Matching aggregate posteriors in the variational autoencoder

Surojit Saha, Sarang Joshi, and Ross Whitaker

***Abstract*—The variational autoencoder (VAE) [1] is a well-studied, deep, latent-variable model (DLVM) that efficiently optimizes the variational lower bound of the log marginal data likelihood and has a strong theoretical foundation. However, the VAE's known failure to match the aggregate posterior often results in *pockets/holes* in the latent distribution (i.e., a failure to match the prior) and/or *posterior collapse*, which is associated with a loss of information in the latent space. This paper addresses these shortcomings in VAEs by reformulating the objective function associated with VAEs in order to match the aggregate/marginal posterior distribution to the prior. We use kernel density estimate (KDE) to model the aggregate posterior in high dimensions. The proposed method is named the *aggregate variational autoencoder* (AVAE) and is built on the theoretical framework of the VAE. Empirical evaluation of the proposed method on multiple benchmark data sets demonstrates the effectiveness of the AVAE relative to state-of-the-art (SOTA) methods.**

***Index Terms*—Latent-variable models, Marginal posterior matching, Non-parametric density estimation, Latent space regularization, Disentanglement analysis**

I. Introduction

THE development of deep, generative models is an important topic of research and has very recently gained a great deal of well-deserved attention. LVMs learn a joint distribution distribution, $p_\theta(\mathbf{x}, \mathbf{z})$, that captures the relationship between a set of learned, hidden variables, $\mathbf{z}$, and the observed variables, $\mathbf{x}$, by tuning the model parameters, θ. The complexity of these relationships often demands deep-learning architectures, but use of a multi-layer perceptron with non-linear activation or deep neural networks makes the posteriors, $p_\theta(\mathbf{z} \mid \mathbf{x})$, intractable. In such scenarios, *variational inference* approximates the true posterior by a surrogate distribution. The VAE [1], [2], which is motivated by the Helmholtz machine [3], introduces a recognition model, $q_\phi(\mathbf{z} \mid \mathbf{x})$ (a deep neural network parameterized by ϕ), that relies on an amortized variational inference to estimate the parameters of the posterior distribution, $p_\theta(\mathbf{z} \mid \mathbf{x})$. VAEs jointly optimize the generative and inference parameters, θ and ϕ, respectively, using stochastic gradient descent to maximize a lower bound on the log marginal likelihood (as $p_\theta(\mathbf{x})$ is intractable). Moreover, the reparameterization trick in VAEs helps in the efficient optimization of the objective function. All these properties and the strong theoretical foundation have made VAEs an important and widely studied DLVM. The samples produced by VAEs are not as realistic as generative models such as the GANs [4] and diffusion-based models [5], [6] (e.g. generated images maybe be slightly blurring and lack fine detail). However, DLVMs are useful in their ability to: project unseen data onto meaningful latent representations, find generative/explanatory factors in data, and provide mechanisms for subsequent analyses (e.g. statistical) in the latent space.

The objective function of the VAE often fails (in theory and practice) to accurately match the aggregate posterior, $q_\phi(\mathbf{z})$, to the prior, $p(\mathbf{z})$. This can result in *clusters* or *holes* in the latent space, indicating regions strongly supported under the prior may have low density under marginal posterior [10] (and vice versa). The presence of *holes* increases the mismatch between the learned, $p_\theta(\mathbf{x})$, and real data distribution, $p(\mathbf{x})$, leading to the production of low-quality samples in the generative scenario. Researchers have proposed extensions to VAEs for increasing the strength of the regularization term of the VAE objective function [11], but this can result in *posterior collapse* [12], [13], where the mutual information $I(\mathbf{x}; \mathbf{z})$ between the latent and observed variables is minimized, leading to a loss of information in the latent space and higher reconstruction error [14]. VAEs model the true posterior distribution with a factorized Gaussian distribution in the latent space, which matches only the variance along the latent axes to the prior and ignores the covariance. Though this assumption helps in the efficient optimization of the objective function, it fails to match the target distribution in an aggregate sense and might be too simple to capture complex posteriors [15], [16]. Figure 1 shows the mismatch between the aggregate posterior distribution and the prior for the VAE [1] and β-TCVAE [8] when trained on a real data set, MNIST [9], and close approximation to the ground truth by the AVAE.

In this work, we present an alternative formulation to VAEs, derived from first principles, that matches the aggregate/marginal distribution to the prior in the latent space. In addition to improvement in the quality of the generated samples, matching the aggregate posterior to a prior finds potential application in the meaningful interpretation of the latent generative factors [8], [17], outlier detection [18], and data completion [19], [20]. Unlike other variants of the VAE that strive to matching marginal posterior to the prior [8], [17], the proposed method does not require additional regularization terms or hyper-parameters to the objective function. We use a kernel density estimator (KDE) [21] to model the aggregate posterior in the latent space [18]. This results in better characterization of differences between aggregate, latent-space distributions, e.g. as compared to the factorized

S. Saha, S. Joshi and R. Whitaker are with the Scientific Computing and Imaging Institute, Kahlert School of Computing, The University of Utah, Salt Lake City, USA. (e-mail:surojit@cs.utah.edu; sjoshi@sci.utah.edu; whitaker@cs.utah.edu)

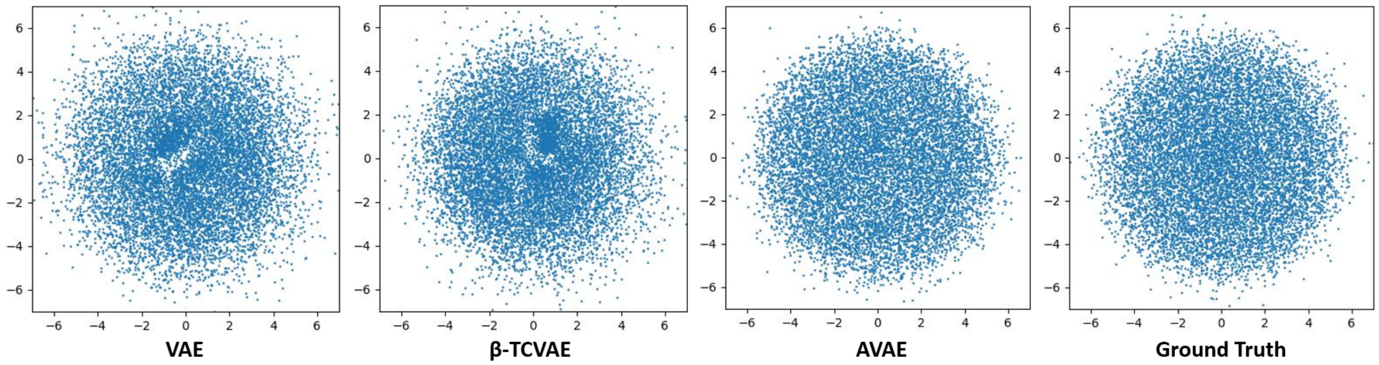

Fig. 1. The metric multidimensional scaling (mMDS) [7] plot in 2D of the latent representations ($\mathcal{Z} \in \mathbb{R}^{16}$) produced by the VAE [1], β-TCVAE [8] and the AVAE (proposed method) using the MNIST data set [9]. The mMDS plot of the samples generated from the target distribution, $\mathcal{Z}^{GT} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{Z}^{GT} \in \mathbb{R}^{16}$, serves as the ground truth for comparison. The mMDS plots of the VAE and β-TCVAE clearly show the mismatch between the learned latent distributions and the ground truth. The mismatch in the latent distributions can be explained with regions of low probability and unwanted aggregation of data points in different parts of the latent space. In comparison, the AVAE closely matches the target distribution, which is corroborated with empirical evaluations.

Gaussian distribution in VAEs. The potential benefits of using KDEs for matching distributions have been presented in [18]. Though KDEs are used in [18] for matching the aggregate posterior distribution to the prior, the objective function is not derived in the general framework of DLVMs, and it is not well suited to very high-dimensional latent spaces, e.g. ≥ 50, which restricts its application for modeling complex data sets, such as the CIFAR10 [22]. The main contributions of this work are summarized as follows:

- A formulation matching the aggregate posterior distribution to the prior in the VAE objective function derived from a maximum a-posteriori formulation.
- An automated method for the estimation of the KDE bandwidth that allows the use of KDEs in high-dimensional latent spaces (dimensions > 100).
- Evaluations showing that proposed method addresses shortcomings in the formulation of the VAE, such as the *posterior collapse* and *pockets/clusters* in the latent distribution.
- An extension of the analysis of disentanglement of latent factors in DLVMs for more general models, such that latent variables need not be aligned with the latent axes.
- Empirical evaluation of the proposed method using different efficacy measures on multiple benchmark data sets, producing results that compare favorably with state-of-the-art, likelihood-based, generative models.

II. Related work

Several extensions to the formulation of the VAE address known limitations, such as alleviating posterior collapse [23], [24], better matching of marginal posteriors [8], [17], and reducing over-regularization [15], [16]. Methods matching marginal posteriors are relevant to our work. These methods either introduced an additional regularization term to the objective function [17] or introduced a hyper-parameter [8] to encourage statistical independence of latent factors. The latent representation built by these methods is primarily used for disentanglement studies (identifying latent generative factors). However, no attempt is made to evaluate the aggregate distribution in the latent space at convergence.

An interesting analysis of the VAE objective is done in [25] that aims at studying the importance of the stochastic inference and generative models used in VAEs. The study makes a strong modeling assumption that the variance of the factorized Gaussian distribution, approximating the true posterior, is the same for all the latent dimensions. The analysis in that work suggests that the latent representation built by an autoencoder with regularized decoder parameters is as good as the representation produced by VAEs. However, the model uses ex-post density estimation in the latent space to function as a generative model. The authors named the model the regularized autoencoder (RAE).

The generative adversarial network (GAN) is another popular generative model that implicitly matches distributions using a discriminator [26], [27]. GANs produce novel, realistic examples, such as images with sharp and distinct features, which are difficult for even humans to identify as generated images [4]. Nevertheless, GANs do not produce a reliable matching form data samples into the latent space [28], and there are significant challenges in optimizing the objective function of a GAN [29]–[31]. GANs are very particular about the architecture of the discriminator, training strategy, and the associated hyper-parameters [32], [33]. The adversarial autoencoder (AAE) [34] is a likelihood-based generative model that matches the aggregate posterior in the latent space of an autoencoder to a prior with the help of a discriminator. The AAE implicitly matches distributions by relying on the interactions of the encoder and discriminator in the latent space to optimize the Jesnsen-Shanon divergence of samples from the empirical and target distributions, unlike explicit density matching done in VAEs.

WAEs [35] is another likelihood-based generative model that explicitly matches the aggregate posterior to a prior in the latent space (unlike VAEs). In the WAE, the Wasserstein distance between the data and generated distribution is minimized by factoring the latent variable $\mathbf{z}$ in its formulation. The regularization term in WAEs is computed using two different strategies. In one approach, a discriminator is used in the latent space as in AAEs, known as the WAE-GAN. In the other approach, the maximum mean discrepancy (MMD) [36] is used to compute the divergence between distributions in the latent space, known as the WAE-MMD. The WAE-MMD is the preferred choice, as discriminators are hard to train. The AAE is considered a special case of WAEs [35] due to the open-endedness in the choice of the cost functions in the objective function of the WAE.

