PROTOTYPICAL VARIATIONAL AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational autoencoders are unsupervised generative models that implement latent space regularization towards a known distribution, enabling stochastic synthesis from straightforward sampling procedures. Many works propose various regularization approaches, but most struggle to compromise between proper regularization and good reconstruction quality. This paper proposes distributing the regularization through the latent space using prototypical anchored clusters, each with an optimal position in the latent space and following a known distribution. Such schema enables obtaining an appropriate number of clusters with solid regularization for better reconstruction quality and improved synthesis control. We experiment with our method using widespread exploratory benchmarks and report that regularization anchored on prototypes' coordinates or cluster centroids neutralizes the adverse effects regularization terms often have on autoencoder reconstruction quality, matching non-regularized autoencoders' performance. We also report appealing results for interpreting data representatives with simple prototype synthesis and controlling the synthesis of samples with prototype-like characteristics from decoding white noise around prototype anchors.

1 INTRODUCTION

Deep generative models learn data distributions and can provide realistic synthesis once trained (Karras et al. (2018); Brock et al. (2019); van den Oord et al. (2017); Bond-Taylor et al. (2021)). Most popular approaches involve energy-based models, variational autoencoders, generative adversarial networks, autoregressive models, normalizing flows, and various hybrid approaches that learn implicit or explicit training data distribution. We focus on the latter.

Variational autoencoders (VAE) (Kingma & Welling (2014)) were one of the first deep generative models to propose learning explicit training set distributions efficiently. They introduced a regularization schema that warrants a latent space with good sampling properties using known distributions, more stable training than GANs, and more efficient sampling mechanisms than autoregressive models (Germain et al. (2015)). Like autoencoders, these models consist of an encoder network that maps the input data x into a latent representation z and a decoder that maps the representation back into the original data. To enable efficient synthesis, they further implement a regularization term that forces the latent space to follow a prior distribution $p(z)$, as observed in Figure 1a. VAEs usually compromise between two targets: high reconstruction quality and suitable regularization of the latent space distribution. A common trick for training VAEs is to assume that posteriors and priors are normally distributed, which allows simple Gaussian reparametrization for end-to-end training (Rezende et al. (2014); Kingma & Welling (2014)).

While standard VAE implementations (Kingma & Welling (2014)) use KL regularization loss to enable mapping the latent space distribution to a known standard normal distribution, many works have explored alternate VAE regularization functions to avoid invalid latent space loci that negatively impact the decoder reconstruction performance. InfoVAE (Zhao et al. (2017)) explored information maximization theories to propose regularization terms using maximum mean discrepancy (MMD) theory (Gretton et al. (2012)). MMD implements a divergence by measuring how different the moments of two distributions are, assuming that two distributions are identical if and only if all their moments are the same, using kernel embeddings. Wasserstein auto-encoders (Tolstikhin et al. (2018); Arbel et al. (2019)) use the Wasserstein distance in two different setups using either MMD or an adversarial training schema to regularize the latent space efficiently. More recently, Vector Quantized Variational Autoencoders (VQ-VAE) (van den Oord et al. (2017)) introduced the quantization of latent space delivering synthesis with improved reconstruction quality, even overcoming the results delivered by many popular GAN (Razavi et al. (2019)). Since the optimized quantized codes are discrete following a categorical distribution, one cannot use them directly to generate new samples. In van den Oord et al. (2017) the authors train a PixelCNN using the codes as priors to generate novel examples, which increases much the complexity for stochastic synthesis compared to merely controlling sampling from known latent distribution as in standard VAE.

Different works explored latent space regularization in autoencoders for deep clustering, assuming that objects from the same class share similar features and should be somewhat grouped in the feature space. Xie et al. (2016) proposed the Deep Embedded Clustering method, a framework that alternately learns feature representations and clustering assignments using pre-defined template cluster distributions. Fard et al. (2020) presented the deep-*k*-means algorithm, which applies *k*-means in an AE embedding space to jointly cluster and learn feature representations. They proposed to use a *k*-means clustering loss as the limit of a differentiable function, enabling training the network using back-propagation (Fard et al. (2020)).

Other deep clustering methods dismiss using decoders to learn latent space distributions. Genevay et al. (2019) proposed a differentiable deep clustering method with cluster size constraints, rewriting the *k*-means clustering algorithm as an optimal transport problem with entropic regularization term. Also benefiting from optimal transport, YM. et al. (2020) imposed an equipartition clustering constraint and used a fast version of the Sinkhorn-Knopp algorithm (Knight (2008)) to find an approximate optimal transpor solution. Recently, Swapping Assignments between multiple Views (SwAV) (Caron et al. (2020b)) combined contrastive learning and prototypical clustering, reporting impressive results in self-supervised learning. Similar to YM. et al. (2020), SwAV performs online clustering under an equipartition constraint for each mini-batch using the Sinkhorn-Knopp algorithm.

