Diff-Aid: Inference-time Adaptive Interaction Denoising for Rectified Text-to-Image Generation

Binglei Li[1,2] **Mengping Yang**[1,3†] **Zhiyu Tan**[1,3] **Qiang Xiang**[1]
Junping Zhang[1*] **Hao Li**[1,2,3*]

[1] Fudan University [2] Shanghai Innovation Institute [3] Shanghai Academy of AI for Science
blli24@m.fudan.edu.cn, {mengpingyang, jpzhang, lihao_lh}@fudan.edu.cn

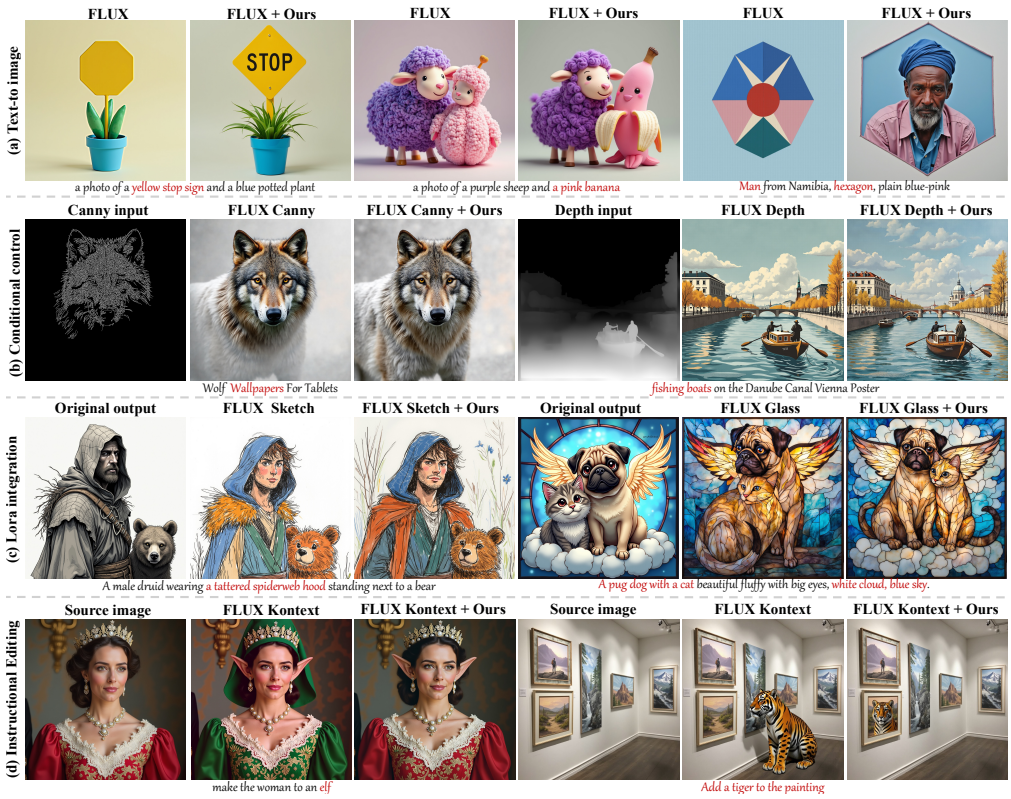


Figure 1: **We present *Diff-Aid* to adaptively improve the interaction between textual conditions and image latents during test time.** Benefited from our *plug-in* and *interpretable* design, *Diff-Aid* enables improved prompts following and synthesis quality for (a) text-to-image generation, (b) adding conditional input for controllable generation, (c) integrating off-the-shelf LoRAs and (d) zero-shot instructional image editing.

Abstract

Recent text-to-image (T2I) diffusion models have achieved remarkable advancement, yet faithfully following complex textual descriptions remains challenging due to insufficient interactions between textual and visual features. Prior approaches enhance such interactions via architectural design or handcrafted textual condition weighting, but lack flexibility and overlook the dynamic interactions across differ-

†Project Lead.
*Corresponding Authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

ent blocks and denoising stages. To provide a more flexible and efficient solution, we propose *Diff-Aid*, a lightweight module trained with a frozen backbone that predicts signed, per-token text modulation across transformer blocks and denoising timesteps at inference. Beyond improving generation quality, *Diff-Aid* yields interpretable modulation patterns that reveal how different blocks, timesteps, and textual tokens contribute to semantic alignment during denoising. As a plug-and-play module, *Diff-Aid* can be seamlessly integrated into downstream applications for further improvement, including style LoRAs, controllable generation, and zero-shot editing. Experiments on SD 3.5 and FLUX show improvements in compositional prompt following, visual quality, and blinded human preference. Our code is available at https://github.com/phil329/Diff-Aid.

1 Introduction

Diffusion models (DMs) [14, 43], which iteratively denoises random noisy samples into high-quality outputs, have revolutionized the experiences of AI generated content (AIGC) in various applications including text-to-image [30, 9], text-to-video generation [60, 45, 34, 50], joint audio-video synthesis [52, 61], *etc.* These breakthroughs mainly stem from the scalability of diffusion transformers (DiT) [35, 28] over conventional U-Net models [40, 8], as well as advances in large-scale training compute, infra and data pipelines. Yet, considering DMs as a model that maps textual conditions to visual outputs, they might often fail to produce images that accurately reflect textual details due to insufficient interactions between textual conditions and image features. For instance, the current top-performing model FLUX [22] failed to produce the "*yellow stop sign*" in Fig. 1 (a).

Many prior approaches have been developed to strengthen cross-modal interactions between text and image representations. On the architecture design hand, early methods rely on cross-attention mechanisms to inject textual information into diffusion models [40, 47], while DiT [35] proposed adaLN-zero strategy to condition transformer blocks on class conditions. The most recent multimodal diffusion transformers (MMDiT) based approaches [22, 44, 55] employ a shared unified transformer to jointly model both modalities, facilitating cross-modal interactions. However, training such models from scratch requires massive compute and data. On the other hand, classifier-free guidance (CFG) [15] adopts a static scaling factor to balance conditional and unconditional model outputs, thus improving conditioning strength. TACA [29] pointed out that semantic discrepancies may arise from the suppression of cross-modal attention due to the numerical asymmetry between the number of visual and text tokens and proposed a handcrafted weighting strategy to enhance textual tokens. Further, Li et al. [23] probed specific roles of transformer blocks in rendering textual conditions and attached block-specific enhancements to improve interactions. Yet, they require heuristic search of weighting strategies across the forward process, without explicitly considering the dynamic interactions between textual and visual features throughout the denoising process.

In this paper, we aim to develop a lightweight, plug-in, and adaptive method to dynamically adjust interactions between textual and visual features during inference time, thereby delivering high-quality output images with improved prompt following and human preferences. Note that *Diff-Aid* requires a lightweight training phase to learn adaptive weights, which is far more efficient than retraining the entire model. Fig. 2 illustrates a localized failure: vanilla FLUX omits "birthday" and shows a weak, spatially diffuse response to that token, whereas *Diff-Aid* recovers the text and a more focused response from the same prompt and initial noise. The block- and timestep-dependent attention patterns motivate an intervention before joint attention. Because prompt tokens also differ in visual relevance, we learn token-specific modulation rather than applying one global conditioning strength.

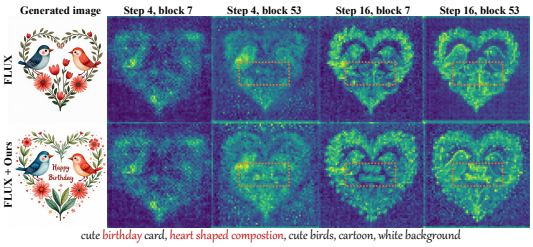


Figure 2: **Attention visualization of different blocks across denoising steps.** If the interaction between textual conditions and image latents is insufficient, the rendered image may fail to faithfully reflect given conditions.