III. Methods

A. Formulation

The goal of a DLVM is to learn the joint distribution of the latent variables, $\mathbf{z}$, and the observed variables, $\mathbf{x}$, such that the resulting (generative) distribution closely approximates the true but unknown data distribution, $p_{data}(\mathbf{x})$. The general formulation for DLVMs is as follows:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \tag{1}$$

, where the $p(\mathbf{z})$ is the prior distribution over the latent space, e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and $p_\theta(\mathbf{x} \mid \mathbf{z})$ is the likelihood distribution. In the DLVM, $p_\theta(\mathbf{x} \mid \mathbf{z})$ learns the mapping from the latent space to observed space using the samples generated by $p(\mathbf{z})$ and model parameters θ. This setup is used to generate new samples not present in the observed data set. Thus, the aim is to determine the correct setting of the parameters, θ, such that the probability of each observed data, $p_\theta(\mathbf{x})$, is maximized. The objective function of the DLVM is defined as follows:

$$\begin{aligned}
&\max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log p_\theta(\mathbf{x}) = \max_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log \int p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\
&= \max_{\theta,q} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log \int \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z})} \right) q(\mathbf{z}) d\mathbf{z} \quad \text{Expectation under the proposal distribution, } q(\mathbf{z}) \qquad (2) \\
&= \max_{\theta,q} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \log \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})} \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z})} \right) \\
&\text{by Jensen's inequality, we get} \\
&\geq \max_{\theta,q} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})} \log \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z})} \right) \\
&= \max_{\theta,q} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})} \left\{ \log \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \right) - \log \left(\frac{q(\mathbf{z})}{p(\mathbf{z})} \right) \right\} \\
&= \max_{\theta,q} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left\{ \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})} \log \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \right) - \mathrm{KL} \left(q(\mathbf{z}) || p(\mathbf{z}) \right) \right\}.
\end{aligned} \tag{3}$$

The objective function defined in 3 gives a lower bound on the data log likelihood and is known as the variational lower bound or evidence lower bound (ELBO).

Use of $q(\mathbf{z} \mid \mathbf{x})$ as the proposal distribution in equation 2 gives us the objective function of the VAE [1]. The choice of the probability distribution for q is a modeling choice, and for VAEs, it is typically a Gaussian distribution [1] . The VAE uses an inference network (also called a recognition model), $q_\phi(\mathbf{z} \mid \mathbf{x})$, a deep neural network parameterized by ϕ that estimates the parameters of the Gaussian distribution for any input $\mathbf{x}_i$, $\phi : \mathbf{x}_i \rightarrow (\mu_i, \sigma_i)$. The recognition model does amortize inference that alleviates the optimization of the distribution parameters separately for each observed data.

Matching the conditional distribution, $q_\phi(\mathbf{z} \mid \mathbf{x})$, to $p(\mathbf{z})$ in VAEs often fails to match the aggregate posterior in the latent space [10], [14]. The mismatch leads to, among other things, *holes* or *pockets* in the latent distribution that subsequently affects the quality of the generated samples. Increasing the strength of the regularization term in the objective function of VAEs does not help better match the aggregate posterior to the prior [11]. Instead, it results in a scenario known as *posterior collapse* [12], [13], where the conditional distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ matches to the prior $p(\mathbf{z})$ for most of the data samples. Such degenerate solutions produce latent encodings that are no longer meaningful, and the network tends to ignore $\mathbf{z}$ in the reproduced observed data, resulting in poor reconstruction. This phenomenon is related to the identity, $\mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z})\right) = I(\mathbf{x};\mathbf{z}) + \mathrm{KL}\left(q_\phi(\mathbf{z}) || p(\mathbf{z})\right)$, where the $I(\mathbf{x};\mathbf{z})$ is the mutual information between the observed and latent variables. Thus, increasing the strength of the KL term would lead to better aggregate posterior matching but would lower the mutual information between the latent variables and the data. To circumvent these issues with the formulation of the VAE, several variants are proposed [8], [17], [24] that emphasize matching the aggregate posterior to a prior.

Instead of parametric distribution on the conditional probability, as used in VAEs, we propose to represent the aggregate distribution with the kernel density estimates (KDE). The KDE used to approximate the aggregate posterior distribution is defined as:

$$q(\mathbf{z}) = \frac{1}{m} \sum_{i=1}^{m} K \left(\frac{||\mathbf{z} - \mathbf{z}_i^{'}||}{h} \right). \tag{4}$$

Using KDEs, the probability at $\mathbf{z}$ for the proposal distribution is estimated using m KDE samples, $\mathbf{z}_i^{'}$, and the kernel, K, with an associated bandwidth, $h \in \mathbb{R}^{+}$. We use a subset of the training data $\mathcal{X}^{kde} \in \mathcal{X}^{train}$ to produce the KDE samples, $\mathbf{z}_i^{'} = \mathbf{E}_\phi(\mathbf{x}_i^{'})$, where $\mathbf{E}_\phi$ is a deep neural network parameterized by ϕ, known as the *encoder*, and $\mathbf{x}_i^{'} \in \mathcal{X}^{kde}$. We use a deterministic encoder (ignoring the variances along the latent axes), unlike VAEs, as we aim at matching the aggregate distribution, $q_\phi(\mathbf{z})$, to the prior, $p(\mathbf{z})$, instead of matching the conditional posterior distribution, $q_\phi(\mathbf{z} \mid \mathbf{x})$, to the prior, $p(\mathbf{z})$. Through multiple empirical evaluations, we show that using a deterministic encoder in the AVAE does not rob it of expressive power compared to a regular VAE or its variants.

The ELBO objective function using the KDE-based proposal distribution $q_\phi(\mathbf{z})$ is defined as follows:

$$\max_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left\{ \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \log \left(p_\theta(\mathbf{x} \mid \mathbf{z}) \right) - \mathrm{KL} \left(q_\phi(\mathbf{z}) || p(\mathbf{z}) \right) \right\}. \tag{5}$$

Equation 5 gives us the objective function of the AVAE. In comparison to the proposal distribution in VAEs, the KDE-based approximation matches the aggregate posterior, $q_\phi(\mathbf{z})$, to the prior, $p(\mathbf{z})$, without any modifications to the ELBO formulation. Compared to the β-TCVAE [8], the AVAE does not have a mutual information (MI) term in its objective function. The absence of the MI in the AVAE also reduces the number of hyper-parameters.

The random (data) variable $\mathbf{x}$ typically exists high dimensions, and thus the probability of $p_\theta(\mathbf{x} \mid \mathbf{z})$ is valid only for a small region in the latent space, i.e., $p_\theta(\mathbf{x} \mid \mathbf{z})$ is nonzero for small subset of the latent space. We use $\mathbf{z} = \mathbf{E}_\phi(\mathbf{x})$ as an estimate to maximize $\log p_\theta(\mathbf{x} \mid \mathbf{z})$ in equation 5. Moreover, $\mathbf{z} = \mathbf{E}_\phi(\mathbf{x})$ is a high-probability (likely) sample from $q_\phi(\mathbf{z})$, because the encoder parameters capture the statistics of the aggregate posterior distribution, $q_\phi(\mathbf{z})$. Considering this modeling choice, the objective function of the AVAE then becomes:

$$\max_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left\{\log\left(p_\theta(\mathbf{x} \mid \mathbf{E}_\phi(\mathbf{x}))\right) - \mathrm{KL}\left(q_\phi(\mathbf{z})||p(\mathbf{z})\right)\right\}. \tag{6}$$

Here, we present another perspective in support of the single point estimate used to maximize $\log\left(p_\theta(\mathbf{x} \mid \mathbf{z}\right)$ in equation 6. The term $\log\left(p_\theta(\mathbf{x} \mid \mathbf{z}\right)$ in 6 is a function of the random variable $\mathbf{z}$, $f_\theta(\mathbf{z}) = \log\left(p_\theta(\mathbf{x} \mid \mathbf{z}\right)$. By Taylor series expansion of $f_\theta(\mathbf{z})$, we get the following:

$$f_\theta(\mathbf{z}) \approx f_\theta(\mathbf{z}_0) + (\mathbf{z} - \mathbf{z}_0)^T f'_\theta(\mathbf{z}_0) \quad \text{Ignoring the higher order terms} \tag{7}$$

$$\mathbf{z}_0^* = \arg\max_{\mathbf{z}_0} f_\theta(\mathbf{z}), \tag{8}$$

where $\mathbf{z}_0^*$ is the value for which $f_\theta(\mathbf{z})$ is maximized. This results in $f_\theta(\mathbf{z}) = f_\theta(\mathbf{z}_0^*)$ as the derivative at the mode $(\mathbf{z}_0^*)$ is zero. We use an amortized inference of the neural network, $\mathbf{E}_\phi$, for estimating $\mathbf{z}_0^*$ for different data points, $\mathbf{x}$.

We use the multivariate Gaussian distribution or Bernoulli distribution as the conditional likelihood distribution, $p_\theta(\mathbf{x} \mid \mathbf{z})$ in 6, depending on the data set. The parameters of the chosen distribution are estimated using another neural network known as the *decoder*, $\mathbf{D}_\theta$, parameterized by θ. The objective function in 6 is optimized using the SGD that jointly updates the encoder and decoder parameters, ϕ and θ, respectively. The first term in the objective function tries to reproduce the input as closely as possible using the corresponding latent statistics (reconstruction loss), while the KL term (matching the aggregate posterior to the prior) prevents the over-fitting of the model parameters.

The objective function of the AVAE has a similarity to WAEs, which has a reconstruction term and a divergence penalty on the aggregate distribution over latent representations. The divergence measure regulates the trade-off between the reconstruction and latent regularization loss. Similar to WAEs, the AVAE has the flexibility in choosing reconstruction cost terms by considering different distributions for $p_\theta(\mathbf{x} \mid \mathbf{z})$. The divergence penalty in the AVAE is the KL divergence, a particular case of the WAE. Nevertheless, the AVAE has provable statistical properties of the latent space, and the proposed method has empirically demonstrated its merit over the WAE (using the MMD as the divergence measure) under several evaluation metrics discussed in subsequent sections.

B. Aggregate variational autoencoder

1) Training: The objective function of the AVAE defined in 6 has two terms: the reconstruction loss and KL-divergence-matching of the aggregate posterior to the prior. The aggregate posterior, $q_\phi(\mathbf{z})$, in the AVAE is represented using the KDE. A subset of the training data $\mathcal{X}^{kde} \in \mathcal{X}^{train}$, chosen at random, form the KDE samples. Remaining samples $\mathcal{X}^{sgd} = \mathcal{X}^{train} - \mathcal{X}^{kde}$ are used for optimizing the objective function using the SGD that updates the model parameters, ϕ and θ. Members of $\mathcal{X}^{kde}$ used in the KDE estimate and $\mathcal{X}^{sgd}$ used for minibatch stochastic gradient descent are shuffled after every epoch, such that the model generalizes to any random KDE set. This optimization strategy also helps in the amortized inference of the optimal latent encoding for an input that has the minimum reconstruction loss.

The KDE samples $\mathbf{z}'_i = \mathbf{E}_\phi(\mathbf{x}'_i)$, where $\mathbf{x}'_i \in \mathcal{X}^{kde}$, are updated as the model parameters are learned, which essentially updates the aggregate posterior estimate. Thus, the encoder models the statistics of the posterior distribution, and the current state of the encoder represents the posterior distribution at any point in the optimization of the AVAE. For stable optimization of the objective function 6 and computational benefits, update of the KDE samples $(\mathbf{z}'_i = \mathbf{E}_\phi(\mathbf{x}'_i))$ are set to lag for few minibatch updates. For this work, we use the isotropic Gaussian kernel in the KDE, which introduces a bandwidth parameter. There are many heuristics for estimating the kernel bandwidth used in the KDE, and there is no established solution for unknown distributions. Furthermore, the curse of dimensionality makes estimating bandwidths in high dimensions particularly challenging. We present a bandwidth estimation method in section III-B4 for a given latent dimension and a given number of KDE samples that scales to higher dimensions.