This paper builds on this literature and presents a novel approach for VAEs latent space regularization using prototypes online clustering. Our approach does not require significant changes in encoder-decoder architectures and implements effective regularization using known distribution clustering around optimal prototype anchor coordinates. We demonstrate that such distributed organization of the latent space neutralizes adverse effects on the reconstruction quality observed when the regularization group together very dissimilar samples, as often happens with standard regularization terms. Our paper experiments with three public computer vision benchmarks (MNIST from LeCun et al. (2010), CIFAR10, and CIFAR100 from Krizhevsky et al. (2012)) and report that our regularization approach neutralizes the commonly observed adverse effects of regularization on autoencoder reconstruction quality with an appropriate number of prototypes, matching non-regularized autoencoders' performance. We report exciting results for model interpretability from prototype anchor synthesis and investigate synthesis control capabilities from decoding white noise around prototype anchors.

We present the following contributions:

- A method for efficient autoencoders regularization using prototypes.
- Improved reconstruction quality using prototypical regularization compared to standard variational autoencoders.
- Increased stochastic synthesis control with decoding samples around prototype anchors.

2 METHOD

2.1 STANDARD VAEs FORMULATION

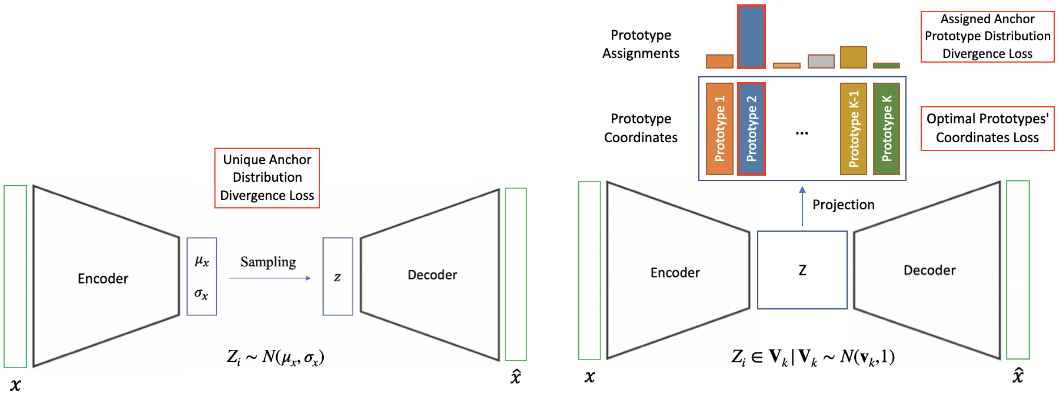
As previously stated, standard variational autoencoders implement regularization through a sampling schema that maps encoded features into a known (commonly normal) distribution 1a. Let us assume an autoencoder that learns encoding q_θ and decoding p_θ distributions through the encoder and decoder networks. A variational autoencoder should then minimize a cost function that is composed by a reconstruction term that targets building outputs very similar to the inputs, and a regularization term, that targets creating a latent space with a known distribution, as following:

$$\mathcal{L}_{VAE} = \mathcal{L}_{reg} + \mathcal{L}_{rec} \tag{1}$$

The reconstruction loss L_{rec} is commonly the mean squared error between a sample x and its corresponding reconstruction $\hat{x}$, or the binary cross-entropy that we use in this paper and is defined as the following:

$$\mathcal{L}_{rec} = -\frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} x_i \log \hat{x}_i + (1 - x_i)\log\,(1 - \hat{x}_i) \tag{2}$$

where $\hat{x}_i$ is the i-th scalar value in the model output, x_i is the corresponding original input, and N is the number of scalar values in x.


Unique Anchor Distribution Divergence Loss
Encoder
μ_x
σ_x
Sampling
z
Decoder
$Z_i \sim N(\mu_x, \sigma_x)$
x
$\hat{x}$
Prototype Assignments
Assigned Anchor Prototype Distribution Divergence Loss
Prototype Coordinates
Prototype 1
Prototype 2
...
Prototype K-1
Prototype K
Optimal Prototypes' Coordinates Loss
Projection
Encoder
Z
Decoder
$Z_i \in \mathbf{V}_k | \mathbf{V}_k \sim N(\mathbf{v}_k, 1)$
x
$\hat{x}$

(a) Vanila Variational Autoencoders

(b) Prototypical Variational Autoencoders

Figure 1: Variational autoencoder with standard and prototypical regularization schemas. While the standard regularization considers a unique anchor distribution targeting $N(0, 1)$, our approach considers regularization using K distributed anchors in the latent space targeting $N(\mathbf{v}_k$,1) distributions.

