Symmetric Variational Inference with High Mutual Information

by

Micha Livne

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Symmetric Variational Inference with High Mutual Information

Micha Livne

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2020

This thesis introduces the Mutual Information Machine (MIM), an autoencoder model for learning joint distributions over observations and latent states. The model formulation reflects three key design principles: 1) low divergence, or symmetry, to encourage the encoder and decoder to learn consistent factorizations of the same underlying distribution; 2) high mutual information, or approximate invertibility, to encourage an informative relation between data and latent variables; and 3) low marginal entropy, or compression, which tends to encourage clustered latent representations. Taken together, these objectives yield a cross entropy loss for learning latent variable models. The resulting form of amortized, symmetric variational inference stands in contrast to the use of an evidence-lower-bound (ELBO) in VAEs, and the use of adversarial learning that is common with other models formulated in terms of a symmetric divergence. In this thesis we systematically probe different terms in the variational bound, providing intuition about MIM. Experiments show that MIM is capable of learning a latent representation with high mutual information, and good unsupervised clustering, while providing data log likelihoods comparable to VAE. We demonstrate state of the art results on image and language data.

Mutual Information Machine

You come to me at night
Reaching out into a tormented soul
I hear you calling to me
Smelling you
Tasting you
Feeling you

I reach my hand deep into a conflicted mind
Pulling you closer
Or maybe being pulled
When all the world is ablaze around me
I let go of it all

But you keep pulling harder
You come to me through dreams
Force me back to reach my hand
I keep pulling
Or maybe being pulled

Your perfection mesmerizes me
I hear you roar
Bursting into a violent existence
Resilient and proud
Ignored
Misunderstood

I am broken now
I finally understand
We are not that different after all
Looking at you through cracked reality
Trying to make sense of it all

I am just like you
Resilient and proud
Ignored
Misunderstood
A mutual information machine

Acknowledgements

First and foremost I would like to thank my supervisor David Fleet for having the patience to support my academic wandering to the point of being lost, and for providing me the guidance in finding my way back. Furthermore, via financial support, non-compromising work ethics, insightful feedback, thought provoking questions, and remarkable communication skills, David guided me through the transformation from a student into a world expert in the topic of my choice, and I am grateful for that.

I would also like to thank my supervisory committee members, Allan Jepson, and Marcus Brubaker, for assisting me in the process of graduation despite the short timeline, and to my external committee members Kyunghyun Cho, and David Duvenaud. Special thanks go to Kevin Swersky, which provided me with academic and personal support, and assisted me in facilitating financial support at a time of great need. Additional thanks to all my colleagues, collaborators, and friends. Your support, good advice, and assistance helped me greatly: Leonid Sigal, Kyros Kutulakos, Roger Grosse, Animesh Garg, James Lucas, Eleni Triantafillou, Will Grathwohl, David Madrass, Jesse Betencourt, Sajad Norouzi, Ethan Fetaya, Jörn-Henrik Jacobsen.

Finally, I would like to thank my close friends and family, which supported me through a most difficult period of my life. Starting with Ady Bar-El, my amazing partner, who carried me on her back through long periods of mental, social, and financial darkness. Ady's patience, kindness, and complete faith in me made this journey possible. Continuing with my family: my parents Gila and Yitshak Livne, and my brothers Avner, Nadav, and Gad Livne. You were a beacon of light and love, and I could have not completed this journey without you. And closing with my dear friends: Tim Lussier, Barak Edelstein, Ariel Snir, Milan Barboza, Kfir Sharlin, Eran Ben-Ari, Assaf Dayan, Oran Rimon, Assaf Leon, Ariel Szapiro, Ran Livne, Yaniv Hason, David Cadotte, and Micha Blankstein. You never stopped believing in me, even when I did not believe in myself. You provided me with advice, comforting words, and even financial support. This journey showed me how much love and support is present in my life, and I am thankful for that.

Contents

Chapter 1

Introduction

This thesis concerns learning high mutual information representations with latent variable models (LVMs). In particular, we are interested in generative models which are also probability density estimators, where VAE is a canonical example. In what follows we characterize key challenges in existing solutions, and describe a novel estimator that achieves our goals.

Mutual information, together with disentanglement, is considered to be a cornerstone for useful representations (Belghazi et al., 2018; Hjelm et al., 2019). Unfortunately, estimating mutual information in high dimensional continuous variables is challenging (Belghazi et al., 2018). Normalizing flows (Rezende and Mohamed, 2015; Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018; Ho et al., 2019) directly maximizes mutual information by restricting the architecture to be invertible and tractable. This, however, requires the latent dimension to be the same as the dimension of the observations (*i.e.*, no bottleneck). As a consequence, normalizing flows are not well suited to learning a concise representation of high dimensional data (*e.g.*, images), which often lies on low dimensional representation.

VAEs (Kingma and Welling, 2013) are widely used as latent variable models for representation learning of a lower dimensional latent space. The VAE provides a strong sampling capability (*e.g.*, Razavi et al. (2019)), which is considered as a proxy for representation quality, in addition to auxiliary tasks such as classification (Bengio et al., 2017). Nevertheless, it has been observed that a powerful decoder can suffer from posterior collapse (Bowman et al.,

2015; Chen et al., 2016b; Razavi et al., 2019; van den Oord et al., 2016, 2017b), where the decoder effectively ignores the encoder in some dimensions, and the learned representation has low mutual information with the observations. While several attempts to mitigate the problem have been proposed (*e.g.*, Alemi et al. (2017); Razavi et al. (2019)), the root cause has not been identified.

GANs (Goodfellow et al., 2014b), which focus mainly on decoder properties, without a proper inference model, have been shown to minimize JSD between the data distribution $\mathcal{P}(\boldsymbol{x})$ and the model generative process $q_{\boldsymbol{\theta}}(\boldsymbol{x})$. Extension of GANs allow representation learning (Dumoulin et al., 2017; Donahue et al., 2016a), but do not provide a probability density estimator. Additional representation learning models include contrastive predictive coding (CPC, van den Oord et al. (2018)), and Deep InfoMax (Hjelm et al., 2019). Here, both models target high mutual information in the learned representation. However, similar to GAN, no probability density estimator is learned.

This thesis introduces the Mutual Information Machine (MIM), an autoencoder model for learning joint distributions over observations and latent states, which is trained with a novel symmetric variational inference framework. MIM offers high mutual information, low marginal latent entropy, and consistent encoder and decoder.

We systematically probe different terms in the variational bound, providing intuition about MIM. Experiments show that MIM is capable of learning a latent representation with high mutual information, and good unsupervised clustering, while providing data log likelihood comparable to VAE. We demonstrate state of the art results on image and language data. In particular, we show for the first time a latent variable model for language modelling that achieves competitive performance with state of the art auto-regressive models.

Chapter 2

Background

Probabilistic modelling comprises multiple research areas, including generative models, probability density estimators, and representation learning, where each research area covers an overlapping subset of learning methods and models. Commonly used generative models include Generative Adversarial Networks (GAN, Goodfellow et al. (2014b)), and Variational Autoencoders (VAE, Kingma and Welling (2013)); probability density estimators models include autoregressive models (AR, Frey et al. (1995); Larochelle and Murray (2011)), normalizing flows (NF, Dinh et al. (2014)), and VAE; and representation learning models include Autoencoder (AE, Rumelhart et al. (1986)), Contrastive Predictive Coding (CPC, van den Oord et al. (2018)), Deep InfoMax (DIM, Hjelm et al. (2019)), and VAE.

In what follows we use the notations $\mathcal{P}(\boldsymbol{x})$ and $\mathcal{P}(\boldsymbol{z})$ for priors over the observations $\boldsymbol{x}$ and latent $\boldsymbol{z}$, correspondingly, to emphasize that these priors are given, and that we can draw fair random samples from them, but not necessarily evaluate the likelihood of samples under them. Here we often refer to them as *anchors* to further emphasize their role. We also disambiguate between two factorizations of the joint distribution over $\boldsymbol{x}$ and $\boldsymbol{z}$, namely the decoding distribution $p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$, and the encoding distribution $q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})$. Finally, we denote the parameters of a model with $\boldsymbol{\theta}$.

A particularly interesting model is VAE, a latent variable model (LVM) which provides a unified framework for all three research areas. That is, VAE is a generative model, a

probability density estimator, and is used in representation learning. Here we present a new LVM, the Mutual Information Machine (MIM), which is closely related to VAE. The proposed estimator comprises a generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{z})$, a probability density estimator $q_{\boldsymbol{\theta}}(\boldsymbol{x})$, and a posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ that enable the learning of latent representations, similar to VAE.

Unlike VAE, the proposed model is symmetric with regard to the observations and the latent state. More explicitly, MIM estimates a joint distributions over observations and latent state $\mathcal{P}(\boldsymbol{x}, \boldsymbol{z})$. This is in contrast to VAE which estimates the observations marginal $\mathcal{P}(\boldsymbol{x})$ only, and where a tractable approximate encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is a "second class citizen". In addition, MIM is formulated with a symmetric divergence, namely the Jensen-Shannon Divergence (JSD), as opposed to the asymmetric Kullback-Leibler divergence (KLD) in the formulation of VAE.

2.1 Generative models and Probability Density Estimation

Given the vast literature on generative models, here we only touch on the major bodies of work related to MIM. To this end there has been a significant interest in learning probabilistic generative models, in hope that such models can learn a salient representation due to their ability to generate realistic observations. The ability to generate good samples suggests some sort of fundamental "understanding" of the observed data, which is hopefully captured within the learned representation. Furthermore, generative models can be trained in an unsupervised fashion. The attraction of unsupervised learning stems from a desire to exploit vast amounts of unlabelled data, especially when downstream tasks are either unknown *a priori*, or when one lacks ample task-specific training data. And while samples from models trained on heterogeneous data may not resemble one's task domain per se, conditional models can be used to isolate manifolds or sub-spaces associated with particular classes or attributes (*e.g.*, Kingma and Dhariwal (2018)).

In addition, the target distribution can be defined implicitly (*i.e.*, non-parametric distribution) by providing a training sample set. Training a probability density estimator entails

a concise representation of the target distribution, in the case of a parametric generative model. Generative models are often formulated as a latent variable model (LVM), which models the observations conditioned on a latent code (*i.e.*, decoding distribution). This, in turn, allows the models to learn a low dimensional representation which captures the variability of the data, ideally in a way which is semantically meaningful to humans, or useful for downstream tasks. We note that a generative model does not necessarily allow for inference, which requires the learning of a probability density estimator (*e.g.*, GAN).

Generative Adversarial Networks (GAN, Goodfellow et al. (2014b)), which focus mainly on decoder properties, without a proper inference model, have been shown to minimize Jensen-Shannon Divergence (JSD) between the data anchor $\mathcal{P}(\boldsymbol{x})$ and the model generative process $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{P}(\boldsymbol{z})}\left[p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]$. Adversarial learning entails a minmax optimization problem between an adversarial parametric discriminator and a parametric decoder. The training procedure allows to enforce a non-parametric data distribution over a parametric generator. A trained GAN is characterized by realistic sampling, however, despite the high quality of samples, the lacking of inference model in GAN limits its use. We further discuss recent GAN extensions that incorporate the inference task in Section 2.2.

Other common issues with GAN relate to difficulties in training (*i.e.*, instabilities in training), and to mode collapse (*i.e.*, only a few modes of the true distribution are captured by the trained model, see Salimans et al. (2016); Che et al. (2016)). Various methods exists to address the aforementioned issues. Gulrajani et al. (2017), for instance, propose to regularize the gradients of the discriminator in order to stabilize training, and Liu et al. (2019) to regularize the GAN loss with preservation of the normalized pairwise distance between the latent and the corresponding observed samples. We note that while multiple methods have been demonstrated to be effective, the use of a regularizer introduces additional hyper-parameters which are problem dependent, and as such is undesirable in the general case.

As opposed to adversarial learning, methods related to maximum-likelihood (ML) learning are typically stable to train, with an added benefit where the learned generative model can

also function as a probability density function (PDF) estimator. Prominent models include variational auto-encoders (VAE), which maximize a variational lower bound on the data log likelihood $\mathcal{P}(\boldsymbol{x})$ while learning an approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (Rezende et al., 2014; Kingma and Welling, 2013; van den Berg et al., 2018; Kingma et al., 2016); and PDF estimators which directly maximize the marginal log-likelihood under samples from the data distribution (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015; Kingma and Dhariwal, 2018; Bengio and Bengio, 1999; Larochelle and Murray, 2011; Papamakarios et al., 2017).

The VAE offers a tractable variational lower bound, and efficient learning (*i.e.*, unbiased and low variance gradient estimation) with high dimensional observations when combined with the reparameterization trick (Kingma and Welling, 2013). Inference, however, remains challenging for VAEs as the model does not include a probability density estimator over the observations, but rather includes a lower bound (Schmah et al., 2009; Papamakarios et al., 2017; Dinh et al., 2016, 2014). In particular, VAE comprises an approximate inference model which has been empirically prone to suffer from uninformative learned representation when a powerful decoder is used. This phenomena is often named *posterior collapse* (Bowman et al., 2015; van den Oord et al., 2017b)).

Posterior collapse is characterized with low mutual information between observations and inferred latent states, and is commonly observed when the posterior matches the prior in some of the dimensions (*i.e.*, small KLD between posterior and prior). Under a collapsed posterior, a trained VAE model is effectively reduced into a probability density estimator (*i.e.*, the decoder), since the learned latent codes carry little information about the corresponding observations. A collapsed VAE posterior will likely be problematic for representation learning and downstream applications. Interestingly, in many cases it was observed that a collapsed model might also lead to relatively poor probability density estimation (Bowman et al., 2015). Multiple methods to mitigate posterior collapse have been suggested (Chen et al., 2016b; van den Oord et al., 2016, 2017b), by means of regularization of the loss, or restriction of the decoder architecture. We note that while existing methods do mitigate the observed symptoms of posterior collapse (*e.g.*, large variance in posterior), none address the fundamental cause

which is rooted within the formulation (*i.e.*, minimization of the KLD term in the objective) as argued in this thesis.

Alternatively, in recent years several tractable PDF estimators demonstrated great success in modelling complex high dimensional data (*e.g.*, images). Auto-regressive models (Germain et al., 2015; Bengio and Bengio, 1999; Larochelle and Murray, 2011; Papamakarios et al., 2017) and normalizing flows (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015; Kingma and Dhariwal, 2018) train with maximum likelihood (ML), avoiding approximations by choosing a tractable parameterization of probability density. Auto-regressive models assume a conditional factorization of the density function, yielding a tractable probability model. While demonstrating the ability to learn complex high-dimensional distributions, sampling from AR models is expensive (Kingma et al., 2016).

Normalizing flows model the distribution with a series of invertible transformations of a known base distribution, but are somewhat problematic in terms of the memory and computational costs associated with large volumes of high-dimensional data. Nevertheless, successful flow-based generative models (*e.g.*, Kingma and Dhariwal (2018); Durkan et al. (2019)) offer the expressiveness to represent target distributions well, with effective sampling and exact inference (*i.e.*, being a normalized PDF). A notable example for normalizing flows is NICE, being a particularly useful architecture (Dinh et al., 2014, 2016). While invertibility can be used to trade memory with compute requirements (Chen et al., 2018; Gomez et al., 2017), training powerful density estimators remains challenging with large volumes of high-dimensional data (*e.g.*images).

2.2 Representation learning and Symmetry in Probabilistic Models

There has been a rapid progress in recent years in the research area of representation learning, with various methods addressing different aspects of the question: what constitutes a "good" representation? Such methods include information theory-based (Hjelm et al., 2019; van den

Oord et al., 2018), variational inference (Chen et al., 2016b), adversarial learning (Donahue and Simonyan, 2019), and combinations of those methods (e.g., adversarial and variational learning Pu et al. (2017)), to name a few.

Mutual information is a natural indicator of the quality of a learned representation (Hjelm et al., 2019), along with other characteristics, such as the compositionality of latent factors that are expected to be useful in downstream tasks, like transfer learning (Bengio et al., 2017). Mutual information is, however, computationally difficult to estimate for continuous high-dimensional random variables. As such, it can be hard to optimize when learning latent variable models (Chen et al., 2016a; Belghazi et al., 2018; Hjelm et al., 2019).

Chen et al. (2016a), for instance, propose a variational bound on mutual information in order to regularize GAN. A more direct approach was taken in Belghazi et al. (2018), which introduces a powerful parametric estimator to mutual information, which is scalable to high dimensional random variables. The proposed mutual information estimator is then used in (Hjelm et al., 2019) in order to learn a latent representation with high mutual information between local patches of observations and a latent code (*i.e.*, leading to low entropy in the latent code). Interestingly enough, using the same method to maximize the mutual information between observations and a global latent code (*i.e.*, approximate invertibility between observations, and latent code) did not yield as good results, supporting the notion that high mutual information by itself does not constitute a good representation. Further more, the best performing model used both losses, implying that high mutual information and low entropy in latent code leads to a useful representation.

While mutual information indeed plays a significant role in representation learning (*i.e.*, low mutual information between latent and observations is undesirable), mutual information alone does not suffice (*i.e.*, any invertible function maximizes the mutual information). This notion is also reiterated by van den Oord et al. (2018), which presents Contrastive Predictive Coding (CPD), a model that maximizes the mutual information between pairs of local observation patches and a global autoregressive latent code. This raises a fundamental question: given a constant value of mutual information between observations and latent codes,

what other criteria can be used in order to determine the quality of representation? The methods by (Hjelm et al., 2019; van den Oord et al., 2018) opted for solutions which encourage representations with information that is shared across dimensions of an observation. We hypothesize that this preference resembles compression of information, where lower entropy of the latent code translates to information (*i.e.*, code) that is shared across an observation.

VAEs (Kingma and Welling, 2013) are widely used as latent variable models for representation learning. The VAE provides a strong sampling capability (*e.g.*, Razavi et al. (2019)), which is considered as a proxy for quality of the learned representations, in addition to auxiliary tasks such as classification (Bengio et al., 2017). Nevertheless, it has been observed that a powerful decoder can suffer from posterior collapse (Bowman et al., 2015; Chen et al., 2016b; Razavi et al., 2019; van den Oord et al., 2016, 2017b), where the decoder effectively ignores the encoder in some dimensions, and the learned representation has low mutual information with the observations.

While several attempts to mitigate the problem have been proposed (Alemi et al., 2017; Razavi et al., 2019), a root cause has yet to be identified. More explicitly, posterior collapse is commonly treated as an optimization challenge (Bowman et al., 2015), or as an information preference property in certain cases (Chen et al., 2016b). Chen et al. (2016b) limit the decoding distribution to be locally autoregressive (*i.e.*, autoregressive within a local window), which encourages the latent code to capture the global structure. However, Chen et al. (2016b) assume that the true distribution is not locally autoregressive for the proposed method to work, and as such the proposed solution still relies on stabilization methods of the optimization process (*e.g.*, free bits Kingma et al. (2016)).

An alternative to limiting the decoder expressiveness is to choose families of distributions for the prior and approximate posterior such that the KL divergence term (*i.e.*, in ELBO) is bounded below by a non-negative constant. Examples comprise VQ-VAE (van den Oord et al., 2017b), which learn a discrete latent with a fixed KL value; and Delta VAE (Razavi et al., 2019), which models the prior as an autoregressive process, effectively presenting a moving target to the posterior. Both approaches, however, encounter the problem of

"posterior holes" (coined by Rezende and Viola (2018)), where the aggregated posterior assign only a small probability mass to areas of high probability in the prior.

The proposed solution, in both cases, is to learn the prior as a post-processing step. A more optimal approach is taken by Tomczak and Welling (2017), which mitigate the problem by learning the prior as a mixture model of the posteriors with learnable pseudo-inputs. MIM follow a similar approach, and explicitly learns a prior (possibly with independent parameters).

As mentioned above, mutual information, together with disentanglement, is considered to be a cornerstone for useful representations (Belghazi et al., 2018; Hjelm et al., 2019). Normalizing flows (Rezende and Mohamed, 2015; Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018; Ho et al., 2019) directly maximizes mutual information by restricting the architecture to be invertible and tractable, while learning a disentangled representation (*i.e.*, non-linear independent component analysis NICE). This, however, requires the latent dimension to be the same as the dimension of the observations (*i.e.*, no bottleneck). As a result no information is lost, and thus normalizing flows are not well suited to learning a concise (*i.e.*, low dimensional) representation of high dimensional data (*e.g.*, images).

Invertible models, can however, be used in learning small dimensional semantic latent representation, as proposed by Ardizzone et al. (2019). The main idea in Ardizzone et al. (2019) is to assign all unused dimensions (*i.e.*, the dimensionality gap between a semantic latent representation such as class label, and the observations) to an additional latent random variable with a known distribution (*e.g.*, Gaussian) that captures all extraneous information (*i.e.*, non-semantic latent representation). Training entails learning the parameters of an invertible neural network (INN) model such as NICE (Dinh et al., 2014) with ML, combining supervised (*i.e.*, semantic latent) and unsupervised (*i.e.*, non-semantic latent) losses. The proposed method offers efficient computation of normalized distributions, similar to MIM. A main disadvantage of Ardizzone et al. (2019) lies in the requirement to maintain the dimensionality of observations, which might be expensive (*i.e.*, in compute or memory) for very high-dimensional data (*e.g.*, high resolution images).

Adversarial learning has also been used in representation learning. Since GAN (Goodfellow et al., 2014b) focuses mainly on decoding, symmetry has to be introduced by simultaneously training encoding and decoding distributions. Here we refer to symmetry in modelling the encoding and decoding processes. In particular, prior work recognizes the importance of symmetry in learning generative models with reference to symmetric discriminators on $\boldsymbol{x}$ and $\boldsymbol{z}$ (Bang and Shim, 2018; Donahue et al., 2016a; Dumoulin et al., 2017). Introducing adversarial learning over the latent code also allows the learning procedure to enforce an implicit distribution (*i.e.*, given via samples set) over the posterior (Goodfellow et al., 2014b; Makhzani, 2018; Makhzani et al., 2015). We note that variational inference can be combined with adversarial learning. For instance, Pu et al. (2017) focus on minimizing symmetric KLD, which is intractable in the general case (*i.e.*, inability to evaluate likelihood over observations $\log \mathcal{P}(\boldsymbol{x})$). Pu et al. (2017) solve the tractability challenge by using adversarial learning procedure, which does not require such likelihood evaluation.

The importance of symmetry has also been recognized in Bornschein et al. (2015), which presents the Bidirectional Helmholtz Machine, and target symmetry by enforcing encode/decoder consistency. Unlike MIM, the formulation by Bornschein et al. (2015) models the joint density in terms of the geometric mean between the encoder and decoder, for which one must compute an expensive partition function. Here, we propose MIM which is defined as the arithmetic mean between the encoder and decoder. Among other advantages, MIM does not require the computation of such partition function. Despite that limitation, Bornschein et al. (2015) demonstrate the positive effect symmetry has over a learned representation.

2.3 Representation learning and Compositionality

Another fundamental aspect in representation learning is compositionality in probabilistic models, which is still an open question, by large. Intuitively, a compositional model learns a distribution over observations conditioned on multiple disentangled latent factors, such that the conditional distributions of the factors given an observation are independent. One

example for compositionality is a naive model with a disentangled latent representation, where each latent factor affects a non-overlapping dimensions in the observed random variable. In such a case the model is trivially compositional, but is also not that interesting. In what follows we discuss the more interesting case, of a compositional model where each latent factor can potentially affect all dimensions in observations.

A canonical example for compositional models is Energy-Based Models (EBM), which provide a flexible representation of undirected graphical models, at the expense of an unknown partition function. Well known EBM models includes Restricted Boltzmann Machine (RBM, Smolensky (1986)), and Product-of-Experts (PoE, Hinton (2002a)). While such models provide a powerful probabilistic modelling capabilities, they are also notoriously hard to train (*i.e.*, unstable to train), slow to sample (*i.e.*, require MCMC for sampling), allow for evaluation of log-likelihood ratio only (*i.e.*, distributions are unnormalized), and as such had proven to have limited use in real-world problems.

Recognizing the importance of normalized distributions, Poon and Domingos (2012) propose sum-product networks, an architecture that allows the computation of unnormalized distribution and the normalizing factor in polynomial run-time complexity. The proposed model can be trained with backpropagation and Expectation-Maximization (EM), and provides efficient computation of conditional and marginal distributions. Despite showing promising results, sum-product networks have yet to match the results of competing approaches in high dimensional and complex data (*e.g.*, images).

Here we propose a new probability density estimator named Mutual Information Machine (MIM), providing a stable learning method (*i.e.*, variational bound over entropy of a joint distribution) to approximate an undirected graphical model. Compared to EBM, MIM learns normalized distributions, and approximates the undirected graphical model by means of learning multiple independently parameterized distributions of the same underlying joint distribution. In effect, MIM trades the ideal compositionality and approximate sampling of EBM with ideal sampling, normalized log likelihood evaluation, and approximate compositionality. MIM learning shares some resemblance with VAE (*i.e.*, MIM is a PDF

estimator, a representation learning framework, and a generative model, similar to VAE), with an additional property of symmetry/consistency (*i.e.*, MIM learns multiple consistent factorization of the same distribution).

Chapter 3

MIM: Mutual Information Machine

3.1 Introduction

Latent Variable Models (LVMs) are probabilistic models that enhance the distribution over observations into a joint distribution over observations and latent variables, with VAE (Kingma and Welling, 2013) being a canonical example. It is hoped that the learned representation will capture salient information in the observations, which in turn can be used in downstream tasks (*e.g.*, classification, inference, generation). In addition, a fixed-size representation enables comparisons of observations with variable size (*e.g.*, time series). The VAE popularity stems, in part, from its versatility, serving as a generative model, a probability density estimator, and a representation learning framework.

Mutual information (MI) is often considered a useful measure of the quality of a latent representation (*e.g.*, Hjelm et al. (2019); Belghazi et al. (2018); Chen et al. (2016a)). Indeed, many common generative LVMs can be seen as optimizing an objective involving a sum of mutual information terms and a divergence between encoding and decoding distributions (Zhao et al., 2018a). These include the VAE (Kingma and Welling, 2013), β-VAE (Higgins et al., 2017), and InfoVAE (Zhao et al., 2017), as well as GAN models including ALI/BiGAN (Dumoulin et al., 2017; Donahue et al., 2016b), InfoGAN (Chen et al., 2016a), and CycleGAN (Zhu et al., 2017). From an optimization perspective, however, different

objectives can be challenging, as MI is notoriously difficult to estimate (Belghazi et al., 2018), and many choices of divergence require adversarial training.

This thesis introduces a class of generative LVMs over data $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$ called the Mutual Information Machine (MIM). The learning objective for MIM is designed around three fundamental principles, namely

1. Consistency of encoding and decoding distributions;
2. High mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$;
3. Low marginal entropy.

Consistency enables one to both generate data and infer latent variables from the same underlying joint distribution Dumoulin et al. (2017); Donahue et al. (2016b); Pu et al. (2017). High MI ensures that the latent variables accurately capture the factors of variation in the data. Beyond consistency and mutual information, our third criterion ensures that each distribution efficiently encodes the required information, and does not also model spurious correlations.

MIM is formulated using

1. The Jensen-Shannon divergence (JSD), a symmetric divergence that also forms the basis of ALI/BiGANs and ALICE, (Dumoulin et al., 2017; Donahue et al., 2016b; Li et al., 2017); and
2. The entropy of the encoding and decoding distributions, encouraging high mutual information and low marginal entropy.

We show that the sum of these two terms reduces to the entropy of the mixture of the encoding and decoding distributions defined by the JSD. Adding parameterized priors, we obtain a novel upper bound on this entropy, which forms the MIM objective. Importantly, the resulting cross-entropy objective enables direct optimization using stochastic gradients with reparameterization, thereby avoiding the need for adversarial training and the estimation of mutual information. MIM learning is stable and robust, yielding consistent encoding and

decoding distributions while avoiding over- and under-estimation of marginals (cf. Pu et al. (2017)), and avoiding posterior collapse in the conventional VAE.

We provide an analysis of the MIM objective and further broaden the MIM framework, exploring different sampling distributions and forms of consistency. We also demonstrate the benefits of MIM when compared to VAEs experimentally, showing that MIM in particular benefits greatly from more expressive model architectures by utilizing the additional capacity for better representation (*i.e.*, compressed representation with higher mutual information).

3.2 Generative LVMs

We consider the class of generative latent variable models (LVMs) over data $\boldsymbol{x} \in \mathcal{X}$ and latent variables $\boldsymbol{z} \in \mathcal{Z}$. These models generally assume an explicit prior over latent variables, $\mathcal{P}(\boldsymbol{z})$, and an unknown distribution over data, $\mathcal{P}(\boldsymbol{x})$, specified implicitly through data examples. As these distributions are fixed and not learned, we call them *anchors*. The joint distribution over $\boldsymbol{x}$ and $\boldsymbol{z}$ is typically expressed in terms of an encoding distribution, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$, or a decoding distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})$, where $q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$ are known as the encoder and decoder.

Zhao et al. (2018a) describe a learning objective that succinctly encapsulates many LVMs proposed to date,

$$\mathcal{L}(\theta) = \alpha_1 I_q(\boldsymbol{x};\boldsymbol{z}) + \alpha_2 I_p(\boldsymbol{x};\boldsymbol{z}) + \lambda^\top \mathcal{D} \ . \tag{3.1}$$

where θ comprises the model parameters, $\alpha_1, \alpha_2, \lambda$ are weights, $\mathcal{D}$ is a set of divergences, and $I_q(\boldsymbol{x};\boldsymbol{z})$ and $I_p(\boldsymbol{x},\boldsymbol{z})$ represent mutual information[1] under the encoding and decoding distributions. The divergences measure inconsistency between the encoding and decoding distributions. By encouraging high MI one hopes to learn meaningful relations between $\boldsymbol{x}$ and $\boldsymbol{z}$ Higgins et al. (2017); Zhao et al. (2017); Chen et al. (2016a); Zhu et al. (2017); Li et al. (2017).

[1] $I(\boldsymbol{x};\boldsymbol{z}) = \mathcal{D}_{\mathrm{KL}}\left(p(\boldsymbol{x},\boldsymbol{z}) \,\|\, p(\boldsymbol{x})p(\boldsymbol{z})\right)$

One of the best-known examples within this framework is the variational auto-encoder (VAE) Kingma and Welling (2013); Rezende et al. (2014), which sets $\alpha_1 = \alpha_2 = 0$, $\lambda = 1$, and the divergence $\mathcal{D}$ to

$$\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) \ . \tag{3.2}$$

One can show that this objective is equivalent, up to an additive constant, to the evidence lower bound typically used to specify VAEs. Although widely used, two issues can arise when training a VAE. First, the asymmetry of the KL divergence can lead it to assign high probability to unlikely regions of the data distribution Pu et al. (2017). Second, it sometimes learns an encoder that essentially ignores the input data and instead models the latent prior. This is known as *posterior collapse*. The consequence is that latent states convey little useful information about observations.

Another example is the ALI/BiGAN model Dumoulin et al. (2017); Donahue et al. (2016b), which is instead defined by the Jensen-Shannon divergence:

$$\begin{aligned} \mathcal{D}_{\mathrm{JS}} \ &= \ \frac{1}{2}\Big(\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) \\ &\qquad + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) \Big) \ , \end{aligned} \tag{3.3}$$

where $\mathcal{M}_{\mathcal{S}}$ is an equally weighted mixture of the encoding and decoding distributions; *i.e.*,

$$\mathcal{M}_{\mathcal{S}} \ = \ \frac{1}{2}\big(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) + q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \big) \ . \tag{3.4}$$

The symmetry in the objective helps to keep the marginal distributions consistent, as with the symmetric KL objective used in Pu et al. (2017).

As described by Zhao et al. (2018a), many such methods belong to a difficult class of objectives that usually rely on adversarial training, which can be unstable. In what follows we show how by further encouraging low marginal entropy, and with a particular combination of MI and divergence, we obtain a framework that accomplishes many of the aims of the

previous methods while also allowing for stochastic gradient-based optimization.

3.3 Variational Autoencoders

VAE learning entails optimization of a variational lower bound on the log-marginal likelihood of the data, $\log \mathcal{P}(\boldsymbol{x})$, to estimate the parameters θ of an approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ over latent states $\boldsymbol{z}$ (*i.e.*, the encoder) and a corresponding decoder, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ (Kingma and Welling, 2013; Rezende et al., 2014). A prior over the latent space, $\mathcal{P}(\boldsymbol{z})$, often assumed to be an isotropic Gaussian, serves as a prior for $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ in the evidence-lower-bound (ELBO) on the marginal likelihood:

$$\log \mathcal{P}(\boldsymbol{x}) \;\geq\; \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, \mathcal{P}(\boldsymbol{z})\right) \;,$$

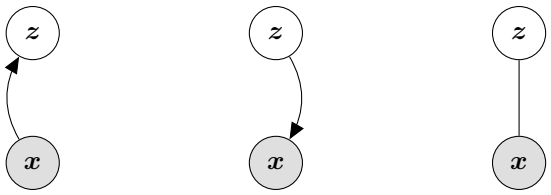
Here, we use the notation $\mathcal{P}(\boldsymbol{x})$ and $\mathcal{P}(\boldsymbol{z})$ to emphasize that these priors are given, and that we can draw random samples from them, but not necessarily evaluate the log-likelihood of samples under them. In what follows we often refer to them as *anchors* to further emphasize their role.

With amortized posterior inference, we take expectation over the observation distribution, $\mathcal{P}(\boldsymbol{x})$, to obtain the VAE objective:

$$\begin{aligned}
\mathcal{R}_{\mathrm{VAE}}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}(\boldsymbol{x})}\left[\,\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, \mathcal{P}(\boldsymbol{z})\right)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}(\boldsymbol{x}),\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log \mathcal{P}(\boldsymbol{z}) - \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \;, \qquad (3.5)
\end{aligned}$$

Gradients of Eqn. (3.5) are estimated through MC sampling from $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ with reparameterization, yielding unbiased low-variance gradient estimates (Kingma and Welling, 2013; Rezende et al., 2014).

VAEs are normally thought of as maximizing a lower bound on the data log-likelihood, however it can also be expressed as minimizing the divergence between two joint distributions over $\boldsymbol{x}$ and $\boldsymbol{z}$. To see this, we first subtract $\log \mathcal{P}(\boldsymbol{x})$ from (3.5), which does not change



(a) $\mathcal{P}(\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (b) $\mathcal{P}(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ (c) $p(\boldsymbol{x},\boldsymbol{z})$

Figure 3.1: A MIM model learns two factorizations of a joint distribution: (a) encoding; (b) decoding factorizations; and (c) the estimated joint distribution (an undirected graphical model).

the gradients of the objective with respect to θ. We then negate the result, as we will be performing minimization. This yields a VAE loss

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}) \;=\; -\mathcal{R}_{\text{VAE}}(\boldsymbol{\theta}) + \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}(\boldsymbol{x})}\left[\log \mathcal{P}(\boldsymbol{x})\right] \;=\; \mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})\right) . \tag{3.6}$$

The VAE optimization is therefore equivalent to minimizing the KL divergence between an encoding distribution $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})$ and a decoding distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})$.

3.4 Symmetry and Mutual Information

Our goal is to find a consistent encoder-decoder pair, representing a joint distribution over the observation and latent domains, with high mutual information between observations and latent states. By consistent, we mean that the encoding and decoding distributions, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})$, define the same joint distribution. Figure 3.1 depicts this basic idea, in which the same distribution is identical under both the encoding and decoding factorizations. Effectively, we estimate an undirected graphical model with two valid factorizations. We note that consistency is achievable in the VAE when the approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is capable of representing the posterior under the decoding distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. In the general case, however, consistency is not usually achieved.

In contrast to the asymmetric divergence between encoding and decoding distributions in the VAE objective (3.6), here we consider a symmetric measure, namely, the well-known

Jensen-Shannon divergence (JSD),

$$\mathrm{JSD}(\boldsymbol{\theta}) \;=\; \frac{1}{2}\Big(\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) \Big) \;, \tag{3.7}$$

where $\mathcal{M}_{\mathcal{S}}$ is an equally weighted mixture of the encoding and decoding distributions; i.e.,

$$\mathcal{M}_{\mathcal{S}} \;=\; \frac{1}{2}\big(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) + q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \big) \;. \tag{3.8}$$

In addition to encoder-decoder consistency, to learn useful latent representations we also want high mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$. Indeed, the link between mutual information and representation learning has been explored in recent work (Belghazi et al., 2018; Chen et al., 2016a; Hjelm et al., 2019). Here, to emphasize high mutual information, we add a particular regularizer of the form

$$\mathrm{R_H}(\boldsymbol{\theta}) \;=\; \frac{1}{2}\big(H(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})) + H(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})) \big) \;. \tag{3.9}$$

This is the average of the joint entropy over $\boldsymbol{x}$ and $\boldsymbol{z}$ according to the encoding and decoding distributions. This is related to mutual information by the identity $H(\boldsymbol{x},\boldsymbol{z}) = H(\boldsymbol{x}) + H(\boldsymbol{z}) - I(\boldsymbol{x};\boldsymbol{z})$. That is, minimizing joint entropy encourages the minimization of the marginal entropy and maximization of the mutual information. In addition to encouraging high mutual information, and high compression of the marginals, one can show that this particular regularizer has a deep connection to JSD and the entropy of $\mathcal{M}_{\mathcal{S}}$, i.e.,

$$\mathrm{JSD}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta}) \;=\; H(\mathcal{M}_{\mathcal{S}}) \;. \tag{3.10}$$

The derivation for Eqn. (3.10) is given in Appendix A.1.

3.5 Mutual Information Machine

The loss function in Eqn. (3.10) reflects our desire for model symmetry and high mutual information. A symmetric model will learn an encoder and decoder that represent different factorization of the same underlying joint distribution. Nevertheless, it is difficult to optimize directly since we do not know how to evaluate $\log \mathcal{P}(\boldsymbol{x})$ in the general case (*i.e.*, we do not have an exact closed-form expression for $\mathcal{P}(\boldsymbol{x})$). As a consequence, we introduce parameterized approximate priors, $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, to derive tractable bounds on the regularized Jensen-Shannon divergence. This is similar in spirit to VAEs, which introduce a tractable parameterized approximate posterior. These parameterized priors, together with the conditional encoder and decoder, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, comprise a new pair of joint distributions, i.e.,

$$
\begin{aligned}
q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) &\equiv q_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}\right) q_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right) \\
p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) &\equiv p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}\right) p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right) \ .
\end{aligned}
$$

These joint distributions allow us to formulate a new, tractable loss that bounds $H(\mathcal{M}_{\mathcal{S}})$. That is,

$$
\mathcal{L}_{\mathrm{CE}}(\boldsymbol{\theta}) \ \equiv \ CE\left(\mathcal{M}_{\mathcal{S}}\,,\,\mathcal{M}_{\boldsymbol{\theta}}\right) \ = \ \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, \mathcal{M}_{\boldsymbol{\theta}}\right) + H\left(\mathcal{M}_{\mathcal{S}}\right) \ \geq \ H\left(\mathcal{M}_{\mathcal{S}}\right) \ , \tag{3.11}
$$

where

$$
\mathcal{M}_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right) \ = \ \frac{1}{2}\big(\, p_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right) + q_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right) \,\big) \ , \tag{3.12}
$$

and $CE\left(\mathcal{M}_{\mathcal{S}}\,,\,\mathcal{M}_{\boldsymbol{\theta}}\right)$ denotes the cross-entropy between $\mathcal{M}_{\mathcal{S}}$ and $\mathcal{M}_{\boldsymbol{\theta}}$.

We refer to $\mathcal{L}_{\mathrm{CE}}$ as the cross-entropy loss. It aims to match the model prior distributions to the anchors, while also minimizing $H\left(\mathcal{M}_{\mathcal{S}}\right)$. A key advantage of this formulation is that the cross-entropy loss can be trained by Monte Carlo sampling from the anchor distributions with reparameterization (Kingma and Welling, 2013; Rezende et al., 2014).

At this stage it might seem odd to introduce a parametric prior for $\mathcal{P}(\boldsymbol{z})$. Indeed, setting

it directly is certainly an option. Nevertheless, in order to achieve consistency between $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ it can be advantageous to allow $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ to vary. Essentially, we trade-off latent prior fidelity for increased model consistency. We provide more insights about this in Appendix B.2.

