Human Faces Generation with Diffusion Models

Frank Lu, Kaan Emre Sanal, Ziyu Wang, Xiaoguang Liang
Faculty of Engineering and Phys. Sciences
University of Surrey
{tl01159, ks02303, zw00953, xl01339}@surrey.ac.uk
https://github.com/frankcholula/fAIce

***Abstract*— This paper presents a comprehensive investigation of diffusion models, with a focus on architectural improvements and experiments with various sampling methods and hyperparameters, including data augmentation and loss functions. We also successfully demonstrated both class-based and text-based guidance, the latter implemented through a fine-tuned stable diffusion model. Using our best models and a dataset of 3,000 CelebAHQ 128×128 images [1], we achieved an FID of 47.9 using the standard UNet backbone through systematic ablations, an FID of 56.9 using our custom-trained latent diffusion model and an FID of 53.3 using a fine-tuned pre-trained LDM, and an FID of 68.2 by a custom-trained DiT model. We also successfully demonstrated both class-based and text-based guidance, the latter implemented through a fine-tuned stable diffusion model.**

***Index Terms*— Diffusion Models, Latent Diffusion, U-Net, Conditional Generation, Stable Diffusion, Diffusion Transformers**

I. Introduction

Generative modelling includes auto-regressive models, flow models, latent variable models, and—most prominently—GANs [2]. While GANs have achieved high image fidelity, they are notoriously difficult to train, requiring extensive hyperparameter tuning of both generator and discriminator networks. They also struggle to cover the full data distribution and are prone to mode collapse. Diffusion models have emerged as a promising alternative, demonstrating exceptional capabilities in high-quality image synthesis.

This paper presents a comprehensive investigation of diffusion models, structured in three sections. Section II establishes the theoretical foundations and current state-of-the-art through an extensive literature review. Section III develops our methodological framework, describing how we designed our experiments based on key findings from Section II. Section IV presents our ablation testing results with detailed analysis as well as conditional generation results from our models.

II. Literature Review

Our literature review follows a thematic rather than chronological approach. Given the extensive literature on diffusion, we focus on the most significant papers relevant to our investigation. For a complete overview, consult Figure 1.

A. Foundation

We begin by reviewing *Denoising Diffusion Probabilistic Models* [3] to establish both an intuitive understanding and a working baseline model. In summary, the training process consists of three steps:

1) Sample initial data $x_0 \sim \mathcal{K}$, from dataset, noise level $\sigma \sim [\sigma_{min}, \sigma_{max}]$, and random noise $\varepsilon \sim \mathcal{N}(0, I)$
2) Generate noisy data via $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon$. This uses the reparameterization trick by sampling ϵ once and forming x_t as a deterministic, differentiable function.
3) Train a model to predict ϵ from x_σ by minimizing squared loss

The process involves training a θ-parameterized neural network $\epsilon_\theta(x, \sigma)$ to minimize the loss function:

$$\mathcal{L}(\theta) = \mathbb{E}\|\epsilon_\theta(x_0 + \sigma_t\epsilon, \sigma_t) - \epsilon\|^2$$

This diffusion training procedure yields a learned denoiser $\epsilon_\theta(x, \sigma)$, which quite elegantly be thought of as an approximate projection operator onto our data manifold $\mathcal{K}$. This provides a powerful geometric perspective that the denoising process simply refines noisy samples to progressively bring them closer to the data manifold. We then aligned this intuition with the more generalised view presented in Song's work, which frames the reverse process as solving a continuous-time SDE [4]:

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt$$

Here, the term $\mathbf{f}(\mathbf{x}, t)$ acts as a deterministic drift, guiding samples toward high-density regions, while the learned score function $\nabla_x \log p_t(\mathbf{x})$ serves as a gradient that projects samples back onto the data manifold over time. In DDPM, this drift term is equal to zero, yielding a purely score-driven trajectory.

B. Algorithmic Improvement

1) Learned Variance: Nichol and Dhariwal [5] identified that fixing the variance $\Sigma_\theta(x_t, t)$ as done in Ho et al. [3] can be suboptimal with fewer diffusion steps and proposed to parameterize $\Sigma_\theta(x_t, t)$ as a neural network output v interpolated as:

$$\Sigma_\theta(x_t, t) = \exp\left(v \log \beta_t + (1-v) \log \tilde{\beta}_t\right)$$

This approach allows the model to dynamically adjust the noise variance for improved sample quality.

2) Deterministic Sampling: Song et al. introduced DDIM, a non-Markovian variant of DDPM that shares the same forward marginals but uses a modified variance schedule to enable deterministic sampling [6]. By setting the reverse noise to zero, they effectively transform the denoising process

into a deterministic mapping from latents to images, which significantly reduces the required sampling steps.

$$x_{t-1} = \alpha_{t-1}\left(\frac{x_t - \sigma_t\hat{\epsilon}(x_t)}{\alpha_t}\right) + \sigma_{t-1}\hat{\epsilon}(x_t)$$

We demonstrate this in Section III-B and show the results in Table X.

