
Personalized Image Generation Based on Diffusion Models

Abstract

This paper explores personalized image generation using diffusion models, specifically focusing on fine-tuning the FLUX.1-dev model through parameter-efficient methods. We conducted comprehensive experiments to optimize hyperparameters such as learning rate, LoRA rank, and training steps, evaluating outcomes with Face Distance Scores and CLIP Scores alongside qualitative analyses. Advanced analyses including ablation studies, interactive Gradio demonstrations, and multi-person fine-tuning further validated our approach. Our results demonstrate significant improvements in personalized generation quality, highlighting critical factors influencing model performance and underscoring effective methodologies for achieving high-fidelity personalized outputs.

1 Introduction

Recent advancements in generative models, particularly diffusion models, have demonstrated significant capabilities in image synthesis, generating high-quality visuals across diverse scenarios. However, conventional diffusion models exhibit notable limitations in addressing personalized image generation tasks, particularly in consistently reproducing specific individual identities or stylistic nuances across varying contexts [1]. This limitation restricts their practical applicability, especially in personalized entertainment, e-commerce, and social media sectors, which increasingly demand highly individualized content generation capabilities.

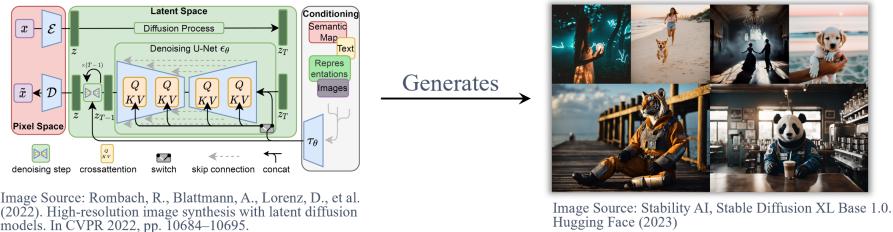


Figure 1: Architecture of the diffusion model and sample outputs.

Motivated by recent developments such as GPT-4o's remarkable text-to-image capabilities and inspired by this semester's deep learning course encompassing Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, Generative Adversarial Networks (GANs), Graph Neural Networks (GNNs), and particularly the innovative Diffusion Models, this work aims to explore the personalization capabilities of diffusion-based generative models.

*Equal contribution.

Specifically, our project focuses on fine-tuning the FLUX.1-dev diffusion model, selected after evaluating multiple open-source foundational models based on their popularity and performance metrics, including likes, downloads, and leaderboard rankings. Our personalization target is the renowned football player Cristiano Ronaldo (CR7), using multiple high-resolution images covering diverse angles and lighting conditions, sourced from publicly accessible platforms with no copyright restrictions.

The contributions of our work are multifaceted. Firstly, we systematically explore hyperparameter spaces, including learning rates, LoRA ranks, and training steps, through comprehensive experiments to optimize model performance. Secondly, our technical approach rigorously adheres to software engineering principles, particularly modularity and information hiding, resulting in structured and maintainable project organization with automated multi-configuration training capabilities and detailed error handling mechanisms. Thirdly, we implement and evaluate personalized image generation using quantitative metrics such as Face Distance Score and CLIP Score, alongside qualitative analyses.

Additionally, we provide an interactive Gradio demo, facilitating intuitive exploration and evaluation of personalized generation outputs. This not only demonstrates our methodological effectiveness but also significantly enhances user engagement and practical applicability. Despite encountering challenges such as substantial computational resource requirements and inherent limitations in existing model architectures, our approach successfully demonstrates the feasibility and effectiveness of personalized image generation with diffusion models.

In summary, our work advances the personalization capabilities of diffusion models by integrating rigorous experimentation, principled engineering practices, and interactive evaluation, thereby contributing both methodologically and practically to the field of personalized generative modeling.

2 Related Work

2.1 Diffusion Models

Diffusion models represent a class of generative models that progressively denoise a noisy input image over a series of steps to produce high-quality outputs [2]. These models learn the data distribution through a self-supervised approach, leveraging image-text pairs without explicit human labeling. Their training objective centers around reconstructing clean images from noisy versions, effectively learning the inherent structural distribution of the data rather than explicit semantic labels.

