FEW SHOT GENERATIVE CLASSIFICATION

A PREPRINT

Kamil Garifullin *
Department of Data Science
The Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, bld 1, 121205
kamil.garifullin@skoltech.ru

Ignat Melnikov
Department of Data Science
The Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, bld 1, 121205
ignat.melnikov@skoltech.ru

Irina Lebedeva
Department of Internet of Things and Wireless Technologies
The Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, bld 1, 121205
irina.lebedeva@skoltech.ru

Artem Alekseev
Department of Data Science
The Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, bld 1, 121205
artem.alekseev@skoltech.ru

Viktoriia Zinkovich
Department of Data Science
The Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, bld 1, 121205
viktoriia.zinkovich@skoltech.ru

May 29, 2024

ABSTRACT

In the following research, we investigate the problem of few-shot learning, which arises when only limited amounts of labeled data are available for training. The central idea is to utilize a generative model, capable of producing specific types of images. By training this model, we aim to gain insight into the underlying data structure. Subsequently, we can extract a set of more meaningful features from the model for the labeled data. With these features and their corresponding labels, it becomes feasible to train a simpler neural network for classification tasks, requiring fewer computational resources and a smaller labeled training set.

Keywords Diffusion model · Variational autoencoder · Generative Adversarial Network

1 Introduction

Training neural networks for classification tasks poses a fundamental challenge in machine learning. Traditional supervised methods often demand significant human effort and computational resources. Creating a training dataset for classification typically involves collecting thousands of images, a process that can take months. Compounding this challenge is the issue of poorly labeled data. This problem arises not only from labels provided by non-experts but also from errors made by experts themselves Zhang et al. [2018]. At the root of these challenges is the immense data hunger of neural networks. Solving this obstacle would enable training a model on vast amounts of unlabeled data and a small labeled dataset without quality loss.

In recent years, there has been a surge of interest in leveraging generative models for various machine learning tasks, including classification. Models such as diffusion, variational autoencoder (VAE), and generative adversarial network (GAN) have shown remarkable capabilities in capturing complex data distributions Kingma and Welling [2019],

*Github repo: github.com/GenerativeClassification

Goodfellow et al. [2014], Austin et al. [2023]. Frameworks like DatasetGAN highlight the potential of GANs in synthesizing labeled image datasets, thereby reducing the need for manual annotation Zhang et al. [2021]. Similarly, label-efficient semantic segmentation with diffusion models demonstrates the effectiveness of these models in learning feature representations directly from raw image data Baranchuk et al. [2022]. Thus, in this report, we aim to explore an alternative approach to addressing classification tasks, inspired by recent advancements in generative modeling.

At the core of our approach is the training of a generative model, capable of producing images that represent the underlying data distribution. This process enables us to gain insights into the intrinsic structure of the data, facilitating the extraction of features directly from the model. These features can then be employed to train a simpler neural network for classification tasks, bypassing the need for extensive training datasets and reducing computational overhead.

2 Methods

In this study, we aim to validate our hypothesis by training generative models such as the Denoising Diffusion Probabilistic Model (DDPM), Variational Autoencoder (VAE), and Generative Adversarial Network (GAN). For each model, we developed an algorithm to extract features, i.e. latent representations of data samples (Section 2.1). Using the extracted features as input, we trained multiple small neural networks (two linear layers with activation) to explore performance across different training set sizes. We tested our approach on the MNIST, CIFAR-10, and CIFAR-100 datasets, using ResNet-18 as a baseline (Section 3). Notably, for the diffusion model, we conducted a hyperparameter search, selecting the optimal forward diffusion time step. We investigated how classification accuracy depends on the noise level in images (Section 2.2).

