Towards Learning Better and Flexible Representations from Variational Autoencoders

Advait, Pulkit, Sanchit, Lakshya

IIT Bombay

6th May 2024

Outline

Towards **Learning** Better and Flexible **Representations** from Variational Autoencoders

- Representation Learning is concerned with learning representations of the data that make it easier to extract useful information.
- The learned representations can then be used to perform multiple downstream tasks such as classification or retrieval.

Towards **Learning** Better and Flexible **Representations** from Variational Autoencoders

- Representation Learning is concerned with learning representations of the data that make it easier to extract useful information.
- The learned representations can then be used to perform multiple downstream tasks such as classification or retrieval.

Towards **Learning** Better and Flexible **Representations** from Variational Autoencoders

- Representation Learning is concerned with learning representations of the data that make it easier to extract useful information.
- The learned representations can then be used to perform multiple downstream tasks such as classification or retrieval.

Towards **Learning** Better and Flexible **Representations** from Variational Autoencoders

- Representation Learning is concerned with learning representations of the data that make it easier to extract useful information.
- The learned representations can then be used to perform multiple downstream tasks such as classification or retrieval.

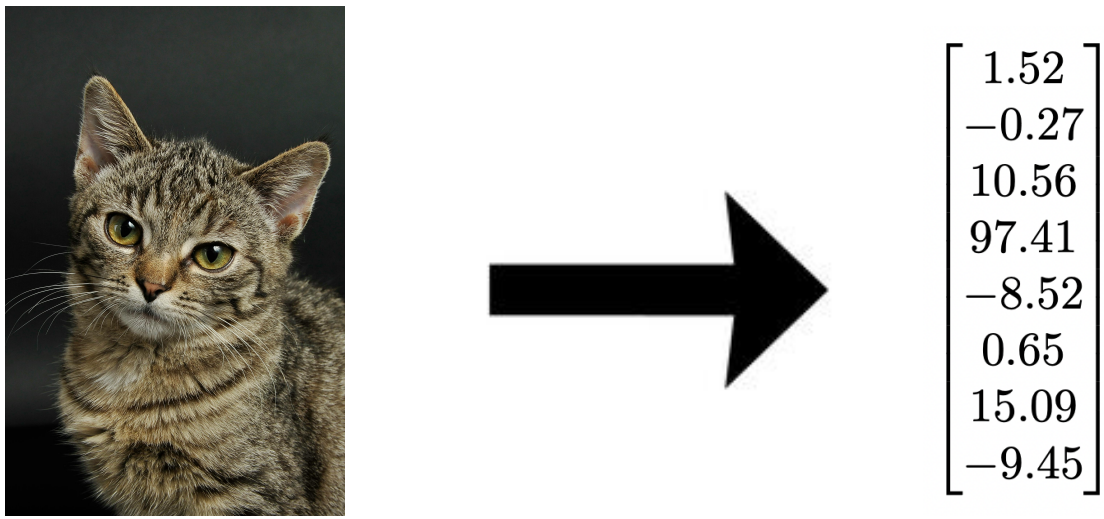


Figure 1: Representating the image of cat as 8-dimensional vector

- An ideal representation should be expressive, meaning that a reasonably-sized learned representation should capture a huge number of possible input configuration.
- Moreover, we expect the process of learning representations to not be computationally expensive.
- Some examples of models used for representation learning :-
  1. Convolutional Neural Networks for Image Representations
  2. Recurrent Neural Networks or Transformers for Word Representations
- In this project, we focus our attention on learning representations of images using **Variational Autoencoders**

Towards Learning Better and Flexible Representations from **Variational Autoencoders**

- Variational autoencoders (VAEs) are a deep learning technique for learning latent representations.
- VAEs consist of an encoder network which models the posterior distribution $q_\phi(z|x)$ and a decoder network which models the likelihood distribution $p_\theta(x|z)$
- VAEs have been used for several applications such as dimensionality reduction and image generation.

Towards Learning Better and Flexible Representations from **Variational Autoencoders**

- Variational autoencoders (VAEs) are a deep learning technique for learning latent representations.
- VAEs consist of an encoder network which models the posterior distribution $q_\phi(z|x)$ and a decoder network which models the likelihood distribution $p_\theta(x|z)$
- VAEs have been used for several applications such as dimensionality reduction and image generation.

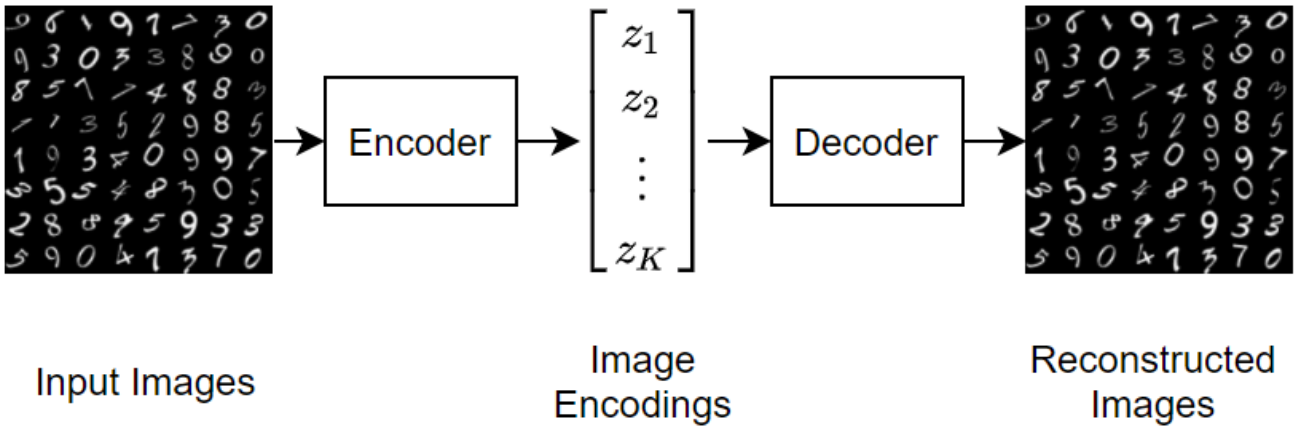


Figure 2: Variational Autoencoder Architecture

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

- After training the VAE on the ELBO objective, the learned $q_\phi(z|x)$ can be used to generate the representation.
- For a given data x', we can obtain a representation by :-
  1. Sampling from the posterior - $z' \sim q_\phi(z|x')$
  2. Finding the most likely representation (MAP) - $z' = \text{argmax}\ q_\phi(z|x')$
- For this project, we used 3 popular image datasets :-

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

- After training the VAE on the ELBO objective, the learned $q_\phi(z|x)$ can be used to generate the representation.
- For a given data x', we can obtain a representation by :-
  1. Sampling from the posterior - $z' \sim q_\phi(z|x')$
  2. Finding the most likely representation (MAP) - $z' = \text{argmax}\ q_\phi(z|x')$
- For this project, we used 3 popular image datasets :-

(a) MNIST Dataset

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

- After training the VAE on the ELBO objective, the learned $q_\phi(z|x)$ can be used to generate the representation.
- For a given data x', we can obtain a representation by :-
  1. Sampling from the posterior - $z' \sim q_\phi(z|x')$
  2. Finding the most likely representation (MAP) - $z' = \text{argmax}\ q_\phi(z|x')$
- For this project, we used 3 popular image datasets :-