2.2 FLUX.1-dev

FLUX.1-dev extends conventional diffusion models by integrating advanced multimodal architectures. It combines CLIP and T5 encoders to extract semantic meaning from prompts, processing this information through multi-layer MM-DiT and Single-DiT Transformer blocks. MM-DiT blocks handle multimodal conditions such as image latent representations and textual prompts, while Single-DiT blocks focus solely on latent diffusion modeling. These modules jointly manage temporal, spatial, and conditional data, utilizing modulation mechanisms for efficient latent synthesis. The VAE decoder finalizes the generation by mapping latent representations back into high-resolution images, employing key techniques such as transposed convolutions, activation functions like Swish and GELU, and batch normalization. FLUX thus advances from abstract semantic fusion to detailed pixel-level refinement, enabling superior semantic control and image generation quality.

2.3 LoRA (Low-Rank Adaptation)

LoRA offers a parameter-efficient fine-tuning technique designed to overcome the computational challenges of full-parameter tuning [3]. Exploiting the concept of intrinsic dimensionality, LoRA decomposes weight increments into two smaller, low-rank matrices. Instead of adjusting the entire weight matrix $W_0 \in \mathbb{R}^{m \times n}$, LoRA fine-tunes by modifying matrices $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$, significantly reducing the number of parameters and computational overhead.

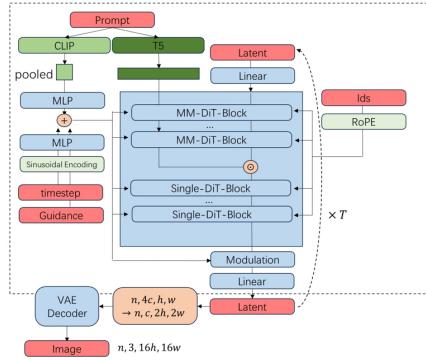


Image Source: Zhou Yifan Blog, 2024,
<https://zhouyifan.net/2024/09/03/20240809-flux1/>

Figure 2: Architecture of FLUX.1-dev model.

2.4 Personalization Techniques

Beyond LoRA, other notable personalization approaches include DreamBooth, which adapts diffusion models to consistently generate a target individual within diverse contexts by explicitly modeling the subject in specific scenarios. ControlNet further extends capabilities by incorporating additional condition signals into the diffusion model, facilitating enhanced controllability over the generated output. These methods, supported by Diffusers' official scripts, collectively provide a robust framework for personalized image generation tasks.

Together, these advancements underpin our project's methodological choices and experimental setup, enabling a comprehensive investigation of personalized image generation within diffusion model frameworks.

3 Method

3.1 Model Selection

We systematically evaluated four prominent diffusion models: black-forest-labs/FLUX.1-dev, stabilityai/stable-diffusion-xl-base-1.0, runwayml/stable-diffusion-v1-5, and CompVis/stable-diffusion-v1-4. The evaluation criteria included download counts, star ratings, community recommendations, fundamental capabilities, compatibility with fine-tuning and inference enhancement techniques, and hardware constraints. After comprehensive consideration, FLUX.1-dev was selected due to its superior multimodal semantic capabilities and compatibility with advanced fine-tuning methods. However, our computational resources ultimately limited the fine-tuning method exclusively to LoRA.



Figure 3: Comparison of Candidate Base Diffusion Models.

3.2 Data Preprocessing

We sourced multiple high-resolution images of the target individual from Pngfre, ensuring diversity in angles and lighting conditions without background and copyright restrictions. These images were subsequently cropped to a uniform resolution of 1024×1024 pixels using the Birme platform, aligning with the resolution requirements for model fine-tuning.

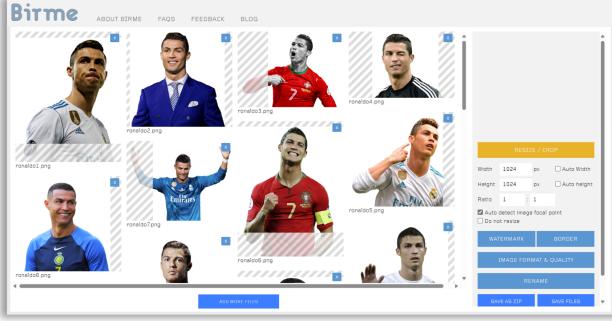


Figure 4: Data Preprocessing – BirMe Cropping Interface.