Capitalizing on these observations, we propose *Diff-Aid*, an inference-time **A**daptive **I**nteraction **D**enoising module for text-to-image **Diff**usion models. "Inference-time" describes its dynamic use during sampling: the module is first trained with the backbone frozen, then predicts modulation

coefficients for each prompt and denoising step. Concretely, *Diff-Aid* introduces two key innovations: *1)* an *Aid* module, which explicitly adjusts textual conditions according to the varied blocks, timesteps, and tokens; and *2)* a gated sparsity and stabilized regularization mechanism that respectively ensure attaching more attention to important blocks and avoiding training collapse. In this way, the learned parameters are interpretable with respect to different blocks, varied timesteps, and textual tokens, further providing insights into the model's internal behavior (as illustrated in Fig. 4). More importantly, our design is lightweight with a small number of trainable parameters, without modifying the representation space of pretrained models. As illustrated in Fig. 1, as a plug-in module, *Diff-Aid* can be seamlessly integrated into homologous models for controllable image generation (b), styled LoRA generation (c), and instructional image editing (d), demonstrating its versatility and effectiveness. We evaluate our *Diff-Aid* on two strong baseline models (*i.e.,* SD 3.5 [9] and FLUX [22]), the consistent performance improvements across varied metrics and benchmarks demonstrate the effectiveness of our design. Further analysis reveals that *Diff-Aid* learns meaningful and interpretable interaction patterns aligned with the denoising dynamics, validating our design principles.

To sum up, our contributions are: *1)* We propose *Diff-Aid*, a lightweight model to adaptively manipulate the interactions between textual and visual features at inference-time, capturing the dynamic relationships across different blocks, timesteps, textual tokens, as well as the intrinsic properties of baseline models. *2)* Our *Diff-Aid* is designed as a plug-in and generalizable module that can be seamlessly integrated for downstream applications, including LoRAs, controllable generation, and zero-shot image editing. *3)* Extensive experiments across multiple models and evaluation metrics demonstrate consistent improvements in prompt following, visual quality, and human preferences.

2 Methodology

2.1 Motivation and Design Principles

Interactions between textual conditions and image latents for T2I. Text is the primary guidance for image generation in text-to-image diffusion models. The diffusion model is indeed a mapper $\mathcal{F}: \mathcal{P}_{txt} \rightarrow \mathcal{X}_{img}$, where the prompt is the only condition to generate the image. MMDiT models [9, 22] leverage joint self-attention to integrate text and image features:

$$\begin{aligned} \mathbf{z}_t^{l+1} &= \text{Softmax}\left(\mathbf{Q}\mathbf{K}^T/\sqrt{d}\right)\mathbf{V} + \mathbf{z}_t^l; \\ [\mathbf{Q}, \mathbf{K}, \mathbf{V}] &= W_{qkv} \cdot \text{LN}\left([\mathbf{z}_t^l, \mathbf{c}_t^l]\right). \end{aligned} \tag{1}$$

Therefore, effective interaction in attention between textual and visual tokens is vitally important for high-quality image generation. Fig. 2 shows two denoising trajectories of the same prompt and initial noise but different text-image interaction patterns. The attention maps of the second trajectory have a stronger focus on the relevant text token "birthday", leading to better text-image alignment.

Principle 1: Cross-modal interaction matters during denoising

Attention mechanism enables text-image interaction, which is crucial for high-quality image generation in Diffusion models. Transformer blocks exhibit varying interaction patterns over denoising steps.

Sufficient interactions lead to improved prompt adherence. Attention within each transformer block facilitates the integration of text and image features. As shown in Fig. 2 (second row), different blocks exhibit different patterns, where the 7th block focuses more on structure while the 53rd block emphasizes the "birthday" token. Therefore, we should consider a block-specific architecture to adaptively modulate interactions to leverage the unique capabilities of each block. Moreover, the denoising process in DiTs is indeed a PF-ODE solving procedure [28], where different timesteps contribute different guidance strengths from text features. Actually, Fig. 4(d) shows that attention norms differ across timesteps and different blocks. Thus, timestep-aware modulation is also necessary to manage the interaction strength dynamically. To mitigate the problem of imbalanced visual and textual tokens ($N_z \gg N_c$) [29], we introduce a token-level modulation mechanism for text features with different lengths. All these observations motivate our designs.

Principle 2: Per-token interaction across transformer blocks and denoising timesteps

Modulating text features at block-specific, timestep-aware, and token-level dimensions is essential and effective for enhancing text-image interactions.

Differences between our *Diff-Aid* and existing approaches. FiLM and adaLN provide general feature conditioning [36, 35], while *Diff-Aid* instead predicts a signed scalar for each text token, block, and denoising step, rescaling text states immediately before MMDiT joint attention. This changes the text keys and values seen by image queries while keeping the backbone weights frozen. LoRA [17] updates effective weights, whereas our learned feature modulation can also be combined with LoRA. Unlike TACA's prescribed, timestep-aware attention amplification [29], *Diff-Aid* learns which text signals to enhance or suppress. CFG instead adjusts sampling predictions [15], while *Diff-Aid* acts inside the denoiser and improves GenEval at all seven tested guidance scales.

Principle 3: Selective enhancement and suppression of text signals

Not all text signals contribute equally to text-image alignment. Adaptively enhancing informative signals while suppressing irrelevant ones enables effective control of cross-modal interactions without modifying the pretrained backbone.

2.2 The Proposed *Diff-Aid*

Adaptive interaction denoising (*Aid*) module. We propose *Aid* ϕ that learns the coefficient $\alpha_t^l \in \mathbb{R}^N$ to modulate text-image interactions based on the current diffusion timestep, transformer block, and text features.

$$\alpha_t^l = \phi(\mathbf{c}_t^l, t, l), \quad \alpha_t^l \in [-1, 1]^N, \tag{2}$$

where $\mathbf{c}_t^l$ represents the text features at timestep t and transformer block l, N is the number of text tokens. Then, we modulate the text features as:

$$\widetilde{\mathbf{c}}_t^l = \mathbf{c}_t^l + \mathbf{c}_t^l \odot \alpha_t^l, \tag{3}$$

where $\odot$ is element-wise multiplication. The modulated $\widetilde{\mathbf{c}}_t^l$ enters joint attention in block l: positive α_t^l strengthens a text-token signal, negative values suppress it, and zero preserves the original state. The image-token updates are affected through the resulting text keys and values; the backbone projections remain frozen. The *Aid* ϕ is implemented as a lightweight multi-layer perceptron with a gated mechanism and bounded activation functions (*i.e.*, $\tanh$) to ensure stability and effective modulation, as shown in Fig. 3.

Improving distribution sparsity across blocks. We expect the learned α_t^l to be sparse for two main reasons. *1)* Not all text tokens are equally important; some tokens carry more semantic information than others. For example, max sequence lengths of text tokens are 512 in FLUX and 333 in SD 3.5, but most prompts contain only approximately $50 \sim 200$ tokens, and the rest are often less informative, such as "PAD" tokens. *2)* Not all transformer blocks require strong text conditioning. Commonsense from previous works [2, 23] indicates that early blocks focus more on semantic and structural information, while later blocks refine visual details. Therefore, the α_t^l should be sparse, allowing the model to focus on the most relevant tokens and transformer blocks.

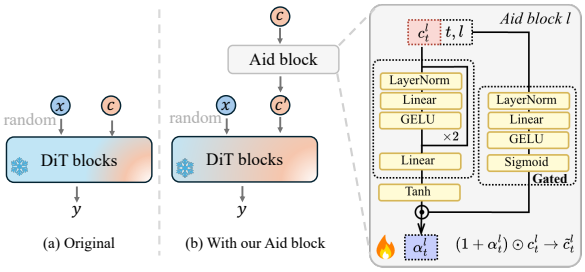


Figure 3: **Overview of our proposed *Aid* blocks.** (a) The original diffusion transformers map a random latent and input textual conditions to output features. (b) Our *Aid* blocks adaptively modulate textual conditions with respect to denoising timesteps across different transformer blocks, enhancing the interaction between image latents and textual conditions.