The regularization loss is implemented in standard VAEs (Kingma & Welling (2014)) as the KL divergence to a known distribution, as follows:

$$\mathcal{L}_{KL}(p, q) = \sum_j KL(p(z|x), q(z)) = -\sum_j \sum_i p(i)\ \log \frac{p(i)}{q(i)} \tag{3}$$

where x is a given sample and $KL(p, N(0, 1))$ is the KL regularization loss between the latent space distribution p_θ and the standard normal distribution for each j latent space dimension, as described in Kingma & Welling (2014).

Another popular regularization term was firstly proposed by Zhao et al. (2017) and builds on maximum mean discrepancy (MMD) theory (Gretton et al. (2012)) to model the regularization loss as the distance between different distribution moments using kernel embeddings, as follows:

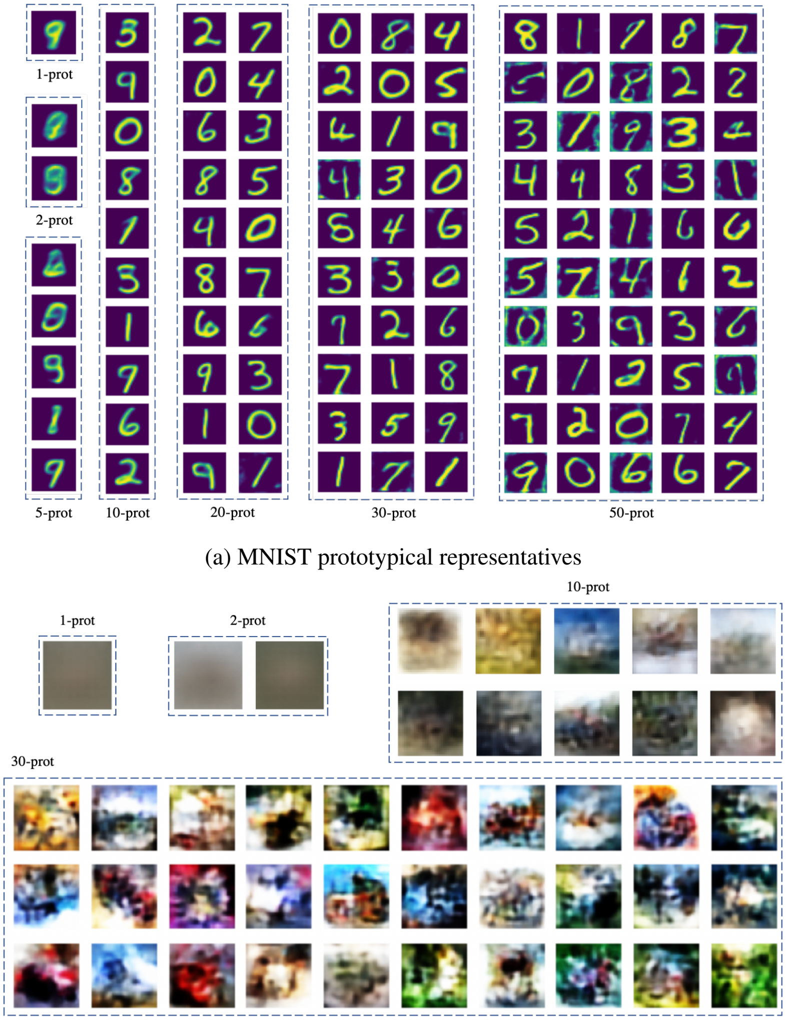
$$\mathcal{L}_{MMD}(p, q) = \mathbb{E}_{p(z),p(z')}[w(z, z')] + \mathbb{E}_{q(z),q(z')}[w(z, z')] - 2\mathbb{E}_{p(z),q(z')}[w(z, z')] \tag{4}$$

where $w(z, z')$ is any universal kernel (in this work we used the Gaussian kernel, such that $w(z, z') = e^{-\frac{\|z - z'\|^2}{2\sigma^2}}$).

Our paper proposes implementing VAE regularization using prototypical online clustering, where each cluster follows a known, normal distribution. We propose using two auxiliary losses 1b, one for finding optimal prototype coordinates, and the other constraining the cluster distribution around the prototype to follow a known distribution, using MMD. Our regularization procedure finds optimal prototype coordinates, assigns samples to the best match prototype using optimal transport, and makes each cluster follow a normal distribution using the MMD loss. In the following, we detail each of those steps and describe the overall optimization schema, including the reconstruction loss.