3.3 Inference Parameter Experiment

To ensure optimal image generation quality and streamline subsequent hyperparameter experimentation, we initially experimented with the inference parameter `guidance_scale`, which balances the adherence to the prompt versus creativity in the output. We tested values of 3.5, 5, and 7.5 using prompts such as `cr7man wears a tank top and blazer` and `cr7man smiling at camera`. Empirical evaluations indicated the optimal guidance scale was 5, providing an ideal balance between likeness to the original subject and image detail clarity.

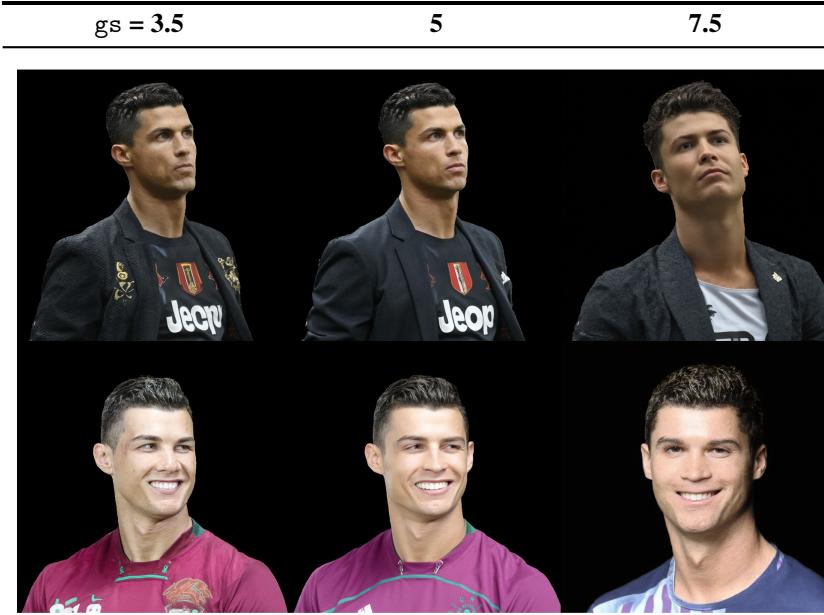


Figure 5: Example of Inference Parameter Experiments.

3.4 Hyperparameter Experiments

Considering the official training script constraints where `lora_alpha` is hardcoded to match `lora_rank`, we focused on three key hyperparameters:

Table 1: Hyperparameter configurations by experimental scenario

Scenario	Exp.	LR	LoRA Rank	Steps	Scheduler
Extreme parameters	1	1e-3	8	500	cosine
	2	1e-4	32	1500	cosine
	3	5e-4	16	1000	cosine
Fixed-step variations	4	1e-3	8	1000	cosine
	5	1e-4	32	1000	cosine
	8	1e-3	32	1000	cosine
Step count effects	9	1e-4	8	1000	cosine
	6	5e-4	16	500	cosine
	7	5e-4	16	1500	cosine
Expressiveness vs. fast training	10	1e-3	32	500	cosine
LR scheduler impact	11	5e-4	16	1000	constant

- **Learning rate** (`learning_rate`): Influences convergence speed and risk of overfitting.
- **LoRA rank** (`lora_rank`): Determines model capacity and complexity.
- **Maximum training steps** (`max_train_steps`): Directly affects training duration and model saturation.

We defined manageable parameter ranges as 1e-3, 5e-4, 1e-4, 8, 16, 32, and 500, 1000, 1500. Instead of performing exhaustive grid searches (27 combinations), we strategically reduced the set to 11 representative experiments inspired by branch coverage, condition coverage, and Modified Condition/Decision Coverage (MCDC) testing techniques. This set included extreme cases (fastest vs. strongest representation), combinations prioritizing either learning speed or representational capacity, isolated effects of training steps, and specific investigations into learning rate schedulers (cosine versus constant).

Notably, our hyperparameter experimentation benefits from:

- A unified JSON configuration for consistency, controlled through separate execution scripts to decouple experiment management.
- Utilization of subprocesses for training execution, enabling real-time streaming and logging of training outputs via pipe connections directly to the console.

3.5 Evaluation Metrics

Our evaluation employed multiple metrics to quantify image generation fidelity:

Face Distance Score: Utilizing the `facenet_pytorch` library, we first employed Multi-task Cascaded Convolutional Networks (MTCNN) for face detection and cropping. Subsequently, InceptionResnetV1 extracted face embedding features. The Face Distance Score was computed as the Euclidean distance between the mean embedding vector of fine-tuning images (benchmark) and the mean embedding vector of generated images from each hyperparameter set.

