity is less than that of image datasets.

In astronomical lightcurve datasets an estimation of the associated error σ of the luminosity μ is generally available. To exploit this information we can estimate $\hat{\mu}$ and $\hat{\sigma}$ with $p(x|z)$ to reduce the cross-entropy with the available statistics of the data μ and σ. Note that the cross-entropy $-\int \mathcal{N}(\mu, \sigma^2) \log \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ have a closed form solution for the Gaussian distribution, a common assumption when dealing with lightcurve astronomical datasets [40].

As explained before, astronomical lightcurve datasets have the advantage that it is possible to optimize models based on the cross-entropy in the observed space but they still have other difficulties. There are two main issues that we have to take into consideration when working with lightcurve datasets:

1. The length of each lighcurve is variable
2. The observation times in a lightcurve are irregularly and sometimes sparsely sampled

These difficulties are of special consideration for generative-inference models, because these models consider an encoder and a decoder, and both should be able to deal with variable length and irregularly sampled time series. One alternative to deal with this difficulty is to use recurrent neural networks (RNN) [39]. RNNs can deal with variable length but they don't scale well for long sequences or when many RNNs modules are stacked. For this reason we will use architectures based on convolutional neural networks [59, 62] to encode and decode lightcurve data. Some encoders approaches can encode data using irregularly sampled time series [100], but there are not many decoders that can decode variable length data with irregularly sampled time series. To use convolutional based architectures we propose a new decoder that takes advantage of the Gaussianity of $p(x|z)$ to infer data sampled at arbitrary times t, similar to a Gaussian process [92]. The optimization of this decoder is possible thanks to the simplicity of autoencoding procedures like VaDE.

The main contributions of this work are:

- A new decoder that is able to deal with irregularly sampled and variable-length times series.
- A new generative-inference model called LC-VaDE based on VaDE to train a latent variable mixture model for astronomical light curves.
- A new anomaly detection score based on information theory.

9.1 Related work

Variational deep embedding (VaDE) is a VAE model that changes the common prior $p(z) = \mathcal{N}(0, I)$ to a learnable mixture of Gaussians. This modification in the prior allows VaDE to perform applications like clustering [2, 46] but to the best of our knowledge it hasn't been

applied to irregularly sampled time series or for anomaly detection in time series datasets. Previous work [123] have used autoencoding models with mixture of Gaussians in its embedding to perform anomaly detection. To the best of our knowledge this is the first attempt of using a generative-inference model with a multimodal prior for this task. There are two main differences of our work with [123]: 1) [123] is not a generative model and 2) its anomaly detection score is based on energy functions [64] instead of information theory. We will compare our model with [123] since it is the most similar method in the literature and it also has been used in astronomical applications [109].

9.2 Astronomical datasets

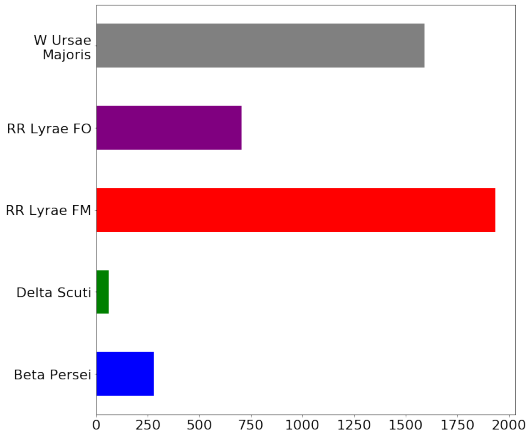
Following previous autoencoding work [80], we consider the All Sky Automated Survey [56] (ASAS) and Linear [86] variable star lightcurve datasets. Fig. 9.1 shows a histogram of the periodic variable star classes present in these datasets. A variable star is a star whose luminosity changes over time. The luminosity of an object i is obtained by taking pictures of the sky for a given passband frequency filter $b = 1, \ldots, B$ and at times $t_{i,j,b}$, where j denotes the j-th picture of object i. Photometry [81] is then used to estimate the magnitude (relative flux) $\mu_{i,j,b}$ and its associated error $\sigma_{i,j,b}$ for every object, picture and band. The tuples $(t_{i,j,b}, \mu_{i,j,b}, \sigma_{i,j,b})$ defines the lightcurve dataset. The number of tuples per object and per band varies across the dataset. Additionally $t_{i,j,b}$ is sampled irregularly, which increases the difficulty of astronomical lightcurve datasets.

For periodic variable stars it is possible to fold the lightcurve using

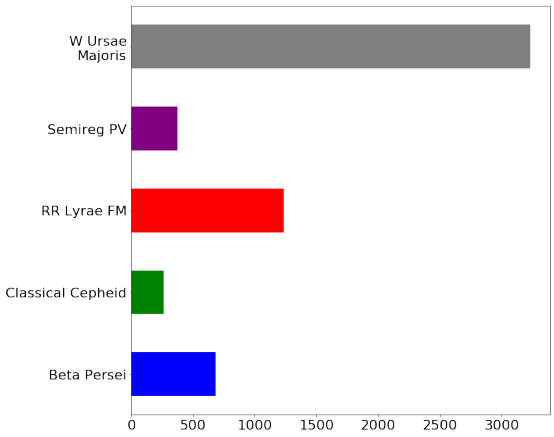
$$t^{\text{fold}} = \mod (t^{\text{unfold}}, p)/p, \tag{9.1}$$

where p is the period of the lightcurve and $\mod$ is the modulus operator. The result of this operation normalizes the observation times t^{unfold} to a phase space $t^{\text{fold}} \in [0, 1]$. In practice we will use t^{fold} in all experiments and denote it simply as t. For an arbitrary $t_{i,j,b}$ the index j can take the values $j \in 1 \ldots T_b$, being T_b the length of the lightcurve i for band b. In the datasets considered in this work the maximum value of T_b is 200 and number of bands B is 1.

For the anomaly detection problem we will separate all the data in a training and test sets. The test set contains N_{anomaly} number of samples from the anomaly class and a equal number of N_{anomaly} samples from the inlier class preserving the balance of classes of the inlier data. The training set is constructed with all the inlier data that is not used for testing. N_{anomaly} is defined as the number of samples from the minority class of the dataset. We use each class of the dataset as outlier in different experiments. We don't use a validation set since in practice it is not feasible to have outlier data available. For each lightcurve we normalize its statistics with the mean m of the magnitudes and standard deviation s of the magnitudes of the lightcurve itself. For training we consider additional features f that can be useful to the model for the task of classification. In this work we will consider the vector $f = (m, s, p)$ as additional features.



(a) Linear dataset



(b) Asas dataset

Figure 9.1: Histogram of the classes available in the Linear and ASAS datasets.

9.3 A decoder for variable length and irregular sample time series

Astronomical lightcurves have variable length and irregular sampling and we need both an encoder and a decoder that can deal with these difficulties. For the encoder we use the interpolation prediction network [100] followed by a CNN architecture[1]. For the decoder we propose a new framework based on Gaussian process. In Gaussian process usually a collection of observed data x_i, indexed by their times t_i, form a multivariate Gaussian distribution that is used to predict data on times t^{pred}. In our case we will learn induction points x_i^{ind}, indexed by fixed times t_i^{ind}, to form a multivariate Gaussian distribution that will be used to predict data on times t^{pred}. We can take t^{pred} from the empirical distribution $t_i, x_i \sim q(x,t)$ to have an estimate $\hat{x}_i$ of the observed data x_i. In what follows we give a general and a more detailed overview of the proposed procedure.

We start by encoding the data $\mu(t_{i,j,b})$, $\sigma(t_{i,j,b})$ into latent variables z. Using a CNN architecture we decode the latent variables z into a fixed amount of induction points $\hat{x}_{i,b}^{\text{ind}}(t_k) = (\hat{\mu}_{i,b}^{\text{ind}}(t_k^{\text{ind}}),\ \hat{\sigma}_{i,b}^{\text{ind}}(t_k^{\text{ind}}))$ that are associated with equally spaced times t_k^{ind} with $k \in 1 \dots I$, being I the number of induction points. These induction points are used to infer $\hat{\mu}(t_{i,j,b})$, $\hat{\sigma}(t_{i,j,b})$ at the irregularly sampled instants $t_{i,j,b}$, which are the times associated to data. We train the model by maximizing the likelihood $\hat{\mu}(t_{i,j,b})$, $\hat{\sigma}(t_{i,j,b})$ with respect to the empirical observations $\mu(t_{i,j,b})$, $\sigma(t_{i,j,b})$. The gradients backpropagate to learn the induction points. We also use backpropagation to learn the kernel parameters θ^{kernel} typically used in Gaussian process. This Gaussian process is amortized since the neural network can compute its parameters in one forward pass.

The coding and decoding processes are shown in Fig. 9.2. Note that any autoencoding model can be used, generative or not. The detailed steps of the autoencoding process are the following

[1]More details of the architecture are given in Section 9.5.2

1. Sample $(x_i, t_i) \sim q_\delta(x, t)$ from the empirical data distribution, where $x_i = (\mu_i, \sigma_i)$.
2. Encode the data $x_i = (\mu_i, \sigma_i)$ with the encoder $q(z|x)$ into a lower dimensional space (see section 9.4 for the architecture used). We set $q(z|x)$ as a Gaussian distribution defined by the statistics $\tilde{\mu}_i, \tilde{\sigma}_i$. Using the reparameterization trick we sample $z_{i,h} \sim q(z|x = x_i)$ with h referring to a random sample from this distribution, in the following we use notation z_i for simplicity.
3. We separate the codification as $z_i = [z_i^{\text{rep}}, z_i^{\text{kernel}}]$. We make this distinction to separate the representation learning information of the lightcurve in z^{rep} from the necessary information to estimate the kernel parameters in z^{kernel} (Eq. (9.3)).
4. Decode z_i^{rep} with $p(x^{\text{ind}}|z)$. The distribution $p(x^{\text{ind}}|z)$, parameterized by a CNN architecture (see details in section 9.5.2), decodes z_i^{rep} into a fixed amount of induction points $\hat{x}_{i,b}^{\text{ind}}(t_k) = (\hat{\mu}_{i,b}^{\text{ind}}(t_k^{\text{ind}}),\ \hat{\sigma}_{i,b}^{\text{ind}}(t_k^{\text{ind}}))$. Note that each induction point $\hat{x}_{i,b}$ is associated with fixed and regularly sampled times t_k defined as

$$t_k = \frac{k}{I+1} \quad k \in 1 \dots I, \tag{9.2}$$

   where t_k does not depend on the data point i. The total number of induction points is given by $I \times B$, where I is defined by the architecture, B is the number of bands. Note that that the number of neurons necessary to decode is $2 \times I \times B$ since x is formed by the tuple $(\hat{\mu},\ \hat{\sigma})$.
5. Decode z_i^{kernel} with $p(\theta^{\text{kernel}}|z)$. The distribution $p(\theta^{\text{kernel}}|z)$, parameterized by a NN architecture (more details in the section 9.5.2), decodes z_i^{kernel} into the kernel parameters θ_i^{kernel}. In practice we use an exponential quadratic kernel for all of our experiments defined as

$$k_{\theta_i}(t, t') = \sigma_{\theta_i}^2 \exp\left(-\frac{||t - t'||^2}{2l_{\theta_i}^2}\right), \tag{9.3}$$

   where its parameters are defined by $\theta_i^{\text{kernel}} = (\sigma_{\theta_i}, l_{\theta_i})$ and are learned through back-propagation (see architecture details in section 9.5.2).