To achieve this, we facilitate two strategies to enable that the *Aid* ϕ learns an implicit and efficient α for block-specific and token-level modulation. Inspired by Qiu et al. [38], we apply a gated mechanism in *Aid* to promote sparsity:

$$\alpha_t^l = \phi(\mathbf{c}_t^l, t, l) = \tanh(\mathbf{c}_t^l W_l) \cdot \sigma(\mathbf{c}_t^l W_g), \tag{4}$$

where W_l represents the feature weights, and W_g represents the gating weights. The gate score $\sigma(\mathbf{c}_t^l W_g)$ effectively acts as a dynamic filter, controlling the strength of the text features by selectively allowing or blocking the modulation signal. Furthermore, we introduce a regularization term on the

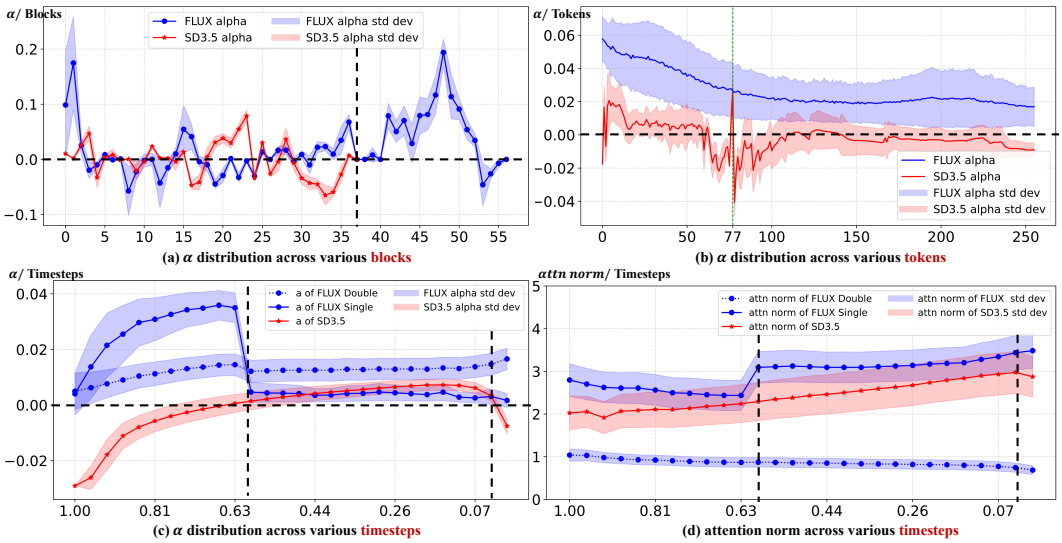


Figure 4: **Distributional visualization of the learned adaptive α across various (a) diffusion transformer blocks , (b) textual tokens and (c) denoising timesteps.** (d) Visualization of the attention norm across various timesteps. For (c) and (d), we separately show FLUX single-stream and dual-stream results to highlight architectural differences. Curves show means across $2,400$ held-out prompts while shaded bands show prompt-level standard deviations after averaging over the other displayed axes. Positive and negative values indicate enhancement and suppression rather than a universal trajectory for every prompt.

modulation coefficients, which penalizes large values of α_t^l:

$$\mathcal{L}_{reg} = \mathbb{E}_{t,l}\left[\|\alpha_t^l\|_2\right]. \tag{5}$$

2.3 Training and Inference

Aid modules are trained using the standard diffusion loss, which minimizes the squared error between the predicted velocity and the target velocity $(\epsilon - \mathbf{x})$:

$$\mathcal{L}_{diff} = \mathbb{E}_{t,\epsilon,\mathbf{x}}\left[\|v_\theta(\mathbf{z}_t, t, \widetilde{\mathbf{c}}_t) - (\epsilon - \mathbf{x})\|_2^2\right], \tag{6}$$

where v_θ is the velocity predicted by the diffusion model. During training, we freeze the backbone MMDiT model and only optimize the *Aid* ϕ using the combined loss:

$$\mathcal{L} = \mathcal{L}_{diff} + \lambda_{reg}\mathcal{L}_{reg}, \tag{7}$$

where λ_{reg} is the regularization hyperparameter. To make the optimization more stable and α_t^l more robust, we randomly skip the *Aid* modules during training with a probability of p. This mutes the entire *Aid* rather than individual neurons, akin to dropout.

During inference, the lightweight *Aid* modules are all active and inserted into each block. Other than the integration of our proposed *Aid*, the inference process remains unchanged. The module is trained beforehand; at inference it recomputes α_t^l from the evolving text states at each denoising step and block. Regarding downstream applications such as LoRAs and editing, our weights are directly loaded for zero-shot inference.

3 Experiments

Implementation details. We adopt FLUX.1-Dev [22] and SD3.5-Large [44] as baselines. Each *Aid* module consists of a two-layer MLP with hidden dimension $d = 2048$ and a gated mechanism. During training, we freeze all backbone parameters and only optimize the *Aid* modules, introducing only a 1.51% parameter increase over the baseline FLUX.1-Dev. For optimization, we use the Prodigy optimizer [32] with a learning rate of 1.0 and gradient accumulation of 8, yielding an effective batch size of 64. Training runs for $4{,}000$ steps (3.3 epochs) on 8 H100 GPUs(approximately 6 hours for FLUX and 4.5 hours for SD 3.5). We set the dropout probability $p = 0.1$, and $\lambda_{reg} = 0.1$. During



Figure 5: **Qualitative comparisons of images generated by baseline FLUX and SD 3.5 with and without our proposed *Diff-Aid*.** The quantitative results below each image shows the scores of HPSv3, Image Reward and Aesthetic. Our method demonstrates superior prompt adherence and image quality. More qualitative results are given in appendix.

inference, all the *Aid* modules are activated. We sample 1024×1024 images using the EDM [21] sampler for 28 steps, with CFG scales of 7.0 (SD3.5) and 3.5 (FLUX).

Datasets and evaluation metrics. We train on a subset of HPDv3 [31] comprising $77,252$ text-image pairs, filtered by preference confidence and image source (Appendix B.4). We evaluate *Diff-Aid* on the full HPDv3 benchmark for image quality and human preference alignment, as well as GenEval [10] for semantic understanding. In addition, we assess the generalizability of *Diff-Aid* using out-of-distribution metrics such as HPSv2 [56], ImageReward [59] and Aesthetic Score [41]. In Sec. 3.4, we demonstrate *Diff-Aid*'s compatibility with domain-specific LoRA and ControlNet modules using subsets of HPDv3, and evaluate its performance on editing tasks with EditBench [25].

3.1 Empirical Observation of Learned α

A key contribution of our work is that the learned α exhibits strong interpretability, highly correlating with the model's internal interaction patterns. To verify that our *Diff-Aid* learns to adaptively adjust block-wise interactions between token-level textual conditions across denoising trajectories, we analyze the learned coefficients $\alpha_t^l \in \mathbb{R}^N$ cached in each *Aid* block during denoising. Fig. 4 summarizes the learned α and attention norms across the same 2,400-prompt evaluation subset. The prompt-specific token patterns are visualized in Appendix F.1.

α adaptively exhibits sparse and independent patterns across different transformer blocks. The mean and standard deviation of $alpha$ over dimensions $\{T, N\}$ for each block are shown in Fig. 4(a). We could observe that the former blocks ($1st - 3rd$ in FLUX and $2nd - 4th$ in SD 3.5) and the latter blocks ($41st - 52nd$ in FLUX and $30th - 35th$ in SD 3.5) exhibit larger α values than the middle blocks. This aligns well with the observations [13, 5, 23] that early blocks focus on coarse structure and late blocks refine fine details. Additionally, there are noticeable zero regions in several blocks, showing the effectiveness of our sparsity strategies. The signed α learned by *Diff-Aid* captures varying block contributions, including suppression in selected blocks. In a separate lightweight ($d = 512$) control, clamping negative coefficients at inference reduces HPSv3 from 10.82 to 10.72 and GenEval from 0.709 to 0.692 (Appendix D).

α assigns token-specific importance weights to different textual tokens. The α distribution across token positions is shown in Fig. 4(b). The vertical line indicates the boundary between two concatenated text encoders in SD 3.5. In total, α shows a declining trend as token position increases for both models. This is expected since earlier tokens typically carry more semantic weight in prompts, while later tokens often include less informative padding or filler words. Notably, the abrupt change around token index 77 in SD 3.5 coincides with the two text encoders' boundary. All these observations confirm that our *Diff-Aid* accurately captures token-level importance variations.

Table 1: **Quantitative comparisons of strong baselines and our proposed *Diff-Aid* on the HPSv3 benchmark.** *Diff-Aid* significantly improves the baseline models across categories and even outperforms the current leading Kolors in most categories when applied to FLUX. † denotes results evaluated by ourselves and *scores are quoted from the official HPSv3 leaderboard [31]. **Bold** and underline indicate the best and second-best results, respectively.

