Recent Advances in Autoencoder-Based Representation Learning

Michael Tschannen
ETH Zurich
michaelt@nari.ee.ethz.ch

Olivier Bachem
Google AI, Brain Team
bachem@google.com

Mario Lucic
Google AI, Brain Team
lucic@google.com

Abstract

Learning useful representations with little or no supervision is a key challenge in artificial intelligence. We provide an in-depth review of recent advances in representation learning with a focus on autoencoder-based models. To organize these results we make use of *meta-priors* believed useful for downstream tasks, such as disentanglement and hierarchical organization of features. In particular, we uncover three main mechanisms to enforce such properties, namely (i) regularizing the (approximate or aggregate) posterior distribution, (ii) factorizing the encoding and decoding distribution, or (iii) introducing a structured prior distribution. While there are some promising results, implicit or explicit supervision remains a key enabler and all current methods use strong inductive biases and modeling assumptions. Finally, we provide an analysis of autoencoder-based representation learning through the lens of rate-distortion theory and identify a clear tradeoff between the amount of prior knowledge available about the downstream tasks, and how useful the representation is for this task.

1 Introduction

The ability to learn useful representations of data with little or no supervision is a key challenge towards applying artificial intelligence to the vast amounts of unlabelled data collected in the world. While it is clear that the usefulness of a representation learned on data heavily depends on the end task which it is to be used for, one could imagine that there exists properties of representations which are useful for many real-world tasks simultaneously. In a seminal paper on representation learning Bengio et al. [1] proposed such a set of *meta-priors*. The meta-priors are derived from general assumptions about the world such as the hierarchical organization or disentanglement of explanatory factors, the possibility of semi-supervised learning, the concentration of data on low-dimensional manifolds, clusterability, and temporal and spatial coherence.

Recently, a variety of (unsupervised) representation learning algorithms have been proposed based on the idea of *autoencoding* where the goal is to learn a mapping from high-dimensional observations to a lower-dimensional representation space such that the original observations can be reconstructed (approximately) from the lower-dimensional representation. While these approaches have varying motivations and design choices, we argue that essentially all of the methods reviewed in this paper implicitly or explicitly have at their core at least one of the meta-priors from Bengio et al. [1].

Given the unsupervised nature of the upstream representation learning task, the characteristics of the meta-priors enforced in the representation learning step determine how useful the resulting representation is for the real-world end task. Hence, it is critical to understand which meta-priors are targeted by which models and which generic techniques are useful to enforce a given meta-prior. In this paper, we provide a unified view which encompasses the majority of proposed models and relate them to the meta-priors proposed by Bengio et al. [1]. We summarize the recent work focusing on the meta-priors in Table 1.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

Table 1: Grouping of methods according to the meta-priors for representation learning from [1]. While many methods directly or indirectly address multiple meta-priors, we only considered the most prominent target of each method. Note that meta-priors such as low dimensionality and manifold structure are enforced by essentially all methods.

Meta-prior	Methods
Disentanglement	β-VAE (6) [2], FactorVAE (8) [3], β-TCVAE (9) [4], InfoVAE (9) [5], DIP-VAE (11) [6], HSIC-VAE (12) [7], HFVAE (13) [8], VIB [9], Information dropout (15) [10], DC-IGN [11], FaderNetworks (18) [12], VFAE (17) [13]
Hierarchical representation[1]	PixelVAE [14], LVAE [15], VLaAE [16], Semi-supervised VAE [17], PixelGAN-AE [18], VLAE [19], VQ-VAE [20]
Semi-supervised learning	Semi-supervised VAE [17], [21], PixelGAN-AE (14) [18], AAE (16) [22]
Clustering	PixelGAN-AE (14) [18], AAE (16) [22], JointVAE [23], SVAE [24]

Meta-priors of Bengio et al. [1]. Meta-priors capture very general premises about the world and are therefore arguably useful for a broad set of downstream tasks. We briefly summarize the most important meta-priors which are targeted by the reviewed approaches.

1. **Disentanglement:** Assuming that the data is generated from independent factors of variation, for example object orientation and lighting conditions in images of objects, disentanglement as a meta-prior encourages these factors to be captured by different independent variables in the representation. It should result in a concise abstract representation of the data useful for a variety of downstream tasks and promises improved sample efficiency.
2. **Hierarchical organization of explanatory factors:** The intuition behind this meta-prior is that the world can be described as a hierarchy of increasingly abstract concepts. For example natural images can be abstractly described in terms of the objects they show at various levels of granularity. Given the object, a more concrete description can be given by object attributes.
3. **Semi-supervised learning:** The idea is to share a representation between a supervised and an unsupervised learning task which often leads to synergies: While the number of labeled data points is usually too small to learn a good predictor (and thereby a representation), training jointly with an unsupervised target allows the supervised task to learn a representation that generalizes, but also guides the representation learning process.
4. **Clustering structure:** Many real-wold data sets have multi-category structure (such as images showing different object categories), with possibly category-dependent factors of variation. Such structure can be captured with a latent mixture model where each mixture component corresponds to one category, and its distribution models the factors of variation within that category. This naturally leads to a representation with clustering structure.

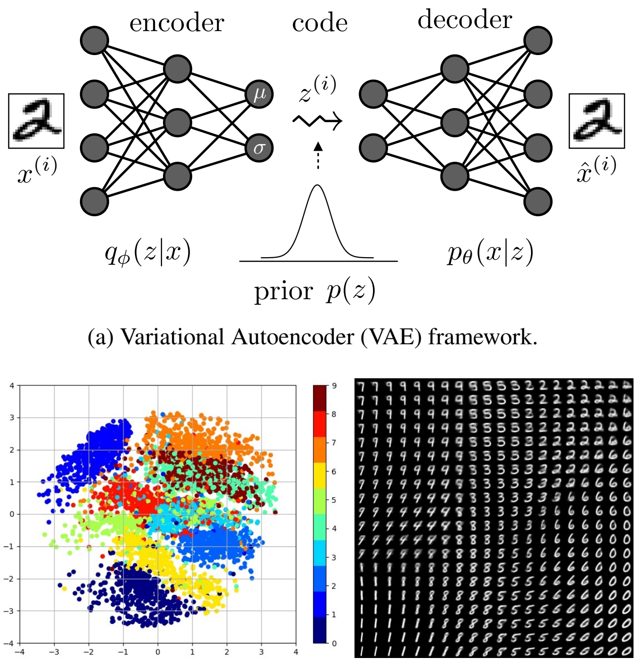
Very generic concepts such as *smoothness* as well as *temporal and spatial coherence* are not specific to unsupervised learning and are used in most practical setups (for example weight decay to encourage smoothness of predictors, and convolutional layers to capture spatial coherence in image data). We discuss the implicit supervision used by most approaches in Section 7.

Mechanisms for enforcing meta-priors. We identify the following three mechanisms to enforce meta-priors:

(i) Regularization of the encoding distribution (Section 3).
(ii) Choice of the encoding and decoding distribution or model family (Section 4).
(iii) Choice of a flexible prior distribution of the representation (Section 5).

For example, regularization of the encoding distribution is often used to encourage disentangled representations. Alternatively, factorizing the encoding and decoding distribution in a hierarchical fashion allows us to impose a hierarchical structure to the representation. Finally, a more flexible prior, say a mixture distribution, can be used to encourage clusterability.

[1] While PixelGAN-AE [18], VLAE [19], and VQ-VAE [20] do not explicitly model a hierarchy of latents, they learn abstract representations capturing global structure of images [18, 19] and speech signals [20], hence internally representing the data in a hierarchical fashion.



(a) Variational Autoencoder (VAE) framework.

(b) Samples from a trained VAE.

Figure 1: Figure (a) illustrates the Variational Autoencoder (VAE) framework specified by the encoder, decoder, and the prior distribution on the latent (representation/code) space. The encoder maps the input to the representation space (*inference*), while the decoder reconstructs the original input from the representation. The encoder is encouraged to satisfy some structure on the latent space (e.g., it should be disentangled). Figure (b) shows samples from a trained autoencoder with latent space of 2 dimensions on the MNIST data set. Each point on the left corresponds to the representation of a digit (originally in 784 dimensions) and the reconstructed digits can be seen on the right. One can observe that in this case the latent representation is clustered (various styles of the same digit are close w.r.t. L_2-distance, and within each group the position corresponds to the rotation of the digit).

Before starting our overview, in Section 2 we present the main concepts necessary to understand variational autoencoders (VAEs) [25, 26], underlying most of the methods considered in this paper, and several techniques used to estimate divergences between probability distributions. We then present a detailed discussion of regularization-based methods in Section 3, review methods relying on structured encoding and decoding distributions in Section 4, and present methods using a structured prior distribution in Section 5. We conclude the review section by an overview of related methods such as cross-domain representation learning [27–29] in Section 6. Finally, we provide a critique of unsupervised representation learning through the rate-distortion framework of Alemi et al. [30] and discuss the implications in Section 7.

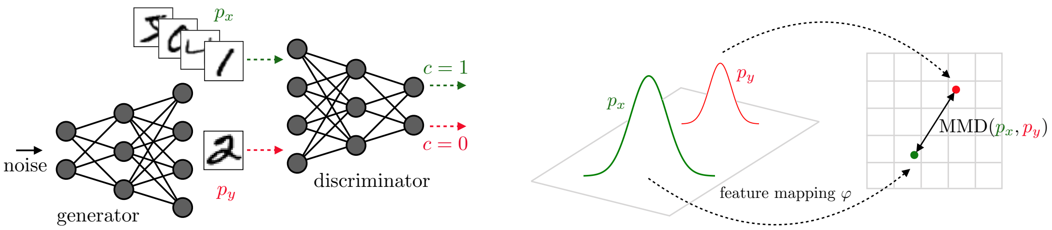
2 Preliminaries

We assume familiarity with the key concepts in Bayesian data modeling. For a gentle introduction to VAEs we refer the reader to [31]. VAEs [25, 26] aim to learn a parametric latent variable model by maximizing the marginal log-likelihood of the training data $\{x^{(i)}\}_{i=1}^{N}$. By introducing an approximate posterior $q_\phi(z|x)$ which is an approximation of the intractable true posterior $p_\theta(z|x)$ we can rewrite the negative log-likelihood as

$$\mathbb{E}_{\hat{p}(x)}[-\log p_\theta(x)] = \mathcal{L}_{\text{VAE}}(\theta, \phi) - \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x) \| p_\theta(z|x))]$$

where

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = \mathbb{E}_{\hat{p}(x)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x) \| p(z))], \tag{1}$$



(a) The main idea behind GANs.

(b) The main idea behind MMD.

Figure 2: Adversarial density ratio estimation vs MMD. Figure (a): GANs use adversarial density ratio estimation to train a generative model, which can be seen as a two-player game: The discriminator tries to predict whether samples are real or generated, while the generator tries to deceive the discriminator by mimicking the distribution of the real samples. Figure (b): The MMD corresponds to the distance between mean feature embeddings.

and $\mathbb{E}_{\hat{p}(x)}[f(x)] = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$ is the expectation of the function $f(x)$ w.r.t. the empirical data distribution. The approach is illustrated in Figure 1. The first term in (1) measures the reconstruction error and the second term quantifies how well $q_\phi(z|x)$ matches the prior $p(z)$. The structure of the latent space heavily depends on this prior. As the KL divergence is non-negative, $-\mathcal{L}_{\text{VAE}}$ lower-bounds the marginal log-likelihood $\mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)]$ and is accordingly called the *evidence lower bound (ELBO)*.

There are several design choices available: (1) The prior distribution on the latent space, $p(z)$, (2) the family of approximate posterior distributions, $q_\phi(z|x)$, and (3) the decoder distribution, $p_\phi(x|z)$. Ideally, the approximate posterior should be flexible enough to match the intracable true posterior $p_\theta(z|x)$. As we will see later, there are many available options for these design choices, leading to various trade-offs in terms of the learned representation.