   The kernel defines the covariances $\hat{\Sigma}^{\text{ind,ind}}$, $\hat{\Sigma}^{\text{ind,obs}}$ and $\hat{\Sigma}^{\text{obs,obs}}$. The sub-index "obs" refers to covariances obtained by using the real times t. The sub-index "ind" refer to the covariances obtained by using the induction points times t^{ind}. The covariance $\hat{\Sigma}^{\text{ind,ind}}$ can be modified depending on the noise $\hat{\sigma}_i^{\text{ind}}(t_k^{\text{ind}}))$ of each induction point as follows:

$$\hat{\Sigma}_i^{\text{ind,ind}} = \begin{bmatrix} k_{\theta_i}(t_1^{\text{ind}}, t_1^{\text{ind}}) + (\hat{\sigma}_i^{\text{ind}}(t_1^{\text{ind}}))^2 & \dots & \dots & k_{\theta_i}(t_I^{\text{ind}}, t_I^{\text{ind}}) \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ k_{\theta_i}(t_I^{\text{ind}}, t_1^{\text{ind}}) & \dots & \dots & k_{\theta_i}(t_I^{\text{ind}}, t_I^{\text{ind}}) + (\hat{\sigma}_i^{\text{ind}}(t_I^{\text{ind}}))^2 \end{bmatrix} \tag{9.4}$$

6. Using the induction points we can compute the posterior distribution $p(\hat{x}|\hat{x}^{\text{ind}}, t^{\text{ind}}, t)$ to estimate $\hat{x}$ at times t that can be compared to the observed data x. Note that the induction points are Gaussian so the posterior $p(\hat{x}|\hat{x}^{\text{ind}}, t^{\text{ind}}, t)$ is Gaussian too and can be computed in closed-form as follows:

$$p(\hat{x}|\hat{x}^{\text{ind}}, t^{\text{ind}}, t) = \mathcal{N}(\underbrace{\hat{\Sigma}^{\text{obsobs,ind}}(\hat{\Sigma}^{\text{ind,ind}})^{-1}\hat{\mu}^{\text{ind}}}_{\hat{\mu}}, \underbrace{\hat{\Sigma}^{\text{obsobs,obsobs}} - \hat{\Sigma}^{\text{obsobs,ind}}(\hat{\Sigma}^{\text{ind,ind}})^{-1}\hat{\Sigma}^{\text{ind,obsobs}}}_{\hat{\Sigma}}) \tag{9.5}$$

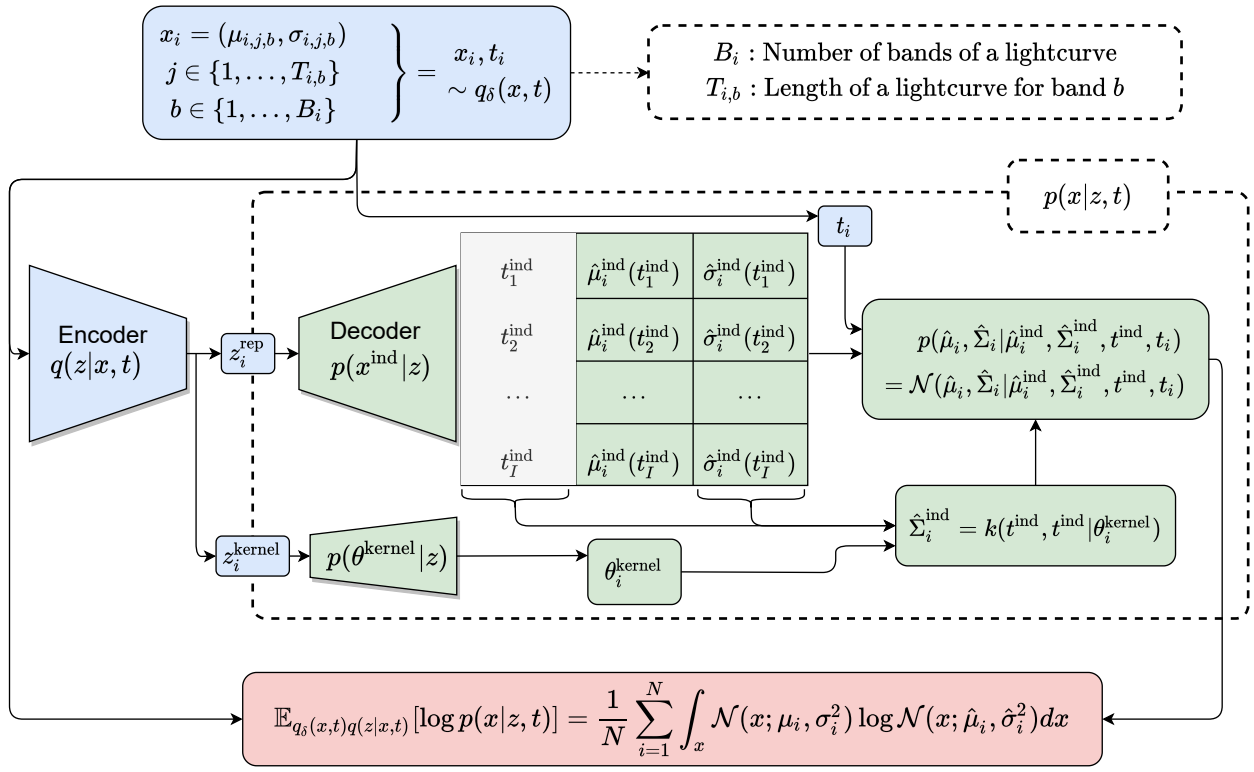


Figure 9.2: Diagram of the autoencoding process for astronomical light curves using the proposed decoder.

7. For simplicity we consider the diagonal of $\hat{\Sigma}$ to define the posterior as a Normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$. Since the posterior distribution is Normal and each sample of the empirical data distribution is also Normal we can estimate the likelihood as

$$\mathbb{E}_{q_\delta(x,t)q(z|x,t)}[\log p(x|z,t)] = \frac{1}{N}\sum_{i=1}^{N}\int_x \mathcal{N}(x;\mu_i,\sigma_i^2)\log\mathcal{N}(x;\hat{\mu}_i,\hat{\sigma}_i^2)dx \tag{9.6}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{b=1}^{B}\sum_{j=1}^{T_b} -\log(2\pi\hat{\sigma}_{i,j,b}^2) - \frac{\sigma_{i,j,b}^2}{2\hat{\sigma}_{i,j,b}^2} - \frac{(\mu_{i,j,b}-\hat{\mu}_{i,j,b})^2}{\hat{\sigma}_{i,j,b}^2} \tag{9.7}$$

We refer to [46] for the demonstration of the cross-entropy for Gaussian distributions.

9.4 Anomaly detection with LC-VaDE

9.4.1 Graphical model

For anomaly detection we will train LC-VaDE with labels to be comparable with the baseline [109]. We will consider the following generative-inference graphical model,

- $p(x, z, y, f|t) = p(x|z,t)p(z|y)p(f|y)p(y)$
- $q(x, z, y, f, t) = q(x, f, t)q(z|x,t)q(y|z,f)$.

The structure of LC-VaDE is similar to VaDE [46], a three variable graphical model. LC-VaDE additionally considers the addition of the time t and features f (described in section 9.2) used for classification. The joint distribution $q(x, f, t)$ includes the observable data x, features f and the time vector t of the observable data. We encode the data with $q(z|x, t)$ using x and times t. We classify the data to a mode of the prior y with $q(y|z, f)$. For $q(y|z, f)$ we use the information from z the codified data and the additional features f. For the prior distribution we use a categorical prior $p(y)$, and the conditional priors $p(z|y)$, $p(f|y)$ that are learnable Gaussians dependant on y. The decoder $p(x|z, t)$ is used to decode the data from z and t. Note that we impose the dependency t in $p(x|z, t)$ since we want to learn the underlying process to generate x, and t are the observations times available from that underlying process.

The proposed graphical model can be obtained by decomposing the inference model as $q(x, z, y, f, t) = q(x, z, y)q(z|x, f, t)q(y|x, z, f, t)$ and the generative model as $p(x, z, y, f|t) = p(x|z, y, f, t)p(z|f, y, t)p(f|y, t)p(y|t)$ and making the following independence assumptions. For $q(z|x, f, t)$ we want the latent variable z to be a codified version of x and not for the additional features f. With this independence in consideration the posterior z simplifies as $q(z|x, f, t) = q(z|x, t)$. For the task of classification executed by $q(y|x, z, f, t)$ we assume the data x and the time t are already codified in z, so to estimate y only z and f are needed simplifying the posterior of y as $q(y|x, z, f, t) = q(y|z, f)$. For the generative model we assume that all latent variables and f are independent of the time t. This assumption is possible since t stands for the available time observations of x and are not part of the underlying process of x. To estimate x we only need z and t since x and f are independent and z contains the categorical information of y. With these considerations the decoder $p(x|z, y, f, t)$ simplifies to $p(x|z, t)$. Since we also assume the independence of z and f, $p(z|y, f, t)$ simplifies to $p(z|y)$. Finally with the time independency consideration $p(f|y, t)$ simplifies to $p(f|y)$.

9.4.2 Loss function of LC-VaDE

For the training of LC-VaDE we consider supervised and unsupervised information. We refer to the model that use both sources of information as LC-VaDES. LC-VaDES minimizes the following loss function

$$\mathcal{L}^{\text{LC-VaDES}} = \mathcal{L}^{\text{LC-VaDE}} + h(y, \tilde{y}), \tag{9.8}$$

where y is the real label and $\tilde{y}$ is the label estimated by the model. The term $h(y, \tilde{y})$ refers to the cross-entropy between both distributions. We obtain $\tilde{y}$ by sampling data $x_i, f_i, t_i \sim q_\delta(x, f, t)$ and encoding it with $\tilde{z}_i \sim q(z|x = x_i, t = x_i)$ and $\tilde{y}_i = q(y|z = z_i, f = f_i)$. $\mathcal{L}^{\text{LC-VaDE}}$ refers to the ELBO of the marginal distribution $p(x|t)$ obtained using Jensen's Shannon inequality as follows:

$$\begin{aligned}
\mathbb{E}_{q(x,t)}[\log p(x|t)] &\geq \mathbb{E}_{q(x,z,y,f,t)}\left[\frac{p(x, z, y, f|t)}{q(x, z, y, f, t)}\right] && (9.9)\\
&\equiv \mathbb{E}_{q(x,t)q(z|x,t)}[[\log p(x|z, t)] + \mathbb{E}_{q(y|z,f)}[\log p(z|y)] \\
&+ \mathbb{E}_{q(f)q(y|z,f)}[\log p(f|y)]] + \mathbb{E}_{q(y)}[\log p(y)] \\
&+ h_q(y|z) + h_q(z|x) && (9.10)\\
&\equiv -\mathcal{L}^{\text{LC-VaDE}}. && (9.11)
\end{aligned}$$

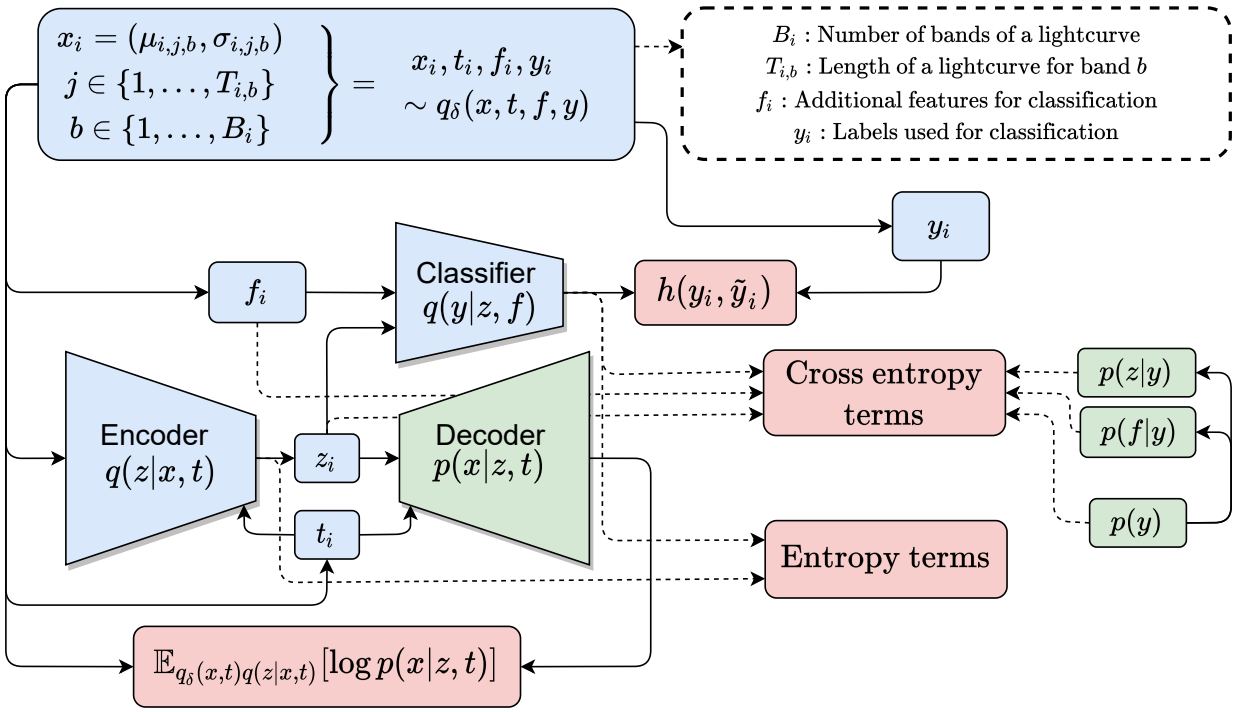


Figure 9.3: Diagram of the LC-VaDE loss function. The terms $\mathbb{E}_{q(y|z,f)}[\log p(z|y)]$, $\mathbb{E}_{q(f)q(y|z,f)}[\log p(f|y)]]$ are represented by the "cross-entropy terms" red box. The terms $h_q(y|z)$, $h_q(z|x)$ are represented by the "entropy terms" red box.

Note that the ELBO is trained in an unsupervised manner and the supervised information is only used for the cross-entropy term included in Eq. (9.8). All the terms in Eq. (9.10) can be computed in closed-form similarly to [46], since all distributions are either Gaussian or categorical. To obtain a closed-form solution for the variable f, the distribution $q(f)$ is assumed to be $\mathcal{N}(f_i, \sigma_{f_i}^2)$ with $\sigma_{f_i}^2 \to 0$. We assumed that the prior distribution $p(y)$ is uniform since we force label balance during training. For the rest of this work we refer to LC-VaDES as LC-VaDE, when there is no ambiguity. Fig. 9.3 shows all the terms involved in the computation of the loss function of LC-VaDE.