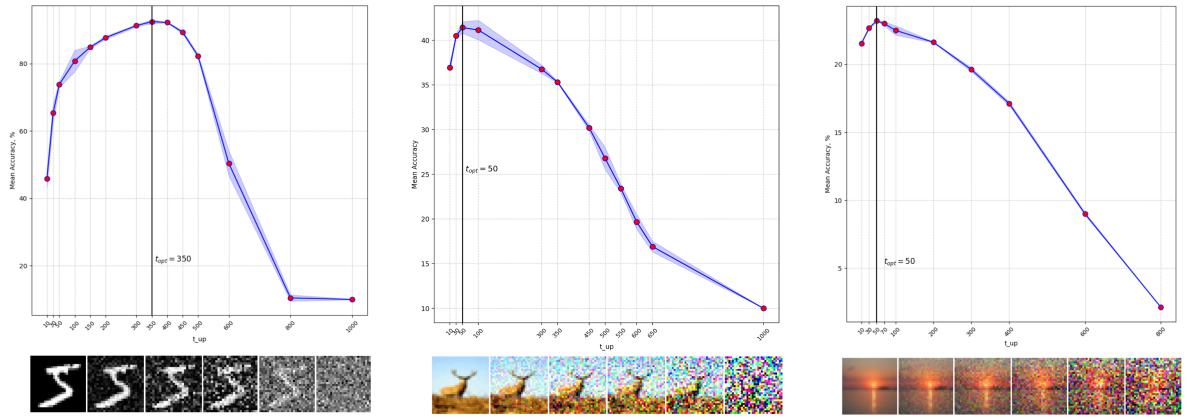
2.1 Feature extraction

- **DDPM**. Extracting features with DDPM is similar to extracting features with VAE. For diffusion model we implemented Unet architecture, so the only difference from VAE case is that we concatenate features only from the decoder part (Appendix 4, Figure 7).
- **Variational Autoencoder**. VAE shows significant performance at learning lower-dimensional feature representations of the data. These latent representations can capture the internal structure of data, which is crucial to achieve our goal. The algorithm can be summarized as follows:
  1. Transformations of the input data are extracted after each VAE layer (separately for decoder and encoder).
  2. The output data of the layers is averaged across the resolutional space and concatenated across the channels.
  3. Features of individual layers are combined and added to three datasets: separately for the decoder, encoder, and stacked (decoder + encoder features) (Appendix 4, Figure 8).
- **GAN**. A typical limitation of common GANs is their non-invertibility, which means they can generate images from random noise but cannot extract embeddings from real images Karras et al. [2020]. To address this in our paper, we train an encoder alongside the generator and discriminator Pidhorskyi et al. [2020]. A special term is added to the GAN loss to ensure it reproduces the same latent vector from a synthetic image used to generate it. These latent vectors are then used as features for images in the labeled dataset. The feature extraction pipeline and neural network architecture used in this work follow the methodology of StyleALAE Architecture in Pidhorskyi et al. [2020]. Figure 2 in their paper illustrates the relevant concept

2.2 Optimal time-step selection for diffusion model

We investigated whether using images generated by the diffusion model at specific noise steps could improve model performance. For the MNIST, CIFAR-10, and CIFAR-100 datasets, we examined the classification accuracy of two-layer neural networks based on image noise levels. Each experiment used 16 images per class and was repeated five times to account for randomness and build confidence intervals.

The optimal noise reduction steps, which significantly improved performance, were 350 for MNIST (Figure 1a) and 50 for both CIFAR-10 (Figure 1b) and CIFAR-100 (Figure 1c) datasets. These steps were used for all subsequent experiments with diffusion models. This way, classification accuracy increased by 21% for a dataset with 4 images per class and by 19% for 16 images per class for MNIST dataset (Figure 9).