In practice, the first term in (1) can be estimated from samples $z^{(i)} \sim q_\phi(z|x^{(i)})$ and gradients are backpropagated through the sampling operation using the *reparametrization trick* [25, Section 2.3], enabling minimization of (1) via minibatch-stochastic gradient descent (SGD). Depending on the choice of $q_\phi(z|x)$ the second term can either be computed in closed form or estimated from samples. For the usual choice of $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \text{diag}(\sigma_\phi(x)))$, where $\mu_\phi(x)$ and $\sigma_\phi(x)$ are deterministic functions parametrized as neural networks, and $p(z) = \mathcal{N}(0, I)$ for which the KL-term in (1) can be computed in closed form (more complicated choices of $p(z)$ rarely allow closed form computation). To this end, we will briefly discuss two ways in which one can measure distances between distributions. We will focus on intuition behind these techniques and provide pointers to detailed expositions.

Adversarial density-ratio estimation. Given a convex function f for which $f(1) = 0$, the f-divergence between p_x and p_y is defined as

$$D_f(p_x \| p_y) = \int f\left(\frac{p_x(x)}{p_y(x)}\right) p_y(x) dx.$$

For example, the choice $f(t) = t \log t$ corresponds to $D_f(p_x \| p_y) = D_{\text{KL}}(p_x \| p_y)$. Given samples from p_x and p_y we can estimate the f-divergence using the density-ratio trick [32, 33], popularized recently through the generative adversarial network (GAN) framework [34]. The trick is to express p_x and p_y as conditional distributions, conditioned on a label $c \in \{0, 1\}$, and reduce the task to binary classification. In particular, let $p_x(x) = p(x|c = 1)$, $p_y(x) = p(x|c = 0)$, and consider a discriminator S_η trained to predict the probability that its input is a sample from distributions p_x rather than p_y, i.e, predict $p(c = 1|x)$. The density ratio can be expressed as

$$\frac{p_x(x)}{p_y(x)} = \frac{p(x|c=1)}{p(x|c=0)} = \frac{p(c=1|x)}{p(c=0|x)} \approx \frac{S_\eta(x)}{1 - S_\eta(x)}, \tag{2}$$

where the second equality follows from Bayes' rule under the assumption that the marginal class probabilities are equal. As such, given N i.i.d. samples $\{x^{(i)}\}_{i=1}^{N}$ from p_x and a trained classifier

S_η one can estimate the KL-divergence by simply computing

$$D_{\text{KL}}(p_x \| p_y) \approx \frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{S_\eta(x^{(i)})}{1 - S_\eta(x^{(i)})} \right).$$

As a practical alternative, some approaches replace the KL term in (1) with an arbitrary divergence (e.g., maximum mean discrepancy). Note, however, that the resulting objective does not necessarily lower-bound the marginal log-likelihood of the data.

Maximum mean discrepancy (MMD) [35]. Intuitively, the distances between distributions are computed as distances between mean embeddings of features as illustrated in Figure 2b. More formally, let $k\colon \mathcal{X} \to \mathcal{X}$ be a continuous, bounded, positive semi-definite kernel and $\mathcal{H}$ be the corresponding reproducing kernel Hilbert space, induced by the feature mapping $\varphi\colon \mathcal{X} \to \mathcal{H}$. Then, the MMD of distributions $p_x(x)$ and $p_y(y)$ is

$$\text{MMD}(p_x, p_y) = \|\mathbb{E}_{x \sim p_x}[\varphi(x)] - \mathbb{E}_{y \sim p_y}[\varphi(y)]\|^2_{\mathcal{H}}. \tag{3}$$

For example, setting $\mathcal{X} = \mathcal{H} = \mathbb{R}^d$ and $\varphi(x) = x$, MMD reduces to the difference between the means, i.e., $\text{MMD}(p_x, p_y) = \|\mu_{p_x} - \mu_{p_y}\|^2_2$. By choosing an appropriate mapping φ one can estimate the divergence in terms of higher order moments of the distribution.

MMD vs f-divergences in practice. The MMD is known to work particularly well with multivariate standard normal distributions. It requires a sample size roughly on the order of the data dimensionality. When used as a regularizer (see Section 3), it generally allows for stable optimization. A disadvantage is that it requires selection of the kernel k and its bandwidth parameter. In contrast, f-divergence estimators based on the density-ratio trick can in principle handle more complex distributions than MMD. However, in practice they require adversarial training which currently suffers from optimization issues. For more details consult [36, Section 3].

Deterministic autoencoders. Some of the methods we review rely on deterministic encoders and decoders. We denote by D_θ and E_ϕ the deterministic encoder and decoder, respectively. A popular objective for training an autoencoder is to minimize the L_2-loss, namely

$$\mathcal{L}_{\text{AE}}(\theta, \phi) = \frac{1}{2} \mathbb{E}_{\hat{p}(x)}[\|x - D_\theta(E_\phi(x))\|^2_2]. \tag{4}$$

If E_ϕ and D_θ are linear maps and the representation z is lower-dimensional than x, (4) corresponds to principal component analysis (PCA), which leads to z with decorrelated entries. Furthermore, we obtain (4) by removing the D_{KL}-term from $\mathcal{L}_{\text{VAE}}$ in (1) and using a deterministic encoding distribution $q_\phi(z|x)$ and a Gaussian decoding distribution $p_\theta(x|z)$. Therefore, the major difference between $\mathcal{L}_{\text{AE}}$ and $\mathcal{L}_{\text{VAE}}$ is that $\mathcal{L}_{\text{AE}}$ does not enforce a prior distribution on the latent space (e.g., through a D_{KL}-term), and minimizing $\mathcal{L}_{\text{AE}}$ hence does not yield a generative model.

3 Regularization-based methods

A classic approach to enforce some meta-prior on the latent representations $z \sim q_\phi(z|x)$ is to augment $\mathcal{L}_{\text{VAE}}$ with regularizers that act on the approximate posterior $q_\phi(z|x)$ and/or the aggregate (approximate) posterior $q_\phi(z) = \mathbb{E}_{\hat{p}(x)}[q_\phi(z|x)] = \frac{1}{N}\sum_{i=1}^{N} q_\phi(z|x^{(i)})$. The vast majority of recent work can be subsumed into an objective of the form

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[R_1(q_\phi(z|x))] + \lambda_2 R_2(q_\phi(z)), \tag{5}$$

where R_1 and R_2 are regularizers and $\lambda_1, \lambda_2 > 0$ the corresponding weights. Firstly, we note a key difference between regularizers R_1 and R_2 is that the latter depends on the entire data set through $q_\phi(z)$. In principle, this prevents the use of mini-batch SGD to solve (5). In practice, however, one can often obtain good mini-batch-based estimates of $R_2(q_\phi(z))$. Secondly, the regularizers bias $\mathcal{L}_{\text{VAE}}$ towards a looser (larger) upper bound on the negative marginal $\log$-likelihood. From this perspective it is not surprising that many approaches yield a lower reconstruction quality (which typically corresponds to a larger negative $\log$-likelihood). For deterministic autoencoders, there is no such concept as an aggregated posterior, so we consider objectives of the form $\mathcal{L}_{\text{AE}}(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[R_1(E(x))]$.

Table 2: Overview over different choices of the regularizers $R_1(q_\phi(z|x))$ and $R_2(q_\phi(z))$. The learning objective is specified in (5). Most approaches use a multivariate standard normal distribution as prior (see Table 3 in the appendix for more details). The last column (Y) indicates whether supervision is used: (✓) indicates that labels are required, while (O) indicates labels can optionally be used for (semi-) supervised learning. Note that some of the regularizers are simplified.

WORK	$\mathcal{L}$.	R_1	R_2	Y
β-VAE [2]	VAE	$D_{\text{KL}}(q_\phi(z \vert x) \Vert p(z))$		
VIB [9]	VAE	$D_{\text{KL}}(q_\phi(z \vert x) \Vert p(z))$		O
PixelGAN-AE[18]	VAE	$-I_{q_\phi}(x; z)$		O
InfoVAE [5]	VAE	$D_{\text{KL}}(q_\phi(z \vert x) \Vert p(z))$	$D_{\text{KL}}(q_\phi(z) \Vert p(z))$	
Info. dropout [10]	VAE	$D_{\text{KL}}(q_\phi(z \vert x) \Vert p(z))$	$\text{TC}(q_\phi(z))$	O
HFVAE [8]	VAE	$-I_{q_\phi}(x; z)$	$R_{\mathcal{G}}(q_\phi(z)) + \lambda_2' \sum_{G \in \mathcal{G}} R_{\mathcal{G}}(q_\phi(z))$	
FactorVAE [3, 4]	VAE		$\text{TC}(q_\phi(z))$	
DIP-VAE [6]	VAE		$\Vert \text{Cov}_{q_\phi(z)}[z] - I \Vert_{\text{F}}^2$	
HSIC-VAE [7]	VAE		$\text{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2}))$	O
VFAE [13]	VAE		$\text{MMD}(q_\phi(z \vert s = 0), q_\phi(z \vert s = 1))$	✓
DC-IGN [11]	VAE			✓
FaderNet. [12]; [37][2]	AE	$-\mathbb{E}_{\hat{p}(x,y)}[\log P_\psi(1 - y \vert E_\phi(x))]$		✓
AAE/WAE [22, 36]	AE		$D_{\text{JS}}(E_\phi(z) \Vert p(z))$	O

In this section, we first review regularizers which can be computed in a fully unsupervised fashion (some of them optionally allow to include partial label information). Then, we turn our attention to regularizers which require supervision.

3.1 Unsupervised methods targeting disentanglement and independence

Disentanglement is a critical meta-prior considered by Bengio et al. [1]. Namely, assuming the data is generated from a few statistically independent factors, uncovering those factors should be extremely useful for a plethora of downstream tasks. An example for (approximately) independent factors underlying the data are class, stroke thickness, and rotation of handwritten digits in the MNIST data set. Other popular data sets are the CelebA face data set [38] (factors involve, e.g., hair color and facial attributes such as glasses), and synthetic data sets of geometric 2D shapes or rendered 3D shapes (e.g., 2D Shapes [2], 3D Shapes [3], 3D Faces [39], 3D Chairs [40]) for which the data generative process and hence the ground truth factors are known (see Figure 4 for an example).

The main idea behind several recent works on disentanglement is to augment the $\mathcal{L}_{\text{VAE}}$ loss with regularizers which encourage disentanglement of the latent variables z. Formally, assume that the data $x \sim p(x|v, w)$ depends on conditionally independent factors v, i.e., $p(v|x) = \prod_j p(v_j|x)$, and possibly conditionally dependent factors w. The goal is to augment $\mathcal{L}_{\text{VAE}}$ such that the inference model $q_\phi(z|x)$ learns to predict v and hence (partially) invert the data-generative process.

Metrics. Disentanglement quality of inference models is typically evaluated based on ground truth factors of variation (if available). Specifically, disentanglement metrics measure how predictive the individual latent factors are for the ground-truth factors, see, e.g., [2, 3, 6, 41, 4, 42]. While many authors claim that their method leads to disentangled representations, it is unclear what the proper notion of disentanglement is and how effective these methods are in the unsupervised setting (see [43] for a large-scale evaluation). We therefore focus on the concept motivating each method rather than claims on how well each method disentangles the factors underlying the data.

3.1.1 Reweighting the ELBO: β-VAE

Higgins et al. [2] propose to weight the second term in (1) (henceforth referred to as the D_{KL}-term) by a coefficient $\beta > 1$,[3] which can be seen as adding a regularizer equal to the D_{KL}-term with

[2] Lample et al. [12], Hadad et al. [37] do not enforce a prior on the latent distribution and therefore cannot generate unconditionally.

[3] Higgins et al. [2] also explore $0 < \beta < 1$ but discovers that this choice does not lead to disentanglement.