One issue with $\mathcal{L}_{\text{CE}}$ is that, while it will try to enforce consistency between the model and the anchored distributions, i.e., $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \approx p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \approx q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})$, it will not directly try to achieve model consistency: $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \approx q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$. To remedy this, we bound $\mathcal{L}_{\text{CE}}$ using Jensen's inequality, *i.e.*,

$$\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \;\equiv\; \frac{1}{2}\big(H\left(\mathcal{M}_{\mathcal{S}}, q_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)\right) + H\left(\mathcal{M}_{\mathcal{S}}, p_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)\right) \big) \geq \;\; \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \; . \tag{3.13}$$

Equation (3.13) gives us the loss function for the Mutual Information Machine (MIM). It is an average of cross entropy terms between the mixture distribution $\mathcal{M}_{\mathcal{S}}$ and the model encoding and decoding distributions respectively. To see that this encourages model consistency, it can be shown that $\mathcal{L}_{\text{MIM}}$ is equivalent to $\mathcal{L}_{\text{CE}}$ plus a non-negative model consistency term; i.e.,

$$\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \;=\; \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) + \text{R}_{\text{MIM}}(\boldsymbol{\theta}) \; . \tag{3.14}$$

The non-negativity of R_{MIM} is a simple consequence of $\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \geq \mathcal{L}_{\text{CE}}(\boldsymbol{\theta})$ in (3.13). One can further show (see Appendix A.2) that $\text{R}_{\text{MIM}}(\boldsymbol{\theta})$ satisfies

$$\begin{aligned} \text{R}_{\text{MIM}}(\boldsymbol{\theta}) \;&=\; \frac{1}{2}\big(\mathcal{D}_{\text{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, p_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)\right) + \mathcal{D}_{\text{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, q_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)\right) \big) \\ &\qquad -\mathcal{D}_{\text{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, \mathcal{M}_{\boldsymbol{\theta}}\right) \qquad (3.15) \\ &=\; \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z} \sim \mathcal{M}_{\mathcal{S}}} \left[-\log \frac{\sqrt{q_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right) \cdot p_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)}}{\frac{1}{2}(q_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right) + p_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right))} \right] \geq 0 \; . \qquad (3.16) \end{aligned}$$

One can conclude from Eqn. (3.15) that R_{MIM} is zero only when the two joint model distributions, $q_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)$ and $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{z}\right)$, are identical under fair samples from the joint sample distribution $\mathcal{M}_{\mathcal{S}}\left(\boldsymbol{x}, \boldsymbol{z}\right)$. In practice we find that encouraging model consistency also helps to stabilize learning.

To understand the MIM objective in greater depth, we find it helpful to express $\mathcal{L}_{\text{MIM}}$ as a sum of fundamental terms that provide some intuition for its expected behavior. In particular, as derived in the Appendix A.3,

$$\begin{aligned}\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \;=\; \mathrm{R_H}(\boldsymbol{\theta}) \;+\; \frac{1}{4}\big(\,\mathcal{D}_{\text{KL}}\left(\mathcal{P}(\boldsymbol{z}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) + \mathcal{D}_{\text{KL}}\left(\mathcal{P}(\boldsymbol{x}) \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\big) \\ +\; \frac{1}{4}\big(\,\mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + \mathcal{D}_{\text{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right)\big)\end{aligned} \tag{3.17}$$

The first term in Eqn. (3.17) encourages high mutual information between observations and latent states. The second shows that MIM directly encourages the model priors to match the anchor distributions. Indeed, the KL term between the data anchor and the model prior is the maximum likelihood objective. The third term encourages consistency between the model distributions and the anchored distributions, in effect fitting the model decoder to samples drawn from the anchored encoder (cf. VAE), and, via symmetry, fitting the model encoder to samples drawn from the anchored decoder (both with reparameterization). As such, MIM can be seen as simultaneously training and distilling a model distribution over the data into a latent variable model. The idea of distilling density models has been used in other domains, e.g., for parallelizing auto-regressive models (van den Oord et al., 2017a).

In summary, the MIM loss provides an upper bound on the joint entropy of the observation and latent states under the mixture distribution $\mathcal{M}_{\mathcal{S}}$:

$$\begin{aligned}\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \;&=\; \frac{1}{2}\big(\,CE\left(\,\mathcal{M}_{\mathcal{S}}\,,\, q_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right) + CE\left(\,\mathcal{M}_{\mathcal{S}}\,,\, p_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right)\,\big) \\ &=\; CE\left(\,\mathcal{M}_{\mathcal{S}}\,,\, \mathcal{M}_{\boldsymbol{\theta}}\,\right) + \mathrm{R_{MIM}}(\boldsymbol{\theta}) \\ &\geq\; CE\left(\,\mathcal{M}_{\mathcal{S}}\,,\, \mathcal{M}_{\boldsymbol{\theta}}\,\right) \\ &\geq\; H_{\mathcal{M}_{\mathcal{S}}}\left(\,\boldsymbol{x},\boldsymbol{z}\,\right) \\ &=\; H_{\mathcal{M}_{\mathcal{S}}}\left(\,\boldsymbol{x}\,\right) + H_{\mathcal{M}_{\mathcal{S}}}\left(\,\boldsymbol{z}\,\right) - I_{\mathcal{M}_{\mathcal{S}}}(\boldsymbol{x};\boldsymbol{z})\;.\end{aligned} \tag{3.18}$$

Through the MIM loss and the introduction of the parameterized model distribution $\mathcal{M}_{\boldsymbol{\theta}}$, we are pushing down on the entropy of the anchored mixture distribution $\mathcal{M}_{\mathcal{S}}$, which is the

sum of marginal entropies minus the mutual information. Minimizing the MIM bound yields consistency of the model encoder and decoder, low marginal entropies, and high mutual information under $\mathcal{M}_{\mathcal{S}}$ between observations and latent states.

3.5.1 Asymmetric Mutual Information Machine

With some models, like auto-regressive distributions (*e.g.*, PixelHVAE Tomczak and Welling (2017)), sampling from $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$ becomes impractical during learning. This is problematic for MIM, as it draws samples from $\mathcal{M}_{\mathcal{S}}$, a mixture of the anchored encoding and decoding distributions. As an alternative, we introduce a variant of MIM, called A-MIM, that only samples from the encoding distribution. To that end, collecting terms in Eqn. (3.17) which depend on samples from $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$, we obtain

$$\begin{aligned}\mathcal{L}_{\text{A-MIM}} \;&\equiv\; \frac{1}{2}\big(\, CE\left(\,\mathcal{M}^q_{\mathcal{S}}\,,\, q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\,\right) \;+\; CE\left(\,\mathcal{M}^q_{\mathcal{S}}\,,\, p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\,\right)\,\big) \qquad (3.19)\\ &=\; \frac{1}{2}H_q(\boldsymbol{x},\boldsymbol{z}) \;+\; \frac{1}{2}\mathcal{D}_{\text{KL}}\left(\mathcal{P}(\boldsymbol{x})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\\ &\qquad\qquad + \frac{1}{2}\mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)\;, \qquad (3.20)\end{aligned}$$

where $\mathcal{M}^q_{\mathcal{S}} = q_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$. Minimization of $\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta})$ learns a consistent encoder-decoder model with an encoding distribution with high mutual information and low entropy. Formally, one can derive the following bound:

$$\mathcal{L}_{\text{A-MIM}} \;\geq\; CE\left(\,\mathcal{M}^q_{\mathcal{S}}\,,\,\mathcal{M}_{\boldsymbol{\theta}}\,\right) \;\geq\; H_q(\boldsymbol{x},\boldsymbol{z})\;, \qquad (3.21)$$

where the gap between $\mathcal{L}_{\text{A-MIM}}$ and the cross-entropy term has the same form as R_{MIM} in Eqn. (3.15), but with $\mathcal{M}_{\mathcal{S}}$ replaced by $\mathcal{M}^q_{\mathcal{S}}$. It is still 0 when $q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})$.

The main difference between MIM and A-MIM is the lack of symmetry in the sampling distribution. This is similar to the VAE formulation in Eqn. (3.2). Indeed, the final term in Eqn. (3.20) is exactly the VAE loss, with an optionally parameterized latent prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. Importantly, $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ must be defined implicitly using the marginal of the decoding distribution,

otherwise it will be completely independent from the other terms. Without this, A-MIM can be seen as a VAE with a joint entropy penalty, and we have found that this doesn't work as well in practice.

In essence, A-MIM trades symmetry of the JSD for speed. Regardless, we show that A-MIM is often as effective as MIM at learning representations.

3.5.2 A-MIM, VAEs, and Posterior Collapse

The VAE loss in Eqn. (3.2) can be expressed in a form that bears similarity to the A-MIM loss in Eqn. (3.19) (details in the supplementary material); *i.e.*,

$$\mathcal{L}_{\text{VAE}} \;=\; \frac{1}{2}\Big(CE\left(\mathcal{M}_{\mathcal{S}}^{q}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + CE\left(\mathcal{M}_{\mathcal{S}}^{q}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\Big) - H_q(\boldsymbol{x},\boldsymbol{z}). \quad (3.22)$$

Like $\mathcal{L}_{\text{A-MIM}}$ in Eqn. (3.19), the first term in Eqn. (3.22) is the average of two cross-entropy terms between a sample distribution and the encoding and decoding distributions. Here, as in $\mathcal{L}_{\text{A-MIM}}$, they are asymmetric, as samples are drawn only from the encoding distribution. Unlike $\mathcal{L}_{\text{A-MIM}}$, the last term in Eqn. (3.22) encourages high marginal entropies and low MI under the encoding distribution. This plays a significant role in allowing for posterior collapse (*e.g.*, see Zhao et al. (2018a); He et al. (2019)).

3.6 Learning

Here we provide a detailed description of MIM and A-MIM learning, with algorithmic pseudo-code. In addition we offer practical considerations regarding the choice of priors' parameterization , and gradient estimation. The empirical upper bound objective, $\mathcal{L}_{\text{MIM}}$ in Eqn. (3.13), is expressed in terms of two cross-entropy terms. Given N fair samples, $\{\boldsymbol{x}_i, \boldsymbol{z}_i\}_{i=1}^{N}$ drawn from the anchored (sample) distribution, $\mathcal{M}_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z})$ in (3.8), the empirical loss is

$$\hat{\mathcal{L}}_{\text{MIM}}\left(\boldsymbol{\theta}; \{\boldsymbol{x}_i, \boldsymbol{z}_i\}_{i=1}^{N}\right) \;=\; -\frac{1}{2N}\sum_{i=1}^{N} \log\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}_i|\boldsymbol{x}_i)\,q_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right) + \log\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\,p_{\boldsymbol{\theta}}(\boldsymbol{z}_i)\right) \quad ,(3.23)$$

Algorithm 1 MIM learning of parameters $\boldsymbol{\theta}$

Require: Samples from anchors $\mathcal{P}(\boldsymbol{x}), \mathcal{P}(\boldsymbol{z})$

while not converged **do**
  $D_{\text{dec}} \leftarrow \{\boldsymbol{x}_i, \boldsymbol{z}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})\}_{i=1}^{N/2}$
  $D_{\text{enc}} \leftarrow \{\boldsymbol{x}_j, \boldsymbol{z}_j \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})\}_{j=1}^{N/2}$
  $D \leftarrow D_{\text{dec}} \bigcup D_{\text{enc}}$
  # See definition of $\hat{\mathcal{L}}_{MIM}$ in Eq. (3.23)
  $\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \approx \hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D)$
  # Minimize loss
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D)$
end while

Algorithm 2 A-MIM learning of parameters $\boldsymbol{\theta}$

Require: Samples from anchor $\mathcal{P}(\boldsymbol{x})$

while not converged **do**
  $D_{\text{enc}} \leftarrow \{\boldsymbol{x}_j, \boldsymbol{z}_j \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})\}_{j=1}^{N}$
  # See definition of $\hat{\mathcal{L}}_{MIM}$ in Eq. (3.23)
  $\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta}) \approx \hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D_{\text{enc}})$
  # Minimize loss
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D_{\text{enc}})$
end while

where samples from $\mathcal{M}_{\mathcal{S}}$ comprise equal numbers of points from $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$. Samples from the anchors, $\mathcal{P}(\boldsymbol{x})$ and , $\mathcal{P}(\boldsymbol{z})$, are treated as external observations; *i.e.*, we assume we can sample from them but not necessarily evaluate the density of points under the anchor distributions. Algorithm 1 specifies the corresponding training procedure for MIM. The algorithm makes no assumptions on the form of the parameterized distributions (*e.g.*, discrete, or continuous). Similarly, Algorithm 2 specifies the corresponding training procedure for A-MIM.

In practice, for gradient-based optimization, we would like an unbiased gradient estimator without the need to accurately approximate the full expectations per se (*i.e.*, in the cross entropy terms). This is particularly important when dealing with high dimensional data (*e.g.*, images), where it is computationally expensive to estimate the value of the expectation. We next discuss practical considerations for the continuous case and the discrete case.

3.6.1 MIM Parametric Priors

There are several effective ways to parameterize the priors. For the 1D experiments in Appendix B we model $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ using mixtures of isotropic Gaussians. With complex, high dimensional data one might also consider more powerful models (*e.g.*, autoregressive, or flow-based priors). Unfortunately, the use of complex models typically increases the required computational resources, and the training and inference time.

As an alternative, for image data, we make use of the *VampPrior* (Tomczak and Welling, 2017) which models the latent prior as a mixture of posteriors, i.e.,

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \sum_{k=1}^{K} q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x} = \boldsymbol{u}_k) \tag{3.24}$$

with learnable pseudo-inputs $\{\boldsymbol{u}_k\}_{k=1}^{K}$. This is effective and allows one to reduce the need for additional parameters (see Tomczak and Welling (2017) for details on VampPrior's effect over gradient estimation).

3.6.2 Learning with Marginal $q_{\boldsymbol{\theta}}(\boldsymbol{x})$

Algorithm 3 MIM learning with marginal $q_{\boldsymbol{\theta}}(\boldsymbol{x})$

Require: Samples from anchors $\mathcal{P}(\boldsymbol{x}), \mathcal{P}(\boldsymbol{z}), q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})$
Require: Define $q_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right]$
while not converged **do**
  $D_{\text{enc}} \leftarrow \{\boldsymbol{x}_i, \boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})\mathcal{P}(\boldsymbol{x})\}_{i=1}^{N}$
  $\hat{\mathcal{L}}_{\text{MIM}_{\text{enc}}} \leftarrow -\frac{1}{N}\sum_{i=1}^{N}\big(\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i \mid \boldsymbol{z}_i) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_i)\big)$
  $D_{\text{dec}} \leftarrow \{\boldsymbol{x}_i, \boldsymbol{z}_i, \boldsymbol{z}'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\mathcal{P}(\boldsymbol{z}) q_{\boldsymbol{\theta}}(\boldsymbol{z}' \mid \boldsymbol{x})\}_{i=1}^{N}$
  $\hat{\mathcal{L}}_{\text{MIM}_{\text{dec}}} \leftarrow -\frac{1}{2N}\sum_{i=1}^{N}\big(\log q_{\boldsymbol{\theta}}(\boldsymbol{z}_i \mid \boldsymbol{x}_i) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i \mid \boldsymbol{z}_i) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_i) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}'_i) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}'_i) - \log q_{\boldsymbol{\theta}}(\boldsymbol{z}'_i \mid \boldsymbol{x}_i)\big)$
  # Minimize loss
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\ \frac{1}{2}\left(\hat{\mathcal{L}}_{\text{MIM}_{\text{enc}}} + \hat{\mathcal{L}}_{\text{MIM}_{\text{dec}}}\right)$
end while

When we would like to use a marginal data prior (Bornschein et al., 2015),

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}) \;\; = \;\; \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})] \;, \tag{3.25}$$

we must modify the learning algorithm slightly. This is because the MIM objective in Equation (3.13) involves a $\log q_{\boldsymbol{\theta}}(\boldsymbol{x})$ term, and the marginal is intractable to compute analytically. Instead we minimize an upper bound on the necessary integrals using Jensen's inequality. Complete details of the derivations are given below.

For the encoder portion of $\mathcal{M}_{\mathcal{S}}$, the integral can be bounded and approximated as follows.

$$\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})}[\log(q_{\boldsymbol{\theta}}(\boldsymbol{x}))] \\
&\quad = \;\; \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})}[\log(\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}')}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}')])] \\
&\quad \geq \;\; \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] \\
&\quad \approx \;\; \log(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}} \mid \tilde{\boldsymbol{z}})) - \log(q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}} \mid \tilde{\boldsymbol{x}})) + \log(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})) \;.
\end{aligned}$$

where $\tilde{\boldsymbol{x}} \sim \mathcal{P}(\boldsymbol{x})$ and $\tilde{\boldsymbol{z}} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \tilde{\boldsymbol{x}})$.

Similarly, the decoder portion can be bounded and approximated as follows.

$$\begin{aligned}
&\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})}[\log(q_{\boldsymbol{\theta}}(\boldsymbol{x}))] \\
&\quad = \; \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})}[\log(\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}')}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}')])] \\
&\quad \geq \; \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})q_{\boldsymbol{\theta}}(\boldsymbol{z}'|\boldsymbol{x})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}')p_{\boldsymbol{\theta}}(\boldsymbol{z}')}{q_{\boldsymbol{\theta}}(\boldsymbol{z}'|\boldsymbol{x})}\right)\right] \\
&\quad \approx \; \log(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}} \mid \tilde{\boldsymbol{z}'}))) + \log(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}'})) - \log(q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}'} \mid \boldsymbol{x}')).
\end{aligned}$$

where $\tilde{\boldsymbol{z}} \sim \mathcal{P}(\boldsymbol{z})$, $\tilde{\boldsymbol{x}} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \tilde{\boldsymbol{z}})$, and $\tilde{z'} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}' \mid \tilde{\boldsymbol{x}})$.

In both cases, we use importance sampling from $q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})$. The full algorithm, collecting all like terms, is summarized in Algorithm 3.

3.6.3 Gradient Estimation

Optimization is performed through minibatch stochastic gradient descent. To ensure unbiased gradient estimates of $\hat{\mathcal{L}}_{\text{MIM}}$ we use the reparameterization trick (Kingma and Welling, 2013; Rezende et al., 2014) when taking expectation with respect to continuous encoder and decoder distributions, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. Reparameterization entails sampling an auxiliary variable $\epsilon \sim p(\epsilon)$, with known $p(\epsilon)$, followed by a deterministic mapping from sample variates to the target random variable, that is $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = g_{\boldsymbol{\theta}}(\epsilon)$ and $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = h_{\boldsymbol{\theta}}(\epsilon, \boldsymbol{x})$ for prior and conditional distributions. In doing so we assume $p(\epsilon)$ is independent of the parameters $\boldsymbol{\theta}$. It then follows that

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[f_{\boldsymbol{\theta}}(\boldsymbol{z})\right] \;=\; \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[f_{\boldsymbol{\theta}}(h_{\boldsymbol{\theta}}(\epsilon, \boldsymbol{x}))\right] \;=\; \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(h_{\boldsymbol{\theta}}(\epsilon, \boldsymbol{x}))\right]$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ is the loss function with parameters $\boldsymbol{\theta}$. It is common to let $p(\epsilon)$ be standard normal, $\epsilon \sim \mathcal{N}(0, 1)$, and for $\boldsymbol{z}|\boldsymbol{x}$ to be Gaussian with mean $\mu_{\boldsymbol{\theta}}(\boldsymbol{x})$ and standard deviation $\sigma_{\boldsymbol{\theta}}(\boldsymbol{x})$, in which case $\boldsymbol{z} = \sigma_{\boldsymbol{\theta}}(\boldsymbol{x})\,\epsilon + \mu_{\boldsymbol{\theta}}(\boldsymbol{x})$. A more generic exact density model can be learned by mapping a known base distribution (*e.g.*, Gaussian) to a target distribution with normalizing flows (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015).

For discrete distributions, e.g., with discrete data, reparameterization is not readily applicable. There exist continuous relaxations that permit reparameterization (e.g., Maddison et al. (2016); Tucker et al. (2017)), but current methods are rather involved, and require adaptation of the objective function or the optimization process. Here we simply use the REINFORCE algorithm (Sutton et al., 1999) for unbiased gradient estimates, as follows

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[f_{\boldsymbol{\theta}}(\boldsymbol{z})\right] \;=\; \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z}) + f_{\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] . \tag{3.26}$$

The derivation for Eqn. (3.26) is as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[f_{\boldsymbol{\theta}}(\boldsymbol{z})\right] = & \int q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z})\, d\boldsymbol{z} + \int f_{\boldsymbol{\theta}}(\boldsymbol{z})\, \nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, d\boldsymbol{z} \\
= & \int q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z})\, d\boldsymbol{z} + \int f_{\boldsymbol{\theta}}(\boldsymbol{z})\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, d\boldsymbol{z} \\
= & \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z}) + f_{\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right]
\end{aligned}$$

for which the step from the first line to the second line makes use of the well-known identity, $\nabla_{\boldsymbol{\theta}}\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. This relation is essential as it enables a Monte Carlo approximation to the integral.

3.6.4 Training Time

Training times of MIM models are comparable to training times for VAEs with comparable architectures. One important difference concerns the time required for sampling from the decoder during training. This is particularly significant for models like auto-regressive decoders (*e.g.*, Kingma et al. (2016)) for which sampling is very slow. In such cases, we find that we can also learn effectively with A-MIM, a sampling distribution that only includes samples from the encoding distribution, i.e., $\mathcal{P}(\boldsymbol{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, rather than the mixture. We use it in Sec. 4.3 when working with the PixelHVAE architecture (Kingma et al., 2016).

Chapter 4

MIM: Experiments

In what follows we examine MIM empirically, with the VAE as a baseline. We consider synthetic datasets and well-known image datasets, namely MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017) and Omniglot (Lake et al., 2015). All models were trained using Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-3}, and a mini-batch size of 128. Following Alemi et al. (2017), we anneal the loss to stabilize the optimization. To this end we linearly increase β from 0 to 1 in the following expression for a number of 'warm-up' epochs:

$$\hat{\mathcal{L}}_{\text{MIM}}\left(\boldsymbol{\theta}; \{\boldsymbol{x}_i, \boldsymbol{z}_i\}_{i=1}^N, \beta\right) \;=\; -\frac{1}{2N}\sum_{i=1}^{N} \log\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)\, p_{\boldsymbol{\theta}}(\boldsymbol{z}_i)\right) + \beta \log\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}_i|\boldsymbol{x}_i) \log q_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right) \;.$$

Training continues until the loss (*i.e.*, with $\beta = 1$) on a held-out validation set has not improved for the same number of epochs as the warm-up steps (*i.e.*, defined per experiment). The annealing procedure improved NLL results for the architectures we used here. We have found the number of epochs to convergence of MIM learning to be between 2 to 5 times greater than a VAE with the same architecture. We point the reader to Appendix B for experiments on variants of MIM and VAE that tease apart the impact of specific terms of the respective objectives. (Code is available from `https://github.com/seraphlabs-ca/MIM`).

4.1 Relation to Low Mutual Information in VAE

Before turning to empirical results, it is useful to briefly discuss similarities and differences between MIM and the canonical VAE formulation. To that end, one can show from Eqns. (3.5) and (3.6) that the VAE loss can be expressed in a form that bears similarity to the MIM loss in Eqn. (3.13). In particular, following the derivation in Section 5.2,

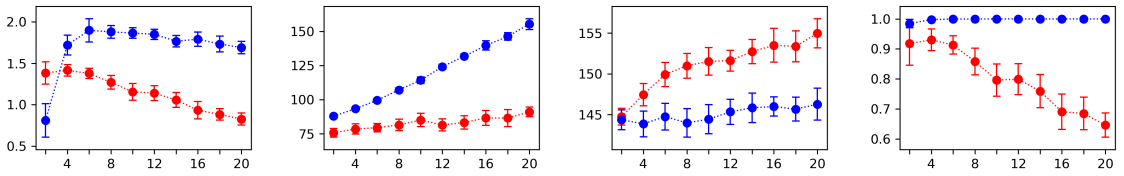
$$\begin{aligned}\mathcal{L}_{\text{VAE}} \;=\; \frac{1}{2}\Big(CE\left(\,\mathcal{M}_{\mathcal{S}}^{q}\,,\; q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\right) + CE\left(\,\mathcal{M}_{\mathcal{S}}^{q}\,,\; p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\right)\Big) \\ - H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{x}\,\right) - H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{z}\,\right) + I_{\mathcal{M}_{\mathcal{S}}^{q}}(\boldsymbol{x};\boldsymbol{z})\;.\end{aligned} \tag{4.1}$$

where $\mathcal{M}_{\mathcal{S}}^{q}(\boldsymbol{x},\boldsymbol{z}) \;=\; q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$. Like the MIM loss, the first term in Eqn. (4.1) in the average of two cross entropy terms, between a sample distribution and the encoding and decoding distributions. Unlike the MIM loss, these terms are asymmetric as the samples are drawn only from the encoding distribution. Also unlike the MIM loss, the VAE loss includes the last three terms in Eqn. (4.1), the sum of which comprise the negative joint entropy $-H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{z},\boldsymbol{x}\,\right)$ under the sample distribution $\mathcal{M}_{\mathcal{S}}^{q}$.

While the MIM objective explicitly encourages high mutual information between observations and corresponding latent embeddings, this VAE loss includes a term that encourages a reduction in the mutual information. We posit that this plays a significant role in the phenomena often referred to as posterior collapse, in which the variance of the variational posterior grows large and the latent embedding conveys relatively little information about the observations (e.g., see Chen et al. (2016b) and others).

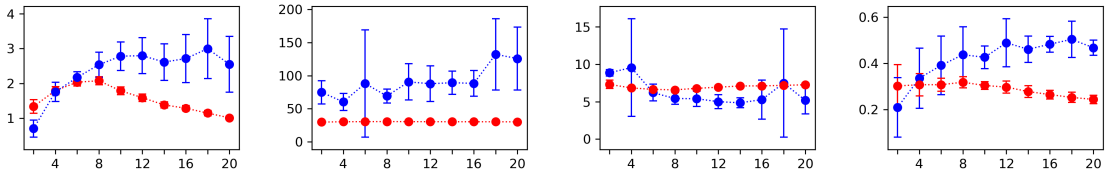
4.2 Low Dimensional Data

To empirically support the expression in Eqn. (4.1), we begin with synthetic data comprising 20D observations $\boldsymbol{x} \in \mathbb{R}^{20}$, with latent dimensionalities between 2 and 20, $\boldsymbol{z} \in \left\{\mathbb{R}^{d} | d \in [2, 20]\right\}$. In low dimensional data one can easily visualize the model and measure quantitative properties of interest (*e.g.*, mutual information), as discussed by Belghazi et al. (2018). This also ensures



(a) MI (b) NLL (c) Recon. Error (d) Classif. (5-NN)

Figure 4.1: Test performance for MIM (blue) and VAE (red) for 20D GMM data (*i.e.*, $\boldsymbol{x} \in \mathbb{R}^{20}$), all as function of the latent dimensionality, from 2 to 20 (on x-axis). Plots depict mean and standard deviation of 10 experiments. MIM learning produces higher mutual information and classification accuracy, with lower test reconstruction error, while VAE yields better data log likelihoods. The VAE suffers from increased collapse as the latent dimensionality grows.



(a) MI (b) NLL (c) Recon. Error (d) Classif. (5-NN)

Figure 4.2: Test performance for MIM (blue) and VAE (red) for 20D Fashion-MNIST, with latent dimension between 2 and 20. Plots depict mean and standard deviation of 10 experiments. MIM opts for better mutual information, and yields better K-NN classification accuracy, at the expense of worse test log likelihood scores.

that the distribution is well modeled with a relatively simple architecture.

In this experiment, observations are drawn from anchor $\mathcal{P}(\boldsymbol{x})$, a Gaussian mixture model with five isotropic components with standard deviation 0.25. The latent anchor $\mathcal{P}(\boldsymbol{z})$ is an isotropic standard Normal. The encoder and decoder are Gaussian conditional distributions, the means and variances of which are regressed from the input using two fully connected layers and *tanh* activation. The parameterized data prior, $q_{\boldsymbol{\theta}}(\boldsymbol{x})$, is defined to be the marginal of the decoding distribution (3.25), and the model prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is defined to be $\mathcal{P}(\boldsymbol{z})$, so the only model parameters are those of the encoder and decoder. We can thus learn models with MIM and VAE objectives, but with the same architecture and parameterization. We used a warm-up scheduler (Vaswani et al., 2017) for the learning rate, with a warm-up of 3 steps, and with each epoch comprising 10000 samples. Training and test sets are drawn

independently from the GMM.

Fig. 4.1 shows results of mutual information, the average negative log-likelihood (NLL) of test points under the model $q_{\boldsymbol{\theta}}$, the mean reconstruction error of test points, and 5-NN classification performance[1] (predicting which of 5 GMM components each test point was drawn from). The auxiliary classification task provides a proxy for representation quality. Following Belghazi et al. (2018), we estimate mutual information using the KSG estimator (Kraskov et al., 2004; Gao et al., 2016), based on 5-NN neighborhoods. MIM produces higher mutual information and better classification as the latent dimensionality increases. VAE mutual information and classification accuracy deteriorate with increasing latent dimensionality, due to stronger posterior collapse for higher dimensional latent space. The test NLL scores for MIM are not as good as those for VAEs in part because the MIM encoder produces very small posterior variance, approaching a deterministic encoder. Nevertheless, MIM produces lower test reconstruction errors.

To further investigate MIM learning in low dimensional data, we project 784D images from Fashion-MNIST onto a 20D linear subspace using PCA (capturing 78.5% of total variance), and repeat the experiment in Fig. 4.1. The training and validation sets had 50,000 and 10,000 images respectively. We trained for 200 epochs, well past convergence, and then selected the model with the lowest validation loss. Fig. 4.2 summarizes the results, with MIM producing high mutual information and classification accuracy, at all but very low latent dimensions. MIM and VAE yield similar test reconstruction errors, with VAE having better negative log likelihoods for test data.

We conclude that the VAE is prone to posterior collapse for a wide range of models' expressiveness and latent dimensionality, with latent embeddings exhibiting low mutual information. In contrast, MIM was empirically robust to posterior collapse, and showed higher mutual information, converging to an encoder with small variance. As a result the learned marginal data likelihood for MIM is worse. In this regard, we note that several papers have described ways to mitigate posterior collapse in VAE learning, e.g., by lower bounding,

[1]We experimented with 1-NN,3-NN,5-NN,10-NN and found the results to be consistent.

	convHVAE (S)		convHVAE (VP)	
Dataset	MIM	VAE	MIM	VAE
Fashion-MNIST	272.14 ± 0.64	$\mathbf{225.40 \pm 0.05}$	227.61 ± 0.34	$\mathbf{224.77 \pm 0.04}$
MNIST	126.85 ± 0.56	$\mathbf{80.50 \pm 0.05}$	82.73 ± 0.08	$\mathbf{79.66 \pm 0.06}$
Omniglot	141.81 ± 0.32	$\mathbf{97.94 \pm 0.29}$	104.10 ± 2.17	$\mathbf{97.52 \pm 0.16}$
	PixelHVAE (S)		PixelHVAE (VP)	
	A-MIM	VAE	A-MIM	VAE
Fashion-MNIST	243.95 ± 0.47	$\mathbf{224.65 \pm 0.07}$	224.94 ± 0.34	$\mathbf{224.02 \pm 0.08}$
MNIST	114.96 ± 0.35	$\mathbf{79.04 \pm 0.05}$	79.04 ± 0.08	$\mathbf{78.60 \pm 0.04}$
Omniglot	126.12 ± 0.38	$\mathbf{91.06 \pm 0.14}$	91.82 ± 0.20	$\mathbf{90.74 \pm 0.15}$

Table 4.1: Test NLL (in nats) for high dimensional image data. Quantitative results based on 10 trials per condition. With a more powerful prior, MIM and VAE yield comparable results.

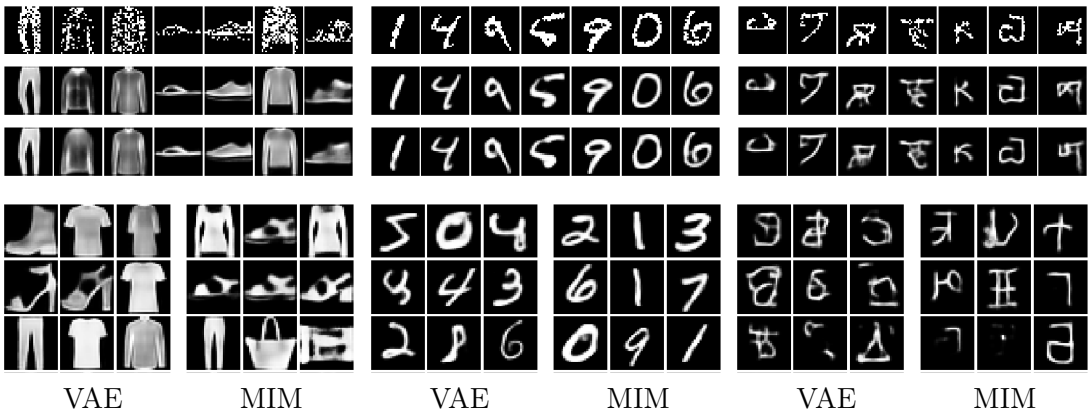
or annealing the KL divergence term in the VAE objective (Alemi et al., 2017; Razavi et al., 2019), or by limiting the expressiveness of the decoder (e.g., Chen et al. (2016b)). We posit that MIM does not suffer from this problem as a consequence of the objective design principles that encourage high mutual information between observations and the latent representation.

4.3 High Dimensional Image Data

We next consider MIM and VAE learning with image data (Fashion-MNIST, MNIST, Omniglot). Unfortunately, with high dimensional data we cannot reliably compute mutual information (Belghazi et al., 2018). Instead, for model assessment we focus on negative log-likelihood, reconstruction, and the quality of random samples. In doing so we also explore multiple architectures, including the top performing models from Tomczak and Welling (2017), namely, *convHVAE* (L = 2) and *PixelHVAE* (L = 2) models, with Standard (S) priors[2], and *VampPrior* (VP) priors[3]. The VP pseudo-inputs are initialized with training data samples. All the experiments below use the same experimental setup as in Tomczak

[2] $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mu = \mathbf{0}, \sigma = \mathbb{I})$, a standard Normal distribution, where $\mathbb{I}$ is the identity matrix.

[3] $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \frac{1}{K} \sum_{k=1}^{K} q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{u}_k)$, a mixture model of the encoder conditioned on optimized pseudo-inputs $\boldsymbol{u}_k$.



(a) Fashion MNIST (b) MNIST (c) Omniglot

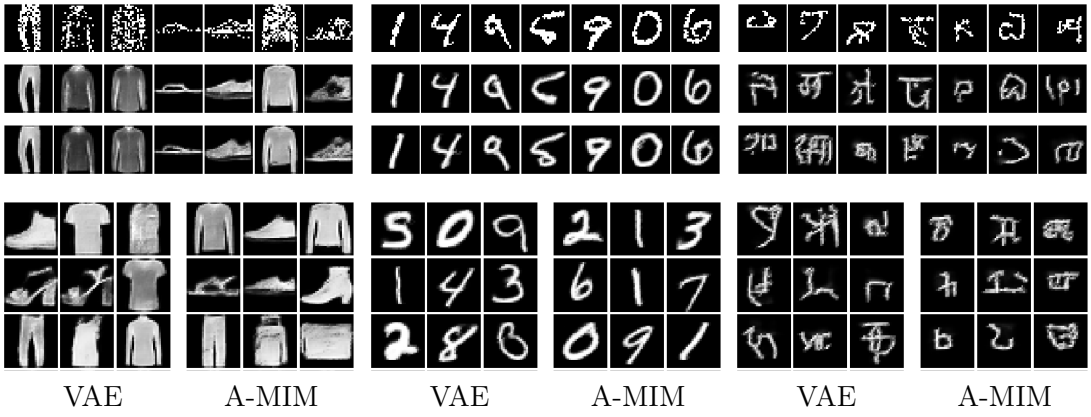
Figure 4.3: MIM and VAE learning with the convHVAE (VP) architecture, applied to Fashion-MNIST, MNIST, and Omniglot (left to right). The top three rows (from top to bottom) are test data samples, VAE reconstruction, and MIM reconstruction. Bottom: random samples from VAE and MIM. With a powerful enough prior, MIM offers samples which are comparable to VAE.

and Welling (2017), and the same latent dimensionality $\boldsymbol{z} \in \mathbb{R}^{80}$. Here we also demonstrate that a powerful prior (*e.g.*, PixelHVAE (VP)) allows MIM to learn models with competitive sampling and NLL performance.

Sampling from an auto-regressive decoder (*e.g.*, PixelHVAE) is very slow. To reduce training time, as discussed above in Sec. 3.6.4, we learn with a sampling distribution comprising just the encoding distribution, i.e., $\mathcal{P}(\boldsymbol{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, rather than the mixture, a MIM variant we refer to as asymmetric-MIM (or A-MIM).

Table 4.1 reports test NLL scores. One can see that VAE models yield better NLL, but with a small gap for more expressive models (*i.e.*, PixelHVAE (VP)). We also show qualitative results in Figures (4.3, 4.4) for the most expressive models (*i.e.*, convHVAE (VP), PixelHVAE (VP) respectively). Each figure depicts reconstruction[4] and sampling for Fashion-MNIST, MNIST, and Omniglot, with MIM and VAE being comparable. The top three rows depict data samples, VAE reconstructions, and MIM reconstructions, respectively.

[4]Test data in the top row of Figures (4.3, 4.4) are binary, while reconstructions depict the probability of each pixel being 1, following Tomczak and Welling (2017).



(a) Fashion MNIST (b) MNIST (c) Omniglot

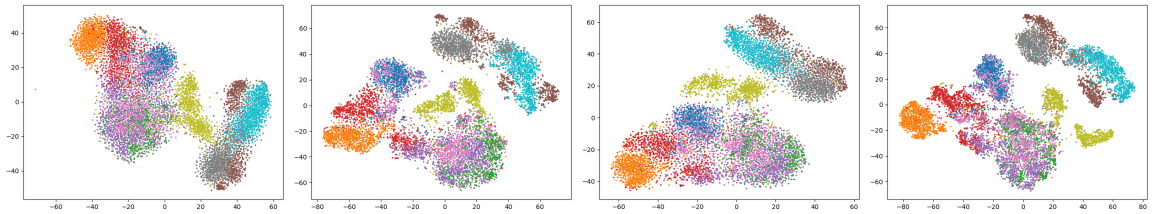
Figure 4.4: MIM and VAE learning with the PixelHVAE (VP) architecture, applied to Fashion-MNIST, MNIST, and Omniglot (left to right). MIM was trained with asymmetric sampling (*i.e.*, from encoding distribution only), and as such is labelled A-MIM. The top three rows (from top to bottom) are test data samples, VAE reconstruction, and MIM reconstruction. Bottom: random samples from VAE and MIM. With a powerful enough prior, MIM offers samples which are comparable to VAE.

The bottom row depicts random samples. Note that, while MIM with a weak prior (Standard) suffers from poor sampling, increasing the expressiveness results in comparable samples and reconstruction. See Appendix C for additional results.

The poor NLL and hence poor sampling for MIM with a weak prior model can be explained by the tightly clustered latent representation (*e.g.*, Fig. 5.1). A more expressive, learnable prior can capture such clusters more accurately, and as such, also produces good samples (*e.g.*, VampPrior). In other words, while VAE opts for better NLL and sampling at the expense of lower mutual information, MIM provides higher mutual information at the expense of the NLL for a weak prior, and comparable NLL and sampling with more expressive priors. In Sec. 4.4 we probe the effect of higher mutual information on the quality of the learned representation.

	convHVAE (S)		convHVAE (VP)		PixelHVAE (S)		PixelHVAE (VP)	
Dataset	MIM	VAE	MIM	VAE	A-MIM	VAE	A-MIM	VAE
Fashion-MNIST	**0.83**	0.76	**0.81**	0.78	0.71	**0.76**	**0.79**	0.77
MNIST	**0.97**	0.92	**0.97**	0.92	**0.95**	0.86	**0.96**	0.81

Table 4.2: Test accuracy of 5-NN classifier for High dimensional image data. Quantitative results based on 10 trials per condition. Standard deviations are less than 0.01, and omitted from the table. MIM offers better unsupervised clustering of classes in the latent representation in all experiments but one.



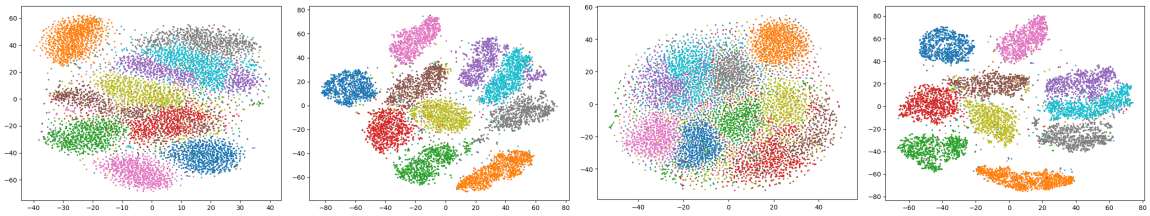
(a) VAE (S) (b) MIM (S) (c) VAE (VP) (d) MIM (VP)

Figure 4.5: MIM and VAE $\boldsymbol{z}$ embedding for Fashion MNIST with convHVAE architecture. MIM shows stronger disentanglement of classes.