(a) MNIST dataset (b) CIFAR-10 dataset (c) CIFAR-100 dataset

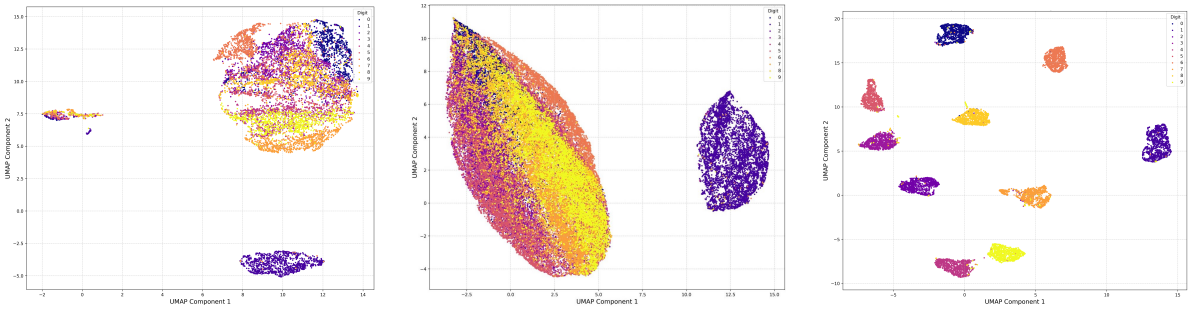
Figure 1: Optimal noise time-step in diffusion process

3 Experiments and Results

3.1 Results for the MNIST dataset

We began our experiments with the simple MNIST dataset to test our hypothesis and obtain visually interpretable results. We compared our models with the baseline ResNet-18 using training sets of varying sizes (e.g. 2, 4, 8 images per class). As it was mentioned before, we trained three generative models: diffusion, VAE, and GAN. To assess each model performance, we used Uniform Manifold Approximation and Projection (UMAP) for feature representation analysis, which allowed us to visualize feature separability. The feature dimensions were 512, 236, and 128 for each trained generative model, i.e. diffusion, VAE, and GAN, respectively.

The visualizations (Figure 2) show a clear separation between image classes, supporting our hypothesis that all models learned MNIST's intrinsic structure in their latent spaces. Interestingly, the handwritten "1" class was more distinguishable compared to other classes in the case of diffusion (Figure 2a) and VAE models (Figure 2b), which is most likely due to its distinct and simple shape, which stands out more clearly compared to other digits.



(a) diffusion model (b) VAE (c) GAN

Figure 2: 2D projection of features from generative models using the UMAP method, **MNIST dataset**

As noted in Section 2.1, we compare three approaches to extracting features from the VAE model: 1) from the encoder, 2) from the decoder, and 3) by concatenating features from both, referred to as the "stacked" method. Our analysis showed that, due to the simplicity of the MNIST dataset, there is negligible performance difference among these methods (Figure 10). Therefore, in the comparison graph of the three generative models, we only highlighted the "stacked" method for VAE, as it has a slight advantage over the "encoder" and "decoder" methods.

Finally, the overall comparison of all trained neural networks on features extracted from three generative models with the baseline ResNet-18 is depicted in Figure 3. To assess the effectiveness of our approach, we analyzed the accuracy of our models on a separate test set. As shown in Figure 3, the GAN model achieved the best results on the MNIST dataset,