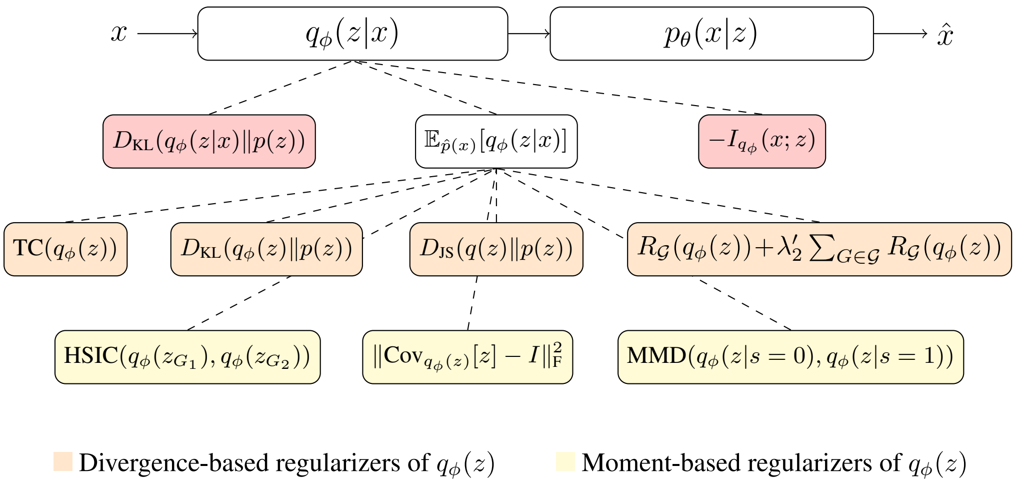

x
$q_\phi(z|x)$
$p_\theta(x|z)$
$\hat{x}$
$D_{\text{KL}}(q_\phi(z|x)\|p(z))$
$\mathbb{E}_{\hat{p}(x)}[q_\phi(z|x)]$
$-I_{q_\phi}(x;z)$
$\text{TC}(q_\phi(z))$
$D_{\text{KL}}(q_\phi(z)\|p(z))$
$D_{\text{JS}}(q(z)\|p(z))$
$R_{\mathcal{G}}(q_\phi(z)) + \lambda_2' \sum_{G\in\mathcal{G}} R_{\mathcal{G}}(q_\phi(z))$
$\text{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2}))$
$\|\text{Cov}_{q_\phi(z)}[z] - I\|_{\text{F}}^2$
$\text{MMD}(q_\phi(z|s=0), q_\phi(z|s=1))$
Divergence-based regularizers of $q_\phi(z)$
Moment-based regularizers of $q_\phi(z)$

Figure 3: Schematic overview over different regularizers. Most approaches focus on regularizing the aggregate posterior and differ in the way the disagreement with respect to a prior is measured. More details are provided in Table 2 and an in-depth discussion in Section 3.

coefficient $\lambda_1 = \beta - 1 > 0$ to $\mathcal{L}_{\text{VAE}}$

$$\mathcal{L}_{\beta\text{-VAE}}(\theta,\phi) = \mathcal{L}_{\text{VAE}}(\theta,\phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x)\|p(z))]. \tag{6}$$

This type of regularization encourages $q_\phi(z|x)$ to better match the factorized prior $p(z)$, which in turn constrains the implicit capacity of the latent representation $z \sim q_\phi(z|x)$ and encourages it be factorized. Burgess et al. [44] provide a through theoretical analysis of β-VAE based on the information bottleneck principle [45]. Further, they propose to gradually decrease the regularization strength until good quality reconstructions are obtained as a robust procedure to adjust the tradeoff between reconstruction quality and disentanglement (for a hard-constrained variant fo β-VAE).

3.1.2 Mutual information of x and z: FactorVAE, β-TCVAE, InfoVAE

Kim and Mnih [3], Chen et al. [4], Zhao et al. [5] all propose regularizers motivated by the following decomposition of the second term in (1)

$$\mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x)\|p(z)] = I_{q_\phi}(x;z) + D_{\text{KL}}(q_\phi(z)\|p(z)), \tag{7}$$

where $I_{q_\phi}(x;z)$ is the mutual information of x and z w.r.t. the distribution $q_\phi(x,z) = q_\phi(z|x)\hat{p}(x) = \frac{1}{N}\sum_{i=1}^N q_\phi(z|x^{(i)})\delta_{x^{(i)}}(x)$. The decomposition (7) was first derived by Hoffman and Johnson [46]; an alternative derivation can be found in Kim and Mnih [3, Appendix C].

FactorVAE. Kim and Mnih [3] observe that the regularizer in $\mathcal{L}_{\beta\text{-VAE}}$ encourages $q_\phi(z)$ to be factorized (assuming $p(z)$ is a factorized distribution) by penalizing the second term in (7), but discourages the latent code to be informative by simultaneously penalizing the first term in (7). To reinforce only the former effect, they propose to regularize $\mathcal{L}_{\text{VAE}}$ with the total correlation $\text{TC}(q_\phi(z)) = D_{\text{KL}}(q_\phi(z)\|\prod_j q_\phi(z_j))$ of $q_\phi(z)$—a popular measure of dependence for multiple random variables [47]. The resulting objective has the form

$$\mathcal{L}_{\text{FactorVAE}}(\theta,\phi) = \mathcal{L}_{\text{VAE}}(\theta,\phi) + \lambda_2 \text{TC}(q_\phi(z)) \tag{8}$$

where the last term is the total correlation. To estimate it from samples, Kim and Mnih [3] rely on the density ratio trick [32, 33] which involves training a discriminator (see Section 2).

β-TCVAE. Chen et al. [4] split up the second term in (7) as $D_{\text{KL}}(p(z)\|q_\phi(z)) = D_{\text{KL}}(q_\phi(z)\|\prod_j q_\phi(z_j)) + \sum_{j=1}^m D_{\text{KL}}(q_\phi(z_j)\|p(z_j))$ and penalize each term individually

$$\mathcal{L}_{\beta\text{-TCVAE}}(\theta,\phi) = \mathcal{L}_{\text{VAE}}(\theta,\phi) + \lambda_1 I_{q_\phi}(x;z) + \lambda_2 \text{TC}(q_\phi(z)) + \lambda_2' \sum_j D_{\text{KL}}(q_\phi(z_j)\|p(z_j)).$$

However, they set $\lambda_1 = \lambda_2' = 0$ by default, effectively arriving at the same objective as FactorVAE in (8). In contrast to FactorVAE, the TC-term is estimated using importance sampling.

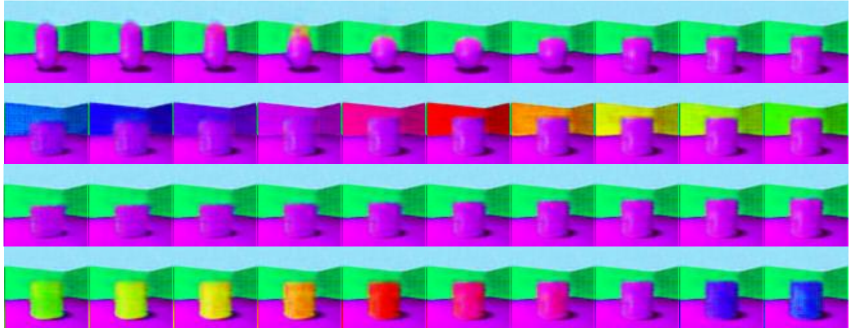


Figure 4: Latent space traversals obtained by varying one latent variable while holding all the others fixed (from left to right), for β-VAE [2] (with $\beta = 16$) trained on the 3D Shapes data set [3]. The variables shown correspond to the object shape, wall color, object height, and object color. Note that the other latent variables simultaneously vary multiple latent factors or are inactive for this model.

InfoVAE. Zhao et al. [5] start from an alternative way of writing $\mathcal{L}_{\text{VAE}}$

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = D_{\text{KL}}(q_\phi(z) \| p(z)) + \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(x|z) \| p_\theta(x|z))], \tag{9}$$

where $q_\phi(x|z) = q_\phi(x, z)/p(z)$. Similarly to [3], to encourage disentanglement, they propose to reweight the first term in (9) and to encourage a large mutual information between $z \sim q(z|x)$ and x by adding a regularizer proportional to $I_{q_\phi}(x; z)$ to (9). Further, by rearranging terms in the resulting objective, they arrive at

$$\mathcal{L}_{\text{InfoVAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x) \| p(z))] + \lambda_2 D_{\text{KL}}(q_\phi(z) \| p(z)). \tag{10}$$

For tractability reasons, Zhao et al. [5] propose to replace the last term in (10) by other divergences such as Jensen-Shannon divergence (implemented as a GAN [34]), Stein variational gradient [48], or MMD [35] (see Section 2).

DIP-VAE. Kumar et al. [6] suggest matching the moments of the aggregated posterior $q_\phi(z)$ to a multivariate standard normal prior $p(z)$ during optimization of $\mathcal{L}_{\text{VAE}}$ to encourage disentanglement of the latent variables $z \sim q_\phi(z)$. Specifically, they propose to match the covariance of $q_\phi(z)$ and $\mathcal{N}(0, I)$ by penalizing their ℓ_2-distance (which amounts to decorrelating the entries of $z \sim q_\phi(z)$) leading to the Disentangled Inferred Prior objective:

$$\mathcal{L}_{\text{DIP-VAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_2 \sum_{k \neq \ell} (\text{Cov}_{q_\phi(z)}[z])^2_{k,\ell} + \lambda_2' \sum_k ((\text{Cov}_{q_\phi(z)}[z])_{k,k} - 1)^2. \tag{11}$$

Noting that for the standard parametrization $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \text{diag}(\sigma_\phi(x)))$, $\text{Cov}_{q_\phi(z)}[z] = \sum_{i=1}^N \text{diag}(\sigma_\phi(x_i)) + \text{Cov}_{\hat{p}(x)}[\mu_\phi(x)]$, $\sigma_\phi(x)$ only contributes to the diagonal of $\text{Cov}_{q_\phi(z)}[z]$, Kumar et al. [6] also consider a variant of $\mathcal{L}_{\text{DIP-VAE}}$ where $\text{Cov}_{q_\phi(z)}[z]$ in (11) is replaced by $\text{Cov}_{\hat{p}(x)}[\mu_\phi(x)]$.

3.1.3 Independence between groups of latents: HSIC-VAE, HFVAE

Groups/clusters, potentially involving hierarchies, is a structure prevalent in many data sets. It is therefore natural to take this structure into account when learning disentangled representations, as seen next.

HSIC-VAE. Lopez et al. [7] leverage the Hilbert-Schmidt independence criterion (HSIC) [49] (cf. Section A) to encourage independence between groups of latent variables, as

$$\mathcal{L}_{\text{HSIC-VAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_2 \text{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2})), \tag{12}$$

where $z_G = \{z_k\}_{k \in G}$ (an estimator of HSIC is defined in (21) in Appendix A). This is in contrast to the methods [3–6] penalizing statistical dependence of all individual latent variables. In addition to controling (in)dependence relations of the latent variables, the HSIC can be used to remove sensitive information, provided as labels s with the training data, from latent representation by using the regularizer $\text{HSIC}(q_\phi(z), p(s))$ (where $p(s)$ is estimated from samples) as extensively explored by Louizos et al. [13] (see Section 3.4).

