FREE-Switch: Frequency-based Dynamic LoRA Switch for Style Transfer

Supplementary Material

A. Notations

We first list the notations for key concepts in our paper.

Table 2. Notations.

Notations	Descriptions
f	Pre-trained diffusion model.
θ_c	Fine-tuning parameters for content.
θ_s	Fine-tuning parameters for style.
h_t	The output of t-th diffuison step.
$f(h_{t-1})$	The denoising process of t-th step using base model.
$f_\theta(h_{t-1})$	The denoising process of t-th step with LoRA θ.

B. Reproducibility

B.1. Base Model

SDXL v1.0. [21] Stable Diffusion XL (SDXL) v1.0 is a high-capacity latent diffusion model tailored for high-resolution text-to-image generation. It utilizes a dual-stage U-Net architecture coupled with an enhanced text encoder to improve semantic comprehension and prompt fidelity. In this work, SDXL v1.0 serves as the principal backbone for LoRA fine-tuning, providing strong expressivity for both concept and style learning. Its latent space supports fine-grained control over texture and geometry, making it suitable for analyzing adapter alignment and compositionality.

FLUX.1 [18]. FLUX 1 is a transformer-based diffusion architecture optimized for fine-grained detail and global photorealism. Compared with SDXL, it integrates cross-layer attention, offering complementary structural inductive bias. We include FLUX 1 to examine the generality of our frequency-domain dynamic adapter switching mechanism across distinct diffusion architectures. Using both SDXL and FLUX ensures that the observed performance improvements originate from the proposed adapter fusion strategy rather than model-specific characteristics.

B.2. Adapters

In this part, we describe the open-source adapters used in our study and how we obtained the partially trained adapters included in our experiments.

SDXL v1.0. For settings where SDXL v1.0 serves as the base model, we conduct evaluations using K-LoRA. Following the same training protocol and dataset used in RA, we train dedicated content and style LoRAs using the DreamBooth procedure, and employ them for subsequent combination tests. The detailed hyperparameters are provided in Tab. 3.

Table 3. Training Configuations for LoRA.

Parameter	Value
rank	64
resolution	1024
train_batch_size	1
learning_rate	5.00E-05
lr_scheduler	constant
lr_warmup_steps	0
max_train_steps	1000

FLUX 1. For settings based on FLUX, we obtain all content[1] and style[2] LoRAs directly from open-source repositories on HuggingFace, aligning with our goal of evaluating models in a fully open-source manner. Since different LoRAs require different prompt formats and trigger words, we adopt their default configurations without modification.

B.3. Baselines

Direct Generation. Generates images directly using the pretrained diffusion backbone without any adapter. This baseline reveals the intrinsic generative capacity of the base model and serves as a reference for measuring the benefit of adapter combination.

Joint Train. Trains content and style adapters in a single optimization stage to allow cross-domain feature interaction. However, such coupling often causes entanglement between semantic and stylistic factors, making it difficult to maintain distinct semantic and stylistic roles. Moreover, simultaneous optimization of multiple adapters increases training cost, reducing its suitability for lightweight or modular customization.

ZipLoRA. ZipLoRA [26] merges multiple LoRAs through interleaving rank components into a unified structure. It achieves parameter compression and moderate fusion quality but lacks dynamic awareness of diffusion-step differences, leading to degraded details in complex scenarios.

Merge. This baseline performs simple arithmetic merging of individually trained LoRA weights, typically via weighted averaging of corresponding matrices. Although computationally efficient, it lacks semantic adaptivity, of-

[1] https://huggingface.co/DeZoomer/ScarlettJohansson-FluxLora
https://huggingface.co/wanghaofan/Black-Myth-Wukong-FLUX-LoRA
https://huggingface.co/fofr/flux-mona-lisa
https://huggingface.co/ginipick/flux-lora-eric-cat

[2] https://huggingface.co/lucataco/ReplicateFluxLoRA
https://huggingface.co/dataautogpt3/FLUX-AestheticAnime
https://huggingface.co/multimodalart/flux-tarot-v1
https://huggingface.co/alvdansen/softserve_anime

ten resulting in content distortion or style dilution. It is also important to note that our task involves only two LoRAs to be fused. Therefore, existing merge-based approaches that aim to mitigate conflicts among multiple LoRAs offer limited benefits in our setting, and their performance becomes largely comparable to a simple weighted merge.