(a) MNIST Dataset

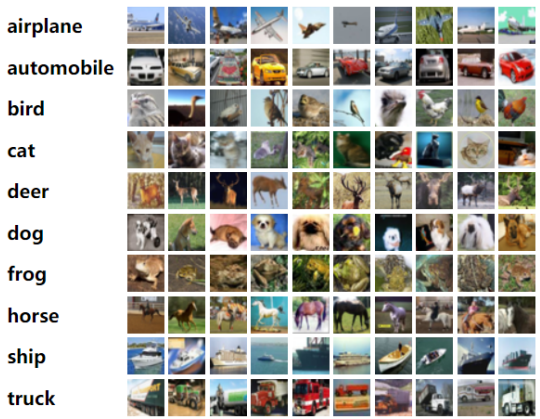
(b) FashionMNIST

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

- After training the VAE on the ELBO objective, the learned $q_\phi(z|x)$ can be used to generate the representation.
- For a given data x', we can obtain a representation by :-
  1. Sampling from the posterior - $z' \sim q_\phi(z|x')$
  2. Finding the most likely representation (MAP) - $z' = \text{argmax}\ q_\phi(z|x')$
- For this project, we used 3 popular image datasets :-

(a) MNIST Dataset

(b) FashionMNIST



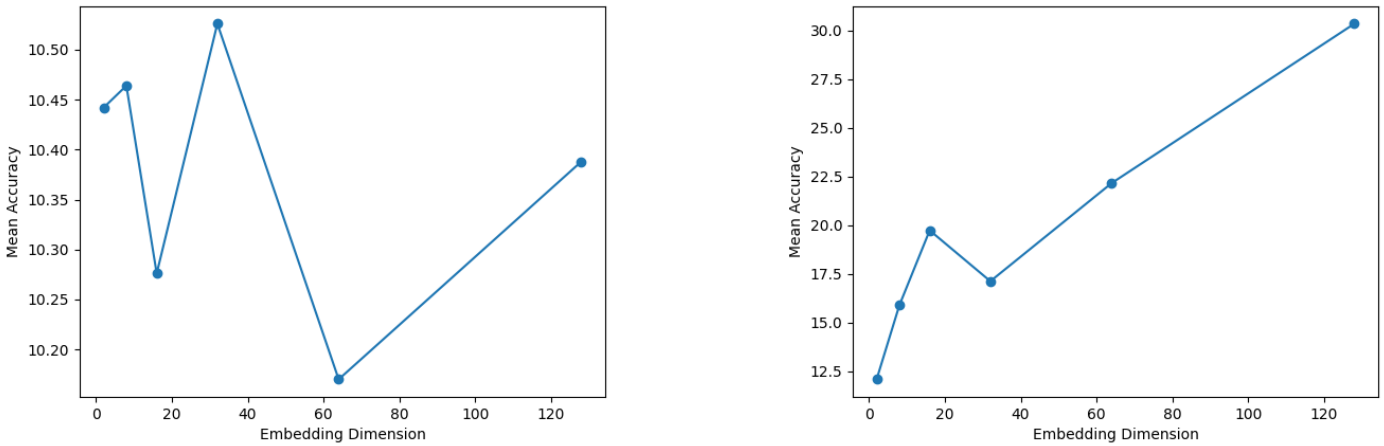
(c) CIFAR10 Dataset

Towards Learning Better and Flexible Representations from Variational Autoencoders

- For the downstream classification task, we used a feedforward Network consisting of 2 linear layers.
- The encoder and decoder networks of VAE consist of 2 convolutional layers and 2 linear layers.
- The classifier and VAE have been implemented in PyTorch and we have used Adam Optimizer for training the models.
- We varied the size of latent representations as [2, 8, 16, 32, 64, 128].
- VAE was trained for 10 epochs with a batch size of 32 meanwhile the classifier was trained for 20 epochs.
- We implemented an early stopping module to prevent overfitting.
- For evaluation, we have employed 5-fold cross validation.

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

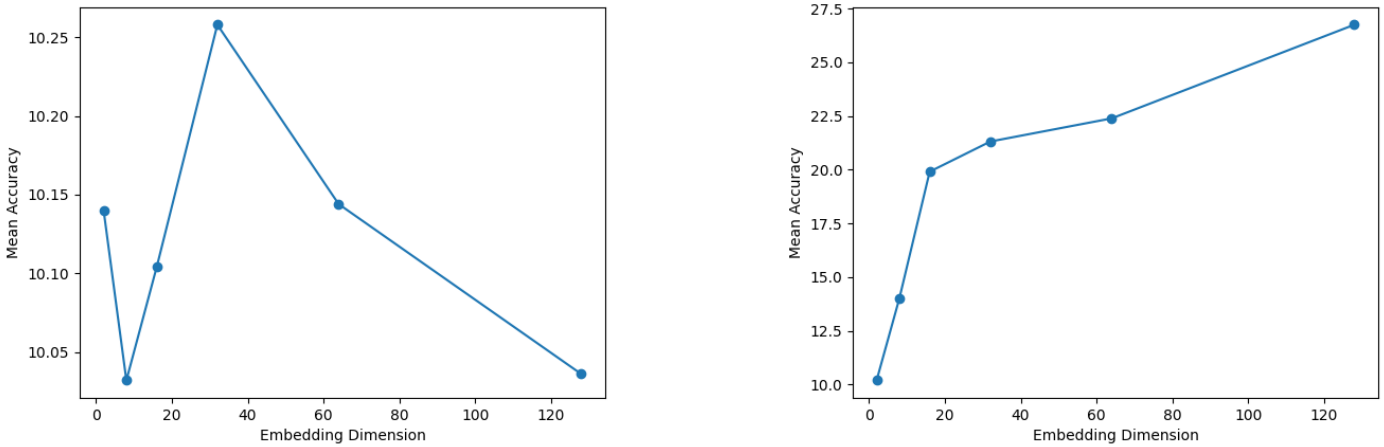
1. MNIST Dataset
   - For Posterior sampling, across all dimensions, the classification accuracy varies around 10 which is comparable to random labeling.
   - For Maximum a posteriori, the classification accuracy increases as we increase the dimensionality of the latent space. However, the best classification accuracy is 30.33 which is low compared to SOTA.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 4: Classification Accuracy of VAE on MNIST Dataset

2. FashionMNIST Dataset
   - Similar to MNIST Dataset, the classification accuracy for posterior sampling varies around 10 which is comparable to random labeling.
   - For Maximum a posteriori, the classification accuracy increases as we increase the dimensionality of the latent space. However, the best classification accuracy is 26.74 which is low compared to SOTA.