HFVAE. Starting from the decomposition (7), Esmaeili et al. [8] hierarchically decompose the D_{KL}-term in (7) into a regularization term of the dependencies between groups of latent variables $\mathcal{G} = \{G_k\}_{k=1}^{n_G}$ and regularization of the dependencies between the random variables in each group G_k. Reweighting different regularization terms allows to encourage different degrees of intra and inter-group disentanglement, leading to the following objective:

$$\begin{aligned}\mathcal{L}_{\text{HFVAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}} &- \lambda_1 I_{q_\phi}(x; z) \\ &+ \lambda_2 \left(-\mathbb{E}_{q_\phi(z)} \left[\log \frac{p(z)}{\prod_{G \in \mathcal{G}} p(z_G)} \right] + D_{\text{KL}}(q_\phi(z) \| \prod_{G \in \mathcal{G}} q_\phi(z_G)) \right) \\ &+ \lambda_2' \sum_{G \in \mathcal{G}} \left(-\mathbb{E}_{q_\phi(z_G)} \left[\log \frac{p(z_G)}{\prod_{k \in G} p(z_k)} \right] + D_{\text{KL}}(q_\phi(z_G) \| \prod_{k \in G} q_\phi(z_k)) \right). \end{aligned} \tag{13}$$

Here, λ_1 controls the mutual information between the data and latent variables, and λ_2 and λ_2' determine the regularization of dependencies between groups and within groups, respectively, by penalizing the corresponding total correlation. Note that the grouping can be nested to introduce deeper hierarchies.

3.2 Preventing the latent code from being ignored: PixelGAN-AE and VIB

PixelGAN-AE. Makhzani and Frey [18] argue that, if $p_\theta(x|z)$ is not too powerful (in the sense that it cannot model the data distribution unconditionally, i.e., without using the latent code z) the term $I_{q_\phi}(x; z)$ in (7) and the reconstruction term in (1) have competing effects: A small mutual information $I_{q_\phi}(x; z)$ makes reconstruction of $x^{(i)}$ from $q_\phi(z|x^{(i)})$ challenging for $p_\theta(x|z)$, leading to a large reconstruction error. Conversely, a small reconstruction error requires the code z to be informative and hence $I_{q_\phi}(x; z)$ to be large. In contrast, if the decoder is powerful, e.g., a conditional PixelCNN [50], such that it can obtain a small reconstruction error without relying on the latent code, the mutual information and reconstruction terms can be minimized largely independent, which prevents the latent code from being informative and hence providing a useful representation (this issue is known as the information preference property [19] and is discussed in more detail in Section 4). In this case, to encourage the code to be informative Makhzani and Frey [18] propose to drop the $I_{q_\phi}(x; z)$ term in (7), which can again be seen as a regularizer

$$\mathcal{L}_{\text{PixelGAN-AE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}}(\theta, \phi) - I_{q_\phi}(x; z). \tag{14}$$

The D_{KL} term remaining in (7) after removing I_{q_ϕ} is approximated using a GAN. Makhzani and Frey [18] show that relying on $\mathcal{L}_{\text{PixelGAN-AE}}$ a powerful PixelCNN decoder can be trained while keeping the latent code informative. Depending on the choice of the prior (categorical or Gaussian), the latent code picks up information of different levels of abstraction, for example the digit class and writing style in the case of MNIST.

VIB, information dropout. Alemi et al. [9] and Achille and Soatto [10] both derive a variational approximation of the information bottleneck objective [10], which targets learning a compact representation z of some random variable x that is maximally informative about some random variable y. In the special case, when $y = x$, the approximation derived in [9] one obtains an objective equivalent to $\mathcal{L}_{\beta\text{-VAE}}$ in (1) (c.f. [9, Appendix B] for a discussion), whereas doing so for [10] leads to

$$\mathcal{L}_{\text{InfoDrop}}(\theta, \phi) = \mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[D_{\text{KL}}(q_\phi(z|x) \| p(z))] + \lambda_2 \text{TC}(q_\phi(z)). \tag{15}$$

Achille and Soatto [10] derive (more) tractable expressions for (15) and establishe a connection to dropout for particular choices of $p(z)$ and $q_\phi(z|x)$. Alemi et al. [30] propose an information-theoretic framework studying the representation learning properties of VAE-like models through a rate-distortion tradeoff. This framework recovers β-VAE but allows for a more precise navigation of the feasible rate-distortion region than the latter. Alemi and Fischer [51] further generalize the framework of [9], as discussed in Section 7.

3.3 Deterministic encoders and decoders: AAE and WAE

Adversarial Autoencoders (AAEs) [22] turn a standard autoencoder into a generative model by imposing a prior distribution $p(z)$ on the latent variables by penalizing some statistical divergence D_f

between $p(z)$ and $q_\phi(z)$ using a GAN. Specifically, using the negative $\log$-likelihood as reconstruction loss, the AAE objective can be written as

$$\mathcal{L}_{\text{AAE}}(\theta, \phi) = \mathbb{E}_{\hat{p}(x)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)]] + \lambda_2 D_f(q(z)\|p(z)). \tag{16}$$

In all experiments in [22] encoder and decoder are taken to be deterministic, i.e., $p(x|z)$ and $q(z|x)$ are replaced by D_θ and E_ϕ, respectively, and the negative $\log$-likelihood in (16) is replaced with the standard autoencoder loss $\mathcal{L}_{\text{AE}}$. The advantage of implementing the regularizer $\lambda_2 D_f$ using a GAN is that any $p(z)$ we can sample from, can be matched. This is helpful to learn representations: For example for MNIST, enforcing a prior that involves both categorical and Gaussian latent variables is shown to disentangle discrete and continuous style information in unsupervised fashion, in the sense that the categorical latent variables model the digit index and continuous random variables the writing style. Disentanglement can be improved by leveraging (partial) label information, regularizing the cross-entropy between the categorical latent variables and the label one-hot encodings. Partial label information also allows to learn a generative model for digits with a Gaussian mixture model prior, with every mixture component corresponding to one digit index.

3.4 Supervised methods: VFAEs, FaderNetworks, and DC-IGN

VFAE. Variational Fair Autoencoders (VFAEs) [13] assume a likelihood of the form $p_\theta(x|z, s)$, where s models (categorical) latent factors one wants to remove (for example sensitive information), and z models the remaining latent factors. By using an approximate posterior of the form $q_\phi(z|x, s)$ and by imposing factorized prior $p(z)p(s)$ one can encourage independence of $z \sim q_\phi(z|x, s)$ from s. However, z might still contain information about s, in particular in the (semi-) supervised setting where z encodes label information y that might be correlated with s, and additional factors of variation z', i.e., $z \sim p_\theta(z|z', y)$ (this setup was first considered in [17]; see Section 4). To mitigate this issue, Louizos et al. [13] propose to add an MMD-based regularizer to $\mathcal{L}_{\text{VAE}}$, encouraging independence between $q(z|s = k)$ and $q(z|s = k')$, i.e.,

$$\mathcal{L}_{\text{VFAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}} + \lambda_2 \sum_{\ell=2}^{K} \text{MMD}(q_\phi(z|s = \ell), q_\phi(z|s = 1)), \tag{17}$$

where $q_\phi(z|s = \ell) = \sum_{i\colon s^{(i)}=\ell} \frac{1}{|\{i\colon s^{(i)}=\ell\}|} q_\phi(z|x^{(i)}, s^{(i)})$. To reduce the computational complexity of the MMD the authors propose to use random Fourier features [52]. Lopez et al. [7] also consider the problem of censoring side information, but use the HSIC regularizer instead of MMD. In contrast to MMD, the HSIC is amenable to side information s of a non-categorical distribution. Furthermore, it is shown in Lopez et al. [7, Appendix E] that VFAE and HSIC are equivalent to censoring in case s is a binary random variable.

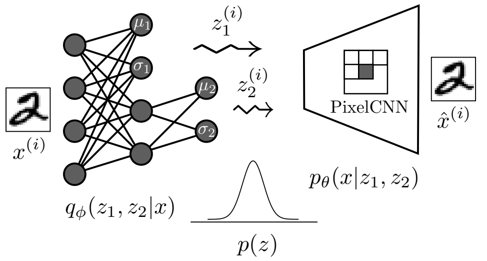
Fader Networks. A supervised method similar to censoring outlined above was explored by Lample et al. [12] and Hadad et al. [37]. Given data $\{x^{(i)}\}_{i=1}^N$ (e.g., images of faces) and corresponding binary attribute information $\{y^{(i)}\}_{i=1}^N$ (e.g., facial attributes such as hair color or whether glasses are present; encoded as binary vector in $\{0, 1\}^K$), the encoder of a FaderNetwork [12] is adversarially trained to learn a feature representation $z = E_\phi(x)$ invariant to the attribute values, and the decoder $D_\theta(y, z)$ reconstructing the original image from z and y. The resulting model is able to manipulate the attributes of a testing image (without known attribute information) by setting the entries of y at the input of D_θ as desired. In particular, it allows for continuous control of the attributes (by choosing non-integer attribute values in $[0, 1]$).

To make $z = E_\phi(x)$ invariant to y a discriminator $P_\psi(y|z)$ predicting the probabilities of the attribute vector y from z is trained concurrently with E_ϕ, D_θ to maximize the $\log$-likelihood $\mathcal{L}_{\text{dis}}(\psi) = \mathbb{E}_{\hat{p}(x,y)}[\log P_\psi(y|E_\phi(x))]$. This discriminator is used adversarially in the training of E_ϕ, D_θ encouraging E_ϕ to produce a latent code z from which it is difficult to predict y using P_ψ as

$$\mathcal{L}_{\text{Fader}}(\theta, \phi) = \mathbb{E}_{\hat{p}(x,y)}\left[\frac{1}{2}\|x - D_\theta(y, E_\phi(x))\|_2^2 - \lambda_1 \log P_\psi(1 - y|E_\phi(x))\right], \tag{18}$$

i.e., the regularizer encourages E_ϕ to produce codes for which P_ψ assigns a high likelihood to incorrect attribute values.

Hadad et al. [37] propose a method similar to FaderNetworks that first separately trains an encoder $z' = E'_{\phi'}(x)$ jointly with a classifier to predict y. The code produced by $E'_{\phi'}$ is then concatenated



(a) Hierarchical encoder + PixelCNN decoder.

Work	Enc	Dec	$P(z)$	Y
LadderVAE [15]	H	H	$\mathcal{N}$	
Variational LadderVAE[16]	H	H	$\mathcal{N}$	
PixelVAE [14]	H	H+A	$\mathcal{N}$	
Semi-supervised VAE [17]	H		$\mathcal{N}$+$\mathcal{C}$	✓
VLAE [19]		A	$\mathcal{N}$/L	

(b) Factorizations used by different models.

Figure 5: Figure (a) shows an example VAE with hierarchical encoding distribution and PixelCNN decoding distribution. Figure (b) gives an overview of factorizations used by different models. We indicate the structure of the encoding (ENC) and decoding (DEC) distribution as follows: (H) hierarchical, (A) autoregressive, (default) fully connected or convolutional feed-feed forward neural network). We indicate the prior distribution as follows: ($\mathcal{N}$) multivariate standard Normal, ($\mathcal{C}$) categorical, (M) mixture distribution, (G) graphical model, (L) learned prior. The last column (Y) indicates whether supervision is used.

with that produced by a second encoder $E''_{\phi''}$ and fed to the decoder D_θ. $E''_{\phi''}$ and D_θ are now jointly trained for reconstruction (while keeping ϕ' fixed) and the output of $E''_{\phi''}$ is regularized as in (18) to ensure that $z'' = E''_{\phi''}$ and $z' = E'_{\phi'}$ are disentangled. While the model from [37] does not allow fader-like control of attributes, it provides a representation that facilitates swapping and interpolation of attributes, and can be use for retrieval. Note that in contrast to all previously discussed methods, both of these techniques do not provide a mechanism for unconditional generation.

DC-IGN. Kulkarni et al. [11] assume that the training data is generated by an interpretable, compact graphics code and aim to recover this code from the data using a VAE. Specifically, they consider data sets of rendered object images for which the underlying graphics code consists of extrinsic latent variables—object rotation and light source position—and intrinsic latent variables, modeling, e.g., object identity and shape. Assuming supervision in terms of which latent factors are active (relative to some reference value), a representation disentangling intrinsic and the different extrinsic latent variables is learned by optimizing $\mathcal{L}_{\text{VAE}}$ on different types of mini-batches (which can be seen as implicit regularization): Mini-batches containing images for which all but one of the extrinsic factors are fixed, and mini-batches containing images with fixed extrinsic factors, but varying intrinsic factors. During the forward pass, the latent variables predicted by the encoder corresponding to fixed factors are replaced with the mini-batch average to force the decoder to explain all the variance in the mini-batch through the varying latent variables. In the backward step, gradients are passed through the latent space ignoring the averaging operation. This procedure allows to learn a disentangled representation for rendered 3D faces and chairs that allow to control extrinsic factors similarly as in a rendering engine. The models generalize to unseen object identities.