2.2 Constraining prototype-anchored clusters distribution

Let K be the number of trainable clusters/prototypes $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_k]$, and $\mathbf{v}_k \in \mathbb{R}^d$ the prototype coordinate or cluster centroid associated with the $k - th$ prototype. To implement the local regularization term given a prototype, we propose a slight modification of MMD loss that considers $q(z)$ to follow a normal distribution with the mean value centered at the $k - th$ prototype with coordinates



(a) MNIST prototypical representatives

(b) CIFAR-10 prototypical representatives

Figure 2: Exploring interpretability using synthesis from prototypes coordinates. For MNIST it is possible to observe that prototype coordinates usually reconstruct to known digits, especially with prototypes larger than 10, as we have 10 digits represented in the dataset. For CIFAR-10, one can notice that the learned clusters are gathered around role models that depict associations between foreground and background patterns, which become more detailed with the increase of the number of prototypes available for anchoring clusters.

$\mathbf{v}_k$. Such regularization term follows the standard MMD regularization implementation but considers only samples x_i assigned to $\mathbf{v}_k$ for each prototype.

Since the regularization loss must account for all $\mathbf{V}$ prototypes, formally, we define it as:

$$\mathcal{L}_{reg} = \sum_{k=1}^{K} \sum_{i} \mathcal{L}_{MMD}(p_\theta(z|x_i^k), N(\mathbf{v}_k, 1)) \tag{5}$$

where $\mathbf{v}_k$ are the latent space coordinates of prototype $\mathbf{V}_k$, x_i^k are the samples assigned to prototype $\mathbf{v}_k$, and $N(\mathbf{v}_k, 1)$ is a normal distribution array with means equal to $\mathbf{v}_k$ and standard deviations equal to 1.

2.3 Finding optimal prototypes' coordinates in the latent space

To run the prototype-anchored regularization schema, we need to establish means for computing optimal prototype coordinates and assigning samples to prototypes. We use a similar approach defined in Caron et al. (2020b), which relies on optimal transport to tune the prototypes by maximizing the coherence between OT outcome and prototypes layer softmax output.

Optimal Transport Optimal transport can be interpreted as the search for the optimal plan that transport a probability vector r onto another probability vector c, which are both presented in the simplex denoted by $\sum_d := \{o \in \mathbb{R}_+^d : o^T\mathbf{1}_d = 1\}$ (Cuturi (2013)), where $\mathbf{1}_d$ is a d-dimensional vector with all elements equal to one. The optimal coupling/transportation plan, can be seen as the joint probability distribution between r and c. Using the notation in Cuturi (2013), lets consider $U(r,c) \in \mathbb{R}^{d\times d}$ as the space of probability distribution with marginals r and c, also know as the transportation polytope:

$$U(r,c) := \{P \in \mathbb{R}_+^{d\times d} | P\mathbf{1}_d = r, P^T\mathbf{1}_d = c\} \tag{6}$$

For two distributions $\mathbf{o}_1$ and $\mathbf{o}_2$ drawn for r and c respectively, any $P \in U(r,c)$ is the joint probability matrix of $(\mathbf{o}_1, \mathbf{o}_2)$. With this notation, the entropy h of the joint probability $P \in U(r,c)$ and the marginal $r \in \sum_d$ can be expressed as

$$h(r) = -\sum_{i=1}^{d} r_i \log r_i, \quad h(P) = -\sum_{i,j=1}^{d} p_{i,j} \log p_{i,j}. \tag{7}$$

Then, the optimal transport solution is the matrix $P \in U(r,c)$ that minimize the following equation

$$d_M(r,c) := \min_{P\in U(r,c)} \langle P, M\rangle, \tag{8}$$

where $\langle ., . \rangle$ is the Frobenius dot-product, $M \in \mathbb{R}^{d\times d}$ is the cost matrix that represents the pairwise cost of transporting bin r_i to bin c_i, and $d_M(r,c)$ is a distance between r and c (Cuturi (2013)).

To avoid undesirable sparse solutions, Cuturi (2013) proposed an entropic regularization term that smooths the prediction and allows for an efficient solver using the Sinkhorn-Knopp algorithm. The entropic function of the joint probability matrix $h(P)$ is strongly concave and subject to $h(P) \leq h(r) + h(c) = h(rc^T)$. Hence, the authors use -$h(P)$ as a regularization function to obtain an approximate solution as follows

$$d_M^{\varepsilon}(r,c) := \min_{P\in U(r,c)} \langle P, M\rangle - \varepsilon h(P), \tag{9}$$

where ε is a trade-off parameter that controls the smoothness of the prediction. For more details about entropic regularization of OT we refer the reader to Cuturi (2013); Peyré et al. (2019).