9.4.3 Anomaly detection score

For anomaly detection we prefer a score based on the information obtained from the latent variables. We avoid including information about the observed space because it could perform badly in situations where outlier lightcurves are less noisy than inlier light curves. Models that include reconstruction error in the observed space could bias the model [123, 73], treating an object as outlier only because it is noisier. For this reason we decide to compress the first two statistics of the data *i.e.* $x_i = (\mu_i, \sigma_i)$. If $x_i = (\mu_i, \sigma_i)$ then its codified version $z_i(x_i)$ should contain all the information of μ_i and σ_i.

We choose to measure the mutual information $\mathcal{I}_q(z, y|x)$ between the codified data $z_i(x_i)$ and each mode from the prior $p(y)$ given the data x_i. Since this mutual information has an

intractable term we use the following bound

$$\begin{aligned}\mathcal{I}_q(z, y|x, f) &= \mathbb{E}_{q(z,y|x,f)}[\log p(z|y)] + h_q(z|x) + \mathbb{E}_{q(x,y)}[D_{KL}(q(z|y, x)||p(z|y))] \\ &\geq \mathbb{E}_{q(z,y|x,f)}[\log p(z|y)] + h_q(z|x) \equiv \text{IAS}(x),\end{aligned} \tag{9.12}$$

where we have assumed $D_{KL}(\cdot) \geq 0$. We call this expression Information Anomaly Score (IAS). Intuitively the IAS score considers three factors: (1) the prediction of the classifier $q(y|z, f)$, (2) the likelihood of $q(z|x)$ in the conditional prior $p(z|y)$ and (3) the entropy $h_q(z|x)$. The factors (1) and (2) are collected by the term

$$\mathbb{E}_{q(z,y|x,f)}[\log p(z|y)] = \mathbb{E}_{q(y|z,f)q(z|x)}[\log p(z|y)] = \sum_{k=1}^{K} q(y|z(x_i), f_i) \int_z \mathcal{N}(\tilde{\mu}_i, \tilde{\sigma}_i^2) \log \mathcal{N}(\mu_k, \sigma_k^2),$$

where for simplicity the subindexes i and k refer to the statistics of the encoded data and the prior parameters, respectively. This term weights the prediction $q(y = k|x, f)$ with the likelihood of the encoded data in the k-th Gaussian of the prior. This means that $\mathbb{E}_{q(z,y|x,f)}[\log p(z|y)]$ and thus the IAS score will be low if either the entropy of $q(y|z)$ is high or the likelihood $\mathbb{E}_{q(y|z,f)}[\log p(z|y)]$ is low. The last factor implies that a high entropy $h_q(z|x)$ augments the IAS score. We interpret the entropy $h_q(z|x = x_a)$ as how much latent space the input x_a fill. Probably an input with higher entropy $h_q(z|x)$ is more confundable with other data than inputs with less entropy. An input x_b with higher entropy $h_q(z|x = x_b)$ probably contains features that are maximized in training, thus the entropy should be higher for inlier inputs than outlier inputs.

9.5 Experiments

In this section the experimental setup for the anomaly detection task is provided. In Section 9.5.1 we explain the metrics used to evaluate the anomaly detection performance. In Section 9.5.2 we give the architectural details of our work and in Section 9.5.3 we present the qualitative and quantitative results. Finally in section 9.5.4 we discuss the impact of our work and future contributions.

9.5.1 Metrics

In a practical anomaly detection problem it is not possible to choose a specific threshold to separate inlier and outlier data since usually a validation set is not available. Due to this we use a grid of thresholds and compute the Area under the Receiver Operating Characteristic curve (AUCROC) and area under the precision-recall (AUCPR), as is commonly used in the literature [26].

9.5.2 Architecture details

Following the experiments shown in previous chapters, we use the BigGAN model techniques [8] as a base for all our experiments. We also employ ResNet [34] architectures and Spectral Normalization [78]. We followed the same residual block components of the generator (in this case the decoder) and encoder shown in figures 6.1 (a) and (b), respectively, although in this case a discriminator is not used. All the 3×3 convolutions use a padding equal to

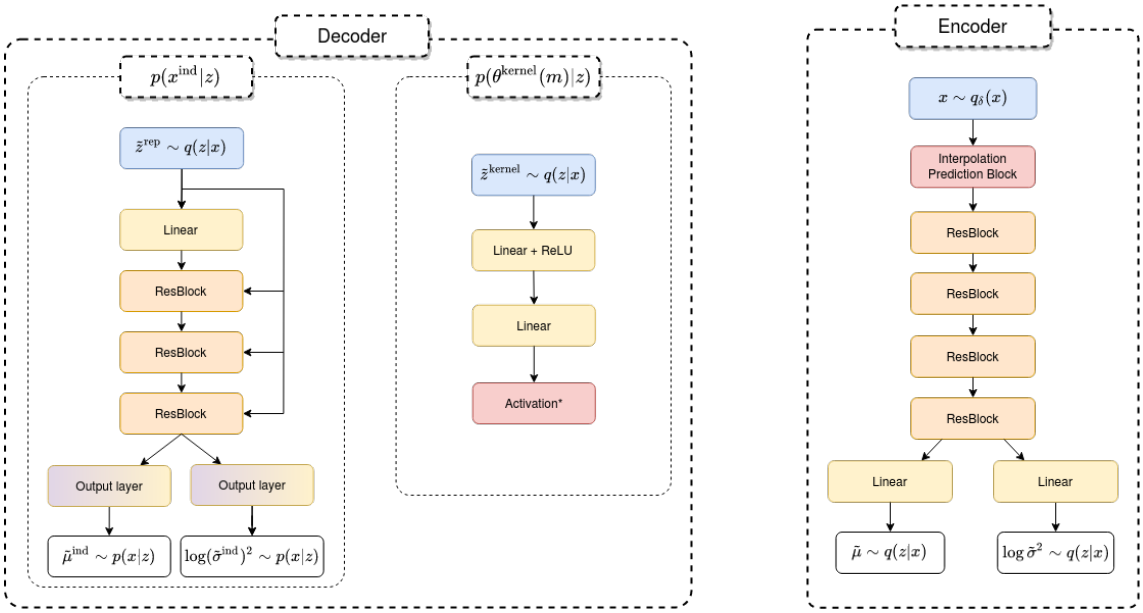
one while 1×1 convolutions have no padding. The upsampling operation of the generator residual block is done using bilinear interpolation ("Resblock up" in Table 6.1) increasing the width dimension by two. The downsampling operation of the encoder residual block is done using average pooling with kernel size three ("Resblock down" in Table 9.2). A general scheme of the decoder and encoder architecture is shown in Fig. 9.4.

The interpolation prediction encoder [100] is a learnable filter operation that can reduce a variable amount of input data into a fixed amount (in our case 64). It performs low-pass and high-pass interpolations and cross-correlation of the input bands (if $D > 1$, D the dimensionality of the input of the network). Because it performs three operations it increases the input dimension by three (see Table 9.2). The first residual block of the encoder inverts the order of the 1×1 Conv and the average pooling and omits the first ReLU activation. We can write the architectures for all datasets in general way as in Fig. 9.4 or more specific as in Tables 6.1, 6.2 and 6.3, where C, J and D may change between datasets. In this case, for both Linear and ASAS and in all our experiments $C = 72$, $J = 10$, $(\dim(z^{\text{rep}}) + \dim(z^{\text{kernel}}) = 10 + 2 = 12)$ and $D = 2$ (Number of bands μ + Number of bands $\sigma = 1 + 1 = 2$). For a general case note that $D = 2 \times B$.

The latent z^{rep} is used to estimate the parameters of the batch normalization layer. We use the same z^{rep} in each decoder block (see Fig. 6.1) and different linear transformations represented in the yellow boxes of Fig. 6.1. Tables 9.1 and 9.2 specify the parameters of the decoder and encoder block, respectively.

In Fig. 9.4 we consider different parameters $p(\theta^{\text{kernel}}|z)(m)$ for different kernel parameters m. In the case of the exponential quadratic kernel the parameter is defined as $\theta_i^{\text{kernel}} = (\sigma_{\theta_i}, l_{\theta_i})$ *i.e.* two sets of parameters are necessary for $p(\theta^{\text{kernel}}|z)$. In Fig. 9.4 the number of hidden units of the linear transformations used for $p(\theta^{\text{kernel}}|z)$ is BW $\times\, 4C$ (see Table 9.1), except for the last which is one. The "Activation*" block is different for different kernels, in particular $\sigma_{\theta_i} = 10 \cdot \text{Sigmoid}(\cdot)$ and $l_{\theta_i} = 1/I \cdot \text{Sigmoid}(\cdot)$.

We use the Adam optimizer [52] with its default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and an initial learning rate of $1\text{e} - 3$ for all networks and experiments.



(a) Decoder (b) Encoder

Figure 9.4: Architectures for the decoder and encoder networks, respectively.

$\tilde{z}_i^{\text{rep}} \in \mathbb{R}^J \sim \mathcal{N}(\tilde{\mu}_i, \tilde{\sigma}_i^2)$
Linear$(J) \rightarrow$ BW $\times 4C$
Resblock up $4C \rightarrow 4C$
Resblock up $4C \rightarrow 4C$
Resblock up $4C \rightarrow 4C$
x2 :Output Layer: BN, ReLU, 3×3 Conv $C \rightarrow 1$

Table 9.1: Decoder $p(x^{\text{ind}}|z)$. BW refers to a bottom width parameter, which it defines the number of induction points I. We use BW $= 3$, given that we have 3 upsampling resblocks the number of induction points is $I = 3 \times 2^3 = 24$.

$x \in \mathbb{R}^{200 \times 1 \times D}$
Interpolation prediction block $D \rightarrow 2D$
Resblock down $2D \rightarrow 4C$
Resblock down $4C \rightarrow 4C$
Resblock down $4C \rightarrow 4C$
Resblock down $4C \rightarrow 4C$
Flatten
$\times 2$: Linear $(32//2^4 \times 4C) \rightarrow J$

Table 9.2: Encoder

9.5.3 Results

In Table 9.3 we show the results of testing LC-VaDE using all the possible classes as outlier. The worst result for both Linear and ASAS datasets comes from the eclipsing binary classes: Beta Persei and W Ursae Majoris. Other classes are close to the perfect score of 100% and when Delta Scuti (in Linear dataset) is chosen as outlier the model reaches the perfect score.

Fig. 9.6 and Fig. 9.7 show reconstructed light curves for each class on the test set for the Linear and ASAS datasets, respectively (In Fig. 9.6 Beta Persei reconstructions are

Table 9.3: AUCROC and AUCPR for Linear and ASAS datasets by outlier class. Classes are ordered by number of data samples in the training set.

Outlier class	AUCROC	AUCPR
Delta Scuti	100.00 ± 0.00	100.00 ± 0.00
Beta Persei	86.55 ± 2.97	85.56 ± 3.63
RR Lyrae FO	96.41 ± 2.02	96.40 ± 1.73
W Ursae Majoris	71.37 ± 12.05	64.72 ± 10.75
RR Lyrae FM	95.72 ± 3.78	93.59 ± 7.09

(a) Linear dataset

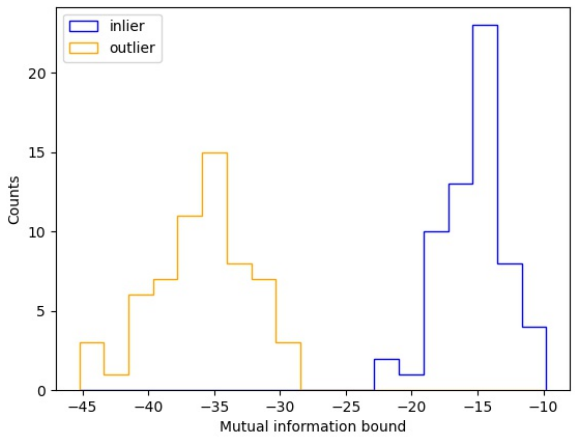
Outlier class	AUCROC	AUCPR
Classical Cepheid	98.58 ± 2.41	98.72 ± 2.07
Semireg PV	99.98 ± 0.15	99.98 ± 0.43
Beta Persei	89.73 ± 1.59	84.76 ± 0.22
RR Lyrae FM	94.78 ± 0.15	89.91 ± 0.43
W Ursae Majoris	81.62 ± 6.15	71.51 ± 4.31

(b) Asas dataset

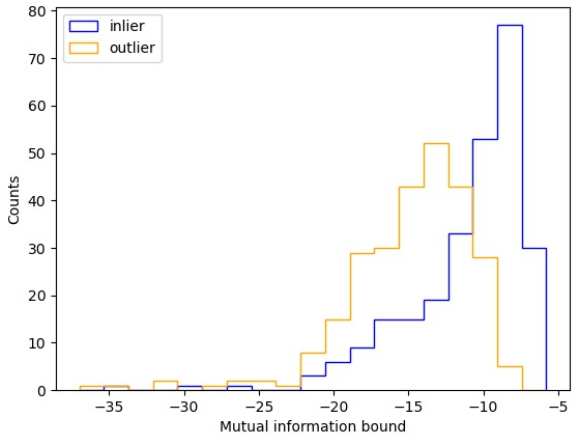
not included because of its low number on test set). For the Linear dataset we chose the case where the Delta Scuti class is taken as outlier, which has a good anomaly detection performance (see Table 9.3). For the ASAS dataset we chose the case where the W Ursae Majoris is taken as outlier, which has the worst anomaly detection performance (see Table 9.3). For the Linear dataset we found that even the outlier class is well reconstructed. On the other hand, for the ASAS dataset the outlier class has bad reconstructions and tend to reconstruct Beta Persei classes rather than the input.