C. Noise-Image Space Parameterization

Recall the original loss in DDPM, where we predict the L2 loss in the noise space:

$$\mathcal{L}(x_0, t) = \|\epsilon - \hat{\epsilon}_\theta(\alpha_t x_0 + \sigma_t\epsilon)\|_2^2$$

We can re-parameterise the loss in the image space [7]:

$$\mathcal{L}(x_0, t) = \|x_0 - \hat{x}_\theta(\alpha_t x_0 + \sigma_t\epsilon)\|_2^2$$

Framing the relationship between the two spaces by substitutions:

$$\frac{\alpha_t^2}{\sigma_t^2}\|x_0 - \hat{x}_\theta(x_t)\|_2^2 = \|\epsilon - \hat{\epsilon}_\theta(x_t)\|_2^2$$

The weighting term $\frac{\alpha_t^2}{\sigma_t^2}$ represents the SNR ratio. Salimans and Ho [8] introduced a final reparameterization trick in terms of angle $\phi = arctan(\sigma_t/\alpha_t)$ (See Figure 2):

$$\mathcal{L}(x_0, t) = \|v - \hat{v}(\alpha_t x_0 + \sigma_t\epsilon)\|_2^2$$

where v can be understood as a velocity term that combines aspects of both noise prediction and image prediction.

$$v = \alpha_t\epsilon - \sigma_t x_0$$

This technique proves especially valuable in the zero SNR regime [9]. We explore this in more detail and show the effects of v-prediction in Section III-B and the results in Table XIII.

III. Methodology

We evaluated model performance using two industry-standard metrics: Fréchet Inception Distance (FID) [10] and Inception Score (IS) [11]. We also used clean-FID [12], a more reliable and objective implementation compared to our custom FID implementation. We conducted our training with 128×128 images across 500 training epochs using a learning rate of 0.0001. Due to computational constraints, we calculated FID scores only at the end of training rather than monitoring them throughout the process to find the lowest score.

A. Architecture Improvement

The loss curves for each model architecture's best-performing configuration are shown in Figure 10. First, we establish a foundation baseline with UNet for subsequent experiments. We then explore alternative backbone architectures in this section as candidates for our final best model.

1) UNet Improvement: Our baseline model uses DDPM's UNet architecture (Figure 3) [13], which features an encoder-decoder structure with skip connections. This design balances local detail preservation with global context understanding. We enhanced this baseline following the improvements in *Improved DDPM* and additional attention layers from *Diffusion Models Beat GANs* [5], creating our Ablation Diffusion Model (ADM) as a baseline model for subsequent tasks.

2) LDM (Latent Diffusion Model): We trained our LDM (Figure 4) from scratch using a two-stage pipeline. A VQ-VAE perceptual autoencoder (encoder E and decoder D) compresses each image into a lower-dimensional latent $z_0 = E(x)$. Its training optimises three terms in a single loss: reconstruction, codebook, and commitment [14].

$$\begin{aligned}\mathcal{L}_{\text{VQ}} = & -\log p_\theta(x \mid z_q) + \|\text{sg}[q_\phi(z \mid x)] - e_k\|_2^2 \\ & + \beta\,\|q_\phi(z \mid x) - \text{sg}[e_{k^*}]\|_2^2\end{aligned}$$

where $sg[\bullet]$ denotes the stop-gradient operator. This stage yields a ceiling FID score when decoding through D, establishing an upper bound for our subsequent diffusion model. Next, we train a diffusion process directly in that learned latent space. Starting from $z_o = E(x)$, we produce noisy latents z_t via the usual forward noising schedule and learn a time-conditional U-Net ϵ_θ (with optional conditioning $\tau_\theta(y)$ injected via cross-attention) to predict the added noise [15]. The objective is

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{E(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right]$$

By operating in this compressed space, we reduce computational cost and memory footprint compared to pixel-space diffusion. At inference time, we sample $z_T \sim \mathcal{N}(0, I)$ and run the learned denoiser in reverse to recover z_0, then pass it through D to reconstruct a full-resolution image.

3) DiT (Diffusion Transformer): Finally, we explored DiT (Figure 5) by first compressing images into a latent space via a VAE trained under the usual AEVB framework [16]. The VAE learns an encoder-decoder pair (E, D) by minimising the ELBO.

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\phi,\theta) = \sum_{i=1}^{n}\Big[& \underbrace{-\mathbb{E}_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i \mid z)]}_{\text{reconstruction}} \\ & + \underbrace{\text{KL}\big(q_\phi(z \mid x_i)\,\|\,p(z)\big)}_{\text{regularizer}}\Big]\end{aligned}$$

using the reparameterization trick $z = \mu(x) + \sigma(x) \odot \epsilon$ so that sampling is fully differentiable. The learned latents $z_0 = E(x_0)$ serves as the input to our Diffusion Transformer. The DiT backbone [17] treats each noised latent z_t as a sequence of patch-tokens, injects timestep conditioning via AdaLN-Zero, and applies stacked transformer blocks (multi-head self-attention + MLP). We then train the network $\epsilon_\theta(z_t, t)$ to the DDPM simple loss:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{z_0,\ \epsilon\sim\mathcal{N}(0,I),\ t}\left\|\epsilon - \epsilon_\theta(z_t, t)\right\|_2^2$$

The sampling and generation procedure is similar to LDM's.

B. Hyperparameters

Using the ablation-diffusion architecture from Section III-A, we ran targeted experiments to isolate each hyperparameter's effect on FID.

- We tested random horizontal flips, Gaussian blur, and centre cropping—individually and combined—and report FID changes.
- We compared linear (Ho et al.) [3], scaled, and cosine (Nichol & Dhariwal) [7] beta schedules—and also tried DDIM/PNDM [18] samplers—to explore speed vs. fidelity trade-offs.
- We extended the length of the forward process from 1K to 2K to 4K steps to assess whether a more fine-grained noising schedule yields higher-quality samples or exhibits diminishing returns.
- Lin et al. [9] found that the noise schedule leaks some information about x_0 at timestep T, creating a mismatch with inference where we assume a perfect Gaussian prior with zero information about x_0. While setting noise to zero at T would resolve this, it would trivialize the noise objective.

$$\mathcal{L}(x_0, t) = \|\epsilon - \hat{\epsilon}_\theta(0 * x_0 + 1 * \epsilon)\|_2^2$$

  To put it simply, at $t = T$, we have $\alpha_T = 0$ and $\sigma_T = 1$, so the network is asked to predict pure noise from pure noise, which carries no learning signal. That's why we switch to v–prediction [8] in the zero-SNR regime.