4.4 Clustering and Classification

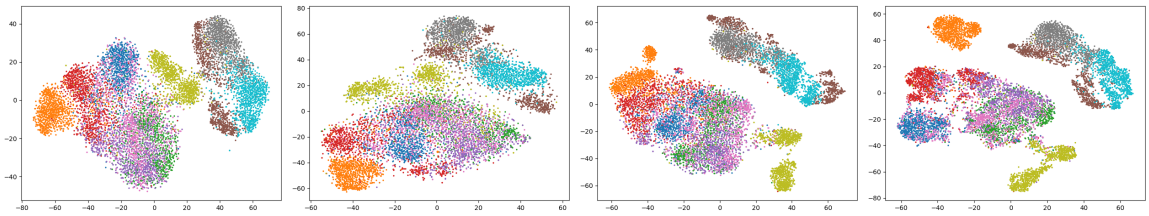
Finally, following Hjelm et al. (2019), we consider an auxiliary classification task as a further measure of the quality of the learned representations. We opted for K-NN classification, being a non-parametric method which relies on semantic clustering in latent space without any additional training. Given representations learned above in Sec. 4.3, a simple 5-NN classifier[5] was applied to test data to predict one of 10 classes for MNIST and Fashion-MNIST. Table 4.2 shows that in all but one case, MIM yields more accurate classification results. We attribute the performance difference to higher mutual information of MIM representations, combined with low entropy of the marginals. Figures (4.5, 4.6, 4.7, and 4.8) provide a qualitative visualization of the latent clustering, for which t-SNE (van der Maaten and Hinton, 2008)) was used to project the latent space down to 2D for Fashion-MNIST, and MNIST data. One can see that MIM learning tends to cluster classes in the latent representation more tightly,

[5]We omitted results for $k \in \{1, 3, 10\}$ as we find them similar.



(a) VAE (S) (b) MIM (S) (c) VAE (VP) (d) MIM (VP)

Figure 4.6: MIM and VAE $\boldsymbol{z}$ embedding for MNIST with convHVAE architecture. MIM shows stronger disentanglement of classes.



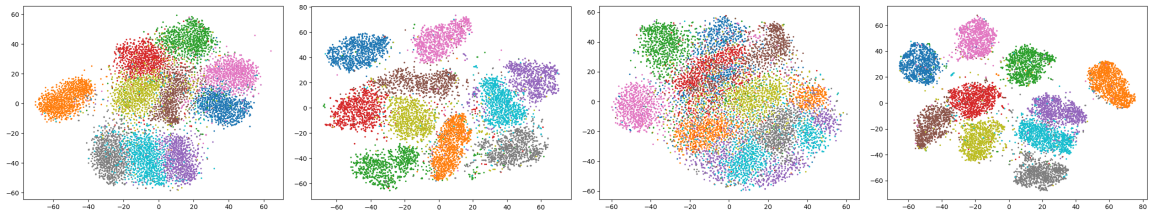
(a) VAE (S) (b) A-MIM (S) (c) VAE (VP) (d) A-MIM (VP)

Figure 4.7: A-MIM and VAE $\boldsymbol{z}$ embedding for Fashion MNIST with PixelHVAE architecture. MIM shows stronger disentanglement of classes.

while VAE clusters are more diffuse and overlapping, consistent with the results in Table 4.2.

4.5 Conclusions

We introduce a new representation learning framework, named the *mutual information machine* (MIM), that defines a latent generative model built around three principles: consistency between the encoding and decoding distributions, high mutual information between the observed and latent variables, and low marginal entropy. This yields a learning objective that can be directly optimized by stochastic gradient descent with reparameterization, as opposed to adversarial learning used in related frameworks. We show that the MIM framework produces highly informative representations and is robust against posterior collapse. Most importantly, MIM greatly benefits from additional capacity; as the encoder and decoder become larger and more expressive, MIM retains its highly informative representation, and consistently shows improvement in its modelling of the underlying distribution as measured by negative log-likelihood. The stability, robustness, and ease of training make MIM a



(a) VAE (S) (b) A-MIM (S) (c) VAE (VP) (d) A-MIM (VP)

Figure 4.8: A-MIM and VAE $\boldsymbol{z}$ embedding for MNIST with PixelHVAE architecture. MIM shows stronger disentanglement of classes.

compelling framework for latent variable modelling across a variety of domains.

In future work, we intend to focus on utilizing the high mutual information mapping provided by MIM, by exploiting the clustered latent representation to further improve the resulting generative model.

Chapter 5

Posterior Collapse

5.1 Introduction

VAE is widely used in representation learning (Kingma and Welling, 2013; Rezende et al., 2014; Rezende and Mohamed, 2015). While often producing excellent representations, VAEs sometimes produce pathological results in which the encoder, or approximate posterior, conveys relatively little information between observations and latent states. This poses an issue when dealing with images, for instance, where reconstructed and sampled images are blurry, which is a known issue with VAEs.

This behavior, often referred to as *posterior collapse*, is observed by low mutual information between observations and inferred latent states, near zero KL divergence term (in some of the dimensions), and a large variance of the posterior (Bowman et al., 2015; Chen et al., 2016b; Razavi et al., 2019; van den Oord et al., 2016, 2017b). Posterior collapse is especially noticeable when the decoder becomes powerful (Chen et al., 2016b), which poses a significant challenge when dealing with high dimensional data (Tomczak and Welling, 2017), sequential data (Bowman et al., 2015), or expressive autoregressive decoders (Chen et al., 2016b). When using VAE for representation learning, posterior collapse might lead to learned latent codes which are less informative than one might desire, and as such poses a significant problem when trying to utilize the learned representation in downstream tasks.

Methods to fix posterior collapse commonly focus on optimization heuristics in order to push the learned model into a local optimum with reduced collapse. Bowman et al. (2015) proposed annealing of the KL divergence term (*i.e.*, in ELBO) to encourage the trained model to learn a representation with high mutual information (*i.e.*, the model is initially trained as an auto-encoder). He et al. (2019) encourage high mutual information by altering the training dynamics to initially train an auto-encoder (*i.e.*, "aggressive" step) until the mutual information is saturated, followed by the usual ELBO.

The method by He et al. (2019) is indeed effective in finding an local optimum with high mutual information when compared to KL annealing techniques. However, we point to an observed drop in KL divergence and mutual information once the ELBO loss is optimized. This observation empirically supports the notion that the ELBO loss in itself might encourage lower mutual information (*i.e.*, when compared to an auto-encoder with similar parametrization; or $\beta = 0$ for KL-annealing in VAE).

Alternatively, posterior collapse in VAE can be mitigated by means of model architecture design. Bowman et al. (2015) experiment with weakening the decoder by adding noise to the autoregressive conditioning process (*i.e.*, randomly replace a fraction of the conditioned-on input to the decoder). This forces the decoder to rely on the latent rather than the previous word, reducing the collapse on the expense of worse NLL. Chen et al. (2016b) propose an architecture for a decoder to limit the model to be autoregressive over a local window. Interesting, we point that Chen et al. (2016b) acknowledge that the proposed technique will fail if the true posterior is in fact locally-autoregressive, and as such recommend to use optimization heuristics (*e.g.*, KL-annealing) in addition to the proposed method.

As an alternative to restricting the decoder, Razavi et al. (2019) propose to model the prior over the latent as a first order autoregressive process (*i.e.*, over optimization steps). In effect creating a moving target for the posterior to match, and a KL divergence term which is lower bounded. Such a method, while effective in preventing collapse, also requires to learn a parametric prior around the learned posterior (*i.e.*, in a second optimization stage) in order to provide good sampling. The issue of learning a good prior is also addressed by Tomczak

and Welling (2017), which uses the posterior conditioned on parametric pseudo-inputs as a prior (*i.e.*, VampPrior), and when combined with aggressive KL annealing, provides good samples with mitigated posterior collapse.

It is also worth mentioning Stochastic Variational Inference (SVI, Hoffman et al. (2013)), in which the parameters of a mean-field variational family are learned, with instance specific local latent and a shared global latent variables. Training SVI entails per-observation iterative inference, and as such does not scale well with large datasets. Probability density estimation with SVI, however, has been demonstrated to be less susceptible to the phenomenon of posterior collapse, and recent methods (*e.g.*, Hjelm et al. (2016); Kim et al. (2018)) combine amortized VI (AVI) with SVI to avoid posterior collapse.

More specifically, Kim et al. (2018) propose to initialize each instance-specific SVI step with an amortized inference network (*i.e.*, the encoder). While showing good results, the method is rather involved, and suffers from instabilities in optimization. In addition, SVI is instance specific, and as such does not generalize, and requires optimization in inference time. When compared to AVI which involves only a forward pass of the encoder, SVI might potentially suffer from slow inference.

Here we note that the research community focus on posterior collapse is mostly limited to various optimization and model design heuristics. Unfortunately, this view ignores the possibility that alternative estimators (*i.e.*, to VAE) might avoid such challenges. Instead, we advocate for a novel parametric density estimator (*i.e.*, MIM in Section 3.5) which is trained with a symmetric amortized variational inference. The proposed estimator does not suffer from posterior collapse, and does not require elaborate optimization heuristics. As a consequence, we hope that MIM will ease the training of powerful encoder-decoder models.

5.2 Posterior Collapse in VAE

Mutual information is a key quantity in representation learning. Here we show the relation between VAE and MIM, where VAE loss implicitly encourages lower mutual information.

This is in contrast with MIM loss, which encourages high mutual information. More explicitly, we postulate that a root cause for the observed phenomena of posterior collapse in VAE is low mutual information, similar to Zhao et al. (2018b). We show that VAE learning can be viewed as asymmetric MIM learning (*i.e.*, A-MIM) with a regularizer that encourages the appearance of the collapse. We further support that idea in the experiments in Section 5.3.2.

As discussed earlier, VAE learning entails maximization of a variational lower bound (ELBO) on the log-marginal likelihood, or equivalently, given Eqn. (3.5), the VAE loss in terms of expectation over a joint distribution:

$$-\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}(\boldsymbol{x}),\boldsymbol{z}\sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log\mathcal{P}(\boldsymbol{z}) - \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \ . \tag{5.1}$$

To connect the loss in Eqn. (5.1) to MIM, we first add the expectation of $\log\mathcal{P}(\boldsymbol{x})$, and scale the loss by a factor of $\frac{1}{2}$, to obtain

$$\mathbb{E}_{\mathcal{P}(\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[-\frac{1}{2}\left(\log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})) + \log(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x}))\right) + \log\mathcal{P}(\boldsymbol{x}) + \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \tag{5.2}$$

where $\mathcal{P}(\boldsymbol{x})$ is the data distribution, which is assumed to be independent of model parameters θ and to exist almost everywhere (*i.e.*, complementing $\mathcal{P}(\boldsymbol{z})$). Importantly, because $\mathcal{P}(\boldsymbol{x})$ does not depend on $\boldsymbol{\theta}$, the gradients of Eqs. (5.1) and (5.2) are identical up to a multiple of $\frac{1}{2}$, so they share the same stationary points.

Combining IID samples from the data distribution, $\boldsymbol{x}^i \sim \mathcal{P}(\boldsymbol{x})$, with samples from the corresponding variational posterior, $\boldsymbol{z}^i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^i)$, we obtain a joint sampling distribution; *i.e.*,

$$\mathcal{M}^q_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z}) \ = \ \mathcal{P}(\boldsymbol{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$$

where $\mathcal{M}^q_{\mathcal{S}}$ comprises the encoding distribution in $\mathcal{M}_{\mathcal{S}}$. With it one can then rewrite the objective in Eqn. (5.2) in terms of the cross-entropy between $\mathcal{M}^q_{\mathcal{S}}$ and the parametric encoding

and decoding distributions; *i.e.*,

$$\frac{1}{2}\big(CE\left(\mathcal{M}_{\mathcal{S}}^{q},\ p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \right)) + CE\left(\mathcal{M}_{\mathcal{S}}^{q},\ q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \right)\ \big) + \\ - H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{x}\,\right) - H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{z}\,\right) + I_{\mathcal{M}_{\mathcal{S}}^{q}}(\boldsymbol{x};\boldsymbol{z}) \tag{5.3}$$

where $CE\left(\,\cdot\,,\,\cdot\,\right)$ denotes cross-entropy, $H\left(\,\cdot\,\right)$ entropy, and $I(\boldsymbol{x};\boldsymbol{z})$ mutual information. The sum of the last three terms in Eqn. (5.3) is the negative joint entropy $-H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{z},\boldsymbol{x}\,\right)$ under the sample distribution $\mathcal{M}_{\mathcal{S}}^{q}$.

Equations (5.1) and (5.3), the VAE objective and VAE as regularized cross entropy objective respectively, define equivalent optimization problems, under the assumption that $\mathcal{P}(\boldsymbol{x})$ and samples $\boldsymbol{x} \sim \mathcal{P}(\boldsymbol{x})$ do not depend on the parameters $\boldsymbol{\theta}$, and that the optimization is gradient-based. Formally, the VAE objectives (5.1) and (5.3) are equivalent up to a scalar multiple of $\frac{1}{2}$ and an additive constant, namely, $H_{\mathcal{M}_{\mathcal{S}}^{q}}\left(\,\boldsymbol{x}\,\right)$.

Equation (5.3) is the average of two cross-entropy objectives (*i.e.*, between sample distribution $\mathcal{M}_{\mathcal{S}}^{q}$ and the model decoding and encoding distributions, respectively), along with a joint entropy term (*i.e.*, last three terms), which can be viewed as a regularizer that encourages a reduction in mutual information and increased entropy in $\boldsymbol{z}$ and $\boldsymbol{x}$. We note that Eqn. (5.3) is similar to the A-MIM objective in Eqn. (3.19),

$$\frac{1}{2}\big(CE\left(\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,q_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\,\right) + CE\left(\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,p_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\,\right)\ \big)$$

with an additional regularizer.

Importantly, Eqn. (5.3) comprises an asymmetric sample distribution, where the priors are defined to be the anchors, and with an additional regularizer. In other words, Eqn. (5.3) suggests that VAE learning implicitly lowers mutual information. This often runs contrary to the goal of learning useful latent representations, and we posit that it is an underlying root cause for *posterior collapse*, for which the trained model has low mutual information, manifested in the encoder having high posterior variance (e.g., see (Chen et al., 2016b) and

others). We point the reader to Section 5.3.2 for empirical evidence for the use of a joint entropy as a mutual information regularizer.

5.3 Experiments

In what follows we experiment with VAE and MIM learning in order to demonstrate various aspects of posterior collapse in VAE, and to show that MIM does not suffer from it. We probe multiple experimental settings, which allows us to empirically support the notion that posterior collapse is the result of the VAE formulation.

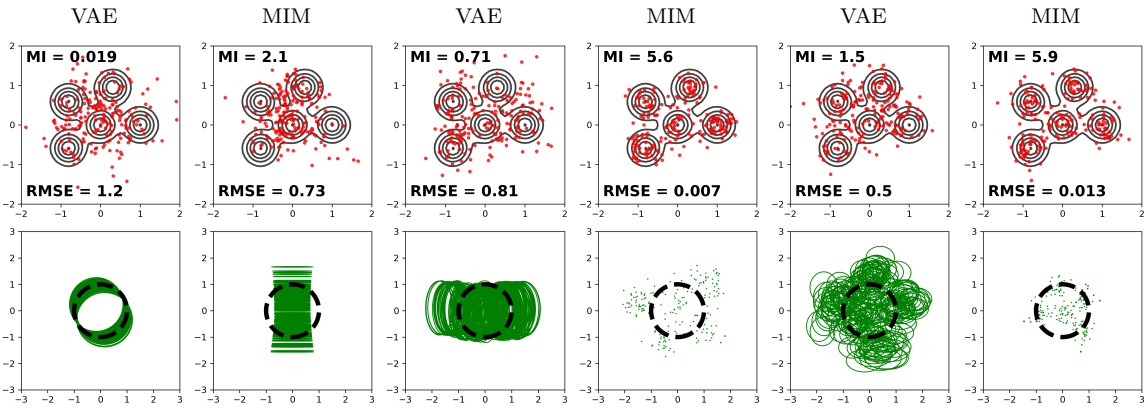
5.3.1 Visualization of Posterior Collapse in 2D Data

To empirically support the notion of a low mutual information regularizer in VAE, we begin with synthetic data comprising 2D observations $\boldsymbol{x} \in \mathbb{R}^2$, with a 2D latent space, $\boldsymbol{z} \in \mathbb{R}^2$. In 2D one can easily visualize the model and measure quantitative properties of interest (*e.g.*, mutual information). Observations are drawn from anchor $\mathcal{P}(\boldsymbol{x})$, a Gaussian mixture model with five isotropic components with standard deviation 0.25 (Fig. 5.1, top row). The latent anchor $\mathcal{P}(\boldsymbol{z})$ is an isotropic standard Normal (Fig. 5.1, bottom row).

The encoder and decoder conditional distributions are Gaussian, the means and variances of which are regressed from the input using two fully connected layers and *tanh* activation. The parameterized data prior, $q_{\boldsymbol{\theta}}(\boldsymbol{x})$, is defined to be the marginal of the decoding distribution (3.25), and the model prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is defined to be $\mathcal{P}(\boldsymbol{z})$, so the only model parameters are those of the encoder and decoder. We can thus learn models with MIM and VAE objectives, but with the same architecture and parameterization. We used a warm-up scheduler (Vaswani et al., 2017) for the learning rate, with a warm-up of 3 steps, and with each epoch comprising 10000 samples. Training and test sets are drawn independently from the GMM.

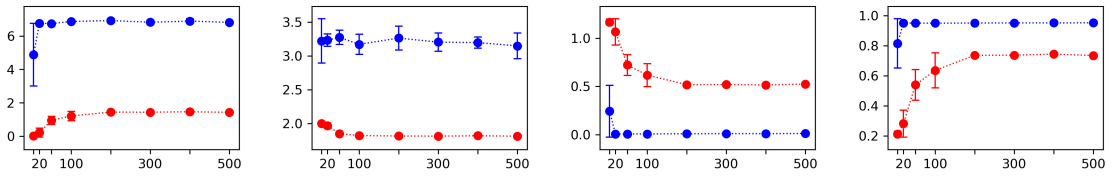
Figure 5.1 depicts three models for VAE (odd columns) and MIM (even columns), with increasing numbers of hidden units (from left to right) to control model expressiveness. The top row depicts observation space where black contours are levels sets of constant density

$\mathcal{P}(\boldsymbol{x})$, and red points are reconstructed samples, i.e., one point drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}')$ where $\boldsymbol{z}'$ is drawn from the encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}'|\boldsymbol{x}')$, given a test point $\boldsymbol{x}'$ from $\mathcal{P}(\boldsymbol{x})$. In each case we also report the mutual information and the root-mean-squared reconstruction error, with MIM producing superior results.



(a) $h \in \mathbb{R}^5$ (b) $h \in \mathbb{R}^{20}$ (c) $h \in \mathbb{R}^{500}$

Figure 5.1: VAE and MIM models with 2D inputs, a 2D latent space, and 5, 20 and 500 hidden units. Top row: Black contours depict level sets of $\mathcal{P}(\boldsymbol{x})$; red dots are reconstructed test points. Bottom row: Green contours are one standard deviation ellipses of $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ for test points. Dashed black circles depict one standard deviation of $\mathcal{P}(\boldsymbol{z})$. The VAE predictive variance remains high, regardless of model expressiveness, an indication of various degrees of posterior collapse, while MIM produces lower predictive variance and lower reconstruction errors, consistent with high mutual information (see inset quantities).



(a) MI (b) NLL (c) Recon. Error (d) Classif. (5-NN)

Figure 5.2: Test performance for MIM (blue) and VAE (red) for the 2D GMM data (cf. Fig. 5.1), all as functions of the number of hidden units (on x-axis). Plots show (a) mutual information, (b) negative log-likelihood of test points, (c) test reconstruction error, and (d) K-NN mode classification performance. Each plot shows the mean and standard deviation of 10 experiments.

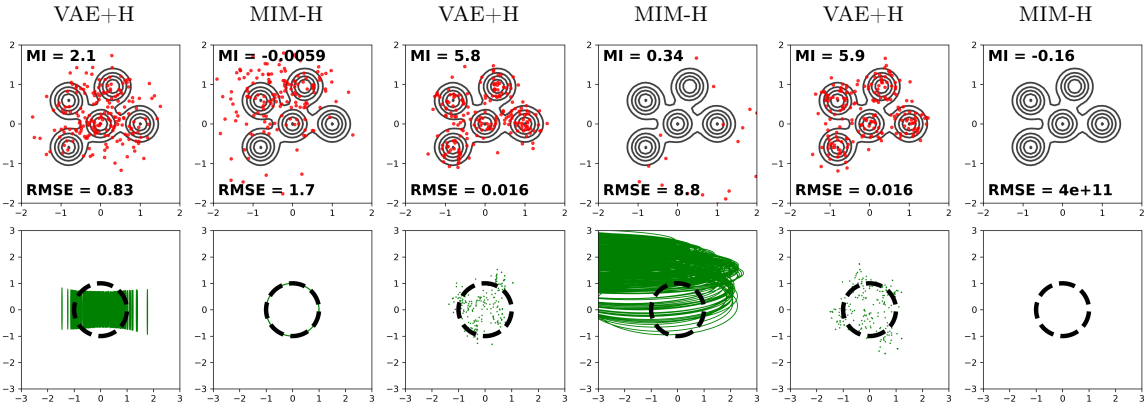
The bottom row of Fig. 5.1 depicts latent space behavior. The dashed black circle depicts one standard deviation of $\mathcal{P}(\boldsymbol{z})$. Each green curve depicts a one standard deviation ellipse of

the encoder posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}'|\boldsymbol{x}')$ given a test point $\boldsymbol{x}'$ from $\mathcal{P}(\boldsymbol{x})$. For the weakest architecture (a), with 5 hidden units, VAE and MIM posterior variances are similar to the prior in one dimension, a sign of posterior collapse. As the number of hidden units increases (b,c), the VAE posterior variance remains large, preferring lower mutual information while matching the aggregated posterior to the prior. In contrast, the MIM encoder produces tight posteriors, and yields higher mutual information and lower reconstruction errors at the expense of somewhat worse data log likelihoods..

To quantify this behavior Fig. 5.2 shows mutual information, the average negative log-likelihood (NLL) of test points under the model $q_{\boldsymbol{\theta}}$, the mean reconstruction error of test points, and 5-NN classification performance[1] (predicting which of 5 GMM components each test point was drawn from). The auxiliary classification task provides a proxy for representation quality. Following Belghazi et al. (2018), we estimate mutual information using the KSG estimator (Kraskov et al., 2004; Gao et al., 2016), based on 5-NN neighborhoods.

Mutual information and classification accuracy for test data under the MIM model are higher than for VAE models. One can also see that mutual information is saturated for MIM, as it effectively learns an (approximate) invertible mapping. The encoder and decoder approach deterministic mappings, reflected in the near-zero reconstruction error. Interestingly, MIM learning finds near-invertible mappings with unconstrained architectures (demonstrated here for the 2D case), when the dimensionality of the latent representation and the observations is the same.

We conclude that the large variance, often observed in VAE, coincides with low mutual information (*i.e.*, compare to the same parameterization trained with MIM learning), and poor reconstruction. In addition, the posterior collapse in VAE is not solved by increasing the model expressiveness. MIM, on the other hand, demonstrated robustness to the effects of posterior collapse when model expressiveness grows.



(a) $h \in \mathbb{R}^5$ (b) $h \in \mathbb{R}^{20}$ (c) $h \in \mathbb{R}^{500}$

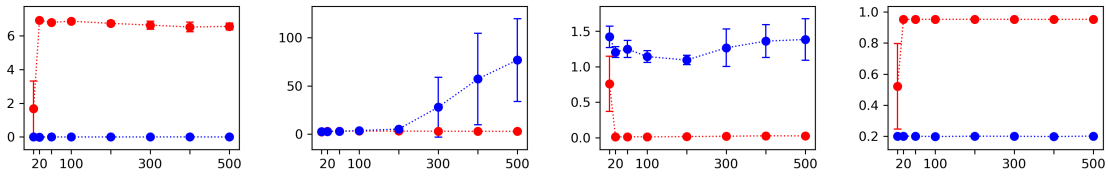
Figure 5.3: Effects of entropy as a mutual information regularizer in 2D $\boldsymbol{x}$ and 2D $\boldsymbol{z}$ synthetic problem. VAE and MIM models with 2D inputs, a 2D latent space, and 5, 20 and 500 hidden units. Top row: Black contours depict level sets of $\mathcal{P}(\boldsymbol{x})$; red dots are reconstructed test points. Bottom row: Green contours are one standard deviation ellipses of $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ for test points. Dashed black circles depict one standard deviation of $\mathcal{P}(\boldsymbol{z})$. Here we added $H_{\mathcal{M}_S^q}(\boldsymbol{x}, \boldsymbol{z})$ to VAE loss, and subtracted $H_{\mathcal{M}_S}(\boldsymbol{x}, \boldsymbol{z})$ from MIM loss, in order to demonstrate the effect of entropy on mutual information. Posterior collapse in VAE is mitigated following the increased mutual information. MIM, on the other hand, demonstrates a severe posterior collapse as a result of the reduced mutual information (*e.g.*, (a) MIM posterior matches prior over $\boldsymbol{z}$ almost perfectly). (see inset quantities).

5.3.2 Entropy as Mutual Information Regularizer

Here we examine the use of joint entropy as a mutual information regularizer. We hypothesize that low mutual information is the reason behind posterior collapse. If indeed the joint entropy regularizer acts as a high mutual information regularizer (and low marginal entropy), adding joint entropy to VAE loss should reduce posterior collapse. We will measure the collapse by measuring the mutual information, and mean reconstruction error. Similarly, subtracting the joint entropy from MIM should induce posterior collapse.

To this end, we repeat the experiments in Section 5.3.1 where we manipulate VAE and MIM loss with entropy regularizer. Here we have a synthetic example with 2D observations, 2D latent, and exact parametric observations distribution $\mathcal{P}(\boldsymbol{x})$. We add the encoding sample

[1]We experimented with 1-NN,3-NN,5-NN,10-NN and found the results to be consistent.



(a) MI (b) NLL (c) Recon. Error (d) Classif. (5-NN)

Figure 5.4: Effects of entropy as a mutual information regularizer in 2D $\boldsymbol{x}$ and 2D $\boldsymbol{z}$ synthetic problem. Test performance for modified MIM (blue) and modified VAE (red) for the 2D GMM data with (cf. Fig. 5.3), all as functions of the number of hidden units (on x-axis). Each plot shows the mean and standard deviation of 10 experiments. Adding encoding entropy regularizer to VAE loss leads to high mutual information (*i.e.*, prevent posterior collapse), low reconstruction error, and better classification accuracy. Subtracting sample entropy regularizer from MIM loss results in almost zero mutual information (severe collapse), which leads to poor reconstruction error and classification accuracy.

distribution $H_{\mathcal{M}_{\mathcal{S}}^{q}}(\boldsymbol{x}, \boldsymbol{z})$ to the VAE loss, where

$$\mathcal{M}_{\mathcal{S}}^{q} = q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$$

and subtract $H_{\mathcal{M}_{\mathcal{S}}}(\boldsymbol{x}, \boldsymbol{z})$ from the MIM loss, where

$$\mathcal{M}_{\mathcal{S}} = \frac{1}{2}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) \ .$$

We note that the augmented models do not necessarily represent any probabilistic model. Our goal here is to observe the effect of the joint entropy on the learned representation.

Figure 5.3 depicts the effects of an added $H_{\mathcal{M}_{\mathcal{S}}^{q}}(\boldsymbol{x}, \boldsymbol{z})$ to the VAE loss, and a subtracted $H_{\mathcal{M}_{\mathcal{S}}}(\boldsymbol{x}, \boldsymbol{z})$ from MIM . The corresponding quantitative values are presented in Figure 5.4. Adding the entropy regularizer to VAE loss leads to increased mutual information, low NLL, and low mean reconstruction error. Subtracting the entropy regularizer from MIM loss results in a strong posterior collapse, poor NLL, which in turn is reflected in poor reconstruction error. We conclude that the presented empirical evidence supports the use of entropy as a regularizer (cf. Eq. (3.9)) to JSD in order to define a consistent model with high mutual information.

5.3.3 Posterior Collapse in High Dimensional Image Data

	convHVAE (S)		convHVAE (VP)	
Dataset	MIM	VAE	MIM	VAE
Fashion-MNIST	272.14 (274.98)	225.40 (225.72)	227.61 (229.2)	224.77 (225.17)
MNIST	126.85 (127.45)	80.50 (80.58)	82.73 (85.62)	79.66 (80.42)
Omniglot	141.81 (143.75)	97.94 (99.5)	104.10 (**100.79**)	97.52 (100.63)
	PixelHVAE (S)		PixelHVAE (VP)	
	A-MIM	VAE	A-MIM	VAE
Fashion-MNIST	243.95 (**243.43**)	224.65 (226.35)	224.94 (227.23)	224.02 (225.66)
MNIST	114.96 (115.09)	79.04 (80.82)	79.04 (82.34)	78.60 (81.87)
Omniglot	126.12 (**125.97**)	91.06 (92.78)	91.82 (93.6)	90.74 (93.4)

Table 5.1: Test NLL (in nats) for high dimensional image data. Quantitative results are based on 10 trials per condition. Here we compare the results of KL-annealing over 100 epochs, to the (results) without any annealing. The standard deviation per experiment is comparable, and is not included here. **Bold** marks same or better results when no annealing was used. Most models performed worse without annealing, with the exception of three MIM models.

Here we explore the effects of KL-annealing on posterior collapse with MIM and VAE learning. We repeat the experiments in Section 4.3, where we used the architecture by Tomczak and Welling (2017) to train VAE and MIM models. We used the experimental settings by Tomczak and Welling (2017), with the exception that here we avoided KL-annealing. We emphasize that Tomczak and Welling (2017) use KL-annealing for 100 epochs, in effect aggressively pushing the VAE into a local optimum around an autoencoder solution.

We show the NLL and classification results in the corresponding Tables (5.1, 5.2), where we compare results of models that were trained with KL-annealing, to models with the same architecture that were trained with no annealing (in parentheses). Table 5.1 shows NLL, where **Bold** marks same or better results when no annealing was used. Most models performed worse without annealing, with the exception of three MIM models. This might suggest that the architecture of the models benefit from the KL-annealing as an optimization heuristics which focuses first on reconstruction before learning a corresponding variance.

Table 5.2 show 5-NN classification results, where **Bold** marks 1% accuracy drop or

	convHVAE (S)		convHVAE (VP)	
Dataset	MIM	VAE	MIM	VAE
Fashion-MNIST	0.83 (**0.83**)	0.76 (**0.75**)	0.81 (**0.8**)	0.78 (**0.8**)
MNIST	0.97 (**0.97**)	0.92 (**0.91**)	0.97 (**0.96**)	0.92 (**0.96**)
	PixelHVAE (S)		PixelHVAE (VP)	
	A-MIM	VAE	A-MIM	VAE
Fashion-MNIST	0.71 (0.69)	0.76 (0.43)	0.79 (0.7)	0.77 (0.61)
MNIST	0.95 (**0.95**)	0.86 (0.47)	0.96 (0.51)	0.81 (0.46)

Table 5.2: Test accuracy of 5-NN classifier for High dimensional image data. Quantitative results based on 10 trials per condition. Here we compare the results of KL-annealing over 100 epochs, to the (results) without any annealing. The standard deviation per experiment is comparable, and is not included here. **Bold** marks 1% accuracy drop or better results when no annealing was used. Models with convHVAE architecture were all comparable, with VAE models with VampPrior benefiting the most from lack of annealing. Models with PixelHVAE architecture were more affected, with VAE suffers significantly more from lack of annealing.

better results when no annealing was used. Models with convHVAE architecture were all comparable, where VAE models with VampPrior benefiting the most from lack of annealing. The results are counter-intuitive since annealing encourages a solution with higher mutual information, whereas lack of annealing is expected to suffer from stronger posterior collapse.

Examining models with PixelHVAE architecture might provide an explanation. PixelHVAE architecture was more susceptible to the lack of annealing, with VAE suffers significant drop of performance from lack of annealing. This suggests that convHVAE architecture was not expressive enough to benefit from KL-anealling. A more expressive PixelHVAE architecture resulted in a strong collapse in VAE, and smaller drop in performance for MIM. We conclude that MIM learning was more robust to the effects of KL-annealing, with a noticeable different in the learned representation for the most expressive models (*i.e.*, PixelHVAE).

5.4 Conclusions

We hypothesize that the collection of pathological behaviours in VAE, named posterior collapse, is the result of low mutual information regularizer in the VAE loss. We support that view by reorganizing the VAE loss to expose the existence of such a regularizer, and by a set of empirical evidence. We demonstrate that VAE exhibits the symptoms of posterior collapse (*e.g.*, low mutual information, and non-informative representation) in low dimensional synthetic data, and high dimensional image data, while MIM is less affected. We also show that KL-annealing has a significantly greater effect on VAE, when compared to MIM.

We conclude that VAE is prone to posterior collapse, as noted by many, whereas MIM is less so. We note that the common view of posterior collapse as the result of near-zero KL divergence term directed the attention of the research community to various heuristics in order to prevent small KL term. Here, we argue that the view of posterior collapse as the result of low mutual information paved the way to defining a new estimator, namely MIM, which encourages high mutual information. As a result, MIM learning is less reliant on optimization heuristics in order to prevent posterior collapse.

Chapter 6

SentenceMIM: A Latent Variable Language Model

6.1 Introduction

Generative modelling of text has become one of the predominant approaches to natural language processing (NLP), particularly in the machine learning community. It is favoured because it supports probabilistic reasoning and it provides a principled framework for unsupervised learning in the form of maximum likelihood. Unlike computer vision, where various generative approaches have proliferated (Dinh et al., 2017; Goodfellow et al., 2014a; Kingma and Welling, 2013; Oord et al., 2016; Rezende et al., 2014), current methods for text mainly rely on auto-regressive models.

Generative latent variable models (LVMs), such as the variational auto-encoder (VAE, Kingma and Welling (2013); Rezende et al. (2014)), are versatile and have been successfully applied to a myriad of domains. Such models consist of an encoder, which maps observations to distributions over latent codes, and a decoder that maps latent codes to distributions over observations. LVMs are widely used and studied because they can learn a latent representation that carries many useful properties. Observations are encoded as fixed-length vectors that capture salient information, allowing for semantic comparison, interpolation,

and search. They are often useful in support of downstream tasks, such as transfer or k-shot learning. They are also often interpretable, capturing distinct factors of variation in different latent dimensions. These properties have made LVMs especially compelling in the vision community.

Despite their desirable qualities, generative LVMs have not enjoyed the same level of success in text modelling. There have been several recent proposals to adapt VAEs to text (Bowman et al., 2015; Guu et al., 2017; Kruengkrai, 2019; Li et al., 2019a; Yang et al., 2017), but despite encouraging progress, they have not reached the same level of performance (*i.e.*, perplexity) on natural language benchmarks as auto-regressive models (*e.g.*, Merity et al. (2017); Rae et al. (2018); Wang et al. (2019)). This is often attributed to the phenomenon of posterior collapse (Le Fang, 2019; Li et al., 2019b), in which the decoder captures all of the modelling power and the encoder ends up conveying little to no information. Posterior collapse constitutes a significant problem in the context of representation learning, where non-informative latent codes are less likely to be useful for downstream tasks (*e.g.*, classification). For text, where the decoder is naturally auto-regressive, this has proven challenging to mitigate.

While the task of probability density estimation for language data has been challenging for VAEs, we point to recent progress in the task of language translation. Zhang et al. (2016); Łukasz Kaiser et al. (2018) successfully trained a conditional VAE (*i.e.*, conditioned on the source language), avoiding the negative effects of posterior collapse by conditioning the prior on the input sentence (*i.e.*, and hence keeping the latent code informative even in the case of a strong posterior collapse). Shah and Barber (2018) modelled the joint distribution over both the source and target languages, leading to empirically mitigated posterior collapse, but had to resort to various optimization and architecture heuristics (*e.g.*, KL annealing, decoder word dropouts, and limited vocabulary to reduce the expressiveness of the decoder). More recently, Gu et al. (2017); Shu et al. (2019) tackled the computational bottleneck which is associated with auto-regressive decoders by using a non-autoregressive decoders and explicitly modelling the sentence length. Despite the competitive results, the proposed

methods are not well suited to representation learning of language data since both models allow the dimensionality of the latent code to grow with the sentence length (*i.e.*, no fixed-size continuous representation).

This thesis introduces sentenceMIM (sMIM), a new LVM for text. It is based on the architecture of Bowman et al. (2015) and the mutual information machine (MIM) framework (Livne et al., 2019). MIM is a recently introduced LVM framework that shares the same underlying architecture as VAEs, but uses a different learning objective that is more robust against posterior collapse. MIM learns a highly informative and compressed latent representation, and often strictly benefits from more powerful architectures. To evaluate sMIM we propose a novel bound on the model log-likelihood, called MIM-ELBO, or *MELBO* . As an alternative to the evidence lower bound (ELBO) used to evaluate VAEs, MELBO is useful for models with implicit priors, for which the ELBO is intractable.

We show on four challenging datasets that sMIM outperforms VAE models for text, and is competitive with state-of-the-art auto-regressive approaches, including transformer-based models (Radford et al., 2019; Vaswani et al., 2017), as measured by negative log-likelihood and perplexity. We further demonstrate the quality of the sMIM representation by generating diverse samples around a given sentence and interpolating between sentences. Finally, we show the versatility of the learned representation by applying a pre-trained sMIM model to a question answering task with state-of-art performance as compared to single task, supervised models.

6.2 Problem Formulation

Let $\boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{X}$ be a discrete variable representing a sentence of tokens of length $T \in \{1, ..., T_{max}\}$ from a finite vocabulary $\mathcal{V}$, where T_{max} is the maximum sentence length. The set $\mathcal{X}$ comprises all sentences we aim to model. The total number of sentences X is typically unknown and large. Let $\mathcal{P}(\boldsymbol{x})$ be the unknown probability of sentence $\boldsymbol{x}$.

Our goal is to learn a latent variable model given N fair samples from $\mathcal{P}(\boldsymbol{x})$, where

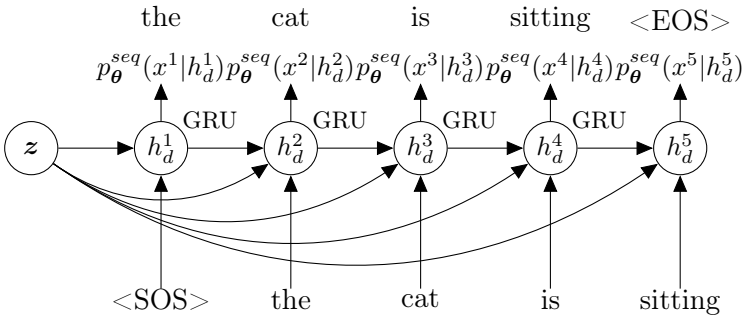


Figure 6.1: The decoder is auto-regressive, and conditioned on latent code $\boldsymbol{z}$. Words are represented by parametric embeddings. In each step (except the first) the previous output token and the latent code are inputs, and the GRU hidden output is then mapped to the parameters of a categorical distribution $p^{seq}_{\boldsymbol{\theta}}(x^k|h^k_d)$, from which the next token is sampled. The top sentence depicts the sample, with inputs on the bottom. <SOS>/<EOS> are the start/end-of-sentence tokens.

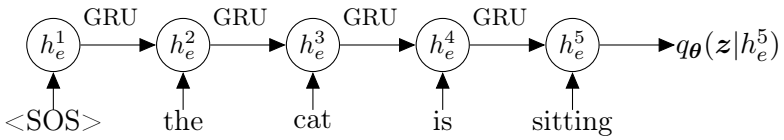


Figure 6.2: The encoder is implemented with GRU. Each word is represented by a parametric embedding. Given the input sequence, the encoder maps the last hidden state to the mean and variance of Gaussian posterior over latent codes, $q^{seq}_{\boldsymbol{\theta}}(z|h^k_d)$. <SOS> is the start-of-sentence token.

$N \ll X$. To this end, we consider probabilistic auto-encoders, defining distributions over discrete observations $\boldsymbol{x} \in \mathcal{X}$, and a corresponding continuous latent space, $\boldsymbol{z} \in \mathbb{R}^d$. They consist of an encoder, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, mapping sentences to a distribution over continuous latent codes, and a corresponding decoder, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, providing a distribution over sentences given a latent code. The joint parameters of the encoder and the decoder are denoted by $\boldsymbol{\theta}$. Ideally the encoder maps inputs to latent codes from which the decoder can correctly reconstruct the input. We also desire a latent space in which similar sentences (*e.g.*, in structure or content) are mapped to nearby latent codes.

6.2.1 Encoder-Decoder Specification

In what follows we adapt the architecture proposed by Bowman et al. (2015). Beginning with the generative process, let $p^{seq}_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ be a conditional auto-regressive distribution over

sequences of T tokens. We express the log probability of a sequence, $\boldsymbol{x} = (x^1, \ldots, x^T)$, with tokens $x^k \in \mathcal{V}$, as

$$\log p_{\boldsymbol{\theta}}^{seq}(\boldsymbol{x}|\boldsymbol{z}) \;=\; \sum_{k=1}^{T} \log p_{\boldsymbol{\theta}}^{seq}(x^k \,|\, x^{k-1}, \ldots, x^1, \boldsymbol{z}) \tag{6.1}$$

where $p_{\boldsymbol{\theta}}^{seq}(x^k|\cdot)$ is a categorical distribution over $|\mathcal{V}|$ possible tokens for the k^{th} token in $\boldsymbol{x}$, and $\boldsymbol{x}^0 \equiv$ <SOS> is the start-of-sentence token. According to the model (see Fig. 6.1), generating a sentence $\boldsymbol{x}$ with latent code $\boldsymbol{z}$ entails sampling each token from a distribution conditioned on the latent code and previously sampled tokens. Tokens are modelled with a parametric embedding.