Fig. 9.8 and Fig. 9.9 show reconstructed light curves on the test set ordered by their IAS for the same runs used in Fig. 9.6 and Fig. 9.7, respectively. For the Linear dataset (Fig. 9.8) we show that the most outlier data (smallest IAS) are effectively the data corresponding to the outlier class. We found that even for data with the smallest IAS its reconstructions are close to the input data. This implies that models that detect anomalous data based solely on the reconstruction error [73] could result in poor anomaly detection performance. For the ASAS dasaset (Fig. 9.9) the performance is bad. In this case the model does not reconstruct the outlier (in this case W ursae majoris) correctly case and the smallest IAS does not always correspond to the outlier class. Fig. 9.5 shows histograms of the mutual information bound (IAS) for the run in Linear and ASAS datasets. The superior performance in the Linear dataset is clearly observed by more separable histograms than in the ASAS dataset.

Table 9.4 compares our model with the most similar baseline [123] in the literature, which is based on energy function (see section 9.1). This model has been used in astronomy [109] and considers RNN architectures for the encoder and decoder (architecture proposed in [80]) to deal with variable length and irregular sampling. For completeness we compare against the original RNN architecture of [109], and also with an energy based anomaly detection model with the same architecture that is used in LC-VaDE. The latter presented very unstable training so we chose only the runs that could converge for the comparison. Finally, to show the relevance of multimodal priors we implemented an unimodal version of LC-VaDE using the same architecture of LC-VaDE. This model can be understood as a VAE model with the addition supervision in the latent space. The anomaly score in this case is computed by measuring the likelihood of the encoded data into the prior $\mathcal{N}(0, I)$.



(a) MI bound with Delta Scuti as outlier



(b) MI bound with W Ursae Majoris as outlier

Figure 9.5: Histogram of the mutual information bound for the Linear dataset using Delta Scuti as outlier (a) and for the ASAS dataset using W Ursae Majoris as outlier.

9.5.4 Discussion

We present a novel generative-inference model called LC-VaDE. With this model we surpasses the most similar baseline in the current literature, improving over the autoencoding mixture model based on energy function [123]. When we used recurrent neural networks and energy function as anomaly score as is done in [109] the results are stable and consistent, but the performance is inferior than when we use our proposed architecture. The proposed architecture improves over the RNN architecture by a significant margin. We found however difficulties in training when we used the Linear dataset. We attribute these instabilities to the need of the covariance regularization for each estimated Gaussian that was noted by [123, 109]. In contrast our proposed model LC-VaDE using the same architecture is stable and with a much better performance than the energy baseline. We test different combination for the class chosen as outlier observing consistent result. We also compare LC-VaDE with its uni-modal version *i.e.* only one Gaussian as prior resulting in a VAE with additional supervision on the embedding. We show that the uni-modal version is outperformed by a large margin which highlights the relevance of multi-modal priors.

For LC-VaDE a new decoder motivated by Gaussian processes was proposed. This decoder can be seen as an amortized Gaussian process since it can generate at any given/request time from the induction points. The decoder is particularly useful for representation learning since it forces the reconstruction of the entire light curve using only the induction points, which are considerable less dimensional than the input of the network. The decoder is suitable to deal with the difficulties associated with light curve datasets: irregular sampling and variable length data. Moreover this decoder estimates the variance (photometric error) *i.e.* the model is capable to capture the variance of the data in the latent space.

From the observations made in Section 9.5.3 we can conclude that the model have low anomaly detection performance when the outlier class is badly reconstructed. For the ASAS dataset the worst performance is obtained when W Ursae Majoris is taken as the outlier

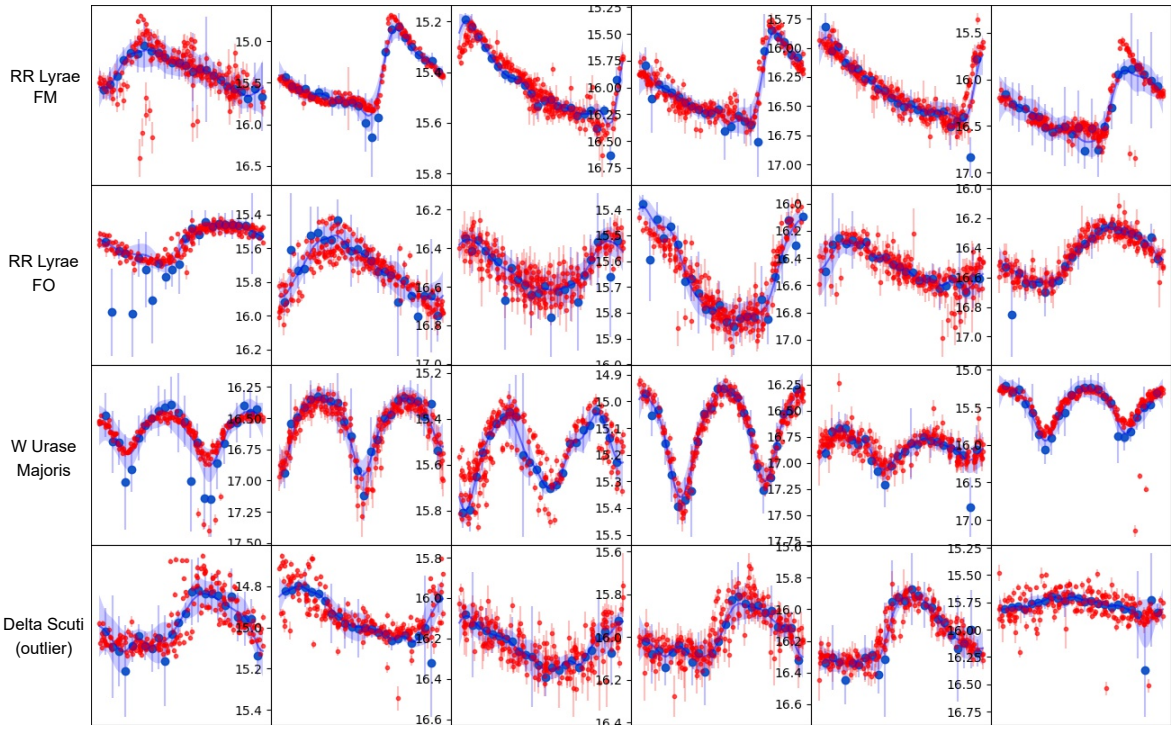


Figure 9.6: Light curve reconstructions in the Linear test set when Delta Scuti is used as the outlier class. Red dots refer to observed data, blue dots are the induction points $\hat{\mu}^{\text{ind}}$ with error bars $\hat{\sigma}^{\text{ind}}$. The blue line corresponds to the predicted data on real times t, and the shaded blue area to its standard deviation.

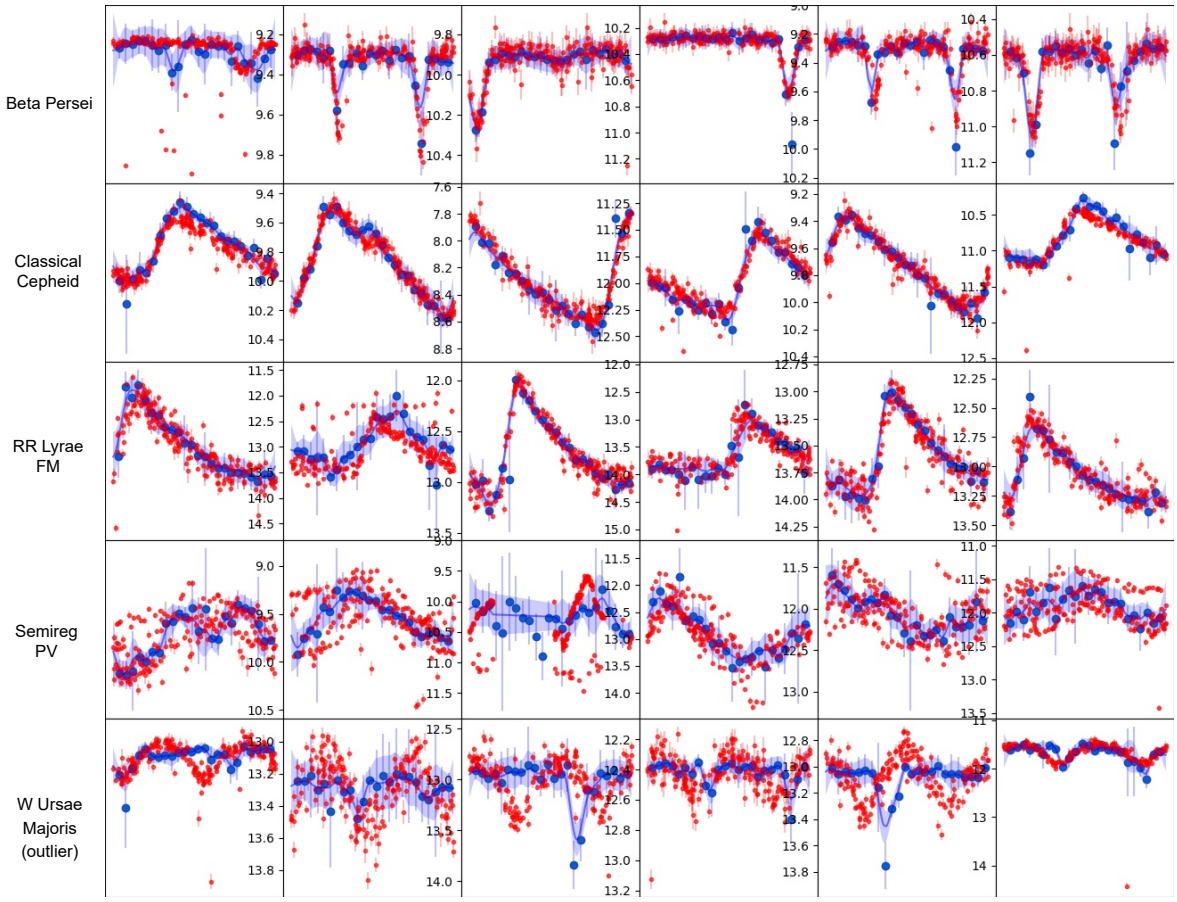


Figure 9.7: Light curve reconstructions in the ASAS test set when W Ursae Majoris is used as the outlier class. Red dots refer to observed data, blue dots are the induction points $\hat{\mu}^{\text{ind}}$ with error bars $\hat{\sigma}^{\text{ind}}$. The blue line corresponds to the predicted data on real times t, and the shaded blue area to its standard deviation.