CLIP Score: To quantify image-text consistency, the CLIPProcessor preprocessed both images and textual prompts uniformly, and the CLIPModel calculated their similarity scores. The mean similarity score for generated images across various prompts provided the CLIP Score, reflecting semantic alignment quality.

Number of Faceless Images: This metric was assessed visually, counting the images that failed to accurately depict recognizable faces.

Collectively, these methodological approaches provided robust and insightful analyses to optimize personalized image generation effectively.

Algorithm 1 Face Distance Score

```
1: Input: target dir  $D_{tgt}$ , experiment dir  $D_{exp}$ 
2: Initialize MTCNN detector and InceptionRes-
   netV1 on device
3:  $\bar{f} \leftarrow \text{MeanEmbedding}(D_{tgt})$ 
4: for all image  $x$  in  $D_{exp}$  do
5:    $f_x \leftarrow \text{ExtractEmbedding}(x)$ 
6:    $d_x \leftarrow \|\bar{f} - f_x\|_2$ 
7: end for
8: return FDScore  $\leftarrow \frac{1}{|D_{exp}|} \sum_x d_x$ 
```

Algorithm 2 CLIP Score

```
1: Input: prompts  $P$ , experiment dir  $D_{exp}$ 
2: Initialize CLIPProcessor & CLIPModel on
   device
3:  $S \leftarrow \emptyset$ 
4: for all prompt  $p$  in  $P$  do
5:   for all image  $x$  in  $D_{exp}$  do
6:     preprocess  $(p, x)$  via CLIPProcessor
7:      $s_{p,x} \leftarrow \text{CLIPSimilarity}(p, x)$ 
8:      $S \leftarrow S \cup \{s_{p,x}\}$ 
9:   end for
10: end for
11: return CLIPScore  $\leftarrow \frac{1}{|S|} \sum_{s \in S} s$ 
```

4 Experiment

4.1 Experimental Results

We evaluated the performance of each hyperparameter configuration across 11 distinct prompts. For each set of hyperparameters, we computed the Face Distance Scores and the CLIP Scores, complemented by qualitative visual assessments. Through this rigorous comparative analysis, we identified the four optimal hyperparameter configurations based on superior quantitative and qualitative outcomes.

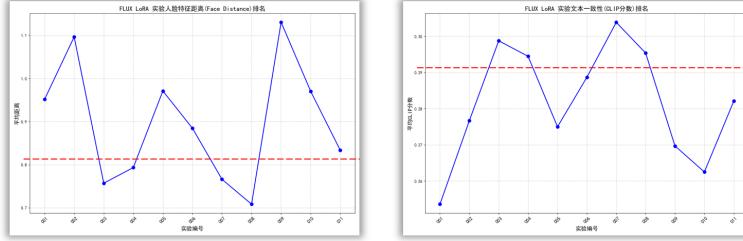


Figure 6: Evaluation Results.

4.2 Experimental Result Analysis

Learning Rate Analysis: By comparing experiments 4, 9, and 5, 8, we observed that under consistent LoRA rank conditions, a higher learning rate ($1e-3$) consistently achieved lower Face Distance Scores and higher CLIP Scores. This indicates that, given appropriate ranks and training durations, higher learning rates substantially outperform lower rates. Intermediate learning rates could mitigate their disadvantages through increased representational strength and extended training periods, whereas the lowest learning rates consistently yielded inferior results, regardless of other parameter adjustments.

LoRA Rank Analysis: Comparisons between experiments 4, 8 and 5, 9 revealed that at identical learning rates, higher LoRA ranks significantly improved Face Distance Scores. However, changes in CLIP Scores were marginal, likely due to the inherent strong cross-modal capabilities of the foundational model. Higher LoRA ranks thus enhance feature comprehension, significantly benefiting Face Distance outcomes.

Training Steps Analysis: Analysis indicated that, with learning rate set to $5e-4$ and rank at 16, model convergence typically occurred around 1000 steps, with no substantial improvements beyond this point. Conversely, at a lower learning rate ($1e-4$) and higher rank (32), prolonged training beyond optimal convergence points led to overfitting, adversely impacting Face Distance scores without significantly affecting CLIP Scores.

