REPLICATING DIFFUSEVAE AND ATTEMPTS AT IMPROVEMENT

Yifan Li, Gaoxiang Ye, Chenhao Zhou, Zhetao Huang
Peking University

ABSTRACT

Variational Autoencoder (VAE) stands as a generative model architecture that facilitates access to a low-dimensional latent space. However, its poor quality in generating samples necessitates improvement. Conversely, diffusion probabilistic models exhibit the capability to stably generate high-quality image reconstructions. Yet they lack a low-dimensional and interpretable latent space. The DiffuseVAE, an integration of two architectures, endows the diffusion model with controllable generative capabilities while preserving its inherent generative ability. Our work builds upon the replication results of this model, along with our attempts at further accelerating the sampling of the model via

1 INTRODUCTION

Generative modeling requires the acquisition of underlying distributions inherent in data within a statistical framework. Subsequently, the generation of new samples is undertaken in an unsupervised manner through explicit or implicit encoding mechanisms. Variational Autoencoders (VAEs) (Kingma & Welling (2013);Rezende & Mohamed (2015)), serving as a general generative modeling framework, are designed to learn explicit low-dimensional latent representations of data through the maximization likelihood estimation.

However, in image synthesis applications, VAE generated samples (or reconstructions) are usually blurry and fail to incorporate high-frequency information due to the over-smooth property of MSE objective. Recently, Denoising Diffusion Probabilistic Models (DDPM) (Sohl-Dickstein et al. (2015); Ho et al. (2020)) have been shown to achieve impressive performance on several image synthesis benchmarks. But conventional diffusion models are lack of a accurate low-dimensional latent space because of the inaccurate of vanilla DDIM Inversion process, thus there is a lack of interpretability and controllable method to generate images with conventional DDPM.

As VAE offers a proper description of low-dimensional latent space and DDPM can achieve the generation of high-frequency information by its natural property of Markov chain, so why not combine this two techniques to construct a better generative model? DiffuseVAE has accomplished this intuitive idea with some variant design based on the original DDPM, and achieves a better performance than both VAE and DDPM. As this is an interesting and intuitive idea which creates a better combination based on two robust generative models, such way of thinking will also promote our future study and research.

This technical report contains four main parts. The preliminaries of this work will be succinctly expounded upon in the Sec. 2. In Sec. 3, we elucidate the replication intricacies and present the results of reproducing DiffuseVAE. In addition, some available attempts based on the foundational model will be demonstrated in Sec. 4.

2 PRELIMINARIES

2.1 VAE: VARIATIONAL AUTOENCODERS

Variational Autoencoder (VAE) (Kingma & Welling (2013);Rezende & Mohamed (2015)) is a type of generative model designed to learn efficient representations of data in an unsupervised manner. It

combines elements of autoencoders with probabilistic models, introducing a probabilistic interpretation of the latent space.

VAE consists of an encoder and a decoder. The encoder maps input data to a probability distribution in the latent space, and the decoder reconstructs data from samples drawn from this distribution. The latent space is treated as a probability distribution, typically a multivariate Gaussian.

VAEs typically provide access to a low-dimensional latent space, but the quality of generated samples is often poor compared to other generative models, as the low-dimensional bottleneck encoding causes the loss of some high-dimensional texture features.

2.2 DDPM: Denoising Diffusion Probabilistic Models

DDPM (Denoising Diffusion Probabilistic Models) (Sohl-Dickstein et al. (2015); Ho et al. (2020)) is a type of generative model designed to capture and model the underlying probability distribution of data.

The core idea of DDPM is rooted in denoising, where the model learns to denoise (remove noise from) a sequence of noisy observations to recover the underlying data distribution. The denoising process is typically achieved through a series of diffusion steps. In each step, the model learns to transform the data from a noisy distribution to a less noisy one, ultimately converging towards the true data distribution.