C. Loss Functions

We compared two pixel-wise reconstruction losses: Mean Squared Error (L2) and Mean Absolute Error (L1).

- MSE loss: Penalises squared differences, favouring smoother outputs by averaging possible values.
- MAE loss penalises absolute differences, better preserves edges and handles outliers.

In addition to pixel-level losses, we incorporate Learned Perceptual Image Patch Similarity [19] as a regularisation term. LPIPS computes distances between deep feature representations extracted from a pretrained network such as AlexNet or VGG, moving beyond simple pixel-wise comparisons. The results are shown in table XIV and the loss curves are shown in Figure 13.

D. Guidance

Classifier guidance [5] proposes training a classifier $p_\phi(y \mid x_t, t)$ on noisy image x_t, providing gradients $\nabla_{x_t} \log p_\phi(y \mid x_t, t)$ to guide towards a class label. The sampling update can be expressed as follows:

$$x_{t-1} = w_0 x_t + w_1 \epsilon_\theta(x_t, t) + w_3 \nabla_{x_t} \log p_\phi(y \mid x_t, t)$$

This leverages the relationship between the score function and the noise prediction, where $\epsilon_\theta \approx \sigma \nabla_{x_t} \log p_\theta(x_t)$, to effectively incorporate class information without significantly reducing sample diversity. However, this requires training an additional classifier. Classifier-free guidance [20] eliminates this overhead by directly integrating class information into the model using Bayes' rule:

$$w \nabla_x \log \left(\frac{p(x \mid y) p(y)}{p(x)} \right) = w \nabla_x \log p(x \mid y) - w \nabla_x \log p(x)$$

The class-dependent gradient is decomposed into a term involving only the generative model's scores. This reformulation leads to the simplified classifier-free guidance equation (in DDPM notation):

$$w\epsilon_\theta(x, y) - w\epsilon_\theta(x)$$

where $\epsilon_\theta(x, y)$ is the conditional noise prediction and $\epsilon_\theta(x)$ is the unconditional counterpart. We demonstrate this approach in Section III-D using binary class labels (male versus female) and a guidance scale w of 0.5. During training, we randomly drop out conditioning, and during sampling, we apply the aforementioned conditional and unconditional score estimates. Results are shown in Figure 15 (male) and Figure 16 (female).

IV. Experiments

A. Architecture Improvement

1) UNet Ablation: Each UNet ablation experiment is carried out with a batch size of 24 on the NVIDIA RTX A4000. The UNet ablation experiments align with Nichol and Dhariwal's findings [5], with increasing attention heads and adding multi-resolution attention delivering the most significant improvements, resulting in a -7.69993 FID improvement (See Table I).

Our heads ablation experiment also demonstrated that fewer channels per head yielded a slight decrease in FID scores. However, to balance performance with computational costs, we opted to maintain 4 heads per layer with 32 channels per head (See Table II).

2) LDM Experiments: We first conducted VQ-VAE ablation experiments to determine the ceiling FID score. This was based on the assumption that effective VQ-VAE encoding and decoding would lead to a lower FID score in our latent diffusion model. All experiments were performed on an NVIDIA RTX A4000 with a latent channel size of 3 and a batch size of 16. VQ-VAE reached its best reconstruction score of 29.20 using a commitment loss weight of 0.4 (See Table III). This weight struck an optimal balance between encoder-codebook coupling, allowing the latent space to maintain both structural integrity through codebook alignment and expressiveness through encoder flexibility. Using these optimal parameters, we conducted LDM ablation experiments with a commitment loss weight of 0.4 on NVIDIA RTX A4000s with a batch size of 64. Through testing various UNet architectures, we discovered that 5 up/down encoder blocks performed best, reducing the FID score by an impressive +10.13 (See Table IV). For reference, we also included results from a pretrained 5-block UNet.

LDMs offer significant savings in GPU memory usage and training time because of the compression capability of VQ-VAE. For comparison, training a conventional diffusion model with a batch size of 64 typically requires approximately 70GB

of GPU memory and takes around 15 hours. In contrast, training an LDM under the same batch size reduces GPU memory consumption to about 15GB and cuts training time to approximately 7 hours.

3) DiT Experiments: Similar to LDM, we began with a series of VAE ablation experiments. We conducted all experiments on the NVIDIA RTX A4000 using 4 latent channels and a batch size of 16, testing various configurations of up and down encoder blocks (See Table V). After finalizing the 3-block encoder-decoder architecture, we implemented our DiT model. We conducted each ablation experiment on the NVIDIA RTX A4000 using 4 output channels and a batch size of 64. The DiT_B_2 model features 12 attention heads, an attention head dimension of 64, and a patch size of 2 (See Table VI).

Like LDMs, VAE's compression capabilities reduce the GPU memory usage required by DiT models and the overall training time. We reduced DiT training time from approximately 30 hours to 11 hours on an A100, with memory usage decreasing from 40 GB to around 15 GB.

B. Hyperparameters

1) Data Augmentation: Basic augmentations, specifically random horizontal flips and centre crops, significantly improved performance, with their combination achieving the best FID score of 48.32. These results indicate that spatial augmentations enhance the model's generalization. Centre crops focus on facial features by removing background distractions, while horizontal flips leverage facial symmetry (See Table VII).