4 Factorizing the encoding and decoding distributions

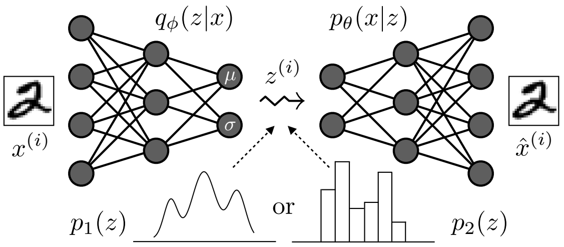
Besides regularization, another popular way to impose a meta-prior is factorizing the encoding and/or decoding distribution in a certain way (see Figure 5 for an overview). This translates directly or indirectly into a particular choice of the model class/network architecture underlying these distributions. Concrete examples are hierarchical architectures and architectures with constrained receptive field. This can be seen as hard constraints on the learning problem, rather than regularization as discussed in the previous section. While this is not often done in the literature, one could obviously combine a specific structured model architecture with some regularizer, for example to learn a disentangled hierarchical representation. Choosing a certain model class/architecture is not only interesting from a representation point of view, but also from a generative modeling perspective. Indeed, certain model classes/architectures allow to better optimize $\mathcal{L}_{\text{VAE}}$ ultimately leading to a better generative model.

Semi-supervised VAE. Kingma et al. [17] harness the VAE framework for semi-supervised learning. Specifically, in the "M2 model", the latent code is divided into two parts z and y where y is (typically discrete) label information observed for a subset of the training data. More specifically, the inference model takes the form $q_\phi(z, y|x) = q_\phi(z|y, x)q_\phi(y|x)$, i.e., there is a hierarchy between y and z. During training, for samples $x^{(i)}$ for which a label $y^{(i)}$ is a available, the inference model is conditioned on y (i.e., $q_\phi(z|y, x)$) and $\mathcal{L}_{\text{VAE}}$ is adapted accordingly, and for samples without label, the label is inferred from $q_\phi(z, y|x)$. This model hence effectively disentangles the latent code into two parts y and z and allows for semi-supervised classification and controlled generation by holding one of the factors fixed and generating the other one. This model can optionally be combined with an additional model learned in unsupervised fashion to obtain an additional level of hierarchy (termed "M1 + M2 model" in [17]).

VLAE. Analyzing the VAE framework through the lens of Bits-Back coding [53, 54], Chen et al. [19] identify the so-called information preference property: The second term in $\mathcal{L}_{\text{VAE}}$ (1) encourages the latent code $z \sim q_\phi(z|x)$ to only store the information that cannot be modeled locally (i.e., unconditionally without using the latent code) by the decoding distribution $p_\theta(x|z)$. As a consequence, when the decoding distribution is a powerful autoregressive model such as conditional PixelRNN [55] or PixelCNN [50] the latent code will not be used to encode any information and $q_\phi(z|x)$ will perfectly match the prior $p(z)$, as previously observed by many authors. While this not necessarily an issue in the context of generative modeling (where the goal is to maximize testing $\log$-likelihood), it is problematic from a representation learning point of view as one wants the latent code $z \sim q_\phi(z|x)$ to store meaningful information. To overcome this issue, Chen et al. [19] propose to adapt the structure of the decoding distribution $p_\theta(x|z)$ such that it cannot model the information one would like z to store, and term the resulting model variational lossy autoencoder (VLAE). For example, to encourage z to capture global high-level information, while letting $p_\theta(x|z)$ model local information such as texture, one can use an autoregressive decoding distribution with a limited local receptive field $p_\theta(x|z) = \prod_j p_\theta(x_j|z, x_{W(j)})$, where $W(j)$ is a window centered in pixel j, that cannot model long-range spatial dependencies. Besides the implications of the information preference property for representation learning, Chen et al. [19] also explore the orthogonal direction of using a learned prior based on autoregressive flow [56] to improve generative modeling capabilities of VLAE.

PixelVAE. PixelVAEs [14] use a VAE with feed-forward convolutional encoder and decoder, combining the decoder with a (shallow) conditional PixelCNN [50] to predict the output probabilities. Furthermore, they employ a hierarchical encoder and decoder structure with multiple levels of latent variables. In more detail, the encoding and decoding distributions are factorized as $q_\phi(z_1, \dots, z_L|x) = q_\phi(z_1|x) \dots q_\phi(z_L|x)$ and $p_\theta(x, z_1, \dots, z_L) = p_\theta(x|z_1)p_\theta(z_1|z_2) \dots p_\theta(z_{L-1}|z_L)p(z_L)$. Here, $z_1, \dots, z_L$ are groups of latent variables (rather than individual entries of z), the $q_\phi(z_j|x)$ are parametric distributions (typically Gaussian with diagonal covariance matrix) whose parameters are predicted from different layers of the same CNN (with layer index increasing in j), $p_\theta(x|z_1)$ is a conditional PixelCNN, and the factors in $p_\theta(z_1|z_2) \dots p_\theta(z_{L-1}|z_L)$ are realized by a feed-forward convolutional networks. From a representation learning perspective, this approach leads to the extraction of high- and low-level features on one hand, allowing for controlled generation of local and global structure, and on the other hand results in better clustering of the codes according to classes in the case of multi-class data. From a generative modeling perspective, this approach obtains testing likelihood competitive with or better than computationally more complex (purely autoregressive) PixelCNN and PixelRNN models. Only $L = 2$ stochastic layers are explored experimentally.

LadderVAE. In contrast to PixelVAEs, Ladder VAEs (LVAEs) [15] perform top-down inference, i.e., the encoding distribution is factorized as $q_\phi(z|x) = q_\phi(z_L|x) \prod_{j=1}^{L-1} q_\phi(z_j|z_{j+1})$, while using the same factorization for $p_\theta(x|z)$ as PixelVAE (although employing a simple factorized Gaussian distribution for $p_\theta(x|z_1)$ instead of a PixelCNN). The $q_\phi(z_j|z_{j+1})$ are parametrized Gaussian distributions whose parameters are inferred top-down using a precision-weighted combination of (i) bottom-up predictions from different layers of the same feed-forward encoder CNN (similarly as in PixelVAE) with (ii) top-down predictions obtained by sampling from the hierarchical distribution $p_\theta(z) = p_\theta(z_1|z_2) \dots p_\theta(z_{L-1}|z_L)p(z_L)$ (see [15, Figure 1b] for the corresponding graphical model representation). When trained with a suitable warm-up procedure, LVAEs are capable of effectively



(a) VAE with a multimodal continuous or discrete prior.

Work	Enc	Dec	$p(z)$	Y
SVAE [24]			G/M	
VQ-VAE [20]		(A)	$\mathcal{C}$/L	
[21]			G	✓
JointVAE [23]			$\mathcal{C}$+$\mathcal{N}$	

(b) Priors employed by different models.

Figure 6: Figure (a) shows an example of a VAE with with a multimodal continuous or discrete prior (each prior gives rise to a different model). Figure (b) gives an overview of the priors employed by different models. We indicate the structure of the encoding (ENC) and decoding (DEC) distribution as follows: (H) hierarchical, (A) autoregressive, (default) fully connected or convolutional feed-feed forward neural network. We indicate the prior distribution as follows: ($\mathcal{N}$) multivariate standard Normal, ($\mathcal{C}$) categorical, (M) mixture distribution, (G) graphical model, (L) learned prior. The last column (Y) indicates whether supervision is used: (✓) indicates that labels are required.

learning deep hierarchical latent representations, as opposed to hierarchical VAEs with bottom-up inference models which usually fail to learn meaningful representations with more than two levels (see [15, Section 3.2]).

Variational Ladder AutoEncoders. Yet another approach is taken by Variational Ladder autoencoders (VLaAEs) [16]: While no explicit hierarchical factorization of $p(z)$ in terms of the z_j is assumed, $p_\theta(z_1|z_2, \dots z_L)$ is implemented as a feed-forward neural network, implicitly defining a top-down hierarchy among the z_j by taking the z_j as inputs on different layers, with the layer index proportional to j. $p_\theta(x|z_1)$ is set to a fixed variance factored Gaussian whose mean vector is predicted from z_1. For the encoding distribution $q_\phi(z|x)$ the same factorization and a similar implementation as that of PixelVAE is used. Implicitly encoding a hierarchy into $p_\theta(z_1|z_2, \dots z_L)$ rather than explicitly as by PixelVAE and LVAE avoids the difficulties described by [15] involved with training hierarchical models with more than two levels of latent variables. Furthermore, Zhao et al. [16] demonstrate that this approach leads to a disentangled hierarchical representation, for instance separating stroke width, digit width and tilt, and digit class, when applied to MNIST.

Finally, Bachman [57] and Kingma et al. [56] explore hierarchical factorizations/architectures mainly to improve generative modeling performance (in terms of testing $\log$-likelihood), rather than exploring it from a representation learning perspective.

5 Structured prior distribution

Instead of choosing the encoding distribution, one can also encourage certain meta-priors by directly choosing the prior distribution $p(z)$ of the generative model. For example, relying on a prior involving discrete and continuous random variables encourages them to model different types of factors, such as the digits and the writing style, respectively, in the MNIST data set, which can be seen as a form of clustering. This is arguably the most explicit way to shape a representation, as the prior directly acts on its distribution.

5.1 Graphical model prior

SVAE. One of the first attempts to learn latent variable models with structured prior distributions using the VAE framework is [24]. Concretely, the latent distribution $p(z)$ with general graphical model structure can capture discrete mixture models such as Gaussian mixture models, linear dynamical systems, and switching linear dynamical systems, among others. Unlike many other VAE-based works, Johnson et al. [24] rely on a fully Bayesian framework including hyperpriors for the likelihood/decoding distribution and the structured latent distribution. While such a structured $p(z)$ allows for efficient inference (e.g., using message passing algorithms) when the likelihood is an exponential family distribution, it becomes intractable when the decoding distribution is parametrized through a neural network as commonly done in the VAE framework, the reason for which the latter includes an approximate posterior/encoding distribution. To combine the tractability of conjugate graphical model inference with the flexibility of VAEs, Johnson et al. [24] employ inference models that output conjugate graphical model potentials [58] instead of the parameters of the approximate posterior distribution. In particular, these potentials are chosen such that they have a form conjugate to the exponential family, hence allowing for efficient inference when combined with the structured $p(z)$. The resulting algorithm is termed structured VAE (SVAE). Experiments show that SVAE with a Gaussian mixture prior learns a generative model whose latent mixture components reflect clusters in the data, and SVAE with a switching linear dynamical system prior learns a representation that reflects behavior state transitions in motion recordings of mouses.

Narayanaswamy et al. [21] consider latent distributions with graphical model structure similar to [24], but they also incorporate partial supervision for some of the latent variables as [17]. However, unlike Kingma et al. [17] which assumes a posterior of the form $q_\phi(z, y|x) = q_\phi(z|y, x)q_\phi(y|x)$, they do not assume a specific factorization of the partially observed latent variables y and the unobserved ones z (neither for $q_\phi(z, y|x)$ nor for the marginals $q_\phi(z|x)$ and $q_\phi(y|x)$), and no particular distributional form of $q_\phi(z|x)$ and $q_\phi(y|x)$. To perform inference for $q_\phi(z, y|x)$ with arbitrary dependence structure, Narayanaswamy et al. [21] derive a new Monte Carlo estimator. The proposed approach is able to disentangle digit index and writing style on MNIST with partial supervision of the digit index (similar to [17]). Furthermore, this approach can disentangle identity and lighting direction of face images with partial supervision assuming the product of categorical and continuous distribution, respectively, for the prior (using the the Gumbel-Softmax estimator [59, 60] to model the categorical part in the approximate posterior).