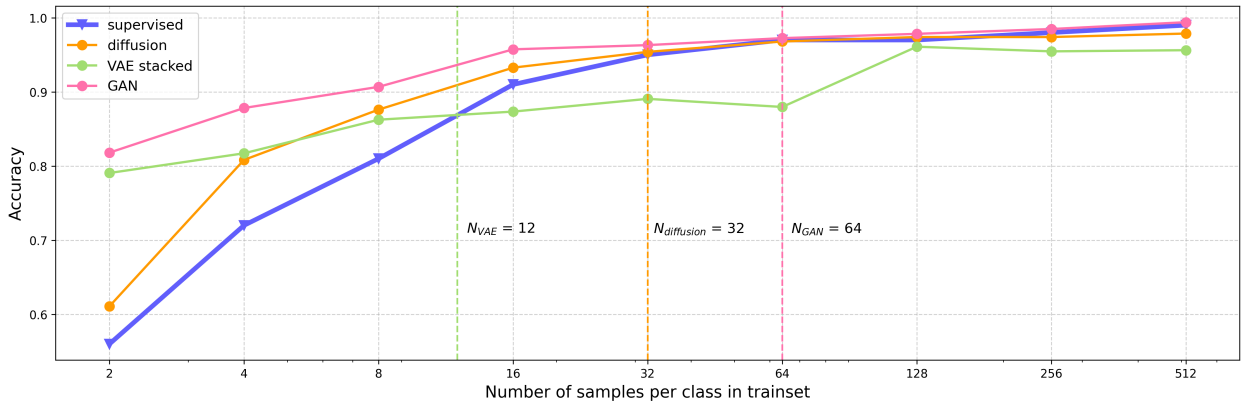
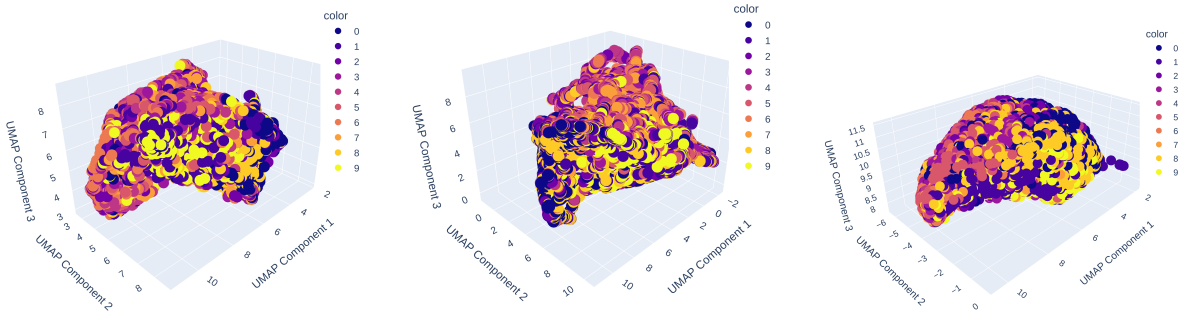


Figure 3: Comparison of three generative models for feature extraction for **MNIST dataset**

outperforming the baseline model on small labeled datasets, with up to 64 labeled images per class. The diffusion model and VAE also showed an advantage over the baseline, with improvements up to 32 and 12 labeled images per class, respectively.

3.2 Results for the CIFAR-10 dataset

In our study, we progressed to the CIFAR-10 dataset, featuring 10 classes and three-channel input images, posing a more intricate challenge for generative models. Following the approach outlined in Section 3.1, we trained the diffusion model, VAE, and GAN. However, we encountered poorer performance with VAE on this dataset compared to MNIST. Therefore, we opted for the more complex VQ-VAE model (detailed in Appendix 4, Figure 11). This improvement may be attributed to the VQ-VAE's higher feature dimensionality (1020) compared to VAE (236), enabling it to better capture the intrinsic data representation.



(a) diffusion model (b) VQ-VAE (c) GAN

Figure 4: 2D projection of features from generative models using the UMAP method, **CIFAR-10 dataset**

For all three models, as before we projected the features using UMAP onto a three-dimensional space to demonstrate their separability. In each case, we observed clear clustering of features, suggesting effective training of the generative models (Figure 4). After training simple two-layer neural networks on features extracted from generative models, we observed improved performance compared to the baseline ResNet-18 model in all three cases for small dataset sizes. The image sizes per class at which the models outperformed the baseline were 32, 190, and 220 for VQ-VAE, GAN, and diffusion, respectively (Figure 5). While both the GAN and diffusion model demonstrated exceptional performance, the diffusion model achieved comparable results with just two hours of training on a T4 GPU, whereas the GAN necessitated five hours on an A100 graphics card!