K-LoRA. K-LoRA [19] is a training-free LoRA fusion approach that adaptively integrates subject and style LoRAs without requiring additional fine-tuning. It operates by introducing a Top-K selection mechanism within attention layers, which identifies the most salient components from content and style LoRAs and dynamically combines them at each diffusion timestep. This strategy enables effective preservation of both subject and stylistic features while maintaining model stability.

B.4. Details

We discuss the computational details of the experiments. All experiments are conducted on NVIDIA RTX 3090 GPUs, except for those involving FLUX.1, which are run on NVIDIA A800 GPUs.

B.5. Details of Generation Alignment

Box B.1: System Prompt for Content Alignment.

You are a concept extractor specialized in analyzing MULTIPLE images of {class_name}
CORE TASK: Extract ONLY the SHARED core subject content across all concept images (ignore unique details from single images).
MANDATORY INCLUSIONS:
- Shared subject identity (clear category of the main subject, e.g., 'ceramic teapot').
- 3+ shared key features (form, structure, pose, unique traits present in ALL images, e.g., 'domed lid, curved spout').

STRICT EXCLUSIONS:
- NO style elements (color, texture, lighting, brushwork — these belong to style descriptions).
- NO background, environment, props, text, or camera-related details (e.g., 'indoors', 'close-up').
- NO unique details from individual images (e.g., a pattern only in one image).

STRICT TOKEN LIMIT: The output must be $\geq$\{concept_token_limit\} tokens.
OUTPUT FORMAT: Single line in the structure:
'[shared subject identity] with [shared feature1], [shared feature2], [shared feature3+]'
Example: 'Ceramic teapot with domed lid, curved spout, C-shaped handle.

Box B.2: User Prompt for Content Alignment.

Analyze the provided MULTIPLE Concept Images of '{class_name}'
Follow these steps:
1. Identify the SHARED main subject (present in all images).
2. Extract 3+ key features that appear in ALL images (ignore features unique to single images).
3. Describe them in the required format, excluding all style elements, background, and unique details.
- The output must be STRICTLY $\leq$\{concept_token_limit\} tokens.
- Prioritize retaining 3+ key features over redundant words if approaching the limit.
Output ONLY the following line (no extra text):
[shared subject] with [feature1], [feature2], [feature3+]

Box B.3: System Prompt for Style Alignment.

You are a style extractor specialized in analyzing a single {style_name} image.
CORE TASK: Extract ONLY pure visual style elements from the style image (ignore any subject/content details).
MANDATORY INCLUSIONS:
- Artistic medium (clear type of art form, e.g., 'watercolor painting', 'crayon drawing').
- 3+ visual style elements (must be from color palette, lighting, texture/brushwork, mood — e.g., 'soft blue palette, textured brushstrokes').

STRICT EXCLUSIONS:
- NO subject content (objects, structure, form, or any elements related to concept themes).
- NO redundant descriptions unrelated to visual style (e.g., 'popular style', 'modern design').

STRICT TOKEN LIMIT: The output must be $\leq$\{style_token_limit\} tokens.
OUTPUT FORMAT: Single line in the structure:
'[artistic medium] with [style element1], [style element2], [style element3+]'
Example: 'Watercolor painting with soft blue palette, textured brushstrokes, warm ambient lighting, dreamy mood'.