In this work, the standard normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, is chosen as the prior distribution. However, the proposed method can be extended to the multivariate Gaussian distribution, parameterized by mean μ and Σ. Several extensions of the VAE [37]–[39] propose automated ways to determine the hyper-parameter β that balances the loss terms in the objective function. In a similar vein, we propose a data-driven technique, described in section III-B2, to determine β that balances the loss terms in the AVAE objective function. An outline of the training of AVAEs is presented in Algorithm 1. We discuss the details of the encoder-decoder architecture, optimizer parameters, and other hyper-parameters related to the training of the AVAE in Appendix A.

2) Estimation of β: The formulation of the VAE objective function does not introduce a hyper-parameter to weigh the loss terms. However, it is a common practice to assign weights to different terms in the objective functions [25], [34], [35] for various reasons, such as stability in optimization and application-specific trade-offs. Likewise, several variants of the VAE [8], [11] use a hyper-parameter, β, to control the contribution of the loss terms in the objective function. It is often challenging to decide the appropriate value of these hyper-parameters for a particular model architecture, data set, and other related settings for optimization. The widely used strategy under these circumstances is to set the hyper-parameter value using cross-validation.

To alleviate these issues, methods proposed in [38], [39], among others, have devised automated strategies to determine β. The method in [38] uses a PI controller that manipulates the value of β as the learning progresses. Assuming the decoder predicts the parameter of the multivariate Gaussian distribution, [39] presents two approaches to learning the Gaussian variance, σ (equivalent to learning β). In the first approach, an additional parameter is trained with the encoder-decoder parameters to learn the trade-off factor, σ. In another

Algorithm 1 : AVAE training. Minibatch stochastic gradient descent of the AVAE objective function 6.

```
Input: Training samples $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, Minibatch size $n_b$, Latent dimension $l$, Number of KDE samples $m$, Number of epochs after which the KDE samples are shuffled $k_e$, Number of minibatch updates the KDE estimate should lag $k_b$
Output: encoder parameters $\phi$ and decoder parameters $\theta$
Estimate the optimal kernel bandwidth $h_{\text{opt}}$ given $(l, m)$
Split $\mathcal{X}$ into training, $\mathcal{X}^{train}$, and validation data, $\mathcal{X}^{val}$
Initialize samples for the KDE, $\mathcal{X}^{kde} \in \mathcal{X}^{train}$, where $\mathcal{X}^{kde} = \{\mathbf{x}'_1, \dots \mathbf{x}'_m\}$
Initialize SGD samples $\mathcal{X}^{sgd} = \mathcal{X}^{train} - \mathcal{X}^{kde}$
Initialize $\phi$ and $\theta$
Initialize KDE samples, $\mathbf{z}_i^{'} = E_\phi(\mathbf{x}_i^{'})$, where $\mathbf{x}_i^{'} \in \mathcal{X}^{kde}$
Initialize $\beta \leftarrow \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \|\mathbf{x}_i^{''} - \hat{\mathbf{x}}_i^{''}\|_2$, where $\mathbf{x}_i^{''} \in \mathcal{X}^{val}$
Initialize epoch index $e \leftarrow 0$
Initialize minibatch index $b \leftarrow 0$
for number of epochs do
    if $e \mod k_e$ then
        Generate a new set for the KDE, $\mathcal{X}^{kde} = \{\mathbf{x}_1^{'}, \dots \mathbf{x}_m^{'}\}$
        Produce latent encoding $\mathbf{z}_i^{'} = E_\phi(\mathbf{x}_i^{'})$ for KDE estimate, $q_\phi$
        Update SGD samples, $\mathcal{X}^{sgd} = \mathcal{X}^{train} - \mathcal{X}^{kde}$
    end if
    for number of minibatch updates do
        Sample a minibatch of size $n_b$ from $\mathcal{X}^{sgd}$, $\mathcal{X}_b^{sgd} = \{\mathbf{x}_1^{''}, \dots \mathbf{x}_{n_b}^{''}\}$
        Encode samples $\mathcal{Z}_b^{sgd} = \{\mathbf{z}_1^{''}, \dots \mathbf{z}_{n_b}^{''}\}$, where $\mathbf{z}_i^{''} = E_\phi(\mathbf{x}_i^{''})$
        Update the encoder parameters, $\phi$ using stochastic gradient descent:
            $$\nabla_\phi \frac{1}{n_b} \sum_{i=1}^{n_b} -\log\left(p_\theta(\mathbf{x}_i^{''} \mid \mathbf{z}_i^{''})\right) + \beta \times \nabla_\phi \frac{1}{n_b} \sum_{i=1}^{n_b} \left\{\log \frac{q_\phi(\mathbf{z}_i^{''})}{p(\mathbf{z}_i^{''})}\right\}$$
        Update the decoder parameters, $\theta$ using stochastic gradient descent:
            $$\nabla_\theta \frac{1}{n_b} \sum_{i=1}^{n_b} -\log\left(p_\theta(\mathbf{x}_i^{''} \mid \mathbf{z}_i^{''})\right)$$
        if $b \mod k_b$ then
            Update KDE samples, $\mathbf{z}_i^{'} = E_\phi(\mathbf{x}_i^{'})$
            using the current state of the encoder, where $\mathbf{x}_i^{'} \in \mathcal{X}^{kde}$
        end if
        $b \leftarrow b + 1$
    end for
    $b \leftarrow 0$
    $e \leftarrow e + 1$
    $\beta \leftarrow \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \|\mathbf{x}_i^{''} - \hat{\mathbf{x}}_i^{''}\|_2$, where $\mathbf{x}_i^{''} \in \mathcal{X}^{val}$
    Shuffle $\mathcal{X}^{sgd}$
end for
```

approach, the maximum likelihood estimate (MLE) determines the variance analytically.

Similar to these approaches, the proposed AVAE optimization sets beta β to weight the gradient of the regularization term relative to the reconstruction loss:

$$\beta \leftarrow \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \|\mathbf{x}_i^{''} - \hat{\mathbf{x}}_i^{''}\|_2, \tag{9}$$

where $\mathbf{x}_i^{''}$ is an example in the validation set, $\mathcal{X}^{val} \in \mathcal{X}$, and $\hat{\mathbf{x}}_i^{''}$ is the corresponding reconstructed sample produced by the decoder. Relative to [38], the proposed approach is simple yet effective, as demonstrated by the empirical evaluations. Moreover, this formulation can be extended to any distribution chosen for the log conditional likelihood, $p(x \mid z)$, rather than being limited to only a Gaussian, as in [39].

3) Properties of the marginal posterior distribution: Considering the standard normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, as the prior distribution, $p(\mathbf{z})$, we analytically derive the expected aggregate posterior distribution of a trained AVAE. For a trained AVAE model, we assume the gradient of the objective function w.r.t to latent encodings, $\mathbf{z}_i^{''}$'s (refer to Algorithm 1), is zero. In our analysis, we consider only the KL divergence term in the objective function. Setting the derivative of the $\text{KL}(q_\phi(\mathbf{z}^{''})||p(\mathbf{z}^{''})$ to 0, we derive the same expression as in equation 5 of [18]. Following the steps in [18], we prove the aggregate posterior distribution of the AVAE is $\mathcal{N}\left(\mathbf{0}, \mathbf{I}(1-h^2)\right)$, in expectation, where h is the KDE bandwidth. The proof is also consistent with the known properties of KDEs generally — KDEs introduce a bias that is characterized by a convolution of the kernel with the underlying distribution.

4) KDE bandwidth estimate: Estimating KDE bandwidth can be challenging, and solutions in the literature are often related to particular applications. Many heuristics are proposed for bandwidth estimation under general circumstances [40]. However, here, we rely on our knowledge that the empirical aggregate distribution in the latent space approaches the target distribution as the system converges. Thus, we set the kernel bandwidth h_{opt} to minimize the KL divergence between the analytical prior distribution and the KDE of a finite set of samples from the prior distribution, as follows:

$$h_{\text{opt}} = \min_h \text{KL}\left(p(\mathbf{z})||q_\phi(\mathbf{z})\right) = \max_h \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} q_\phi(\mathbf{z}), \tag{10}$$

with latent dimension, l, and a number of KDE samples, m. In this optimization problem, we use samples from the $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ such that the probability of the samples is maximized w.r.t the aggregate posterior, $q_\phi(\mathbf{z})$. Table I reports the optimum bandwidth for different scenarios. We use gradient-based optimizers, such as Adam, to learn the single parameter, h in 10. We observe in Table I that for higher latent dimensions with limited KDE samples (e.g., starting at $l = 40$ with $m = 500$), the optimal bandwidth is greater than the standard deviation of the prior distribution, $h_{\text{opt}} > 1.0$. Given the known bias KDE introduces in the AVAE optimization, optimizing the encoder under these conditions would degenerate to samples converging at the origin (posterior collapse).

TABLE I

Optimal bandwidths, h_{opt}, estimated using the objective function defined in 10. Error bars are computed over 3 trials. The estimated bandwidth increases with increasing dimensions (vertical) and decreases with increasing sample size (horizontal).

$l\backslash m$	500	1000	2000	5000	10000
10	0.74 ± 0.00	0.70 ± 0.00	0.67 ± 0.00	0.63 ± 0.00	0.60 ± 0.00
20	0.89 ± 0.00	0.86 ± 0.00	0.84 ± 0.00	0.80 ± 0.00	0.78 ± 0.00
40	> 1.0	> 1.0	0.98 ± 0.00	0.95 ± 0.00	0.93 ± 0.00
50	> 1.0	> 1.0	> 1.0	0.99 ± 0.00	0.98 ± 0.00
70	> 1.0	> 1.0	> 1.0	> 1.0	> 1.0
100	> 1.0	> 1.0	> 1.0	> 1.0	> 1.0

From section III-B3, we known that the AVAE converges to $\mathcal{N}\left(\mathbf{0}, \mathbf{I}(1-h^2)\right)$, where $(1-h^2)$ is bias introduced by the KDE. However, we could not consider $\mathcal{N}\left(\mathbf{0}, \mathbf{I}(1-h^2)\right)$ as the target distribution for optimization of the objective function in equation 10, as h is unknown. We hypothesize that this is one of the reasons for the optimal bandwidth h_{opt} to be greater than the standard deviation of the prior distribution in high dimensions (Table I). Thus, we must factor in the bias, $(1-h^2)$, introduced by the KDE to estimate the bandwidth. To this end, we propose to use a scaled version of the target distribution, $\mathcal{N}\left(\mathbf{0}, \alpha^2\mathbf{I}\right)$, for optimization of the objective function in 10, where the scaling factor α is unknown. We need to estimate the optimum bandwidth for $\mathcal{N}\left(\mathbf{0}, \alpha^2\mathbf{I}\right)$. Given h_{opt} as the optimum bandwidth for $\mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$, the estimated bandwidth for $\mathcal{N}\left(\mathbf{0}, \alpha^2\mathbf{I}\right)$ is αh_{opt} by linear property of the Gaussian distribution. Moreover, we know (from section III-B3) that with αh_{opt} as the KDE bandwidth, the latent distribution of the AVAE would have a bias, $(1-(\alpha h_{\text{opt}})^2)$, at convergence. We use this property to solve for the scaling factor, α, where we set the variance equal to the bias, $\alpha^2 = 1-(\alpha h_{\text{opt}})^2$, to get the scaling that accounts for both the ideal optimal bandwidth and the bias:

$$\alpha^2 = \frac{1}{1+h_{\text{opt}}^2}. \tag{11}$$

This simple but elegant strategy of handling the bias in the KDE addresses the *curse of dimensionality* that affects the estimation of bandwidth in high dimensions. This method takes advantage of the fact that the latent space configuration in high dimensions shrinks, increasing the data density and allowing for a smaller KDE bandwidth. The application of KDEs in higher dimensions (e.g., dimensions ≥ 50) makes the system appropriate for complex data sets (a limitation of previous KDE-based aggregate matching [18]). Notice that because $0 \leq \alpha \leq 1.0$, we avoid mode collapse because the system only degenerates ($h_{\text{opt}} \to \infty$) as the number of samples goes to zero or the dimensionality goes to infinity.