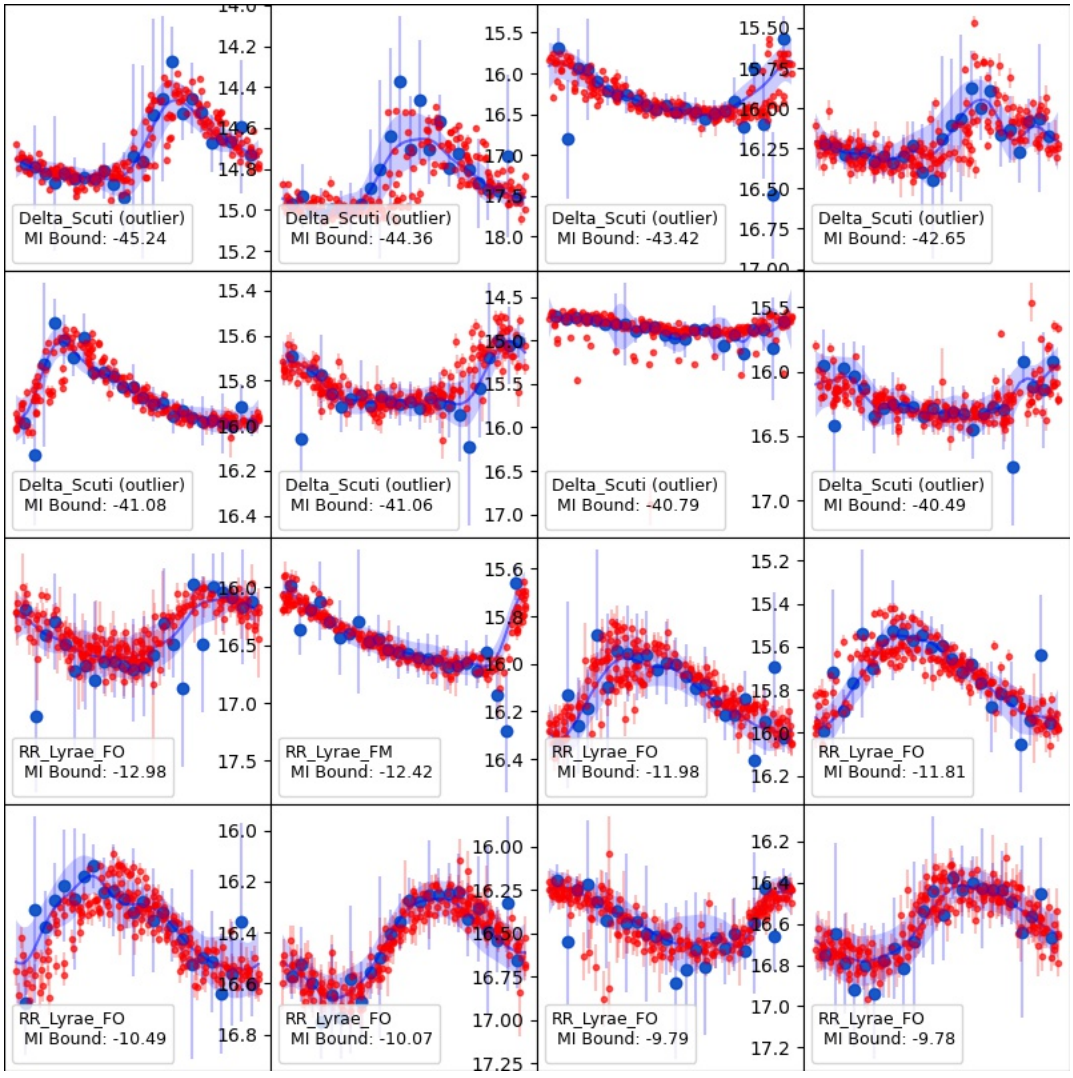


Figure 9.8: Examples sorted by the mutual information bound (IAS) for the Linear test set when Delta Scuti is selected as the outlier class. The two top/bottom rows correspond to data with the lowest/highest IAS. Red dots refer to observed data, blue dots are the induction points $\hat{\mu}^{\text{ind}}$ with error bars $\hat{\sigma}^{\text{ind}}$. The blue line corresponds to the predicted data on real times t, and the shaded blue area to its standard deviation.

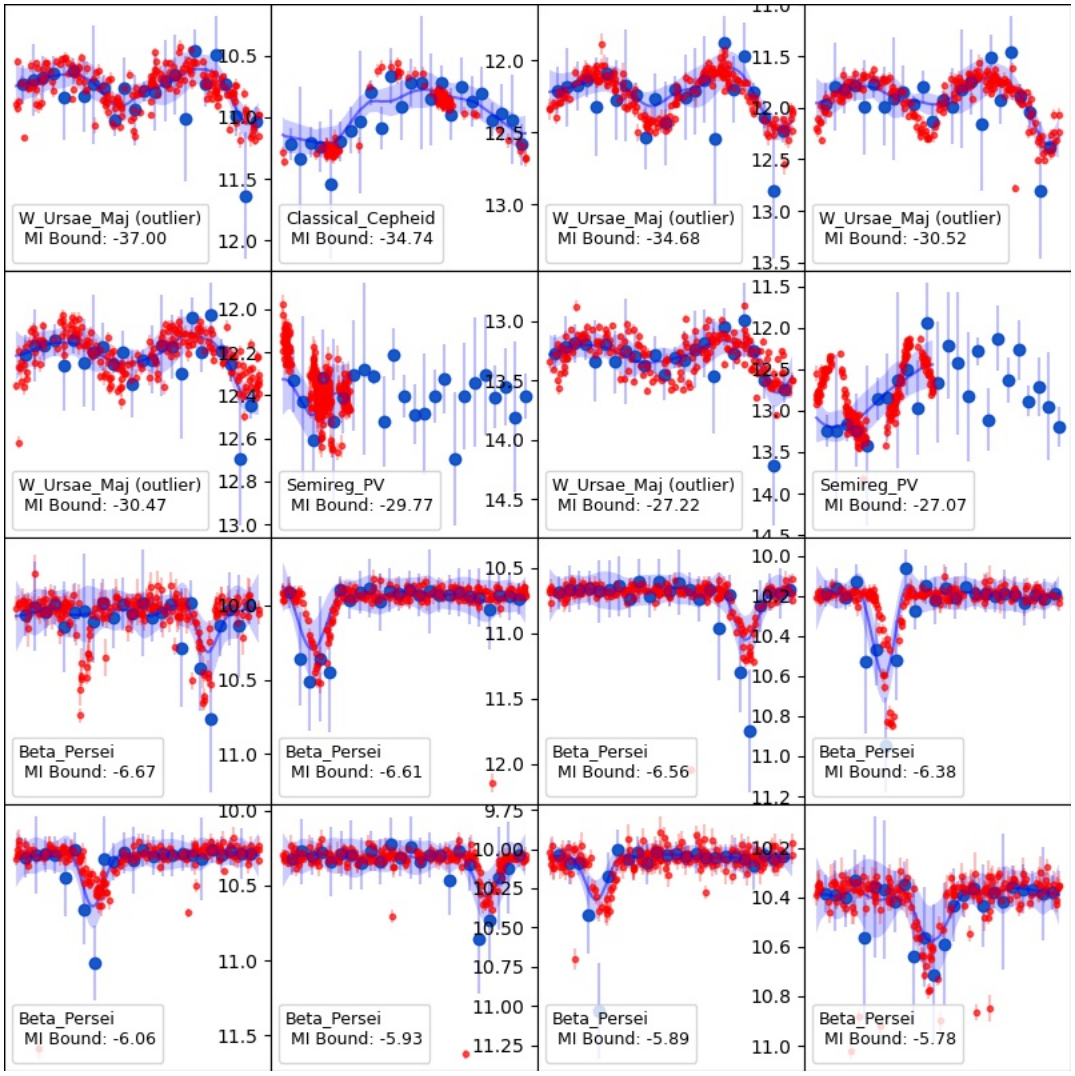


Figure 9.9: Ordered data by mutual information bound (IAS). The top 2 rows/bottom correspond to data with low/high IAS. In this case the W Ursae Majoris is left as the outlier data to be detected. Red dot point refers to real data, blue points are the induction points $\hat{\mu}^{\text{ind}}$ with error bars $\hat{\sigma}^{\text{ind}}$. Blue line corresponds to the predicted data on real times t, its standard deviation with light blue surrounding color.

Table 9.4: AUCROC and AUCPR for the Linear and ASAS dataset. The outlier class to be detected is the minority class of the correspoding dataset, i.e. Delta Scuti and Classical Cepheid for Linear and ASAS datasets, respectively. LC-VaDE statistics were obtained using four runs, and baseline models through three runs. (S) refers to the same architecture used in LC-VaDE.

	Datasets			
	Linear		ASAS	
Methods	AUCROC	AUCPR	AUCROC	AUCPR
Energy GMM (RNN) [109]	62.80 ± 0.53	62.96 ± 0.90	88.20 ± 0.06	81.97 ± 0.01
Energy GMM (S)	98.15 ± 0.79	97.89 ± 0.98	94.39 ± 0.47	89.78 ± 1.01
VAE baseline (S)	46.86 ± 6.26	48.75 ± 7.74	60.23 ± 5.63	63.12 ± 7.98
LC-VaDE (ours)	100.00 ± 0.00	100.00 ± 0.00	98.58 ± 2.41	98.72 ± 2.07

class. This is contrasted with bad reconstruction observed in Fig. 9.7. However, the best performance of Linear dataset occurs when Delta Scuti is taken as outlier (see Table 9.3) and the reconstructions of the Delta Scuti are good. This empirical fact motivates the creation of a new anomaly detection score that takes the IAS score in consideration but also the reconstruction error in the observed space.

LC-VaDE is a generative-inference model for the representation learning task in light curve astronomical data since it can use both unsupervised information and supervised information that will be key to the LSST survey. The proposed decoder forces even more the codification of a useful embedding by learning induction points that generate the entire input light curve. There are several interesting research lines to expand this model: 1) Learn the times of the induction points forcing the model to learn more relevant features, 2) Consider other kernels like the spectral kernel [113] allowing the possibility of learning the periods of the light-curves and 3) Trying more challenging datasets like the ZTF survey.

Chapter 10

Conclusions

This thesis presents a thorough study of generative-inference models, *i.e.* generative models based on neural network architectures that also consider an inference model. The inference model is useful to reach the most relevant features of the observed space $\mathcal{X}$ compressed into the latent space $\mathcal{Z}$. The study comprehends the formalization of the graphical models considered, the generalization of generative-inference model into a common mathematical framework from a information theory perspective, the empirical demonstration of the behaviour expected by this perspective, the study of generative-inference with more flexible graphical models and the proposition of new models for clustering and anomaly detection applications.

For the general case of a generative-inference model of two variables $x \in \mathcal{X}$ and $z \in \mathcal{Z}$ we propose two desired properties of generative-inference models: one associated with the graphical model and the generative capabilities of the model, and other associated with the representation learning capabilities of the model. These desired properties have not been formalized in the literature so it is a step forward to understand the generative and representation learning capabilities of generative-inference models.

With the two desired properties formalized, we study how these properties can be achieved. We found two perspectives to be useful for this task: 1) Matching the joint distribution of generative-inference models and 2) a new proposed perspective associating generative-inference model with the mutual information of their distributions. We expand the first perspective of matching joint distribution already explored in the literature including models that match the marginal distributions. We note this perspective is useful since it tell us about the optimizations that can be utilized to achieve the two desired properties that we enunciated.

Although the joint matching distribution perspective tell us about what loss functions can be used to achieve the desired properties, it doesn't tell us directly about the representation capabilities of the model. To achieve such perspective we associate generative-inference model loss functions with the mutual information of their distributions. This perspective is remarkably useful since just from identifying the loss function of the model we can have a close idea of how the model will be behave in representation learning as well as in generation.

We also proposed a new generative-inference model based on this theory.

We validate the theory presented using a state of the art architecture, comparing various generative-inference models under similar optimization conditions. In simpler datasets, when no capacity restriction exists, we note that the theory accurately predicts the empirical results in the sense of comparing various models and identifying what models should behave better in representation learning or generation. We also observed the relevance of the entropy $h_q(z|x)$ in the generative capabilities of the model and its generalization capacity. For more complex datasets we tested mainly GAN based models and we were able to conclude what model behave better.

We noted that the prior distribution $p(z)$ is fundamental for both generation and representation learning and models that have a multi-modal prior instead of an uni-modal prior like $\mathcal{N}(0, I)$ are studied. We derive, based on the two theoretical perspectives, the models that can enter in these categories. We found that models decomposed using $D_{KL}(p(x, z, y)||q(x, z, y))$ are not present in the literature, so we presented it as a new model called MPCC. We experiment extensively showing state of the art performances in clustering and generation (as of July 2020) on CIFAR10 benchmark. The results also show that multi-modal priors outperform uni-modal priors.

Finally, to validate the usability of generative-inference models in practice, we applied them in real world light curve applications for the task of anomaly detection. To deal with the variable length and irregular sampling of light curves we proposed a new decoder based on Gaussian processes. We also expand the graphical model of a multi-modal generative-inference model to make it suitable to use in light curve applications, we called this model LC-VaDE. With this model we surpass the most similar baselines and we also show the superiority of multi-modal priors in representation learning.

The theoretical and empirical results of this thesis are extensive. Because we associate generative-inference model by a general framework we were able to understand them from a broader perspective, translating the theoretical findings in practice and proposing new models not existing in the current literature. We found unsupervised metrics that can be used in practice to select generative-inference models by their representation or generative capabilities and we also show the relevance of the prior distribution for both tasks. This thesis serves as a guide for the understanding and the application of generative-inference models in representation learning and generation. The hypotheses made in this thesis were helpful to guide this research, understand generative-inference models and experiment with them.

10.1 Future work

In this thesis we studied the general formulation of generative inference models with two variables $x \in \mathcal{X}$, $z \in \mathcal{Z}$ and three variables $x \in \mathcal{X}$, $z \in \mathcal{Z}$, $y \in \mathcal{Y}$, respectively. In the literature the two variable graphical model has been extended for more latent variables $z_1, z_2 \dots z_k$, particularly for VAE models in [15, 73] to improve convergence in more complex datasets. The likelihood is estimated hierarchically and although these architectures differs greatly from GAN architectures, this way to estimate likelihood could be used for models that bound

$\mathcal{I}_p(x, z)$, which hasn't been explored in the literature. We also plan to explore more deeply the models obtained by decomposing $D_{KL}(p(x, z)||q(x, z))$ or $D_{KL}(p(x, z, y)||q(x, z, y))$. The current models that optimize these decompositions are AIM [69] and MPCC (Chapter 7), respectively. These models optimize $D_{KL}(p(x)||q(x))$ adversarially instead of $D_{KL}(p(x|z)||q(x))$ (see Chapters 4 and 7). We hypothesize that optimizing $D_{KL}(p(x|z)||q(x))$ should improve the generation capabilities of the model despite worse representation learning capabilities (in MPCC this disadvantage may not be present given its multi modal prior).