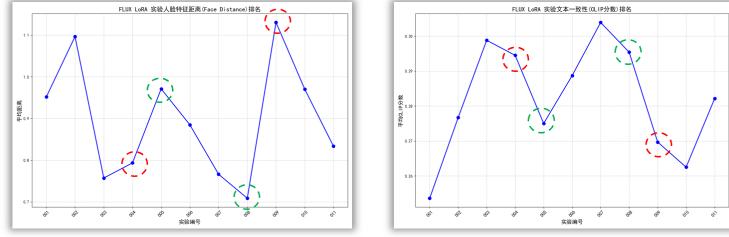


Figure 7: Evaluation of Learning Rate.

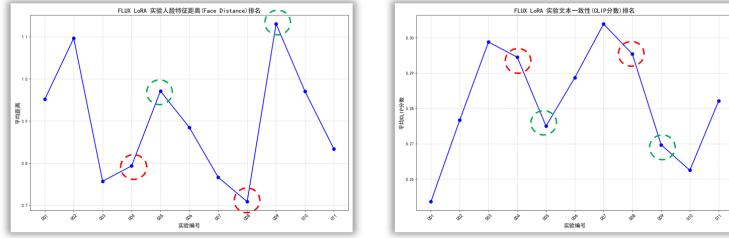


Figure 8: Evaluation of LoRA Rank.

4.3 Advanced Analysis: Ablation Study

Given the robustness and inherent knowledge of the FLUX model, we conducted an ablation study to ascertain whether model performance improvements stemmed from genuine learning of personalized features or internal token substitution strategies. Specifically, we removed the personalized training effect of the "cr7man" token and its referential meaning within the original model to evaluate the actual impact of fine-tuning. Results indicated substantial performance differences between models before and after fine-tuning with the "cr7man" token. Conversely, prompts explicitly naming "Cristiano Ronaldo" yielded minimal variation, confirming that our fine-tuning procedure effectively encoded personalized characteristics. Further validation was provided by introducing "David Tao" or "Tao Zhe," a figure unknown to the original model, demonstrating significant performance gains post fine-tuning and reinforcing the efficacy of our personalized fine-tuning approach.

4.4 Advanced Analysis: Gradio Demo

To enhance the interactivity and practical applicability of our approach, we developed a Gradio-based interactive demo. This demo allows dynamic loading of fine-tuned models, on-the-fly addition of training images, and immediate scoring of generated images using CLIP and Face Distance metrics, facilitating intuitive exploration and user-driven evaluations.

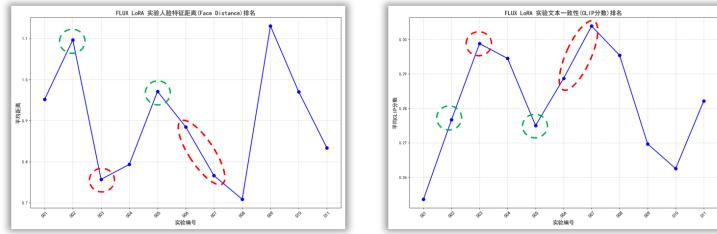


Figure 9: Evaluation of Training Steps.

4.5 Advanced Analysis: Multi-Person Group Photos

We extended our methodology by jointly fine-tuning the model on two distinct individuals, Cristiano Ronaldo and David Tao, subsequently merging these fine-tuned weights. The resulting capability to generate coherent multi-person group images demonstrated the robustness and flexibility of our personalization techniques, yielding visually compelling and realistic multi-individual compositions.

5 Conclusion

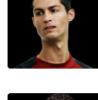
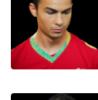
In this study, we successfully demonstrated personalized image generation leveraging the FLUX.1-dev diffusion model. Strategic selection and thorough optimization of hyperparameters, coupled with robust evaluation metrics, established clear insights into the impacts of learning rate, LoRA rank, and training steps. Ablation studies clarified the genuine learning outcomes attributable to our fine-tuning procedures, while advanced demonstrations such as Gradio interactivity and multi-person group photo generation showcased practical applicability and extensibility. This work confirms the efficacy of our personalized fine-tuning methodology and presents a solid foundation for future advancements in personalized image synthesis with diffusion models.