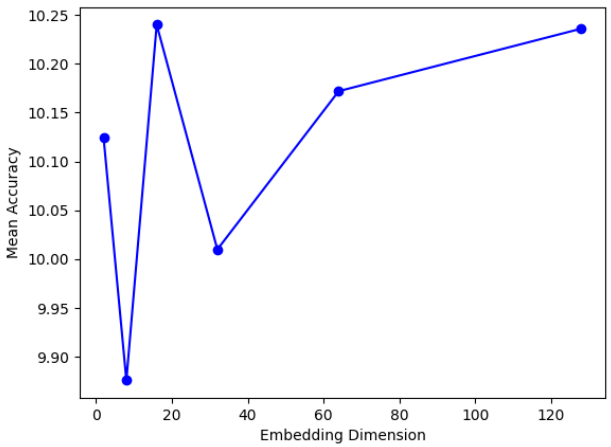


(a) Posterior Sampling (b) Maximum a posteriori

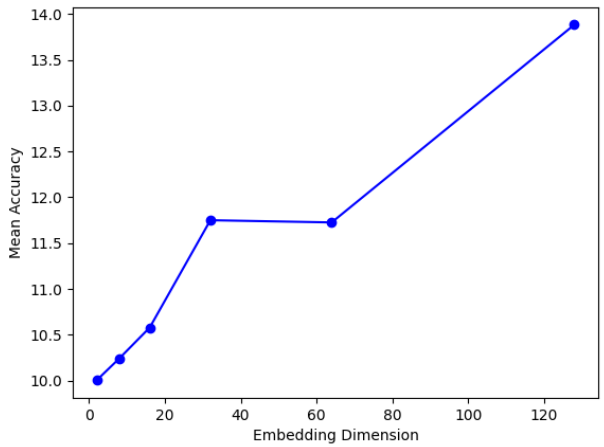
Figure 5: Classification Accuracy of VAE on FashionMNIST Dataset

3. CIFAR10 Dataset

- The classification accuracy for posterior sampling varies around 10 which is comparable to random labeling.
- For Maximum a posteriori, the classification accuracy increases as we increase the dimensionality of the latent space. However, the best classification accuracy is 13.88 which is low compared to SOTA.



(a) Posterior Sampling

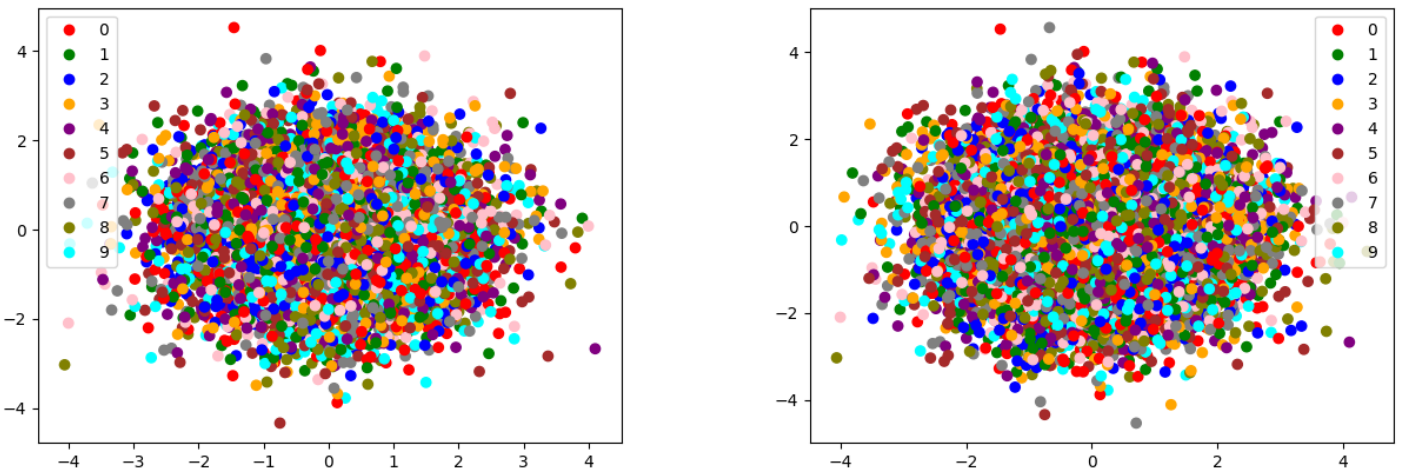


(b) Maximum a posteriori

Figure 6: Classification Accuracy of VAE on CIFAR10 Dataset

Towards Learning Better and Flexible Representations from Variational Autoencoders

- In order to understand why the classification accuracy was so low, we plotted the latent space representations.
- It was observed that input data belonging to the different classes have similar representations in latent space.

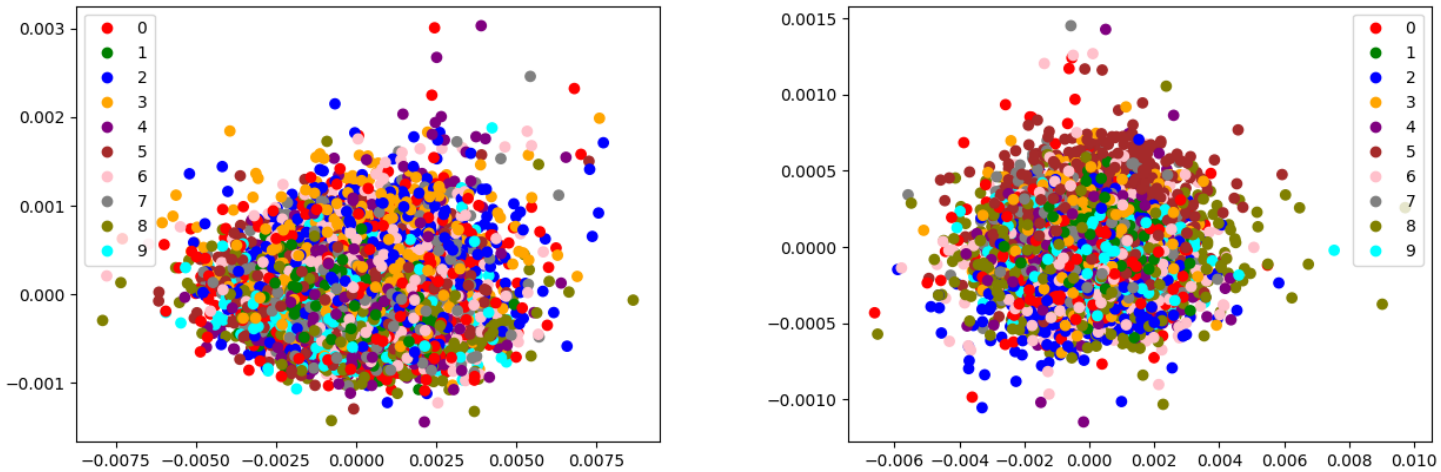


(a) MNIST Dataset (b) FashionMNIST Dataset

Figure 7: Latent Space of VAE for Posterior Sampling

Towards **Learning** Better and Flexible **Representations from Variational Autoencoders**

- Due to similar representations, the classifier wasn't able to distinguish between two classes thus leading to poor classification.
- For higher dimensional latent representations, we used Principal Component Analysis (PCA) to project down the representations to 2-dimensional space for plotting the latent space.



(a) MNIST Dataset (b) FashionMNIST Dataset

Figure 8: Latent Space of VAE for Maximum a posteriori

Towards **Learning Better** and Flexible **Representations from Variational Autoencoders**

- Our next objective was to improve these latent space representations for better performance in downstream classification task.
- For this, we referred to Zhao et. al., 2017 which proposed a new family of VAEs called InfoVAE.
- The paper discusses about two major problems of VAE :-
  1. The approximate inference distribution is often significantly different from the true posterior.
  2. When the conditional distribution is sufficiently expressive, the latent variables are often ignored.
- The paper argues that the above two problems arise due to ELBO objective used to train VAEs and proposes a new loss function.
- The following tutorial serves as a good start for understanding InfoVAE.