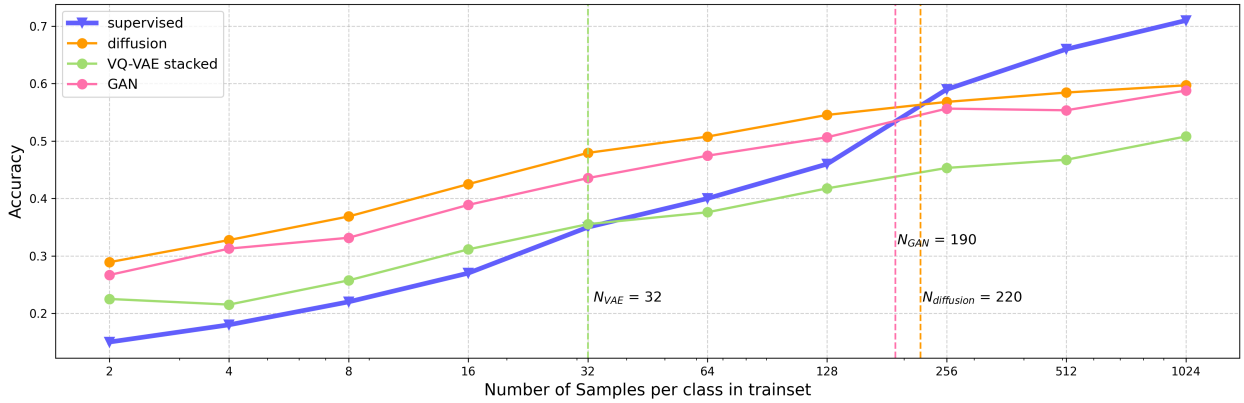


Figure 5: Comparison of three generative models for feature extraction for **CIFAR-10 dataset**

3.3 Results for the CIFAR-100 dataset

Finally, we applied the same pipeline to investigate the CIFAR-100 dataset, aiming to assess the effectiveness of our method on more complex real-world datasets. We projected the features onto a three-dimensional space, but observed less distinct separability compared to previous cases, likely due to the higher dimensionality of the features. However, the diffusion model emerged as the top performer for this dataset, significantly outperforming GAN and VQ-VAE. The diffusion model surpasses ResNet-18 with up to 256 images per class, while GAN and VQ-VAE only show an advantage with up to 50 and 16 images, respectively (Figure 6).

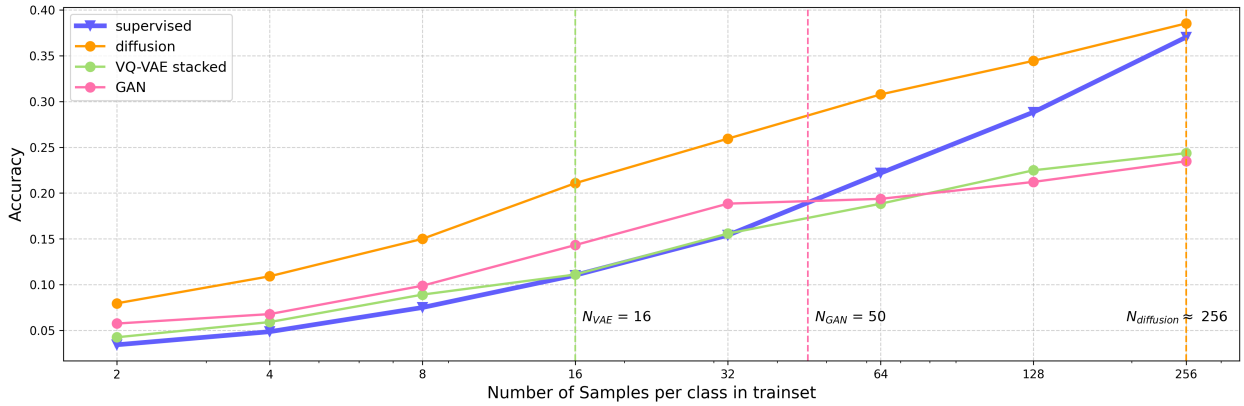


Figure 6: Comparison of three generative models for feature extraction for **CIFAR-100 dataset**

4 Conclusion

In summary, our research successfully confirms the hypothesis that leveraging generative models—specifically, diffusion, variational autoencoders (VAEs), and generative adversarial networks (GANs)—can address challenges posed by small or poorly labeled datasets. By extracting features from the hidden layers of these models, we could train simpler classification neural networks effectively.

Comparing the performance of all three models with the ResNet-18 baseline, we observe the diffusion model achieving the best results on CIFAR-10 and CIFAR-100 datasets, while GAN outperforms all models on the MNIST dataset. Additionally, the diffusion model requires fewer resources for training, typically a few hours on a T4 GPU compared to the GAN, which necessitates training on an A100 graphics card. Despite this, we believe in the potential of GANs and anticipate that refining hyperparameters or increasing the dimensionality of the latent space vector will enhance its performance.