References

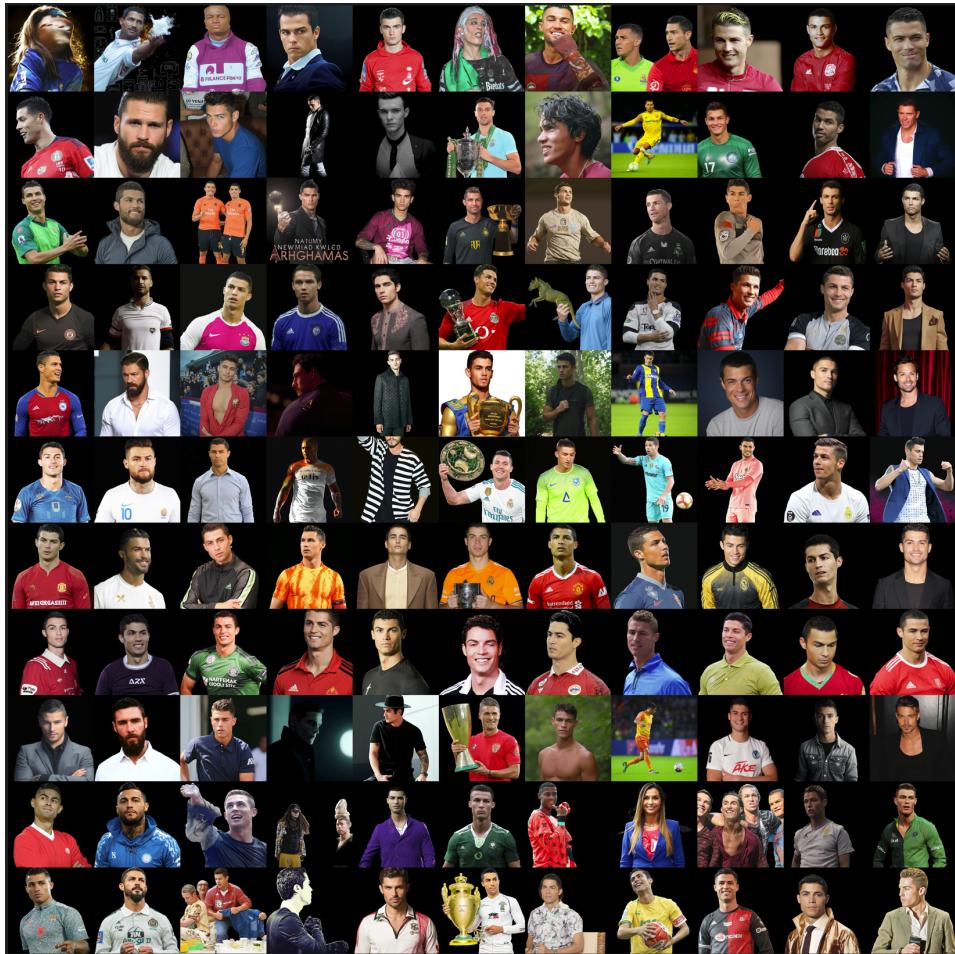
- [1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.
- [2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GENeral NEural SImulation System*. New York: TELOS/Springer-Verlag.
- [3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

A Appendix / supplemental material

Chosen Exps

Prompt	learning rate=5e-4 rank=16 steps=1000	learning rate=1e-3 rank=8 steps=1000	learning rate=5e-4 rank=16 steps=1500	learning rate=1e-3 rank=32 steps=1000
a photo of cr7man				
cr7man with a beard and a white shirt.				
cr7man in casual clothes				
cr7man, cinematic style, dramatic lighting, movie poster aesthetic, high contrast				
cr7man, fashion magazine style, editorial photography, stylish outfit, modern aesthetic				
cr7man holding a trophy				
cr7man outdoors in natural environment				
cr7man playing football on the field				
cr7man smiling at camera				
portrait of cr7man in studio lighting				
cr7man wears a tank top and blazer.				
Clip Score	0.2988	0.2945	0.3039	0.2954
Face Score	0.7571	0.7933	0.7661	0.7085

All Exps



B Early Diffusion Models: Basic Principles

Diffusion models are generative models that learn to reverse a gradual noising process. The forward diffusion process gradually adds Gaussian noise to data over T timesteps:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I)$$

where β_t is a variance schedule. The forward process can be written in closed form:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

where $\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$.

The reverse process learns to denoise:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

The training objective minimizes the variational lower bound:

$$L = \mathbb{E}_{x_0, \epsilon, t} [|\epsilon - \epsilon_\theta(x_t, t)|^2]$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.