Towards **Learning Better** and Flexible **Representations from Variational Autoencoders**

- In particular, we used Maximum Mean Discrepancy (MMD) instead of KL Divergence to quantify the distance between two distributions.
- Maximum Mean Discrepancy can be efficiently implemented using the kernel trick. Let $k(.\,,.)$ be any positive definite kernel then,

$$\begin{aligned}\mathcal{L}_{\text{MMD}}(q\,||\,p) = &\mathbb{E}_{p(z),\,p(z')}[k(z,\,z')] \\ &+ \mathbb{E}_{q(z),\,q(z')}[k(z,\,z')] \\ &- 2\mathbb{E}_{q(z),\,p(z')}[k(z,\,z')]\end{aligned}$$

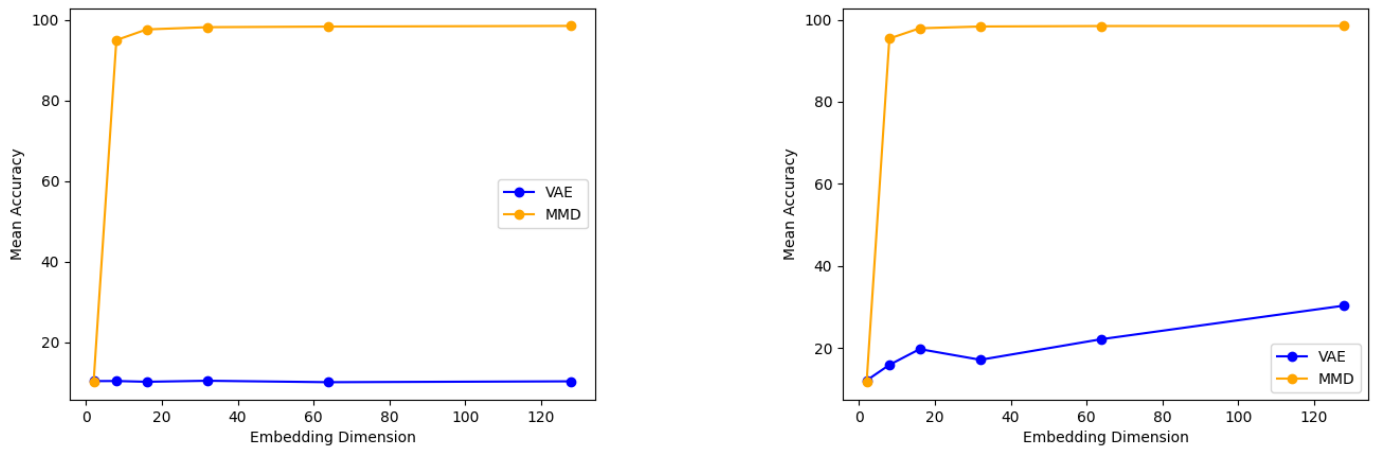
- In our project, we utilized Gaussian Kernel with $\sigma = 1$ to implement Maximum Mean Discrepancy.

$$k(z,\,z') = e^{-\frac{||\,z - z'\,||^2}{2\sigma^2}}$$

- For computing the MMD distance, we first simply generated n samples from the prior distribution $p(z)$ and compared these generated samples with the encoder output.
- For training InfoVAE, we used the same hyperparameters as that of standard VAE. The additional hyperparameter n was set to 200.
- We also implemented an early stopping module to prevent overfitting and employed 5-fold cross validation for evaluation.
- We used the following Github repository for reference while implementing our model.

Towards **Learning Better** and Flexible **Representations from Variational Autoencoders**

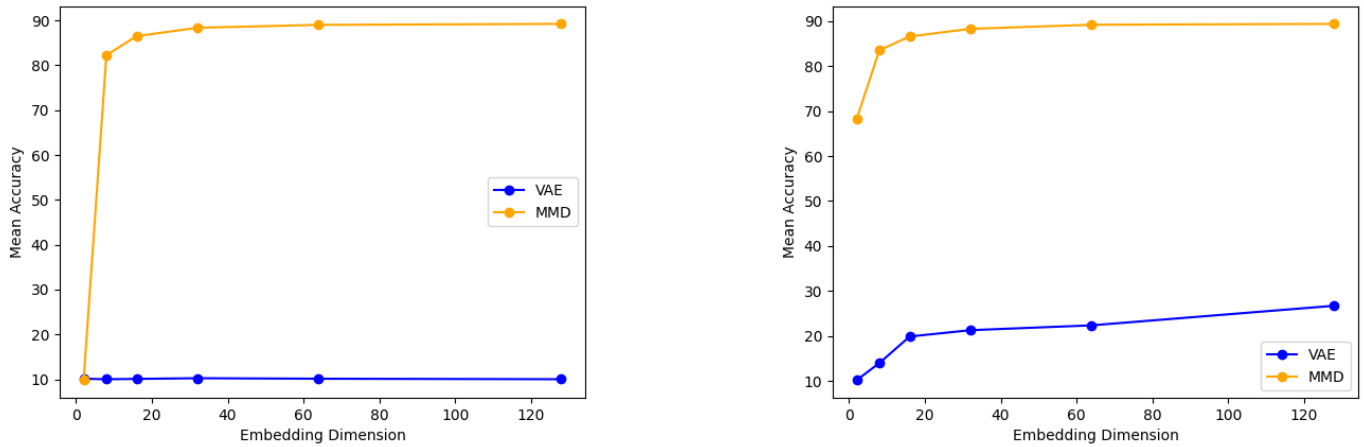
1. MNIST Dataset
   - Barring the 2-dimensional case, the classification accuracy for posterior sampling approaches 98% which is comparable to SOTA.
   - The classification accuracy for Maximum a posteriori follows a similar pattern as that of Posterior Sampling.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 9: Classification Accuracy of InfoVAE on MNIST Dataset

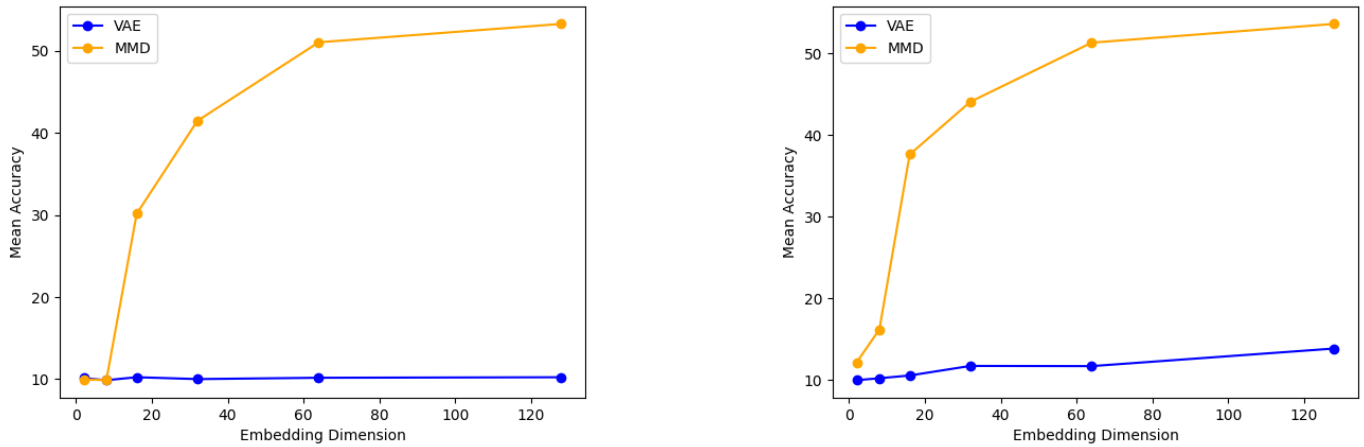
2. FashionMNIST Dataset
   - Similar to MNIST Dataset, classification accuracy for posterior sampling approaches 90%.
   - The classification accuracy for Maximum a posteriori follows a similar pattern as that of Posterior Sampling.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 10: Classification Accuracy of InfoVAE on FashionMNIST Dataset