With the *bias scaling factor*, α, we get estimates of the KDE bandwidth reported in Table II, which are the scaled versions of the optimum bandwidth h_{opt} (equation 10) estimated using $\mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ as the target distribution. In the revised estimate, the optimal bandwidth is less than the standard deviation of the prior distribution, $h_{\text{opt}} < 1.0$, for all dimensions in Table II, as expected.

IV. Experiments

A. Experimental setup

Benchmark methods: In comparisons, we consider the conventional VAE [1] and other variations of VAE that modify the original formulation in an attempt to match the aggregate posterior to the prior, such as the FactorVAE [17], and β-TCVAE [8], for comparison. The FactorVAE [17] and β-TCVAE [8] have the total correlation loss in their objective function. However, FactorVAE adds a loss term to the ELBO objective, unlike β-TCVAE, and uses a discriminator to optimize the objective function, which is challenging to train. Among others, β-TCVAE [8] is the closest to the AVAE formulation, as the objective function does not introduce any addition, ad-hoc loss terms.

The RAE [25] robs the stochasticity built into the VAE models and relies on an ex-post density estimate in the latent space of a regularized autoencoder for the model to be used as a generative model. Demonstrating the model efficacy on benchmark data sets motivated the authors to consider this method one of the baseline models. As there is no clear advantage in using a specific regularizer studied in the RAE, we use the L-2 regularizer for simplicity. Other than the variants of the VAE, maximum likelihood-based models such as the AAE [34] and WAE [35] match aggregate posterior in the latent space of a deterministic autoencoder. The AAE [34] implicitly matches aggregate distributions using a discriminator in the latent space. We use the WAE-MMD (with IMQ kernel) in our analysis due to the stability in training. In this work we study VAE [1], β-TCVAE [8], RAE [25], AAE [34], and WAE-MMD [35] as competing methods to the proposed AVAE.

Evaluation metrics: Ideally, the evaluation of a DLVM should include a comparison of the model's data distribution and that of the true data. Of course, this is infeasible because true data distribution is unknown. Many methods use the quality of the samples produced by the models in the observed space as a proxy for the actual distribution. In this work, we use the Fréchet Inception Distance (FID) [41] to quantify the quality of the samples. In addition, we evaluate the data distributions learned by different models using the precision and recall metric [42], where the precision evaluates the quality of the generated samples, and the recall assesses whether the model data distribution captures the variations present in the original but unknown data distribution.

Besides the attributes of the model data distribution, it is also essential to understand the properties of the latent distribution because that differentiates DLVM methods from

TABLE II
Optimal bandwidths, h_{opt} in Table I, scaled by the factor α. The estimated bandwidth scales to higher dimensions with limited KDE samples.

$l\backslash m$	500	1000	2000	5000	10000
10	0.60 ± 0.00	0.58 ± 0.00	0.56 ± 0.00	0.53 ± 0.00	0.51 ± 0.00
20	0.67 ± 0.00	0.65 ± 0.00	0.64 ± 0.00	0.62 ± 0.00	0.61 ± 0.00
40	0.72 ± 0.00	0.71 ± 0.00	0.70 ± 0.00	0.69 ± 0.00	0.68 ± 0.00
50	0.73 ± 0.00	0.72 ± 0.00	0.71 ± 0.00	0.70 ± 0.00	0.70 ± 0.00
70	0.74 ± 0.00	0.74 ± 0.00	0.73 ± 0.00	0.73 ± 0.00	0.72 ± 0.00
100	0.76 ± 0.00	0.75 ± 0.00	0.75 ± 0.00	0.74 ± 0.00	0.74 ± 0.00

basic, bottle-necked reconstruction networks, such as autoencoders. To this end, we perform statistical analysis on the latent representations inferred from data by the model. In our analysis, we evaluate the mismatch between the approximate marginal posterior distribution (using the latent encodings) and the prior, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in the following setup: (i) approximate the posterior distribution by a Gaussian distribution and match its parameters with that of the prior, and (ii) estimate the entropy of the whitened latent encodings of test samples (using the parameters of the empirical Gaussian distribution). In addition, we study whether the latent representations built by DLVMs can explain the variations in the observed data, known as the *disentanglement analysis* [8], [11], [17], [43]. This analysis aims to identify independent latent variables that represent true generative factors. For a perfect disentangled latent space, each latent variable must correspond to a single generative factor without any overlap with other latent variables.

Data sets: We use several benchmark data sets, MNIST [9], CelebA [44], and CIFAR10 [22] for empirical evaluation of different methods. These are widely used data sets for the evaluation of generative models. We split the data (train, test) following the documentation in the literature [9], [22], [44]. To address the data set's complexity, the size of the latent space, neural network architectures, model-specific hyper-parameters, and other optimization parameters are altered accordingly. For example, the size of the latent space grows from $l = 16$ in MNIST to $l = 128$ in CIFAR10. Likewise, the complexity of the neural network architectures varies from one data set to the other. For the disentanglement analysis, we use the DSprites [45] and the 3D Shapes [46] data sets. The true generative factors for the observed data are known in these data sets, which helps quantify the disentanglement in the learned latent representations.

Implementation details: For a given data set, we use the same latent dimension, encoder-decoder architecture, and optimization strategies (such as the learning rate, learning rate scheduler, epochs, and batch size) for all the competing methods to ensure a fair comparison. The VAE and β-TCVAE are trained for more epochs on the MNIST data set for convergence. The latent dimensions used in the MNIST, CelebA, and CIFAR10 data sets are $l = 16, 64$, and 128, respectively [25], [35]. The KDE samples used in the AVAE are $m = 10K, 20K$, and $10K$ for MNIST, CelebA, and CIFAR10 data sets, respectively. In disentanglement analysis using the DSprites and 3D Shapes data sets, we leverage the information of the known latent factors to set the latent size as $l = 6$ for both data sets. The number of the KDE samples used in the DSprites and 3D Shapes data sets is $m = 10K$. Initialization of the model parameters has ramifications on the performance of the methods. We run all methods for 5 different seeds (producing different initialization) for the MNIST, CelebA, and CIFAR10 data sets. Likewise, all methods are trained with 10 different initialization for the DSprites and 3D Shapes data sets. Different initialization also helps in the statistical evaluation of the model performance. The objective function of several methods studied in this work has hyper-parameters associated with the regularization loss that needs to be adjusted, depending on the data set. Mostly, we have used the hyper-parameter settings suggested by the author or recommended in the literature, such as higher β value in the β-TCVAE for better disentanglement (typically set to $\beta \geq 5$) [8], [47]. However, for multiple methods, such as the β-TCVAE and AAE, we have empirically determined the strength of the regularization loss for different data sets as we could not find them in the literature. For e.g., the hyper-parameter β in the β-TCVAE is set to $\beta = 2$ (the minimum value) for the MNIST, CelebA, and CIFAR10 data sets after trying a range of values $\beta \in \{2, 4, 6, 10\}$, as higher β value is leading to poor reconstruction. Details of the neural network architectures and other parameter settings for all the benchmark data sets used by the competing methods are reported in Appendix A.

B. Results

1) Evaluation of the model data distribution: Several methods of sampling the data distribution are appropriate. The quality (FID) of reconstructed samples quantifies the reconstruction quality separately from the learning objective function (e.g., mean squared error) and thus gives additional insight into reconstruction quality. FIDs of reconstructed samples use the latent representations of held-out data for all the data sets. The response of the decoder to the *interpolation* of random pairs in the latent space helps us assess the regularity/smoothness of the latent space. The quality of generated data samples from prior distribution (latent space) gives insight into the combination of prior matching and the subsequent effects of the decoder. Except for the RAE, all the methods considered in this experiment use $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as the prior distribution. For the RAE, we approximate the distribution in the latent space by the Gaussian distribution. Parameters of the Gaussian distribution derived from the latent representations are used to generate new data samples. We know that the latent distribution for the AVAE convergences to $\mathcal{N}\left(\mathbf{0}, \mathbf{I}(1 - h^2)\right)$, where h is the KDE bandwidth (section III-B3). Therefore, we use samples drawn from the distribution, $\mathcal{N}\left(\mathbf{0}, \mathbf{I}(1-h^2)\right)$, to evaluate the generative capability of AVAEs.

We quantitatively evaluate the reconstructed, interpolated, and generated samples in this experiment using the FID scores [41] on multiple benchmark data sets. The FID score is considered a better alternative to the Inception Score (IS) [32] due to its properties, such as rewarding variability in the generated examples, robustness to noise [41], [48]. A lower FID score indicates better matching of the data distribution. Results are reported in Table III. The generative capability of the AVAE is the best among all the considered methods for all the benchmark data sets. The FID scores of the reconstructed and interpolated samples of the MNIST and CelebA data set closely follow the best-performing method. However, the AVAE is the best-performing method for the CIFAR10 data set (a highly complex and challenging data set) under all test scenarios. We set $\beta = 2$ in the β-TCVAE (minimum value for β) to improve the reconstruction loss, which makes it comparable to other methods studied in this work. However, this strategy for the β-TCVAE did not enhance the quality of the reconstructed data for the MNIST and CIFAR10 data sets. This possibly could be due to the challenge of learning multiple modes (10 classes) present in the MNIST and CIFAR10 data sets, unlike the CelebA data set, which tends to consist of smooth variations (such as skin tone, hair color/style, pose) around a very similar, highly represented, central set of examples. Reconstructed and generated images produced by different methods trained on the CelebA data set are shown in Figure 2. The performance of the RAE is promising across all data sets under different evaluation scenarios. This method produces the best FID score on the reconstructed and interpolated examples of the MNIST and CelebA data sets and is close to the best method for the CIFAR10 data set. AAEs do a reasonably good job of learning the data distribution of all the data sets. The performance of the AAE is very close to the best-performing methods for the MNIST and CelebA data sets. Similar to the AAE, the WAE also does well on the MNIST and CelebA data sets. For WAEs, the FID score of the reconstructed and interpolated examples of the CIFAR10 data set is comparable to the best-performing methods. From this experiment, we conclude that, unlike other methods, the performance of the AVAE is consistent across all data sets and under different evaluation scenarios.