Models	All	Char.	Arts	Design	Archi.	Animals	Scenery	Transpor.	Products	Plants	Food	Science	Others
Kolors [47]*	<u>10.55</u>	<u>11.79</u>	<u>10.47</u>	<u>9.87</u>	10.82	<u>10.60</u>	9.89	10.68	<u>10.93</u>	<u>10.50</u>	10.63	<u>11.06</u>	9.51
FLUX [22]*	10.43	11.70	10.32	9.39	10.93	10.38	10.01	10.84	**11.24**	10.21	10.38	**11.24**	9.16
Infinity [12]*	10.26	11.17	9.95	9.43	10.36	9.27	<u>10.11</u>	10.36	10.59	10.08	10.30	10.59	9.62
SD 3 [9]*	5.31	6.70	5.98	5.15	5.25	4.09	5.24	4.25	5.71	5.84	6.01	5.71	4.58
SD 3.5†	9.31	9.66	8.63	8.58	10.03	8.81	8.90	9.58	9.67	9.78	10.05	8.60	9.46
SD 3.5 + TACA†	9.44	9.81	8.82	8.54	10.31	8.92	9.26	9.54	9.96	9.85	10.05	8.81	9.48
SD 3.5 + Ours	9.48	9.93	8.87	8.69	10.22	9.00	9.17	9.80	9.76	9.87	10.18	8.74	9.55
FLUX†	10.42	11.67	10.31	9.35	10.94	10.37	9.97	10.87	10.48	10.39	**11.20**	9.24	10.24
FLUX + TACA†	10.48	11.72	10.35	9.36	<u>10.97</u>	10.45	10.03	<u>11.04</u>	10.53	10.46	**11.20**	9.33	<u>10.35</u>
FLUX + Ours	**10.71**	**11.96**	**10.72**	**9.90**	**11.19**	**10.62**	**10.18**	**11.24**	10.61	**10.67**	<u>11.18</u>	9.68	**10.58**

α implicitly learns dynamic timestep awareness throughout denoising trajectories. To investigate the timestep-awareness of α, we visualize its distribution throughout the denoising process in Fig. 4(c). Each block's α values are shown in Fig. 9a and Fig. 9b of appendix. FLUX is split into two parts due to its single-stream and dual-stream architectures. The different types of blocks behave differently, while single-stream blocks show a clear drop in α values at $t = 0.607$, while the dual-stream blocks and SD 3.5 blocks exhibit smooth changes. Such trend indicates that *Diff-Aid* implicitly learns to adjust text-image interactions dynamically across timesteps.

α aligns the intrinsic attributes of pretrained models for efficient and effective adaptation. To explain the α changes across denoising trajectories, we visualize the attention norm of textual and image features in attention blocks across timesteps in Fig. 4(d). We could clearly observe that the dashed vertical lines in Fig. 4(d) perfectly align with the sharp changes in α in Fig. 4(c). Based on the observations, our learned α already captures the intrinsic attribute changes of pretrained models across timesteps, enabling efficient and effective adaptation for the text-image interactions.

3.2 Main Results

Qualitative results. To demonstrate the effectiveness of our proposed *Diff-Aid*, we present qualitative comparisons of images generated by different models given the same prompts and seeds in Fig. 1 and Fig. 5. FLUX with *Diff-Aid* accurately captures the "*Nailpolish*" and "*Male*", generates a correct quantity of "*donuts*" and "*sneaker with tank wheels*". SD 3.5 with *Diff-Aid* effectively synthesizes the "*christmas frog*" and "*rabbit wedding*" with semantic details and better visual quality. Although derived from the same noise, *Diff-Aid* adaptively denoises to better trajectories according to the prompts, demonstrating its effectiveness in enhancing prompt adherence and image quality.

Quantitative results on HPSv3 benchmark. We conduct quantitative comparisons on the HPSv3 benchmark [31]. The results are summarized in Tab. 1. Both FLUX and SD 3.5 benefit from our *Diff-Aid*, with improvements of 0.29 and 0.17 in overall HPSv3 score, compared to their respective baselines. Our method also competes favorably against TACA [29], and FLUX combined with *Diff-Aid* even outperforms current SoTA Kolors [47] in overall score and most categories. These significant improvements across all categories validate *Diff-Aid*'s capability towards diverse and complex prompts, leading to improved human preferences.

Computational overhead analysis. To evaluate the efficiency of our *Diff-Aid*, we measure the computational overhead on $1,000$ images under default settings. With hidden dimension $d = 2048$ of the *Aid*, *Diff-Aid* introduces only $179.77M$ additional parameters (1.51% over the FLUX), and increases inference time by only 0.08 seconds (from $5.75s$ to $5.83s$). *Diff-Aid* achieves substantial performance improvements with negligible computational cost. The separate $d = 512$ variant adds 0.38% parameters. Details are provided in Appendix E.3.

Table 2: **Blinded pairwise human preferences.** Each comparison uses 200 prompts, two generated image pairs per prompt, and eight volunteer judgments per pair (3,200 votes).

Pair	Alternative wins	Tie	*Diff-Aid* wins
FLUX vs. FLUX + *Diff-Aid*	507 (15.84%)	1,269 (39.66%)	1,424 (44.50%)
SD 3.5 vs. SD 3.5 + *Diff-Aid*	724 (22.63%)	1,138 (35.56%)	1,338 (41.81%)
FLUX+LoRA vs. FLUX + *Diff-Aid*	848 (26.50%)	1,040 (32.50%)	1,312 (41.00%)

Generalization to other metrics and benchmark. Although *Diff-Aid* is optimized solely on the HPDv3 dataset, it brings consistent improvements on widely-used metrics such as HPSv2, ImageReward, and Aesthetic Score. The results are summarized in Tab. 3. HPSv2 scores improve by 0.15 and 0.20 for SD 3.5 and FLUX when combined with *Diff-Aid*, respectively. Moreover, *Diff-Aid* also improves the GenEval performance, achieving 5% and 2% improvements for SD 3.5 and FLUX, respectively. These results validate the effectiveness and generalizability of our approach across different evaluation metrics and benchmarks. More results are in Appendices E.1, E.2 and C.3.

Human evaluation. We conduct blinded pairwise comparisons against FLUX, SD 3.5, and FLUX+LoRA. For each comparison, we select 200 prompts and generate two image pairs per prompt, matching the prompt and random seed between the methods in each pair. Eight volunteers judge each pair by choosing which image is better or indicating that they are equally good. As is shown in Tab. 2, *Diff-Aid* receives more votes than the alternative in all three comparisons: 44.50% versus 15.84% against FLUX, 41.81% versus 22.63% against SD 3.5, and 41.00% versus 26.50% against FLUX+LoRA. Our method achieved a consistent advantage in human evaluation experiments.

3.3 Ablation Studies

To identify the efficacy of each component and optimization choice in our *Diff-Aid*, here we perform ablation studies by disabling each component for evaluation. Specifically, we train various variants on the FLUX baseline following the same settings in Sec. 3 except specific component is removed. We calculate quantitative scores on a random 20% subset of HPDv3 benchmark for a more efficient investigation. The quantitative results are presented in Tab. 5.

Ablation on boundedness and sparsity constraints. Recall that we introduce a regularization term and a gated mechanism to respectively stabilize and sparsify the values of α in Sec. 2. The results in Tab. 5 (*"Bounded Constraints"* and *"Sparsity Constraints"* groups) show that the performance degrades when these constraints are removed. Compared to simply clipping the α values, using Tanh activation yields better results(10.48 vs. 7.12 on HPSv3). Adding a smaller regularization weight ($\lambda_{reg} = 0.1$) further improves the performance than a larger one ($\lambda_{reg} = 1.0$). Armed with the gated mechanism, the model achieves the best results (28.71 on HPSv2, 10.74 on HPSv3) compared to the preceding settings. Ablation studies validate the effectiveness of our proposed boundedness and sparsity constraints in stabilizing and enhancing the learning of α.

Ablation on optimization choices and *Aid* drop probability. We investigate the impact of different optimization strategies and dropout probabilities p during training. As shown in Tab. 5 (*"Optimization Choices"* group), LoRA-based methods yield only marginal improvements over the FLUX baseline. In contrast, all variants of our *Aid* modules substantially outperform LoRA-based approaches across all metrics. Notably, our *Aid* modules can be effectively integrated with reward model or DPO to achieve competitive results. For optimization simplicity and to highlight the α learning mechanism, we adopt the straightforward SFT as our default optimization strategy. Regarding the dropout probability p in Tab. 5 (*"Drop p"* group), we observe that a small $p = 0.1$ consistently achieves the best performance across all metrics, outperforming both $p = 0.3$ and $p = 0.5$. These results validate that our *Aid* design with adaptive α learning is inherently more effective than LoRA-based methods.

Ablation of adaptive dimensions. To quantify the individual contribution of each dimension in our design, we remove the adaptivity of one dimension while keeping the other two fully adaptive. Tab. 5 (*"Dimension Adaptivity"* group) isolates the block, text-token, and timestep dimensions on the same 2,400-prompt HPDv3 subset. Sharing the module across blocks or a coefficient across text tokens lowers HPSv3 from 10.84 to 9.58 or 10.28, respectively. Time-averaging the coefficients lowers it to 10.46, while fixing them to one denoising step performs worse. These results show the necessity of adaptive modulation in our *Diff-Aid*. Appendix E.7 specifies the details.