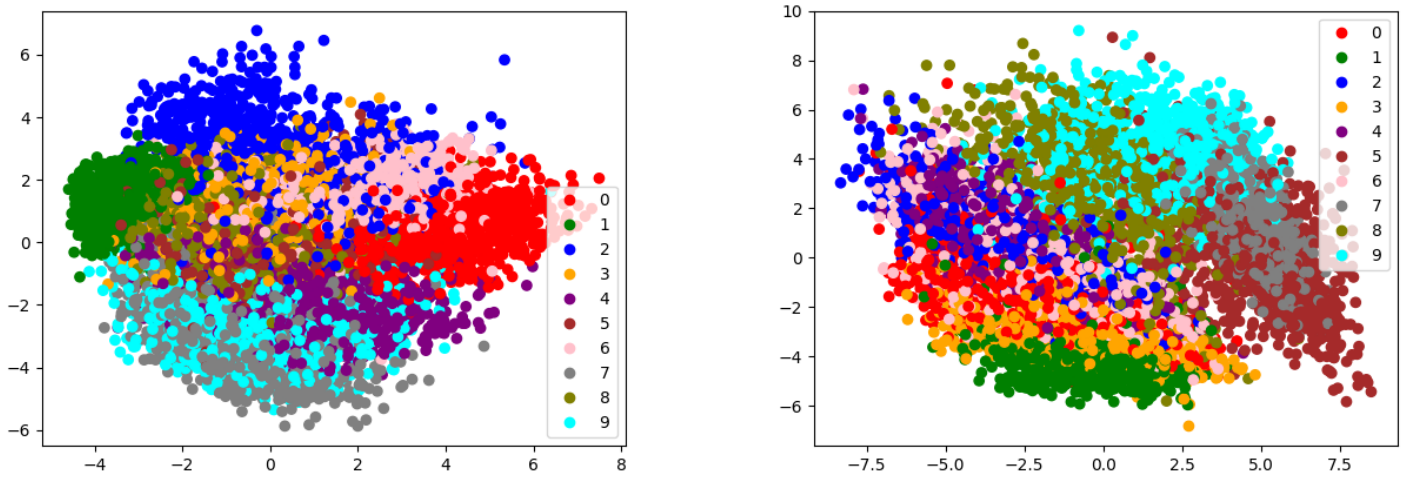
2. CIFAR10 Dataset
   - Similar to MNIST Dataset, classification accuracy for posterior sampling approaches 53%.
   - The classification accuracy for Maximum a posteriori follows a similar pattern as that of Posterior Sampling.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 11: Classification Accuracy of InfoVAE on CIFAR10 Dataset

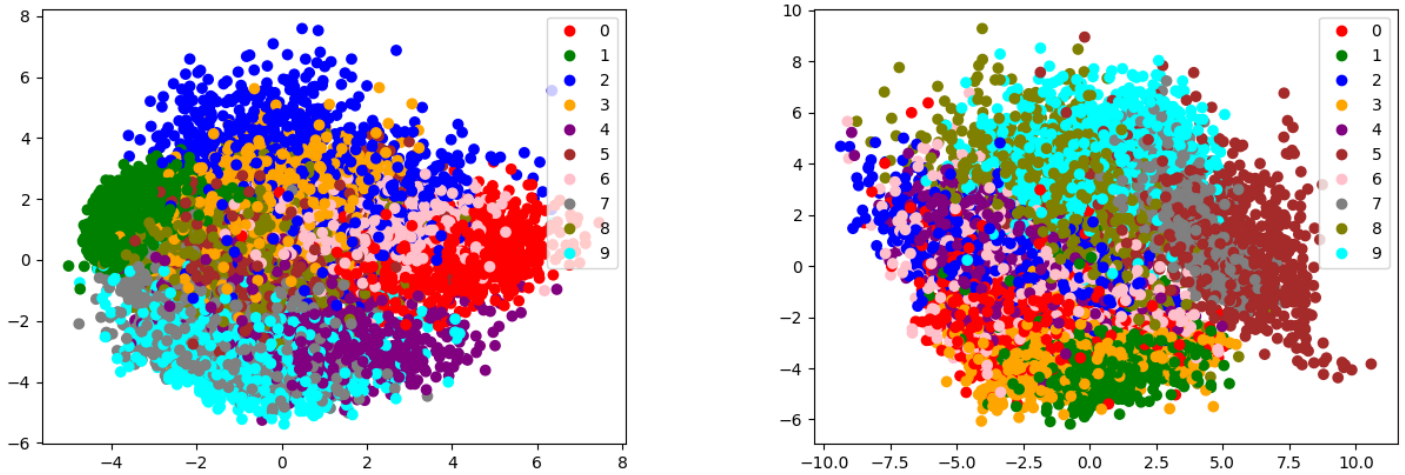
- After plotting the latent space of InfoVAE, we observed the formation of distinct clusters in the latent space corresponding to each class.
- Due to the formation of distinct clusters in the latent space, we are able to achieve such good performance in the downstream classification task.



(a) MNIST Dataset (b) FashionMNIST

Figure 12: Latent Space of InfoVAE for Posterior Sampling

Towards Learning Better and Flexible Representations from Variational Autoencoders



(a) MNIST Dataset (b) FashionMNIST

Figure 13: Latent Space of InfoVAE for Maximum a posteriori

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- So far, we utilized InfoVAE to improve the representations learnt from Variational Autoencoder architecture.
- However, in practice, we would be generating representations for a large number of images and share them with different clients with varying computational resources.

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- So far, we utilized InfoVAE to improve the representations learnt from Variational Autoencoder architecture.
- However, in practice, we would be generating representations for a large number of images and share them with different clients with varying computational resources.

- So far, we utilized InfoVAE to improve the representations learnt from Variational Autoencoder architecture.
- However, in practice, we would be generating representations for a large number of images and share them with different clients with varying computational resources.

$$\begin{bmatrix} 1.52 & \dots & 3.59 \\ -0.27 & \dots & 6.42 \\ 10.56 & \dots & -0.32 \\ 97.41 & \dots & -5.36 \\ -8.52 & \dots & 63.29 \\ 0.65 & \dots & 3.38 \\ 15.09 & \dots & 5.18 \\ -9.45 & \dots & -7.82 \end{bmatrix}$$

Figure 14: Representating a collection of cat images as 8-dimensional vectors

- Storing numerous such representations would be memory intensive and sharing these representations to other users could also prove to be a difficult and lossy process.
- Moreover, a user in the network may not possess the required compute to use these representations in their downstream tasks.
- In this context, rigid fixed-capacity representations can be either over or under-accommodating to the task at hand.