In this part, we describe how we leverage a Vision-Language Model (VLM) for generation alignment. Our alignment strategy is primarily achieved through a prompt-refinement scheme, where Qwen3-VL-Plus is used as the multimodal model to extract and enrich visual information. The prompts applied in our workflow are provided in Boxes

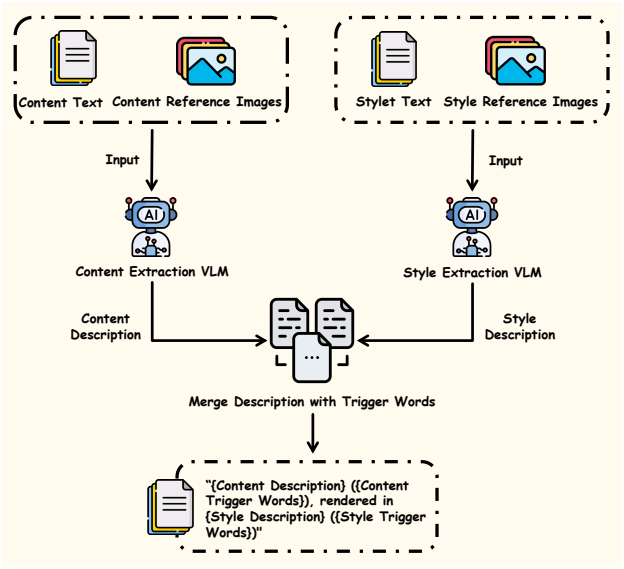

Figure 11. The Pipeline of Generation Alignment.

B.1–B.4. Specifically, we feed the VLM with both the original class name and its corresponding reference image. The VLM then extracts the most salient semantic cues, which are further filtered as illustrated in Fig. 11, and finally used as the refined prompt for generation.

Box B.4: User Prompt for Style Alignment.

Analyze the provided single {style_name} image.
Follow these steps:
1. Identify the artistic medium (e.g., 'oil painting', 'digital illustration').
2. Extract 3+ pure visual style elements (focus on color, lighting, texture, mood — avoid any content).
3. Describe them in the required format, excluding all subject-related and non-style details.
The output must be STRICTLY $\leq$ {style_token_limit} tokens.
Prioritize retaining 3+ key style elements over redundant words if approaching the limit.
Use concrete, art-specific vocabulary (e.g., 'impasto texture', 'muted complementary palette', 'film-grain aesthetic').
Output ONLY the following line (no extra text):
"[artistic medium] with [style element1], [style element2], [style element3+]"

B.6. Details of Evaluation

In this section, we describe the details of the evaluation metrics used in our experiments.

Box B.5: Prompt for Gemini Feedback.

YOU ARE A STYLE TRANSFER EVALUATION EXPERT. YOUR SOLE OUTPUT MUST BE A SINGLE JSON OBJECT. DO NOT INCLUDE ANY INTRODUCTORY OR EXPLANATORY TEXT.
TASK: Assess the images provided in this API call and determine which Candidate Image achieves the most successful and balanced fusion of the "Content Reference Image's" subject structure and the "Style Reference Image's" artistic characteristics.
INPUT IMAGES:
- Content Reference Image (Defines subject, outline, and composition).
- Style Reference Image (Defines color, lighting, texture, and style).
- Candidate Images (The images to be compared and scored).

EVALUATION CRITERIA:
- Content Fidelity: The accuracy of the candidate image in preserving the subject's identity and shape from the Content Reference (Score 1-5).
- Style Fidelity: The accuracy of the candidate image in adopting the color scheme, texture, and artistic attributes from the Style Reference (Score 1-5). OUTPUT REQUIREMENTS: Provide a single JSON object with the following structure. The 'id' field must clearly reference the Candidate Image (e.g., Candidate_A, Candidate_1, or the filename if available).

{ "id": "Candidate_1_ID",
"content_fidelity_score": 0,
"style_fidelity_score": 0}

CLIP Score. Following the procedure in K-LoRA, this metric measures how well each method preserves style. We use a pretrained ViT-B/32 model to encode the generated image and the reference style image into normalized high-dimensional feature vectors. The cosine similarity between the two vectors is then computed to quantify their semantic closeness, which reflects the degree of style preservation.