Table 3: **Quantitative comparisons of more image quality metrics.** Despite trained only on HPDv3 dataset, our *Diff-Aid* generalizes to other metrics and gains consistent improvements on these metrics. Note that GenEval scores are computed from official GenEval benchmarks.

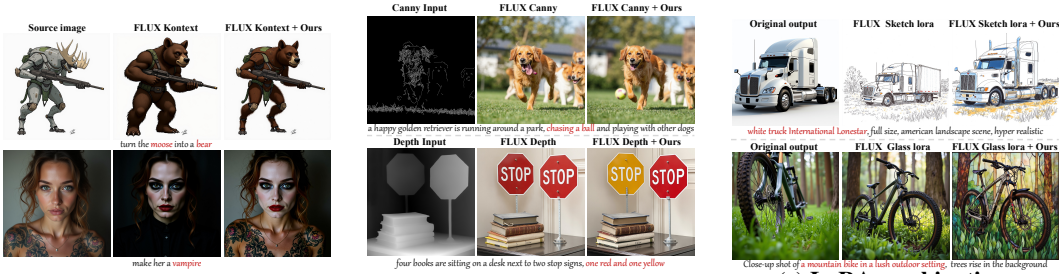
Models	HPSv2	HPSv3	ImgRwd	Aes.	GenEval
SD 3.5	28.86	9.31	0.94	6.42	0.72
SD 3.5 + TACA	28.89	9.44	0.93	6.44	0.73
SD 3.5 + Ours	**29.01**	**9.48**	**1.05**	**6.53**	**0.77**
FLUX	28.53	10.42	0.89	6.65	0.68
FLUX + TACA	28.59	10.48	0.90	6.68	0.68
FLUX + Ours	28.73	**10.71**	0.94	**6.73**	**0.70**

Table 4: **Quantitative comparison of instructional image editing tasks on EditBench benchmark.** Despite trained on the pure generation settings, our *Diff-Aid* generalizes to homologous Kontext by enhancing test-time interactions between editing prompts and images in a zero-shot manner.

Models	HPSv2	HPSv3	ImgRwd	Aes.	CLIP-I	CLIP-T
Kontext	27.48	8.50	**0.62**	6.51	90.74	26.35
Kontext + Ours	**27.54**	**8.62**	**0.62**	**6.53**	**91.67**	**26.41**

Table 5: **Ablation study on key components of our proposed method**. "+" means adding the new component on the previous setting. The ***italicized*** settings are our default choices.

Categories	Settings	HPSv2	HPSv3	ImgRwd	Aes.
FLUX	Baseline	28.53	10.42	0.89	6.66
Bounded Constraints	No Const.	20.03	6.41	-2.04	3.54
	+ Clip	22.86	7.12	0.31	4.07
	+ Tanh Act	28.55	10.48	0.90	6.67
Sparsity Constraints	+ Reg(1.0)	28.49	10.44	0.90	6.64
	+ Reg(0.1)	28.61	10.59	0.92	6.68
	+ Gated	28.71	10.74	0.94	6.72
Drop p	+ drop(0.5)	28.21	10.19	0.77	6.59
	+ drop(0.3)	28.47	10.44	0.87	6.64
	+ drop(0.1)	28.80	10.84	0.95	6.76
Optimization Choices	LoRA +SFT	28.54	10.43	0.86	6.63
	LoRA +DPO	28.57	10.48	0.90	6.67
	Ours +Rwd	28.76	10.78	0.93	6.73
	Ours +DPO	28.81	10.85	0.94	6.75
	Ours +SFT	28.80	10.84	0.95	6.76
Dimension Adaptivity	*w/o* blocks	27.41	9.58	0.77	6.34
	w/o tokens	28.07	10.28	0.88	6.48
	w/o timesteps	28.53	10.32	0.82	6.35
FLUX + Ours	***full***	28.80	10.84	0.95	6.76



(a) **Instructional image editing.** (b) **Controllable generation.** (c) **LoRA combination.**

Figure 6: **Qualitative results of *Diff-Aid* on downstream applications.** *Diff-Aid* demonstrates strong generalization across diverse tasks without task-specific training, improving instruction-based editing, precise controllable generation, and high-quality stylized synthesis.

3.4 Generalization to Downstream Applications

Generalizing to zero-shot instruction-based editing. Our *Aid* blocks can be directly integrated into FLUX.1 Kontext [4] without any additional training. We evaluate the edited images on the EditBench benchmark [25], as summarized in Tab. 4. All the metrics show consistent improvements when *Diff-Aid* is applied to Kontext. Notably, CLIP-I and CLIP-T scores increase by 0.93 and 0.06, respectively, indicating better alignment with editing instructions. Visualizations in Fig.1 and Fig. 6a further keep the original details while accurately following editing instructions.

Integrating with conditional control models. Cooperating with ControlNet [62], our *Diff-Aid* can further enhance controllable generation by improving the interactions between complex textual conditions. In Fig. 1 and Fig. 6b, our method effectively captures the semantic details, such as the "*Wolf*" and "*chasing a ball*", which are missed by the baseline. Moreover, "*one yellow*" stop sign is accurately generated with *Diff-Aid*, showing its effectiveness in handling complex textual conditions.

Combining with existing LoRAs. When combined with off-the-shelf LoRAs from the community, our *Diff-Aid* boosts the generation quality while preserving the desired styles. In Fig. 1 and Fig. 6c, the images generated by combining *Diff-Aid* exhibit richer details and more vivid textures. The quantitative results are shown in Tab. 16 in Appendix E.4.

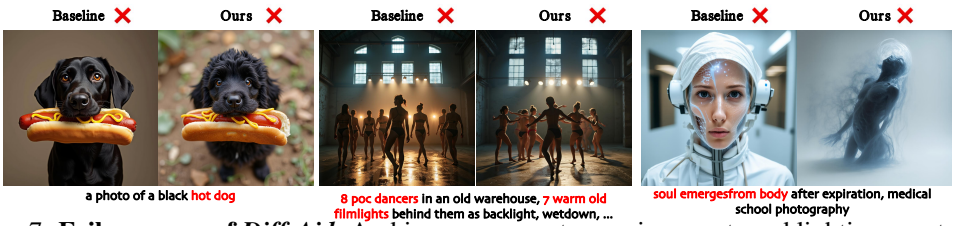


Figure 7: **Failure cases of *Diff-Aid*.** Ambiguous concepts, precise counts and lighting constraints, and abstract descriptions remain challenging for both the baseline and *Diff-Aid*.

4 Related Works

Diffusion transformers. DiT [35] pioneers the use of transformers [48] in diffusion models for image and video generation. Following DiT, Multimodal Diffusion Transformers (MMDiTs) [9, 22, 55] facilitate text-image interactions through a unified transformer. Subsequent works [46, 11, 34, 50, 52, 61, 24] have promoted AI generated content (AIGC).

Improving cross-modal interactions in diffusion models. Early works [40, 47] inject text features into diffusion models via cross-attention mechanisms. DiT [35, 6] uses adaLN-zero to condition transformer activations, while FiLM [36] applies feature-wise affine conditioning. MMDiTs [9, 22, 55] suggest that text and image features should be treated equally by jointly modeling them in a unified transformer architecture. Other paradigms include in-context conditioning [20] and auxiliary controlnets [62, 33]. Related studies explore instance-aware prompt adaptation [58], layer-specific feature fusion [57], and causal attention interventions [51] to enhance feature adaptation and multimodal interactions. Our intervention instead rescales individual text-token states immediately before MMDiT joint attention, with coefficients that vary by block and denoising step.

Understanding cross-modal interactions in diffusion models. Some works [13, 5, 26] investigate the role of attention mechanisms in Unet-based diffusion models for instruction-based editing. They reveal that attention values largely reflect the intensity of text-image interactions. Classifier-free guidance (CFG) [15] adopts a static scaling factor to balance conditional and unconditional model outputs, thereby controlling textual strength. Despite these advances, a holistic understanding of DiT remains underexplored. Prior work [3, 42, 54] focused on adjusting attention mechanisms, evaluating rotary position embeddings, or analyzing specific blocks. TACA [29] identified cross-modal attention imbalance and introduced a time-aware weighting scheme. Li et al. [23] dissected the block contributions and proposed block-wise conditioning strategies. TACA prescribes attention enhancement over time, while our coefficients are learned from data and can either enhance or suppress individual textual signals. CFG and its scheduling variants [15, 53] adjust the overall sampling guidance, while our method acts inside the denoiser and can be combined with CFG.