C FLUX Model Architecture

FLUX represents a state-of-the-art text-to-image diffusion model that incorporates several architectural innovations:

Flow Matching Framework:

$$L_{FM} = \mathbb{E}_{t, x_0, x_1} [|v_\theta(x_t, t) - (x_1 - x_0)|^2]$$

with $x_t = tx_1 + (1 - t)x_0$.

Multi-Modal Architecture integrates:

- **Text Encoder:** T5-XXL
- **Vision Transformer:** DiT
- **Cross-Attention:**

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

D Low-Rank Adaptation (LoRA)

LoRA adds trainable low-rank matrices to adapt weights:

$$W = W_0 + BA$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$.

Forward pass:

$$h = W_0x + BAx$$

Advantages:

- Parameter efficiency: $r(d + k) \ll dk$
- Modular adaptation
- No extra inference cost when merged

E Textual Inversion

Textual Inversion learns a new pseudo-token v_* using:

$$L_{TI} = \mathbb{E}_{x,t,\epsilon} [|\epsilon - \epsilon_\theta(x_t, t, P(v_*))|^2]$$

Steps:

1. Initialize pseudo-word S_*
2. Freeze model θ
3. Optimize v_*
4. Embed in prompt: "A photo of S_* "

F DreamBooth

DreamBooth fine-tunes the whole model with:

$$L_{DB} = \mathbb{E}[|\epsilon - \epsilon_\theta(x_t, t, c)|^2] + \lambda \mathbb{E}[|\epsilon - \epsilon_\theta(x'_t, t, c_{pr})|^2]$$

Terms:

- Unique identifier (e.g., "sks")
- Class noun (e.g., "person")
- Prior preservation to prevent drift

G Custom Tokens

Extend embedding with:

$$E_{\text{new}} = [E_{\text{original}}; E_{\text{custom}}]$$

Initialization:

1. Random: $\mathcal{N}(0, \sigma^2 I)$
2. Based on similar tokens
3. Learned via Textual Inversion

Training:

$$L_{CT} = \mathbb{E}[|\epsilon - \epsilon_\theta(x_t, t, f(c, E_{\text{new}}))|^2]$$

H Prompt Tuning

Tune prompt embeddings $P \in \mathbb{R}^{L \times d}$:

$$L_{PT} = \mathbb{E}[|\epsilon - \epsilon_\theta(x_t, t, [P; E(c)])|^2]$$

Gradient:

$$P \leftarrow P - \alpha \nabla_P L_{PT}$$

Types: Hard (tokens) vs Soft (embeddings)

I ControlNet

Adds conditioning through spatial control:

$$y_c = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + \mathcal{Z}(c; \Theta_{z1}); \Theta_c); \Theta_{z2})$$

Training:

$$L_{CN} = \mathbb{E}[|\epsilon - \epsilon_\theta(x_t, t, c_{\text{text}}, c_{\text{control}})|^2]$$

Controls: Edges, depth, pose, segmentation.

J Token Merging (ToMe)

Reduce tokens by merging:

$$S_{i,j} = \frac{f_i^T f_j}{|f_i||f_j|}, \quad f_{\text{merged}} = \alpha f_i + (1 - \alpha) f_j$$

Reduction:

$$N_l = N_0 \cdot r^l, \quad r \in (0, 1)$$

K Prompt Mixing

Mix prompts for compositional generation:

Linear interpolation:

$$c_{\text{mix}} = \sum w_i c_i, \quad \sum w_i = 1$$

Attention mixing:

$$A_{\text{mixed}} = \sum w_i A_i$$

Classifier-free guidance:

$$\epsilon_{\text{mix}} = \epsilon_{\theta}(x_t, t, \emptyset) + \sum w_i s_i (\epsilon_{\theta}(x_t, t, c_i) - \epsilon_{\theta}(x_t, t, \emptyset))$$

Regional prompting:

$$\epsilon = \sum M_r \odot \epsilon_{\theta}(x_t, t, c_r), \quad \sum M_r = 1$$

Temporal mixing (video):

$$c_t = (1 - \alpha(t))c_{\text{start}} + \alpha(t)c_{\text{end}}$$

where $\alpha(t)$ controls transition timing.

References

- [1] Xulu Zhang, Xiaoyong Wei, Wentao Hu, Jinlin Wu, Jiaxin Wu, Wengyu Zhang, Zhaoxiang Zhang, Zhen Lei, and Qing Li. A survey on personalized content synthesis with diffusion models. *arXiv preprint arXiv:2405.05538*, 2024.
- [2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.