In this thesis we maximize the correlation of observed variables $x \in \mathcal{X}$ and latent variables $z \in \mathcal{Z}$ by maximizing the likelihood, *i.e.* reducing the reconstruction error. Recent approaches have tackled this approach using contrastive learning techniques that can also be associated with a maximization of a bound of the mutual information [88]. These techniques achieve state of the art results in complex datasets [12, 13] and could be useful to regularize the discriminator or the encoder resulting in better representation learning and/or generative capabilities of the model.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[2] Nicolás Astorga, Pablo Huijse, Pavlos Protopapas, and Pablo Estévez. Mpcc: Matching prior and conditionals for clustering. In *European Conference on Computer Vision*, 2020.

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 15535–15545. Curran Associates, Inc., 2019.

[4] Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. *ArXiv*, abs/1808.10356, 2018.

[5] Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. *CoRR*, abs/1808.10356, 2018.

[6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors,

Advances in Neural Information Processing Systems, volume 33, pages 9912–9924. Curran Associates, Inc., 2020.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[11] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, Oct 2017.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[13] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[14] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[15] Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021.

[16] Frédéric Chyzak and Frank Nielsen. A closed-form formula for the Kullback-Leibler divergence between Cauchy distributions. 8 pages, December 2019.

[17] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6510–6520. Curran Associates, Inc., 2017.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[19] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[20] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *ArXiv*, abs/1907.02544, 2019.

[21] Hao-Wen Dong and Yi-Hsuan Yang. Towards a deeper understanding of adversarial losses. *ArXiv*, abs/1901.08753, 2019.

[22] G. Dorta, Sara Vicente, L. Agapito, N. Campbell, and Ivor J. A. Simpson. Training

vaes under structured residuals. *ArXiv*, abs/1804.01050, 2018.

[23] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[24] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *ICLR*, 2017.

[25] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 441–452. Curran Associates, Inc., 2018.

[26] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[27] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

[28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[30] Guillermo L. Grinblat, Lucas C. Uzal, and Pablo M. Granitto. Class-splitting generative adversarial networks, 2017.

[31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5769–5779, USA, 2017. Curran Associates Inc.

[32] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R. Venkatesh Babu. Deligan : Generative adversarial networks for diverse and limited data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[33] Philip Häusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *GCPR*, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

770–778, June 2016.

[35] Olivier J Henaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding, 2020.

[36] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017.

[37] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[38] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[40] Zafiirah Hosenie, Robert Lyon, Benjamin Stappers, and Arrykrishna Mootoovaloo. Imbalance learning for variable star classification. *Royal Astronomical Society. Monthly Notices*, February 2020.

[41] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1558–1567. JMLR.org, 2017.

[42] Pablo Huijse, Pablo A. Estevez, Pavlos Protopapas, Jose C. Principe, and Pablo Zegers. Computational intelligence challenges and applications on large-scale astronomical time series databases. *IEEE Computational Intelligence Magazine*, 9(3):27–39, 2014.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[44] Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2235–2244, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[45] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for

unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

[46] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1965–1972. AAAI Press, 2017.

[47] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc., 2016.

[48] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[49] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[50] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1857–1865. PMLR, 06–11 Aug 2017.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[53] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[54] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014.

[55] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[56] C. S. Kochanek, B. J. Shappee, K. Z. Stanek, T. W.-S. Holoien, Todd A. Thompson, J. L. Prieto, Subo Dong, J. V. Shields, D. Will, C. Britt, and et al. The all-sky automated survey for supernovae (asas-sn) light curve server v1.0. *Publications of the Astronomical Society of the Pacific*, 129(980):104502, Aug 2017.

[57] Adam Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 15512–15522. Curran Associates, Inc., 2019.

[58] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[60] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[61] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[62] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[63] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[64] Yann LeCun, Sumit Chopra, Raia Hadsell, Fu Jie Huang, and et al. A tutorial on energy-based learning. In *PREDICTING STRUCTURED DATA*. MIT Press, 2006.

[65] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 4–15, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[66] Chongxuan LI, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4088–4098. Curran Associates, Inc., 2017.

[67] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. Mmd gan: Towards deeper understanding of moment matching network. In I. Guyon,

U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[68] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pages 5495–5503, 2017.

[69] Hanbo Li, Yaqing Wang, Changyou Chen, and Jing Gao. AIM: Adversarial inference by matching priors and conditionals, 2019.

[70] Jae Hyun Lim and Jong Chul Ye. Geometric gan, 2017.

[71] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[72] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[73] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling, 2019.

[74] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1445–1453, New York, New York, USA, 20–22 Jun 2016. PMLR.

[75] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[76] Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York, 1988.

[77] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2391–2400, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[78] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[79] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4610–4617. AAAI Press, 2019.

[80] Brett Naul, Joshua S. Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2:151–155, November 2018.

[81] Tim Naylor. An optimal extraction algorithm for imaging photometry. *Monthly Notices of the Royal Astronomical Society*, 296(2):339–346, May 1998.

[82] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

[83] Yao Ni, Dandan Song, Xi Zhang, Hao Wu, and Lejian Liao. Cagan: Consistent adversarial training enhanced gans. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2588–2594. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[84] F. Nielsen and R. Nock. Entropies and cross-entropies of exponential families. In *2010 IEEE International Conference on Image Processing*, pages 3621–3624, Sep. 2010.

[85] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 271–279. Curran Associates, Inc., 2016.

[86] Lovro Palaversa, Željko Ivezić, Laurent Eyer, Domagoj Ruždjak, Davor Sudar, Mario Galin, Andrea Kroflin, Martina Mesarić, Petra Munk, Dijana Vrbanec, Hrvoje Božić, Sarah Loebman, Branimir Sesar, Lorenzo Rimoldini, Nicholas Hunt-Walker, Jacob VanderPlas, David Westman, J. Scott Stuart, Andrew C. Becker, Gregor Srdoč, Przemyslaw Wozniak, and Hakeem Oluseyi. Exploring the variable sky with LINEAR. III. classification of periodic light curves. *The Astronomical Journal*, 146(4):101, sep 2013.

[87] John R Percy. *Understanding variable stars*. Cambridge University Press, 2007.

[88] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[89] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[90] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[91] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[92] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[93] Douglas Reynolds. *Gaussian Mixture Models*, pages 659–663. Springer US, Boston, MA, 2009.

[94] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[95] Eitan Richardson and Yair Weiss. On gans and gmms. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5847–5858. Curran Associates, Inc., 2018.

[96] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *CoRR*, abs/1802.06847, 2018.

[97] Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 460–468. Curran Associates, Inc., 2016.

[98] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[99] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[100] Satya Narayan Shukla and Benjamin Marlin. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2019.

[101] Joram Soch and Carsten Allefeld. Kullback-leibler divergence for the normal-gamma distribution. *arXiv preprint arXiv:1611.01437*, 2016.

[102] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2019.

[103] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[104] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[105] Yaling Tao, Kentaro Takagi, and Kouta Nakata. Rdec: Integrating regularization into deep embedded clustering for imbalanced datasets. In Jun Zhu and Ichiro Takeuchi, editors, *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 49–64. PMLR, 14–16 Nov 2018.

[106] Željko Ivezić *et al.* LSST: From science drivers to reference design and anticipated data products. *The Astrophysical Journal*, 873(2):111, mar 2019.

[107] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

[108] Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference, 2017.

[109] Benny T.-H. Tsang and William C. Schultz. Deep neural network classifier for variable stars with novelty detection capability. *The Astrophysical Journal*, 877(2):L14, may 2019.

[110] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4790–4798. Curran Associates, Inc., 2016.

[111] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5756–5766. Curran Associates, Inc., 2017.

[112] L. N. Wasserstein. Markov processes over denumerable products of spaces describing large systems of automata. In *Probl. Inform. Transmission*, 1969.

[113] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation, 2013.

[114] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy P. Lillicrap. LOGAN: latent optimisation for generative adversarial networks. *CoRR*, abs/1912.00953, 2019.

[115] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[116] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 478–487, 2016.

[117] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[118] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[119] Yang Yu and Wen-Ji Zhou. Mixture of gans for clustering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3047–3053. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[120] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017.

[121] Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information autoencoding family: A lagrangian perspective on latent variable generative models. *arXiv preprint arXiv:1806.06514*, 2018.

[122] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[123] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

Appendix A

Demonstration of KL divergence between $q(z)$ and $p(z)$ (Chapter 6)

We look to obtain a closed form solution of $D_{KL}(q(z)||p(z))$. The demonstration start by taking the following generic Gaussian distributions as $p(x) = (2\pi)^{-\frac{D}{2}}|\Sigma_1|^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2}(x-\mu_1)^T\Sigma_1(x-\mu_1)}$ and $q(x) = (2\pi)^{-\frac{D}{2}}|\Sigma_2|^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2}(x-\mu_2)^T\Sigma_2(x-\mu_2)}$.

$$
\begin{aligned}
& D_{KL}(p(x)||q(x)) && (1)\\
&= \int \left[\log p(x) - \log q(x)\right] p(x)dx && (2)\\
&= \int \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - (x-\mu_1)\Sigma_1^{-1}(x-\mu_1)^T \\
&\qquad + (x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\Bigg] p(x)dx && (3)\\
&= \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - \mathrm{Tr}\{\mathbb{E}[(x-\mu_1)(x-\mu_1)^T]\Sigma_1^{-1}\} \\
&\qquad + \mathbb{E}[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)]\Bigg] && (4)\\
&= \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - \mathrm{Tr}\{I_d\} + (\mu_1-\mu_2)\Sigma_2^{-1}(\mu_1-\mu_2)^T \\
&\qquad + \mathrm{Tr}\{\Sigma_2^{-1}\Sigma_1\}\Bigg] && (5)\\
&= \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - D + (\mu_1-\mu_2)\Sigma_2^{-1}(\mu_1-\mu_2)^T \\
&\qquad + \mathrm{Tr}\{\Sigma_2^{-1}\Sigma_1\}\Bigg] && (6)
\end{aligned}
$$

Appendix B

Additional empirical results (Chapter 6)

For the completeness of our results we also test different generative-inference models for FMNIST dataset. In Table B.1 and in Table B.2 we observed the main metrics studied in Chapter 6 for latent dimensions $J = 10$ and $J = 20$ respectively. In Fig. B.1 and Fig. B.2 we observe similar tendencies than the observed in Chapter 6 although more generative-inference models have $\mathcal{I}_q(x, z) < 0$ *i.e.* the suppositions to compute this metric start to fail. Finally to show that in this dataset the reconstructions also start to fail we show them for VAE, ALI and AIM models in Fig. B.3, Fig. B.4 and Fig. B.5 respectively.

Model	**LL in $\mathcal{X}$**	LL in $\mathcal{Z}$	$D_{KL}^{\mathcal{Z}}$	D_{KL}^{VAE}	$\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$	$h_q(z\|x)$	IS	FID	Acc%
VAE ($\beta = 0.33$)	-4.27	-3.22	0.15	24.43	24.33	-10.36	7.25	203.14	76.00
	± 0.02	± 0.03	± 0.07	± 0.08	± 0.09	± 0.10	± 0.04	± 6.24	± 0.42
VAE ($\beta = 1.00$)	-4.08	-2.16	3.79	16.44	12.93	-2.55	7.76	173.01	71.89
	± 0.01	± 0.02	± 1.57	± 0.13	± 1.58	± 0.04	± 0.02	± 8.01	± 0.92
VAE ($\beta = 30.00$)	-2.60	-0.38	31.16	2.61	-28.55	11.58	5.36	869.86	62.24
	± 0.00	± 0.00	± 0.12	± 0.02	± 0.12	± 0.02	± 0.06	± 17.77	± 0.39
WVAE ($\beta = 0.33$)	-4.32	-2.73	0.10	61.31	61.33	-47.33	6.48	298.05	76.90
	± 0.01	± 0.07	± 0.05	± 0.18	± 0.18	± 0.27	± 0.03	± 11.62	± 0.23
WVAE ($\beta = 1.00$)	-4.33	-2.77	0.22	61.29	61.33	-47.57	6.55	290.79	77.10
	± 0.01	± 0.10	± 0.00	± 0.23	± 0.22	± 0.22	± 0.01	± 3.63	± 0.60
WVAE ($\beta = 30.00$)	-4.27	-2.31	0.14	62.36	62.37	-48.51	6.07	349.08	76.37
	± 0.01	± 0.08	± 0.05	± 0.28	± 0.29	± 0.27	± 0.06	± 9.70	± 0.39
MMD ($\beta = 0.33$)	-4.35	-1.27	1.98	83.03	64.72	-43.09	3.83	972.40	78.03
	± 0.00	± 0.13	± 0.01	± 0.25	± 0.66	± 0.76	± 0.22	± 53.15	± 0.58
MMD ($\beta = 1.00$)	-4.36	-1.74	1.74	73.95	63.73	-43.87	4.42	819.98	77.84
	± 0.01	± 0.19	± 0.03	± 2.14	± 1.27	± 1.33	± 0.17	± 72.13	± 0.33
MMD ($\beta = 30.00$)	-4.33	-3.04	0.45	62.38	61.85	-47.97	6.08	353.25	77.87
	± 0.01	± 0.02	± 0.02	± 0.57	± 0.56	± 0.61	± 0.04	± 6.97	± 0.59
JS-VAE ($\beta = 0.33$)	-4.31	-2.55	0.11	61.20	61.22	-46.76	6.32	334.88	77.39
	± 0.00	± 0.12	± 0.04	± 0.40	± 0.40	± 0.42	± 0.02	± 3.74	± 0.33
JS-VAE ($\beta = 1.00$)	-4.29	-2.40	0.02	60.86	60.92	-46.69	6.39	328.56	76.69
	± 0.03	± 0.23	± 0.02	± 0.66	± 0.67	± 0.75	± 0.13	± 30.39	± 0.28
JS-VAE ($\beta = 30.00$)	-4.14	-2.02	0.03	60.99	61.01	-46.69	6.02	386.39	74.91
	± 0.04	± 0.21	± 0.02	± 0.79	± 0.79	± 0.83	± 0.02	± 10.68	± 0.59
AIM	-2.85	-2.26	2.44	9.12	7.76	3.97	8.80	18.67	71.46
	± 0.01	± 0.00	± 0.02	± 0.30	± 0.35	± 0.34	± 0.02	± 1.91	± 0.99
ALI	-1.56	-0.18	35.56	17.49	-17.75	-3.64	7.56	174.40	47.82
	± 0.63	± 0.07	± 48.35	± 12.55	± 60.91	± 12.71	± 1.54	± 197.57	± 26.54
Veegan	-1.58	-1.72	2.31	6.89	4.61	7.26	6.14	393.04	62.74
	± 0.59	± 0.62	± 0.58	± 1.20	± 1.22	± 0.65	± 1.81	± 253.67	± 5.46
ALICE	-3.54	-1.20	11.11	23.39	12.29	-9.20	7.86	239.66	66.51
	± 0.01	± 0.07	± 1.71	± 0.43	± 1.45	± 0.69	± 0.12	± 22.85	± 0.41