2) Scheduler Configuration: We first increased the number of training steps since this provides a finer approximation to the underlying continuous diffusion SDE, as described in the original DDPM paper (See Table VIII). Next, we tuned the beta schedule while keeping both training steps and inference steps fixed at 1,000 to determine the best beta schedule for DDPM (See Table IX). Finally, we tested two alternative schedulers - DDIM and PNDM. DDIM allows for faster sampling by taking larger steps in the reverse process. PNDM uses higher-order numerical methods for even more efficient sampling. Our results show that DDIM with a cosine schedule achieves the best FID of 52.62 at 100 steps, while PNDM with a linear schedule reaches 56.78 FID using just 50 steps. Both drastically reduce inference time versus standard DDPM (See Tables X and XI). The loss curves are shown in Figure 11.

3) Controlling Stochasticity with η: Due to computational and time constraints, we conducted these experiments using DDIM with a linear beta schedule, 100 inference steps, and a batch size of 64. In DDIM, η interpolates between full stochasticity (η=1) and a deterministic path (η=0). We found η = 0.5 achieved the best performance with a 50.80 FID, showing a +1.83 improvement over both $\eta = 0$ and $\eta = 0.75$. In contrast, too little noise ($\eta = 0.25$) degraded FID to 55.30, demonstrating that minimal stochasticity disrupts generation without aiding exploration. Moreover, the IS score decreases as η increases, indicating a trade-off between fidelity and diversity. Overall, moderate noise injection provides the optimal balance. See Figures 6 to 9 and Table XII.

4) Prediction Type and Zero SNR: Using DDIM with a cosine beta schedule (η = 0.0, batch size 16), we found that v-prediction combined with zero SNR yielded better FID scores (See Table XIII).

C. Loss Functions

This section explores the performance of different loss functions and LPIPS configurations. Using the DDPM pipeline with a linear beta schedule, we conducted experiments with a batch size of 16 and 4,000 forward training steps. Table 3.3 demonstrates that L2 loss consistently outperforms L1, while LPIPS using AlexNet achieves better results than the deeper VGG network (contrary to Zhang et al.'s findings [19]), maintaining comparable run times on our dataset (See Table XIV). Figure 12 shows how different LPIPS weights affect the FID scores when using L2 and L1 with Alex LPIPS net, respectively. L2 with an LPIPS AlexNet weight of 0.05 achieved the best performance with an FID of 47.91. The results demonstrate that LPIPS regularisation improves perceptual realism by better matching human visual perception of textures and structures, even when traditional pixel-wise metrics suggest otherwise.

D. Guidance

1) Conditioning on labels: As specified in Section III-D, we implemented classifier-free guidance (See Figures 14) to 16. Since we simply concatenate the conditional and unconditional output, the guidance scale is effectively 0.5.

The Inception score effectively measures fidelity, while FID measures diversity. As we increase guidance, both the Inception score and FID increase correspondingly as shown in XVI.

2) Conditioning on texts: See Figures 17 and 18 for the outputs of the image captions shown in Appendix A-A. We show the comparisons between our own fine-tuned model vs. a fully fine-tuned model using Stable Diffusion v1.5.

V. Conclusion and Future Work

We achieved notable improvements through comprehensive experiments as demonstrated above. We have learned that targeted ablations significantly improved UNet performance, while comparative analysis of architectures (LDM and DiT) revealed important efficiency and image-quality trade-offs. Our successful experiments with class- and text-based guidance, implemented through a fine-tuned Stable Diffusion variant, demonstrate an effective approach to leveraging pre-trained diffusion models over the image generation process.

In the future, we would like to pursue continuous-time flow matching and one-step inference methods such as NitroFusion [21], as well as end-to-end training of VAE and diffusion model [22] and multi-modal conditioning via frameworks like ImageBind [23]. Beyond image synthesis, we look forward to investigating diffusion models' utility in other domains such as policy optimization in reinforcement learning [24] and other broader systems built on generative models.

REFERENCES

[1] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. [Online]. Available: http://arxiv.org/abs/1710.10196

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. [Online]. Available: http://arxiv.org/abs/1406.2661

[3] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. [Online]. Available: http://arxiv.org/abs/2006.11239

[4] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. [Online]. Available: http://arxiv.org/abs/2011.13456

[5] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis. [Online]. Available: http://arxiv.org/abs/2105.05233

[6] J. Song, C. Meng, and S. Ermon. Denoising Diffusion Implicit Models. [Online]. Available: http://arxiv.org/abs/2010.02502

[7] A. Nichol and P. Dhariwal. Improved Denoising Diffusion Probabilistic Models. [Online]. Available: http://arxiv.org/abs/2102.09672

[8] T. Salimans and J. Ho. Progressive Distillation for Fast Sampling of Diffusion Models. [Online]. Available: http://arxiv.org/abs/2202.00512

[9] S. Lin, B. Liu, J. Li, and X. Yang. Common Diffusion Noise Schedules and Sample Steps are Flawed. [Online]. Available: http://arxiv.org/abs/2305.08891

[10] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. [Online]. Available: http://arxiv.org/abs/1706.08500

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. [Online]. Available: http://arxiv.org/abs/1606.03498

[12] G. Parmar, R. Zhang, and J.-Y. Zhu. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. [Online]. Available: http://arxiv.org/abs/2104.11222

[13] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. [Online]. Available: http://arxiv.org/abs/1505.04597

[14] A. Razavi, A. van den Oord, and O. Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. [Online]. Available: http://arxiv.org/abs/1906.00446

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. [Online]. Available: http://arxiv.org/abs/2112.10752

[16] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. [Online]. Available: http://arxiv.org/abs/1312.6114

[17] W. Peebles and S. Xie. Scalable Diffusion Models with Transformers. [Online]. Available: http://arxiv.org/abs/2212.09748

[18] L. Liu, Y. Ren, Z. Lin, and Z. Zhao. Pseudo Numerical Methods for Diffusion Models on Manifolds. [Online]. Available: http://arxiv.org/abs/2202.09778

[19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. [Online]. Available: http://arxiv.org/abs/1801.03924