5.2 Discrete latent variables

JointVAE. JointVAE [23] equips the β-VAE framework with heterogeneous latent variable distributions by concatenating continuous latent variables z with discrete ones c for improved disentanglement of different types of latent factors. The corresponding approximate posterior is factorized as $q_\phi(c|x)q_\phi(z|x)$ and the Gumbel-Softmax estimator [59, 60] is used to obtain a differentiable relaxation of the categorical distribution $q_\phi(c|x)$. The regularization strength λ_1 in the (a constrained variant of) β-VAE objective (6) is gradually increased during training, possibly assigning different weights to the regularization term corresponding to the discrete and continuous random variables (the regularization term in (6) decomposes as $D_{\mathrm{KL}}(q_\phi(z|x)q_\phi(c|x)\|p(z)p(c)) = D_{\mathrm{KL}}(q_\phi(z|x)\|p(z)) + D_{\mathrm{KL}}(q_\phi(c|x)\|p(c))$). Numerical results (based on visual inspection) show that the discrete latent variables naturally model discrete factors of variation such as digit class in MNIST or garment type in Fashion-MNIST and hence disentangle such factors better than models with continuous latent variables only.

VQ-VAE. van den Oord et al. [20] realize a VAE with discrete latent space structure using vector quantization, termed VQ-VAE. Each latent variable z_j is taken to be a categorical random variable with K categories, and the approximate posterior $q_\phi(z_j|x)$ is assumed deterministic. Each category is associated with an embedding vector $e_k \in \mathbb{R}^D$. The embedding operation induces an additional latent space dimension of size D. For example, if the latent representation z is an $M \times M \times 1$ feature map, the embedded latent representation $\tilde{z}$ is a $M \times M \times D$ feature map. The distribution $q_\phi(\tilde{z}_j|x)$ is implemented using a deterministic encoder network $E_\phi(x)$ with D-dimensional output, quantized w.r.t. the embedding vectors $\{e_k\}_{k=1}^K$. In summary, we have

$$q_\phi(\tilde{z}_j = e_k|x) = \begin{cases} 1 & \text{if } k = \arg\min_\ell \|E_\phi(x) - e_\ell\|, \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

The embeddings e_k can be learned individually for each latent variable z_j, or shared for the entire latent space. Assuming a uniform prior $p(z)$, the second term in $\mathcal{L}_{\mathrm{VAE}}$ (1) evaluates to $\log K$ as a consequence of $q_\phi(z|x)$ being deterministic and can be discarded during optimization. To backpropagate gradients through the non-differentiable operation (19) a straight-through type estimator [61] is used. The embedding vectors e_k, which do not receive gradients as a consequence of using a straight-through estimator, are updated as the mean of the encoded points $E_\phi(x^{(i)})$ assigned to the corresponding category k as in (mini-batch) k-means.

VQ-VAE is shown to be competitive with VAEs with continuous latent variables in terms of testing likelihood. Furthermore, when trained on speech data, VQ-VAE learns a rudimentary phoneme-level language model in a completely unsupervised fashion, which can be used for controlled speech generation and phoneme classification.

Many other works explore learning (variational) autoencoders with (vector-)quantized latent representation with a focus on generative modeling [62, 63, 59, 60] and compression [64], rather than representation learning.

6 Other approaches

Early approaches. Early approaches to learn abstract representations using autoencoders include stacking single-layer autoencoders [65] to build deep architectures and imposing a sparsity prior to the latent variables [66]. Another way to achieve abstraction is to require the representation to be robust to noise. Such a representation can be learned using denoising autoencoders [67], i.e., autoencoders trained to reconstruct clean data points from a noisy version. For a broader overview over early approaches we refer to [1, Section 7].

Sequential data. There is a considerable number of recent works leveraging (variational) autoencoders and the techniques similar to those outlined in Sections 3–5 to learn representations of sequences. Yingzhen and Mandt [68] partition the latent code of a VAE into subsets of time varying and time invariant variables (resulting in a particular factorization of the approximate posterior) to learn a representation disentangling content and pose/identity in video/audio sequences. Hsieh et al. [69] use a similar partition of the latent code, but additionally allow the model to decompose the input into different parts, e.g., modelling different moving objects in a video sequence. Somewhat related, Villegas et al. [70], Denton and Birodkar [71], Fraccaro et al. [72] propose autoencoder models for video sequence prediction with separate encoders disentangling the latent code into pose and content. Hsu et al. [73] develop a hierarchical VAE model to learn interpretable representations of speech recordings. Fortuin et al. [74] combine a variation of VQ-VAE with self-organizing maps to learn interpretable discrete representations of sequences. Further, VAEs for sequences are also of great interest in the context of natural language processing, in particular with autoregressive encoders/decoders and discrete latent representations, see, e.g., [75–77] and references therein.

Using a discriminator in pixel space. An alternative to training a pair of probabilistic encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$ to minimize a reconstruction loss is to learn ϕ, θ by matching the joint distributions $p_\theta(x|z)p(z)$ and $q_\phi(z|x)\hat{p}(x)$. To achieve this, adversarially learned inference (ALI) [78] and bidirectional GAN (BiGAN) [79] leverage the GAN framework, learning $p_\theta(x|z)$, $q_\phi(z|x)$ jointly with a discriminator to distinguish between samples drawn from the two joint distributions. While this approach yields powerful generative models with latent representations useful for downstream tasks, the reconstructions are less faithful than for autoencoder-based models. Li et al. [80] point out a non-identifiability issue inherent with the distribution matching problem underlying ALI/BiGAN, and propose to penalize the entropy of the reconstruction conditionally on the code.

Chen et al. [81] augment a standard GAN framework [34] with a mutual information term between the generator output and a subset of latent variables, which proves effective in learning disentangled representations. Other works regularize the output of (variational) autoencoders with a GAN loss. Specifically, Larsen et al. [82], Rosca et al. [83] combine VAE with standard GAN [34], and Tschannen et al. [84] equip AAE/WAE with a Wasserstein GAN loss [85]. While Larsen et al. [82] investigate the representation learned by their model, the focus of these works is on improving the sample quality of VAE and AAE/WAE. Mathieu et al. [86] rely on a similar setup as [82], but use labels to learn disentangled representations.

Cross-domain disentanglement. Image-to-image translation methods [87, 88] (translating, e.g., semantic label maps into images) can be implemented by training encoder-decoder architectures to translate between two domains (i.e., in both directions) while enforcing the translated data to match the respective domain distribution. While this task as such does not a priori encourage learning of meaningful representation, adding appropriate pressure does: Sharing parts of the latent representation between the translation networks [27–29] and/or combining domain specific and shared translation networks [89] leads to disentangled representations.

7 Rate-distortion tradeoff and usefulness of representation

In this paper we provided an overview of existing work on autoencoder-based representation learning approaches. One common pattern is that methods targeting rather abstract meta-priors such as disentanglement (e.g., β-VAE [2]) were only applied to synthetic data sets and very structured real data sets at low resolution. In contrast, fully supervised methods, such as FaderNetworks [12], provide representations which capture subtle properties of the data, can be scaled to high-resolution data, and allow fine-grained control of the reconstructions by manipulating the representation. As such, there is a rather large disconnect between methods which have some knowledge of the downstream task and the methods which invent a proxy task based on a meta-prior. In this section, we consider this aspect through the lens of rate-distortion tradeoffs based on appropriately defined notions of rate and distortion. Figure 7 illustrates our arguments.

Rate-distortion tradeoff for unsupervised learning. It can be shown that models based purely on optimizing the marginal likelihood might be completely useless for representation learning. We will closely follow the elegant exposition from Alemi et al. [30]. Consider the quantities

$$\begin{aligned}
H &= -\int p(x)\log p(x)\,\mathrm{d}x && = \mathbb{E}_{p(x)}[-\log p(x)]\\
D &= -\iint p(x)q_\phi(z|x)\log p_\theta(x|z)\,\mathrm{d}x\,\mathrm{d}z && = \mathbb{E}_{p(x)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)]]\\
R &= \iint p(x)q_\phi(z|x)\log\frac{q_\phi(z|x)}{p(z)}\,\mathrm{d}x\,\mathrm{d}z && = \mathbb{E}_{p(x)}[D_{\mathrm{KL}}(q_\theta(z|x)\|p(z))]
\end{aligned}$$

where H corresponds to the *entropy* of the underlying data source, D the *distortion* (i.e., the reconstruction negative $\log$-likelihood), and R the *rate*, namely the average relative KL divergence between the encoding distribution and the $p(z)$. Note that the ELBO objective is now simply ELBO $= -\mathcal{L}_{\text{VAE}} = -(D+R)$ (or $-(D+\beta R)$ for β-VAE). Alemi et al. [30] show that the following inequality holds:

$$H - D \leq R.$$

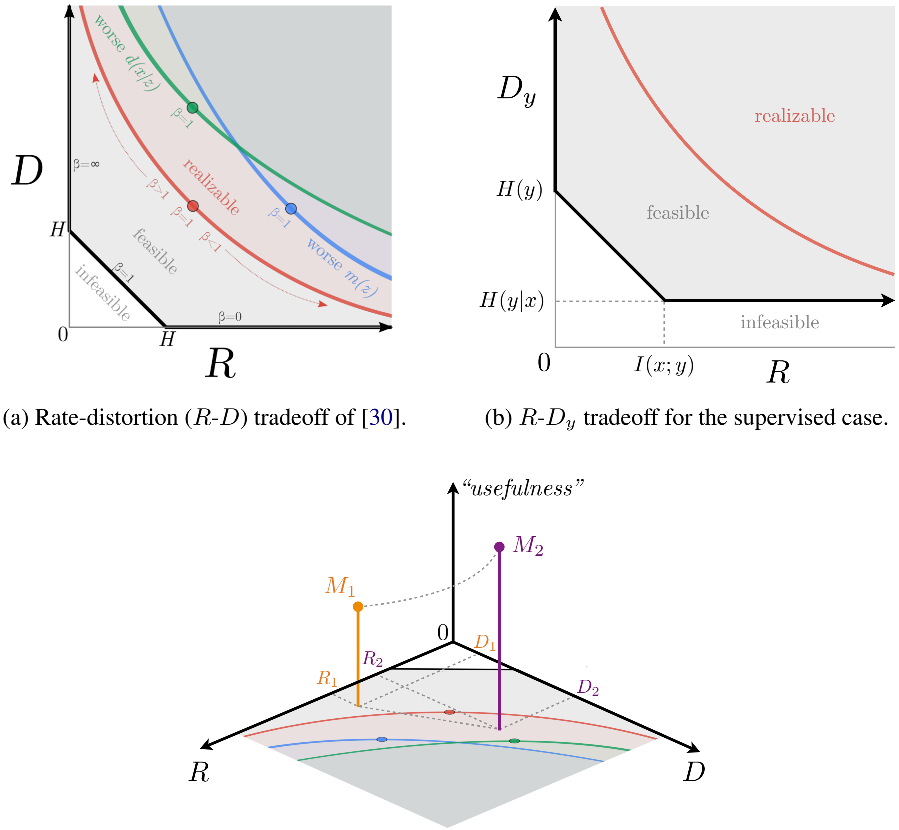
Figure 7 shows the resulting rate-distortion curve from Alemi et al. [30] in the limit of arbitrary powerful encoders and decoders. The horizontal line $(R, 0)$ corresponds to the setting where one is able to encode and decode the data with no distortion at a rate of H. The vertical line $(0, D)$ corresponds to the zero-rate setting and by choosing a sufficiently powerful decoder one can reach the distortion of H. A critical issue is that any point on the line $D = H - R$ achieves the same ELBO. As a result, models based purely on optimizing the marginal likelihood might be completely useless for representation learning [30, 90] as there is no incentive to choose a point with a high rate (corresponding to an informative code). This effect is prominent in many models employing powerful decoders which function close to the zero-rate regime (see Section 4 for details). As a solution, Alemi et al. [30] suggest to optimize the same model under a constraint on the desired rate σ, namely to solve $\min_{\phi,\theta} D + |\sigma - R|$. However, is this really enough to learn representations useful for a specific downstream task?