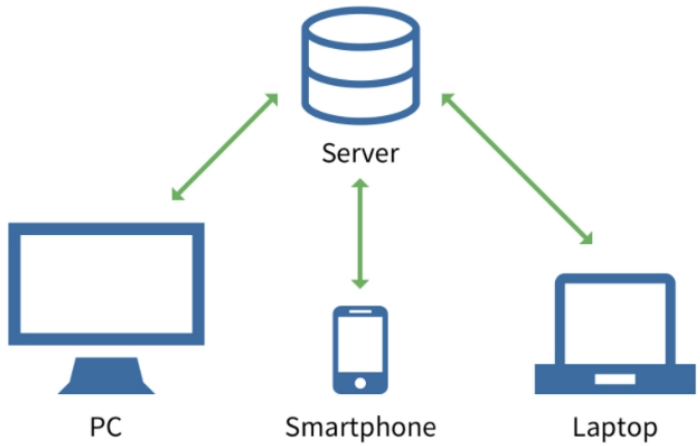


Figure 15: Client-Server Model

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- Our objective in the latter half of the project would be to make the representations learnt from VAEs flexible.
- In order to achieve this goal, we refer to Kusupati et al., 2022 which proposes **Matryokshka Representation Learning**.

- Our objective in the latter half of the project would be to make the representations learnt from VAEs flexible.
- In order to achieve this goal, we refer to Kusupati et al., 2022 which proposes **Matryokshka Representation Learning**.

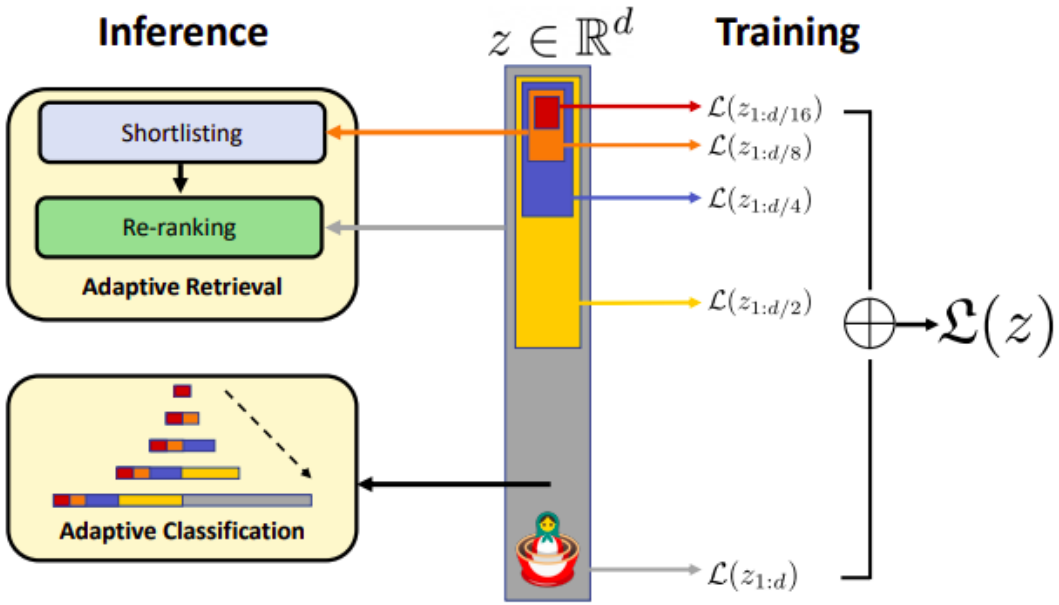


Figure 16: Matryoshka Representation Learning

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- The paper argues that gradient-based training of deep learning models tends to diffuse information across the entire representation vector leading to rigidity.
- To induce flexibility in the learned representation, Matryoshka Representation Learning encodes information at different granularities
- In this way, a single embedding can adapt to the computational constraints of multiple downstream tasks.
- There have been several applications of Matryoshka Representation Learning :-
  1. MatFormer (Kudugunta, Sneha, et al., 2023)
  2. AdANNS (Rege, Aniket, et al., 2024)
- In this project, we intend to implement Matryokshka Representation Learning in the VAE algorithm in order to obtain flexible latent representations.

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- The paper argues that gradient-based training of deep learning models tends to diffuse information across the entire representation vector leading to rigidity.
- To induce flexibility in the learned representation, Matryoshka Representation Learning encodes information at different granularities
- In this way, a single embedding can adapt to the computational constraints of multiple downstream tasks.
- There have been several applications of Matryoshka Representation Learning :-
  1. MatFormer (Kudugunta, Sneha, et al., 2023)
  2. AdANNS (Rege, Aniket, et al., 2024)
- In this project, we intend to implement Matryokshka Representation Learning in the VAE algorithm in order to obtain flexible latent representations.

Towards Learning Better and **Flexible Representations** from Variational Autoencoders

- For $d \in \mathbb{N}$, consider a set $\mathcal{M} \subset [d]$ of representation sizes.
- For a datapoint x in the input domain $\mathcal{X}$, our goal is to learn a d-dimensional flexible representation vector $z \in \mathbb{R}^d$ using a deep neural network $F(\,\cdot\,;\theta_F) : X \to \mathbb{R}^d$ parameterized by weights θ_F
- For every $m \in \mathcal{M}$, Matryoshka Representation Learning (MRL) enables each of the first m dimensions of the embedding vector, $z_{1:m} \in \mathbb{R}^m$ to be independently capable of being a transferable and general purpose representation of the datapoint x.
- Suppose we are given a labelled dataset $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$ where $x_i \in \mathcal{X}$ is an input point and $y_i \in [L]$ is the label of x_i
- MRL optimizes the multi-class classification loss for each of the nested dimension $m \in \mathcal{M}$ using a separate linear classifier, parameterized by $W^{(m)} \in \mathbb{R}^{L \times m}$

$$\min_{\theta_F,\, \{W^{(m)}\}_{m \in \mathcal{M}}} \frac{1}{N} \sum_{i \in [N]} \sum_{m \in \mathcal{M}} \mathcal{L}(W^{(m)} \cdot F(x_i\,;\theta_F)_{1:m}\,;\; y_i)$$

Towards **Learning** Better and **Flexible Representations from Variational Autoencoders**

- We adopt the objective of Matryokshka Representation Learning to the Variational Autoencoder architecture in the following way :-
  1. The Encoder network $q(\cdot\,;\phi)$ is taken as $F(\cdot\,;\theta_F)$.
  2. For every $m \in \mathcal{M}$, we use a separate Decoder network $p(\cdot\,;\theta_m)$
  3. This new VAE architecture is trained upon the following objective :-

$$\min_{\phi,\,\{\theta_m\}_{m\in\mathcal{M}}} \frac{1}{N} \sum_{i\in[N]} \sum_{m\in\mathcal{M}} \mathcal{L}_{\text{vae}}(x_i,\ q(x_i\,;\phi)_{1:m},\ p(q(x_i\,;\phi)_{1:m}\,;\theta_m))$$