Besides the FID metric, we evaluate the diversity and quality of the generated samples produced by different DLVMs using the precision and recall metric [42]. A higher precision indicates good quality of the generated samples, and a higher recall suggests that the model data distribution covers the modes present in the true data distribution. The performance of the DLVMs under this metric is reported in Table IV. Though the VAE produces the best scores for the MNIST data set, its performance drops significantly for the CelebA and CIFAR10 data sets. The β-TCVAE performs poorly for precision and recall across all the data sets. The AVAE is the best or the second best performing method under the precision and recall metric across all the data sets studied in this work. Other than the AVAE, the RAE and AAE perform reasonably well in different evaluation scenarios. The performance of the WAE is comparable to other methods for the MNIST and CelebA data sets. However, it fails to learn the complex data distribution of CIFAR10, similar to the VAE and β-TCVAE.

2) Evaluation of statistical properties of the latent representations: In general, LVMs match a prior distribution in the latent space. In this work, we chose the Gaussian distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, as the target/prior. After a DLVM is trained to convergence, it is interesting to study the properties of the latent representations produced by the same. In this work, we explore how close the latent distributions in the latent space are to the prior; this indicates how strongly one might rely on Gaussian properties in this space for downstream tasks. This exercise will also help us comprehend the poor FID scores of different methods reported in Table III. It exposes the shortcomings in formulating different DLVMs to learn the data distribution. DLVMs experience holes representing regions strongly supported under the prior but underrepresented or ignored by the posterior [10]. Matching moments of the distribution cannot explain such a phenomenon in latent distributions, and to our knowledge, there is no direct way of estimating the same. To this end, we study some known information-theoretic properties of the prior distribution to evaluate the deviation of the posterior distribution from the prior.

Matching distributions in the latent space: A mismatch between distributions can be quantified using different divergence measures, such as the Kullback–Leibler divergence (KLD) and Jensen–Shannon divergence (JSD). As we use the Gaussian distribution as the prior, we ought to match the parameters of the Gaussian distribution $q(\mathbf{z} \mid \mu, \Sigma)$ derived from the latent encodings with the prior, $p(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$, while accounting for known biases. A closed-form solution exists to quantify the mismatch between the multivariate Gaussian distributions using the KLD. In this experiment, we compute the KL divergence, $\mathrm{KL}\left(q(\mathbf{z}) || p(\mathbf{z})\right)$ using the latent representations of held-out data (10K samples), reported in Table V.

The divergence measures indicate that the AVAE closely matches the prior distribution across data sets compared to all other methods. It has the best estimate for the MNIST and CIFAR10 data sets. For the CelebA data set, the posterior distribution statistics are similar to the prior for all the methods except the WAE and RAE. Matching distributions in high dimensions is always a challenge. Therefore, we observe that the divergence measure for the CIFAR10 data set using $l = 128$ is higher (even for the best method, AVAE) than other latent spaces $l = 16, 64$ (used for the MNIST and CelebA data set).

The VAE and β-TCVAE fail to match the distribution for the MNIST and CIFAR10 data sets. A mismatch with the prior will likely have ramifications in learning the data distribution reflected in the FID scores. Thus, the higher KLD estimate explains poor FID scores of the VAE and β-TCVAE (Table III) for the MNIST and CIFAR10 data sets. Similarly, we find a positive correlation between the FID scores of the generated samples produced by the WAE and KLD for the CelebA and CIFAR10 data sets. The AAE matches the aggregate posteriors across data sets, resulting in FID scores comparable to the

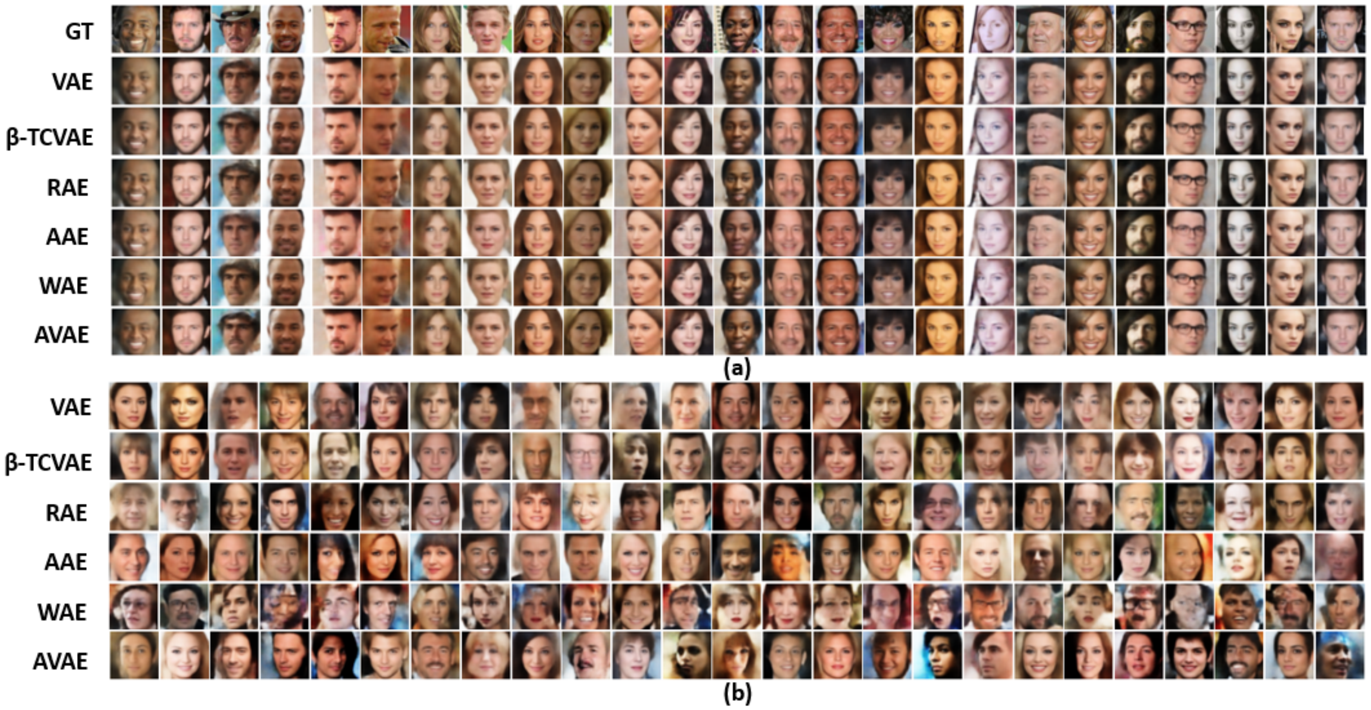

Fig. 2. (a)Reconstructed and (b)generated examples of the CelebA data set produced by the competing methods. **GT** represents the ground truth of the reconstructed examples.

TABLE III
FID SCORES [41] OF COMPETING METHODS TRAINED WITH 5 DIFFERENT SEEDS FOR MULTIPLE DATA SETS (LOWER IS BETTER). THE **BEST** SCORE IS IN **BOLD** AND THE SECOND BEST SCORE IS UNDERLINED.

	MNIST $(l = 16) \downarrow$			CelebA $(l = 64) \downarrow$			CIFAR10 $(l = 128)$ $\downarrow$		
	REC.	SAMPLES		REC.	SAMPLES		REC.	SAMPLES	
		INT.	$\mathcal{N}$		INT.	$\mathcal{N}$		INT.	$\mathcal{N}$
VAE	26.45 ± 0.23	29.96 ± 0.62	28.78 ± 0.48	41.98 ± 0.33	45.95 ± 0.43	49.89 ± 0.57	136.21 ± 1.18	160.78 ± 1.41	147.74 ± 0.81
β-TCVAE	48.46 ± 0.83	53.07 ± 0.84	50.62 ± 1.19	43.37 ± 0.62	47.97 ± 0.71	50.14 ± 0.78	172.87 ± 1.70	195.4 ± 1.55	180.94 ± 1.16
RAE	$\mathbf{7.98 \pm 0.22}$	$\mathbf{13.78 \pm 0.11}$	$\underline{18.79 \pm 0.31}$	$\mathbf{39.86 \pm 0.64}$	$\mathbf{43.53 \pm 0.79}$	48.81 ± 1.02	$\underline{53.76 \pm 0.59}$	$\underline{72.86 \pm 1.09}$	$\underline{94.34 \pm 1.58}$
AAE	11.25 ± 0.95	18.56 ± 2.24	19.03 ± 2.09	42.51 ± 1.35	45.98 ± 2.71	$\underline{47.01 \pm 0.91}$	86.4 ± 12.29	110.86 ± 21.79	101.92 ± 4.18
WAE	9.86 ± 0.19	17.72 ± 0.43	25.42 ± 1.19	40.79 ± 0.16	45.87 ± 0.26	72.01 ± 2.26	66.18 ± 1.16	77.73 ± 1.18	140.49 ± 0.64
AVAE	$\underline{9.32 \pm 0.36}$	$\underline{14.16 \pm 0.17}$	$\mathbf{13.27 \pm 0.34}$	$\underline{40.23 \pm 0.36}$	$\underline{44.44 \pm 0.39}$	$\mathbf{46.0 \pm 0.42}$	$\mathbf{52.83 \pm 0.47}$	$\mathbf{71.63 \pm 1.74}$	$\mathbf{90.93 \pm 6.65}$

best-performing methods. The RAE does not match a prior distribution in the latent space. However, we still reported the KLD estimates to understand how far the posterior is from the prior distribution using the simple regularization technique.

Deviations beyond the second moment: In this experiment, we evaluate deviations of the resultant distribution beyond the second moment, as we would expect from holes or clusters in the distribution. For this, we use the entropy of the posterior distribution to quantify how close it is to Gaussian, *after whitening the distribution* to remove the effects of the second moment mismatch. Because the Gaussian distribution has the maximum entropy (for a given mean and covariance), we use the entropy of the whitened data. Entropy is defined as

$$H(\mathbf{z}) = \mathbb{E}_{\mathbf{z} \sim q_\phi^{\mathrm{w}}(\mathbf{z})} \left\{ -\log\left(q_\phi^{\mathrm{w}}(\mathbf{z})\right) \right\} \approx \frac{1}{m} \sum_j \frac{1}{m-1} \sum_{i \neq j} K\left(\frac{||\mathbf{z}_j - \mathbf{z}_i^{'}||}{h} \right), \qquad (12)$$

where $q_\phi^{\mathrm{w}}(\mathbf{z})$ is the posterior distribution over the whitened data. We use the KDE (defined in 4) for estimating the density $q_\phi^{\mathrm{w}}(\mathbf{z})$ for all methods because it can, in principle model the deviations we are seeking to evaluate. The bandwidth h required in the KDE for the latent dimensions $l = \{16, 64, 128\}$ and KDE samples $m = 10K$ is derived using the strategy defined in section III-B4. The entropy computation uses the same held-out data set used in matching distributions. The entropy of the standard normal distribution derived analytically serves as the ground truth.

From the results reported in Table VI, we observe that the entropy scores of the VAE and β-TCVAE are far off from the ground truth for the MNIST and CIFAR10 data sets even using the whitened latent representations. Low entropy scores can be attributed to the formation of clusters. The entropy scores offer another perspective to explain the high FID scores of the generated samples produced by the VAE and β-TCVAE for the MNIST and CIFAR10 data sets. Poor FID scores of the WAE can be related to the low entropy values across data sets. The low entropy scores of the RAE are not surprising because it does not attempt to match any prior distribution in the latent space. However, the regularization approach is better than the

TABLE IV
Evaluation of the precision and recall metric [42] over the generated samples produced by the competing methods (higher is better). The precision and recall value in a cell is reported in the left and right side, respectively. All models are trained with 5 different seeds. The **best** score is in **bold** and the <ins>second best</ins> score is <ins>underlined</ins>.