Learning the prototype coordinates Using the entropic regularized OT, finding optimal prototypes boils down to minimize the cross-entropy between the softmax of the prototypes assignments $\mathbf{p}$ and the codes/assignment $\mathbf{q}$ obtained using the OT algorithm:

$$\mathcal{L}_{prot} = -\sum_k \mathbf{q}^k \log \mathbf{p}^k, \quad where \quad \mathbf{p}^k = \frac{\exp(\frac{1}{\tau}(\mathbf{z}^\mathsf{T}\mathbf{v}_k)}{\sum_{k'} \exp(\frac{1}{\tau}(\mathbf{z}^\mathsf{T}\mathbf{v}_{k'})}, \tag{10}$$

where the prototype assignments are the dot product between the feature vector $\mathbf{z}$ and each prototype $\mathbf{v}_k$. Using the notation of Caron et al. (2020a), given B features in a mini-batch, we compute the codes $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_B]$ in order to maximize the similarity between the set of features $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_B]$ and the prototypes $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]$ as follows

$$\min_{\mathbf{Q}\in\mathcal{Q}} \mathbf{Tr}(\mathbf{Q}^\mathsf{T}\mathbf{V}^\mathsf{T}\mathbf{Z}) + \varepsilon h(\mathbf{Q}). \tag{11}$$

Similar to Caron et al. (2020a); YM. et al. (2020) we also impose an equipartition constraint to ensure that the cluster assignments partition the data in groups of equal size. Hence, the space of the OT solutions reads:

Algorithm 1 Prototypical Variational Autoencoders training loop

```
Input: batch of samples $\mathcal{D} = \{\mathcal{B}_i\}_{i=1}^{N}$
Input: $\varepsilon > 0$, epochs
Input: warm-up=10
Initialize: encoder $e_\theta$, decoder $d_\theta$, and prototypes $\mathbf{V}$ with random weights
for i = 1 to epochs do
    for each $\mathcal{B}_i$ in $\mathcal{D}$ do
        if i > warm-up then
            Obtain the feature vectors $\mathbf{Z}_i$
            Compute the prototype scores $\mathbf{V}^\mathsf{T}\mathbf{Z}_i$
            Compute the codes $\mathbf{Q}_i$ through Sinkhorn constrained to equipartition
            Convert prototype scores to probabilities $\mathbf{P}_i$
            Compute prototype loss $\mathcal{L}_{prot}$ using $\mathbf{P}_i$ and $\mathbf{Q}_i$
            Update $e_\theta$ weights and $\mathbf{V}$ with a gradient step using $\mathcal{L}_{prot}$
            Compute MMD regularization loss $\mathcal{L}_{reg}$
            Update $e_\theta$ weights with a gradient step using $\mathcal{L}_{reg}$
        end if
        Obtain the feature vectors $\mathbf{Z}_i$
        Decode the features vectors
        Compute reconstruction loss $\mathcal{L}_{rec}$
        Update $\theta$ with a gradient step using $\mathcal{L}_{rec}$
    end for
end for
```

$$\mathcal{Q} = \{\mathbf{Q} \in \mathbb{R}_+^{K\times B} | \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^T\mathbf{1}_K = \frac{1}{B}\mathbf{1}_B\}. \tag{12}$$

This formulation is equivalent to 9 and can be efficiently solved using iterative matrix multiplication using the Sinkhorn-Knopp algorithm (Cuturi (2013)).

2.4 Optimizing an autoencoder with prototypical latent space regularization

For running the autoencoder optimization, we implemented three sequential gradient optimization procedures at each training step, as shown in the pseudo-code Algorithm 1. The first optimizes the encoder and prototype weights using $\mathcal{L}_{prot}$ and targets to find optimal prototype coordinates using the OT lead. The second optimizes the encoder weights using $\mathcal{L}_{reg}$ to make the samples assigned to each prototype follows a normal distribution centered at the fixed prototype coordinates. The third optimizes the encoder and decoder weights using a binary cross-entropy reconstruction loss ($\mathcal{L}_{rec}$).

We also implement a warm-up training schema, where we train the autoencoder using only the reconstruction loss, so we have reasonable embedding to introduce the regularization. The same warm-up configuration is used for VAE-KL and InfoVAE models to enable a fair performance comparison.