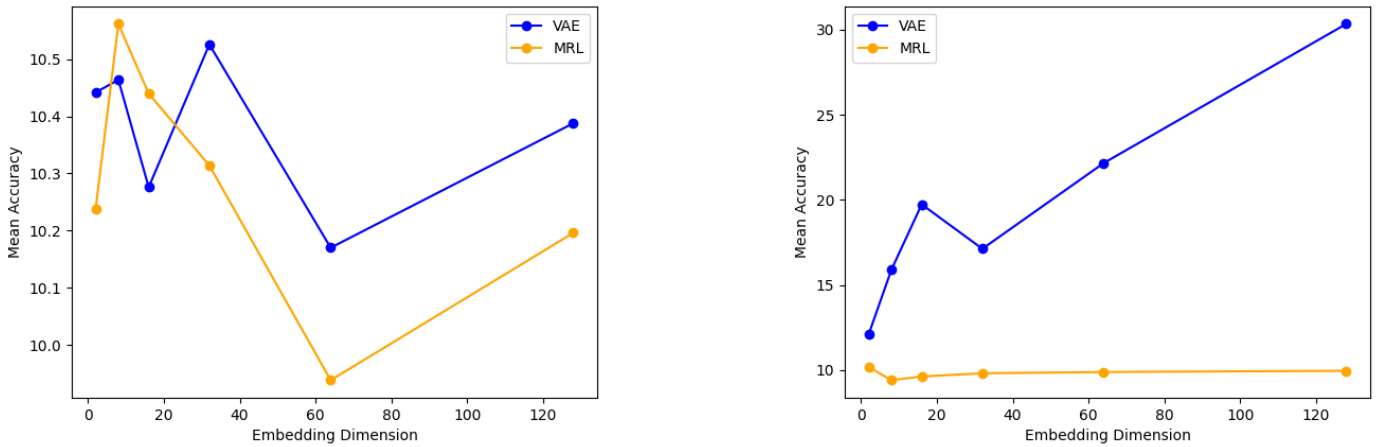
- After training the Encoder Network, we train a classifier separately for every $m \in \mathcal{M}$ using the first m dimensions of the representation $z = q(x\,;\phi)$

Towards **Learning** Better and **Flexible Representations from Variational Autoencoders**

- Similar to previous case, we used a feedforward Network consisting of 2 linear layers for classification.
- The classifiers are trained on softmax cross-entropy loss function.
- The encoder and decoder networks of VAE consist of 2 convolutional layers and 2 linear layers just like before.
- We trained both the standard Variational Autoencoder architecture and InfoVAE architecture in this setting.
- We took $\mathcal{M} = [2, 8, 16, 32, 64, 128]$ as the sizes of the representations.
- VAE and classifiers were trained for 20 epochs with a batch size of 32.
- We implemented an early stopping module to prevent overfitting.
- For evaluation, we have employed 5-fold cross validation.
- We compared the results of MRL with individually trained models.

Towards Learning Better and Flexible Representations from Variational Autoencoders

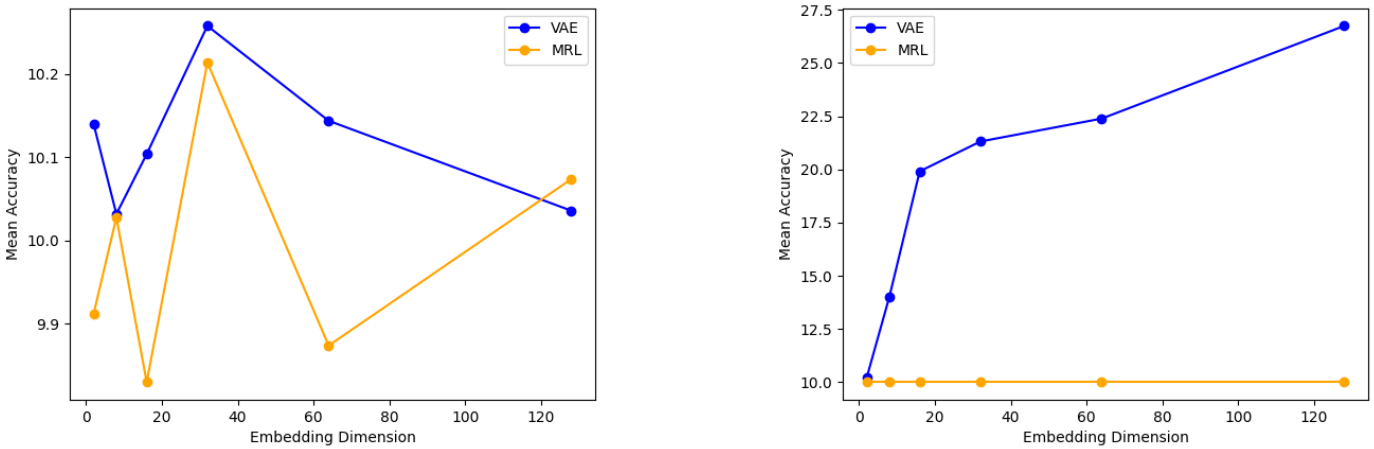
1. MNIST Dataset
   - Classification accuracy for MRL in posterior sampling setting also varies around 10% which is comparable to random labeling.
   - However, for Maximum a posteriori, Matryoshka Representation Learning is not able to keep up with individually trained VAE models.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 17: Classification Accuracy of MRL-VAE on MNIST Dataset

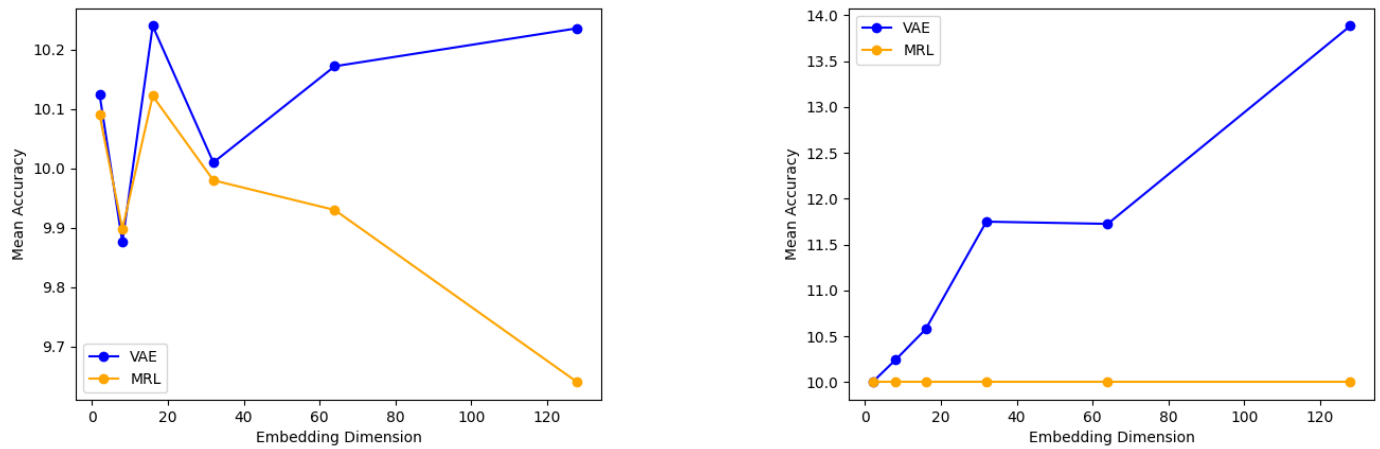
2. FashionMNIST Dataset
   - Similar to MNIST Dataset, Matryoshka Representation Learning follows a similar trend as that of individually trained VAE models.
   - Meanwhile for Maximum a posteriori, Matryoshka Representation Learning is not able to keep up with individually trained VAE models.



(a) Posterior Sampling (b) Maximum a posteriori

Figure 18: Classification Accuracy of MRL-VAE on FashionMNIST Dataset

3. CIFAR10 Dataset
   - For CIFAR10 dataset as well, classification accuracy for MRL in posterior sampling setting varies around 10%.
   - Similarly, Matryoshka Representation Learning is not able to keep up with individually trained VAE models.