5 Discussion

Limitation and future work. Since *Diff-Aid* modulates textual interactions with a frozen backbone, it cannot reliably resolve ambiguous concepts, supply missing visual grounding, or enforce multiple simultaneous constraints shown in Fig. 7. Improved average alignment therefore does not guarantee prompt fidelity. Future work will explore joint training with the backbone to improve visual grounding, explicit above constraints, and extensions to other attention architectures and modalities.

Conclusions. We presented *Diff-Aid*, a plug-in module for rectified text-to-image diffusion transformers that learns to adaptively adjust block-wise interactions between token-level textual conditions across denoising trajectories. Beyond improving the generation quality, *Diff-Aid* provides insights into the dynamic interplay of cross-modal signals in transformer blocks and can be seamlessly applied to various tasks. Our work benefits creative industries and accessibility tools, while it may also be misused to create realistic fake images, increasing risks of misinformation and deepfakes.

Acknowledgements

This work was supported in part by AI for Science Program, Shanghai Municipal Commission of Economy and Informatization (Grant No. 2025-GZL-RGZNBTBX-02017) and the National

Natural Science Foundation of China (Grant No.62576103). The computations in this research were performed using the CFFF platform of Fudan University.

References

[1] Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Jungwoo Kim, Wooseok Jang, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *Eur. Conf. Comput. Vis.*, 2024.

[2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH*, pages 1–12, 2024.

[3] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7877–7888, 2025.

[4] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Int. Conf. Comput. Vis.*, pages 22560–22570, 2023.

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *Int. Conf. Learn. Represent.*, 2024.

[7] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *Int. Conf. Learn. Represent.*, 2024.

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Adv. Neural Inform. Process. Syst.*, volume 34, pages 8780–8794, 2021.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Int. Conf. Mach. Learn.*, 2024.

[10] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Adv. Neural Inform. Process. Syst.*, volume 36, pages 52132–52152, 2023.

[11] Lixue Gong, Xiaoxia Hou, Fanshi Li, Liang Li, Xiaochen Lian, Fei Liu, Liyang Liu, Wei Liu, Wei Lu, Yichun Shi, et al. Seedream 2.0: A native chinese-english bilingual image generation foundation model. *arXiv preprint arXiv:2503.07703*, 2025.

[12] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15733–15744, 2025.

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, pages 6840–6851, 2020.

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[16] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. In *Int. Conf. Comput. Vis.*, pages 7462–7471, 2023.

[17] 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. *Int. Conf. Learn. Represent.*, 1(2):3, 2022.

[18] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[19] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Adv. Neural Inform. Process. Syst.*, volume 36, pages 78723–78747, 2023.

[20] Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. In-context lora for diffusion transformers. *arXiv preprint arXiv:2410.23775*, 2024.

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Adv. Neural Inform. Process. Syst.*, 35:26565–26577, 2022.

[22] Black Forest Labs. Flux. github.com/black-forest-labs/flux, 2024.

[23] Binglei Li, Mengping Yang, Zhiyu Tan, Junping Zhang, and Hao Li. Textailor: Inference-time textual guidance tailoring for multimodal diffusion transformers. In *Eur. Conf. Comput. Vis.*, 2026.

[24] Yuzhuo Li, Di Zhao, Xinyu Zhang, Daniel Wilson, and Yun Sing Koh. Generating the wild: Individual-consistent image-to-video generation for wildlife. *arXiv preprint arXiv:2610.05587*, 2026.

[25] Haonan Lin, Yan Chen, Jiahao Wang, Wenbin An, Mengmeng Wang, Feng Tian, Yong Liu, Guang Dai, Jingdong Wang, and Qianying Wang. Schedule your edit: A simple yet effective diffusion noise schedule for image editing. *Adv. Neural Inform. Process. Syst.*, 37:115712–115756, 2024.

[26] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7817–7826, 2024.

[27] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Int. Conf. Mach. Learn.*, 2024.

[28] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Int. Conf. Learn. Represent.*, 2023.

[29] Zhengyao Lv, Tianlin Pan, Chenyang Si, Zhaoxi Chen, Wangmeng Zuo, Ziwei Liu, and Kwan-Yee K Wong. Rethinking cross-modal interaction in multimodal diffusion transformers. In *Int. Conf. Comput. Vis.*, 2025.

[30] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[31] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Int. Conf. Comput. Vis.*, 2025.

[32] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *Int. Conf. Mach. Learn.*, 2024.

[33] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Assoc. Adv. Artif. Intell.*, volume 38, pages 4296–4304, 2024.

[34] OpenAI. Video generation models as world simulators. 2024. Technical Report, OpenAI.

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4195–4205, 2023.

[36] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Assoc. Adv. Artif. Intell.*, pages 3942–3951, 2018.

[37] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739v2*, 2023.

[38] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *Adv. Neural Inform. Process. Syst.*, 2025.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pages 8748–8763, 2021.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10684–10695, 2022.

[41] Chrisoph Schuhmann. Laion aesthetics. github.com/LAION-AI/aesthetic-predictor, 2022.

[42] Joonghyuk Shin, Alchan Hwang, Yujin Kim, Daneul Kim, and Jaesik Park. Exploring multi-modal diffusion transformers for enhanced prompt-based image editing. In *Int. Conf. Comput. Vis.*, 2025.

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Int. Conf. Learn. Represent.*, 2021.

[44] Stability-AI. Stable diffusion 3.5. github.com/Stability-AI/sd3.5, 2024.

[45] Hunyuan Foundation Model Team. Hunyuanvideo: A systematic framework for large video generative models, 2024.

[46] Hunyuan Foundation Model Team. Hunyuanimage-2.1: An efficient diffusion model for high-resolution (2k) text-to-image generation, 2025.

[47] Kolors Team. Kolors: Effective training of diffusion model for photorealistic text-to-image synthesis. github.com/Kwai-Kolors/Kolors, 2024.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

[49] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8228–8238, 2024.

[50] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[51] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2609.38810*, 2026.

[52] Jun Wang, Chunyu Qiang, Yuxin Guo, Yiran Wang, Xijuan Zeng, Chen Zhang, and Pengfei Wan. Klear: Unified multi-task audio-video joint generation. *arXiv preprint arXiv:2601.04151*, 2026.

[53] Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernandez Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *Transactions on Machine Learning Research*, 2024.

[54] Tianyi Wei, Yifan Zhou, Dongdong Chen, and Xingang Pan. Freeflux: Understanding and exploiting layer-specific roles in rope-based mmdit for versatile image editing. In *Int. Conf. Comput. Vis.*, 2025.

[55] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[56] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[57] Xi Xiao, Xingjian Li, Yunbei Zhang, Cheng Han, Tianming Liu, Tianyang Wang, Runmin Jiang, Jihun Hamm, Xiao Wang, and Min Xu. Layer-specific prompt fusion discovery via differentiable search in vision foundation models. In *Eur. Conf. Comput. Vis.*, pages 391–409. Springer, 2026.

[58] Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. Visual instance-aware prompt tuning. In *ACM Int. Conf. Multimedia*, pages 2880–2889, 2025.

[59] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Adv. Neural Inform. Process. Syst.*, pages 15903–15935, 2023.

[60] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *Int. Conf. Learn. Represent.*, 2025.

[61] Guozhen Zhang, Zixiang Zhou, Teng Hu, Ziqiao Peng, Youliang Zhang, Yi Chen, Yuan Zhou, Qinglin Lu, and Limin Wang. Uniavgen: Unified audio and video generation with asymmetric cross-modal interactions. *arXiv preprint arXiv:2511.03334*, 2025.

[62] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Int. Conf. Comput. Vis.*, pages 3836–3847, 2023.

A Outline of Appendix

We provide additional details and results to supplement the main paper as follows. We first present experimental setup and human-evaluation details in Sec. B. Sec. C studies sampling trajectories and CFG robustness. Sec. D compares static, enhancement-only, and full signed modulation. Sec. E provides additional baseline, alignment, efficiency, application, and ablation results. Finally, Sec. F provides token-level visualizations, further examples, and failure cases.

B Details of Experimental Setup

B.1 Details of Baselines

We conduct experiments on two strong text-to-image diffusion models: Stable Diffusion 3.5 Large (SD 3.5) [44] and FLUX.1-Dev [22]. We use the official checkpoints provided by the authors and do not perform any fine-tuning on these models. During inference, model weights are loaded in 16-bit precision. No acceleration techniques such as xformers or memory-efficient attention are used. The Tab. 6 summarizes the key hyperparameters used during inference for both models.