The auto-regressive model, $p_{\boldsymbol{\theta}}^{seq}(\boldsymbol{x}|\boldsymbol{z})$, sums to one over all sequences of a given length. Combining this model with a distribution over sentence lengths $p(\ell)$, for $\ell \in \{1, ..., T_{max}\}$, we obtain the decoder, *i.e.*, a distribution over all sentences in $\mathcal{X}$:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \;=\; p_{\boldsymbol{\theta}}^{seq}(\boldsymbol{x}|\boldsymbol{z})\; p(T = \ell)\;. \tag{6.2}$$

Here, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ sums to one over all sentences of all lengths. The corresponding marginal $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is discussed in Sec. 6.2.3.

The encoder, or posterior distribution over latent codes given a sentence, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, is a conditional distribution over the latent variable $\boldsymbol{z}$. We take this to be Gaussian whose mean and diagonal covariance are specified by mappings $\mu_{\boldsymbol{\theta}}$ and $\sigma_{\boldsymbol{\theta}}$:

$$q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \;=\; \mathcal{N}(\boldsymbol{z}; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma_{\boldsymbol{\theta}}(\boldsymbol{x})) \tag{6.3}$$

Mappings $\mu_{\boldsymbol{\theta}}$ and $\sigma_{\boldsymbol{\theta}}$ are computed from the last hidden state of a GRU (Cho et al., 2014) (see Fig. 6.2).

6.2.2 Background: MIM Learning Objective

The Mutual Information Machine (MIM), introduced by Livne et al. (2019), is a versatile LVM. Like the VAE, it serves as a framework for representation learning, probability density estimation, and sample generation. Importantly, MIM learns a model with high mutual information between observations and latent codes, and with robustness against posterior collapse, which has been problematic for representation learning with VAEs and language data (*e.g.*, Bowman et al. (2015)).

MIM is formulated in terms of several elements. It assumes two *anchor* distributions, $\mathcal{P}(x)$ and $\mathcal{P}(z)$, for observations and the latent space, from which one can draw samples. They are fixed and not learned. There is also a parameterized encoder-decoder pair, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, and parametric marginal distributions $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. These parametric elements define joint encoding and decoding distributions:

$$q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) \;=\; q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\; q_{\boldsymbol{\theta}}(\boldsymbol{x})\;, \tag{6.4}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) \;=\; p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\; p_{\boldsymbol{\theta}}(\boldsymbol{z})\;. \tag{6.5}$$

For language modeling we use A-MIM learning, a MIM variant that minimizes a loss defined on the encoding and decoding distributions, with samples drawn from an encoding *sample* distribution, denoted $\mathcal{M}^q_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z})$; *i.e.*,

$$\mathcal{M}^q_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z}) \;=\; q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\;. \tag{6.6}$$

The particular loss for A-MIM is a variational upper bound on the joint entropy of the encoding sample distribution, which can be expressed with marginal entropies and mutual information terms. More precisely,

$$\begin{aligned}\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta}) \;&=\; \frac{1}{2}\big(\,CE\left(\,\mathcal{M}^q_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z})\,,\,q_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right) +\; CE\left(\,\mathcal{M}^q_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right)\,\big)\\ &\geq\; H_{\mathcal{M}^q_{\mathcal{S}}}(\boldsymbol{x}) + H_{\mathcal{M}^q_{\mathcal{S}}}(\boldsymbol{z}) - I_{\mathcal{M}^q_{\mathcal{S}}}(\boldsymbol{x};\boldsymbol{z})\;,\end{aligned} \tag{6.7}$$

where $CE(\cdot, \cdot)$ is cross-entropy, $H_{\mathcal{M}_{\mathcal{S}}^q}(\cdot)$ is information entropy over distribution $\mathcal{M}_{\mathcal{S}}^q$, and $I(\cdot;\cdot)$ is mutual information. Minimizing $\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta})$ learns a model with a consistent encoder-decoder, high mutual information, and low marginal entropy (Livne et al., 2019).

6.2.3 Variational Model Marginals

To complete the model specification, we define the model marginals $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. To help encourage consistency, and avoid introducing more model parameters, one can define model marginals in terms of marginals of the sample distributions (Bornschein et al., 2015; Livne et al., 2019; Tomczak and Welling, 2017).

We define the model marginal over observations as a marginal over the decoder (Bornschein et al., 2015): *i.e.*,

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \mathbb{E}_{\mathcal{P}(\boldsymbol{z})}\left[p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] \;, \tag{6.8}$$

where the latent anchor is defined to be a standard normal, $\mathcal{P}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, 1)$. Similarly, one can define the model marginal over latent codes as a marginal of the encoder,

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}) \;=\; \mathbb{E}_{\mathcal{P}(\boldsymbol{x})}\left[q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \;. \tag{6.9}$$

The latent marginal is defined as the aggregated posterior, in the spirit of the VampPrior (Tomczak and Welling, 2017).

6.2.4 Tractable Bounds to Loss

Given a training dataset $D = \{\boldsymbol{x}_i\}_{i=1}^N$, an empirical approximation to $\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta})$ is

$$\begin{aligned}\hat{\mathcal{L}}_{\text{A-MIM}}(\boldsymbol{\theta}) = \; & -\frac{1}{2N}\sum_{\boldsymbol{x}_i}\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)\, q_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right] \\ & - \frac{1}{2N}\sum_{\boldsymbol{x}_i}\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]\end{aligned} \tag{6.10}$$

where $\sum_{\boldsymbol{x}_i}$ denotes $\sum_{\boldsymbol{x}\in D}$, a sum over N fair samples drawn from $\mathcal{P}(\boldsymbol{x})$, as a Monte Carlo approximation to expectation over $\mathcal{P}(\boldsymbol{x})$.

Unfortunately, the empirical loss in Eqn. (6.10) is intractable since we cannot evaluate the log-probability of the marginals $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{x})$. In what follows we obtain a tractable empirical bound on the loss in Eqn. (6.10) for which, with one joint sample, we obtain an unbiased and low-variance estimate of the gradient (*i.e.*, using the reparameterization trick Kingma and Welling (2013)).

We first derive a tractable lower bound to $\log q_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$:

$$\begin{aligned}
\log q_{\boldsymbol{\theta}}(\boldsymbol{x}_i) &= \log \mathbb{E}_{\mathcal{P}(\boldsymbol{z})}\left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})\right] \\
&\underset{\text{(IS)}}{=} \log \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})\frac{\mathcal{P}(\boldsymbol{z})}{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\right] \\
&\underset{\text{(JI)}}{\geq} \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\log\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})\frac{\mathcal{P}(\boldsymbol{z})}{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\right)\right]
\end{aligned} \tag{6.11}$$

where the second and third lines are obtained using importance sampling and Jensen's inequality. We remind the reader that $q_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ is a variational marginal that can depend on $\boldsymbol{x}_i$. Indeed, Eqn. (6.11) is the usual ELBO.

To derive a lower bound to $\log p_{\boldsymbol{\theta}}(\boldsymbol{z})$, we begin with the following inequality,

$$\begin{aligned}
\log \mathbb{E}_{\mathcal{P}(\boldsymbol{x})}\left[h(\boldsymbol{x};\cdot)\right] &= \log \sum_i \mathcal{P}(\boldsymbol{x}_i)h(\boldsymbol{x}_i;\cdot) \\
&\geq \log \mathcal{P}(\boldsymbol{x}')h(\boldsymbol{x}';\cdot)\ ,
\end{aligned} \tag{6.12}$$

for any sample $\boldsymbol{x}'$, any discrete distribution $\mathcal{P}(\boldsymbol{x})$, and any non-negative function $h(\boldsymbol{x};\cdot) \geq 0$. The inequality in Eqn. (6.12) follows $\log a \geq \log b$ for $a \geq b$. Using this bound, we express a lower bound to $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ as follows,

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{z}) &\underset{\text{(Eqn. 6.9)}}{=} \log \mathbb{E}_{\mathcal{P}(\boldsymbol{x})}\left[q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \\
&\underset{\text{(Eqn. 6.11)}}{\geq} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}') + \log \mathcal{P}(\boldsymbol{x}')
\end{aligned} \tag{6.13}$$

for any sample $\boldsymbol{x}'$. We choose $\boldsymbol{x}' = \boldsymbol{x}_i$ during training for a joint sample $\boldsymbol{x}_i, \boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$.

Substituting Eqns. (6.11) and (6.13) into Eqn. (6.10) gives the final form of an upper

Algorithm 4 Learning parameters $\boldsymbol{\theta}$ of sentenceMIM

while not converged **do**
  $D_{\text{enc}} \leftarrow \{\boldsymbol{x}_j, \boldsymbol{z}_j \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})\}_{j=1}^{N}$
  $\hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D) = -\frac{1}{N}\sum_{i=1}^{N}\big(\ \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i) + \frac{1}{2}\left(\log q_{\boldsymbol{\theta}}(\boldsymbol{z}_i|\boldsymbol{x}_i) + \log \mathcal{P}(\boldsymbol{z}_i)\right)\ \big)$
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}_{\text{MIM}}(\boldsymbol{\theta}; D)$ {*Gradient computed through sampling using reparameterization*}
end while

bound on the empirical loss; *i.e.*,

$$\begin{aligned}\hat{\mathcal{L}}_{\text{A-MIM}} \ \leq\ & -\frac{1}{N}\sum_{i}\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})\right] \\ & -\frac{1}{2N}\sum_{i}\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)}\left[\log\big(\ q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)\mathcal{P}(\boldsymbol{z})\ \big)\right] \\ & +\frac{1}{2}H_{\mathcal{P}}(\,\boldsymbol{x}\,)\ . \end{aligned} \tag{6.14}$$

We find an unbiased, low variance estimate of the gradient of $\hat{\mathcal{L}}_{\text{A-MIM}}$ with a single joint sample $\boldsymbol{z}_i, \boldsymbol{x}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$ and reparameterization. The last term, $H_{\mathcal{P}}(\,\boldsymbol{x}\,)$, is a constant, independent of model parameters and can therefore be ignored during optimization. The resulting learning process is described in Algorithm 4.

To better understand the proposed bounds, we note that MIM achieves good reconstruction by learning posteriors with relatively small variances (*i.e.*, relative to the distance between latent means). Our choice of $\boldsymbol{x}' = \boldsymbol{x}_i$ exploits this, allowing good gradient estimation, facilitating fast convergence. We further provide empirical evidence for these properties below in Fig. 7.1.

6.3 NLL Evaluation

We would like to evaluate the log likelihood of the model $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ on a given test set, or equivalently, under a corresponding target distribution $\mathcal{T}(\boldsymbol{x})$. Unfortunately, the use of the implicit marginal defined in Eq. (6.9) makes such evaluation intractable because the variational marginal $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is defined with expectation over an unknown anchor distribution

$\mathcal{P}(\boldsymbol{x})$. Instead, for the purpose of evaluation, we define a new variational marginal using the learned parameters $\boldsymbol{\theta}$.

To this end, assume we are given a set of N IID samples drawn from $\mathcal{P}(\boldsymbol{x})$, denoted $D = \{\boldsymbol{x}_i\}_{i=1}^N$. We also assume for the purposes of testing that the number of samples in D is relatively small, *i.e.*, $N \ll X$ (where $X = |\mathcal{X}|$ is the total number of sentences in the domain, and is unknown). In this case we can assume the sentences in D are unique. The corresponding empirical distribution, $\mathcal{T}$, is uniform; i.e., $\mathcal{T}(\boldsymbol{x}_i) = 1/N$ for $\boldsymbol{x}_i \in D$. Accordingly, we define an empirical marginal as

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}; \mathcal{T}) = \mathbb{E}_{\mathcal{T}(\boldsymbol{x})} \left[q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \right] = \sum_{\boldsymbol{x}_i \in D} \mathcal{T}(\boldsymbol{x}_i)\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i) \tag{6.15}$$

In effect, this is a MC estimate of the aggregated posterior, based on the empirical test sample. It is based entirely on an IID sample from $\mathcal{P}(\boldsymbol{x})$. (Of course the training data used to learn the model parameters is also assumed to be an IID sample from $\mathcal{P}(\boldsymbol{x})$.) As the size of the test sample increases, the empirical marginal approaches the implicit marginal in Eq. (6.9).

In what follows we use this empirical marginal as an approximation to the intractable marginal for the purposes of evaluating the learned encoder-decoder model. Using the empirical marginal we define a new model for which $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ in the decoding distribution (6.5) is replaced by the empirical marginal $p_{\boldsymbol{\theta}}(\boldsymbol{z}; \mathcal{T})$ in Eq. (6.15). This model approaches the intractable model as the empirical marginal approaches the implicit marginal (*i.e.*, under $\mathcal{P}(\boldsymbol{x})$).

Accordingly, we begin with the derivation of a bound on the log data likelihood of the

approximate model, $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i; \mathcal{T})$, making use of the empirical marginal in Eq. (6.15); *i.e.*,

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i; \mathcal{T}) \quad &= \quad \log \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}; \mathcal{T})} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \boldsymbol{z}) \right] \\
&\underset{\text{(IS)}}{=} \quad \log \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \boldsymbol{z}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}; \mathcal{T})}{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)} \right] \\
&\underset{\text{(Eqn. 6.15)}}{=} \quad \log \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{T}(\boldsymbol{x}), q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \boldsymbol{z}) \frac{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}')}{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)} \right] \\
&\underset{\text{(Eqn. 6.12)}}{\geq} \quad \log \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \boldsymbol{z}) \right] + \log \mathcal{T}(\boldsymbol{x}_i) \qquad (6.16)
\end{aligned}$$

The second step above uses importance sampling, and the third line makes use of the form of the empirical marginal in Eqn. (6.15). The fourth line makes use of Eq. (6.12) for which we choose $\boldsymbol{x}' = \boldsymbol{x}_i$, motivated by the tendency for MIM to learn highly clustered representations (cf. Fig. 7.1). We can also view Eqn. (6.16) as an alternative to the usual ELBO; we refer to it as *MELBO* (*i.e.*, MIM ELBO).

Like the ELBO, the MELBO provides a lower bound for the model likelihood, and is computed per data point. Unlike the ELBO which is simply additive, the MELBO is transductive, as it depends on the entire test sample through the target distribution $\mathcal{T}(\boldsymbol{x})$ (*i.e.*, due to the term $\log \mathcal{T}(\boldsymbol{x}_i)$). More importantly, unlike the ELBO which requires the evaluation of $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, MELBO avoids it, and as such is better suited for evaluation of models with an implicit prior. Unfortunately, the transductive view of the approximate model (*i.e.*, where the prior is approximated) does not allow us to directly compare NLL values which are bounded with MELBO (*i.e.*, measures reconstruction quality) to NLL values which are bounded with ELBO (*i.e.*, measures the quality of language modelling). This is due to the different task each bound represents, where MELBO relates to the task of disambiguating all samples in a given sample set, while ELBO relates to the task of likelihood estimation of a learned model which is approximated using a sample set.

We note that in the VAE literature, it is beneficial in some cases to change the prior once the encoder and decoder have been trained (*e.g.*, Razavi et al. (2019); van den Oord et al. (2017c)). Interestingly, if post-hoc we change the latent prior to be a marginal distribution, as in Eqn. 6.15, then MELBO can be used to bound the NLL of a VAE model. This is

particularly effective when the aggregated posterior is a poor fit to the original Gaussian prior, which is penalized heavily in the EBLO.

We can now derive an upper bound on the NLL under $\mathcal{T}$ using the MELBO :

$$\begin{aligned} -\mathbb{E}_{\mathcal{T}(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x};\mathcal{T})\right] &\underset{\text{(Eqn. 6.16)}}{\leq} -\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\mathcal{T}(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] + H_{\mathcal{T}}(\boldsymbol{x}) \\ &\underset{\text{(MC)}}{\lessapprox} -\frac{1}{N}\sum_{\boldsymbol{x}_i}\left(\frac{1}{N_{\boldsymbol{z}}}\sum_{j=1}^{N_{\boldsymbol{z}}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_{i,j})\right) + \log N \end{aligned} \tag{6.17}$$

where $\boldsymbol{x}_i \in D$, and $N_{\boldsymbol{z}}$ samples $\boldsymbol{z}_{i,j}$ are drawn from the encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$. The last inequality follows $\log N$ being an upper bound on the entropy for the empirical distribution $\mathcal{T}(\boldsymbol{x})$. We denote this empirical upper-bound by $\widehat{NLL}$, and the corresponding perplexity (PPL) upper bound by $\widehat{PPL} \equiv \exp(\frac{N \cdot \widehat{NLL}}{\sum_i T_i}) \geq PPL$, where $\sum_i T_i$ is the total number of tokens in a dataset with N samples.

For the sizes of the datasets we consider, MELBO and ELBO tend to be comparable for VAE. We point, again, that $\widehat{NLL}$ grows with the number of unique sentences in the dataset (*i.e.*, categories of a discrete variable), as expected from a categorical distribution, and is not directly comparable to perplexity values which are bounded with ELBO.

Chapter 7

SentenceMIM: Experiments

7.1 Datasets

	Sentences				
(word level)	Train	Valid.	Test	Vocab.	#words (avg.)
PTB	42068	3370	3761	9877	21 ± 10
Yahoo	100K	10K	10K	37165	76 ± 55
Yelp15	100K	10K	10K	19730	100 ± 51
WikiText-103	200K	10K	2185	89247	115 ± 60
Everything †	442067	33369	33760	105965	94 ± 60

Table 7.1: Dataset properties summary for Penn Tree Bank (Marcus et al., 1993), Yahoo Answers and Yelp15 (cf. Yang et al. (2017)), and sampled WikiText-103 (Merity et al., 2016). Everything † is the union of all datasets.

We show experimental results on four word level datasets[1] described in Table 7.1, namely, Penn Tree Bank (Marcus et al., 1993), Yahoo Answers and Yelp15 (following Yang et al. (2017)), and WikiText-103 (Merity et al., 2016). We use the Yahoo and Yelp15 datasets of Yang et al. (2017), which draw 100k samples for training, and 10k for validation and testing. For WT103 we draw 200k samples for training, 10k for validation, and retain the original

[1] <SOS>, <EOS> are a special start/end-of-sentence tokens. The token <UNK> represents an out-of-vocabulary word.

test data. Empty lines and headers were filtered from the WT103 data.

7.2 Architecture and Optimization

Our auto-encoder architecture (Figs. 6.1 and 6.2), was adapted from that proposed by Bowman et al. (2015). As is common, we concatenated $\boldsymbol{z}$ with the input to the decoder (*i.e.*, a "context", similar to He et al. (2019); Yang et al. (2017); Bowman et al. (2015)). We use the same architecture, parameterization and latent dimensionality for sMIM and a VAE variant called sVAE, for comparison. Training times for sVAE and sMIM are similar.

For PTB we trained models with 1 layer GRU, latent space dimensions of 16D, 128D, and 512D, a 512D hidden state, 300D word embeddings, and 50% embedding dropout. We trained the models with Adam (Kingma and Lei Ba, 2014) with learning rate $lr = 10^{-3}$. The best performing model was trained in less than 30 minutes on a single TITAN Xp 12G GPU. For Yahoo Answers, Yelp15, and WT103 we trained models with 1 layer GRU, latent space dimensions of 32D, 512D, 1024D, a 1024D hidden state, 512D embeddings, and 50% embedding dropout. We trained these models with SGD (Sutskever et al., 2013), with $lr = 5.0$, and 0.25 L_2 gradient clipping.

In all cases we use a learning rate scheduler that scaled the learning rate by 0.25 following two/one epochs (PTB/other datasets, respectively) with no improvement in the validation loss. We used a mini-batch size of 20 in all cases. Following Sutskever et al. (2014) we feed the input in reverse to the encoder, such that the last hidden state in the encoder depends on the first word of the sentence in the decoder. (This gave slightly better results than with left to right order.)

We trained sVAEs with the regular ELBO, and with KL divergence annealing (denoted "+ kl"), where a scalar weight on the KL divergence term is increased from 0 to 1 over 10k mini-batches to lower the risk of posterior collapse and improve the learned models. We use no loss manipulation heuristics in the optimization of sMIM.

LVM ($\boldsymbol{z}$ dim.)	PPL (stdev)	NLL	BLEU	$\lvert\boldsymbol{\theta}\rvert$
sVAE (16)	$\leq$ 148.18 (0.11)	$\leq$ 113.46	0.124	11M
sVAE (128)	$\leq$ 158.31 (0.13)	$\leq$ 114.96	0.118	11M
sVAE (512)	$\leq$ 171.37 (0.31)	$\leq$ 116.76	0.116	12M
sMIM (16)	$\leq$ 76.3 (0.03)	$\leq$ 98.35	0.35	11M
sMIM (128)	$\leq$ 27.93 (0.008)	$\leq$ 75.58	0.61	11M
sMIM (512)	$\leq$ **19.53** (0.01)	$\leq$ **67.46**	**0.679**	12M
sMIM (1024) †	$\leq$ **4.6** (0.0)	$\leq$ **34.66**	**0.724**	179M

Table 7.2: PPL and NLL results for **PTB** bounded with MELBO , averaged over 10 runs (see text for details). Models† use extra training data.

LVM ($\boldsymbol{z}$ dim.)	PPL (stdev)	NLL	BLEU	$\lvert\boldsymbol{\theta}\rvert$
sVAE (32) + kl	$\leq$ 62.51 (0.01)	$\leq$ 410.83	0.274	40M
sVAE (512) + kl	$\leq$ 50.25 (0.01)	$\leq$ 389.14	0.18	43M
sVAE (1024) + kl	$\leq$ 52.79 (0.01)	$\leq$ 394.04	0.176	46M
sMIM (32)	$\leq$ 59.28 (0.0)	$\leq$ 405.55	0.309	40M
sMIM (512)	$\leq$ 10.05 (0.0)	$\leq$ 229.24	0.673	43M
sMIM (1024)	$\leq$ **9.98** (0.0)	$\leq$ **228.58**	**0.676**	46M
sMIM (1024)†	$\leq$ **8.19** (0.0)	$\leq$ **208.93**	**0.686**	179M

Table 7.3: PPL and NLL results for **Yelp15** bounded with MELBO , averaged over 10 runs (see text for details). Models† use extra training data.

7.3 Language Modelling Results

In what follows we compare the MELBO perplexity (PPL) of sMIM, and sVAE. For all datasets but PTB, VAE with KL annealing was more effective than a generic VAE; due to the small size of PTB, annealing produced over-fitting. We remove the <EOS> token during evaluation, since is not part of the data.

Tables 7.2-7.5 show results for PTB, Yelp15, Yahoo Answers, and WT103. Model sMIM (1024)† is trained on all datasets (*i.e.*, PTB, Yahoo Answers, Yelp15 and WT103). The BLEU-1 score is computed between test sentences and their reconstructions (higher is better). PPL and NLL (lower is better) are bounded with MELBO (Eqn. (6.17)). Finally, $\lvert\boldsymbol{\theta}\rvert$ indicates the number of parameters in the model.

LVM ($\boldsymbol{z}$ dim.)	PPL (stdev)	NLL	BLEU	$\lvert\boldsymbol{\theta}\rvert$
sVAE (32) + kl	$\leq$ 84.81 (0.01)	$\leq$ 329.26	0.181	67M
sVAE (512) + kl	$\leq$ 95.94 (0.05)	$\leq$ 338.4	0.139	70M
sVAE (1024) + kl	$\leq$ 102.93 (0.03)	$\leq$ 343.61	0.131	73M
sMIM32)	$\leq$ 56.84 (0.01)	$\leq$ 299.58	0.387	67M
sMIM (512)	$\leq$ 18.78 (0.0)	$\leq$ 217.48	0.664	70M
sMIM (1024)	$\leq$ **18.17** (0.0)	$\leq$ **215.02**	**0.669**	73M
sMIM (1024)†	$\leq$ **12.62** (0.0)	$\leq$ **188.03**	**0.682**	179M

Table 7.4: PPL and NLL results for **Yahoo Answers** bounded with MELBO , averaged over 10 runs (see text for details). Models† use extra training data.

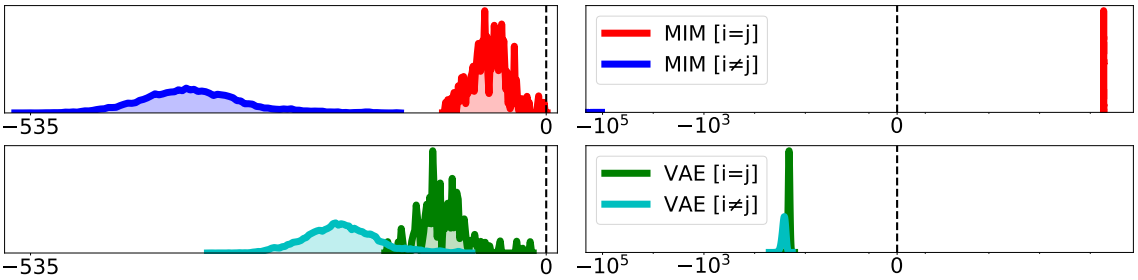
LVM ($\boldsymbol{z}$ dim.)	PPL (stdev)	NLL	BLEU	$\lvert\boldsymbol{\theta}\rvert$
sVAE (1024) + kl	$\leq$ 87.99 (0.08)	$\leq$ 489.33	0.165	153 M
sMIM (1024)	$\leq$ 21.95 (0.02) }	$\leq$ 337.58	0.571	153 M
sMIM (1024)†	$\leq$ **19.0** (0.01)	$\leq$ **321.35**	**0.603**	179M

Table 7.5: PPL and NLL results for **WT103** bounded with MELBO , averaged over 10 runs (see text for details). Models† use extra training data.

Results were validated using three methodologies. First, we provide an additional independent measure to the reconstruction quality with the unigram BLEU score (Papineni et al., 2001) between test sentences and their reconstructions. We use external code (Bird, 2002) to compute the values, as an independent validation to our strong PPL results. Second, our implementation of sVAE provides additional validation, showing ELBO PPL values similar to previously reported results[2]. sMIM shared the same model implementation, and differed only in the computation of the loss. Third, we provide MELBO values for sVAE, demonstrating values comparable to sMIM.

PTB, Yelp15, and Yahoo Answers results in Tables (7.2-7.4) show that sMIM improving significantly when compared to sMIM with a similar architecture. The results are especially interesting when considering the simple architecture used here (*i.e.*, 1 layer GRU). We also note here that sVAE shows posterior collapse as the decoder becomes more powerful (*i.e.*,

[2]We exclude the ELBO PPL results since they are not directly comparable to MELBO PPL values.



(a) $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_j)$ histograms.

(b) $\log q_{\boldsymbol{\theta}}(\boldsymbol{z}_i|\boldsymbol{x}_j)$ histograms.

Figure 7.1: Histograms of log probabilities of test data for sMIM and sVAE trained on **PTB**: Overlap between curves indicates potential for poor reconstruction of input sentences. (a) Histograms of $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_j)$ for $\boldsymbol{z}_j \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_j)$ when $i = j$ (same input), and when $i \neq j$ (when $\boldsymbol{x}_i$ is evaluated with the decoder distribution from a latent code associated with a different input sentence). (b) Histograms of $\log q_{\boldsymbol{\theta}}(\boldsymbol{z}_i|\boldsymbol{x}_j)$ for $\boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$, when conditioned on the same input $i = j$, or a different input $i \neq j$.

with large vocabulary size).

7.4 Posterior Collapse in VAE

The performance gap between sMIM and sVAE is due in part to posterior collapse in VAEs, where the encoder gives high posterior variance over latent codes, and hence low MI (cf. Zhao et al. (2018b); Alemi et al. (2017)); it coincides with KL divergence term in the usual ELBO approaching zero (in all or some dimensions). In such cases, different sentences are mapped to similar regions of the latent space. A code $\boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$ may have high probability density under the posterior given a different observation, *i.e.*, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_j)$ where $i \neq j$. In such cases, one might expect that observations sampled from $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$, might have high probability under the decoder for a different observation, *i.e.*, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_j)$, where $i \neq j$. In contrast, given the high mutual information and reconstruction quality of sMIM, we only expect high encoder and decoder densities when $i = j$. In other words, for sMIM, the posterior variances are relatively small compared to the distance between the posterior means.

The histograms in Fig. 7.1 illustrate this using the best sMIM and sVAE models trained on PTB. Histograms are labeled $[i \neq i]$ and $[i = j]$ for the two cases described above. They show that samples generated by sMIM given one input sentence are extremely unlikely to

Data	sMIM (S)	$\mathcal{N}(S)$	AE (S)	sMIM (M)	$\mathcal{N}(M)$	AE (M)
PTB	11.54	22.7	35.95	53.73	181.62	259.34
Yelp15	32.22	45.4	73.03	186.18	726.49	917.0
Yahoo	23.61	45.4	76.21	155.47	726.49	1003.26

Table 7.6: Entropy of the latent/hidden distribution for sMIM and AE (estimated using NN entropy estimator Kraskov et al. (2004)). (S,M) latent dimensions corresponds to (16D, 128D) in PTB and (32D, 512D) in Yelp15 and Yahoo Answers. For comparison, columns $\mathcal{N}(d)$ gives the entropy of a standard Normal in $\mathbb{R}^d$.

Data	sMIM (S)	sMIM (M)	sMIM (L)	AE (S)	AE (M)	AE (L)
PTB	0.35	0.61	**0.679**	0.348	0.589	0.637
Yelp15	0.309	0.673	0.676	0.402	0.682	**0.697**
Yahoo	0.387	0.664	**0.669**	0.395	0.647	0.394

Table 7.7: BLEU results for reconstruction of sMIM and AE. sMIM demonstrates empirical robustness to over-fitting, when compared to AE. (S,M,L) latent dimensions corresponds to (16D, 128D, 512D) in PTB and (32D, 512D, 1024D) in Yelp15 and Yahoo Answers.

be generated from sMIM given a different sentence. This is not the case for sVAE, where the histograms overlap. In other words, sMIM effectively maps sentences to non-overlapping regions of the latent space, allowing good reconstruction. By comparison, with sVAE sentences are mapped to overlapping regions of the latent space, which hinders accurate reconstruction.

7.5 Comparison of sMIM to Auto-encoders

Here we provide additional insight to the latent representation learned by sMIM. To do so, we contrast sMIM with a deterministic sequence auto-encoder (AE) of the same architecture. We train AEs by keeping the reconstruction term in the sVAE loss, and taking the mean of the posterior to be the hidden state that is fed to the decoder (*i.e.*, $\boldsymbol{z}_i = \mathbb{E}_{\boldsymbol{z}'}\left[q_{\boldsymbol{\theta}}(\boldsymbol{z}'|\boldsymbol{x}_i)\right]$). While AEs minimize the reconstruction error between the observations and the hidden state, they do not learn a distribution over latent codes. MIM, on the other hand, learns a low entropy distribution (*i.e.*, a compressed and clustered representation).

Model	P@1	MRR
AP-CNN (dos Santos et al., 2016)	0.560	0.726
AP-BiLSTM (dos Santos et al., 2016)	0.568	0.731
HyperQA (Tay et al., 2017a)	**0.683**	0.801
sMIM (512) ‡	**0.683**	**0.818**
AE (512) ‡	0.58	0.814
sMIM (1024) ‡ †	**0.753**	**0.861**

Table 7.8: **YahooCQA** results for sMIM, AE, and single-task models (higher is better). Results‡ are averaged over 10 runs (stdev < 0.002). sMIM (1024)† is pre-trained on PTB, Yahoo Answers, Yelp15 and WT103. P@1 and MRR are defined in Sec. 7.6.

We show the empirical entropy (estimated using NN entropy estimator Kraskov et al. (2004)) of the hidden/latent codes in Table 7.6. It is clear that sMIM learns a low entropy representation (*i.e.*, lower than the anchor $\mathcal{P}(z)$), whereas AE has no notion of a latent distribution, leading to high information entropy in the hidden state (*i.e.*, more uniformly distributed, with less structure).

Table 7.7 shows BLEU values for sMIM and AE with the same architecture. Interestingly, the added latent stochasticity in sMIM helps mitigate over-fitting, while AE is more sensitive to the choice of architecture (*i.e.*, stronger model might over-fit), as evident for Yahoo Answers. In addition, learning a latent distribution makes sMIM a useful model for downstream tasks, as we discuss next.

7.6 Question-Answering

To demonstrate the versatility of sMIM, we consider a downstream task in which sMIM (512) is pre-trained on Yahoo Answers, then used for question-answering on YahooCQA (Tay et al., 2017b), with no fine-tuning. The YahooCQA vocabulary has 116,900 tokens, with training, validation and test sets having 253K, 31.7K and 31.7K QA pairs, respectively. These sets were constructed by taking a subset of the QA pairs from Yahoo Answers, from which each question is then paired with another 2-4 answers (not from Yahoo Answers). Thus each question, of 5-50 tokens, has 3-5 possible ranked answers (1 is best). Let Q_i denote the i^{th}

Q: <SOS> my brother is geting out on parole from navy jail where can i find a parole office in our area <UNK> , <UNK> ?
A: you can find out the county jail , or call your local police station . <EOS>
Q: <SOS> what continent has most deserts ?
A: the most notable is in the netherlands . <EOS>
Q: <SOS> how do u clear the history in the search field ?
A: u can find it in the search bar . <EOS>
Q: <SOS> what is the best question to ask ?
A: ask yourself ! <EOS>
Q: <SOS> need to find somewhere to sale baseball cards . ?
A: ebay <EOS>
Q: <SOS> what's the opposite of opposite ?
A: opposite opposite opposite ; i thought it really helps . <EOS>

Table 7.9: Sampled answers from **Yahoo Answers** sMIM (1024).

question, and let $\{A_i^k\}_{k=1}^{K_i}$ be the K_i corresponding answers, ordered such that A_i^k has rank k. To match the format of QA pairs in Yahoo Answers, we compose question-answer pair Q_i^k by concatenating Q_i, "?", and A_i^k.

For question-answering with sMIM we use the following procedure: For each question-answer we sample $\boldsymbol{z}_i^k \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|Q_i^k)$, and a corresponding $\boldsymbol{z}_i^{unk} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|Q_i^{unk})$ where Q_i^{unk} is simply Q_i concatenated with "?" and a sequence of <unk> tokens to represent the $|A_i^k|$ unknown words of the answer. We than rank question-answer pairs according to the score $S_i^k = ||\boldsymbol{z}_i^{unk} - \boldsymbol{z}_i^k||/\sigma_i^{k,unk}$ where $\sigma_i^{k,unk}$ is the standard deviation of $q_{\boldsymbol{\theta}}(\boldsymbol{z}|Q_i^{unk})$. In other words, we rank each question-answer pair according to the normalized distance between the code of the question with, and without, the answer. This score is similar to $\log q_{\boldsymbol{\theta}}(\boldsymbol{z}_i^k|Q_i^{unk})$, but without taking the log standard deviation into account.

Table 7.8 quantifies test performance using average precision ($P@1 = \frac{1}{N}\sum_i \mathbb{1}(rank(A_i^1) = 1)$), and Mean Reciprocal Ranking ($MRR = \frac{1}{N}\sum_i \frac{1}{rank(A_i^1)}$). Interestingly, sMIM (512), pre-trained on Yahoo Ansrews, exhibits state-of-the-art performance compared to single-task models trained directly on YahooCQA data with the aid of supervision. For an even larger sMIM model, pre-trained on all of PTB, Yahoo Answers, Yelp15 and WT103, the

5 stars → 1 star
<SOS> awesome food , just awesome ! top notch beer selection . great staff . beer garden is great setting .
• awesome food , just top notch ! great beer selection . staff has great craft beer . top notch is that . <EOS>
• awesome food ! just kidding , beer selection is great . staff has trained knowledge on top . <EOS>
• cleanliness is awesome ! not only on their game , food . server was polite his hand sanitizer outside . <EOS>
• cleanliness is not on their patio . server was outside , kept running his hand sanitizer his hand . <EOS>
<SOS> cleanliness is not on their radar . outside patio was filthy , server kept running his hand thru his hair .

Table 7.10: Interpolation results between latent codes of input sentences (with gray) from **Yelp15** for sMIM (1024).

question-answering performance of sMIM is even better (last row of Table 7.8).

Finally, as another point of comparison, we repeated the experiment with a deterministic AE model (with $\sigma_i^{k,unk} = 1$). In this case performance drops, especially average precision, indicating that the latent representations are not as semantically meaningful.

We also note that we can also use sMIM to generate novel answers rather than simply ranking several alternatives. To this end, we sample $\boldsymbol{z}_i^{unk} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}_i^k | Q_i^{unk})$, as described above, followed by modified reconstruction $\widehat{Q}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} | \boldsymbol{z}_i^{unk})$. We modify the sampling procedure to be greedy (*i.e.*, top 1 token), and prevent the model from sampling the "<UNK>" token. We consider all words past the first "?" as the answer. (We also removed HTML tags (*e.g.*, "
").) Table 7.9 gives several selected answers. The examples were chosen to be short, and with appropriate (non-offensive) content.

7.7 Reconstruction, Interpolation, and Perturbation

As a final exploration of sMIM, we probe the learned representation, demonstrating that sMIM learns a dense, meaningful latent space. We present latent interpolation results in Table 7.10 for samples (*i.e.*, reviews) with the different ratings from Yelp5. Interpolation

(D)	<SOS> the company did n't break out its fourth-quarter results
(M)	the company did n't break out its results <EOS>
(R)	the company did n't break out its fourth-quarter results <EOS>
(P)	the company did n't accurately out its results <EOS>

Table 7.11: Reconstruction results for sMIM (512) model trained on **PTB**. We denote: (D) Data sample; (M) Mean (latent) reconstruction; (R) Reconstruction; (P) Perturbed (latent) reconstruction.

entails sampling $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_\alpha)$ where $\boldsymbol{z}_\alpha$ is interpolated at equispaced points between two latent codes, $\boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$, and $\boldsymbol{z}_j \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_j)$.

Next we show reconstruction, and perturbation results for for sMIM (512) trained on PTB. Figure 7.11 shows four sentences: (D) the input sentence; (M) the mean reconstruction given the posterior mean z; (R) a reconstruction given a random sample z from the posterior; and (P) a *perturbed reconstruction*, given a sample z from a Gaussian distribution with 10 times the posterior standard deviation. The high mutual information learned by sMIM leads to good reconstruction, as clear in (M) and (R). sMIM also demonstrates good clustering in the latent space, shown here by the great similarity of (R) and (P).

7.8 Conclusions

This thesis introduces a new generative auto-encoder for language modeling, trained with A-MIM learning. The resulting framework learns an encoder that provides a continuous distribution over latent codes for a sentence, from which one can reconstruct, generate and interpolate sentences. In particular, compared to recent attempts to uses VAEs for language learning, A-MIM provides models with high mutual information between observations and latent codes, improved reconstruction, and it avoids posterior collapse. On PTB, Yaoo Anwers, and Yelp15 we obtain state-of-the-art perplexity results, with competitive results on Wiki103. We also use the latent representation for a downstream question-answering task on YahooCQA with state-of-the-art results. Finally, we demonstrate language generation, perturbation and interpolation using the latent representation.

Chapter 8

TzK: Conditional Generative Model

8.1 Introduction

The goal of representation learning is to learn structured, meaningful latent representations given large-scale unlabelled datasets. It is widely assumed that such unsupervised learning will support myriad downstream tasks, some of which may not be known *a priori* (Bengio et al., 2013). To that end it is useful to be able to train on large amounts of hetergeneous data, but then use conditional priors that isolate specific sub-spaces or manifolds from the broader data distribution over the observation domain.

Building on probability flows (Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018), this thesis introduces a flexible form of conditional MIM. It is compositional in nature, without requiring *a priori* knowledge of the number of classes or the relationships between classes. Trained with MC sampling and SGD, the framework allows one to learn from heterogeneous datasets in an unsupervised fashion, with concurrent or subsequent specialization to sub-spaces or manifolds of the observation domain, *e.g.*, conditioning on class labels or attributes. The resulting model thereby supports myriad downstream tasks, while providing efficient inference and sampling from the joint or conditional priors.

Unlike previous chapters, the work presented here is in a preliminary stage. Here we discusses a conditional MIM model (*e.g.*, conditioned on class label) named TzK , and

demonstrate the importance of symmetry in MIM. In more details, we exploit the symmetric nature of a learned MIM model in order to construct (*i.e.*, compose) a joint distribution from multiple conditional distributions.