Table B.1: Relevant metrics for the generative-inference models considered in FMNIST dataset with $J = 10$. LL in $\mathcal{X}$ and LL in $\mathcal{Z}$ refer to the MSE in the observed and latent space respectively. $D_{KL}^{\mathcal{Z}} = D_{KL}(q(z)||p(z))$ is computed in closed form assuming $q(z)$ as multivariate Gaussian. $D_{KL}^{\text{VAE}} = D_{KL}(q(z|x)||p(z))$. In **bold** we represent the main unsupervised learning metrics of our study; $\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$ measures the MI of the inference model q and **LL in $\mathcal{X}$** measures the likelihood in $\mathcal{X}$.

Model	**LL in $\mathcal{X}$**	LL in $\mathcal{Z}$	$D_{KL}^{\mathcal{Z}}$	D_{KL}^{VAE}	$\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$	$h_q(z\|x)$	IS	FID	Acc%
VAE ($\beta = 0.33$)	-4.56	-1.44	12.68	29.62	17.65	-1.95	7.19	184.96	77.20
	± 0.01	± 0.01	± 1.65	± 0.17	± 1.79	± 0.19	± 0.03	± 4.25	± 0.38
VAE ($\beta = 1.00$)	-4.15	-0.73	41.89	16.95	-24.58	11.07	7.78	169.36	72.78
	± 0.00	± 0.03	± 2.61	± 0.18	± 2.85	± 0.26	± 0.03	± 3.00	± 0.88
VAE ($\beta = 30.00$)	-2.59	-0.17	83.45	2.62	-80.80	25.72	5.43	857.41	63.78
	± 0.00	± 0.00	± 0.21	± 0.02	± 0.23	± 0.02	± 0.02	± 6.03	± 0.40
WVAE ($\beta = 0.33$)	-4.81	-2.10	1.59	113.24	113.20	-86.47	5.99	356.97	81.02
	± 0.02	± 0.05	± 0.20	± 0.23	± 0.25	± 0.07	± 0.02	± 4.22	± 0.33
WVAE ($\beta = 1.00$)	-4.82	-2.02	1.39	113.13	113.12	-86.21	5.91	377.29	80.81
	± 0.01	± 0.00	± 0.12	± 0.22	± 0.25	± 0.14	± 0.01	± 4.13	± 0.21
WVAE ($\beta = 30.00$)	-4.58	-1.43	1.02	124.49	124.44	-97.21	5.46	470.43	77.51
	± 0.02	± 0.09	± 0.13	± 0.52	± 0.52	± 0.52	± 0.07	± 12.97	± 0.45
MMD ($\beta = 0.33$)	-4.82	-1.04	2.05	140.96	119.69	-82.93	3.90	936.09	81.26
	± 0.03	± 0.05	± 0.04	± 2.32	± 0.88	± 0.96	± 0.16	± 47.45	± 0.21
MMD ($\beta = 1.00$)	-4.82	-1.40	1.84	131.97	118.23	-84.00	4.12	839.27	81.03
	± 0.02	± 0.07	± 0.04	± 2.41	± 1.21	± 0.96	± 0.15	± 30.59	± 0.10
MMD ($\beta = 30.00$)	-4.80	-2.44	2.85	116.99	113.44	-87.91	4.88	597.30	81.05
	± 0.02	± 0.02	± 0.27	± 0.74	± 0.85	± 0.77	± 0.05	± 6.16	± 0.06
JS-VAE ($\beta = 0.33$)	-4.79	-1.91	0.07	112.78	112.89	-84.33	5.86	403.76	80.71
	± 0.01	± 0.02	± 0.02	± 0.76	± 0.77	± 0.73	± 0.10	± 12.88	± 0.07
JS-VAE ($\beta = 1.00$)	-4.74	-1.77	0.05	112.14	112.26	-83.74	5.85	402.94	80.51
	± 0.03	± 0.08	± 0.02	± 0.90	± 0.89	± 0.94	± 0.07	± 7.28	± 0.24
JS-VAE ($\beta = 30.00$)	-4.15	-0.64	0.35	97.26	97.24	-68.02	5.48	497.19	75.86
	± 0.09	± 0.19	± 0.01	± 3.00	± 3.03	± 3.00	± 0.08	± 4.86	± 0.32
AIM	-2.89	-0.87	13.30	6.79	-5.40	20.48	8.72	22.03	75.70
	± 0.03	± 0.01	± 0.57	± 0.17	± 0.74	± 0.18	± 0.04	± 2.77	± 0.67
ALI	-1.95	0.25	5.20	45.23	40.17	-16.98	8.71	31.85	77.54
	± 0.05	± 0.07	± 0.89	± 6.13	± 6.48	± 5.62	± 0.01	± 0.66	± 0.67
Veegan	-2.16	-0.72	11.26	7.96	-2.96	20.08	7.70	123.05	71.00
	± 0.57	± 0.40	± 0.77	± 1.21	± 0.17	± 0.63	± 1.35	± 127.64	± 2.85
ALICE	-3.52	-0.44	40.37	19.01	-21.42	9.43	8.09	175.50	70.09
	± 0.04	± 0.04	± 1.51	± 4.34	± 4.20	± 4.49	± 0.06	± 35.16	± 0.31

Table B.2: Relevant metrics for the generative-inference models considered in FMNIST dataset with $J = 20$. LL in $\mathcal{X}$ and LL in $\mathcal{Z}$ refer to the MSE in the observed and latent space respectively. $D_{KL}^{\mathcal{Z}} = D_{KL}(q(z)||p(z))$ is computed in closed form assuming $q(z)$ as multivariate Gaussian. $D_{KL}^{\text{VAE}} = D_{KL}(q(z|x)||p(z))$. In **bold** we represent the main unsupervised learning metrics of our study; $\tilde{\boldsymbol{\mathcal{I}}}_{\boldsymbol{q}}(\boldsymbol{x},\boldsymbol{z})$ measures the MI of the inference model q and **LL in $\mathcal{X}$** measures the likelihood in $\mathcal{X}$.

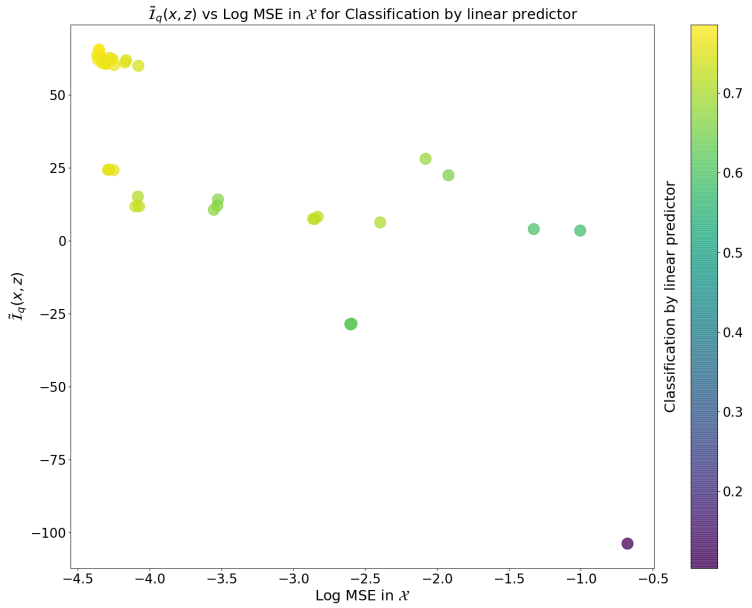


Figure B.1: Tendency of all generative-inference models considered using $\tilde{I}_q(x,z)$ vs Log MSE in the observed space for classification accuracy by linear predictor on the latent space in FMNIST with $J = 10$.

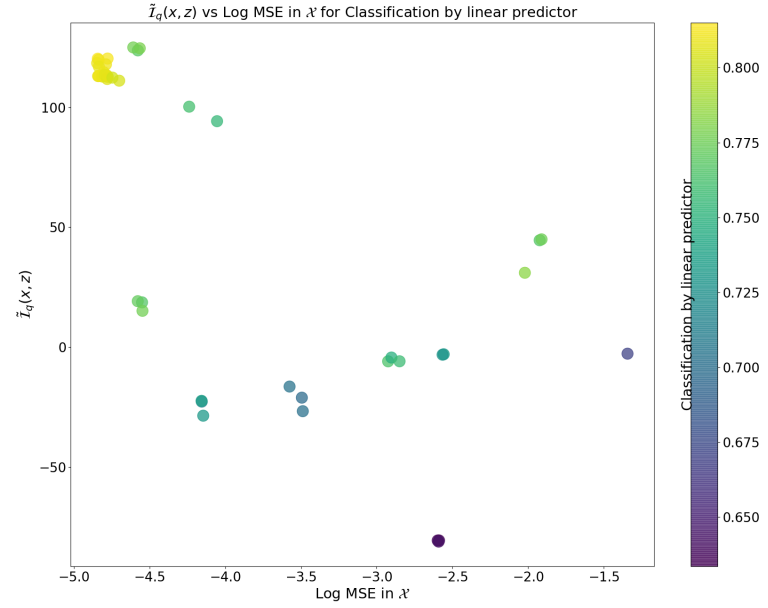


Figure B.2: Tendency of all generative-inference models considered using $\tilde{I}_q(x,z)$ vs Log MSE in the observed space for classification accuracy by linear predictor on the latent space in FMNIST with $J = 20$.

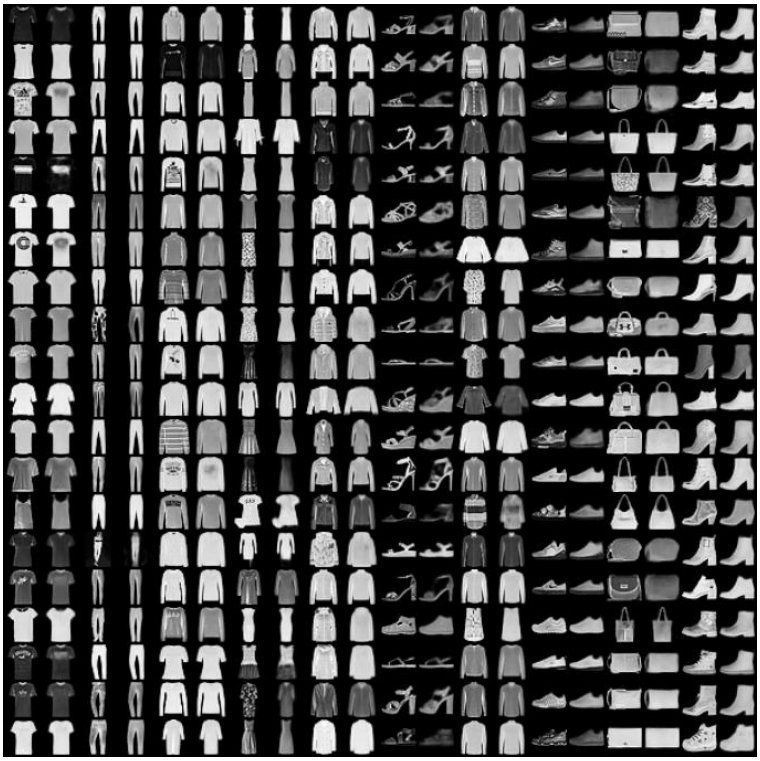


Figure B.3: Reconstructions for VAE model by classes. Odd columns represent real data and even columns correspond to their reconstructions.