The rate-distortion-usefulness tradeoff. Here we argue that even if one is able to reach any desired rate-distortion tradeoff point, in particular targeting a representation with specific rate R, the learned representation might still be useless for a specific downstream task. This stems from the fact that

(i) it is unclear which part of the total information (entropy) is stored in z and which part is stored in the decoder, and

(ii) even if the information relevant for the downstream task is stored in z, there is no guarantee that it is stored in a form that can be exploited by the model used to solve the downstream task.

For example, regarding (i), if the downstream task is an image classification task, the representation should store the object class or the most prominent object features. On the other hand, if the downstream task is to recognize relative ordering of objects, the locations have to be encoded instead. Concerning (ii), if we use a linear model on top of the representation as often done in practice, the representation needs to have structure amenable to linear prediction.



(a) Rate-distortion (R-D) tradeoff of [30].

(b) R-D_y tradeoff for the supervised case.

(c) Rate-distortion-usefulness tradeoff.

Figure 7: Figure (a) shows the Rate-distortion (R-D) tradeoff from [30], where D corresponds to the reconstruction term in the (β-)VAE objective, and the rate to the KL term. Figure (b) shows a similar tradeoff for the supervised case considered in [10, 9]. The ELBO $-\mathcal{L}_{\text{VAE}} = -(R + D)$ does not reflect the usefulness of the learned representation for an unknown downstream task (see text), as illustrated in Figure (c).

We argue that there is no *natural* way to incorporate this desiderata directly into the classic R-D tradeoff embodied by the ELBO. Indeed, the R-D tradeoff per se does not account for *what* information is stored in the representation and *in what form*, but only for *how much*.

Therefore, we suggest a third dimension, namely *"usefulness"* of the representation, which is orthogonal to the R-D plane as shown in Figure 7. Consider two models M_1 and M_2 whose rates satisfy $R_1 > R_2$ and $D_1 < D_2$ and which we want to use for the (a priori unknown) downstream task y (say image classification). It can be seen that M_2 is more useful (as measured, for example, in terms of classification accuracy) for y even though it has a smaller rate and and a larger distortion than M_2. This can occur, for example, if the representation of M_1 stores the object locations, but models the objects themselves with the decoder, whereas M_1 produces blurry reconstructions, but learns a representation that is more informative about object classes.

As discussed in Sections 3, 4, and 5, regularizers and architecture design choices can be used to determine what information is captured by the representation and the decoder, and how it is modeled. Therefore, the regularizers and architecture not only allow us to navigate the R-D plane but simultaneously also the "usefulness" dimension of our representation. As usefulness is always tied to (i) a task (in the previous example, if we consider localization instead of classification, M_1 would be more useful than M_2) and (ii) a model to solve the downstream task, this implies that one cannot *guarantee* usefulness of a representation for a task unless it is known in advance. Further, the better the task is known the easier it is to come up with suitable regularizers and network architectures, the extreme case being the fully supervised one. On the other hand, if there is little information one can rely on a generic meta-prior that might be useful for many different tasks, but will likely not lead to a very good representation for all the tasks (recall that the label-based FaderNetwork [12] scales to higher-resolution data sets than β-VAE [2] which is based on a weak disentanglement meta-prior). How well we can navigate the "usefulness" dimension in Figure 7 (c) is thus strongly tied to the amount of prior information available.

A rate-distortion tradeoff for supervised learning. For arbitrary downstream tasks it is clear that it is hard to formalize the "usefulness" dimension in Figure 7. However, if we consider a subset of possible downstream tasks, then it may be possible to come up with a formalization. In particular, for the case where the downstream task is to reconstruct (predict) some auxiliary variable y, we formulate an R-D tradeoff similar to the one of Alemi et al. [30] for a fully supervised scenario involving labels, and show that in this case, the R-D tradeoff naturally reflects the usefulness for the task at hand. Specifically, we rely on the variational formulation of the information bottleneck principle proposed by [10, 9]. Using the terminology of [10], the goal in supervised representation learning is to learn a *minimal* (in terms of code length) representation z of the data x that is *sufficient* for a task y (in the sense that it contains enough information to predict y). This can be formulated using the *information bottleneck (IB) objective* [45] $\max_z I(y; z) - \beta I(z; x)$, where $\beta > 0$. By introducing parametrized distributions $p_\theta(y|z)$, $q_\phi(z|x)$ as in the derivation of VAEs (see Section 2) and by defining distortion as

$$D_y = -\iint p(x, y) q_\phi(z|x) \log p_\theta(y|z)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z = \mathbb{E}_{p(x,y)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(y|z)]],$$

where $p(x, y)$ is the (true) joint distribution of x and y and $p(z)$ is a fixed prior, one obtains a variational approximation of the IB objective as $-(D_y + \beta R)$ (see [10, 9] for details).

Figure 7 (b) illustrates the R-D_y tradeoff. The best we can hope for is that z stores all information about y contained in x, i.e., $R = I(x; y)$. In the limit of arbitrarily complex $p_\theta(x|z)$ such a z yields the minimum achievable distortion, which corresponds to the conditional entropy of y given x, $H(y|x)$. As the rate decreases below $I(x; y)$ the distortion inevitably increases. When $R = 0$ the code does not store any information and we have $p_\theta(y|z) = p_\theta(y)$, and hence for arbitrarily complex $p_\theta(x|z)$, $D_y = \mathbb{E}_{p(x,y)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(y|z)]] = \mathbb{E}_{p(y)}[-\log p_\theta(y)]] = H(y)$. As in the rate-distortion tradeoff for VAEs, all these extreme points are only achievable in the limit of infinite capacity encoders $p_\theta(x|z)$ and decoders $q_\phi(z|x)$. In practice, only models with a larger optimal IB objective $-(D_y + \beta R)$ are realizable.

In the supervised case considered here, the distortion corresponds to the $\log$-likelihood of the target y predicted from the learned representation z. Therefore, given a model trained for a specific point in the R-D_y plane, we know the predictive performance in terms of the negative $\log$-likelihood (or, equivalently, the cross-entropy) of that specific model.

Finally, we note that the discussed rate-distortion tradeoffs for the unsupervised and supervised scenario can be unified into a single framework, as proposed by Alemi and Fischer [51]. The resulting formulation recovers models such as semi-supervised VAE besides (β-)VAE, VIB, and Information dropout, but is no longer easily accessible through a two-dimensional rate-distortion plane. Alemi and Fischer [51] further establish connections of their framework to the theory of thermodynamics.

8 Conclusion and Discussion

Learning useful representations with little or no supervision is a key challenge towards applying artificial intelligence to the vast amounts of unlabelled data collected in the world. We provide an in-depth review of recent advances in representation learning with a focus on autoencoder-based models. In this study we consider several properties, *meta-priors*, believed useful for downstream tasks, such as disentanglement and hierarchical organization of features, and discuss the main research directions to enforce such properties. In particular, the approaches considered herein either (i) regularize the (approximate or aggregate) posterior distribution, (ii) factorize the encoding and decoding distribution, or (iii) introduce a structured prior distribution. Given the current landscape, there is a lot of fertile ground in the intersection of these methods, namely, combining regularization-based approaches while introducing a structured prior, possibly using a factorization for the encoding and decoding distributions with some particular structure.

Unsupervised representation learning is an ill-defined problem if the downstream task can be arbitrary. Hence, all current methods use strong inductive biases and modeling assumptions. Implicit or explicit supervision remains a key enabler and, depending on the mechanism for enforcing meta-priors, different degrees of supervision are required. One can observe a clear tradeoff between the degree of supervision and how useful the resulting representation is: On one end of the spectrum are methods targeting abstract meta-priors such as disentanglement (e.g., β-VAE [2]) that were applied mainly to toy-like data sets. On the other end of the spectrum are fully supervised methods (e.g., FaderNetworks [12]) where the learned representations capture subtle aspects of the data, allow for fine-grained control of the reconstructions by manipulating the representation, and are amenable to higher-dimensional data sets. Furthermore, through the lens of rate-distortion we argue that, perhaps unsurprisingly, maximum likelihood optimization alone can't guarantee that the learned representation is useful at all. One way to sidestep this fundamental issue is to consider the ”usefulness” dimension with respect to a given task (or a distribution of tasks) explicitly.

References

[1] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-VAE: Learning basic visual concepts with a constrained variational framework,” in *International Conference on Learning Representations*, 2017.

[3] H. Kim and A. Mnih, “Disentangling by factorising,” in *Proc. of the International Conference on Machine Learning*, 2018, pp. 2649–2658.

[4] T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, “Isolating sources of disentanglement in variational autoencoders,” in *Advances in Neural Information Processing Systems*, 2018.

[5] S. Zhao, J. Song, and S. Ermon, “InfoVAE: Information maximizing variational autoencoders,” *arXiv:1706.02262*, 2017.

[6] A. Kumar, P. Sattigeri, and A. Balakrishnan, “Variational inference of disentangled latent concepts from unlabeled observations,” in *International Conference on Learning Representations*, 2018.

[7] R. Lopez, J. Regier, M. I. Jordan, and N. Yosef, “Information constraints on auto-encoding variational bayes,” in *Advances in Neural Information Processing Systems*, 2018.

[8] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. van de Meent, “Structured disentangled representations,” *arXiv:1804.02086*, 2018.

[9] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” in *International Conference on Learning Representations*, 2016.

[10] A. Achille and S. Soatto, “Information dropout: Learning optimal representations through noisy computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[11] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, “Deep convolutional inverse graphics network,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2539–2547.

[12] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer *et al.*, “Fader networks: Manipulating images by sliding attributes,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5967–5976.

[13] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, “The variational fair autoencoder,” in *International Conference on Learning Representations*, 2016.

[14] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, “PixelVAE: A latent variable model for natural images,” in *International Conference on Learning Representations*, 2017.

[15] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “Ladder variational autoencoders,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3738–3746.

[16] S. Zhao, J. Song, and S. Ermon, “Learning hierarchical features from deep generative models,” in *Proc. of the International Conference on Machine Learning*, 2017, pp. 4091–4099.

[17] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[18] A. Makhzani and B. J. Frey, “PixelGAN autoencoders,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1975–1985.

[19] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, “Variational lossy autoencoder,” in *International Conference on Learning Representations*, 2017.

[20] A. van den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[21] S. Narayanaswamy, T. B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, “Learning disentangled representations with semi-supervised deep generative models,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5925–5935.

[22] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv:1511.05644*, 2015.

[23] E. Dupont, “Learning disentangled joint continuous and discrete representations,” in *Advances in Neural Information Processing Systems*, 2018.

[24] M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta, “Composing graphical models with neural networks for structured representations and fast inference,” in *Advances in Neural Information Processing Systems*, 2016, pp. 2946–2954.

[25] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations*, 2014.

[26] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” in *Proc. of the International Conference on Machine Learning*, 2014, pp. 1278–1286.

[27] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. F. Wang, “Detach and adapt: Learning cross-domain disentangled deep representation,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[28] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, “Diverse image-to-image translation via disentangled representations,” in *Proc. of the European Conference on Computer Vision*, 2018, pp. 35–51.

[29] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio, “Image-to-image translation for cross-domain disentanglement,” in *Advances in Neural Information Processing Systems*, 2018.

[30] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy, “Fixing a broken ELBO,” in *Proc. of the International Conference on Machine Learning*, 2018, pp. 159–168.