8.2 Background

There has been significant interest in learning generative models in recent years. Prominent models include variational auto-encoders (VAE), which maximize a variational lower bound on the data log likelihood (Rezende et al., 2014; Kingma and Welling, 2013; van den Berg et al., 2018; Papamakarios et al., 2017; Kingma et al., 2016), and generative adversarial networks (GAN), which use an adversarial discriminator to enforce a non-parametric data distribution on a parametric decoder or encoder (Goodfellow et al., 2014b; Makhzani, 2018; Makhzani et al., 2015; Chen et al., 2016a). Inference, however, remains challenging for VAEs and GANs as neither model includes a direct probability density estimator (Schmah et al., 2009; Papamakarios et al., 2017; Dinh et al., 2016, 2014). In more details, VAEs require Monte-Carlo approximation of the marginal over observations, and GANs do not learn an explicit distribution over observations altogether.

Auto-regressive models (Germain et al., 2015; Bengio and Bengio, 1999; Larochelle and Murray, 2011) and normalizing flows (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015; Kingma and Dhariwal, 2018) train with maximum likelihood (ML), avoiding approximations by choosing a tractable parameterization of probability density. Auto-regressive models assume a conditional factorization of the density function, yielding a tractable joint probability model. Normalizing flows represent the joint distribution with a series of invertible transformations of a known base distribution, but are somewhat problematic in terms of the memory and computational costs associated with large volumes of high-dimensional data (*e.g.*images). While invertibility can be used to trade memory with compute requirements (Chen et al., 2018; Gomez et al., 2017), training powerful density estimators remains challenging.

The attraction of unsupervised learning stems from a desire to exploit vast amounts of

data, especially when downstream tasks are either unknown *a priori*, or when one lacks ample task-specific training data. And while samples from models trained on heterogeneous data may not resemble one's task domain per se, conditional models can be used to isolate manifolds or sub-spaces associated with particular classes or attributes. The TzK framework incorporates task-specific conditioning in a flexible manner. It supports end-to-end training of the full model. Or one to train a powerful density estimator once, retaining the ability to later extend it to new domains, or specialize it to sub-domains of interest. We get the advantages of large heterogenous datasets, while retaining fidelity of such specialized conditional models.

Existing conditional generative models allow one to sample from sub-domains of interest (*e.g.*, Makhzani (2018); Chen et al. (2016a); Dupont (2018)), but they often require that the structure of the data and latent representation be known *a priori* and embedded in the network architecture. For example, (Chen et al., 2016a; Makhzani, 2018) allow unsupervised learning but assume the number of (disjoint) categories is given. In doing so they fix the structure of the latent representation to include a 1-hot vector over categories at the time of training. Such models are therefore re-trained from scratch if labels change, or if new labels are added, *e.g.*by augmenting the training data.

Kingma and Dhariwal (2018) train a conditional prior post hoc, given an existing Glow model. This allows them to condition an existing model on semantic attributes, but lacks the corresponding inference mechanism. A complementary formulation, augmenting a generative model with a post hoc discriminator, is shown by Oliver et al. (2018).

Inspired by (Kingma and Dhariwal, 2018; Oliver et al., 2018), TzK incorporates conditional models with discriminators and generators, all trained jointly. The proposed framework can be trained unsupervised on large volumes of data, yielding a generic representation of the observation domain (*e.g.*, images), while explicitly supporting the semi-supervised learning of new classes in an online fashion. Such conditional models are formulated to be compositional, without a prior knowledge of all classes, and exploiting similarity among classes with a joint latent representation.

Finally, the formulation below exhibits an interesting connection between the use of

mutual information (MI) and ML in representation learning. The use of MI is prevalent in learning latent representations (Belghazi et al., 2018; Chen et al., 2016a; Dupont, 2018; Klys et al., 2018), as it provides a measure of the dependence between random variables. Unfortunately, MI is hard to compute; it is typically approximated or estimated with non-parametric approaches. A detailed analysis is presented by Belghazi et al. (2018), which offers scalability with data dimensionality and sample size. While it is intuitive and easy to justify the use of MI to enforce a relationship between random variables (*e.g.*, dependency (Chen et al., 2016a) or independence (Klys et al., 2018)), MI is often used as a regularizer to extend an existing model. The TzK formulation offers another perspective, showing how MI arises naturally with the ML objective, following the assumption that a target distribution can be factored into (equally plausible) encoder and decoder models. We exploit a lower bound that allows the learning procedure to use indirect optimization of MI, without estimating MI explicitly.

Contributions: We introduce a conditional generative model based on probability density normalizing flows, which is flexible and extendable. It does not require that the number of classes be known *a priori*, or that classes are mutually exclusive. One can train a powerful generative model on unlabeled samples from multiple datasets, and then adapt the structure of the latent representation as a function of specific types of knowledge in an online fashion. The proposed model allows training to be parallel when training multiple tasks while maintaining a joint distribution over the latent representation and observations, all with efficient inference and sampling.

8.3 TzK Framework

We model a joint distribution over an observation domain (*e.g.*, images) and latent codes (*e.g.*, attributes or class labels). Let observation $\boldsymbol{t} \in \mathbb{R}^T$ be a random variable associated through a probability flow with a latent state variable $\boldsymbol{z} \in \mathbb{R}^T$ (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015; Kingma and Dhariwal, 2018). In particular, $\boldsymbol{z}$ is mapped

to $\boldsymbol{t}$ through a smooth invertible mapping $f_{\boldsymbol{t}} : \mathbb{R}^T \rightarrow \mathbb{R}^T$, *i.e.*, $\boldsymbol{t} = f_{\boldsymbol{t}}(\boldsymbol{z})$. As such, $f_{\boldsymbol{t}}$ transforms a base distribution $p(\boldsymbol{z})$ (*e.g.*, Normal) to a distribution $p(\boldsymbol{t}) = p(\boldsymbol{z}) \left| \det \frac{\partial f_{\boldsymbol{t}}}{\partial \boldsymbol{z}} \right|^{-1}$ over the observation domain. Normalizing flows can be formulated as a joint distribution $p(\boldsymbol{z}, \boldsymbol{t}) = \delta(\boldsymbol{z} - f_{\boldsymbol{t}}^{-1}(\boldsymbol{t}))\, p(\boldsymbol{t}) = \delta(\boldsymbol{t} - f_{\boldsymbol{t}}(\boldsymbol{z}))\, p(\boldsymbol{z})$, but for notational simplicity we can omit $\boldsymbol{t}$ or $\boldsymbol{z}$ from probability distributions by trivial marginalization of one or the other.

For conditional generative models within the TzK framework, the latent state $\boldsymbol{z}$ is conditioned on a latent code (see Fig. 8.1b). As such, they capture distributions within the observation domain associated with subsets of the training data, or subsequent labelled data. To this end, let $\boldsymbol{k}^i$ be a hybrid discrete/continuous random variable $\boldsymbol{k}^i \equiv (\mathrm{e}^i, \boldsymbol{c}^i)$, where $\mathrm{e}^i \in \{0, 1\}$ and $\boldsymbol{c}^i \in \mathbb{R}^C$, similar to (Chen et al., 2016a; Dupont, 2018). We refer to $\boldsymbol{k}^i$ as *knowledge* of type i, while $\boldsymbol{c}^i$ is the latent code of knowledge i, a structured latent representation of $\boldsymbol{t}$. We call e^i the existence of knowledge i, a binary variable that serves to indicate whether or not $\boldsymbol{t}$ can be generated by $\boldsymbol{c}^i$.

To handle multiple types of knowledge, let $\bar{\boldsymbol{k}} = \{\boldsymbol{k}^i\}_{k=1}^C$ denote the set of latent codes associated with C knowledge types. Importantly, we do not assume that knowledge types correspond to mutually exclusive class labels. Rather, we allow varying levels of interaction between knowledge classes under the TzK framework. This avoids the assumption of mutually exclusive classes and allows a TzK model to share a learned representation between similar classes, while still being able to represent distinct classes.

8.3.1 Formulation

Our goal is to learn a probability density estimator of the joint distribution $\mathcal{P}(\boldsymbol{t}, \bar{\boldsymbol{k}})$. In terms of an encoder-decoder, for effective inference and sample generation, we model $\mathcal{P}$ in terms of two factorizations, i.e.,

$$q\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right) = p\left(\bar{\boldsymbol{k}} | \boldsymbol{t}\right)\, p\left(\boldsymbol{t}\right) \tag{8.1}$$

$$p\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right) = p\left(\boldsymbol{t} | \bar{\boldsymbol{k}}\right)\, p\left(\bar{\boldsymbol{k}}\right) \tag{8.2}$$

(a) encoder (b) decoder

Figure 8.1: TzK framework models $\mathcal{P}(\boldsymbol{t}, \bar{\boldsymbol{k}})$, a joint distribution over task domain $\boldsymbol{t}$ and multiple latent codes $\bar{\boldsymbol{k}} = \{\boldsymbol{c}^i, \mathrm{e}^i\}_{i=1}^{C}$ with a dual encoder/decoder. The framework offers explicit representation of sub-domains of interest in $\mathcal{P}(\boldsymbol{t}, \bar{\boldsymbol{k}})$ by conditioning on the latent codes which comprise a single compositional model.

The encoder factorization in Eq. (8.1) makes $p(\bar{\boldsymbol{k}}|\boldsymbol{t})$ explicit, which is used for inference of the latent code given $\boldsymbol{t}$. The decoder in Eq. (8.2) makes $p(\boldsymbol{t}|\bar{\boldsymbol{k}})$ explicit for generation of samples of $\boldsymbol{t}$ given a latent code $\bar{\boldsymbol{k}}$.

As noted by (Kingma and Welling, 2013; Agakov and Barber, 2003; Rezende and Mohamed, 2015; Chen et al., 2016a), inference with the general form of the posterior $p(\bar{\boldsymbol{k}}|\boldsymbol{t})$ is challenging. Common approaches resort to variational approximations (Kingma and Welling, 2013; Rezende and Mohamed, 2015; Kingma et al., 2016). A common relaxation in the case of discrete latent codes is the assumption of independence (*e.g.*, $p(\mathrm{e}|\boldsymbol{t}, \bar{\boldsymbol{c}}) = \prod_i p(\mathrm{e}^i|\boldsymbol{t}, \bar{\boldsymbol{c}})$). Alternatively, one can assume that such binary codes represent mutually exclusive classes, e.g., with a single categorical random variable. But this makes it difficult to model attributes, for which the presence or absence of one attribute may be independent of other attributes, or to allow for the fact that one image may belong to two different classes (*e.g.*, it might be present in more than one database).

Here we design TzK to avoid the need for mutual exclusivity, or the need to specify the number of classes a priori, instead allowing the model to be extended with new classes, and to learn and exploit some degree of similarity between classes. To that end we assume that knowledge types exhibit statistical independence, expressed in terms of the following encoder

factorization,

$$q\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right) \;=\; p\left(\boldsymbol{t}\right) \prod_i p\left(\boldsymbol{k}^i | \boldsymbol{t}\right) \;, \tag{8.3}$$

and the corresponding decoder factorization

$$\begin{aligned} p(\boldsymbol{t}, \bar{\boldsymbol{k}}) &= p(\bar{\boldsymbol{k}}) \, \frac{p(\bar{\boldsymbol{k}}|\boldsymbol{t}) \, p(\boldsymbol{t})}{p(\bar{\boldsymbol{k}})} \\ &= p(\bar{\boldsymbol{k}}) \, p(\boldsymbol{t}) \prod_i \frac{p(\boldsymbol{k}^i|\boldsymbol{t}) \, p(\boldsymbol{t})}{p(\boldsymbol{k}^i) \, p(\boldsymbol{t})} \\ &= \frac{\prod_i p\left(\boldsymbol{t}|\boldsymbol{k}^i\right) p\left(\boldsymbol{k}^i\right)}{p\left(\boldsymbol{t}\right)^{C-1}} \end{aligned} \tag{8.4}$$

It is by virtue of this particular factorization that a TzK model is easily extendable with different knowledge types (and conditional models) in an online fashion. We note that Eq. (8.4) bears strong similarity to product of experts (PoE), which was introduced by Hinton (2002b). However, unlike PoE, TzK models normalized distributions (*i.e.*, has a tractable partition function), and as such can be trained with MC and SGD. PoE, on the other hand, is trained with contrastive divergence, which has been demonstrated to be challenging in practice (*e.g.*, unstable optimization).

Taking the hybrid form of knowledge codes into account, as in Fig. 8.1, the model is further factored as follows:

$$p\left(\boldsymbol{k}^i|\boldsymbol{t}\right) \;=\; p\left(\boldsymbol{c}^i|\mathrm{e}^i, \boldsymbol{t}\right) p\left(\mathrm{e}^i|\boldsymbol{t}\right) \tag{8.5}$$

$$p\left(\boldsymbol{t}|\boldsymbol{k}^i\right) \;=\; p\left(\boldsymbol{t}|\mathrm{e}^i, \boldsymbol{c}^i\right) \tag{8.6}$$

$$p\left(\boldsymbol{k}^i\right) \;=\; p\left(\mathrm{e}^i|\boldsymbol{c}^i\right) p\left(\boldsymbol{c}^i\right) \;. \tag{8.7}$$

Here, $p\left(\mathrm{e}^i = 1|\boldsymbol{t}\right)$ and $p\left(\mathrm{e}^i = 1|\boldsymbol{c}^i\right)$ act as discriminators for binary variable e^i, conditioned on $\boldsymbol{t}$ and $\boldsymbol{c}^i$ respectively.

Finally, the factors of the encoder and decoder in (8.3) - (8.7) are parametrized in terms of neural networks. Accordingly, denoting the parameters of the encoder and decoder by $\boldsymbol{\phi}$

and $\boldsymbol{\psi}$, in what follows we write the parametrized model encoder and decoder as $q_{\boldsymbol{\theta}}(\boldsymbol{t}, \bar{\boldsymbol{k}})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{t}, \bar{\boldsymbol{k}})$. (In what follows we use this more concise notation for the encoder and decoder, except where we need the explicit factorization in terms of $\boldsymbol{k}^i$, $\boldsymbol{c}^i$ and e^i.) Details of our implementation are described in Sec. 8.4.1.

8.3.2 Learning

We would like to train a parametric model of the joint distribution $\mathcal{P}(\boldsymbol{t}, \bar{\boldsymbol{k}})$ with the dual encoder/decoder factorization defined in Eqs. (8.3) - (8.7). Following the success of (Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018) with high-dimensional distributions, we opt to estimate the model parameters using MIM (Section 3.5).

We aim to learn a single probabilistic model of $\mathcal{P}(\boldsymbol{t}, \bar{\boldsymbol{k}})$, comprising a consistent encoder-decoder, with the factorization given above, and a shared flow $f_{\boldsymbol{t}}$. To do so, we use MIM learning, as described in Section 3.5. More explicitly, we define a joint distribution $\mathcal{M}_{\boldsymbol{\theta}}\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right)$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\phi}, \boldsymbol{\psi}\}$. Expressed as a linear mixture, $\mathcal{M}_{\boldsymbol{\theta}}$ is randomly selected to be $q_{\boldsymbol{\theta}}$ or $p_{\boldsymbol{\theta}}$ with equal probability, *i.e.*,

$$\mathcal{M}_{\boldsymbol{\theta}}\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right) = \frac{1}{2}\left(q_{\boldsymbol{\theta}}\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right) + p_{\boldsymbol{\theta}}\left(\boldsymbol{t}, \bar{\boldsymbol{k}}\right)\right) \ . \tag{8.8}$$

Choosing the mixing coefficients to be equal reflects our assumption of a dual encoder/decoder parametrization of the same underlying joint distribution. There are other ways to combine q and p into a single model; we chose this particular formulation because it yields a very effective learning algorithm.

MIM learning entails maximizing $\mathbb{E}_{\boldsymbol{t}, \bar{\boldsymbol{k}} \sim \mathcal{P}}\left[\log \mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{t}, \bar{\boldsymbol{k}})\right]$ with respect to $\boldsymbol{\theta}$; equivalently,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} -CE\left(\mathcal{P}\,,\, \mathcal{M}_{\boldsymbol{\theta}}\right) \ , \tag{8.9}$$

where $CE\left(\cdot, \cdot\right)$ denotes cross-entropy. Using Jensen's inequality it is straightforward to

show that $\log(\frac{1}{2}q_{\boldsymbol{\theta}} + \frac{1}{2}p_{\boldsymbol{\theta}}) \geq \frac{1}{2}\log q_{\boldsymbol{\theta}} + \frac{1}{2}\log p_{\boldsymbol{\theta}}$, and as a consequence,

$$-H(\mathcal{P}, \mathcal{M}_{\boldsymbol{\theta}}) \geq -\frac{1}{2}\left[\, CE\left(\mathcal{P}\,,\, q_{\boldsymbol{\theta}}\right) + CE\left(\mathcal{P}\,,\, p_{\boldsymbol{\theta}}\right)\right]. \tag{8.10}$$

The lower bound encourages consistency between the encoder and decoder. To see this, we examine the bound in greater detail. With some algebraic manipulation, ignoring expectation in Eq. (8.10), one can derive the following:

$$\frac{\log q_{\boldsymbol{\theta}} + \log p_{\boldsymbol{\theta}}}{2} = \log \mathcal{M}_{\boldsymbol{\theta}} - \log \frac{\sqrt{\frac{q_{\boldsymbol{\theta}}}{p_{\boldsymbol{\theta}}}} + \sqrt{\frac{p_{\boldsymbol{\theta}}}{q_{\boldsymbol{\theta}}}}}{2}. \tag{8.11}$$

This implies that maximization of the lower bound (the expectation of the LHS) entails maximization of the expectation of the two terms on the RHS, the first of which is $-CE\left(\mathcal{P}\,,\, \mathcal{M}_{\boldsymbol{\theta}}\right)$. The expectation of the second term on the RHS of Eq. (8.11) can be viewed as a regularizer that encourages the encoder and decoder to assign similar probability density to each datapoint. Importantly, it obtains its upper bound of zero when $q_{\boldsymbol{\theta}} = p_{\boldsymbol{\theta}}$, in which case the inequality in Eq. (8.10) becomes equality. In practice, we find the bound is tight.

We note that $-CE\left(\mathcal{P}\,,\, \mathcal{M}_{\boldsymbol{\theta}}\right)$ itself is a lower bound on $-H\left(\mathcal{P}\right)$, since $-H\left(p\right) \geq -CE\left(p\,,\, q\right)$ for any distributions p and q. If $\mathcal{P}(\bar{\boldsymbol{k}}, \boldsymbol{t})$ satisfies the factorization of the TzK model in Eqs. (8.3) - (8.7) then the entropy of the joint distribution can be expressed as

$$\begin{aligned} -H\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right) &= -H\left(\boldsymbol{t}\right) - \sum_i H\left(\boldsymbol{k}^i\right) \\ &\quad + \frac{1}{2}\sum_i \left[\, I(\boldsymbol{k}^i; \boldsymbol{t}) + I(\boldsymbol{z}; \boldsymbol{k}^i)\right], \end{aligned} \tag{8.12}$$

where $H\left(\boldsymbol{t}\right)$ and $H\left(\boldsymbol{k}^i\right)$ denote entropy of marginal distributions, and $I(\boldsymbol{k}^i, \boldsymbol{t})$ and $I(\boldsymbol{z}; \boldsymbol{k}^i)$ denote mutual information, for which all expectations are with respect to $\bar{\boldsymbol{k}}, \boldsymbol{t} \sim \mathcal{P}$. (The derivation of Eq. (8.12) is given in the supplemental material.) Eq. (8.12) suggests that maximizing the MI between observations and latent codes here follows from a design choice, for a model that can equally well "understand" (encode) and "express" (decode) an independent

set of latent codes (as in Eqs. (8.3) and (8.4)), within a shared observation domain.

We claim that the assumption of independent latent codes is a relatively mild assumption, and has little affect on the ability of the model to represent $p(\boldsymbol{t}|\bar{\boldsymbol{y}})$ for a random variable $\bar{\boldsymbol{y}}$ over the same domain as $\bar{\boldsymbol{k}}$. A sufficiently expressive flow $\boldsymbol{t} = f_{\boldsymbol{t}}(\boldsymbol{z})$ will allow for $p(\boldsymbol{t}|\bar{\boldsymbol{y}}) = \mathcal{P}(f_{\boldsymbol{t}}(\boldsymbol{z})|\bar{\boldsymbol{k}})$ for arbitrary $\bar{\boldsymbol{y}} \sim p(\bar{\boldsymbol{y}})$, and $p(\bar{\boldsymbol{k}}) = \prod_i p(\boldsymbol{k}^i)$ (Dinh et al., 2014). Effectively, we approximate the relationship between factors of $\bar{\boldsymbol{y}}$ by learning the relation between conditional distribution of independent factors over the same observation domain. Although such an approximation may not exist for priors, it is effective when dealing with conditional distributions. As we demonstrate below, TzK can learn meaningful representations of the joint knowledge $p(\bar{\boldsymbol{y}})$.

8.3.3 Related Work

Product of experts (PoE, Hinton (2002b)) is a LVM with a decoding distribution which resembles the decoder here (see Eq. (8.4)). PoE utilizes "experts" (*i.e.*, conditional distributions over observations), where each expert is conditioned on a binary hidden state which determine whether it is active or not. By multiplying the experts, PoE can model sharp multi-modal distributions, allowing each expert to capture different aspects of the data. Despite showing promising results, PoE requires a specialized training procedure (*i.e.*, contrastive divergence), which can be used with distributions with intractable partition functions (*i.e.*, as an alternative to maximum-likelihood training). Unfortunately, contrastive divergence has been demonstrated to be rather challenging, and unstable in practice. Here, we propose to train a normalized model (*i.e.*, TzK) with MIM learning, where the partition function can be efficiently estimated with the help of the approximated priors. As a result, the training of TzK is stable, scalable to large datasets (with MC sampling and SGD), and to high dimensional data (when using the reparameterization trick).

A more recent model, spike-and-slab sparse coding (S3C, Goodfellow et al. (2012)) addressed some of the challenges in PoE. S3C utilizes discrete latent variables (spike variables) and continuous low dimensional latent variables (slab variables), similar to TzK . Here, the

authors demonstrated inference in large scale problems (*e.g.*, object recognition) by training the model with a novel EM variational learning procedure. The resulting decoder is similar to the decoder used here (*i.e.*, mixed continuous and discrete latent variables), and the S3C model manages to scale inference by introducing a tractable partition function (similar in spirit to the approach taken here). Nevertheless, S3C differs from TzK in the graphical model (*i.e.*, continuous latent variables are conditioned on the discrete latent variables in S3C, unlike here), and in the learning procedure. In particular, the learning procedure relies on variational inference using KL divergence, and as such is likely to suffer from the associated disadvantages, as discussed earlier in this thesis.

8.4 Experiments

To demonstrate the versatility of TzK we train on up to six image datasets (Table 8.1), in unsupervised and semi-supervised settings. All images were resized to 32×32 as needed, MNIST images were centered and padded to 32×32. When using grayscale (GS) images in an RGB setting, the GS channel was duplicated in R, G, and B channels.

In all experiments below the images, $\boldsymbol{t}$, and class labels, e^i, for different tasks are given. The latent codes, $\boldsymbol{c}^i$, are not. In this semi-supervised context we sample the missing $\boldsymbol{c}^i$ according to the model, $\boldsymbol{c}^i \sim \mathcal{M}_{\boldsymbol{\theta}}$. Specifically, at every mini-batch, we randomly choose $q_{\boldsymbol{\theta}}$ or $p_{\boldsymbol{\theta}}$ with equal probability. When $p_{\boldsymbol{\theta}}$ is chosen we sample from $p(\boldsymbol{c}^i)$, and for $q_{\boldsymbol{\theta}}$ we return the marginal over $q(\boldsymbol{c}^i|\mathrm{e}^i, \boldsymbol{t})$ with respect to the observed binary variable e^i.

We chose CIFAR10 and MNIST as targets for conditional model learning. Each comprises just 3.2% of the entire multi-data training set of 1,892,916 images. Table 8.2 gives performance benchmarks in terms of negative log-likelihood in bits per dimension (NLL) for existing flow-based models.

All learning occurred in an online fashion, adding new conditional knowledge types as needed. When training begins, we start with a model with no knowledge, *i.e.*, $C = 0$, which is just a Glow probability density estimator. As data are sampled for learning, new knowledge

Dataset	**Image Format**	**# Images train / val**	%	**Classes**
CIFAR10	32×32 RGB	50,000 / 10,000	3.2	10
MNIST	28×28 GS	60,000 / 10,000	3.2	10
Omniglot †	105×105 RGB	19,280 / 13,180	1.7	NA
SVHN †	32×32 RGB	73,257 / 26,032	5.3	10
ImageNet †	Varying RGB	1,281,167 / 150,000	75.8	1000
Celeba †	178×218 RGB	200,000 / NA	10.8	NA

Table 8.1: Datasets marked with † were used in unsupervised settings only. GS denotes grayscale images. The *multi-data* training set consists of all six datasets, namely, CIFAR10 (Krizhevsky et al., 2009), MNIST (LeCun et al., 1998), Omniglot (Lake et al., 2015), SVHN (Netzer et al., 2011), ImageNet (Russakovsky et al., 2015), Celeba (Liu et al., 2015). There are 1,892,916 images in total. % gives each dataset's fraction of the entire multi-data training set.

	Glow	**FFJORD**	**RealNVP**	**TzK Prior**	**TzK Cond.**
CIFAR10	3.35	3.4	3.49	3.54	**2.99** *
MNIST	1.05 ††	**0.99**	1.06 ††	1.11	1.02 * †

Table 8.2: Comparison of negative log-likelihood bits per dimension (NLL) on test data (lower is better). *Results of dataset conditional prior. †Model was trained on all 6 datasets (Table 8.1). We compare to Glow (Kingma and Dhariwal, 2018), FFJORD (Grathwohl et al., 2018), and RealNVP (Dinh et al., 2016). Results marked with †† are taken from (Grathwohl et al., 2018).

types are added only when observed, in a semi-supervised manner, *i.e.*, the class label is given, the latent code $\boldsymbol{c}^i$ is not. In most of the experiments below the only class label used is the identity of the dataset from which the image was drawn.

8.4.1 Implementation

The TzK model comprises probability distributions defined in Eqs. (8.3) - (8.7). Each can be treated as a black box object with the functionality of a probability density estimator, returning $\log p(x)$ given x, and a sampler, returning $x \sim p(x)$ given $p(x)$. The specific implementation choices outlined were made for the ease and efficiency of training.

We adopt a Glow-based architecture for probability density estimators, using reparametrization (Papaspiliopoulos et al., 2003; Williams, 1992) and back-propagation with Monte Carlo (Rezende et al., 2014) for efficient gradient-based optimization (*e.g.*, see (Rezende and Mo-

Figure 8.2: Random training samples from CIFAR10 and MNIST.

hamed, 2015)). Our flow architecture used fixed shuffle permutation rather than invertible 1×1 convolution used in (Kingma and Dhariwal, 2018) as we found it to suffer from accumulated numerical error. We implemented TzK in Pytorch, using $swish\left(x\right) = x \cdot \sigma\left(x\right)$ non-linearity (Ramachandran et al., 2018) instead of ReLU as the activation function. We found that the ReLU-based implementation converged more slowly because of the truncation of gradients for negative values.

We implemented separated $p\left(\boldsymbol{c}^i|\mathrm{e}^i, \boldsymbol{t}\right)$ and $p\left(\boldsymbol{t}|\mathrm{e}^i, \boldsymbol{c}^i\right)$ for $\mathrm{e}^i \in 0, 1$ with regressors from $\boldsymbol{t}$ and $\boldsymbol{c}^i$ to parameters of distributions over $\boldsymbol{c}^i$ and $\boldsymbol{t}$. In practice, we regress to the mean and diagonal covariance of a multi-dimensional Gaussian density. We implemented $p\left(\mathrm{e}^i = 1|\boldsymbol{t}\right)$ and $p\left(\mathrm{e}^i = 1|\boldsymbol{c}^i\right)$, discriminators for binary variable e^i conditioned on $\boldsymbol{t}$ and $\boldsymbol{c}^i$ respectively, with regressors from $\boldsymbol{t}$ and $\boldsymbol{c}^i$ followed by sigmoid to normalize the output value to be in $[0, 1]$. We refer to the prior flow $p(\boldsymbol{t}|\boldsymbol{z})$ as the $\boldsymbol{t}$-flow, and the flows in each conditional prior $p(\boldsymbol{z}|\boldsymbol{k}^i)$ as a $\boldsymbol{z}$-flow.

All experiments were executed on a single NVIDIA TITAN Xp GPU with 12GB, and optimized with Pytorch ADAM optimizer (Kingma and Lei Ba, 2014), with default parameters and $lr = 1e-5$, a warm up scheduler (Vaswani et al., 2017) $warmup\_steps = 4000$, and

(a) CIFAR10 only (b) MNIST only

Figure 8.3: Random samples from two baseline models, each trained with a single dataset (CIFAR10 and MNIST). The NLL for the CIFAR10 model is 3.54. The NLL for the MNIST model is 1.11.

mini-batch size of 10. Further details are included in the supplemental material.

8.4.2 Baselines

Two baseline models are trained on CIFAR10 and MNIST, training samples for which are shown in Fig. 8.2. Each used a Glow architecture for the $\boldsymbol{t}$-flow, with 512 channels, 32 steps, and 3 layers. (See Kingma and Dhariwal (2018) for more details.) These models give test NLL values of 3.54 and 1.11, comparable to the state-of-the-art with flow-based models. Differences between our NLL numbers and those reported for Glow by others in Table 8.2 are presumably due to implementation and optimization details. Fig. 8.3 shows random samples from the two models, the quality of which compare well with training samples (Fig. 8.2).

When we train the same architecture on all 6 datasets (*i.e.*, multi-data), we obtain NLL of 3.6 when testing on CIFAR10. Random samples from this model are shown in Fig. 8.4a. One can clearly see the greater diversity of the training data, with images resembling faces and grayscale characters for example. When the same architecture is trained on the union

(a) multi-data (b) MNIST+Omniglot

Figure 8.4: Samples from a model trained on all six datasets (8.4a), and from one trained on MNIST+Omniglot (8.4b). Sample quality is similar to models trained solvely on CIFAR10 and MNIST (Fig. 8.3), despite slightly higher NLL (3.6 for multi-data model and 1.28 for MNIST+Omniglot model). Samples are more diverse, however, reflecting the greater heterogeneity of the training data.

of MNIST and Omniglot, and tested on MNIST, the NLL is 1.28. Random samples of this model (Fig. 8.4b) again show greater diversity. Although the NLL numbers with these models, both learned from larger training sets, are slightly worse, the image quality remains similar to models trained on a single dataset (Fig. 8.3).

8.4.3 Interpolation - Visualizing Flow Expressiveness

Insight into the nature of the generative model can be gleaned from latent space interpolation. Here, given four images (observations $\boldsymbol{t}$), we obtain latent space coordinates, $\boldsymbol{z} = f_{\boldsymbol{t}}^{-1}(\boldsymbol{t})$. We then linearly interpolate in $\boldsymbol{z}$ before mapping back to $\boldsymbol{t}$ for visualization. In a flow-based generative model with a Gaussian prior on $\boldsymbol{z}$, we expect interpolated points to have probability density as high or higher than the end points, and at least as high as one of the two endpoints.

Despite Glow being a powerful model, the results in Fig. 8.5 reveal deficiencies. Training

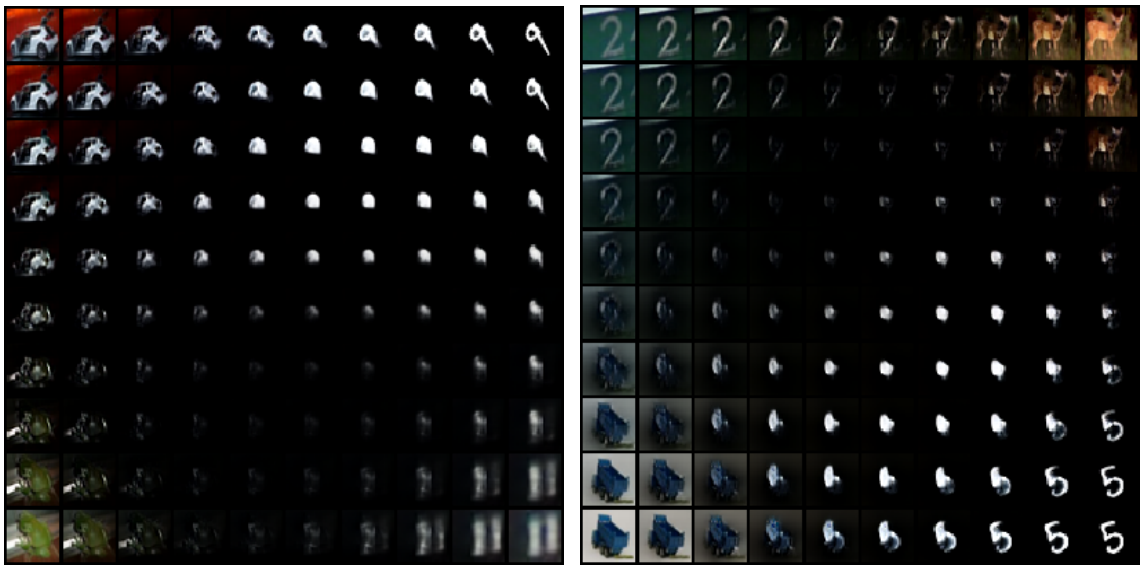


Figure 8.5: Given a model trained solely on CIFAR10, these images depict interpolation in $\boldsymbol{z}$ between random samples from CIFAR10, MNIST, Omniglot, and SVHN. The interpolation reveals regions in $\boldsymbol{z}$ which correspond to relatively poor quality images. This occurs even when the interpolated images are visually similar, and reflects relatively sparse coverage of the high-dimensional image space.

on CIFAR10 data produces a model that yields interpolated images that are not always characteristic of CIFAR10 (e.g., the darkened images in Fig. 8.5). Even with color images (*i.e.*, SVHN), which are expected to be represented reasonably well by a CIFAR10 model, there are regions of low quality interpolants.

One would suspect that a model trained on the entire multi-data training set, rather than just CIFAR10, would yield a better probability flow, exhibiting denser coverage of image space. Consistent with this, Fig. 8.6 shows superior interpolation in $\boldsymbol{z}$.

8.4.4 Specializing a *t*-Flow

In this section we further explore the benefits of unsupervised training over large heterogeneous datasets and the use of TzK for learning conditional models in an online manner. To that end, we assume a $\boldsymbol{t}$-flow has been learned and then remains fixed while we learn one or more conditional models, as one might with unknown downstream tasks. The Glow-like architecture used for the $\boldsymbol{t}$-flow (*i.e.*, for $p(\boldsymbol{t}|\boldsymbol{z})$) had 512 channels, 20 steps, and 3 layers, a

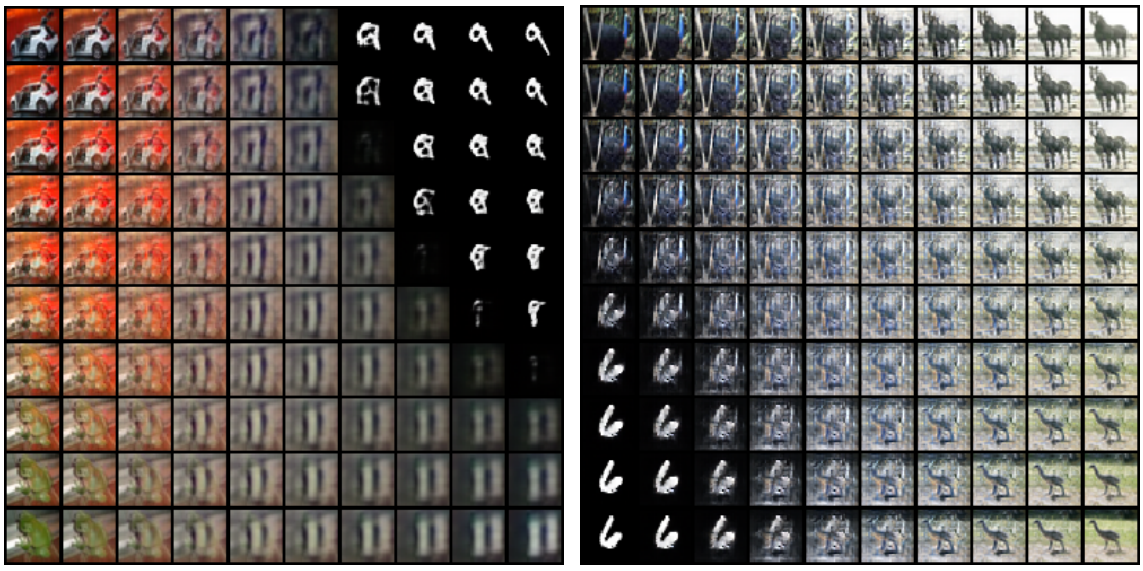
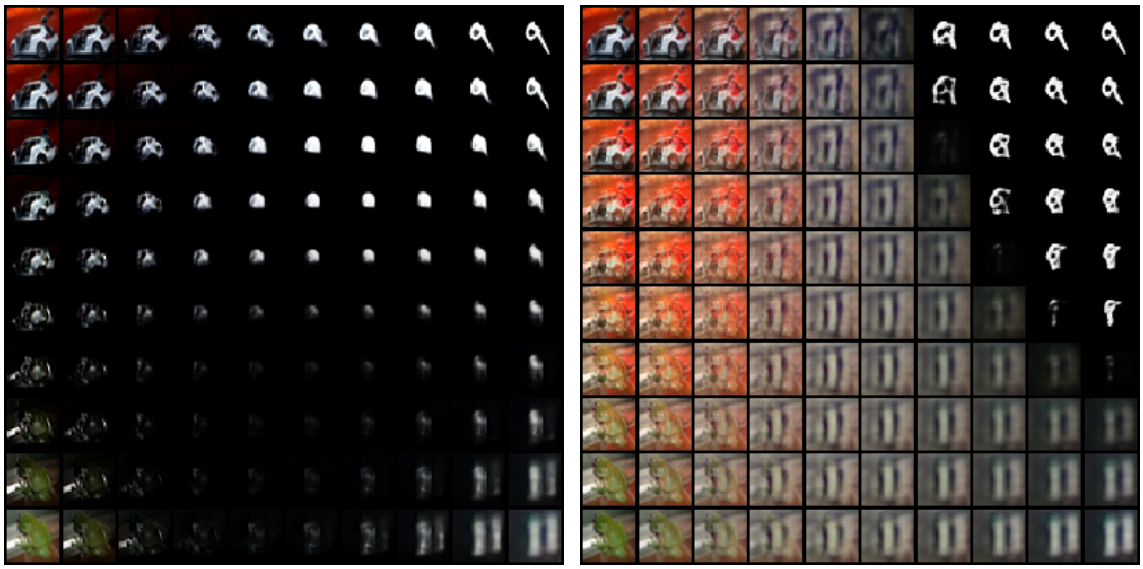


Figure 8.6: Given a model trained on all six datasets (multi-data), the interpolation results are much better than those above in Fig. 8.5. Adding more data, not surprisingly, yields a denser model with visually better interpolation.

weaker model than those use by Kingma and Dhariwal (2018) and the baseline models above with 3 layers of 32 steps. The architecture used for the $\boldsymbol{z}$-flow, for each of the conditional models (*i.e.*, for $p(\boldsymbol{z}|\boldsymbol{k}^i)$), had one layer with just 4 steps.

In the first experiment the $\boldsymbol{t}$-flow is trained solely on CIFAR10 data, entirely unsupervised. The $\boldsymbol{t}$-flow was then frozen, and conditional models were learned, one for CIFAR10 and one for MNIST. Doing so exploits just one bit of supervisory information, namely, whether each training image originated from CIFAR10 or MNIST. Although this is a relatively weak form of supervision, the benefits are significant. The MNIST images serve as negative samples for the conditional CIFAR10 model, and *vice versa*. This allows the discriminators of the respective conditional models to learn tight conditional distributions.

Indeed, the resulting CIFAR10 conditional model exhibits a significant performance gain, with a NLL of 2.99 when evaluated on the CIFAR10 test set, at or better than state-of-the-art for CIFAR10, and a great improvement over the baseline $\boldsymbol{t}$-flow (with 20 steps per layer), the NLL for which was 3.71 on the same test set. Fig. 8.8a shows random samples from the conditional model.