3 Experiments

The scope of our contribution is explicitly related to functional regularization of autoencoders' latent space, and therefore we compare our proposal to other two variational autoencoders, specifically those using KL (Kingma & Welling (2014)) and MMD (Zhao et al. (2017)) divergence regularizations, and also to a baseline non-regularized autoencoder.

3.1 Datasets and experimental setting

Our experiments used the MNIST, CIFAR-10, and CIFAR-100 public benchmarks, released under the Creative Commons Attribution-Share Alike 3.0 and MIT licenses, respectively. We evaluated the latent space organization characteristics and the impact on the reconstruction quality.

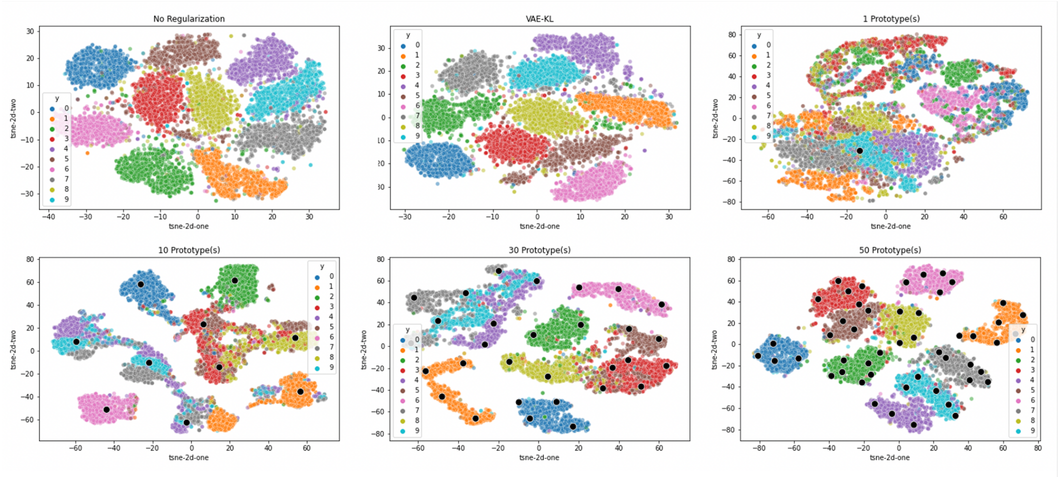


Figure 3: t-SNE plots considering a non-regularized autoencoder, a variational autoencoder with KL divergence loss, and our models with 1, 10, 30 and 50 prototypes for the MNIST dataset. One can observe that increasing the number of prototypes implement disambiguation of digits in the latent space.

The encoder architecture consists of three (two for MNIST experiments) convolution blocks and a bottleneck dense layer. We employed two down-sampling stages (one for each convolutional block) to reduce the spatial input dimension by four before submitting the outcome to the bottleneck dense layer. After the convolutional and dense layers, we also applied ReLU activation functions. The decoder receives an input array z with the size of the latent space dimension that is ingested to a dense layer to be reshaped into 256 activation maps of size 4×4 (7×7 for MNIST). These maps serve as input to consecutive transposed convolution layers that up-sample the data to the original size. A final convolution using three filters (one for MNIST) is applied to deliver the final outcome.

For the baseline variational autoencoder models, we implemented two standard dense layers for optimizing μ_x and σ_x that hold the standard normal distribution parameters from which the sampling function derives z. For implementing our prototypical VAE model, we replaced this module with a single dense layer with 256 nodes that serves as a projection head. At inference, the dimension of the projection head is $B \times 256$, which represents a 256-value embedding for each sample in the batch to be consumed by the decoder. The prototype dense layer receives this embedding and outputs a $B \times K$ tensor, which is the activation for each sample in the batch considering each prototype. The weights in the prototype layers, with a dimension of $256 \times K$ also represent the K prototypes coordinates.

We used the Adam optimizer for optimizing all trained models with a learning rate of 0.001, beta1 as 0.9, and beta2 as 0.999. We trained the models for 600 epochs for CIFAR-10 and 100 epochs for MNIST, with 256 data samples per batch, sufficient to achieve convergence. We set initial seeds to Tensorflow environment and NumPy library to allow a fairer comparison between different trained models and enable our tests' reproducibility. All experiments were carried out using V100 GPUs.