References

Richard Zhang, Phillip Isola, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. 2018.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi:10.1561/2200000056. URL `http://dx.doi.org/10.1561/2200000056`.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort, 2021.

Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models, 2022.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Stanislav Pidhorskyi, Donald Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders, 2020.

Contributions

Explicitly stated contributions of each team member to the Final Project.

- **Kamil Garifullin**
  - Found an important error in the training script and trained a diffusion model on the MNIST, CIFAR-10, and CIFAR-100 datasets.
  - Created script to search for the optimal diffusion forward noise step, which significantly improved models performance.
  - Constructed a UMAP for projecting features extracted from the diffusion model into 3D space.
- **Viktoriia Zinkovich**
  - Trained GAN on the MNIST, CIFAR-10 and CIFAR-100 datasets.
  - Conducted literature analysis to find the method of features extraction for GAN model.
  - Prepared the GitHub Repo and the Final report.
- **Ignat Melnikov**
  - Implemented three methods for extracting features from the VQ-VAE model: 1) from the Encoder part; 2) from the Decoder part; 3) from both the Encoder and Decoder (stacked).
  - Improved VAE performance on the CIFAR-10 dataset by finding and training a stronger VQ-VAE model (e.g., accuracy increased by 30% for a dataset size of 160).
  - Trained baseline ResNet-18 to compare its performance with models developed in that research.
- **Irina Lebedeva**
  - Implemented features extraction method for VAE model.
  - Compared the performance of three methods for extracting features from the VAE model.
  - Constructed a UMAP for projecting features extracted from the VAE models into 2D and 3D space.
- **Artem Alekseev**
  - Trained a simple classifier (a two-layer neural network) on features derived from the latent space of the GAN model for different sizes of labeled datasets.
  - Trained StyleGAN for MNIST dataset to compare performance with StyleGAN2 architecture.

Appendix

Appendix 1: Models architectures for Feature extraction

Figures 7 and 8 show the schemes for extracting features from the DDPM and VAE models accordingly.