[20] J. Ho and T. Salimans. Classifier-Free Diffusion Guidance. [Online]. Available: http://arxiv.org/abs/2207.12598

[21] D.-Y. Chen, H. Bandyopadhyay, K. Zou, and Y.-Z. Song. NitroFusion: High-Fidelity Single-Step Diffusion through Dynamic Adversarial Training. [Online]. Available: http://arxiv.org/abs/2412.02030

[22] X. Leng, J. Singh, Y. Hou, Z. Xing, S. Xie, and L. Zheng. REPA-E: Unlocking VAE for End-to-End Tuning with Latent Diffusion Transformers. [Online]. Available: http://arxiv.org/abs/2504.10483

[23] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. ImageBind: One Embedding Space To Bind Them All. [Online]. Available: http://arxiv.org/abs/2305.05665

[24] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. [Online]. Available: http://arxiv.org/abs/2303.04137

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. [Online]. Available: http://arxiv.org/abs/2010.11929

[26] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu. Neural Wavelet-domain Diffusion for 3D Shape Generation. [Online]. Available: http://arxiv.org/abs/2209.08725

[27] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the Design Space of Diffusion-Based Generative Models. [Online]. Available: http://arxiv.org/abs/2206.00364

[28] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow Matching for Generative Modeling. [Online]. Available: http://arxiv.org/abs/2210.02747

[29] Y. Lipman, M. Havasi, P. Holderrieth, N. Shaul, M. Le, B. Karrer, R. T. Q. Chen, D. Lopez-Paz, H. Ben-Hamu, and I. Gat. Flow Matching Guide and Code. [Online]. Available: http://arxiv.org/abs/2412.06264

[30] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. [Online]. Available: http://arxiv.org/abs/2112.10741

[31] M. Psenka, A. Escontrela, P. Abbeel, and Y. Ma. Learning a Diffusion Model Policy from Rewards via Q-Score Matching. [Online]. Available: http://arxiv.org/abs/2312.11752

[32] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. [Online]. Available: http://arxiv.org/abs/2205.11487

[33] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. [Online]. Available: http://arxiv.org/abs/1503.03585

[34] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency Models. [Online]. Available: http://arxiv.org/abs/2303.01469

[35] Y. Song and S. Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. [Online]. Available: http://arxiv.org/abs/1907.05600

[36] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter. Diffusion Models Are Real-Time Game Engines. [Online]. Available: http://arxiv.org/abs/2408.14837

[37] P. Vincent, "A Connection Between Score Matching and Denoising Autoencoders," vol. 23, no. 7, pp. 1661–1674. [Online]. Available: https://direct.mit.edu/neco/article/23/7/1661-1674/7677

[38] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, H. Guo, T. Chen, and W. Zhang. Diffusion Models for Reinforcement Learning: A Survey. [Online]. Available: http://arxiv.org/abs/2311.01223