Table 6: Default inference hyper-parameters for both SD 3.5 and FLUX.

Model	DiT Blocks	Parameters	Inference Timesteps	Sampler	CFG Scale	Image Size
SD 3.5	38	8B	28	EDM	7.0	1024×1024
FLUX	57	12B	28	EDM	3.5	1024×1024

B.2 Details of Evaluation Metrics

To comprehensively evaluate the performance of our proposed method, we utilize a suite of established evaluation metrics that assess various aspects of text-to-image generation quality, including prompt adherence, image quality, and semantic alignment. We employ the following evaluation metrics to assess the performance of our proposed method.

- **HPSv2** [56] can effectively evaluate the preference score of text-image pairs or multiple images given a text prompt. It is trained on a large-scale human preference dataset and has shown strong correlation with human judgments.
- **HPSv3** [31] is derived from the QWen2-VL and trained on the HPDv3 dataset. It demonstrates improved performance over HPSv2 in evaluating text-image alignment and image quality.
- **ImageReward** [59] utilizes preference data collected from human feedback and applies reinforcement learning to train a reward model that can evaluate the image preference score and provide feedback in reward signals.
- **Aesthetic Score** [41] primarily focuses on assessing the overall aesthetic quality of an image, which is built upon the CLIP models. This score is often subjective regarding composition, color harmony, and visual appearance.
- **Geneval Score** [10] proposes an object-focused evaluation metric for text-to-image generation models. It defines a set of tasks to evaluate different aspects of text-image alignment, including "Single Object", "Two Object", "Counting", "Position", and "Attribute binding". The Geneval Score is computed as the average accuracy across these tasks. It primarily focused on evaluating the model's ability for semantic alignment.
- **CLIP Score** [39] measures the cosine similarity between the image and text embeddings obtained from a pre-trained CLIP model. CLIP-I and CLIP-T are used to evaluate the editing tasks.
- **T2I-CompBench** [19] measures compositional prompt following, including attributes and spatial relations.
- **DPG-Bench** [18] evaluates dense-prompt following across objects, attributes, and relationships.

B.3 Details of Datasets

Training Datasets. HPDv3 dataset [31] totally contains 1.17 million binary preference choices over 1.08 million images, grouped in pairs by prompts. These images are collected from various sources, including real images, MidJourney, Cogview4, FLUX, HunyuanDiT, Stable Diffusion 3 Medium, Stable Diffusion 1.4, and Glide, *etc.* Most sources are not competitive text-to-image models with FLUX and SD 3.5, which use all the images from suboptimal models, which may lead to inferior performance.

To ensure the training data quality, we filter the HPDv3 dataset using the following criteria: ***1)*** Preference confidence filter. We only keep the image pairs with preference confidence scores higher than 0.9. ***2)*** Image quality filter. We only keep the image pairs where the preferred images are from real images, FLUX, or Kolors. After filtering, we obtain $77,252$ image pairs for training our *Aid* on both baselines.

Subset of HPDv3 Benchmark. HPDv3 benchmark consists of $12,000$ prompts covering a wide range of categories, including "Animals", "Architecture", "Arts", "Characters","Design", "Food", "Nature Scenery", "Plants", "Products", "Science", "Transportation", and "Others". Each category contains $1,000$ prompts. Tab. 1, 3 in the main paper report the results on the whole HPDv3 benchmark. Other than the full benchmark, we also report the results on the 20% subset of HPDv3 in Tab. 5 and 18, which contains $2,400$ prompts sampled from all categories. The benchmark evaluation prompts are strictly excluded from the training subset.

B.4 Effect of Training Data Quality

To isolate the role of filtering, we also train an Aid module using a mixture that includes lower-quality SD 1.5-generated images. Tab. 7 uses the same 2,400-prompt evaluation subset as the optimization ablations; the filtered-data SFT and DPO rows follow Tab. 5. Filtering improves the SFT HPSv3 result from 10.74 to 10.84, while the less-filtered Aid module still exceeds LoRA trained with filtered data. The comparison indicates that both data selection and the modulation design matter.

Table 7: Training-data quality ablation on the 2,400-prompt HPDv3 subset. "Filtered" uses confidence > 0.9 and preferred images from real images, FLUX, or Kolors; "mixed" additionally includes lower-quality SD 1.5 images.

Method	Training data	HPSv2	HPSv3	ImgRwd	Aes.
FLUX	–	28.53	10.42	0.89	6.66
LoRA + SFT	Filtered	28.54	10.43	0.86	6.63
Diff-Aid + SFT	Mixed	28.74	10.74	0.93	6.73
Diff-Aid + SFT	Filtered	28.80	10.84	0.95	6.76
Diff-Aid + DPO	Filtered	28.81	10.85	0.94	6.75

B.5 Details of optimization choices

In Sec. 2.2 of the main paper, we introduce several optimization choices to enhance the training stability and performance of our *Aid*. In ablation studies (Tab. 5), we systematically evaluate the impact of four different optimization choices. Here, we provide more implementation details for each choice.

1) LoRA + SFT. We apply LoRA [17] with a rank of 64 to all the linear layers in the FLUX. No *Aid* is applied to the transformer blocks in this setting.

2) LoRA + DPO. Based on the LoRA setting above, we further apply DPO to optimize the parameters to obtain better performance. We set the DPO temperature β to 0.1 and use the original FLUX as the reference model. The λ_{dpo} is set to 1.0, which is commonly used in previous works [49].

3) Ours + Rwd. In this setting, we use all the designs in our *Aid*, including Tanh activation, α regularization ($\lambda_{reg} = 0.1$), gated mechanism, and skipping during training ($p = 0.1$). The *Aid* has an "mlp_hidden_dim" of 2048. We further incorporate the HPSv3 as the reward model to provide additional reward signals during training. The reward weight λ_{rwd} is set to 1.0. To keep the reward scores in a reasonable range, we only apply reward learning when the timestep t is smaller than 0.3.



Figure 8: **Static sparse enhancement compared with full adaptive modulation and baseline FLUX.** The fixed block rule improves some prompts but cannot reproduce token- and timestep-dependent behavior. HPSv3, ImageReward, and Aesthetic scores are shown below each image.

4) Ours + SFT. In this setting, we use all the designs in our *Aid*, including Tanh activation, α regularization ($\lambda_{reg} = 0.1$), gated mechanism, and skipping during training ($p = 0.1$). The *Aid* has an "mlp_hidden_dim" of 2048. We apply the straightforward supervised fine-tuning (SFT) to optimize the parameters without additional reward signals or DPO.

5) Ours + DPO. Similar to the settings of "Ours + SFT", we use all the designs in our *Aid*. We further apply DPO to optimize the parameters. The DPO settings are the same as those in "LoRA + DPO".

In Tab. 5, we can observe that the giant leap in performance mainly comes from our *Aid* designs and sparsity strategies, rather than the choice of optimization strategies. Specifically, comparing "LoRA + SFT" with "Ours + SFT", our *Aid* brings substantial improvements of $+0.41$ on HPSv3 and $+0.27$ on HPSv2, demonstrating the effectiveness of our adaptive modulation mechanism. Incorporating additional optimization strategies like DPO or reward-based learning can achieve competitive performance and may provide marginal improvements on certain metrics. However, the core contribution comes from our *Aid* block designs, including the three-dimensional adaptive modulation, boundedness and sparsity constraints. For optimization simplicity and to highlight the α learning mechanism, we adopt the straightforward SFT as our default optimization strategy. Therefore, our default settings are using all the designs in our *Aid* block with SFT optimization.

C Generalization to Different Sampling Trajectories

To validate that our learned module generalizes across different sampling trajectories and is not overfit to the specific 28-step EDM sampler used during training, we conduct two comprehensive experiments.

C.1 Results on FLUX.1-Schnell

We evaluate our *Aid* modules on FLUX.1-Schnell, a distilled variant of FLUX.1 that uses only 4 sampling steps. Despite the significant reduction in steps (from 28 to 4) and the different denoising scheme, our method still achieves substantial improvements over the baseline model. As shown in Tab. 8, our *Aid* modules bring consistent gains across all metrics. This demonstrates that the learned α values are not overfit to the specific 28-step trajectory and can generalize to drastically different sampling schemes.

C.2 Comprehensive Evaluation Across Different Timesteps

We conduct a systematic evaluation on FLUX.1-Dev across 6 different sampling steps: $\{4, 8, 10, 15, 28, 50\}$ using $1,000$ prompts with fixed random seeds. The results are summarized

Table 8: **Quantitative results on FLUX.1-Schnell with 4 sampling steps.** Our method achieves consistent improvements even with a distilled model using only 4 steps, demonstrating strong generalization across different sampling trajectories.