[31] C. Doersch, “Tutorial on variational autoencoders,” *arXiv:1606.05908*, 2016.

[32] X. Nguyen, M. J. Wainwright, and M. I. Jordan, “Estimating divergence functionals and the likelihood ratio by convex risk minimization,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847–5861, 2010.

[33] M. Sugiyama, T. Suzuki, and T. Kanamori, “Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation,” *Annals of the Institute of Statistical Mathematics*, vol. 64, no. 5, pp. 1009–1044, 2012.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[35] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, 2012.

[36] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *International Conference on Learning Representations*, 2018.

[37] N. Hadad, L. Wolf, and M. Shahar, "A two-step disentanglement method," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 772–780.

[38] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. of the IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.

[39] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in *Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance*. Ieee, 2009, pp. 296–301.

[40] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3762–3769.

[41] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *International Conference on Learning Representations*, 2018.

[42] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the f-statistic loss," in *Advances in Neural Information Processing Systems*, 2018.

[43] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," *arXiv:1811.12359*, 2018.

[44] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-VAE," *arXiv:1804.03599*, 2018.

[45] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[46] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[47] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM Journal of Research and Development*, vol. 4, no. 1, pp. 66–82, 1960.

[48] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," in *Advances In Neural Information Processing Systems*, 2016, pp. 2378–2386.

[49] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *International Conference on Algorithmic Learning Theory*. Springer, 2005, pp. 63–77.

[50] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[51] A. A. Alemi and I. Fischer, "TherML: Thermodynamics of machine learning," *arXiv:1807.04162*, 2018.

[52] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, 2008, pp. 1177–1184.

[53] G. E. Hinton and D. Van Camp, “Keeping the neural networks simple by minimizing the description length of the weights,” in *Proc. of the Annual Conference on Computational Learning Theory*, 1993, pp. 5–13.

[54] A. Honkela and H. Valpola, “Variational learning and bits-back coding: An information-theoretic view to bayesian learning,” *IEEE Transactions on Neural Networks*, vol. 15, no. 4, pp. 800–810, 2004.

[55] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” in *International Conference on Machine Learning*, 2016, pp. 1747–1756.

[56] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improved variational inference with inverse autoregressive flow,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4743–4751.

[57] P. Bachman, “An architecture for deep, hierarchical generative models,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4826–4834.

[58] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[59] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with Gumbel-softmax,” in *International Conference on Learning Representations*, 2017.

[60] C. J. Maddison, A. Mnih, and Y. W. Teh, “The concrete distribution: A continuous relaxation of discrete random variables,” in *International Conference on Learning Representations*, 2016.

[61] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv:1308.3432*, 2013.

[62] A. Mnih and K. Gregor, “Neural variational inference and learning in belief networks,” in *Proc. of the International Conference on Machine Learning*, 2014, pp. 1791–1799.

[63] A. Mnih and D. J. Rezende, “Variational inference for monte carlo objectives,” in *Proc. of the International Conference on Machine Learning*, 2016, pp. 2188–2196.

[64] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1141–1151.

[65] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Advances In Neural Information Processing Systems*, 2007, pp. 153–160.

[66] M. A. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, “Efficient learning of sparse representations with an energy-based model,” in *Advances in Neural Information Processing Systems*, 2007, pp. 1137–1144.

[67] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proc. of the International Conference on Machine Learning*, 2008, pp. 1096–1103.

[68] L. Yingzhen and S. Mandt, “Disentangled sequential autoencoder,” in *Proc. of the International Conference on Machine Learning*, 2018, pp. 5656–5665.

[69] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, “Learning to decompose and disentangle representations for video prediction,” in *Advances in Neural Information Processing Systems*, 2018.

[70] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, “Decomposing motion and content for natural video sequence prediction,” in *International Conference on Learning Representations*, 2017.

[71] E. L. Denton and V. Birodkar, “Unsupervised learning of disentangled representations from video,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4414–4423.

[72] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther, “A disentangled recognition and nonlinear dynamics model for unsupervised learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3601–3610.

[73] W.-N. Hsu, Y. Zhang, and J. Glass, “Unsupervised learning of disentangled and interpretable representations from sequential data,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1878–1889.

[74] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, “Deep self-organization: Interpretable discrete representation learning on time series,” *arXiv:1806.02199*, 2018.

[75] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, “Generating sentences from a continuous space,” in *Proc. of the SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[76] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward controlled generation of text,” in *Proc. of the International Conference on Machine Learning*, 2017, pp. 1587–1596.

[77] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, “A hierarchical latent variable encoder-decoder model for generating dialogues.” in *AAAI*, 2017, pp. 3295–3301.

[78] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially learned inference,” in *International Conference on Learning Representations*, 2017.

[79] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” in *International Conference on Learning Representations*, 2017.

[80] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, “Alice: Towards understanding adversarial learning for joint distribution matching,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5495–5503.

[81] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[82] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” *arXiv:1512.09300*, 2015.

[83] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, “Variational approaches for auto-encoding generative adversarial networks,” *arXiv:1706.04987*, 2017.

[84] M. Tschannen, E. Agustsson, and M. Lucic, “Deep generative models for distribution-preserving lossy compression,” in *Advances in Neural Information Processing Systems*, 2018.

[85] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proc. of the International Conference on Machine Learning*, vol. 70, 2017, pp. 214–223.

[86] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, “Disentangling factors of variation in deep representation using adversarial training,” in *Advances in Neural Information Processing Systems 29*, 2016, pp. 5040–5048.

[87] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5967–5976.

[88] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proc. of the IEEE International Conference on Computer Vision*, 2017, pp. 2242–2251.

[89] A. Liu, Y.-C. Liu, Y.-Y. Yeh, and Y.-C. F. Wang, “A unified feature disentangler for multi-domain image translation and manipulation,” in *Advances in Neural Information Processing Systems*, 2018.

[90] F. Huszar, “Is maximum likelihood useful for representation learning?” http://www.inference.vc/maximum-likelihood-for-representation-learning-2.

A Estimators for MMD and HSIC

Expanding (3) and estimating μ_{p_x}, μ_{p_y} as means over samples $\{x^{(i)}\}_{i=1}^N, \{y^{(i)}\}_{i=1}^M$, one obtains an unbiased estimator of the MMD as

$$\begin{aligned}\widehat{\text{MMD}}(p_x, p_y) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i}^{N} k\left(x^{(i)}, x^{(j)}\right) + \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j \neq i}^{M} k\left(y^{(i)}, y^{(j)}\right) \\ - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} k\left(x^{(i)}, y^{(j)}\right).\end{aligned} \tag{20}$$

The Hilbert-Schmidt independence criterion (HSIC) is a kernel-based independence criterion with the same underlying principles as the MMD. Given distributions $p_x(x)$ and $p_y(y)$ the goal is to determine whether $p(x,y) = p_x(x)p_y(y)$ and measure the degree of dependence. Intuitively, if the distributions p_x and p_y are parametrized with parameters α and β, i.e. $p_x = p_\alpha$ and $p_y = p_\beta$ minimizing HSIC(p_α, p_β) w.r.t. α and β encourages independence between p_α and p_β. Given samples $\{x^{(i)}\}_{i=1}^N, \{y^{(i)}\}_{i=1}^N$ from two distributions p_x and p_y on $\mathcal{X}$ and $\mathcal{Y}$, and kernels $k\colon \mathcal{X} \to \mathcal{X}$ and $\ell\colon \mathcal{Y} \to \mathcal{Y}$, the HSIC can be estimated as

$$\begin{aligned}\widehat{\text{HSIC}}(p_x, p_y) = \frac{1}{N^2} \sum_{i,j} k\left(x^{(i)}, x^{(j)}\right) \ell\left(y^{(i)}, y^{(j)}\right) + \frac{1}{N^4} \sum_{i,j,k,l}^{N} k\left(x^{(i)}, x^{(j)}\right) \ell\left(y^{(k)}, y^{(l)}\right) \\ - \frac{2}{N^3} \sum_{i,j,k}^{N} k\left(x^{(i)}, x^{(j)}\right) \ell\left(y^{(i)}, y^{(k)}\right).\end{aligned} \tag{21}$$

We refer to Lopez et al. [7, Section 2.2] for a detailed description and generalizations.

B Overview table

Table 3: Summary of the most important models considered in this paper. The objective is given by $\mathcal{L}.(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[R_1(q_\phi(z|x))] + \lambda_2 R_2(q_\phi(z))$, where $q_\phi(z) = \mathbb{E}_{\hat{p}(x)}[q_\phi(z|x)]$ is the aggregate posterior, R_1 and R_2 are regularizers, and $\lambda_1, \lambda_2 > 0$ are the corresponding regularization weights. The detailed description of the regularizers is provided in Section 3. We indicate the structure of the encoding and decoding distribution as follows: (H) hierarchical, (A) autoregressive, (default) fully connected or convolutional feed-feed forward neural network). We indicate the prior distribution as follows: ($\mathcal{N}$) multivariate standard Normal, ($\mathcal{C}$) categorical, (M) mixture distribution, (G) graphical model, (L) learned prior. We indicate whether labels are used as follows: (✓) Labels are required for (semi-)supervised learning, (O) labels can optionally be used for (semi-)supervised learning.

WORK	$\mathcal{L}.$	R_1	R_2	ENC	DEC	$p(z)$	Y
[2]	VAE	$D_{\mathrm{KL}}(q_\phi(z \mid x) \Vert p(z))$				$\mathcal{N}$	
[9]	VAE	$D_{\mathrm{KL}}(q_\phi(z \mid x) \Vert p(z))$					O
[5]	VAE	$D_{\mathrm{KL}}(q_\phi(z \mid x) \Vert p(z))$	$D_{\mathrm{KL}}(q_\phi(z) \Vert p(z))$			$\mathcal{N}$	
[10]	VAE	$D_{\mathrm{KL}}(q_\phi(z \mid x) \Vert p(z))$	$\mathrm{TC}(q_\phi(z))$				O
[18]	VAE	$-I_{q_\phi}(x; z)$			A	$\mathcal{N}/\mathcal{C}$	O
[8]	VAE	$-I_{q_\phi}(x; z)$	$R_{\mathcal{G}}(q_\phi(z)) + \lambda_2' \sum_{G \in \mathcal{G}} R_{\mathcal{G}}(q_\phi(z))$			$\mathcal{N}$	
[3, 4]	VAE		$\mathrm{TC}(q_\phi(z))$			$\mathcal{N}$	
[6]	VAE		$\Vert \mathrm{Cov}_{q_\phi(z)}[z] - I \Vert_{\mathrm{F}}^2$			$\mathcal{N}$	
[7]	VAE		$\mathrm{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2}))$			$\mathcal{N}$	O
[13]	VAE		$\mathrm{MMD}(q_\phi(z \mid s = 0), q_\phi(z \mid s = 1))$			$\mathcal{N}$	✓
[11]	VAE					$\mathcal{N}$	✓
[17]	VAE			H		$\mathcal{N}$+$\mathcal{C}$	✓
[19]	VAE				A	$\mathcal{N}$/L	
[14]	VAE			H	H+A	$\mathcal{N}$	
[15]	VAE			H	H	$\mathcal{N}$	
[16]	VAE			H	H	$\mathcal{N}$	
[24]	VAE					G/M	
[21]	VAE					G	✓
[23]	VAE					$\mathcal{C}$+$\mathcal{N}$	
[20]	VAE				A	$\mathcal{C}$/L	
[12, 37][4]	AE	$-\mathbb{E}_{\hat{p}(x,y)}[\log P_\psi(1 - y \mid E_\phi(x))]$					✓
[22, 36]	AE		$D_{\mathrm{JS}}(E_\phi(z) \Vert p(z))$			$\mathcal{N}/\mathcal{C}$/M	O

[4] Lample et al. [12], Hadad et al. [37] do not enforce a prior on the latent distribution and therefore cannot generate unconditionally.