APPENDIX A
FIGURES

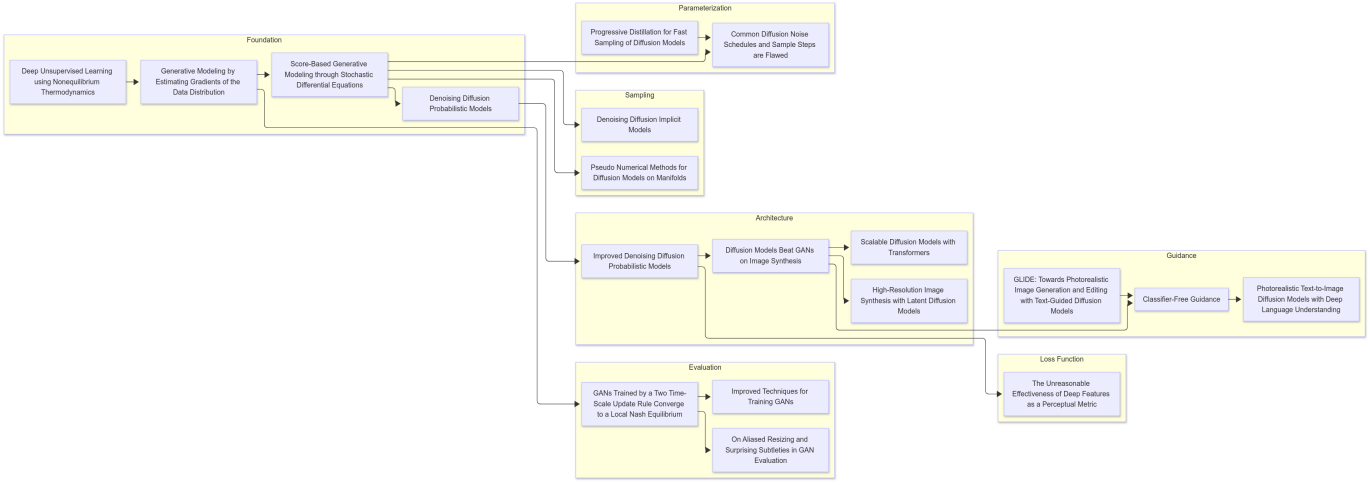

Fig. 1. Mermaid Chart

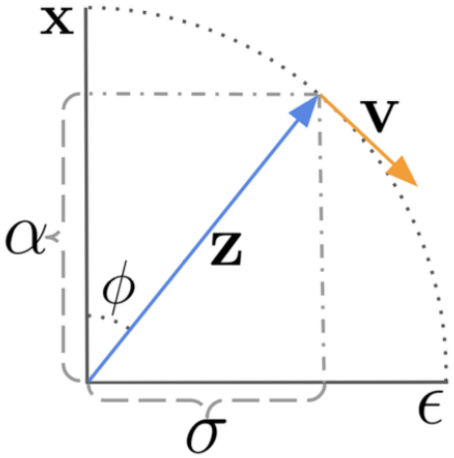

Fig. 2. Parameterizing the diffusion process in terms of ϕ and v_ϕ

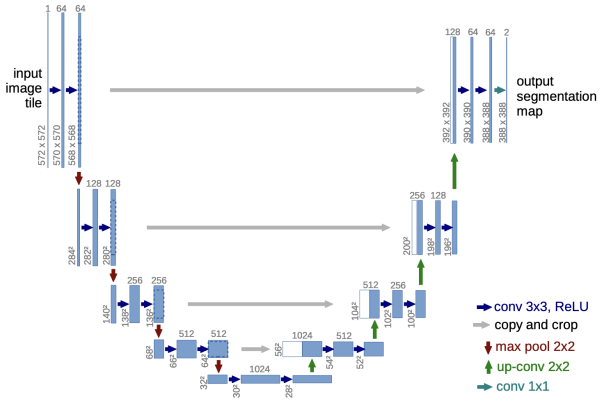

Fig. 3. U-Net Architecture

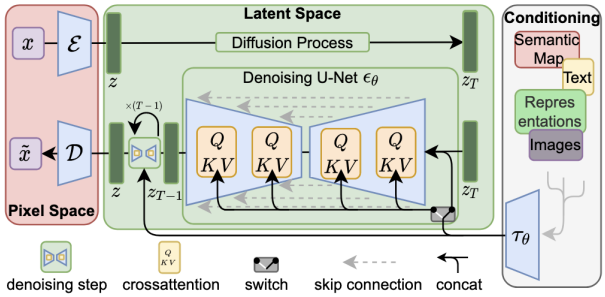

Fig. 4. Latent Diffusion Model Architecture

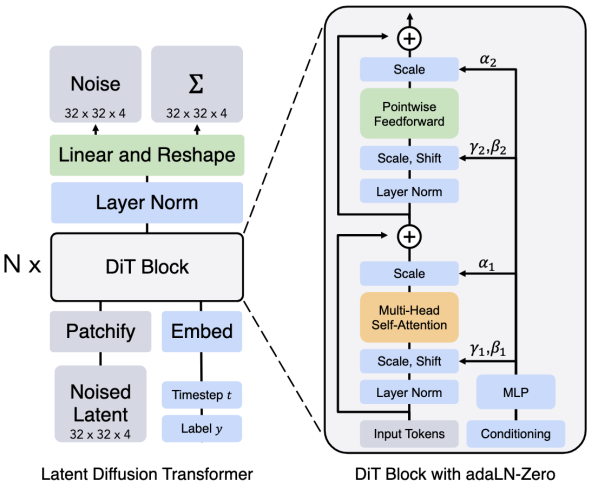

Fig. 5. Diffusion Transformer Architecture

Fig. 6. $\eta = 0$ (least diverse)

Fig. 7. $\eta = 0.25$ (less diverse)

Fig. 8. $\eta = 0.5$ (good balance)

Fig. 9. $\eta = 0.75$ (most diverse)