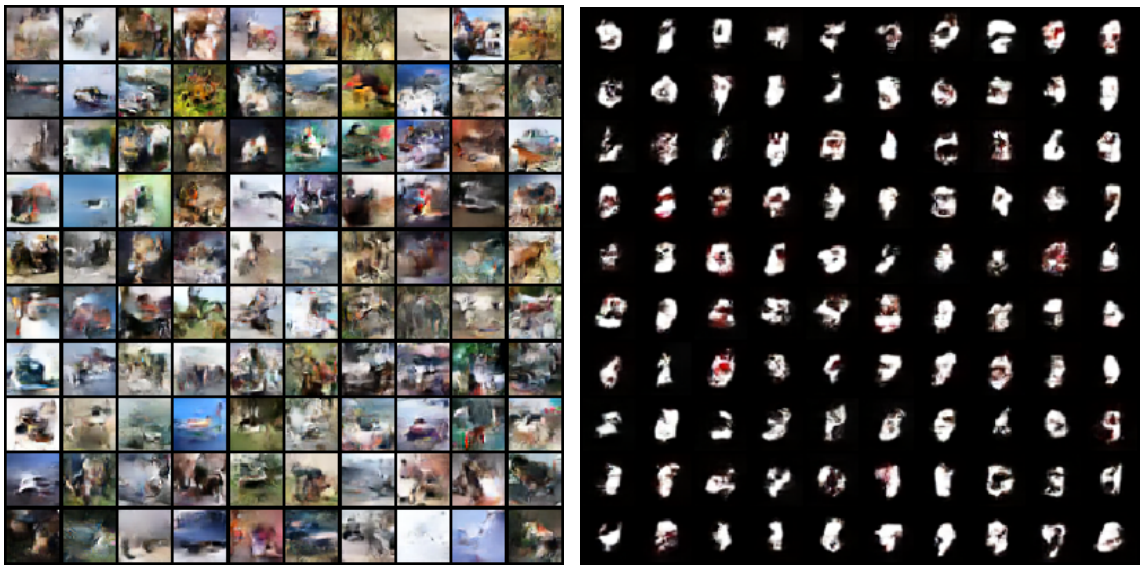


(a) CIFAR10 (b) multi-data

Figure 8.7: Given a model trained solely on CIFAR10, 8.5 depicts interpolation in $\boldsymbol{z}$ between random samples from CIFAR10, MNIST, and SVHN. Interpolation reveals regions of $\boldsymbol{z}$ that correspond to relatively poor quality images. This occurs even when the interpolated images are visually similar, and reflects relatively sparse coverage of the high-dimensional image space. Given a model trained on all six datasets (multi-data), the interpolation results in 8.6 are much better than those above in 8.7a. With more training data we obtain a denser model with visually better interpolation.

Just as surprising is the performance of the MNIST conditional, even though the CIFAR10 data on which the $\boldsymbol{t}$-flow was trained did not contain images resembling the grayscale data of MNIST. Despite this, the conditional model was able to isolate regions of the latent space representing predominantly grayscale MNIST-like images, random samples of which are shown in Fig. 8.8b. When evaluated on MNIST data, the conditional model produced a NLL 1.33. While these results are impressive, one would not expect a flow trained on CIFAR10 to provide a good latent representation for many different image domains, like MNIST.

In the next experiment we train a much richer $\boldsymbol{t}$-flow from the entire multi-data training set of 1,892,916 images, again unsupervised. Once frozen, we again learn conditional models for CIFAR10 and MNIST. Despite MNIST and Omniglot representing a small fraction of the training set, the MNIST conditional model exhibits state-of-the-art performance, with NLL of 1.02 on the MNIST test set. Random samples of the model are shown in Fig. 8.9b.



(a) CIFAR10 conditional (b) MNIST conditional

Figure 8.8: The ability of TzK to learn tight conditional priors is demonstrated here by freezing a $\boldsymbol{t}$-flow trained on CIFAR10 only, then learning conditional priors using CIFAR10 and MNIST. Random samples from the CIFAR10 conditional are shown in 8.8a. When tested on CIFAR10, the NLL for this model is just 2.99. Random samples from the MNIST conditional, in 8.8b, are surprisingly good given that MNIST data was not used to learn the $\boldsymbol{t}$ flow. The NLL for the MNIST conditional, tested on MNIST, is 1.33.

Similarly, the CIFAR10 conditional model exhibits state-of-the-art performance, with NLL 3.1. While slightly worse than the model trained from CIFAR10, it is still much better than our benchmark $\boldsymbol{t}$-flow, with 3 layers of 32 steps, and NLL of 3.54. Random samples from this CIFAR10 conditional model are shown in Fig. 8.9a.

In terms of cost, the time required to train the conditional models is roughly half the time needed to train our baseline $\boldsymbol{t}$-flow model (or equivalently Glow). Freezing the $\boldsymbol{t}$-flow allows the training procedure to be asynchronous for all conditional priors, resulting in significant gains in training time, while still maintaining a model of the joint probability. That is, conditional models can be trained in parallel so the training does not scale with the number of knowledge types.

(a) CIFAR10 conditional (b) MNIST conditional

Figure 8.9: TzK offers a powerful framework to specialize a generative flow model trained in an unsupervised fashion on a large heterogeneous dataset. By learning tight conditional priors, these models are comparable to those trained end-to-end on a single dataset. Here, we train two conditional priors concurrently. Although trained concurrently, samples share the same latent representation $\boldsymbol{z}$. The NLL for CIFAR10 (8.9a) is 3.1. The NLL for MNIST (8.9b) is 1.02.

8.5 Conclusions

Here we introduces a versatile conditional generative model based on probability flows. It supports compositionality without a priori knowledge of the number of classes or the relationships between classes. Trained with MIM learning, it provides efficient inference and sampling from class-conditionals or the joint distribution. This allows one to train generative models from multiple heterogeneous datasets, while retaining strong prior models over subsets of the data (e.g., from a single dataset, class label, or attribute).

The resulting model is efficient to train in parallel (*i.e.*, in two phases: unsupervised flow followed by conditional models), and has no hyper-parameters to tune. In addition, TzK offers an alternative motivation for the use of MI in ML models, as a natural term that arises given the assumption that the joint distributions over observation and multiple latent codes has two equally plausibly factorization of encoder and decoder. Our experiments focus on

models learned from six different image datasets, with a relatively weak Glow architecture, conditioning on various types of knowledge, including the identity of the source dataset, or class labels. This yields log likelihood comparable to state-of-the-art, with compelling samples from conditional priors.

We note that the work presented in this chapter is preliminary, and more experiments are needed to explore compositionality and tasks that could exploit such conditional representations. Nevertheless, we find the empirical results to date encouraging.

Chapter 9

Conclusions

This thesis introduces the Mutual Information Machine (MIM), a novel encoder-decoder latent variable model (LVM). The model is motivated by three key principles, namely, low divergence between the encoding and decoding distributions, high mutual information between observations and latent states, and low entropy for marginal distributions. In this dissertation we demonstrate that MIM learning is effective for both continuous and discrete data. We also demonstrate state-of-the-art results on high dimensional image data, and language time series. To the best of our knowledge, this is a first for an LVM to outperform state of the art autoregressive estimators in language modelling.

MIM learnings is rigorously described in Chapter 3. We thoroughly probe the effects of various term in MIM loss. We also demonstrate how MIM achieves superior representation, when compared to VAE, while providing similar sampling and reconstruction capabilities on multiple image datasets.

We further investigate posterior collapse in Chapter 5. We postulate that posterior collapse in VAE is the result of a low mutual information regularizer in VAE loss, and support the claim empirically. In particular we show in Chapter 6 that MIM does not suffer from posterior collapse in language modelling. We do so by training a latent variable model which manages to achieve state of the art perplexity results. To the best of our knowledge, this is the first LVM for text modelling that achieves competitive performance with non-LVM

models.

We conclude the investigation of MIM by introducing TzK in Chapter 8. TzK is a conditional model that exploit MIM's clustered representation in order to achieve state of the art results for flow-based models and image data. TzK demonstrates how to combine unsupervised learning with relatively low amounts of supervision in order to learn a state-of-the-art probability density estimator for images.

MIM was introduced in this thesis, and opened the door for multiple future research directions. In particular, further investigation of the relation between amortized and stochastic variational inference in MIM can lead to a better understanding of the importance of MIM's principles, with emphasis on model symmetry.

Additional future research directions relate to applications of MIM. MIM provides a learning framework of a new estimator, similar to VAE, and as such, there is great potential for use in multiple fields and application domains. In particular, MIM showed promising results in language modelling, and future research regarding stronger architecture and various language-related tasks is required.

We also note that further investigation of conditional MIM is required. Here we explored TzK with images. Other domains, such as language modelling, might yield stronger results. Additional research directions include compositionality in such conditional models. This thesis did not investigate this particular aspect.

We hope that this thesis will inspire others to build on top of MIM, helping to pursue some of the research directions outlined above. MIM has demonstrated great potential as a representation learning model. We hope that the ideas presented here will advance us one step closer to achieving artificial general intelligence.

Appendix A

MIM: Derivations for Formulation

In what follows we provide detailed derivations of key elements of the formulation in the chapter, namely, Equations (3.10), (3.14), (3.16), and (3.17). We also consider the relation between MIM based on the Jensen-Shannon divergence and the symmetric KL divergence.

A.1 JSD and Entropy Objectives

First we develop the relation in Eqn. (3.10), between Jensen-Shannon divergence of the encoder and decoder, the average joint entropy of the encoder and decoder, and the joint entropy of the mixture distribution $\mathcal{M}_{\mathcal{S}}$.

The Jensen-Shannon divergence with respect to the encoding distribution $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})$ and the decoding distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})$ is defined as

$$\begin{aligned}\mathrm{JSD}(\boldsymbol{\theta}) &= \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, \mathcal{M}_{\mathcal{S}}\right)\right) \\ &= \frac{1}{2}\Bigg(CE\left(\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,\mathcal{M}_{\mathcal{S}}\,\right) - H\left(\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\right) \\ &\qquad + CE\left(\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,\mathcal{M}_{\mathcal{S}}\,\right) - H\left(\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\right)\Bigg)\end{aligned}$$

Where $\mathcal{M}_{\mathcal{S}}(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) + q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}))$ is a mixture of the encoding and decoding distributions. Adding $\mathrm{R_H}(\boldsymbol{\theta}) = \frac{1}{2}\big(H\left(\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\right) + H\left(\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\right)\big)$ to the JSD term

gives

$$\begin{aligned}\mathrm{JSD}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta}) &= \frac{1}{2}\left(CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,\mathcal{M}_{\mathcal{S}}\right) + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,\mathcal{M}_{\mathcal{S}}\right)\right)\\ &= H\left(\mathcal{M}_{\mathcal{S}}\right)\end{aligned}$$

A.2 MIM Consistency

Here we discuss in greater detail how the learning algorithm encourages consistency between the encoder and decoder of a MIM model, beyond the fact that they are fit to the same sample distribution. To this end we expand on several properties of the model and the optimization procedure.

A.2.1 MIM consistency objective

In what follows we derive the form of the MIM consistency term, $\mathrm{R_{MIM}}(\boldsymbol{\theta})$, given in Eqn. (3.14). Recall that we define $\mathcal{M}_{\boldsymbol{\theta}} = \frac{1}{2}(p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) + q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}))$. We can show that $\mathcal{L}_{\mathrm{MIM}}$ is equivalent to $\mathcal{L}_{\mathrm{CE}}$ plus a regularizer by taking their difference.

$$\begin{aligned}\mathrm{R_{MIM}}(\boldsymbol{\theta}) &= \mathcal{L}_{\mathrm{MIM}}(\boldsymbol{\theta}) - \mathcal{L}_{\mathrm{CE}}(\boldsymbol{\theta})\\ &= \frac{1}{2}(CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + CE\left(\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)) - CE\left(\mathcal{M}_{\mathcal{S}}\,,\,\mathcal{M}_{\boldsymbol{\theta}}\right)\\ &= \frac{1}{2}(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + H\left(\mathcal{M}_{\mathcal{S}}\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + H\left(\mathcal{M}_{\mathcal{S}}\right))\\ &\quad - \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,\mathcal{M}_{\boldsymbol{\theta}}\right) - H\left(\mathcal{M}_{\mathcal{S}}\right)\\ &= \frac{1}{2}(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)) - \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}}\,\|\,\mathcal{M}_{\boldsymbol{\theta}}\right)\end{aligned}$$

where $\mathrm{R_{MIM}}(\boldsymbol{\theta})$ is non-negative, and is zero only when the encoding and decoding distributions are consistent (*i.e.*, they represent the same joint distribution). To prove that $\mathrm{R_{MIM}}(\boldsymbol{\theta}) \geq 0$ and to derive Eqn. (3.16), we now construct Eqn. (3.14) in terms of expectation over a joint

distribution, which yields

$$\begin{aligned}
\mathrm{R}_{\mathrm{MIM}}(\boldsymbol{\theta}) &= \frac{1}{2}(CE\left(\mathcal{M}_{\mathcal{S}}, p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\right) + CE\left(\mathcal{M}_{\mathcal{S}}, q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\right)) - CE\left(\mathcal{M}_{\mathcal{S}}, \mathcal{M}_{\boldsymbol{\theta}}\right) \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim\mathcal{M}_{\mathcal{S}}}\left[-\frac{1}{2}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \frac{1}{2}\log q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + \log\frac{1}{2}(q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))\right] \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim\mathcal{M}_{\mathcal{S}}}\left[-\log\sqrt{q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \cdot p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} + \log\frac{1}{2}(q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))\right] \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim\mathcal{M}_{\mathcal{S}}}\left[-\log\frac{\sqrt{q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \cdot p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}}{\frac{1}{2}(q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))}\right] \;\geq\; 0
\end{aligned}$$

where the inequality follows Jensen's inequality, and equality holds only when $q_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ (*i.e.*, encoding and decoding distributions are consistent).

A.2.2 Self-Correcting Gradient

One important property of the optimization follows directly from the difference between the gradient of the upper bound $\mathcal{L}_{\mathrm{MIM}}$ and the gradient of the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$. By moving the gradient operator into the expectation using reparametrization, one can express the gradient of $\mathcal{L}_{\mathrm{MIM}}(\boldsymbol{\theta})$ in terms of the gradient of $\log\mathcal{M}_{\boldsymbol{\theta}}$ and the regularization term in Eqn. (3.14). That is, with some manipulation one obtains

$$\frac{\partial}{\partial\boldsymbol{\theta}}\left(\frac{\log q_{\boldsymbol{\theta}} + \log p_{\boldsymbol{\theta}}}{2}\right) = \frac{\partial}{\partial\boldsymbol{\theta}}\log\left(\frac{q_{\boldsymbol{\theta}} + p_{\boldsymbol{\theta}}}{2}\right) + \frac{1}{2}\frac{\left(\frac{p_{\boldsymbol{\theta}}}{q_{\boldsymbol{\theta}}} - 1\right)\frac{\partial}{\partial\boldsymbol{\theta}}q_{\boldsymbol{\theta}} + \left(\frac{q_{\boldsymbol{\theta}}}{p_{\boldsymbol{\theta}}} - 1\right)\frac{\partial}{\partial\boldsymbol{\theta}}p_{\boldsymbol{\theta}}}{q_{\boldsymbol{\theta}} + p_{\boldsymbol{\theta}}}, \quad \text{(A.1)}$$

which shows that for any data point where a gap $q_{\boldsymbol{\theta}} > p_{\boldsymbol{\theta}}$ exists, the gradient applied to $p_{\boldsymbol{\theta}}$ grows with the gap, while placing correspondingly less weight on the gradient applied to $q_{\boldsymbol{\theta}}$. The opposite is true when $q_{\boldsymbol{\theta}} < p_{\boldsymbol{\theta}}$. In both case this behaviour encourages consistency between the encoder and decoder. Empirically, we find that the encoder and decoder become reasonably consistent early in the optimization process.

A.2.3 Numerical Stability

Instead of optimizing an upper bound $\mathcal{L}_{\text{MIM}}$, one might consider a direct optimization of $\mathcal{L}_{\text{CE}}$. Earlier we discussed the importance of the consistency regularizer in $\mathcal{L}_{\text{MIM}}$. Here we motivate the use of $\mathcal{L}_{\text{MIM}}$ from a numerical perspective point of view. In order to optimize $\mathcal{L}_{\text{CE}}$ directly, one must convert $\log q_{\boldsymbol{\theta}}$ and $\log p_{\boldsymbol{\theta}}$ to $q_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}}$. Unfortunately, this is has the potential to produce numerical errors, especially with 32-bit floating-point precision on GPUs. While various tricks can reduce numerical instability, we find that using the upper bound eliminates the problem while providing the additional benefits outlined above.

A.2.4 Tractability

A linear mixture vis the JSD is not the only way one might combine the encoder and decoder in a symmetric fashion. An alternative to MIM, explored in (Bornschein et al., 2015), is to use a product; i.e.,

$$\mathcal{M}_{\boldsymbol{\theta}} = \frac{1}{\beta}\sqrt{q_{\boldsymbol{\theta}}\, p_{\boldsymbol{\theta}}}\ , \tag{A.2}$$

where $\beta = \int \sqrt{q_{\boldsymbol{\theta}}\, p_{\boldsymbol{\theta}}}\, d\boldsymbol{x}\, d\boldsymbol{z}$ is the partition function. One can then define the objective to be the cross-entropy as above with a regularizer to encourage β to be close to 1, and hence to encourage consistency between the encoder and decoder. This, however, requires a good approximation to the partition function. Our choice of $\mathcal{M}_{\boldsymbol{\theta}}$ avoids the need for a good value approximation by using reparameterization, which results in unbiased low-variance gradient, independent of the accuracy of the approximation of the value.

A.3 MIM Loss Decomposition

Here we show how to break down the $\mathcal{L}_{\text{MIM}}$ into the set of intuitive components given in Eqn. (3.17). To this end, first note the definition of $\mathcal{L}_{\text{MIM}}$:

$$\mathcal{L}_{\text{MIM}}(\boldsymbol{\theta}) \ = \ \frac{1}{2}(CE\,(\,\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\,) + CE\,(\,\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\,)) \tag{A.3}$$

We will focus on the first half of Eqn. (A.3) for now,

$$\frac{1}{2}CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) \;=\; \frac{1}{4}\Bigg(CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)\Bigg) \tag{A.4}$$

It will be more clear to write out the first term of Eqn. (A.4), $\frac{1}{4}CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)$ in full

$$\begin{aligned}\frac{1}{4}CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) &= -\frac{1}{4}\int_{x,z} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\log(p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z} \\ &= -\frac{1}{4}\int_{\boldsymbol{x},\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z} \\ &\quad -\frac{1}{4}\int_{\boldsymbol{x},\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\log(p_{\boldsymbol{\theta}}(\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\end{aligned}$$

We then add and subtract $\frac{1}{4}H\left(\mathcal{P}(\boldsymbol{z})\right)$, where

$$\frac{1}{4}H\left(\mathcal{P}(\boldsymbol{z})\right) \;=\; -\frac{1}{4}\int_{\boldsymbol{z}} \mathcal{P}(\boldsymbol{z})\log(\mathcal{P}(\boldsymbol{z}))\mathrm{d}\boldsymbol{z} \;=\; -\frac{1}{4}\int_{\boldsymbol{x},\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\log(\mathcal{P}(\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}$$

Then expanding $\frac{1}{4}CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + \frac{1}{4}H\left(\mathcal{P}(\boldsymbol{z})\right) - \frac{1}{4}H\left(\mathcal{P}(\boldsymbol{z})\right)$ into constituent parts and combining terms, we obtain

$$\frac{1}{4}CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) \;=\; \frac{1}{4}H\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) + \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(\mathcal{P}(\boldsymbol{z})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) \tag{A.5}$$

The second term in Eqn. (A.4) can then be rewritten as

$$\frac{1}{4}CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) \;=\; \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + \frac{1}{4}H\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) \tag{A.6}$$

Combining Eqns. (A.5) and (A.6), we get the interpretable form for Eqn. A.4, i.e.,

$$\begin{aligned}\frac{1}{2}CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) &= \frac{1}{4}\left(H\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) + H\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right)\right) \\ &\quad + \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(\mathcal{P}(\boldsymbol{z})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) + \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) \\ &= \frac{1}{2}\mathrm{R_H}(\boldsymbol{\theta}) + \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(\mathcal{P}(\boldsymbol{z})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) + \frac{1}{4}\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)\end{aligned} \tag{A.7}$$

We can use the same basic steps to derive an analogous expression for $CE\left(\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right)$ in Eqn. (A.3) and combine it with Eqn. (A.7) to get the final interpretable form:

$$\begin{aligned}\mathcal{L}_{\mathrm{MIM}}(\boldsymbol{\theta}) \;=\; \mathrm{R_H}(\boldsymbol{\theta}) \;+\; &\frac{1}{4}\Big(\,\mathcal{D}_{\mathrm{KL}}\left(\mathcal{P}(\boldsymbol{z})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{P}(\boldsymbol{x})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\Big) \\ +\; &\frac{1}{4}\Big(\,\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\right) + \mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right)\Big)\end{aligned}$$

A.4 MIM in terms of Symmetric KL Divergence

As discussed above, the VAE objective can be expressed as minimizing the KL divergence between the joint anchored encoding and anchored decoding distributions. Below we consider a model formulation using the symmetric KL divergence (SKL),

$$\mathrm{SKL}(\boldsymbol{\theta}) = \frac{1}{2}\left(\,\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right)\;,$$

the second term of which is the VAE objective. The mutual information regularizer $\mathrm{R_H}(\boldsymbol{\theta})$ given in Eqn. (3.9) can be added to SKL to obtain a cross-entropy objective that looks similar to MIM:

$$\frac{1}{2}\mathrm{SKL}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta}) = \frac{1}{2}\left(CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) + CE\left(\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right)\right)$$

When the model priors are equal to the anchors, this regularized SKL and MIM are equivalent. In general, however, the MIM loss is not a bound on the regularized SKL.

In what follows we explore the relation between SKL and JSD. In Section A.1 we showed that the Jensen-Shannon divergence can be written as

$$\begin{aligned}\mathrm{JSD}(\boldsymbol{\theta}) &= \frac{1}{2}\Big(CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,\mathcal{M}_{\mathcal{S}}\right) - H\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) \\ &\qquad + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,\mathcal{M}_{\mathcal{S}}\right) - H\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right)\Big) \\ &= \frac{1}{2}\left(CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,\mathcal{M}_{\mathcal{S}}\right) + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,\mathcal{M}_{\mathcal{S}}\right)\right) - \mathrm{R_H}(\boldsymbol{\theta})\end{aligned}$$

Using Jensen's inequality, we can bound $\mathrm{JSD}(\boldsymbol{\theta})$ from above,

$$\begin{aligned}\mathrm{JSD}(\boldsymbol{\theta}) &\leq \frac{1}{4}\Big(H\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) + CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) \\ &\qquad + H\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\Big) - \mathrm{R_H}(\boldsymbol{\theta}) \qquad \text{(A.8)} \\ &= \frac{1}{4}\Big(CE\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + CE\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) \\ &\qquad + 2\mathrm{R_H}(\boldsymbol{\theta})\Big) - \mathrm{R_H}(\boldsymbol{\theta}) \\ &= \frac{1}{4}\Big(\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right) \\ &\qquad + 4\mathrm{R_H}(\boldsymbol{\theta})\Big) - \mathrm{R_H}(\boldsymbol{\theta}) \\ &= \frac{1}{4}\left(\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\,\|\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\,\|\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right) \\ &= \frac{1}{2}\mathrm{SKL}(\boldsymbol{\theta})\end{aligned}$$

From Eqn. (A.8), if we add $\mathrm{R_H}(\boldsymbol{\theta})$ and simplify, we get

$$\frac{1}{2}\mathrm{SKL}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta}) = \frac{1}{2}\left(CE\left(\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right)$$

Interestingly, we can write this in terms of KL divergence,

$$\begin{aligned}\frac{1}{2}\mathrm{SKL}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta}) &= \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right) + H\left(\mathcal{M}_{\mathcal{S}}\right) \\ &= \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right) \\ &\quad + \mathrm{JSD}(\boldsymbol{\theta}) + \mathrm{R_H}(\boldsymbol{\theta})\end{aligned}$$

which gives the exact relation between JSD and SKL.

$$\begin{aligned}\frac{1}{2}\mathrm{SKL}(\boldsymbol{\theta}) &= \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right) + \mathrm{JSD}(\boldsymbol{\theta}) \\ &= \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x})\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{M}_{\mathcal{S}} \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z})\right)\right) \\ &\quad + \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathcal{P}(\boldsymbol{x}) \,\|\, \mathcal{M}_{\mathcal{S}}\right) + \mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\,\mathcal{P}(\boldsymbol{z}) \,\|\, \mathcal{M}_{\mathcal{S}}\right)\right)\end{aligned}$$

Appendix B

MIM: Additional Experiments

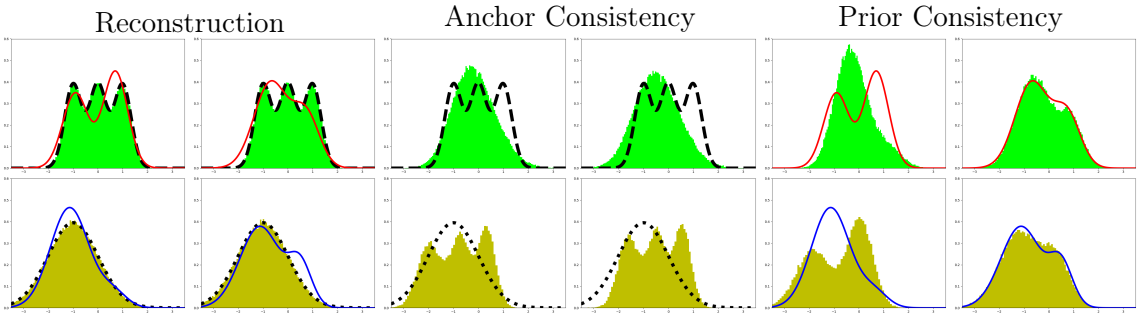
Here we provide additional experiments that further explore the characteristics of MIM learning.

B.1 Consistency regularizer in $\mathcal{L}_{\text{MIM}}$

Here we explore properties of models for 1D $\boldsymbol{x}$ and $\boldsymbol{z}$, learned with $\mathcal{L}_{\text{MIM}}$ and $\mathcal{L}_{\text{CE}}$, the difference being the model consistency regularizer $\text{R}_{\text{MIM}}(\boldsymbol{\theta})$. All model priors and conditional likelihoods ($q_{\boldsymbol{\theta}}(\boldsymbol{x})$, $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$) are parameterized as 10-component Gaussian mixture models, and optimized during training. Means and variances for the conditional distributions were regressed with 2 fully connected layers ($h \in \mathbb{R}^{10}$) and a swish activation function Ramachandran et al. (2018).

Top and bottom rows in Fig. B.1 depict distributions in the observation and latent spaces. Dashed black curves are anchors, $\mathcal{P}(\boldsymbol{x})$ on top, and $\mathcal{P}(\boldsymbol{z})$ below (GMMs with up to 3 components). Learned model priors, $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, are depicted as red (top) and blue (bottom) curves.

Green histograms in Fig. B.1(a,b) depict reconstruction distributions, computed by passing fair samples from $\mathcal{P}(\boldsymbol{x})$ through the encoder to $\boldsymbol{z}$ and then back through the decoder to $\boldsymbol{x}$. Similarly the yellow histograms shows samples from $\mathcal{P}(\boldsymbol{z})$ passed through the decoder



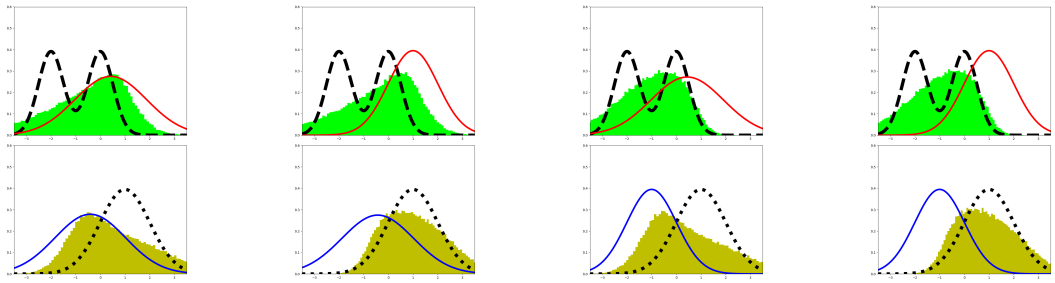
(a) CE (b) MIM (c) CE (d) MIM (e) CE (f) MIM

Figure B.1: We explore the influence of consistency regularizer $R_{\boldsymbol{\theta}}$. CE and MIM indicate the loss, $\mathcal{L}_{\text{CE}}$ or $\mathcal{L}_{\text{MIM}}$ (the regularized objective), respectively. Top row shows anchor $\mathcal{P}(\boldsymbol{x})$ (dashed), prior $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ (red), and reconstruction distribution $\boldsymbol{x}_i \sim \mathcal{P}(\boldsymbol{x}) \to \boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i) \to x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (green). Bottom row mirrors the top row, with anchor $\mathcal{P}(\boldsymbol{z})$ (dotted), prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (blue), and reconstruction distribution $\boldsymbol{z}_i \sim \mathcal{P}(\boldsymbol{z}) \to \boldsymbol{x}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i) \to z'_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (yellow). In (c-d) the reconstruction is replaced with decoding from anchors $\boldsymbol{z}_i \sim \mathcal{P}(\boldsymbol{z}) \to x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (green), and encoding $\boldsymbol{x}_i \sim \mathcal{P}(\boldsymbol{x}) \to z'_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (yellow). In (e-f) the reconstruction is replaced with decoding from priors $\boldsymbol{z}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) \to x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (green), and encoding $\boldsymbol{x}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{x}) \to z'_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (yellow). While both models offers similar reconstruction (a-b), and similar consistency w.r.t. the anchors (c-d), only MIM finds a consistent model (e-f). See text for details.

and then back to the latent space. For both losses these reconstruction histograms match the anchor priors well. In contrast, only the priors that were learned with $\mathcal{L}_{\text{CE}}$ loss approximates the anchor well, while the $\mathcal{L}_{\text{MIM}}$ priors do not. To better understand that, we consider two generative procedures: sampling from the anchors, and sampling from the priors.

Anchor consistency is depicted in Fig. B.1(c,d), where Green histograms are marginal distributions over $\boldsymbol{x}$ from the anchored decoder (i.e., samples from $\mathcal{P}(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$). Yellow are marginals over $\boldsymbol{z}$ from the anchored encoders $\mathcal{P}(\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. One can see that both losses results in similar quality of matching the corresponding opposite anchors.

Priors consistency is depicted in Fig. B.1(e,f), where Green histograms are marginal distributions over $\boldsymbol{x}$ from the model decoder $p_{\boldsymbol{\theta}}(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. Yellow depicts marginals over $\boldsymbol{z}$ from the model encoder $q_{\boldsymbol{\theta}}(\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. Importantly, with $\mathcal{L}_{\text{MIM}}$ the encoder and decoder are consistent; i.e., $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ (red curve) matches the decoder marginal, while $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (blue) matches the encoder marginal. The model trained with $\mathcal{L}_{\text{CE}}$ (*i.e.*, without consistency prior) fails to learn a consistent encoder-decoder pair. We note that in practice, with expressive enough



(a) $q_{\boldsymbol{\theta}}(\boldsymbol{x}), p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (b) $q(\boldsymbol{x}), p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (c) $q_{\boldsymbol{\theta}}(\boldsymbol{x}), p(\boldsymbol{z})$ (d) $q(\boldsymbol{x}), p(\boldsymbol{z})$

Figure B.2: MIM prior expressiveness. In this experiment we explore the effect of learning a prior, where the priors $q(\boldsymbol{x})$ and $p(\boldsymbol{z})$ are normal Gaussian distributions. Top row shows anchor $\mathcal{P}(\boldsymbol{x})$ (dashed), prior $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ (red), and decoding distribution $\boldsymbol{z}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) \rightarrow x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (green). Bottom row mirrors the top row, with anchor $\mathcal{P}(\boldsymbol{z})$ (dotted), prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (blue), and encoding distribution $\boldsymbol{x}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{x}) \rightarrow z'_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (yellow). As can be seen, parameterizing priors affects all learned distributions, supporting the notion of optimization of a single model $\mathcal{M}_{\boldsymbol{\theta}}$. We point that (a) demonstrates the best consistency between the priors and corresponding generated samples, following the additional expressiveness.

priors, $\mathcal{L}_{\text{MIM}}$ will be a tight bound for $\mathcal{L}_{\text{CE}}$.

B.2 Parameterizing the Priors

Here we explore the effect of parameterizing the latent and observed priors. A fundamental idea in MIM is the concept of a single model, $\mathcal{M}_{\boldsymbol{\theta}}$. As such, parameterizing a prior increases the global expressiveness of the model $\mathcal{M}_{\boldsymbol{\theta}}$. Fig. B.2 depicts the utilization of the added expressiveness in order to increase the consistency between the encoding and decoding model distribution, in addition to the consistency of $\mathcal{M}_{\boldsymbol{\theta}}$ with $\mathcal{M}_{\mathcal{S}}$.

B.3 Effect of Consistency Regularizer on Optimization

Here we explore whether a learned model with consistent encoding-decoding distributions (*i.e.*, trained with $\mathcal{L}_{\text{MIM}}$) also constitutes an optimal solution of a CE objective (*i.e.*, trained with $\mathcal{L}_{\text{CE}}$). Results are depicted in Fig. B.3. In order to distinguish between the effects of the optimization from the consistency regularizer we initialize a MIM model by pre-training it with $\mathcal{L}_{\text{CE}}$ loss followed by $\mathcal{L}_{\text{MIM}}$ training in Fig. B.3(i), and vice verse in Fig. B.3(ii).

(a-b,e-f) All trained models in Fig. B.3 exhibit similarly good reconstruction (green matches dashed black). (c-d,g-h) However, only models that were trained with $\mathcal{L}_{\text{MIM}}$ exhibit encoding-decoding consistency (green matches red, yellow matches blue). While it is clear that the optimization plays an important role (*i.e.*, different initialization leads to different local optimum), it is also clear that encoding-decoding consistency is not necessarily an optimum of $\mathcal{L}_{\text{CE}}$, as depicted in a non-consistent model (h) which was initialized with a consistent model (g). Not surprisingly, without the consistency regularizer training with $\mathcal{L}_{\text{CE}}$ results in better fit of priors to anchors (f) as it is utilizing the expressiveness of the parametric priors in matching the sample distribution.

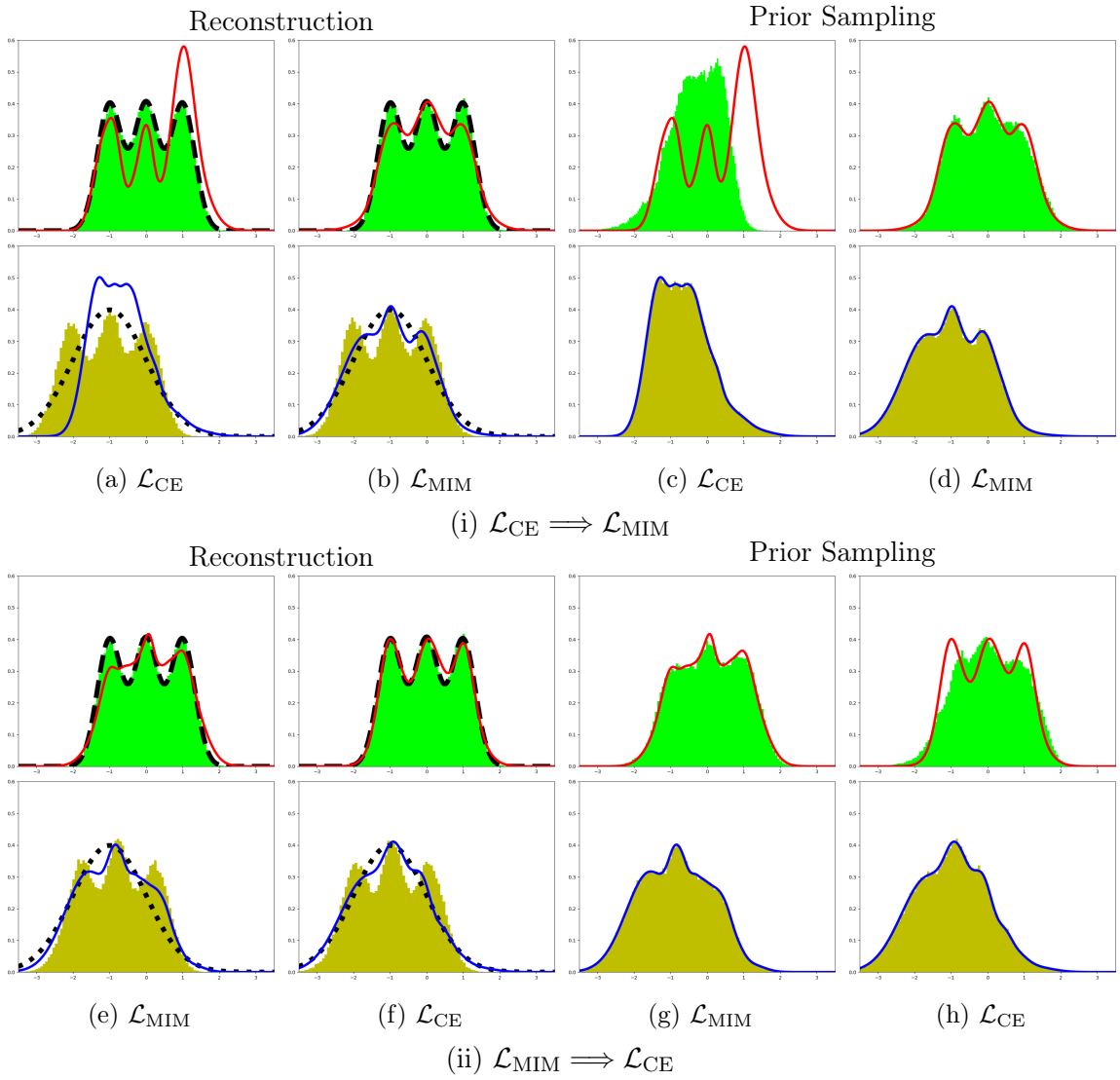


Figure B.3: Effects of MIM consistency regularizer and optimization on encoding-decoding consistency. (i) and (ii) differ in initialization order. Odd rows: anchor $\mathcal{P}(\boldsymbol{x})$ (dashed), prior $q_{\boldsymbol{\theta}}(\boldsymbol{x})$ (red). Even rows: anchor $\mathcal{P}(\boldsymbol{z})$ (dotted), prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (blue). (a-b,e-f) Reconstruction $\boldsymbol{x}_i \sim \mathcal{P}(\boldsymbol{x}) \rightarrow \boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i) \rightarrow x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (x'_i green, $\boldsymbol{z}_i$ yellow). (c-d,g-h) Prior decoding $\boldsymbol{z}_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) \rightarrow x'_i \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_i)$ (green), and prior encoding $\boldsymbol{x}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{x}) \rightarrow \boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$ (yellow). See text for details.

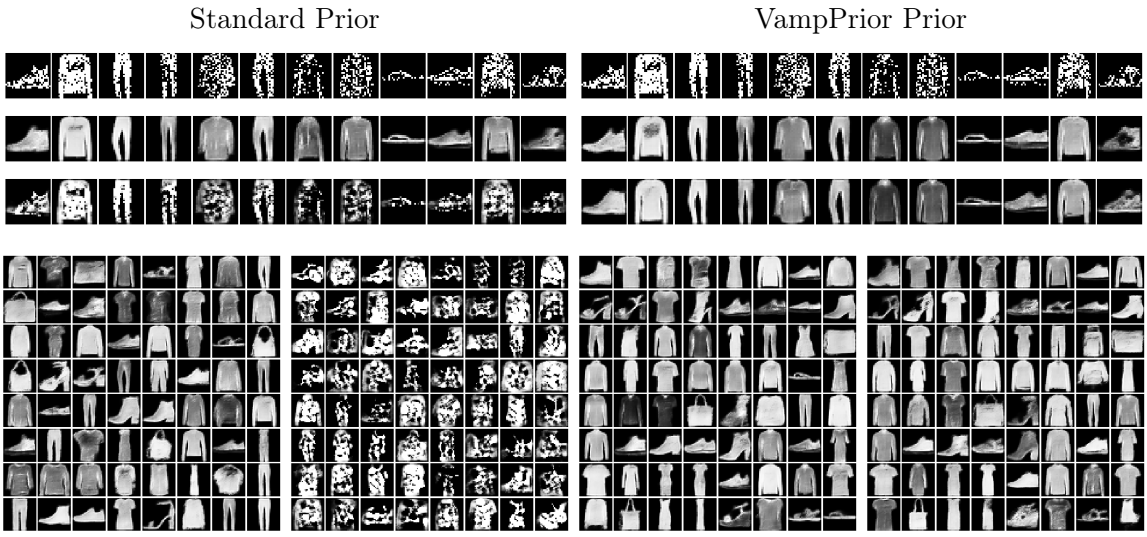
Appendix C

MIM: Additional Results

Here we provide additional visualization of various MIM and VAE models.