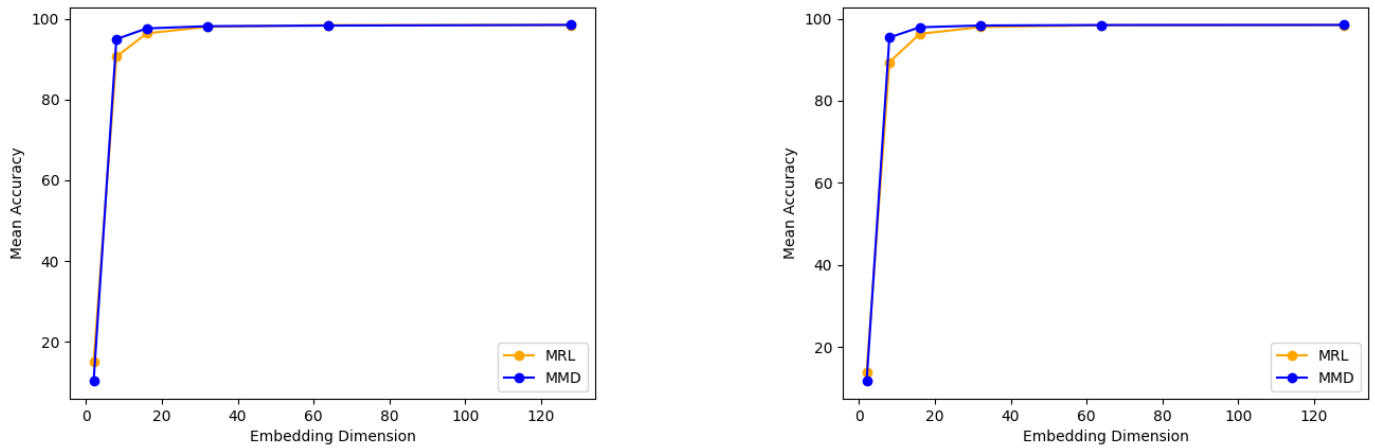


(a) Posterior Sampling (b) Maximum a posteriori

Figure 19: Classification Accuracy of MRL-VAE on CIFAR10 Dataset

Towards Learning Better and Flexible Representations from Variational Autoencoders

1. MNIST Dataset
   - Unlike VAE, classification accuracy for MRL on InfoVAE architecture is able to keep up with the individually trained models in both Posterior Sampling and Maximum a posteriori settings.



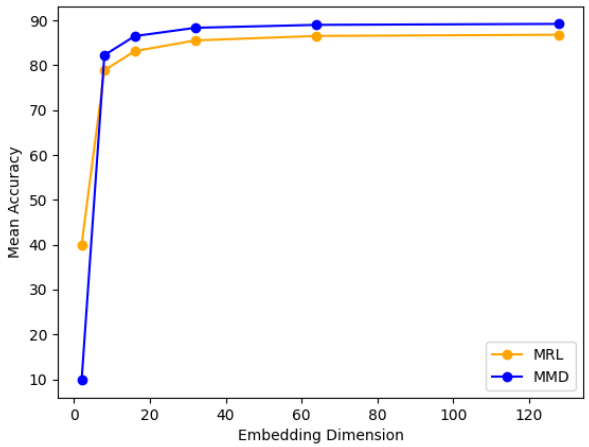
(a) Posterior Sampling (b) Maximum a posteriori

Figure 20: Classification Accuracy of MRL-MMD on MNIST Dataset

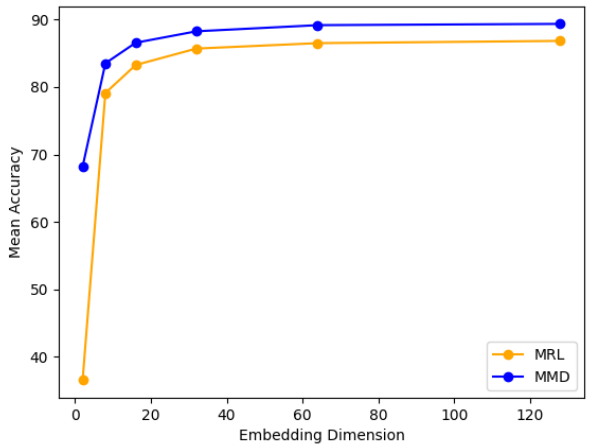
Towards Learning Better and Flexible Representations from Variational Autoencoders

2. FashionMNIST Dataset

- Similar to MNIST Dataset, Matryokshka Representation Learning is able to keep up with individually trained models and achieve high classification accuracy in both settings.



(a) Posterior Sampling



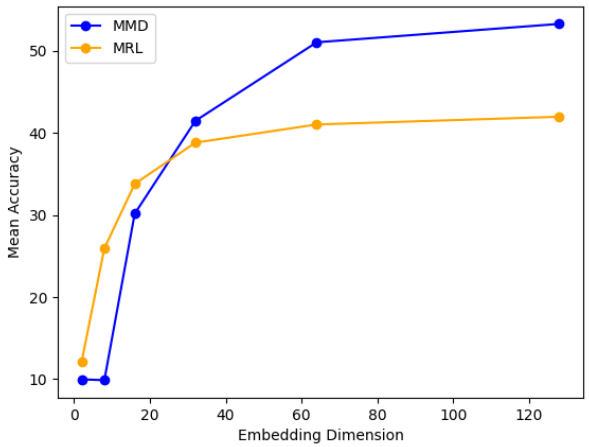
(b) Maximum a posteriori

Figure 21: Classification Accuracy of MRL-MMD on FashionMNIST Dataset

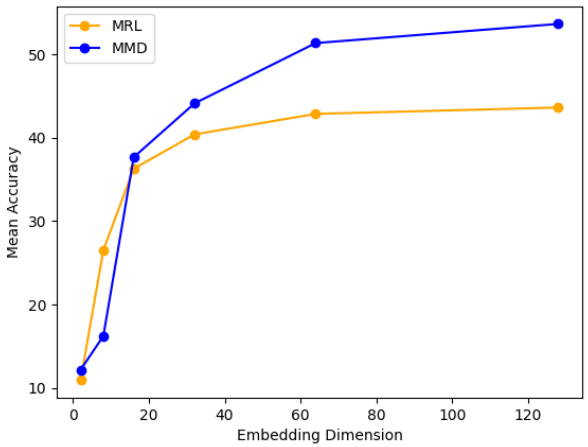
Towards Learning Better and Flexible Representations from Variational Autoencoders

3. CIFAR10 Dataset

- For CIFAR10 dataset as well, Matryokshka Representation Learning is able to keep up with individually trained models and achieve high classification accuracy in both settings.



(a) Posterior Sampling



(b) Maximum a posteriori

Figure 22: Classification Accuracy of MRL-MMD on CIFAR10 Dataset

Towards Learning Better and Flexible Representations from Variational Autoencoders

- In summary, we were able to implement Matryokshka Representation Learning in InfoVAE architecture and were able to obtain flexible representations without much loss in accuracy.
- The MRL technique didn't work in case of standard VAE architecture and wasn't able to keep up with individually trained models.
- Therefore, with the help of Matryokshka Representation Learning and InfoVAE architecture, we were able to learn better and flexible representations from Variational Autoencoders