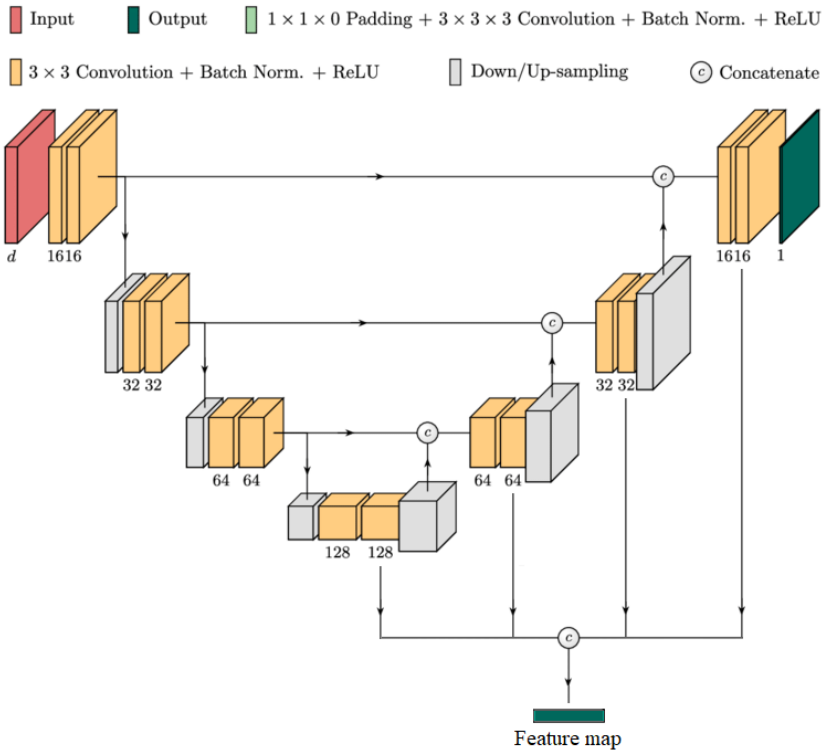


Figure 7: Feature extraction scheme for diffusion model

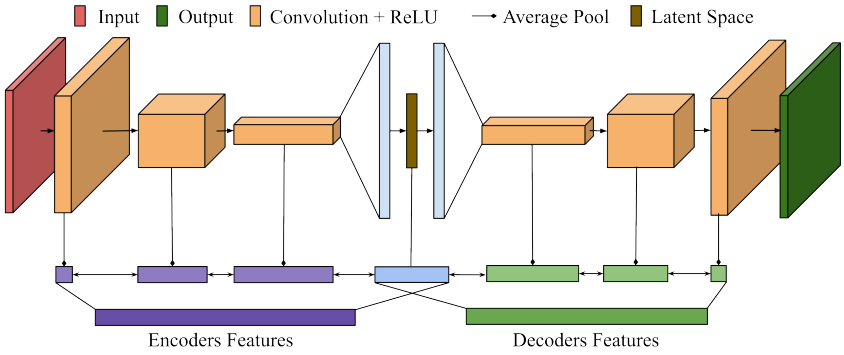


Figure 8: Feature extraction scheme for VAE model

Appendix 2: Forward diffusion process

The graph below illustrates the impact of selecting the optimal image noise step on the diffusion model's performance on the example of the MNIST dataset (Figure 9). For example, with 4 images per class, the accuracy of the model trained on diffusion features, considering the optimal noise time step, increased from 60% to 81%.

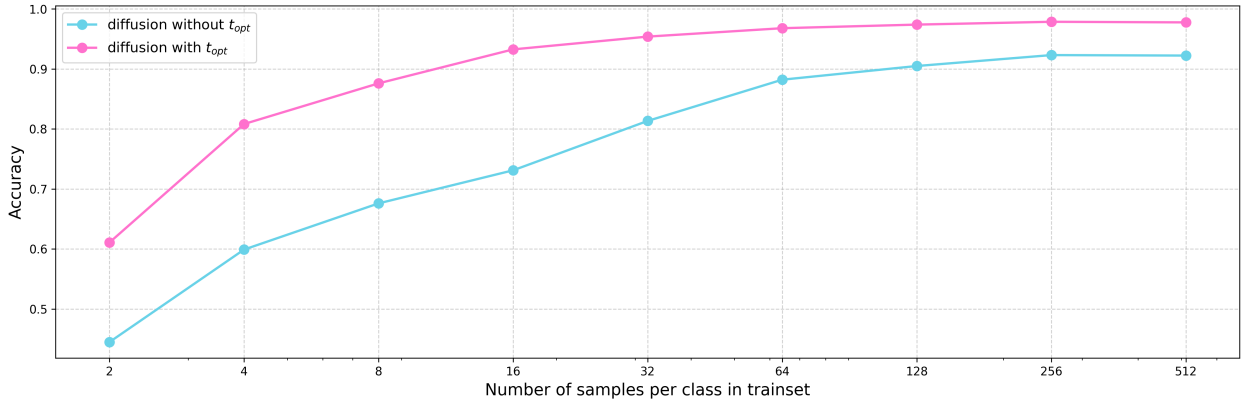


Figure 9: Comparison of models trained on images with and without noise from the optimal noise step

Appendix 3: Methods of feature extraction for VAE model

For feature extraction in the VAE model, three methods were proposed: 1) from the encoder part, 2) from the decoder part and 3) from both encoder and decoder parts (stacked). As depicted in Figure 10, there is practically no difference in which way to extract the features. However, the **stacked** method is slightly better than the others, so it was employed to compare the results of the VAE model with those of other models.