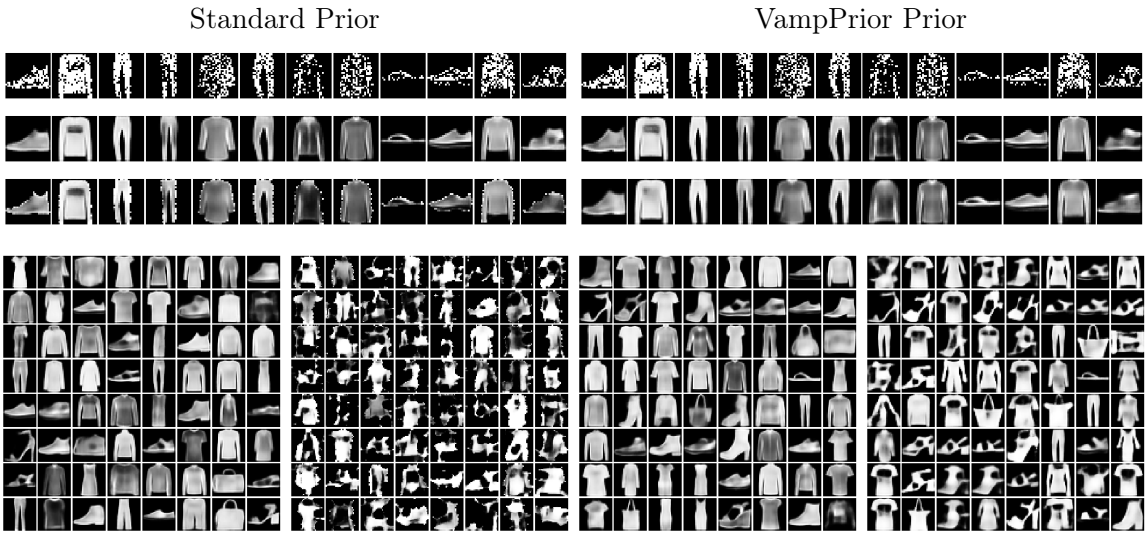
C.1 Reconstruction and Samples for MIM and A-MIM

In what follows we show samples and reconstruction for MIM (*i.e.*, with convHVAE architecture), and A-MIM (*i.e.*, with PixelHVAE architecture). We demonstrate, again, that a powerful enough encoder allows the model to generate samples which are comparable in quality to VAE samples.



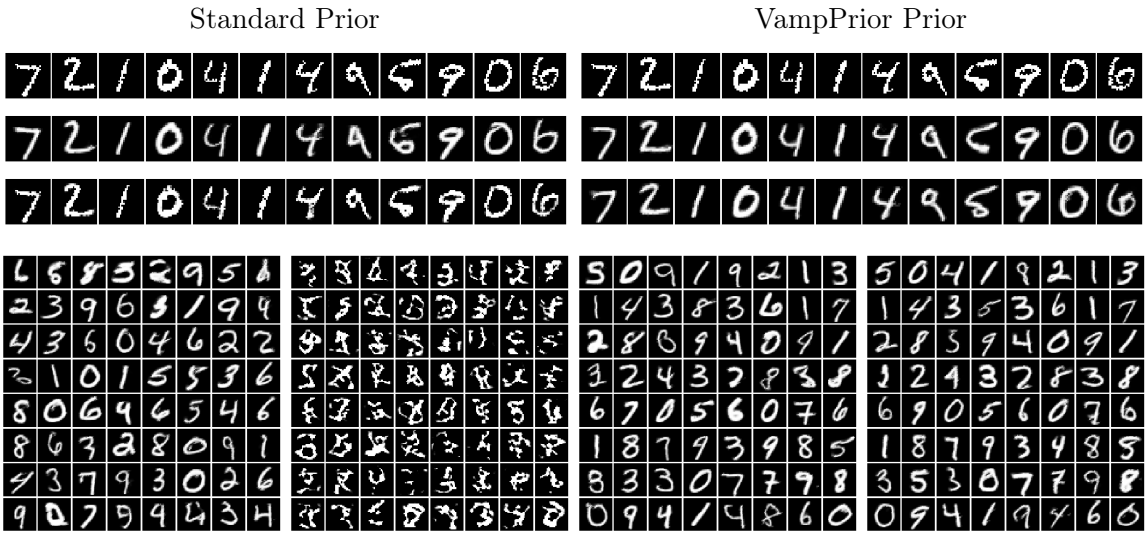
(a) VAE (S) (b) A-MIM (S) (c) VAE (VP) (d) A-MIM (VP)

Figure C.1: MIM and VAE learning with PixelHVAE for Fashion MNIST. MIM was trained with asymmetric sampling (*i.e.*, from encoding distribution only), and as such is labelled A-MIM. The top three rows (from top to bottom) are test data samples, VAE reconstruction, MIM reconstruction. Bottom: random samples from VAE and MIM. (c-d) We initialized all pseudo-inputs with training samples, and used the same random seed for both models. As a result the samples order is similar. We note that with a standard prior, MIM provides better reconstructions, whereas with Vamprior MIM offers comparable reconstructions. We consider that a result of VampPrior being a non-ideal match for MIM, as MIM tends to learn posteriors with low variance. Using Vamprior leads to increased variance, and thus a degradation in the quality of reconstructions.



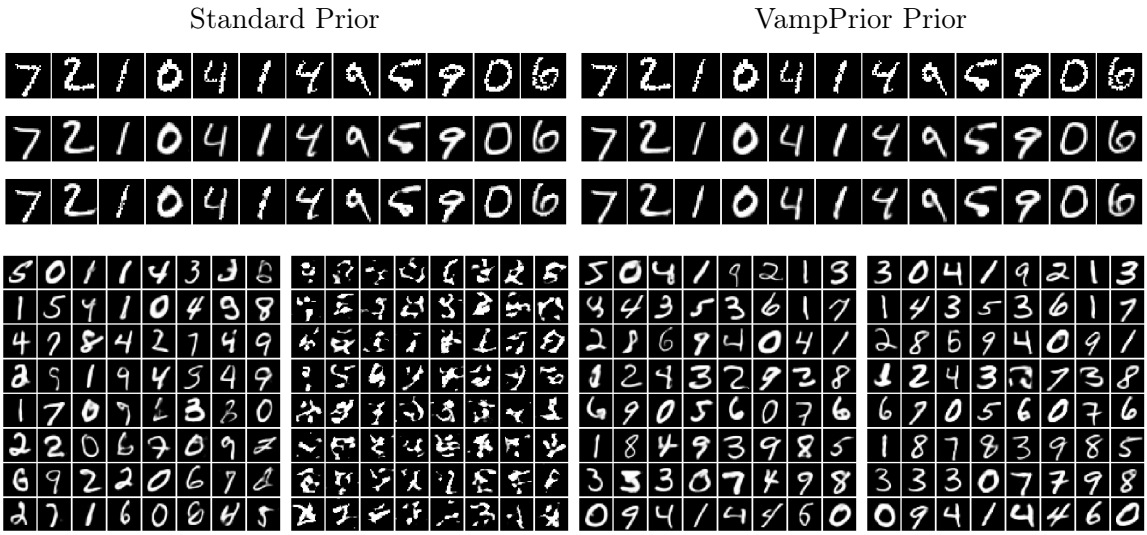
(a) VAE (S) (b) MIM (S) (c) VAE (VP) (d) MIM (VP)

Figure C.2: MIM and VAE learning with convHVAE for Fashion MNIST. The top three rows (from top to bottom) are test data samples, VAE reconstruction, MIM reconstruction. Bottom: random samples from VAE and MIM.



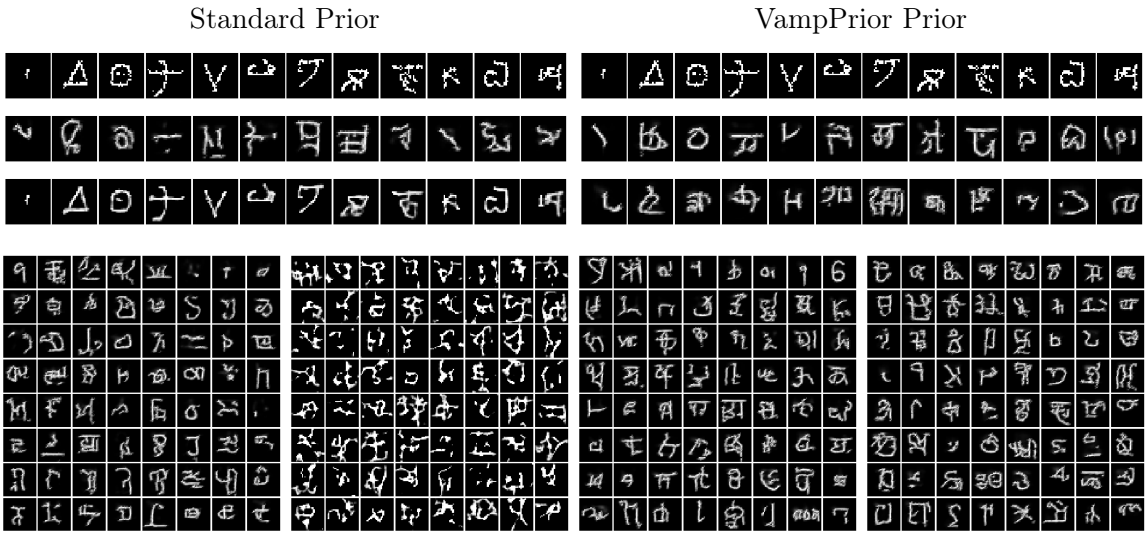
(a) VAE (S) (b) A-MIM (S) (c) VAE (VP) (d) A-MIM (VP)

Figure C.3: MIM and VAE learning with PixelHVAE for MNIST. MIM was trained with asymmetric sampling (*i.e.*, from encoding distribution only), and as such is labelled A-MIM. Top three rows are test data samples, followed by VAE and MIM reconstructions. Bottom: random samples from VAE and MIM. (c-d) We initialized all pseudo-inputs with training samples, and used the same random seed for both models. As a result the samples order is similar. We note that with a standard prior, MIM provides better reconstructions, whereas with Vamprior MIM offers comparable reconstructions. We consider that a result of VampPrior being a non-ideal match for MIM, as MIM tends to learn posteriors with low variance. Using Vamprior leads to increased variance, and thus a degradation in the quality of reconstructions.



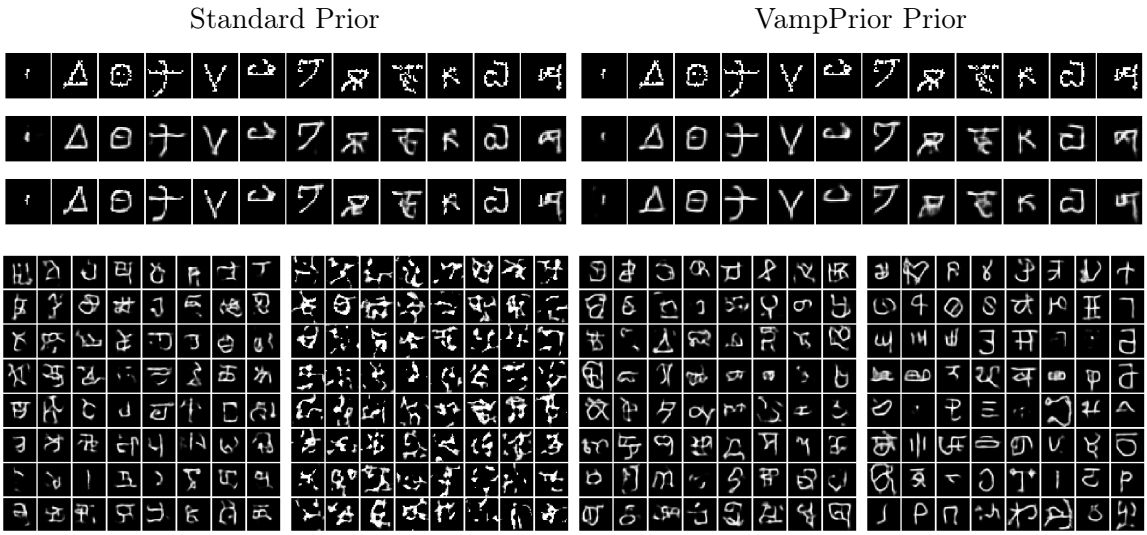
(a) VAE (S) (b) MIM (S) (c) VAE (VP) (d) MIM (VP)

Figure C.4: MIM and VAE learning with convHVAE for MNIST. Top three rows are test data samples, followed by VAE and MIM reconstructions. Bottom: random samples from VAE and MIM.



(a) VAE (S) (b) A-MIM (S) (c) VAE (VP) (d) A-MIM (VP)

Figure C.5: MIM and VAE learning with PixelHVAE for Omniglot. MIM was trained with asymmetric sampling (*i.e.*, from encoding distribution only), and as such is labelled A-MIM. Top three rows are test data samples, followed by VAE and MIM reconstructions. Bottom: random samples from VAE and MIM. We note that with a standard prior, MIM provides better reconstructions, whereas with Vamprior MIM offers comparable reconstructions. We consider that a result of VampPrior being a non-ideal match for MIM, as MIM tends to learn posteriors with low variance. Using Vamprior leads to increased variance, and thus a degradation in the quality of reconstructions.



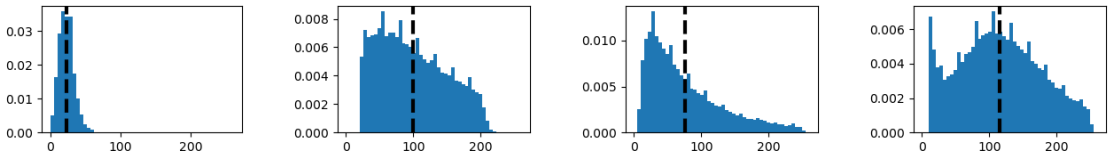
(a) VAE (S) (b) MIM (S) (c) VAE (VP) (d) MIM (VP)

Figure C.6: MIM and VAE learning with convHVAE for Omniglot. Top three rows are test data samples, followed by VAE and MIM reconstructions. Bottom: random samples from VAE and MIM.

Appendix D

SentenceMIM: Additional Experiments

D.1 Distribution of Sentence Lengths

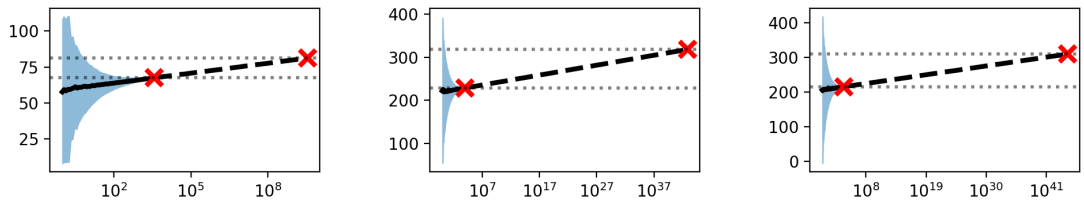


(a) PTB (b) Yelp15 (c) Yahoo Answers WT103

Figure D.1: Here we present histograms of sentence lengths per dataset. The dashed line is the average sentence length.

Fig. D.1 shows histograms of sentence lengths. Notice that PTB sentences are significantly shorter that other datasets. As a result, sMIM is somewhat better able to learn a representation that is well suited for reconstruction. Other datasets, with longer sentences, are more challenging, especially with the simple architecture used here (*i.e.*, 1 later GRU). We believe that implementing sMIM with an architecture that better handles long-term dependencies (*e.g.*, transformers) might help.

D.2 Effect of Sample Set Size on MELBO



(a) PTB (b) Yelp15 (c) Yahoo Answers

Figure D.2: Plots show the effect of sample set size (K on x axis) on NLL computed with MELBO upper bound (y axis). For each plot we draw K random samples from a test set of size N, and compute the bound (*i.e.*, denoted a trial). We repeat the trial $\max(N - K, 500)$ times, and compute the mean NLL and the standard deviation. The solid curve depicts the mean NLL; blue shade is 1 standard deviation (over multiple trials). The dashed line is the extrapolated NLL (*i.e.*, see text for details). Red cross marks are NLL values of best performing sMIM model (left mark, for $K = N$), and best performing non-sMIM model (right mark). We note that for $K \gtrapprox 10^3$ the variance in all cases cannot account for the NLL gap.

Here we consider how MELBO , as a bound on NLL, depends on number of test samples. Our goal is to empirically show that bounding NLL with MELBO is robust to the test set used, and that the bound has a reasonably low variance. Fig. D.2 shows the dependence of MELBO on the size of a test sample set. For each value of K, up to the full test set size, N, we randomly sample K points in each of several trials, and then plot the mean and standard deviation of the MELBO bound over trials. The solid line shows the mean NLL, as a function of K, the standard deviation of which is shown in blue. Once K is 1000 or more, the standard deviation is very small, indicating that the specific test sample does not have a significant effect on the bound. In particular, at that point the standard deviation is less than 3.1% of the MELBO bound.

The dashed curve is the extrapolated NLL bound, assuming the average reconstruction error remain constant. Red crosses indicate the MELBO bounds for the full test set ($K = N$) and for a test set sufficiently large that the bound equals the NLL of the best performing non-sMIM model; the required sample sizes are orders of magnitude above N. This also indicates that the sizes of existing test sets do not account for the large gap in perplexity

between sMIM and other models.

At this point we would like to remind the reader that it is unclear whether MELBO and ELBO, when used as NLL bounds, are comparable, and that our goal here is to examine the effect of the sample set size on MELBO . While we demonstrated here that the gap between MELBO and ELBO is unlikely to be explained by the sample set size, it is possible that the large NLL gap can be explained by the gap between the approximate model (*i.e.*, where the prior is constructed with samples from the target sample set) and the true model (*i.e.*, where the true prior is defined with expectation over true observations distribution). We leave further exploration of that question to future research.

D.3 Comparison of NLL in MIM and VAE

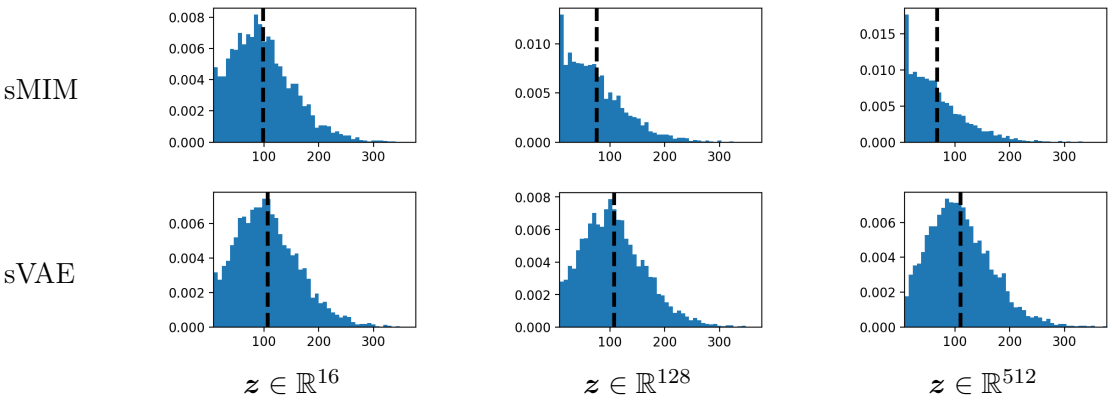


Figure D.3: Histograms of MELBO (sMIM) and ELBO (sVAE) values versus latent dimension for **PTB**. Dashed black line is the mean.

Figures D.3-D.5 depict histograms of ELBO/MELBO values for sentences, for sVAE and sMIM with different latent dimensions. While a less expressive sMIM behaves much like sVAE, the difference is clearer as the expressiveness of the model increases. Here, sVAE does not appear to effectively use the increased expressiveness for better modelling. We hypothesize that the added sVAE expressiveness is used to better match the posterior to the prior, resulting in posterior collapse. sMIM uses the increased expressiveness to increase mutual information.

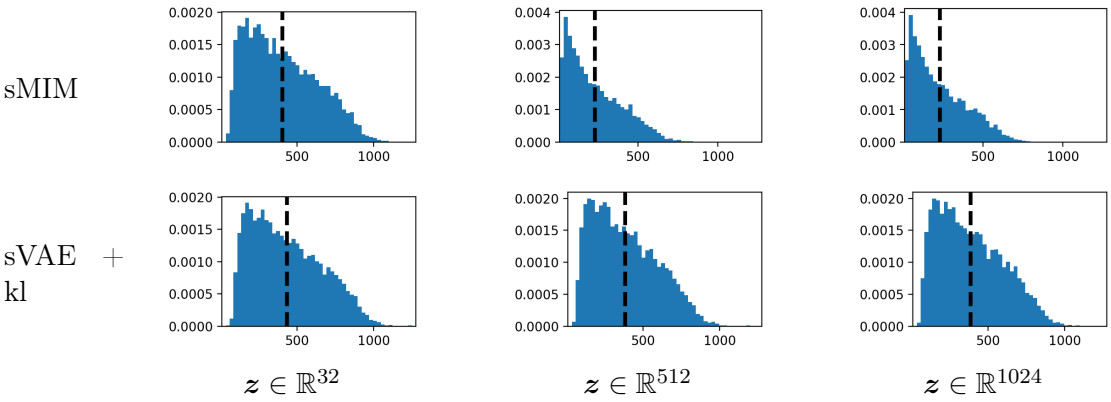


Figure D.4: Histograms of MELBO (sMIM) and ELBO (sVAE) values versus latent dimension for **Yelp15**. Dashed black line is the mean.

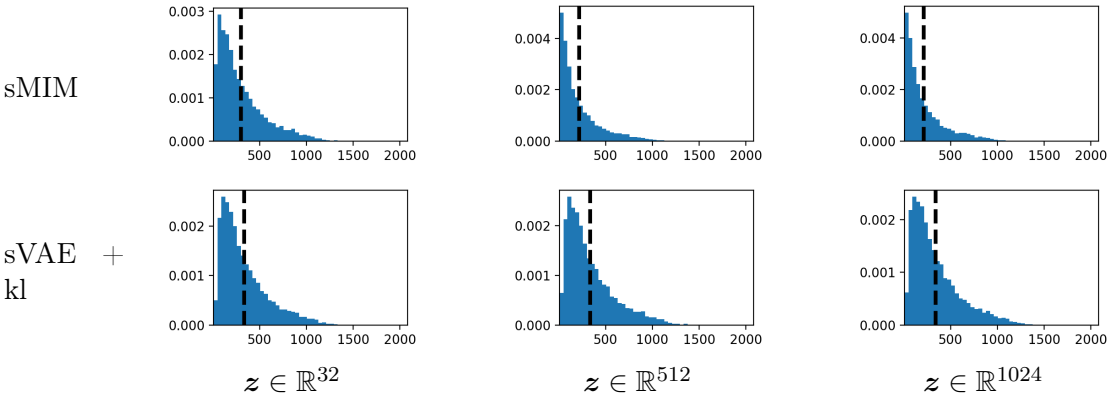


Figure D.5: Histograms of MELBO (sMIM) and ELBO (sVAE) versus latent dimension for **Yahoo Answers**.

Appendix E

SentenceMIM: Additional Results

E.1 Reconstruction

	sMIM (512)	sMIM (1024) †
(D)	<SOS> there was no panic	
(M)	there was no panic <EOS>	there was no panic <EOS>
(R)	there was no orders <EOS>	there was no panic <EOS>
(P)	there was no panic <EOS>	there was no shortage panic <EOS>
(AE)	there was no panic <EOS>	
(D)	<SOS> the company did n't break out its fourth-quarter results	
(M)	the company did n't break out its fourth-quarter results <EOS>	the company did n't break out its results results <EOS>
(R)	the company did n't break out its results <EOS>	the company did n't break out its results <EOS>
(P)	the company did n't break out its fourth-quarter results <EOS>	the company did n't break out its results results <EOS>
(AE)	the company did n't break out results <EOS>	
(D)	<SOS> it had planned a strike vote for next sunday but that has been pushed back indefinitely	
(M)	it had a weakening for promotional planned but that has pushed aside back but so far away <EOS>	it had planned planned a planned for next week but that continues has been pushed back pushed <EOS>
(R)	it had a planned strike for energy gifts but so that has planned airlines but block after six months <EOS>	it had planned a strike planned for next sunday but that has been pushed back culmination pushed <EOS>
(P)	it had a strike with stateswest airlines but so that it has slashed its spending but so far said he would be subject by far <EOS>	it had planned a strike for hardcore but has been pushed every year that leaves back <EOS>
(AE)	it had been a five-year vote but for a week that drilling humana strike back back has planned back <EOS>	

Table E.1: Reconstruction results for models trained on **PTB**. We denote: (D) Data sample; (M) Mean (latent) reconstruction; (R) Reconstruction; (P) Perturbed (latent) reconstruction; (AE) Reconstruction of AE.

Here we provide reconstruction results for PTB (Fig. E.1), Yelp15 (Fig. E.2), and Yahoo Answers (Fig. E.3). Each figure shows (D) Data sample; (M) Mean (latent) reconstruction (*i.e.*, $\boldsymbol{z}_i = \mathbb{E}\left[q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)\right]$); (R) Reconstruction (*i.e.*, $\boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i)$); (P) Perturbed (latent) reconstruction (*i.e.*, $\boldsymbol{z}_i \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_i; \mu_i, 10\sigma_i)$); (AE) Reconstruction of AE. We compare the best performing sMIM model to an AE with the same architecture, and to sMIM (1024) † (*i.e.*, the model trained on the Everything dataset).

Interestingly, AEs tend to perform worse for longer sentences, when compared to sMIM. We attribute this to the higher latent entropy, which leads to non-semantic errors (*i.e.*, nearby latent codes are less similar compared to MIM). Another interesting point is how the reconstruction (R), is better in many cases than the reconstruction given the mean latent code from the encoder (M) (*i.e.*, which have the highest probability density). We attribute that to the fact that most probability mass in a high dimensional Gaussian in $d >> 1$ dimensional space and σ standard deviation is concentrated in around a sphere of radius $r \approx \sigma\sqrt{d}$. As a result the probability mass around the mean is low, and sampling from the mean is less likely to represent the input sentence $\boldsymbol{x}_i$. This also explains how perturbations of up to 10 standard deviations might result in good reconstructions. Finally, we point how sMIM (1024) †, trained on Everything, does a better job handling longer sentences.

	sMIM (1024)	sMIM (1024) †
(D)	**(3 stars)** <SOS> decent price . fast . ok staff ... but it is fast food so i ca n't rate any higher than 3 .	
(M)	decent italians . fast . price ok ... but it is higher than any other fast food i ca n't rate so higher rate jusqu . <EOS>	decent oxtail . ok . fast price ... but staff it is so fast i ca n't rate any food 3 . <EOS>
(R)	decent price . superior . decent staff ... but ok fast food is n't so it i ' d rate higher any higher quality than 3 . <EOS>	decent price . fast staff . fast ok ... but it is so fast food i rate 3 higher than any . <EOS>
(P)	decent price . ok . fast food ... but it is ok . so i ca n't rate any higher rate as fast food is marginal . <EOS>	decent price . fast . wu ... fast food ! but it staff so ok i ca n't rate 3 stars . . <EOS>
(AE)	decent price . fast staff . ok ... but it is fast food so i ca n't rate any rate than 3 . <EOS>	
(D)	**(4 stars)** <SOS> excellent wings . great service . 100 % smoked wings . great flavor . big meaty . i will definitely be back . okra is great too .	
(M)	excellent wings . great service . 100 % wings . big meaty wings . great flavor . i definitely will be back . lake is great too . <EOS>	excellent service . great wings . 100 % superior . great flavor . great fries . definitely will be back . i had too big fat . <EOS>
(R)	excellent wings . great service . 100 % wings . wings flavor . definitely great . 100 % . i will be back . <EOS>	excellent service . great flavor . 100 % wings . excellent . great big guts . definitely will be back from . i had great wings . <EOS>
(P)	excellent wings . great service . wings flavours wings . 100 % big . mmmmm overwhelmed . i ' m definitely hooked . bye disgusted is great but will be back . i definitely go . <EOS>	great burger . excellent service . 100 % fat bowls . great carnitas . great flavor . i will definitely be back . i avoid too late . <EOS>
(AE)	excellent excellent . great service . 100 % wings . 100 % big burritos . 100 % . i will definitely be back . great too too is ultra <EOS>	
(D)	**(5 stars)** <SOS> delicious ! the sandwiches are really good and the meat is top quality . it ' s also nice grabbing an exotic item from the shelf for dessert .	
(M)	delicious ! the meat really are good and the quality is nice . it ' s also tempting top notch lovers from the roasters an item top . <EOS>	delicious ! the sandwiches are really good and the quality is top notch . it ' s an exotic item popping also generates from the top spices . <EOS>
(R)	delicious ! the sandwiches are really good and the meat is quality . it ' s also nice dessert for shipping from the top floor an unhygienic machine . <EOS>	delicious ! the sandwiches are really good and the quality is top notch . it ' s also charging an item assortment from the grocery store for dessert . <EOS>
(P)	delicious sandwiches ! the servers are really good and the quality is top notch . it ' s also an item for meat quality memories . <EOS>	who ! the meat are really good and the quality is top notch ' s . it also seems top notch item has yet and an unexpected range for the pistachio . i do cross like john tomatoes from my experience . <EOS>
(AE)	delicious ! the sandwiches are really good and the quality is top notch . it ' s also caught meat also fixing an item from the top for nice hash . <EOS>	

Table E.2: Reconstruction results for models trained on **Yelp15**. We denote: (D) Data sample; (M) Mean (latent) reconstruction; (R) Reconstruction; (P) Perturbed (latent) reconstruction; (AE) Reconstruction of AE.

	sMIM (1024)	sMIM (1024) †
(D)	**(Sports)** <SOS> are you regular or goofy ? regularly goofy	
(M)	are you regular or regular ? regular <EOS>	are you regular or regularly ? regular johnny <EOS>
(R)	are you regular regular or nintendo ? regular ice-cream <EOS>	are you regular or regularly ? regularly gethsemane <EOS>
(P)	are you or regular worms regular ? regular goldfish by benjamin <EOS>	are you regular or early regularly regularly regularly <EOS>
(AE)	are you sex or two frustrated <EOS>	
(D)	**(Health)** <SOS> how do you start to like yourself ? i was taught by my parents .	
(M)	how do you start to like yourself ? i would like to meet my parents by . <EOS>	how do you start to like yourself ? i was taught by my parents . <EOS>
(R)	how do you start to yourself like ? i was taught my parents by parents . <EOS>	how do you start to like yourself ? i was taught by my parents . <EOS>
(P)	how do you start to like yourself ? i am 27 by my self . <EOS>	how do you start to like yourself ? start by i was taught my foot . <EOS>
(AE)	how do you like to after by christmas day ? i like to aid my boss by my brother and state ! <EOS>	
(D)	**(Business & Finance)** <SOS> how can i find someone in spain ? i'm in spain today , what do you want ?	
(M)	how can i find someone in spain ? i'm in harlem limo , now what do you want ? <EOS>	how can i find someone in spain ? spain in spain ? i'm talking , what did you want ? <EOS>
(R)	where can i find someone in spain ? in spain today , what do you want ? <EOS>	how can i find someone in spain ? spain in spain today , what do you want ? <EOS>
(P)	how can i find someone in stone ? in nassau i'm sure civilian , what ? you want today ! <EOS>	how can i find someone in spain ? i'm in spain today ? what maytag , do you think ? <EOS>
(AE)	how can i find someone in africa investment , ca ? working 6.0 in future with susan toughie <EOS>	

Table E.3: Reconstruction results for models trained on **Yahoo Answers**. We denote: (D) Data sample; (M) Mean (latent) reconstruction; (R) Reconstruction; (P) Perturbed (latent) reconstruction; (AE) Reconstruction of AE.

E.2 Interpolation

Here we provide interpolation results for PTB (Fig. E.4), Yelp15 (Fig. E.5), and Yahoo Answers (Fig. E.6). We compare the best performing sMIM model to sMIM (1024) †. Interestingly, both models appear to have learned a dense latent space, with sMIM (1024) † roughly staying within the domain of each dataset. This is surprising since the latent space of sMIM (1024) † jointly represents all datasets.

sMIM (512)	sMIM (1024) †
<SOS> thanks to modern medicine more couples are growing old together	
• to growing small businesses are growing more rapidly growing <EOS>	• thanks to modern medicine more modern couples are growing together than <EOS>
• growing to more areas are growing preventing black trends <EOS>	• thanks to modern cancer more are growing peaceful couples form <EOS>
• growing to the growing industry are growing more rapidly growing than <EOS>	• thanks to medicine rosen modern more are growing together governing <EOS>
• growing to the exact industry has been growing more sophisticated six months <EOS>	• thanks to moolah the modern premises are more sensitive together <EOS>
• politics the growing issue are not to mention closely although other prospective products <EOS>	• programm thanks to the cutbacks schedules is not an church system <EOS>
• the system is growing enough to make not radical an article <EOS>	• humana remains the loyalty to instituting dynamic is an orthodox montage <EOS>
• the system is reducing compliance not to consider an article <EOS>	• the strategies is not paying the non-food system an individual member <EOS>
• the system is the problem system not an effective <EOS>	• the system is not the individual problem member an can <EOS>
• the system is the system not knowing an individual <EOS>	• the system is not the individual problem an individual member <EOS>
• the system is the system not an encouraging problem <EOS>	• the system is not the individual problem an individual member <EOS>
<SOS> the system is the problem not an individual member	
• the system is the system not an investment fund <EOS>	• the system is the ringers not an individual member <EOS>
• the system is the problem not an office <EOS>	• the system is not the problem an individual member <EOS>
• the system is not the problem for an individual <EOS>	• the problem is not the indies system an individual <EOS>
• the system is not clear the veto <EOS>	• the merksamer is not the problem system an individual <EOS>
• the system is not encouraging to the securities <EOS>	• mr . the herald is not an individual problem <EOS>
• xtra the system is not even critical <EOS>	• qintex producers is the president's to comment <EOS>
• sony denies the declines to secure <EOS>	• sony preferences itself is the bidding to comment <EOS>
• everyone brought the stock to comment <EOS>	• sony sony itself is to comment <EOS>
• sony which declines to comment <EOS>	• sony sony itself to comment <EOS>
• kellogg declines to induce itself <EOS>	• sony declines itself to sony <EOS>
<SOS> sony itself declines to comment	

Table E.4: Interpolation results between latent codes of input sentences (with gray) from **PTB**.

sMIM (1024)	sMIM (1024) †
(3 star) <SOS> as bbq in phoenix goes - this is one of the better ones . get there early - they fill up fast !	
• as in china phoenix - this is one of the better ones fast get . fill there early - they fill up early ! <EOS>	• as in phoenix goes this is - better than one of the newest ones . get there early - they fill up fast ! <EOS>
• as far in san jose - this is one of the better ones . fast get up early ! there they fill up fast for u ! <EOS>	• as shore goes in phoenix - this is one of the better bbq . fast ! they get up there early - men dinner . <EOS>
• oxtail yo buffet in pittsburgh as the owners goes - better . this is not one of those fast food places . fill up there get the hot ! <EOS>	• veal as rocks as this goes in the phoenix area . - one of food is not better quick enough they get . 2 enchiladas up ! <EOS>
• ah circle k ! not as bad in the food . thankfully - this one is one of the best bbq joints here ! service was fast friendly . <EOS>	• kohrs as molasses as comparing goes in the food . not sure is one of this better ones - the only ones for fat . thumbs squeeze there ! <EOS>
• bin spaetzle food not the best . wicked spoon ! service is brutal only fast for the hot mexican in lv . everything else on this planet as can you get . <EOS>	• = frozen food ! not the best . only frozen hot as for you shall pick the ice cream - . loved everything else on wednesday ! <EOS>
• food not the best service . knocking only 99 cents ! for the hot buffet everything . beef & broccoli on the vip polo you can pass . <EOS>	• = food not . the best frozen service ! only five stars for the water suppose . hot things you can smell on budget . <EOS>
• food not the best . service = horrible ! only plopped for the paella everything & rum . you can find everything on the strip . <EOS>	• food = not the best . frozen service ! only $ 21 for the frozen hot chocolate . everything else can you tell on romance . <EOS>
(2 star) <SOS> food = not the best . service = horrible ! only known for the frozen hot chocolate . everything else you can pass on .	
• food not the best . fuck service only ! ! horrible cannolis for the fajitas unusual known . everything you can pass on graduate . <EOS>	• food = not the best . frozen hot service ! only website for the frozen hot chocolate . you can grab everything else on . <EOS>
• food not suck . the best service ever ! just horrible everything for the frozen hot chocolate . you can probably survive on everything else . <EOS>	• food = not the best . frozen service ! only for five stars during the san francisco frozen chicken . everything else on could not give thumbs . <EOS>
• blech food ! not the best dish anywhere else . service = <unk> for the frozen hot chocolate and dessert bartenders ! everything you can only expect better at this shuffle . <EOS>	• gelato food ! not sure the best . frozen seared only wish you can mix for the frozen hot chocolate frozen . service on and everything else explains . <EOS>
• 32 words ! not amazing food . the best <unk> music and service they had can earned a better meal at xs . everything else on bill for me . <EOS>	• hilariously = ! food is not the best meal . hibachi cover service and they only wished a frozen yogurt for hot girl . better luck at <unk> and on the latter experience . <EOS>
• husbands cher ! wish they had <unk> dessert at the bellagio and not a great lunch selection . food better tasting wise but sadly serves and dessert selection . <EOS>	• wish ! methinks buffet is ingrediants at the <unk> food and a better tasting . they woulda frozen lunch but not memorable and satisfying tasting better ambiance . <EOS>
• yummy ! wish they had <unk> at lunch and a dessert selection but a better value and great value than beef suggestion company . <EOS>	• wish ! wish they had <unk> at 10am and a dessert selection but better food a better and better tasting selection . great value ! <EOS>
• yummy ! wish they had <unk> dessert at lunch and a selection but a tiramisu better value and freshness value food taste better than ihop . <EOS>	• wish ! wish they had lunch at <unk> and a dessert fountain but better than a selection and great tasting food servings better tasting . <EOS>
(4 star) <SOS> yummy ! wish they had <unk> at lunch and a better dessert selection but a great value and better tasting food than wicked spoon .	

Table E.5: Interpolation results between latent codes of input sentences (with gray) from **Yelp15**.

sMIM (1024)	sMIM (1024) †
(Business & Finance) <SOS> are u shy or outgoing ? both , actually	
• are u or wishing vidio ? both , actually <EOS>	• are u shy or k ? both , actually <EOS>
• are u or stressed caffiene ? both , actually make a smile <EOS>	• are u minded or rem ? actually , both <EOS>
• witch are u or how lucky ? both <EOS>	• are u transparent or shy ? it'd actually , add-on <EOS>
• are u kidding or spraying ? both <EOS>	• are u untouchable cubed or programe ? both , actually like <EOS>
• how does wile or are you ? to both use , instead like it . <EOS>	• wha do u are roselle or marketed ? you start , by both my inbox <EOS>
• how do u choose to start or ? like i cant think , are actually better by my work . <EOS>	• how do u simplify phases towards you ? are proving , like no smiles . <EOS>
• how do you start to alienate yourself ? i are like or drone , my actually feels . <EOS>	• how do you burp confidence ? to start i was like , shareaza the new by hindering . <EOS>
• how do you start to yourself or like ? i like my math side . <EOS>	• how do you start to race ? i like kazaa when my was cheated . <EOS>
• how do you start to like yourself ? i think my parents is by focusing . <EOS>	• how do you start to start like ? i was taught by my parents . <EOS>
• how do you start to yourself like ? i was taught by my parents . <EOS>	• how do you start to like yourself ? i was taught by my parents . <EOS>
(Health) <SOS> how do you start to like yourself ? i was taught by my parents .	
• how do you start to yourself by allowing ? i like my parents yr . <EOS>	• how do you start to like yourself ? i was taught by new england . <EOS>
• how do you start to yourself like i ? my parents was by mario practitioner . <EOS>	• how do you start to like yourself ? i was taught by my parents . <EOS>
• how do you start to cite yourself ? i like by my consequences in 1981 . <EOS>	• how do i start you to beethoven ? like israel was my grandmother by fielders . <EOS>
• how do i start girls like to ? you can find yourself in my states , by today . <EOS>	• how do you start to find ? i like aggieland in my testicles was listening . <EOS>
• how do you start yourself drunk ? i can find in something like to my country , what by jane . <EOS>	• how can i do compuserve attain ? start to comment in spain you like , was my real pics . <EOS>
• how can i start those neeed in america ? do you like to rephrase an invention , what i'm spinning ? <EOS>	• how can i find blueprints do you ? i'm in spain like queens to chelsea , arrange . <EOS>
• how can i find someone in spain ? i'm guessing today by pascal , what do you want to ? <EOS>	• how can i find uneasy profiles in spain ? i'm sure what you do , like today's ? <EOS>
• how can i find an attorney in spain ? i'm studying chicken's what , do you want to ? <EOS>	• how can i find someone in spain ? i'm in spain today , what do you want ? <EOS>
• how can i find someone in spain ? in spain i'm studying , what do you want ? <EOS>	• how can i find someone in spain ? i'm in tanks today , what do you want to ? <EOS>
• how can i find someone in spain ? i'm in italy today , what do you want ? <EOS>	• how can i find someone in spain ? i'm guessing in spain today , what do you want ? <EOS>
(Business & Finance) <SOS> how can i find someone in spain ? i'm in spain today , what do you want ?	

Table E.6: Interpolation results between latent codes of input sentences (with gray) from **Yahoo Answers**.