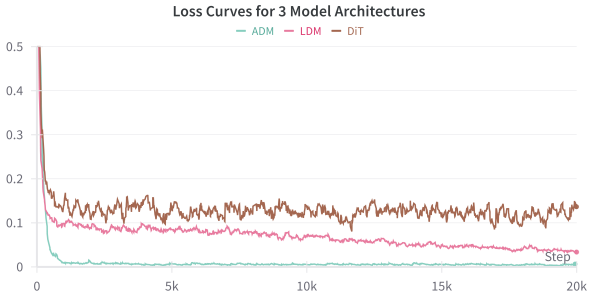

Fig. 10. Loss Curves for 3 Model Architectures

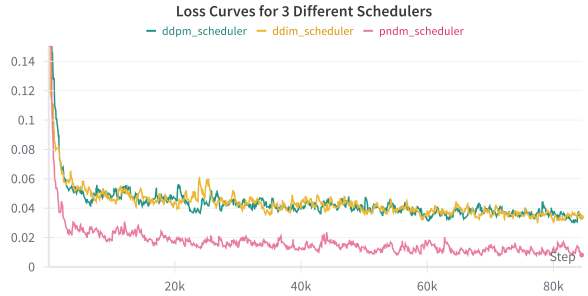

Fig. 11. Loss Curves for 3 Different Schedulers

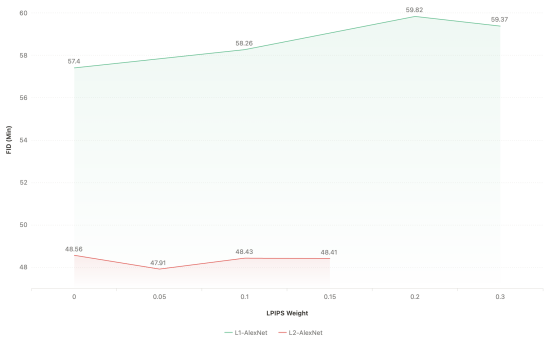

Fig. 12. LPIPS Weight vs. FID

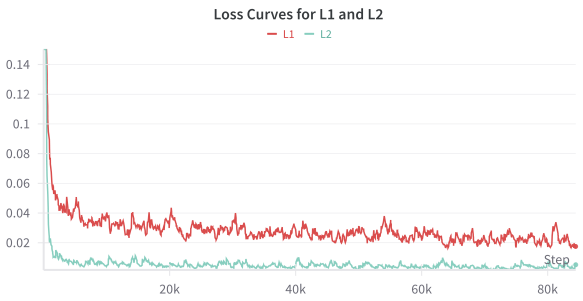

Fig. 13. Loss Curves for L1 and L2

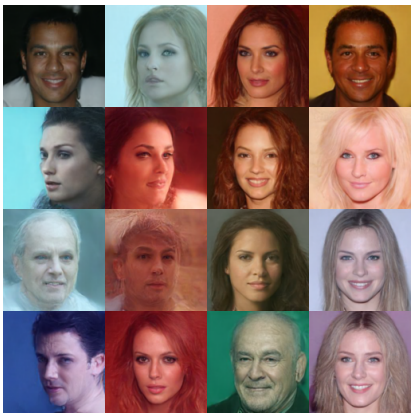

Fig. 14. Unconditioned Output

Fig. 15. Male Conditioned Output

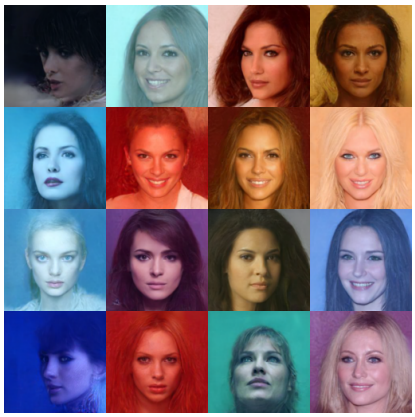

Fig. 16. Female Conditioned Output

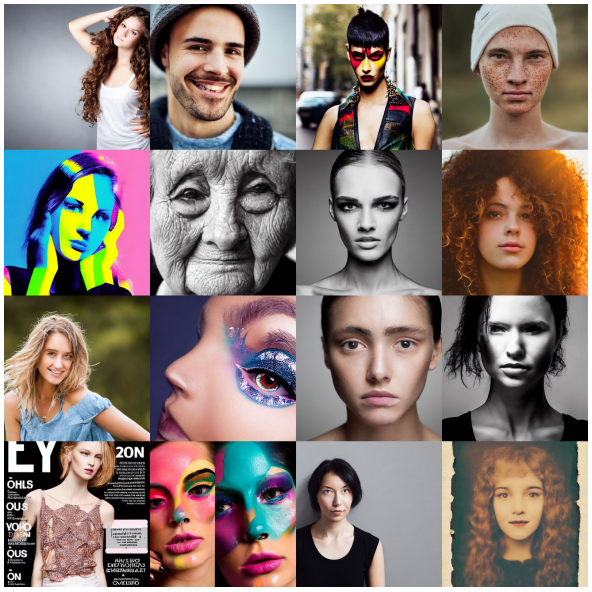

Fig. 17. Fully Fine-tuned with pre-trained "stable-diffusion-v1-5" model

Fig. 18. Our own fine-tuned model with the first three layers frozen

A. *Image Caption Descriptions*

1) Portrait of a young woman with long wavy hair, soft studio lighting, high contrast, 4k resolution, professional headshot.

2) Close-up of a smiling man with sharp jawline, cinematic lighting, shallow depth of field, bokeh background.
3) High-fashion model with dramatic makeup, sharp cheekbones, intense gaze, in a vibrant urban setting.
4) Natural portrait of a person with freckles, soft lighting, unretouched, authentic expression.
5) Retro 80s style portrait, neon colors, grainy texture, bold shadows, high contrast.
6) Black and white portrait of an elderly woman with wrinkles, deep shadows, textured background.
7) Fashion editorial headshot, strong jawline, clean background, high key lighting, Vogue style.
8) Glamour shot of a person with curly hair, golden hour lighting, soft focus, warm tones.
9) Candid portrait, natural light, slight smile, outdoor background, wind-blown hair.
10) Close-up of a person with expressive eyes, intricate makeup, glowing skin, ethereal lighting.
11) Headshot of a person in a studio, neutral background, intense stare, perfectly lit skin.
12) Profile shot, high-contrast lighting, dramatic shadow play, fine art portrait style.
13) Fashion magazine cover style, high-definition, sharp details, confident expression.
14) Editorial beauty portrait, colorful makeup, symmetrical face, close-up, glossy finish.
15) Minimalist headshot, natural light, smooth skin texture, calm expression.
16) Vintage portrait, faded colors, slight grain, nostalgic 70s film effect, natural hair.

Fig. 19. Ablation Diffusion Model using DDPM linear schedule

Fig. 20. Latent Diffusion Model with DDIM using scaled linear schedule

APPENDIX B
TABLES

Channels	Depth	Heads	Attention Resolutions	BigGAN up/down sample	FID (500K)	Baseline Comparison
128	2	1	16	×	70.57	-
128	4	1	16	×	63.84	-6.73
96	4	1	16	×	69.76	-0.81
64	4	1	16	×	74.72	+4.15
128	2	4	16	×	63.63	-6.94
128	2	1	32,16,8	×	65.07	-5.5
128	2	1	16	✓	68.72	-1.85
128	2	4	32,16,8	✓	**62.87**	**-7.70**

TABLE I: U-Net Ablation

Attention Resolution	Heads	Channels per Head	FID (500K)	Baseline Comparison
16	4	64	63.63	-
16	2	128	63.75	+ 0.12
16	8	32	**61.86**	**-1.01**

TABLE II: U-Net Heads Ablation

Model Architecture	Commitment Loss Weight	Reconstruction FID	Baseline Comparison
VQModel	0.1	30.91	-
VQModel	0.3	30.69	-0.22
VQModel	0.35	30.71	-0.20
VQModel	0.4	29.20	-1.70
VQModel	0.5	**29.36**	**-1.55**