3.2 Exploring latent data prototypical clusters

We used the MNIST dataset for exploring our model skill for clustering the latent space around prototypes. Considering that MNIST data comprises a straightforward representation of digits ranging from 0 to 9, we expected the network to optimize clusters around the digit shapes. Therefore, as cluster centroids, the prototypes should be decoded into a faithful representation of digits if the number of prototypes is sufficient to discriminate unique digit characteristics, which might be good for model interpretability uses. Figure 2a depicts the decoded prototypes for different configurations of number of prototypes K. It is possible to observe that for $K = 1, 2, 5$, the number of prototypes used is insufficient to deliver single-digit homogeneous clusters, as we have ten classes in this dataset. With $\mathcal{V} = 10$, we observe that most digits have an associated cluster, but digits 4 and 5 seem not to be represented by their own clusters. That impression is also supported by Figure 3. When $K = 20$

it seems the model delivers at least one cluster for each digit, even if some clusters seem to group ambiguous samples. Observing Figures 2a and the t-SNE (Van der Maaten & Hinton (2008)) plots (3), it is possible to visually conclude that 30 prototypes deliver a fair clustering, and 50 prototypes can separate the digits unequivocally in the latent space.

Considering the CIFAR-10 prototypes decoded information, one can visually inspect interesting outcomes that depict background/foreground relationships and shed some light on which criteria the model uses for gathering samples in the latent space. When we use only one prototype, the corresponding prototype decoded image is brownish, almost uniform. With two prototypes, the method seems to cluster the samples around a light background and dark foreground role model cluster and another one with the exact opposite. We observe diverse, interesting patterns with ten prototypes, some dedicated to samples with blue-sky background, others with a grass-like background. For 30-prototypes, the patterns are more intricate, and we observe that prototypes begin to detail for the background/foreground similarity criteria optimized in the training process.

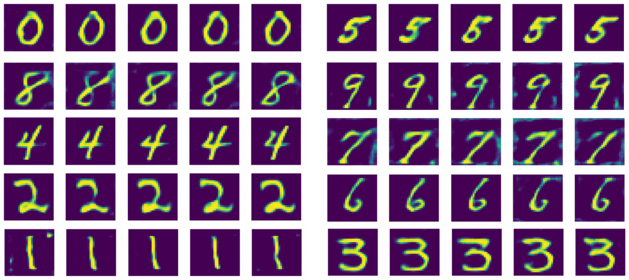


Figure 4: Stochastic synthesis examples considering 10 prototypes manually chosen corresponding to the 10 digits from our 30-prototype trained model.

For exploring controllable stochastic synthesis outgrowths, we selected ten prototypes from our 30-prototype trained model, one for each digit, and output stochastic synthesized samples generated by simply adding white noise to the respective prototype coordinate. As observed in Figure 4, by simply adding noise to selected prototype coordinates and feeding that to the decoder, we can synthesize stable, high-quality samples for the same digit. As depicted in Figure 2a, since the 10-prototype model was not able to cluster every digit, we decided to select 10 different digits from the 30-prototype model manually.

3.3 Exploring regularization effects on the reconstruction quality

We evaluated the impact on reconstruction quality using the CIFAR datasets to investigate if our regularization term could alleviate the reconstruction quality burden usually observed in autoencoders with regularized latent spaces. One can visually inspect in Figure 5 that with a proper amount of prototypes, our method can deliver reconstructions with quality comparable to non-regularized autoencoders, which means that it manages to neutralize adverse blurring effects caused by the regularization. It is also possible to observe that VAE-ML and InfoVAE counterparts deliver considerably more blurred outcomes and that configurations with 1 and 2 prototypes are insufficient to revert the regularization effects completely, as expected, but 10 and 30 prototypes seem to be sufficient for that.

We also evaluated the reconstruction quality during the training process using the mean-squared error considering training and testing samples' reconstruction performance. Figure 6 show that our method can match non-regularized autoencoder reconstruction performance with ten prototypes for both CIFAR-10 and CIFAR-100 datasets. One can also visually inspect that the result with one prototype is comparable to VAE-KL and InfoVAE as expected, and a significant improvement is already observed using two clusters. Increasing the number of prototypes beyond 10 does not increase reconstruction quality performance for CIFAR-10 and CIFAR-100 datasets.