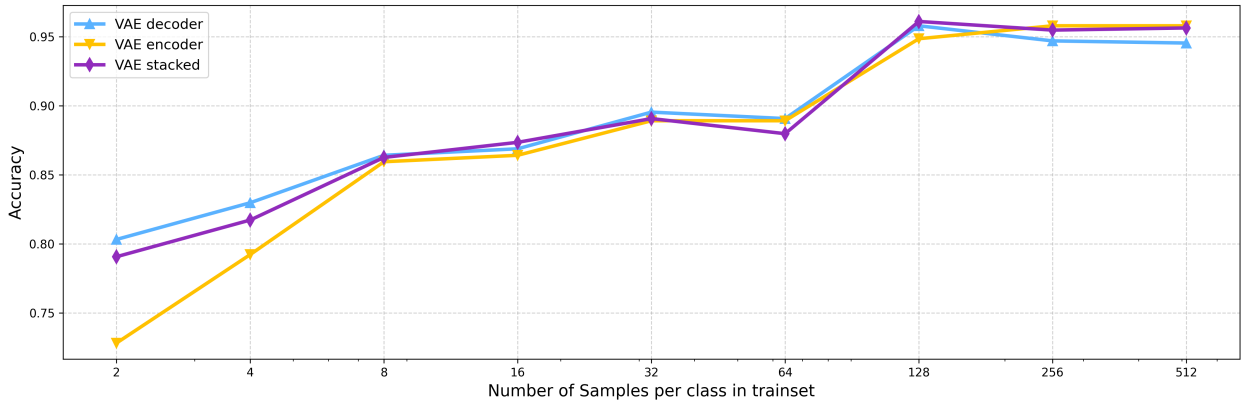


Figure 10: Comparison of three ways of features extraction from VAE model

Appendix 4: Comparison of VAE and VQ-VAE models on the CIFAR-10 dataset

The graph below compares the VAE model to the VQ-VAE model on the CIFAR-10 dataset. The new VQ-VAE model improved classification accuracy, increasing it from 27% to 33% for 16 images per class. Since the feature extraction method (decoder vs. stacked) had minimal impact, we used the VQ-VAE stacked method throughout our work. (Figure 11).

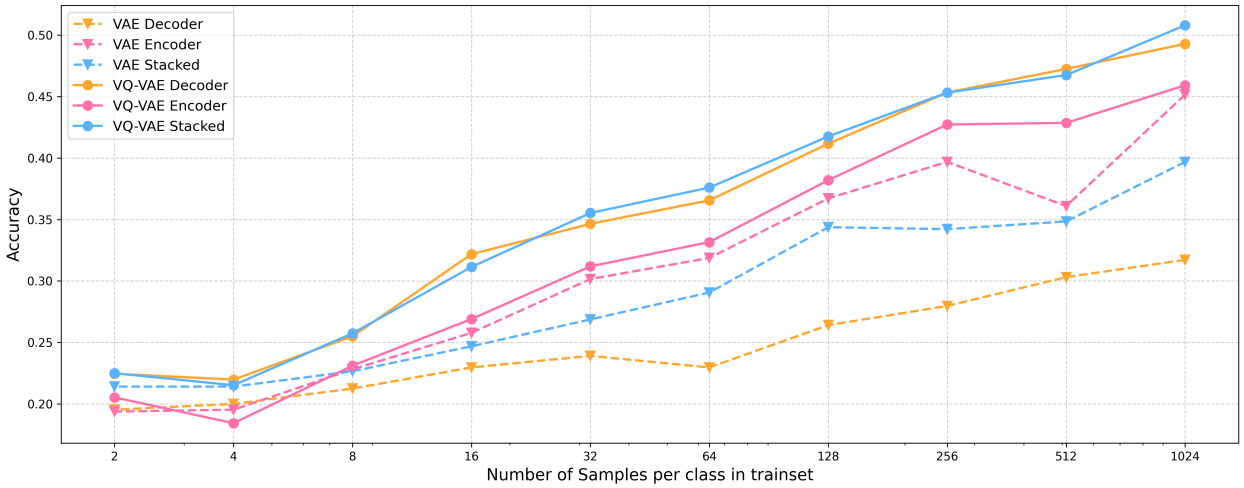


Figure 11: Comparison of VAE and VQ-VAE models on the CIFAR-10 dataset