DDPM models have demonstrated effectiveness in generating high-quality samples and performing tasks such as image generation. However, the denoising diffusion process involves multiple steps, and achieving convergence may require significant computational resources. And it lacks the low-dimensional, interpretable latent space, which is required for the controllable image synthesis.

2.3 DiffuseVAE: Combine VAEs with Diffusion Models

DiffuseVAE is a novel generative framework that integrates a standard VAE within a diffusion model by conditioning the diffusion model samples on the VAE generated reconstructions.

The resulting model significantly improves the generation of blurry samples commonly associated with VAEs. Additionally, it equips the diffusion model with a low-dimensional VAE-inferred latent space, which can be utilized for downstream tasks such as controlled synthesis and image attribute manipulation.

In essence, DiffuseVAE (Pandey et al. (2022)) proposes a generative model that combines the strengths of both VAE and diffusion models. The integration of Variational Autoencoder (VAE) into the framework of the Diffusion Model is undertaken with the dual objective of preserving the stable generative quality inherent in the Diffusion Model, while concurrently endowing the latter with a low-dimensional latent space. This augmentation facilitates the utilization of the Diffusion Model for downstream controlled generative tasks.

3 Experimental Results

We replicate the improvement in image sampling quality presented in the paper and perform an ablation study on the effect of injecting the latent code generated by the VAE into the diffusion U-Net at each time step. As for details, we don't retrain the VAE but use the pre-trained checkpoint provided in the official Github repository, because we think the main novelty doesn't lie on VAE but on the combination of VAE and DDPM. We conduct out training and testing on 3 RTX 2080Ti GPUs. We stop the training and get the ultimate checkpoints after 20 epochs once the curve of loss has been converged. We present our quantitative and qualitative results here.

3.1 Replication

We ran the model on 2 datasets, CIFAR10 and CelebA-256. We believe that with the former acting as a proof of concept for the model, and the latter one presenting a more challenging and real-life use case for the model, the selection of these 2 provide a revealing study into DiffuseVAE's capabilities.

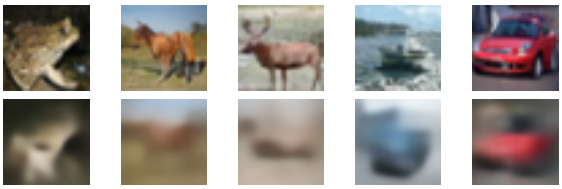

Figure 1: CIFAR10 images sampled with DiffuseVAE (top) vs. VAE (bottom). No latent code guidance used.

Figure 2: CelebA-256 images sampled with DiffuseVAE (top) vs. VAE (bottom). No latent code guidance used.

For models trained on the CIFAR10 dataset, the DiffuseVAE model and VAE exhibit dramatically different sampling quality (see Fig. 1). While the VAE is only capable of generating images consisting of rough geometric shapes and color gradients, the DiffuseVAE generated images have clear edges and defined shapes, and are easily identifiable. For quantitative results, see Fig. 4 and Fig. 5.

As for the more challenging CelebA-256 dataset, the results are similar (see Fig. 2). The VAE generated images, while more recognizable than their CIFAR10 counterparts, are still relatively blurry and exhibit smooth color gradients. The DiffuseVAE-sampled images are much sharper and retain richer detail.

Note that for both datasets (and more prominently in the CelebA-256 dataset), the silhouette of the characters are retained. This indicates the low-quality samples generated by the VAE's ability to be an effective and controllable precursor for high-quality image synthesis.