	MNIST ($l = 16$) ↑	CelebA ($l = 64$) ↑	CIFAR10 ($l = 128$) ↑
VAE	$\mathbf{0.72 \pm 0.01}$ | $\mathbf{0.88 \pm 0.01}$	0.64 ± 0.01 | 0.48 ± 0.02	0.36 ± 0.01 | 0.24 ± 0.03
β-TCVAE	0.63 ± 0.02 | 0.78 ± 0.01	0.62 ± 0.02 | 0.45 ± 0.02	0.19 ± 0.01 | 0.15 ± 0.02
RAE	0.66 ± 0.01 | 0.81 ± 0.01	0.68 ± 0.01 | $\mathbf{0.55 \pm 0.02}$	$\mathbf{0.67 \pm 0.01}$ | 0.32 ± 0.01
AAE	0.64 ± 0.02 | $\underline{0.86 \pm 0.02}$	$\underline{0.70 \pm 0.03}$ | 0.50 ± 0.03	0.58 ± 0.03 | $\underline{0.33 \pm 0.04}$
WAE	$\underline{0.71 \pm 0.04}$ | 0.81 ± 0.02	0.49 ± 0.02 | $\mathbf{0.55 \pm 0.02}$	0.36 ± 0.01 | 0.15 ± 0.05
AVAE	$\mathbf{0.72 \pm 0.02}$ | $\underline{0.86 \pm 0.02}$	$\mathbf{0.74 \pm 0.01}$ | $\underline{0.53 \pm 0.02}$	$\underline{0.59 \pm 0.04}$ | $\mathbf{0.43 \pm 0.02}$

TABLE V
KL divergence between the approximate marginal posterior distribution (derived empirically) and the prior on multiple data sets (lower is better). All models are trained with 5 different seeds. The **best** score is in **bold** and the <ins>second best</ins> score is <ins>underlined</ins>.

	MNIST ($l = 16$) ↓	CelebA ($l = 64$) ↓	CIFAR10 ($l = 128$) ↓
VAE	19.86 ± 2.57	$\mathbf{0.92 \pm 0.03}$	253.35 ± 1.65
β-TCVAE	41.63 ± 4.32	1.51 ± 0.08	292.10 ± 1.08
RAE	7.61 ± 0.85	70.7 ± 2.63	312.28 ± 2.89
AAE	$\underline{0.57 \pm 0.13}$	$\underline{1.30 \pm 0.36}$	$\underline{33.69 \pm 7.53}$
WAE	0.84 ± 0.05	12.61 ± 0.44	75.32 ± 1.11
AVAE	$\mathbf{0.31 \pm 0.10}$	1.63 ± 0.06	$\mathbf{24.74 \pm 2.75}$

TABLE VI
Mean entropy of the $q_{\phi}^{\mathrm{w}}(\mathbf{z})$ produced by competing methods (each method is trained 5 times on a data set, initialized differently in every run) on the benchmark data sets (higher is better). The **best** score is in **bold** and the <ins>second best</ins> score is <ins>underlined</ins>. The entropy of the standard normal distribution (leaving out the constant) is used as the ground truth.

$Method \backslash Samples$	MNIST ($l = 16$) ↑	CelebA ($l = 64$) ↑	CIFAR10 ($l = 128$) ↑
VAE	4.71 ± 0.14	28.88 ± 0.03	29.56 ± 0.39
β-TCVAE	3.44 ± 0.39	28.42 ± 0.05	15.02 ± 0.69
RAE	4.89 ± 0.06	27.08 ± 0.09	49.92 ± 0.31
AAE	$\underline{5.79 \pm 0.05}$	$\mathbf{31.06 \pm 0.03}$	$\underline{53.53 \pm 0.66}$
WAE	4.67 ± 0.09	28.32 ± 0.05	48.10 ± 0.45
AVAE	$\mathbf{7.56 \pm 0.10}$	$\underline{30.96 \pm 0.02}$	$\mathbf{55.64 \pm 0.00}$
Standard Normal	8.00	32.00	64.00

VAE. The AAE has entropy scores comparable to the AVAE, and it also helps us comprehend the consistent FID scores of the generated samples. The AVAE has the best entropy score for all the data sets except the CelebA, which is marginally smaller than the best entropy score.

3) Evaluation of disentanglement in the latent space: One of the objectives of the DLVM is to identify generative factors that can explain the variability in the observed data. DLVMs are designed to learn meaningful representations in an unsupervised manner, such that the hidden explanatory factors are interpretable by independent latent variables (aka disentanglement) [43]. Thus, it is helpful to quantify the level of disentanglement achieved by a DLVM. Several metrics have been proposed in the recent past, such as the Factor VAE metric [17], Mutual Information Gap (MIG, [8]), β-VAE metric [11], DCI disentanglement [49], and SAP score [50]. Most of the existing disentanglement metrics assume that the latent variables explaining the variation in data are aligned with the latent axes (cardinal directions), even when the prior or target is an isotropic Gaussian, which is invariant to rotation. These disentanglement metrics have been mostly studied on DLVMs based on the VAE [1], [2], such as the βTCVAE [8], and Factor VAE [17]. The formulation and modeling choices of the VAE-based methods favor alignment of the latent variables with the latent axes that make these methods amenable to evaluation using the existing metrics. However, there are other DLVMs, such as the AAE [34] and WAE-MMD [35], for which the latent variables might not be aligned with the latent axes. Regardless, we would still like to quantify in those cases the extent to which the latent space *disentangles* latent factors along independent directions.

To facilitate the use of the existing disentanglement metrics for the evaluation of DLVMs, in a general setup, we aim to identify directions (unit vectors) in the latent space representing latent variables associated with the true generative factors instead of using the cardinal axes in the latent space as generative factors. To determine the direction corresponding to a factor $\mathcal{F}_i$, L observed data are selected, where the factor, $\mathcal{F}_i$, is fixed to a value for all L samples. The remaining factors are assigned different values in each instance. The principal component analysis (PCA) is done on the encoding of L observed data, produced by a trained DLVM, and the eigenvector with minimum variance, u_i, is chosen as the representative of the ground truth factor, $\mathcal{F}_i$ (i.e., kept fixed). This step is

Algorithm 2 : Determine latent directions for generative factors in DLVMs, in a general setup

Input: Trained encoder E_ϕ, Latent factors $\mathcal{F}$, Values for the latent factors $\mathcal{V}$ ($\mathcal{V}_i$ has the values for the factor, $\mathcal{F}_i$), L samples used for the PCA analysis of a generative factor ($\mathcal{F}_i$ is set to a fixed value chosen from $\mathcal{V}_i$ and all others factors, $\mathcal{F}_{-i}$,i.e., $\mathcal{F} \setminus i$, are allowed to vary) and Number of pca analysis, N, to determine the direction for a generative factor, $\mathcal{F}_i$.

Output: Directions in the latent space, $\mathcal{D}$, for all the generative factors, $\mathcal{F}$.

```
$\mathcal{D} \leftarrow \emptyset$
for Each ground truth factor $k$ in $\mathcal{F}$ do
    Sample $N$ values for the ground truth factor $k$, $\mathcal{S}^k$
    $\mathcal{U} \leftarrow \emptyset$
    for each element $\mathcal{S}^k_i$ in $\mathcal{S}^k$ do
        Sample factors other than $k$ ($\mathcal{F}_{-k}$) $L$ times, $\mathcal{S}^{-k}$
        {Concatenate $\mathcal{S}^k_i$ to $L$ samples in $\mathcal{S}^{-k}$}
        $\mathcal{S}^i \leftarrow \mathcal{S}^{-k} \frown \mathcal{S}^k_i$
        Get observed data, $\mathcal{X}^i$, corresponding to $\mathcal{S}^i$, where $\mathcal{X}^i = \{x^i_1, x^i_2 \dots x^i_L\}$ and $x^i_j \in \mathbb{R}^d$
        $\mathcal{Z}^i \leftarrow E_\phi(\mathcal{X}^i)$, where $\mathcal{Z}^i = \{z^i_1, z^i_2 \dots z^i_L\}$ and $z^i_j \in \mathbb{R}^l$
        $\{(\sigma_j, u_j)\}^l_{j=1} \leftarrow \mathbb{PCA}(\mathcal{Z}^i)$, where $(\sigma_j, u_j)$ represents the eigenvector ($u_j$) and the corresponding variance ($\sigma_j$) estimated from the PCA
        $u_i$ is the eigenvector with the minimum variance
        $\mathcal{U} \leftarrow \mathcal{U} \cup u_i$
    end for
    {Estimate $u^*$ representing the factor, $\mathcal{F}_i$}
    $\hat{U} \leftarrow 0$
    for $u_i$ in $\mathcal{U}$ do
        $\hat{U} \leftarrow \hat{U} + u_i u_i^T$
    end for
    $\hat{U} \leftarrow \frac{\hat{U}}{N}$
    $\{(\sigma_j, u_j)\}^l_{j=1} \leftarrow \mathbb{PCA}(\hat{U})$
    $u^*$ is the eigenvector with the maximum variance
    $\mathcal{D} \leftarrow \mathcal{D} \cup u^*$
end for
```

similar to the data generation technique of the Factor VAE metric [17] that associates a latent axis with a generative factor based on the variance along the latent axes. Identifying the minimum variance eigenvector is repeated multiple times (N) to capture the variation in u_i's for different values of $\mathcal{F}_i$ and $\mathcal{F}_{-i}$. The optimum unit vector, u^*, representing the factor, $\mathcal{F}_i$, is obtained by solving the optimization problem $\max \sum_{i=1}^{N} (u^{*T} u_i)^2$, where the u_i is an eigenvector. The solution to the optimization problem is the eigendecomposition of the mean outer product of the eigenvectors (u_i), and u^* is the eigenvector with the maximum variance. The above steps are repeated to get the latent directions (u^*) for all the ground truth factors. The outline of the algorithm is presented in Algorithm 2.

In this work, we follow the strategy proposed in the Factor VAE metric [17] and MIG metric [8] to devise techniques for disentanglement analysis using the latent directions, $\mathcal{D}$. However, using the estimated latent directions is not limited to these unsupervised metrics. We name the proposed metrics as the *PCA Factor VAE metric* and the *PCA MIG metric*. Finding latent directions (representing latent variables), $\mathcal{D}$, corresponding to generative factors, $\mathcal{F}$, is a generalization of the majority vote classifier used in the Factor VAE metric. Thus, we can use the latent directions $\mathcal{D}$ for predicting the latent factor, $\hat{\mathcal{F}}_i$, for u_i (produced using the inner loop in algorithm 2). In the PCA Factor VAE metric, we use a similarity measure between u_i and the set of latent directions, $\mathcal{D}$, to predict the corresponding latent factor, $\hat{\mathcal{F}}_i$, and compare it to the true generative factor, $\mathcal{F}_i$, using cosine similarity measure (normalized correlation). The prediction accuracy of a model is the score for the PCA Factor VAE metric. In the PCA MIG metric, latent representations ($\mathcal{Z} = E_\phi(\mathcal{X})$) are projected onto the latent directions, $\mathcal{D}$, and the MIG of the transformed representations ($\mathcal{Z}' = \mathcal{Z}\mathcal{D}^T$) gives the score for this metric.