TABLE III: VQ-VAE Ablation Experiments

Model Architecture	VAE Model	FID Score	Baseline Comparison
4 Block UNet	VQ-VAE(loss-weight-0.4)	67.05	-
5 Block UNet	VQ-VAE(loss-weight-0.4)	56.92	-10.13
6 Block UNet	VQ-VAE(loss-weight-0.4)	57.60	-9.45
5 Block UNet	VQ-VAE(pre-train-model)	**53.35**	**-13.70**

TABLE IV: LDM Model Architecture Ablation

Model Architecture	KL Loss Weight	Reconstruction FID	Baseline Comparison
AutoencoderKL(4 Block)	0.5	69.7	-
AutoencoderKL(4 Block)	0.2	63.55	-6.15
AutoencoderKL(4 Block)	0.1	58.95	-10.75
AutoencoderKL(4 Block)	0.05	52.55	-17.14
AutoencoderKL(3 Block)	0.05	**38.41**	**-31.29**

TABLE V: VAE Architecture Ablation

Model Architecture	VAE Model	Guidance Scale	FID	Baseline Comparison
DiT_B_2	VAE(loss-weight-0.02)	1	95.79	-
DiT_B_2	VAE(loss-weight-0.02)	1	84.29	-11.5
DiT_B_2	VAE(loss-weight-0.1)	4	84.40	-11.39
DiT_B_2	VAE(loss-weight-0.1)	0	75.09	-20.70
DiT_B_2	VAE(loss-weight-0.1)	1	74.38	-21.41
DiT_B_2	VAE(loss-weight-0.05)	1	**68.25**	**-27.53**

TABLE VI: DiT Model Architecture Ablation

Data Augmentation	FID (500K)	Baseline Comparison
Baseline	61.26	-
Center Crop	51.69	-9.57
Random Horizontal Flip	49.63	-11.62
Gaussian Blur	52.66	-8.59
Random Horizontal Flip + Gaussian Blur	52.24	-9.01
Random Horizontal Flip + Center Crop	49.62	-11.64
Random Horizontal Flip + Center Crop (4K Steps)	**48.32**	**-12.94**

TABLE VII: Data Augmentation

Scheduler	Training Steps	FID
DDPM	1000	61.26
DDPM	2000	52.76
DDPM	**4000**	**48.56**

TABLE VIII: Training Steps vs. FID for DDPM

Scheduler	Beta Schedule	Training Steps	FID	Baseline Comparison
DDPM	**Linear**	1000	**61.26**	-
DDPM	Scaled Linear	1000	67.04	5.79
DDPM	Cosine	1000	70.70	9.46

TABLE IX: Beta Schedule vs. FID

Scheduler	Beta Schedule	Inference Steps	FID	Baseline Comparison
DDIM	Linear	100	59.94	-
DDIM	Scaled Linear	100	55.77	-4.16
DDIM	Cosine	100	**52.63**	**-7.30**

TABLE X: DDIM Beta Schedule vs. FID

Scheduler	Beta Schedule	Inference Steps	FID	Baseline Comparison
PNDM	**Linear**	50	**56.79**	-
PNDM	Scaled Linear	50	58.25	1.46
PNDM	Cosine	50	71.12	14.33

TABLE XI: PNDM Beta Schedule vs. FID

η	FID	IS	Baseline FID Comparison
0.0 (baseline)	52.63	2.40 ± 0.22	-
0.25	55.3	2.33 ± 0.17	2.67
0.5	**50.8**	**2.29 ± 0.22**	**-1.83**
0.75	52.19	2.27 ± 0.32	-0.51

TABLE XII: η vs. FID

Pipeline	Prediction Type	Zero SNR	FID (500K)	Baseline FID Comparison
DDIM	ϵ	No	52.63	-
DDIM	v	Yes	**49.83**	**-2.80**

TABLE XIII: Prediction Type and Zero SNR vs. FID

Loss Function	LPIPS Enabled	LPIPS Network	LPIPS Weight	FID (500K)	Run Time	Baseline Comparison
L2	No	-	0	48.56	15h 23m 34s	-
L2	Yes	VGG-16	0.1	49.23	15h 11m 51s	0.67
L2	Yes	AlexNet	0.1	**48.43**	14h 56m 27s	**-0.13**
L1	No	-	0	57.4	15h 25m 8s	8.84
L1	Yes	VGG-16	0.1	59.84	14h 46m 28s	11.28
L1	Yes	AlexNet	0.1	58.26	14h 58m 11s	9.70

TABLE XIV: Loss Function vs. FID

Condition	Guidance Scale	IS	FID (500K)
None	-	2.08	62.87
Male	0.5	2.14	70.78
Female	0.5	2.17	83.37

TABLE XV: Conditioning On Male vs. Female Label

Group Member	**Contribution Score**	**Description**
Frank Lu	**32**	I completed the literature review section and was responsible for overall project management, planning, and task delegation. I also conducted architecture ablation experiments and designed the layout for all experiments. Additionally, I fine-tuned a LoRA Stable Diffusion model and implemented classifier-free guidance. I refactored and maintained the majority of the codebase while reviewing pull requests from other teammates.
Kaan Emre Sanal	**18**	I contributed to args.py and handled the Sections III.B, IV.B.1, and IV.B.2. I ran several experiments and handled most of the LaTeX documentation. I acknowledge that my contributions were limited. My coding skills need improvement, and I faced challenges due to falling behind on lectures and experiencing a hectic Easter break.
Ziyu Wang	**18**	I mainly completed the sections on DDIM, v-prediction and zero SNR in hyperparameter tuning, as well as the loss function section. My contribution was limited due to gaps in my fundamental knowledge and practical skills regarding diffusion coding.
Xiaoguang Liang	**32**	I primarily authored the literature review section, the LDM and DiT components of the methodology section, and conducted experiments with LDM, DiT, and Stable Diffusion. I also contributed to developing the core functionality of the project.

TABLE XVI: Contribution Table