4 Conclusion

This paper proposed an efficient regularization approach for variational autoencoders using prototype-anchored latent space clustering, where each cluster follows a known distribution. Our approach

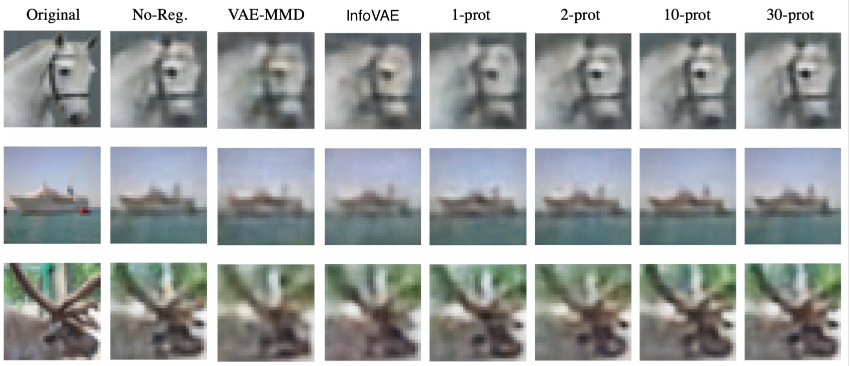
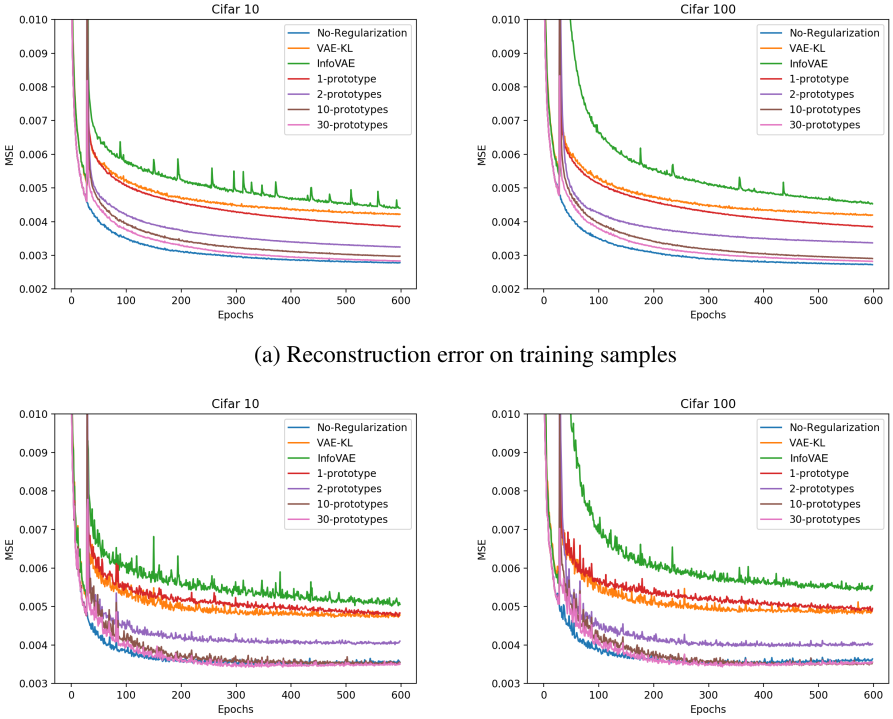


Figure 5: Reconstructed CIFAR-10 samples for different models.



(a) Reconstruction error on training samples

(b) Reconstruction error on testing samples

Figure 6: Reconstruction mean-squared error evolution for training and testing sets during training. One can observe that at converge, with a proper number of prototypes, our model matches non-regularized autoencoders' performance.

enables obtaining an appropriate number of clusters with solid regularization for better reconstruction quality and improved synthesis control. We also explored prototype decoding to understand better the similarity criteria used for gathering samples in the latent space and observed an increased control for stochastic synthesis using prototypes as reference landmarks for creating samples with anticipated characteristics. We reported our results using public benchmarks and showed our model implements an effective organization of the latent space that alleviates the adverse effects of regularization on autoencoder reconstruction quality, matching non-regularized autoencoders' performance.

References

Michael Arbel, Anna Korba, Adil SALIM, and Arthur Gretton. Maximum mean discrepancy gradient flow. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/944a5ae3483ed5c1e10bbccb7942a279-Paper.pdf.

Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models, 2021.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020a.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters*, 138:185–192, 2020.

Aude Genevay, Gabriel Dulac-Arnold, and Jean-Philippe Vert. Differentiable deep clustering with cluster size constraints. Technical report, arXiv, 2019. URL https://arxiv.org/abs/1910.09036. 1910.09036.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 881–889, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/germain15.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Philip A. Knight. The sinkhorn-knopp algorithm: Convergence and applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, March 2008. ISSN 0895-4798. doi: 10.1137/060659624. URL https://doi.org/10.1137/060659624.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *University of Toronto*, 2012. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, number 2 in Proceedings of Machine Learning Research, pp. 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 478–487, 2016.

Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyx-jyBFPr.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017. URL http://arxiv.org/abs/1706.02262.