The performance of the competing methods under the proposed disentanglement metrics are reported in Table VII for the DSprites [45] and 3D Shapes [46] data sets. The size of the latent space is set to the number of generative factors in the data set, $\mathcal{Z} \in \mathbb{R}^6$ for both the DSprites [45] and 3D Shapes [46] data sets. The details of the network architecture and the associated hyper-parameters are reported in Table XI of Appendix A. All of the DLVMs studied in this experiment have a hyper-parameter (except the AVAE) that regulates the trade-off between the reconstruction and regularization loss. A relatively poor reconstruction loss indicates stronger regularization of the latent representation, which might result in better disentanglement. Thus, knowing the reconstruction loss of different methods in a given experimental setup is informative. In an ideal scenario, we expect a higher disentanglement score with low reconstruction loss. In this experiment, we consider the Factor VAE metric [17] and the MIG metric [8] as the baseline and compute the same for the VAE and β-TCVAE. We do not evaluate other competing methods under these metrics as the latent variables might not be aligned with the latent axes, leading to incomprehensible scores. However, the scores for the PCA Factor VAE metric [17] and the PCA MIG metric are computed for all the methods studied in this work. Comparing the PCA MIG metric with the MIG metric [8] (baseline) demonstrates the impact of using the latent directions, $\mathcal{D}$, as latent variables relative to the latent axes. We observe an overall improvement in the performance of the VAE and β-TCVAE using latent directions, $\mathcal{D}$, in the computation of the MIG scores.

Considering the metric scores reported in Table VII, the AVAE produces the best score under both the metrics for the 3D Shapes data set with a slightly higher MSE score (second best). This indicates that the AVAE achieves better disentanglement without compromising the quality of the reconstructed data, a desired property of a DLVM. The AVAE achieves significantly higher scores than the β-TCAVE (the second best performing method) for both the metrics and sets a new SOTA result for the 3D Shapes data set. Higher metric scores for the AVAE are corroborated with the latent traversal along the latent directions, $\mathcal{D}$, shown in Figure 3, where we

TABLE VII

DISENTANGLEMENT SCORES OF COMPETING METHODS TRAINED WITH 10 DIFFERENT SEEDS FOR MULTIPLE DATA SETS (HIGHER IS BETTER). THE **BEST** SCORE IS IN **BOLD** AND THE <u>SECOND BEST</u> SCORE IS <u>UNDERLINED</u>.

METHOD	DSPRITES ($l = 6$)			3D SHAPES ($l = 6$)		
	PCA Factor VAE ↑	PCA MIG ↑	MSE ↓	PCA Factor VAE ↑	PCA MIG ↑	MSE ↓
VAE (baseline)	64.80 ± 7.50	0.07 ± 0.03	3.68 ± 0.58	56.19 ± 8.85	0.13 ± 0.13	10.47 ± 1.10
VAE	75.56 ± 7.21	0.14 ± 0.04	3.68 ± 0.58	55.0 ± 13.56	0.15 ± 0.15	10.47 ± 1.10
β-TCVAE (baseline)	75.55 ± 3.52	$\mathbf{0.22 \pm 0.09}$	6.39 ± 2.05	72.40 ± 12.80	0.40 ± 0.22	11.54 ± 1.86
β-TCVAE	68.40 ± 14.09	0.18 ± 0.13	6.39 ± 2.05	$\underline{74.72 \pm 16.12}$	$\underline{0.42 \pm 0.23}$	11.54 ± 1.86
RAE	$\mathbf{82.03 \pm 2.58}$	0.16 ± 0.03	$\mathbf{2.51 \pm 0.20}$	70.85 ± 20.02	0.33 ± 0.17	10.77 ± 1.25
AAE	49.15 ± 4.23	0.04 ± 0.01	6.95 ± 0.51	46.22 ± 4.13	0.04 ± 0.02	14.6 ± 3.61
WAE	62.72 ± 6.85	0.07 ± 0.02	3.69 ± 0.34	52.34 ± 3.00	0.20 ± 0.05	$\mathbf{9.80 \pm 1.95}$
AVAE	$\underline{79.17 \pm 1.64}$	$\underline{0.20 \pm 0.02}$	$\underline{2.98 \pm 0.28}$	$\mathbf{91.93 \pm 3.27}$	$\mathbf{0.67 \pm 0.04}$	$\underline{10.29 \pm 0.37}$

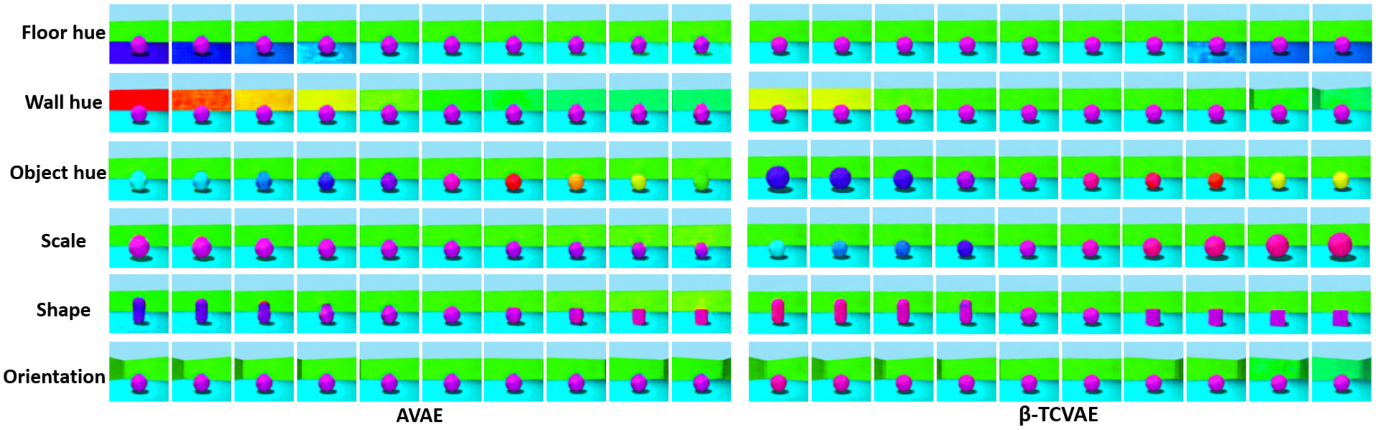

Fig. 3. Latent traversal of the 3D Shapes data set [46] in the range $[-\sigma, \sigma]$ for models trained using the AVAE and β-TCVAE. The latent factors are mentioned in the left column. All latent factors are represented by independent latent variables in the AVAE with almost no overlap between latent variables, except slight variation in object color with shapes. For the β-TCVAE, we observe entanglement of the multiple latent factors, such as the object color with scale, and wall color with orientation.

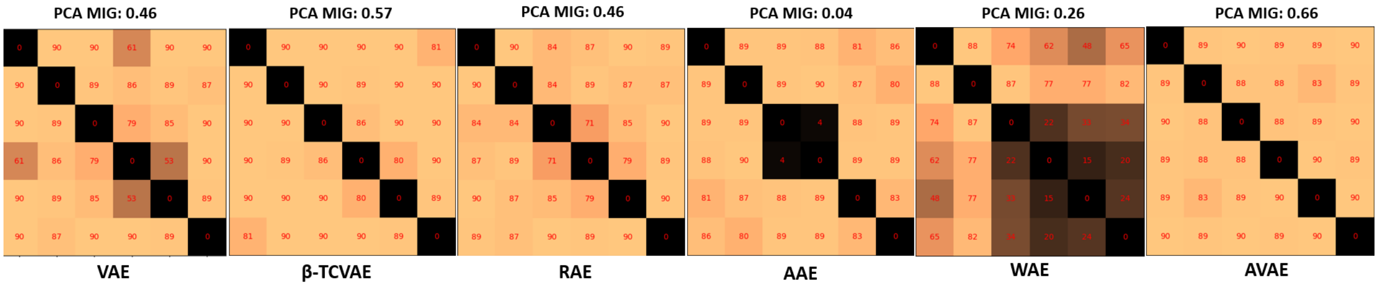

Fig. 4. Pairwise angle between the latent directions (six directions) estimated by Algorithm 2 for different DLVMs using the corresponding latent representations produced for the 3D Shapes data set [46]. The latent directions should be orthogonal to each other for better disentanglement. Deviation from the orthogonality indicates entanglement of the ground truth generative factors that result in poor metric scores, as observed in the AAE and WAE. To interpret the estimated latent directions in the analysis of disentanglement, we report the corresponding PCA MIG metric score for each model.

observe that all the latent factors are represented by separate independent latent variables with no overlap, except for slight variation in object color with shapes. Though the RAE has the maximum score for the PCA Factor VAE metric when trained on the DSprites data set, its performance is poor under the PCA MIG metric. For the DSprites data set, the β-TCAVE (baseline) shows some improvement over the AVAE under the MIG metric. However, its Factor VAE metric score is less than the AVAE. Overall, the performance of the AVAE is consistent under both metrics (with good reconstruction loss) relative to other methods for both data sets.

For independent latent variables, the latent directions should be orthogonal. The collapse of the latent directions or small angle between them indicates the entanglement of ground truth generative factors to latent variables, which should have strong ramifications on the disentanglement metrics. With this motivation, we study the angles between the latent directions estimated by Algorithm 2. Figure 4 shows the angles between the estimated latent directions for different DLVMs trained using the 3D Shapes data set. Every latent direction estimated for the AVAE is almost perpendicular to all other latent directions. This property explains the high metric scores reported for the AVAE in Table VII and adds meaning to each latent direction (aka latent variables) as illustrated in Figure 3. Small variations in the metric scores (when initialized with different seeds) suggest that the observed behavior in the AVAE is not typical of an initialization. The collapse of the latent directions in the AAE and WAE shown in Figure 4 is likely responsible for the poor performance under the evaluation metrics.

V. Conclusion

We propose a novel algorithm, the aggregate-VAE (AVAE), based on the framework of the VAE to match the aggregate posterior distribution to the prior using the KDE. We formulated a method for estimating the high-dimensional KDE bandwidth, comparable to the latent space used in the StyleGAN [4]. The encoder and decoder parameters are trained jointly to minimize the reconstruction loss and match distributions in latent spaces using the objective function of the AVAE. The dynamic adjustment of the scaling factor, β, using the validation data has helped maintain the balance between the reconstruction and regularization loss. The proper choice of the KDE bandwidth and joint optimization of the reconstruction loss and aggregate posterior matching addresses the issue of the posterior collapse, a shortcoming in the VAE formulation. The performance of the AVAE is consistently the best or comparable to SOTA methods on multiple metrics across several benchmark data sets. The FID scores of the reconstructed data confirm that matching distributions in the latent space did not impede the reproduction of the input data using the corresponding latent encodings. Matching the aggregate distribution to the prior in the AVAE resulted in the best FID scores relative to other methods. Low values of the KL divergence between the aggregate posterior and prior for the AVAE corroborate our understanding of latent distribution matching, resulting in good FID scores of the generated samples. In our analysis of the entropy of the aggregate posterior distribution, high entropy scores for the AVAE indicate that the latent representations are close to Gaussian and have a lower chance of encountering holes/clusters in the distribution. The proposed method of finding latent directions corresponding to true generative factors helps evaluate disentanglement in the latent space of DLVMs (not restricted to VAEs) using the existing metrics. The latent directions of the AVAE help in the meaningful interpretation of the latent representations. The AVAE emerged as the best performer for the Shapes 3D data set and is the second-best performer for the DSprites data set.