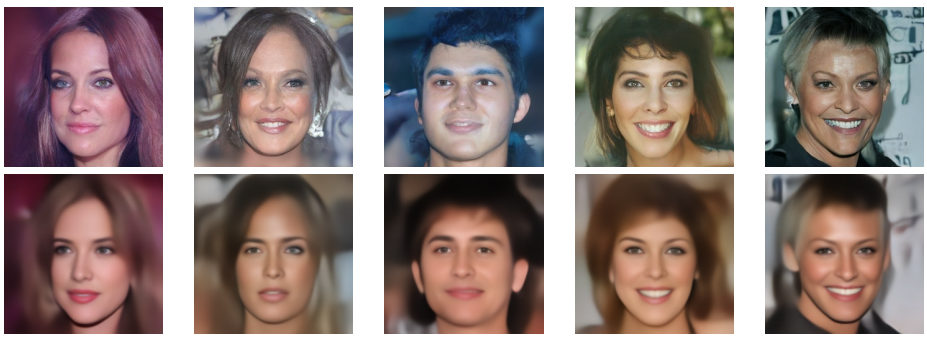

Figure 3: CelebA-256 images sampled with DiffuseVAE (top) vs. VAE (bottom). Latent code guidance used.

	CIFAR10	CelebA-64	CelebA-256
VAE	147.13	101.98	78.13
DiffuseVAE, Form 1, w/o Latent Code Guidance	9.23	16.89	6.09
DiffuseVAE, Form 1, w/ Latent Code Guidance	8.98	16.02	5.98
DiffuseVAE, Form 2, w/o Latent Code Guidance	9.12	17.23	5.79
DiffuseVAE, Form 2, w/ Latent Code Guidance	8.65	15.64	6.23

Figure 4: FID score for images generated by different models on different datasets

	CIFAR10
VAE	3.01
DiffuseVAE, Form 1, w/o Latent Code Guidance	8.21
DiffuseVAE, Form 1, w/ Latent Code Guidance	9.01
DiffuseVAE, Form 2, w/o Latent Code Guidance	9.55
DiffuseVAE, Form 2, w/ Latent Code Guidance	9.75

Figure 5: Inception scores for images generated by different models on the CIFAR10 dataset

3.2 Ablation Studies

The low-quality images sampled from VAE controls the sampling process of the DDPM in two ways:

1. By using the VAE-sampled image as x_0 in the diffusion model.
2. By passing the latent code learnt by the VAE as context into the diffusion U-Net at each time step.

The former of these methods proves indispensable to the DiffuseVAE method. Thus, we focus our discussion on the effect of the later method on the sampling quality.

Quantitatively, the inclusion of latent code guidance yield minor improvements for the FID score (see Fig. 4). However, as for the CelebA-256 dataset, quantitatively speaking, latent code guidance significantly reduces the occurrence of artifacts such as mismatched eyes and uncanny face shapes, resulting in a greater increase in perceived quality than that in FID. This is also reflected in that the inception scores, which utilizes a GAN to assess the perceived quality of a image, see more significant improvements than FID scores (see Fig. 5).

4 Discussion

During the course of our work, we have made numerous attempts which have unfortunately failed to yield satisfying results. To show the original extent of our ideas and the amount of labor put into the work, we list them in this section.

4.1 Quantization

DiffuseVAE, along with improvements in sampling quality, also allows for faster generation. To further this speed advantage, we attempted to quantize the diffusion neural network in a DiffuseVAE model.

Through this process, we referenced a number of papers, all of which share the same general idea: by modifying existing post-training quantization (PTQ) methods to take a time step as input, the resulting PTQ algorithm can quantize full-precision diffusion models to around 4-bit for weights and 8-bit for activation, with minimal to acceptable losses in quality. Due to the similarity in their overarching structure, we will not go into detail as to each paper's main idea.

We first turned to the Q-Diffusion (Li et al. (2023)) project. We successfully created a simplified CNN with off-the-shelf Pytorch modules (`nn.Linear`, `nn.Conv2d`, etc.) and a simplified U-Net with custom `ResNetBlocks` and `AttentionBlocks` provided in the codebase provided by the

paper. However, upon testing, we discovered that the quantized models did not yield statistically significant increases in speed or decreases in size. After examining the code, we discovered that its implementation chooses to store the quantized weights in float data type, thus resulting in unchanged size and inference speed, only decreased accuracy that simulates that of a "true" quantized model.