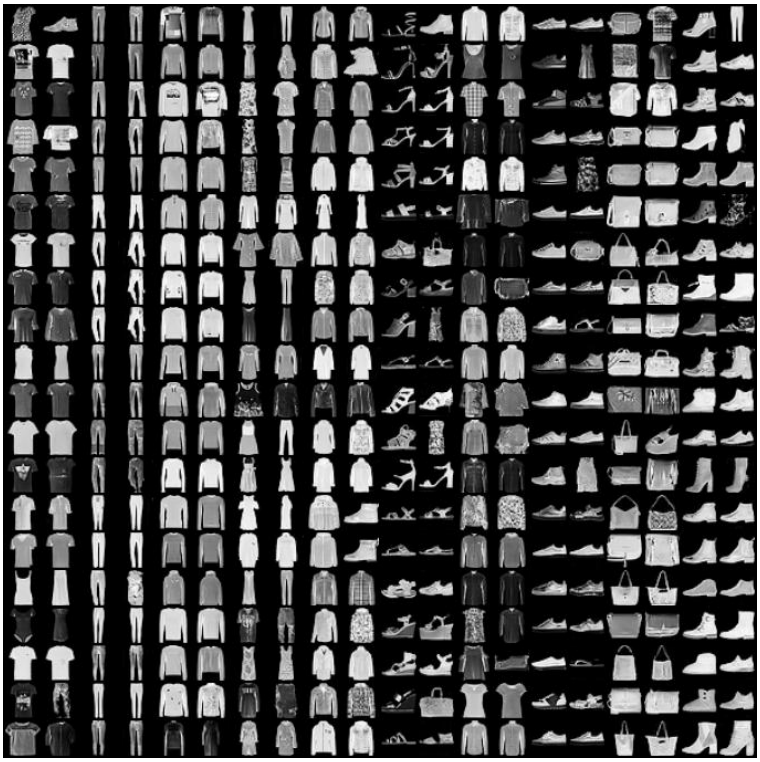


Figure B.4: Reconstructions for ALI model by classes in FMNIST. Odd columns represent real data and even columns correspond to their reconstructions.

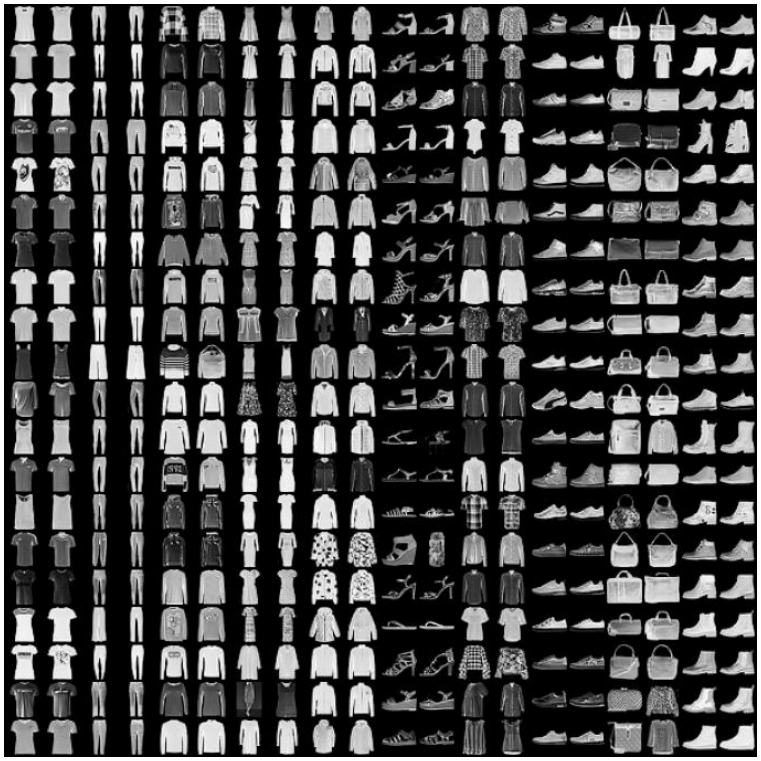


Figure B.5: Reconstructions for AIM model by classes. Odd columns represent real data and even columns correspond to their reconstructions.

Appendix C

Additional observation of Chapter 8: Optimization problems MPCC

We observed two types of errors which restrict the architecture and the optimization techniques. Theses difficulties are particularly relevant for the CIFAR10 and CIFAR20 datasets which present the more complex distributions. We used the default parameters of the CIFAR10 architecture unless otherwise stated.

The first problem is associated with the batch size. We found that we can't optimize MPCC with a big batch size while using a large learning rate of the prior parameter η_p. Note that the latter is necessary to obtain good accuracy performance as it was shown in the thesis (Table 8.5 from Chapter 8). The batch size is relevant to increase the IS and FID scores [8]. Artifacts or saturation problems would appear when doing a small modification in the optimization. The examples shown in Fig. C.6 (a) use a batch size slightly larger than the one used in the thesis (Chapter 8) of 50. We observe that using a slightly larger batch size (64) with a prior learning rate of $\eta_p = 8 \cdot 10^4$ the results change drastically and the generated images show notable saturation.

Mode collapse is an important topic in GANs research and is the second problem that we observed in MPCC. Usually it is associated with the limitations in generation quality caused by the model, which memorize only a small part of the real distribution affecting the performance of the GAN. In MPCC the mode collapse problem can make an entire cluster collapse. Setting $D_{step} = 4$ solves this problem partially for a large amount of models and is sufficient to obtain good performance. In Fig. C.6 (b) we show samples from a model trained with $bs = 64$ and $\eta_p = 2 \cdot 10^{-4}$ where we can see how a mode collapse problem looks in MPCC. We observed that when using a large prior learning rate $\eta_p = 6 \cdot 10^{-4}$ this problem would regularly appear after 150,000 iterations. This doesn't occur for the best configuration of MPCC (the one reported in the thesis) and setting $\eta_p = 2 \cdot 10^{-4}$ even after a large number of iterations, however we would like to increase η_p further as we observed that it correlates with better clustering accuracy (Table 8.5 in the thesis).

(a) Saturation problems (b) Mode collapse problems

Figure C.6: Generated images with bad optimization setting at iteration 50000. Sub-figure (a) shows images associated with saturation problems and (b) with mode collapse problems. Each row represents a different cluster.

Appendix D

Additional qualitative results of MPCC (Chapter 8)

In this section we provide additional reconstructions and samples of MPCC for the CIFAR-10 dataset in Figures D.7 and D.8, and for the MNIST dataset in D.9 and D.10. To give more insight about MPCC's capacity we also include samples for datasets with a high number of classes, CIFAR-20 and Omniglot in Figures D.11 and D.12 respectively.

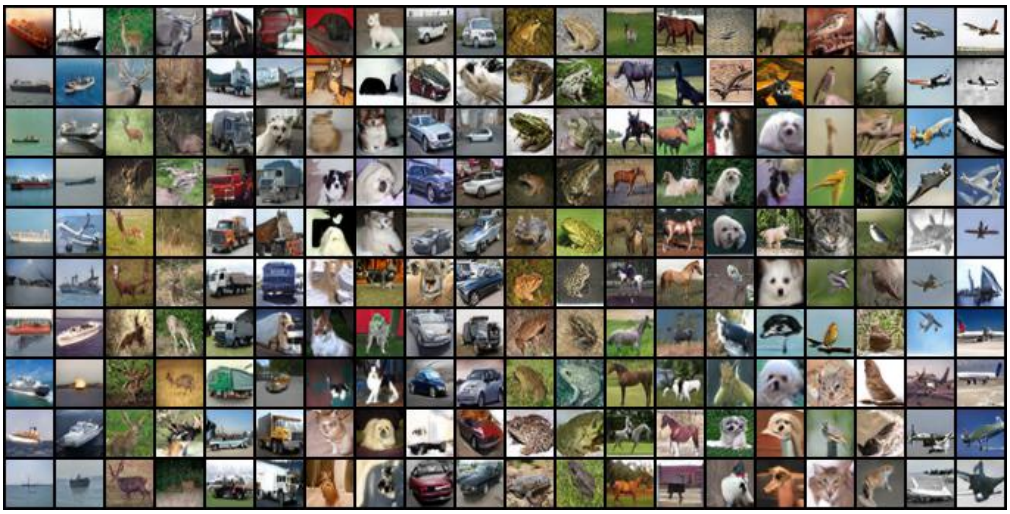


Figure D.7: Generated images for the CIFAR-10 dataset. Every two columns we set a different value for the categorical latent variable y. *i.e.* the samples shown correspond to a different conditional latent space $z \sim p(z|y)$.

Figure D.8: Reconstructions for the CIFAR-10 dataset. Odd columns represent real data and even columns correspond to their reconstructions. The real label is used to sort the column pairs.

Figure D.9: Generated images for the MNIST dataset. Every two columns we set a different value for the categorical latent variable y. *i.e.* the samples shown correspond to a different conditional latent space $z \sim p(z|y)$.

Figure D.10: Reconstructions for the MNIST dataset. Odd columns represent real data and even columns correspond to their reconstructions. The real label is used to sort the column pairs.

Figure D.11: Generated images for CIFAR-20 dataset. In every row we set a different value for the categorical latent variable y, *i.e.* the samples shown correspond to a different conditional latent space $z \sim p(z|y)$.

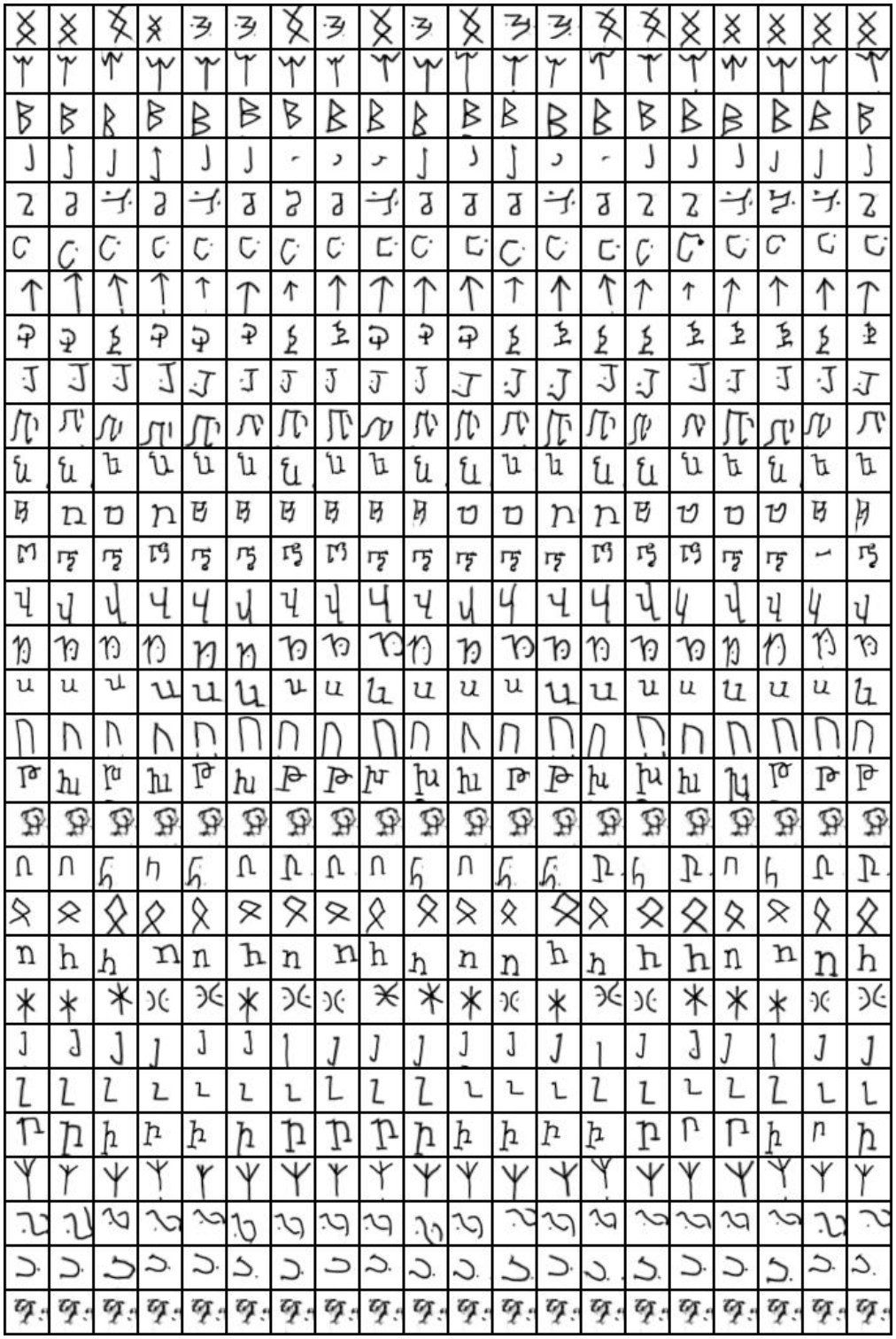


Figure D.12: Generated images for Omniglot dataset. In every row we set a different value for the categorical latent variable y, *i.e.* the samples shown correspond to a different conditional latent space $z \sim p(z|y)$. 30 cluster were randomly chosen.