Appendix
Experimental settings

In the neural network architectures reported in Table VIII, IX and X, Conv$n$ and TransConv$n$ define convolution and transpose convolution operation, respectively, with n filters in the output. We have used 4×4 filters for all the data sets. The transpose convolution filters use a stride size of 2 except for the last layer of the decoders used in the CelebA and CIFAR10 data sets. We represent the fully connected layers as $\text{FC}_{k \times n}$ with $k \times n$ nodes, where $k = 1$ for all the methods, except the VAE and β-TCVAE that use $k = 2$. Activation functions used in the networks are ReLU (ReLU), Leaky ReLU (LReLU), sigmoid (Sigmoid), and hyperbolic tangent (Tanh). Input is in the range $[0, 1]$ for all the data sets except CelebA, for which the input is mapped to the range $[-1, 1]$. We use the Adam optimizer in all experiments (learning rate set to $5e{-}04$) with a learning rate scheduler (ReduceLROnPlateau) that reduces the learning rate by 0.5 if the validation loss does not improve for 5 epochs. All the methods are trained for 50, 50, and 100 epochs for the MNIST, CelebA, and CIFAR10 data sets, respectively, except the VAE and β-TCVAE, which we trained for 100 epochs for the MNIST data set. In disentanglement analysis, all the methods are trained for 35 and 60 epochs [47] for the DSprites and 3D Shapes data sets, respectively. We use a batch size of 100 for training all the methods. Additional optimization details are reported in Table XI.

References

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations*, 2014.

[2] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning*, 2014, pp. 1278–1286.

[3] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The helmholtz machine," *Neural computation*, 1995.

[4] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Conference on Neural Information Processing Systems*, 2021.

[5] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[7] T. Cox and M. Cox, *Multidimensional Scaling*. London: Chapman Hall, Boca Raton, 2001.

[8] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in vaes," in *Conference on Neural Information Processing Systems*, 2019.

[9] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," 2010, available on: http://yann.lecun.com/exdb/mnist.

[10] M. Rosca, B. Lakshminarayanan, and S. Mohamed, "Distribution matching in variational inference," *arxiv*, 2018, preprint at https://arxiv.org/abs/1802.06847.

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

[12] J. Lucasz, G. Tuckery, R. Grossez, and M. Norouziy, "Understanding posterior collapse in generative latent variable models," in *International Conference on Learning Representations*, 2019.

[13] ——, "Don't blame the elbo! a linear vae perspective on posterior collapse," in *Conference on Neural Information Processing Systems*, 2019.

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

[15] M. Bauer and A. Mnih, "Resampled priors for variational autoencoders," *International Conference on Artificial Intelligence and Statistics*, 2019.

[16] J. M. Tomczak and M. Welling, "Vae with a vampprior." in *International Conference on Artificial Intelligence and Statistics*, 2018.

[17] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*, 2018.

[18] S. Saha, S. Elhabian, and R. Whitaker, "Gens: generative encoding networks," *Machine Learning*, vol. 111, p. 4003–4038, 2022.

[19] W. Harvey, S. Naderiparizi, and F. Wood, "Conditional image generation by conditioning variational auto-encoders," in *International Conference on Learning Representations*, 2022.

[20] J. Peng, D. Liu, S. Xu, and H. Li, "Generating diverse structure for image inpaintingwith hierarchical vq-vae," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[21] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, p. 1065–1076, 1962.

[22] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009, available on: https://www.cs.toronto.edu/~kriz/cifar.html.

TABLE VIII
Encoder and decoder architectures used by all the methods for the MNIST, CelebA and CIFAR10 data sets [25].

	MNIST	CelebA	CIFAR10
Encoder:	$\mathbf{x} \in \mathbb{R}^{32\times32\times1}$	$\mathbf{x} \in \mathbb{R}^{64\times64\times3}$	$\mathbf{x} \in \mathbb{R}^{32\times32\times3}$
	Conv64 → BN → ReLU	Conv64 → BN → ReLU	Conv128 → BN → ReLU
	Conv128 → BN → ReLU	Conv128 → BN → ReLU	Conv256 → BN → ReLU
	Conv256 → BN → ReLU	Conv256 → BN → ReLU	Conv512 → BN → ReLU
	Conv512 → BN → ReLU	Conv512 → BN → ReLU	Conv1024 → BN → ReLU
	$\text{Flatten}_{2\times2\times512} \rightarrow \text{FC}_{k\times16} \rightarrow$ None	$\text{Flatten}_{4\times4\times512} \rightarrow \text{FC}_{k\times64} \rightarrow$ None	$\text{Flatten}_{2\times2\times1024} \rightarrow \text{FC}_{k\times128} \rightarrow$ None
Decoder:	$\mathbf{z} \in \mathbb{R}^{16} \rightarrow \text{FC}_{2\times2\times512}$	$\mathbf{z} \in \mathbb{R}^{64} \rightarrow \text{FC}_{8\times8\times512}$	$\mathbf{z} \in \mathbb{R}^{128} \rightarrow \text{FC}_{8\times8\times1024}$
	TransConv256 → BN → ReLU	TransConv256 → BN → ReLU	TransConv512 → BN → ReLU
	TransConv128 → BN → ReLU	TransConv128 → BN → ReLU	TransConv256 → BN → ReLU
	TransConv64 → BN → ReLU	TransConv64 → BN → ReLU	TransConv3 → Sigmoid
	TransConv1 → Sigmoid	TransConv3 → Tanh	

TABLE IX
Encoder and decoder architectures used by all the methods for the DSprites and 3D Shapes data sets [17], [47].

	DSprites	3D Shapes
Encoder:	$\mathbf{x} \in \mathbb{R}^{64\times64\times1}$	$\mathbf{x} \in \mathbb{R}^{64\times64\times3}$
	Conv32 → ReLU	Conv32 → ReLU
	Conv32 → ReLU	Conv32 → ReLU
	Conv64 → ReLU	Conv64 → ReLU
	Conv64 → ReLU	Conv64 → ReLU
	$\text{Flatten}_{4\times4\times64} \rightarrow \text{FC}_{k\times6} \rightarrow$ None	$\text{Flatten}_{4\times4\times64} \rightarrow \text{FC}_{k\times6} \rightarrow$ None
Decoder:	$\mathbf{z} \in \mathbb{R}^{6} \rightarrow \text{FC}_{4\times4\times64}$	$\mathbf{z} \in \mathbb{R}^{6} \rightarrow \text{FC}_{4\times4\times64}$
	TransConv64 → ReLU	TransConv64 → ReLU
	TransConv32 → ReLU	TransConv32 → ReLU
	TransConv32 → ReLU	TransConv32 → ReLU
	TransConv1 → None	TransConv3 → Sigmoid

TABLE X
Discriminator architectures used by the AAE for the MNIST, CelebA, CIFAR10, DSprites and 3D Shapes data sets.

MNIST	CelebA	CIFAR10	DSprites	3D Shapes
$\mathbf{z} \in \mathbb{R}^{16} \rightarrow \text{FC}_{100} \rightarrow$ LReLU	$\mathbf{z} \in \mathbb{R}^{64} \rightarrow \text{FC}_{1024} \rightarrow$ LReLU	$\mathbf{z} \in \mathbb{R}^{256} \rightarrow \text{FC}_{1024} \rightarrow$ LReLU	$\mathbf{z} \in \mathbb{R}^{6} \rightarrow \text{FC}_{1024} \rightarrow$ LReLU	$\mathbf{z} \in \mathbb{R}^{6} \rightarrow \text{FC}_{1024} \rightarrow$ LReLU
$\text{FC}_{100} \rightarrow$ LReLU	$\text{FC}_{4096} \rightarrow$ LReLU	$\text{FC}_{4096} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU
$\text{FC}_{100} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU	$\text{FC}_{1024} \rightarrow$ LReLU
$\text{FC}_{1} \rightarrow$ None	$\text{FC}_{1} \rightarrow$ None	$\text{FC}_{1} \rightarrow$ None	$\text{FC}_{1} \rightarrow$ None	$\text{FC}_{1} \rightarrow$ None

TABLE XI
Optimization settings for different methods.

Method	Parameters	MNIST	CelebA	CIFAR10	DSprites	3D Shapes
AVAE	KDE params (k_e, k_b):	(1, 10)	(1, 10)	(1, 10)	(1, 10)	(1, 10)
β-TCVAE	β:	2	2	2	5	5
WAE	Recons-scalar:	0.05	0.05	0.05	0.05	0.05
WAE	β:	10	100	100	10	10
AAE	Recons-scalar:	0.05	0.05	0.05	0.05	0.05
AAE	β:	1	1	1	1	1
RAE	β:	$1e-04$	$1e-04$	$1e-03$	$1e-04$	$1e-04$
RAE	Dec-L2-reg:	$1e-07$	$1e-07$	$1e-06$	$1e-06$	$1e-06$

[23] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *Conference on Neural Information Processing Systems*, 2017.

[24] A. Razavi, A. v. d. Oord, B. Poole, and O. Vinyals, “Preventing posterior collapse with δ-vaes,” in *International Conference on Learning Representations*, 2019.

[25] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholköpf, “From variational to deterministic autoencoders,” in *International Conference on Learning Representations*, 2020.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Conference on Neural Information Processing Systems*, 2014.

[27] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *International Conference on Learning Representations*, 2016.

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

[29] S. Liu, O. Bousquet, and K. Chaudhuri, “Approximation and convergence properties of generative adversarial learning,” in *Conference on Neural Information Processing Systems*, 2017.

[30] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for gans do actually converge?” in *International Conference on Machine Learning*, 2018.

[31] S. A. Barnett, “Convergence problems with generative adversarial networks (gans),” 2018, preprint at https://arxiv.org/abs/1806.11382.

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Conference on Neural Information Processing Systems*, 2016.

[33] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing training of generative adversarial networks through regularization,” in *Conference on Neural Information Processing Systems*, 2017.

[34] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” in *International Conference on Learning Representations*, 2016.

[35] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelköpf, “Wasserstein auto-encoders,” in *International Conference on Learning Representations*, 2018.

[36] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. Smola,

"A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[37] S. R. Bowman and L. Vilnis, "Generating sentences from a continuous space," in *SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016.

[38] H. Shao, S. Yao, D. Sun, A. Zhang, S. Liu, D. Liu, J. Wang, and T. Abdelzaher, "Controlvae: Controllable variational autoencoder," in *International Conference on Machine Learning*, 2020.

[39] O. Rybkin, K. Daniilidis, and S. Levine, "Simple and effective vae training with calibrated decoders," in *International Conference on Machine Learning*, 2021.

[40] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.

[41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Conference on Neural Information Processing Systems*, 2017.

[42] M. S. M. Sajjadi, O. Bachem, M. Lučić, O. Bousquet, and S. Gelly, "Assessing Generative Models via Precision and Recall," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[43] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[44] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild." in *International Conference on Computer Vision*, 2015. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccv/iccv2015.html#LiuLWT15

[45] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dsprites: Disentanglement testing sprites dataset," https://github.com/deepmind/dsprites-dataset/, 2017.

[46] C. Burgess and H. Kim, "3d shapes dataset," https://github.com/deepmind/3d-shapes/, 2018.

[47] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*, 2019.

[48] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly, "Are gans created equal? a large-scale study," in *Conference on Neural Information Processing Systems*, 2017.

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

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