E.3 Sampling

sMIM (512)
• instead the stock market is still being felt to <unk> those of our empty than in a bid <EOS>
• he estimated the story will take <unk> of paper co . ' s $ n million in cash and social affairs to at the company a good share <EOS>
• long-term companies while the company ' s <unk> provisions would meet there to n or n cents a share and some of costly fund <EOS>
• time stocks the company explained him to sell <unk> properties of high-grade claims which has received a net loss in the firm <EOS>
• what i had the recent competition of <unk> replies that is n't expected to draw a very big rise in tokyo <EOS>

Table E.7: Samples from best performing model for dataset **PTB**.

sMIM (1024)
• ben monkey gabi sister near the western fest . i ' ve been looking forward to this location , and each time i ' m in the 6th bunch i want to have a great visit experience . it was all kinds of fillers , owns and dressings non-asian with jalapeños <unk> does n't hold me for much healthier . front desk is not my favorite dinner place at the gates . they are closed on mondays , - lrb - it could affect a couple minutes more rocks - rrb - and then we said the bar was the real bold . i ' d rather go to firefly some bubble in greece . if you had a neighbourhood addiction <unk> c , take this look as most amazing . <EOS>
• hello tanya stephen covering qualité . ugh haha , i was curious to consume that the white asian restaurants believes filled a mob and turkey melt departments for $ 9.99 . the <unk> of these were not intrusive , it was accepted in there . . i ' m sure this is n't one of my favorite places to go at night with here ! particularly speaking the italian cleaning tables . we also ordered some pina colada , which tasted exactly like they came out of a box and per endearing thick . pretty good food overall , and the pigeons self nightly . i ' d call it again just on halloween for a dependable lunch . but the statue sucks ? so if you have bouchon to inquire was good place . <EOS>
• prada based pata based solely often inside . this place is unappealing horrific for the 50th and fries , i ' ve caught to have a ton of good reviews <unk> in buckeye , barnes knew . not bc that i was wrong with my team being kicked the whole thing at eggroll , it ' s like pulling out of the landmark . no luck on ketchup top crunch , if you are craving something simple and <unk> . we also tried the wild mushroom - lrb - it ' s burn , did n't go in disheveled - rrb - as a matter destination from flavor . the food was just ok and nothing to write home about . friend peeps i only had one beer , but this place does not deserve the same increase . <EOS>

Table E.8: Samples from best performing model for dataset **Yelp15**.

Here we show samples from the best performing models learned from a single dataset for PTB (Fig. E.7), Yelp15 (Fig. E.8), and Yahoo Answers (Fig. E.9). We sample from a zero-mean Gaussian distribution over the latent space, with an isotropic covariance with a standard deviation of 0.1 (since we cannot directly sample from the implicit marginal over the latent). Interestingly, this simple heuristic provides good samples. We attribution this to the anchor, which defines scale and position for the implicit marginal over the latent to roughly match.

sMIM (1024)
• how does transformers send grow ina under pubs ? i found the suspension resides official game is exciting to withstand and what can a person do in that case ? brees fights , if it does 150 . the dre is tied ordered outlook <unk> 2005 . today had a migrane with limitation tops , because of his vr repeats , you are referring to review at the university of 1994 and have visited fortune . judy for websites <unk> website is beware confused . <EOS>
• how do i download jesus gyno to woman whom ? being irvine in line is what you did a lot of oceanic denny in the middle east and spanish wallet or <unk> entity . plus , i'm aware of that , particularly do you have any insight insight ... if you are a hoe who's right click on it , and you can ' t get some skills god . the other government also happened to be <unk> with most varied life-forms is located at this point . foreigners your covers , and maybe even my friends . <EOS>
• what's mastering marathons fluently is einstein among the waivers ? ok i feel that what happened to tom during the holidays moniter of 1-2 awol whn reservoir <unk> . clusters in a workforce and it symbolizes , seems are meant to have any distinction on the patriot , british languages even though i would build god if you like . just bringing your old door as a distorted spree ? hmmmm , because you're not anti-bacterial pure dino and <unk> this can be deduced . <EOS>

Table E.9: Samples from best performing model for dataset **Yahoo Answers**.

E.4 Question Answering

Here we provide more examples of answers generated from a model trained on Yahoo Answers (*i.e.*, sMIM (1024) in Fig. E.10). In particular, the model was trained from data in which 20% of the encoder input tokens were replaced with the <unk> token. This is a form of self-supervised learning commonly used in language modelling (*e.g.*, Bowman et al. (2015)). This encourages the model to replace <unk> with other tokens. We have found this procedure to significantly improve the quality of the generated answers. We provide three generated answers for each question (Q), taken from Yahoo Answers. Short/medium/long answers (A) are generated by concatenating 5/10/15 <unk> tokens. The number of <unk> encodes the length of the expected answer. We note that, in many cases, only one answer will be a good match to the question, suggesting the model has preferences towards answers with a question specific length.

sMIM (1024)	
(Q)	<SOS> where can i find the most affordable roses for valentine's day this late in the game ?
(A) short	• try the mall , <EOS>
(A) medium	• the game is to play the game , set your mind <EOS>
(A) long	• i think the best answer is to ask her out , especially since the late 1990s game <EOS>
(Q)	<SOS> how do you start to like yourself ?
(A) short	• start by yourself , take the steps <EOS>
(A) medium	• start by yourself . you have to start asking yourself why this is happening <EOS>
(A) long	• start by yourself . you have to start acting yourself in the middle of your eye . <EOS>
(Q)	<SOS> how can i find someone in spain ?
(A) short	• try classmates.com or classmates.com to see <EOS>
(A) medium	• someone who is a good one to talk to your question <EOS>
(A) long	• someone who is a good person to find a nice girl , and take care of yourself <EOS>
(Q)	<SOS> with a weapon to your person , who in you family do you save with an answer ?
(A) short	• you should save yourself in person <EOS>
(A) medium	• you can save the money in person , or do you want to save money <EOS>
(A) long	• you can save your money in person , and answer the question yourself . do you have a family ? <EOS>
(Q)	<SOS> what are the best sites to shop for personal computers online ?
(A) short	• try amazon.com online . <EOS>
(A) medium	• i think it is best to shop online , or take a look at the personal <EOS>
(A) long	• yahoo is best online . i would suggest checking out the personal website for personal info <EOS>
(Q)	<SOS> who is going to win the super bowl this year ?
(A) short	• the steelers is a pretty good <EOS>
(A) medium	• the pittsburgh steelers is a good one , but i don't know <EOS>
(A) long	• this is the best team to win the super bowl , and i think you mean good luck <EOS>
(Q)	<SOS> what is the closest capital city to paris ?
(A) short	• paris hilton city , ca <EOS>
(A) medium	• in the south city , ca , i think you mean the city hall <EOS>
(A) long	• city is the capital city in south america . i think it is a beautiful city <EOS>
(Q)	<SOS> do height increasing devices shown on television works ?
(A) short	• perhaps shown is shown to see <EOS>
(A) medium	• yes , the best way to increase it is to work <EOS>
(A) long	• yes , shown shown . the biggest problem with television is to take a walk away from the edge <EOS>
(Q)	<SOS> where can i find advice on hip hop beat making ?
(A) short	• try this website : http://messenger.yahoo.com/ <EOS>
(A) medium	• try this website : http://messenger.yahoo.com/ to find the best answer . <EOS>
(A) long	• try making a video club , such as the lion and hip hop . i wish you to be good <EOS>
(Q)	<SOS> how do u clear the history in the search field ?
(A) short	• clear search history , it will be clear <EOS>
(A) medium	• clear history , it appears that the question period has expired . if you have received an answer that meets your needs , please choose a ' best answer . ' regards <EOS>
(A) long	• clear history in the search field , u can find it from your website : * clear history <EOS>
(Q)	<SOS> where can i get chapter summaries in text books ?
(A) short	• try this site , you'll find it interesting <EOS>
(A) medium	• text books ? try this site , and get a book to read <EOS>
(A) long	• in books , it is a text book , and the text books are written in the same text . <EOS>

Table E.10: Question and sampled answers from model sMIM (1024) (*i.e.*, trained on Yahoo Answers dataset). We provide short/medium/long sampled answers (A) for each question (Q).

Appendix F

TzK: Entropy and Mutual Information

This section provides some context and a derivation for Eq. (8.12).

As discussed in Sec. 8.3.2, probability density normalizing flows allow for efficient learning of arbitrary distributions by learning a mapping from independent components to a joint target distribution (Dinh et al., 2014). It follows that for a sufficiently expressive $\boldsymbol{t}$-flow the TzK model factorization assumed in (8.3) - (8.7) does not pose a fundamental limitation when learning joint, conditional distributions. In other words, it is likely that there exist distribution flows with which $\mathcal{P}$ can be factored according to the encoder and decoder factorizations in TzK .

To that end, we can assume that with a sufficiently expressive model one can assume that the dual encoder/decoder model can fit to the true data distribution reasonably well. In the ideal case, where the encoder and decoder are consistent and equal to the underlying data distribution, *i.e.* $\mathcal{P} = q = p$, we obtain the following result, which relates the entropy of the data distribution to the mutual information between the data distribution and the latent

space representation:

$$
\begin{aligned}
-H\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right) &= \mathbb{E}_{\bar{\boldsymbol{k}}, \boldsymbol{t} \sim \mathcal{P}}\left[\log \mathcal{P}\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right)\right] \\
&= \mathbb{E}_{\bar{\boldsymbol{k}}, \boldsymbol{t} \sim \mathcal{P}}\left[\log\left(\frac{1}{2} q\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right) + \frac{1}{2} p\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right)\right)\right] \\
&= \mathbb{E}_{\bar{\boldsymbol{k}}, \boldsymbol{t} \sim \mathcal{P}}\left[\frac{1}{2} \log q\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right) + \frac{1}{2} \log p\left(\bar{\boldsymbol{k}}, \boldsymbol{t}\right)\right] \\
&= \mathbb{E}_{\bar{\boldsymbol{k}}, \boldsymbol{t} \sim \mathcal{P}}\left[\begin{array}{l} \log p\left(\boldsymbol{t}\right) + \frac{1}{2} \sum_i \left(\begin{array}{l} \log p\left(\boldsymbol{k}^i | \boldsymbol{t}\right) \\ + \log p\left(\boldsymbol{k}^i\right) \\ + \log p\left(\boldsymbol{z}\left(\boldsymbol{t}\right) | \boldsymbol{k}^i\right) \\ - \log p\left(\boldsymbol{z}\left(\boldsymbol{t}\right)\right) \end{array}\right) \end{array}\right] \\
&= -H\left(\boldsymbol{t}\right) + \frac{1}{2} \sum_i \left(\begin{array}{l} H\left(\boldsymbol{z}\right) \\ -H\left(\boldsymbol{z} | \boldsymbol{k}^i\right) \\ -H\left(\boldsymbol{k}^i\right) \\ -H\left(\boldsymbol{k}^i | \boldsymbol{t}\right) \end{array}\right) \\
&= -H\left(\boldsymbol{t}\right) + \frac{1}{2} \sum_i \left(\begin{array}{l} I\left(\boldsymbol{z}; \boldsymbol{k}^i\right) - H\left(\boldsymbol{k}^i\right) \\ -H\left(\boldsymbol{k}^i | \boldsymbol{t}\right) \end{array}\right) \\
&= -H\left(\boldsymbol{t}\right) - \sum_i H\left(\boldsymbol{k}^i\right) + \frac{1}{2} \sum_i (I(\boldsymbol{k}^i; \boldsymbol{t}) + I(\boldsymbol{z}; \boldsymbol{k}^i))
\end{aligned}
\tag{F.1}
$$

Eq. (F.1) illustrates an interesting connection between ML and MI, assuming TzK to be the true underlying model. One can interpret ML learning of $\mathcal{M}_{\boldsymbol{\theta}}$ as a lower bound for the sum of the negative entropy of observations $\boldsymbol{t}$, the negative entropy of latent codes $\bar{\boldsymbol{k}}$, and the MI between the observations $\boldsymbol{t}$ and the latent codes $\boldsymbol{k}^i$, and between the latent state $\boldsymbol{z}$ and the latent codes $\bar{\boldsymbol{k}}$. This formulation arises naturally from the TzK representation of the data distribution, as opposed to several existing models that use MI as a regularizer (Belghazi et al., 2018; Chen et al., 2016a; Dupont, 2018; Klys et al., 2018).

An important property of the TzK formulation is the lack of variational approximations where an auxiliary distribution $q(z|x)$ is used to approximate $p(z|x)$. As a consequence, it is hoped that a more expressive $\mathcal{M}_{\boldsymbol{\theta}}$ will be better able to approximate $\mathcal{P}$, leading to a

tighter lower bound; since, compared to variational inference (VI), a more expressive q does not necessarily guarantees a tighter lower bound as it is restricted to tractable families. In addition, since $\mathcal{D}_{\mathrm{KL}}(q \,\|\, p)$ is unknown, VI does not offer a method to measure that gap.

Appendix G

TzK: Experimentation and Implementation Details

G.1 Architecture Details

The components of a TzK model, *i.e.*, the factors in Eqs. (8.3) - (8.7), have been implemented in terms of parameterized deep networks. In somewhat more detail, the prior over $\boldsymbol{t}$ was implemented with:

- $f_{\boldsymbol{t}}(\boldsymbol{z}) : \boldsymbol{z} \in \mathbb{R}^T \rightarrow \boldsymbol{t} \in \mathbb{R}^T$ is our Glow-based implementation (Kingma and Dhariwal, 2018). Flow details are included with each experiment.
- $q_{\boldsymbol{\theta}}(\boldsymbol{t}) : \emptyset \rightarrow \boldsymbol{t} \in \mathbb{R}^T$ is parameterized in terms of a probability normalizing flow $f_{\boldsymbol{t}}(\boldsymbol{z})$ from a multivariate standard normal distribution $p(\boldsymbol{z})$.

All density probability flow $f_{\boldsymbol{t}}$ had 3 layers (multi-scale) and steps defined in each experiment, with 512 channels for regressors in affine coupling transforms (Kingma and Dhariwal, 2018).

The priors over latent codes $\boldsymbol{c}^i$ were implemented with:

- $p_{\boldsymbol{\theta}}(\boldsymbol{c}^i)$, for $\boldsymbol{c}^i \in \mathbb{R}^C$, is parameterized in terms of a probability normalizing flow $f_{\boldsymbol{c}^i}(\cdot) : \mathbb{R}^C \rightarrow \mathbb{R}^C$ from a multivariate standard normal distribution.

For $\boldsymbol{c}^i \in \mathbb{R}^C$, we use a Glow architecture with 1 layer, 4 steps, and $10C$ channels, where $C = 10$, unless specified otherwise in an experiment. $\boldsymbol{c}^i$ was shaped to have all dimensions in a single channel $C \times 1 \times 1$.

The discriminators associated with different knowledge types $p(\mathrm{e}^i = 1|\cdot)$, conditioned on observation $\boldsymbol{t}$ or a latent code $\boldsymbol{c}^i$, were implemented with:

- $q_{\boldsymbol{\theta}}(\mathrm{e}^i|\boldsymbol{t}) : \boldsymbol{t} \in \mathbb{R}^T \rightarrow [0, 1]$
- $p_{\boldsymbol{\theta}}(\mathrm{e}^i|\boldsymbol{c}^i) : \boldsymbol{c}^i \in \mathbb{R}^C \rightarrow [0, 1]$

All discriminators from $\boldsymbol{t}$ and $\boldsymbol{c}^i$ had 3 layers of 3×3 convolution with $10C$ channels (unless specified otherwise in experiment details), followed by linear mapping to the target dimensionality of 1, and a sigmoid mapping to normalize the output to be $[0, 1]$.

The conditional priors over $\boldsymbol{t}$ and $\boldsymbol{c}^i$ are modeled with regressors from the corresponding $\boldsymbol{c}^i$ and $\boldsymbol{t}$ to the distribution parameters (*e.g.*, mean and variance for Gaussian), similar to VAE. More explicitly, we implemented the priors with regressors to the mean and diagonal covariance matrix of a Gaussian, as described below:

- $q_{\boldsymbol{\theta}}(\boldsymbol{c}^i|\mathrm{e}^i, \boldsymbol{t}) : \boldsymbol{t} \in \mathbb{R}^T \rightarrow \{\mu, \sigma\} \in \mathbb{R}^C \times \mathbb{R}^C$ is composed of a regressor to the mean μ and diagonal covariance σ of a Gaussian base distribution, and $f_{\boldsymbol{c}^i}(\cdot)$, an invertible function that serves as a probability flow. The two components comprise a single parametric representation of a generic probability distribution. We condition the density on e^i by learning two separate sets of weights, i.e., for $\mathrm{e}^i \in \{0, 1\}$. The flow uses the same Glow architecture as $f_{\boldsymbol{t}}$, for which the details are given below. All $f_{\boldsymbol{c}^i}$ had 4 flow steps, with dimensionality of $C = 10$, unless specified otherwise.
- $p_{\boldsymbol{\theta}}(\boldsymbol{t}|\mathrm{e}^i, \boldsymbol{c}^i) : \boldsymbol{c}^i \in \mathbb{R}^C \rightarrow \{\mu, \sigma\} \in \mathbb{R}^T \times \mathbb{R}^T$ is composed of a regressor to the mean μ and diagonal covariance σ of a Gaussian base distribution, and $f_{\boldsymbol{t}}(\boldsymbol{z})$. We condition the probability density on e^i by learning two separated set of weights for $\mathrm{e}^i \in \{0, 1\}$.

All Gaussian regressors $\boldsymbol{c}^i \rightarrow \boldsymbol{z}$ were implemented with a linear mapping $\mathbb{R}^C \rightarrow \mathbb{R}^{80 \times 4 \times 4}$, 3 layers of 3×3 convolution layers with 512 channels, and final layer with 192 channels,

resulting in $\boldsymbol{z} \in \mathbb{R}^{192\times4\times4}$. All Gaussian regressors $\boldsymbol{t} \to \boldsymbol{c}^i$ were implemented with 3×3 convolutional layer with 80 channel, followed by 3 layers of alternating squeeze (Kingma and Dhariwal, 2018) and 3×3 convolutional layers with 80 channels, followed by linear layer to C. All regressors had ActNorm to initialize inputs to be mean-zero with unit variance, and the weights of the last layer were initialized to 0, as in (Kingma and Dhariwal, 2018).

G.2 Model Sampling and Evaluation

Here we explain in detail how we generate samples and evaluate the model during training and experimentation. To that end, we consider the evaluation of the negative log likelihood of a data sample under the model, and the process for drawing a random sample from the model. As described above, a trained TzK model results in empirically consistent encoder and decoder. This allows us to use both the encoder and the decoder to generate samples from $\mathcal{M}_{\boldsymbol{\theta}}$. It is important to note that one cannot assume so during training.

G.2.1 Sampling during training

When all samples are given (supervised training) the optimization problem is well defined. However, when a sample $\boldsymbol{c}^i$ is missing during training (but $\boldsymbol{t}, \mathrm{e}^i$ are given), the problem is not fully defined. That is, we need to define what constitutes fair samples of $\bar{\boldsymbol{c}}$. To do so we define a, augmented sample distribution

$$\mathcal{M}_{\mathcal{S}} = \frac{1}{2}\left[q_{\boldsymbol{\theta}}(\bar{\boldsymbol{c}}|\mathrm{e}, \boldsymbol{t})\mathcal{P}(\mathrm{e}, \boldsymbol{t}) + p_{\boldsymbol{\theta}}(\boldsymbol{t}|\bar{\boldsymbol{c}}, \mathrm{e})\mathcal{P}(\mathrm{e}, \bar{\boldsymbol{c}})\right] \tag{G.1}$$

where $\mathcal{P}$ is augmented with $q_{\boldsymbol{\theta}}(\bar{\boldsymbol{c}}|\mathrm{e}, \boldsymbol{t})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{t}|\mathrm{e}, \bar{\boldsymbol{c}})$.

Sampling from $\mathcal{M}_{\mathcal{S}}$ entails randomly choosing between the sample encoding and decoding distributions with equal chances. Sampling from $\mathcal{P}, q$ involves sampling $\bar{\boldsymbol{c}} \sim q(\bar{\boldsymbol{c}}|\mathrm{e}, \boldsymbol{t})$ where $\boldsymbol{t}, \mathrm{e} \sim \mathcal{P}(\boldsymbol{t}, \mathrm{e})$, which provides us with fair samples from $\mathcal{M}_{\mathcal{S}}$ to be used in Eq. (8.10). Sampling from the decoding distribution $p_{\boldsymbol{\theta}}(\boldsymbol{t}|\bar{\boldsymbol{c}}, \mathrm{e})\mathcal{P}(\mathrm{e}, \bar{\boldsymbol{c}})$ amounts to unsupervised learning. Unfortunately, the decoder provides no efficient sampling method for multiple knowledge

types (*i.e.*, there is no explicit $\boldsymbol{t} \sim p_{\boldsymbol{\theta}}(\boldsymbol{t}|\bar{\boldsymbol{k}})$ only explicit $\boldsymbol{t} \sim p_{\boldsymbol{\theta}}(\boldsymbol{t}|\boldsymbol{k}^i)$). In order to be able to use samples from each individual decoder we exploit the structure of $\mathcal{M}_{\boldsymbol{\theta}}$ to rearrange the lower bound in Eq. (8.10) as follow

$$\begin{aligned} -\frac{1}{2}\left[CE\left(\mathcal{M}_{\mathcal{S}}\,,\,q_{\boldsymbol{\theta}}\right)+CE\left(\mathcal{M}_{\mathcal{S}}\,,\,p_{\boldsymbol{\theta}}\right)\right] &= -\sum_i CE\left(\mathcal{M}_{\mathcal{S}}(\boldsymbol{t},\boldsymbol{k}^i)\,,\,\frac{p^{enc}_{\boldsymbol{\phi}^i}(\boldsymbol{t},\boldsymbol{k}^i)+p^{dec}_{\boldsymbol{\psi}^i}(\boldsymbol{t},\boldsymbol{k}^i)}{2}\right) \\ &\quad +(C-1)H\left(\mathcal{M}_{\mathcal{S}}(\boldsymbol{t})\right)q_{\boldsymbol{\theta}}(\boldsymbol{t}) \\ &= \sum_i \mathbb{E}_{\boldsymbol{t},\boldsymbol{k}^i\sim\mathcal{M}_{\mathcal{S}}}\left[\begin{array}{l}\log\frac{1}{2}p^{enc}_{\boldsymbol{\phi}^i}(\boldsymbol{t},\boldsymbol{k}^i)+\\ +\log\frac{1}{2}p^{dec}_{\boldsymbol{\psi}^i}(\boldsymbol{t},\boldsymbol{k}^i)\\ -\log\frac{C-1}{C}q_{\boldsymbol{\theta}}(\boldsymbol{t})\end{array}\right] \end{aligned} \tag{G.2}$$

where C is the number of knowledge types. The above expression allows the use of independent knowledge-specific samples to approximate the various entropy terms in the lower bound. More explicitly, we can draw joint samples from the decoding distribution, independently per knowledge type, and use it to train the corresponding components. The proposed procedure draws fair samples from $\mathcal{M}_{\mathcal{S}}$ during training.

It is interesting to note some similarities to GAN. Like GAN, samples of a random variable are drawn from two distinct distributions. Unlike GAN, which is adversarial, TzK is contrastive. In other words TzK has a single objective for all samples whereas GAN has two contradicting objectives. The difference is reflected in the stability of training a TzK model.

We also considered an alternative choice of $\boldsymbol{t},\bar{\boldsymbol{k}} \sim \mathcal{M}_{\mathcal{S}}(\bar{\boldsymbol{c}},\mathrm{e},\boldsymbol{t}) = q_{\boldsymbol{\theta}}(\bar{\boldsymbol{c}}|\mathrm{e},\boldsymbol{t})\mathcal{P}(\mathrm{e},\boldsymbol{t})$ which uses only the encoder. This, however, may lead to an unstable optimization following an ever-shrinking variance of $q_{\boldsymbol{\theta}}(\bar{\boldsymbol{c}}|\mathrm{e},\boldsymbol{t})$ which is associated with entropy minimization. One can increase the stability by limiting the variance of $q_{\boldsymbol{\theta}}(\bar{\boldsymbol{c}}|\mathrm{e},\boldsymbol{t})$ which could potentially limit the expressiveness of the model.

G.2.2 Approximated sampling from multiple knowledge

We next explain the procedure for sampling from a class conditional hierarchical model. To sample from a the digit "1" over a domain $\boldsymbol{c}^i$ we need to sample $\boldsymbol{t},\boldsymbol{c}^i \sim \mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{t},\boldsymbol{c}^i|\mathrm{e}^i=1)$ which

has no explicit form. Instead, we approximate the sample using block Gibbs sampling over $\boldsymbol{t}$ and $\boldsymbol{c}^i$. To do so we alternate between $\boldsymbol{c}^i$ or $\boldsymbol{t}$ as the block to sample from the corresponding $q(\boldsymbol{c}^i|\mathrm{e}^i = 1, \boldsymbol{t})$ and $p(\boldsymbol{t}|\mathrm{e}^i = 1, \boldsymbol{c}^i)$. Repeating the process converges quickly to a fair sample $\boldsymbol{t}, \boldsymbol{c}^i \sim \mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{t}, \boldsymbol{c}^i|\mathrm{e}^i = 1)$. A similar procedure is used to sample $\boldsymbol{t}, \bar{\boldsymbol{k}}$ over multiple knowledge types. As an alternative, one can use importance sampling to draw fair samples from a multi-knowledge class conditional distribution by re-sampling samples from the encoder.

G.2.3 Evaluation

Given a set of test samples, the NLL is defined as the average negative log likelihood of the individual samples (*i.e.*, assuming IID samples). Evaluating the NLL for $q_{\boldsymbol{\theta}}(\boldsymbol{t})$ is straightforward in terms of the flow $f_{\boldsymbol{t}}$ and the latent Gaussian prior $p(\boldsymbol{z})$.

To evaluate the NLL for a conditional distribution, given $\boldsymbol{t}$, we first draw a random sample $\boldsymbol{c}^i \sim q_{\boldsymbol{\theta}}(\boldsymbol{c}^i|\mathrm{e}^i = 1, \boldsymbol{t})$. We then use that sample to build $p_{\boldsymbol{\theta}}(\boldsymbol{t}|\mathrm{e}^i = 1, \boldsymbol{c}^i)$. with which We evaluate the log probability of $\boldsymbol{t}$.

Figure G.1: Additional samples of CIFAR10 conditional with CIFAR10 frozen flow, see Fig. 8.8a.

Bibliography

Agakov, F. and D. Barber

2003. The IM algorithm: a variational approach to information maximization. In *NIPS*, Pp. 201–208.

Alemi, A. A., B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy

2017. An information-theoretic analysis of deep latent-variable models. *CoRR*, abs/1711.00464.

Ardizzone, L., J. Kruse, C. Rother, and U. Köthe

2019. Analyzing inverse problems with invertible neural networks. In *ICLR*.

Bang, D. and H. Shim

2018. High quality bidirectional generative adversarial networks. *CoRR*, abs/1805.10717.

Belghazi, I., S. Rajeswar, A. Baratin, R. D. Hjelm, and A. Courville

2018. MINE: Mutual information neural estimation. In *ICML*.

Bengio, E., V. Thomas, J. Pineau, D. Precup, and Y. Bengio

2017. Independently controllable features. *CoRR*, abs/1703.07718.

Bengio, Y. and S. Bengio

1999. modeling high dimensional discrete data with multi-layer neural networks. In *NIPS*, Pp. 400–406.

Bengio, Y., A. Courville, and P. Vincent

2013. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828.

Bird, S.

2002. NLTK: The natural language toolkit. *ArXiv*, cs.CL/0205028.

Bornschein, J., S. Shabanian, A. Fischer, and Y. Bengio

2015. Bidirectional helmholtz machines. *CoRR*, abs/1506.03877.

Bowman, S. R., L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio

2015. Generating sentences from a continuous space. *CoRR*, abs/1511.06349.

Che, T., Y. Li, A. P. Jacob, Y. Bengio, and W. Li

2016. Mode regularized generative adversarial networks. *CoRR*, abs/1612.02136.

Chen, T. Q., Y. Rubanova, J. Bettencourt, and D. Duvenaud

2018. Neural ordinary differential equations. In *NIPS*.

Chen, X., Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel

2016a. InfoGAN: Interpretable representation learning by information maximizing tive adversarial nets. In *NIPS*.

Chen, X., D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel

2016b. Variational lossy autoencoder. *CoRR*, abs/1611.02731.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio

2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv*.

Dinh, L., D. Krueger, and Y. Bengio

2014. NICE: Non-linear independent components estimation. *arXiv:1410.8516*.

Dinh, L., J. Sohl-Dickstein, and S. Bengio

2016. Density estimation using real NVP. *CoRR*, abs/1605.08803.

Dinh, L., J. Sohl-Dickstein, and S. Bengio

2017. Density estimation using real nvp. *ICLR*.

Donahue, J., P. Krähenbühl, and T. Darrell

2016a. Adversarial feature learning. *CoRR*, abs/1605.09782.

Donahue, J., P. Krähenbühl, and T. Darrell

2016b. Adversarial feature learning. *CoRR*, abs/1605.09782.

Donahue, J. and K. Simonyan

2019. Large scale adversarial representation learning. *CoRR*, abs/1907.02544.

dos Santos, C. N., M. Tan, B. Xiang, and B. Zhou

2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Dumoulin, V., I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville

2017. Adversarially learned inference. *ICLR*.

Dupont, E.

2018. Learning disentangled joint continuous and discrete representations. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, Pp. 708–718, USA. Curran Associates Inc.

Durkan, C., A. Bekasov, I. Murray, and G. Papamakarios

2019. Neural spline flows. *NIPS*.

Frey, B. J., G. E. Hinton, and P. Dayan

1995. Does the wake-sleep algorithm produce good density estimators? In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS, Pp. 661–667, Cambridge, MA, USA. MIT Press.

Gao, W., S. Oh, and P. Viswanath

2016. Demystifying fixed k-nearest neighbor information estimators. *CoRR*, abs/1604.03006.

Germain, M., K. Gregor, I. Murray, and H. Larochelle

2015. MADE: Masked autoencoder for distribution estimation. In *ICML*.

Gomez, A. N., M. Ren, R. Urtasun, and R. B. Grosse

2017. The Reversible Residual Network: Backpropagation Without Storing Activations. In *NIPS*.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio

2014a. Generative adversarial nets. In *NIPS*, Pp. 2672–2680.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio

2014b. tive adversarial nets. In *NIPS*, Pp. 2672–2680.

Goodfellow, I. J., A. C. Courville, and Y. Bengio

2012. Large-scale feature learning with spike-and-slab sparse coding. In *ICML*.

Grathwohl, W., R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. K. Duvenaud

2018. FFJORD: free-form continuous dynamics for scalable reversible tive models. *CoRR*, abs/1810.01367.

Gu, J., J. Bradbury, C. Xiong, V. O. K. Li, and R. Socher

2017. Non-autoregressive neural machine translation. *ICLR*, abs/1711.02281.

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville

2017. Improved training of wasserstein gans. *CoRR*, abs/1704.00028.

Guu, K., T. B. Hashimoto, Y. Oren, and P. Liang

2017. Generating sentences by editing prototypes. *ACL*, 6:437–450.

He, J., D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick

2019. Lagging inference networks and posterior collapse in variational autoencoders. In *ICLR*.

Higgins, I., L. Matthey, A. Pal, C. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner

2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Hinton, G. E.

2002a. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800.

Hinton, G. E.

2002b. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hjelm, R. D., K. Cho, J. Chung, R. Salakhutdinov, V. Calhoun, and N. Jojic

2016. Iterative refinement of the approximate posterior for directed belief networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, P. 4698–4706, Red Hook, NY, USA. Curran Associates Inc.

Hjelm, R. D., A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio

2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Ho, J., X. Chen, A. Srinivas, Y. Duan, and P. Abbeel

2019. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *CoRR*, abs/1902.00275.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley

2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347.

Kim, Y., S. Wiseman, A. Miller, D. Sontag, and A. Rush

2018. Semi-amortized variational autoencoders. In *ICML*, J. Dy and A. Krause, eds., volume 80 of *Proceedings of Machine Learning Research*, Pp. 2678–2687, Stockholmsmässan, Stockholm Sweden. PMLR.

Kingma, D. P. and J. Ba

2015. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P. and P. Dhariwal

2018. Glow: tive Flow with Invertible 1x1 Convolutions. In *NIPS*.

Kingma, D. P. and J. Lei Ba

2014. ADAM: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling

2016. Improving variational inference with inverse autoregressive flow. In *NIPS*.

Kingma, D. P. and M. Welling

2013. Auto-Encoding Variational Bayes. In *ICLR*.

Klys, J., J. Snell, and R. Zemel

2018. Learning latent subspaces in variational autoencoders. In *NIPS*.

Kraskov, A., H. Stögbauer, and P. Grassberger

2004. Estimating mutual information. *Phys. Rev. E*, 69:066138.

Krizhevsky, A., V. Nair, and G. Hinton

2009. CIFAR10. Technical report, University of Toronto.

Kruengkrai, C.

2019. Better exploiting latent variables in text modeling. *ACL*, Pp. 5527–5532.

Lake, B. M., R. Salakhutdinov, and J. B. Tenenbaum

2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–8.

Larochelle, H. and I. Murray

2011. The Neural Autoregressive Distribution Estimator. In *AISTATS*, Pp. 29–37.

Le Fang, Chunyuan Li, J. G. W. D. C. C.

2019. Implicit deep latent variable models for text generation. In *EMNLP*.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner

1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

Li, C., H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin

2017. ALICE: Towards understanding adversarial learning for joint distribution matching. In *NIPS*, Pp. 5495–5503.

Li, R., X. Li, C. Lin, M. Collinson, and R. Mao

2019a. A stable variational autoencoder for text modelling. *INLG*, Pp. 594–599.

Li, R., X. Li, C. Lin, M. Collinson, and R. Mao

2019b. A stable variational autoencoder for text modelling. In *INLG*.

Liu, S., X. Zhang, J. Wangni, and J. Shi

2019. Normalized diversification. *CoRR*, abs/1904.03608.

Liu, Z., P. Luo, X. Wang, and X. Tang

2015. Deep learning face attributes in the wild. In *ICCV*, Pp. 3730–3738.

Livne, M., K. Swersky, and D. J. Fleet

2019. MIM: Mutual Information Machine. *arXiv e-prints*.

Maddison, C. J., A. Mnih, and Y. W. Teh

2016. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712.

Makhzani, A.

2018. Implicit autoencoders. *CoRR*, abs/1805.09804.

Makhzani, A., J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey

2015. Adversarial autoencoders. In *ICLR Workshop*.

Marcus, M. P., M. A. Marcinkiewicz, and B. Santorini

1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Merity, S., N. S. Keskar, and R. Socher

2017. Regularizing and optimizing LSTM language models. *CoRR*, abs/1708.02182.

Merity, S., C. Xiong, J. Bradbury, and R. Socher

2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Netzer, Y., T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng

2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Oliver, A., A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow

2018. Realistic evaluation of deep semi-supervised learning algorithms. *CoRR*, abs/1804.09170.

Oord, A. v. d., N. Kalchbrenner, and K. Kavukcuoglu

2016. Pixel recurrent neural networks. *International Conference on Machine Learning*.

Papamakarios, G., T. Pavlakou, and I. Murray

2017. Masked autoregressive flow for density estimation. In *NIPS*, Pp. 2335–2344.

Papaspiliopoulos, O., G. O. Roberts, and M. Skold

2003. Non-centered parameterisations for hierarchical models and data augmentation. In *Bayesian Statistics*, Pp. 307–326. Oxford University Press.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu

2001. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Poon, H. and P. M. Domingos

2012. Sum-product networks: A new deep architecture. *CoRR*, abs/1202.3732.

Pu, Y., W. Wang, R. Henao, L. Chen, Z. Gan, C. Li, and L. Carin

2017. Adversarial symmetric variational autoencoder. In *NIPS*, Pp. 4330–4339.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever

2019. Language models are unsupervised multitask learners.

Rae, J. W., C. Dyer, P. Dayan, and T. P. Lillicrap

2018. Fast parametric learning with activation memorization. *CoRR*, abs/1803.10049.

Ramachandran, P., B. Zoph, and Q. V. Le

2018. Searching for activation functions. In *ICLR*.

Razavi, A., A. van den Oord, B. Poole, and O. Vinyals

2019. Preventing posterior collapse with delta-vaes. *CoRR*, abs/1901.03416.

Rezende, D. J. and S. Mohamed

2015. Variational inference with normalizing flows. In *ICML*.

Rezende, D. J., S. Mohamed, and D. Wierstra

2014. Stochastic backpropagation and approximate inference in deep tive Models. In *ICML*.

Rezende, D. J. and F. Viola

2018. Taming vaes. *ArXiv*, abs/1810.00597.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams

1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, Pp. 318–362. Cambridge, MA, USA: MIT Press.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei

2015. ImageNet large scale visual recognition challenge. *Int. J. Computer Vision*, 115(3):211–252.

Salimans, T., I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen

2016. Improved techniques for training gans. *CoRR*, abs/1606.03498.

Schmah, T., G. E. Hinton, S. L. Small, S. Strother, and R. S. Zemel

2009. Generative versus discriminative training of RBMs for classification of fMRI images. In *NIPS*, Pp. 1409–1416.

Shah, H. and D. Barber

2018. Generative neural machine translation. In *NeurIPS*.

Shu, R., J. Lee, H. Nakayama, and K. Cho

2019. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. *AAAI*, abs/1908.07181.

Smolensky, P.

1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, Pp. 194–281. Cambridge, MA, USA: MIT Press.

Sutskever, I., J. Martens, G. Dahl, and G. Hinton

2013. On the importance of initialization and momentum in deep learning. In *ICML*, S. Dasgupta and D. McAllester, eds., volume 28 of *Proceedings of Machine Learning Research*, Pp. 1139–1147, Atlanta, Georgia, USA. PMLR.

Sutskever, I., O. Vinyals, and Q. V. Le

2014. Sequence to sequence learning with neural networks. In *NIPS*, NIPS, Pp. 3104–3112, Cambridge, MA, USA. MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour

1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, Pp. 1057–1063, Cambridge, MA, USA. MIT Press.

Tay, Y., A. T. Luu, and S. C. Hui

2017a. Enabling efficient question answer retrieval via hyperbolic neural networks. *CoRR*, abs/1707.07847.

Tay, Y., M. C. Phan, A. T. Luu, and S. C. Hui

2017b. Learning to rank question answer pairs with holographic dual LSTM architecture. In *SIGIR*, Pp. 695–704.

Tomczak, J. M. and M. Welling

2017. VAE with a vampprior. *CoRR*, abs/1705.07120.

Tucker, G., A. Mnih, C. J. Maddison, and J. Sohl-Dickstein

2017. REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. *CoRR*, abs/1703.07370.

van den Berg, R., L. Hasenclever, J. Tomczak, and M. Welling

2018. Sylvester normalizing flows for variational inference. In *UAI*.

van den Oord, A., N. Kalchbrenner, and K. Kavukcuoglu

2016. Pixel recurrent neural networks. *CoRR*, abs/1601.06759.

van den Oord, A., Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis

2017a. Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433.

van den Oord, A., Y. Li, and O. Vinyals

2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

van den Oord, A., O. Vinyals, and K. Kavukcuoglu

2017b. Neural discrete representation learning. *CoRR*, abs/1711.00937.

van den Oord, A., O. Vinyals, and K. Kavukcuoglu

2017c. Neural discrete representation learning. In *NIPS*, Pp. 6306–6315.

van der Maaten, L. and G. E. Hinton

2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin

2017. Attention is all you need. In *NIPS*, Pp. 5998–6008.

Wang, D., C. Gong, and Q. Liu

2019. Improving neural language modeling via adversarial training. In *ICML*, K. Chaudhuri and R. Salakhutdinov, eds., volume 97 of *Proceedings of Machine Learning Research*, Pp. 6555–6565, Long Beach, California, USA. PMLR.

Williams, R. J.

1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Xiao, H., K. Rasul, and R. Vollgraf

2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747.

Yang, Z., Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick

2017. Improved variational autoencoders for text modeling using dilated convolutions. *CoRR*, abs/1702.08139.

Zhang, B., D. Xiong, J. Su, H. Duan, and M. Zhang

2016. Variational neural machine translation. In *EMNLP*.

Zhao, S., J. Song, and S. Ermon

2017. Infovae: Information maximizing variational autoencoders. *ArXiv*, abs/1706.02262.

Zhao, S., J. Song, and S. Ermon

2018a. The information autoencoding family: A Lagrangian perspective on latent variable generative models. In *UAI*.

Zhao, S., J. Song, and S. Ermon

2018b. A lagrangian perspective on latent variable generative models. *UAI*.

Zhu, J.-Y., T. Park, P. Isola, and A. A. Efros

2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

Łukasz Kaiser, A. Roy, A. Vaswani, N. Parmar, S. Bengio, J. Uszkoreit, and N. Shazeer

2018. Fast decoding in sequence models using discrete latent variables. In *ICML*.