Though unfortunate, we learned the basic pipeline of creating a custom full-precision model, and quantizing it by passing it into a quantized model class. This pipeline appears standard as it showed up multiple times in papers we've later examined.

Afterwards, we referenced the Post-training Quantization (Shang et al. (2023)) project. We soon gave up on running its code as it did not provide pre-trained checkpoints, and training one ourselves can be very costly, time-consuming, and require expensive GPU rentals, especially for the ImageNet dataset, with its large image size and numerous categories.

We then turned to the PTQD method (He et al. (2023)). However, we also found this hard to replicate due to discrepancies in the calibration dataset format generated by the provided script and that used to quantize the model. We attempted to fix the discrepancy by tweaking the image dimension and batch size, but eventually discovered that this is irreconcilable: the quantize script expects a time step in the data loader, which the code that generates the calibration dataset in the DDLM forward process is fundamentally incapable of this. We scrapped this code base as well.

Another potential obstacle is the existence of custom CUDA kernels. Many high-performance quantization schemes for language generative models come with their own custom CUDA kernel. Further attempts to sample a quantized model efficiently might necessitate a custom CUDA kernel, for which although open-source versions exists, minor modifications to the quantization scheme might need major changes to the kernels.

4.2 KNOWLEDGE DISTILLATION

Apart from quantization, we also explored the possibility of acceleration and model compression through knowledge distillation. Our discussion were focused on distilling DDPM with the aim of reducing denoising steps for speeding-up and lowering model size. Therefore, we reviewed various papers related to knowledge distillation of diffusion models.

We first turned to the Progressive Distillation method (Salimans & Ho (2022)). This work distills a teacher model into a student model with fewer steps through an iterative process. But the official code base is built on JAX instead of Pytorch, and is required to be runned on TPUs. Considering the constraints of time and computational resources, we gave up the replication of this paper on Pytorch.

Then we referenced the Adversarial Distillation(Sauer et al. (2023)) and the Guided Diffusion Distillation(Meng et al. (2023)). Unfortunately, we found that the official code repository for the former is mainly a products release page, while the latter does not have open-source code. As a result, we had no access to the referable code that reproduces their distillation methods thus giving up replicating their work from scratch.

4.3 UTILIZE PCA ON THE LATENT CODE

As we can see, the performance of model can increase with the guidance of the latent code extracted from the VAE. An intuitive idea is to guide the training or sampling of the DDPM with a lower dimensional latent code, just like the conventional classifier guidance. Because the dataset we used is simple enough, for example, Celeba64 and Celeba256 both contains low or high resolution human faces which have few complex features compared with other natural images in the real world. Thus this idea should be an effective way to enhance the interpretability of DDPM. However, when we add PCA into the training process, the loss of training will immediately increase and finally become 'NAN'. We try some other tricks to save this idea, such as (1) add PCA only at the early time steps when the SNR is high; (2) only add PCA at the inference phase; (3) try more PCA dimension with 32, 64, even 128. But these changes did not work at all. From this perspective we can also see that the spiking of idea is always not dependable, only serious mathematics can help to construct a wonderful DDPM which is strict to training strategy.

5 CONCLUSION

In this work, we successfully replicated the novel framework, which facilitates the realization of a two-stage training strategy for Variational Autoencoders (VAEs) and diffusion models. The experiment results have proved its effectiveness in generating high-quality samples.

Concurrently, we also explore to optimize the existing framework through strategies such as various quantization approaches, knowledge distillation, and Principal Component Analysis (PCA). While constrained by various limitations that hindered optimal realization, it is hoped that these ideas may serve as inspiration for future research in the corresponding field.

REFERENCES

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *arXiv preprint arXiv:2305.10657*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2023.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. pp. 1530–1538, 2015.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1972–1981, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. pp. 2256–2265, 2015.