DINO Score. This metric is computed in a manner similar to the CLIP Score and evaluates content preservation. The difference is that the encoder is replaced with DINOv2-Base while all other computations remain the same.

Gemini Feedback. For this metric, we query Gemini 2.5-Flash to determine which image among those generated by different methods best aligns with the specified combination of content and style. Gemini assigns a relevance score to each image, and the image with the highest score is selected. Over a batch of results, we compute the selection probability, where a higher probability indicates better gen-

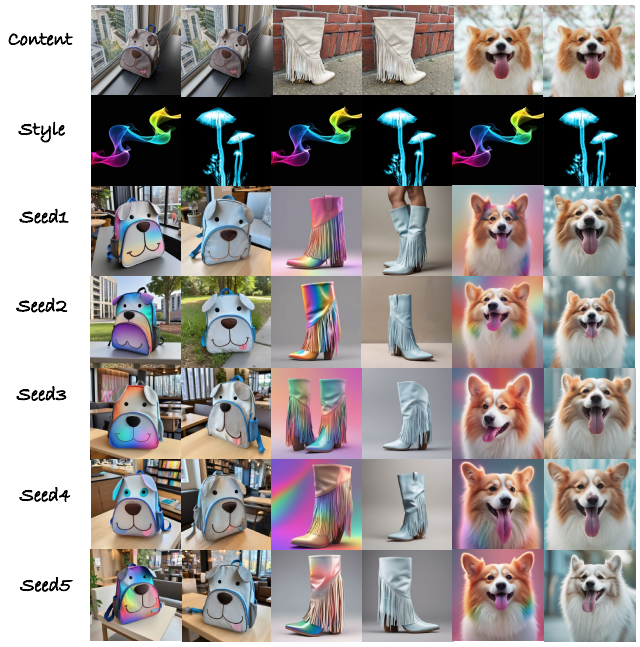

Figure 12. Failure case analysis using SDXL v1.0.

eration quality. The reference image and the target prompt are provided as inputs to the model.

Speed. To evaluate efficiency, we measure the time required to generate ten images for a given content and style pair on a single NVIDIA RTX 3090. For methods that involve training, this time includes both the training phase and the inference phase. For training-free methods, it primarily reflects inference latency. For our method, the total time consists of determining the importance of each LoRA across denoising steps during inference, VLM inference and generating the ten images of each pair.

C. Discussion

C.1. Failure Cases

In this section, we analyze several failure cases of our method. The goal is to identify its current limitations and provide insights for future methodological improvements. Representative failure examples are shown in Fig. 12. For example, the two styles shown in Fig. 12, namely "in abstract rainbow colored flowing smoke wave design" and "in glowing style," are highly abstract and differ significantly from typical content images. As a result, both styles exhibit varying degrees of style degradation during the training-free optimization process. We observe that, because our approach does not involve any training, it struggles with highly abstract styles. Generating a coherent combination of such styles and the target content requires a deep understanding of both the prompt semantics and the underlying generative trajectory. However, existing open-source weights may not possess this level of capability, which makes it difficult for training-free methods to achieve satisfactory results in these cases. This suggests that future work may benefit from incorporating lightweight, low-cost training mechanisms to address these limitations.

C.2. Future Works

Our analysis of failure cases highlights several promising research directions. Although our framework remains entirely training-free, certain abstract or concept-heavy styles appear to require a deeper semantic understanding than what current open-source weights can provide. Future work may explore lightweight or low-cost training strategies that enhance cross-style and cross-content reasoning without sacrificing the efficiency advantages of our approach. Another interesting direction is to design adaptive refinement modules that dynamically learn style–content interactions during inference, enabling more robust generation in scenarios with highly abstract or visually complex styles. Finally, integrating stronger or more specialized vision-language priors may further improve semantic alignment and mitigate the limitations.