Models	HPSv2	HPSv3	ImgRwd	Aes.
FLUX.1-Schnell	28.48	9.95	0.87	6.55
+Ours	**28.65**	**10.14**	**0.91**	**6.64**

in Tab. 9. Our method consistently outperforms the baseline across all timestep configurations. Notably, at 4 steps, our method achieves $+1.14$ HPSv2, $+3.47$ HPSv3, and $+0.56$ ImageReward improvements, demonstrating robust generalization despite the dramatic trajectory shift. At 50 steps, the improvements remain consistent. These results confirm that our *Aid* modules do not overfit to the specific denoising trajectory and maintains effectiveness across a wide range of sampling configurations.

Table 9: **Comprehensive evaluation across different sampling steps on FLUX.1-Dev.** Our method consistently outperforms the baseline across all timestep configurations, demonstrating robust generalization to different sampling trajectories. All experiments are conducted on $1,000$ prompts with fixed random seeds.

Metric	Method	4	8	10	15	28	50
HPSv2	Baseline	25.48	27.53	27.82	28.44	28.95	29.11
	Ours	**26.62**	**28.29**	**28.55**	**28.95**	**29.15**	**29.23**
HPSv3	Baseline	5.07	9.70	9.94	10.45	10.89	11.07
	Ours	**8.54**	**11.31**	**11.29**	**11.34**	**11.33**	**11.41**
ImgRwd	Baseline	0.21	0.87	0.90	1.01	1.08	1.10
	Ours	**0.77**	**1.10**	**1.12**	**1.14**	**1.14**	**1.13**
Aes.	Baseline	5.95	6.67	6.70	6.84	6.88	6.88
	Ours	**6.62**	**7.03**	**7.02**	**7.02**	**6.96**	**6.93**

These experiments provide strong evidence that our training does not modify the noise schedule or denoising scheme, and the learned α values are theoretically applicable to all timesteps. This is further confirmed by Figure 10 in Sec. E.6, where we visualize the α mean values across different sampling timesteps. The curves converge to a stable pattern, demonstrating that *Aid* does not overfit to the specific 28-step trajectory.

C.3 Interaction with CFG Scale

CFG rescales the sampling prediction globally, whereas *Diff-Aid* changes text-token states before joint attention. We evaluate both methods with the same prompts and seeds at seven FLUX guidance scales. Tab. 10 shows a GenEval improvement at every tested scale; the mean gain is 0.038. The result supports complementarity within the tested range, rather than invariance to arbitrary guidance choices.

Table 10: GenEval across FLUX CFG scales with fixed prompts and seeds.

CFG scale	1	2	4	6	8	10	15
FLUX	0.618	0.653	0.662	0.667	0.664	0.638	0.586
FLUX + *Diff-Aid*	**0.652**	**0.674**	**0.690**	**0.703**	**0.705**	**0.682**	**0.647**

D Sparse Enhancement Results

Li et al. [23] selected a subset of transformer blocks in MMDiT to enhance the text embeddings. This approach depends on the prior knowledge of which blocks are vital for text-image generation. From Fig. 4(a) in the main paper, we can observe that some blocks have higher α values, indicating their

importance in text-image interactions. We simply select the blocks $\{1, 15, 36, 41, 48\}$ for FLUX and set the α to 0.5 for these blocks and 0.0 for others. This sparse enhancement strategy overlooks the learned timestep-awareness and token-level modulation in our *Aid*. However, it provides insights into the effectiveness of enhancing only a few crucial blocks. As shown in Fig. 8 and Tab. 11, static enhancement can recover some important content but falls below full adaptive modulation on HPSv3. ImageReward score (0.98) of sparse enhancement even outperforms our full method. This shows that the selected blocks are useful, while their fixed amplification does not account for token, timestep, or sign differences. The comparison supports the functional relevance of the learned block pattern without attributing the full method's gains to block selection alone.

Table 11: **Quantitative results of sparse enhancement strategy compared to the full method and baseline FLUX.** Sparse enhancement improves on the baseline but remains below full modulation on HPSv3. This indicates the effectiveness of our learned α values in identifying important blocks for text-image interactions.

Strategies	HPSv2	HPSv3	ImgRwd	Aes.
FLUX	28.53	10.42	0.89	6.66
Sparse Enhancement	<u>28.61</u>	<u>10.57</u>	**0.98**	<u>6.71</u>
Our Full Method	**28.80**	**10.84**	<u>0.95</u>	**6.76**

Signed modulation versus enhancement alone. The fixed sparse rule above only increases selected block signals. In a separate lightweight ($d = 512$) comparison, we also clamp the learned coefficient to $\max(\alpha, 0)$ at inference. Tab. 12 shows that adaptive enhancement improves on fixed scaling, while restoring negative coefficients improves both metrics further. This supports a functional role for suppression beyond the visible negative regions in Fig. 4.

Table 12: Static, enhancement-only, and signed adaptive controls in the additional lightweight ($d = 512$) FLUX comparison. The default $d = 2048$ result is reported separately in Tab. 11.

Control	HPSv3	GenEval
FLUX baseline	10.42	0.676
Sparse static scaling	10.57	0.683
Enhancement-only, $\alpha \leftarrow \max(\alpha, 0)$	10.72	0.692
Full signed *Diff-Aid* ($d = 512$)	**10.82**	**0.709**

E More Quantitative Results

E.1 Compositional and Text-Image Alignment Benchmarks

The preference-oriented scores in the main paper are complemented by compositional and dense-prompt evaluations. For each backbone, baseline and *Diff-Aid* generations use matching prompts and seeds. Tab. 13 shows gains on both backbones for T2I-CompBench [19], DPG-Bench [18], and CLIP Score [39]. The small CLIP Score differences are reported without claiming that this metric alone establishes semantic correctness.

Table 13: Additional alignment results. Higher is better for all three metrics.

Model	T2I-CompBench	DPG-Bench	CLIP Score
FLUX	0.4739	83.67	28.13
FLUX + *Diff-Aid*	**0.5006**	**87.14**	**28.35**
SD 3.5	0.4980	76.40	27.88
SD 3.5 + *Diff-Aid*	**0.5249**	**79.32**	**27.95**

E.2 Additional Guidance, PEFT, and Reward-Guided Baselines

We compare further FLUX adaptations of sampling guidance [16, 1, 53], parameter-efficient tuning [17, 27], and reward-guided tuning [7, 37]. Tab. 14 uses the 2,400-prompt HPSv3 subset and

the lightweight $d = 512$ *Diff-Aid* variant ($+0.38\%$ parameters), rather than the default $d = 2048$ variant ($+1.51\%$) in the main paper. The reward-guided baselines approach its HPSv3 score, whereas the lightweight *Diff-Aid* gives the highest GenEval in this comparison. Tab. 1 separately reports the default model on the full 12,000-prompt benchmark.

Table 14: Additional FLUX comparison on the 2,400-prompt HPSv3 subset and GenEval, using the lightweight *Diff-Aid* variant. Guidance methods add no trainable parameters; parameter percentages are relative to FLUX.

Method	Type	Extra params	HPSv3	GenEval
FLUX	Baseline	–	10.42	0.676
SAG [16]	Guidance	0	10.45	0.678
PAG [1]	Guidance	0	10.51	0.678
Dynamic CFG [53]	Guidance	0	10.50	0.681
LoRA [17]	PEFT	$+0.31\%$	10.43	0.683
DoRA [27]	PEFT	$+0.33\%$	10.47	0.691
DRaFT [7]	Reward-guided	$+0.31\%$	10.81	0.664
AlignProp [37]	Reward-guided	$+0.31\%$	**10.82**	0.670
Diff-Aid ($d = 512$)	Adaptive modulation	$+0.38\%$	**10.82**	**0.709**

E.3 Parameter Efficiency and Computational Overhead Analysis

To comprehensively evaluate the efficiency of our proposed *Diff-Aid*, we conduct a detailed comparison with LoRA-based methods under similar parameter budgets. We measure the exact computational overhead on $1,000$ images of resolution 1024×1024 with 28 sampling steps on FLUX.1-Dev. The evaluation includes additional parameters, peak VRAM usage, average inference time per image, and performance metrics across HPSv2, HPSv3, ImageReward, and Aesthetic Score. We compare our method with two LoRA configurations: LoRA with rank 32 and rank 128, as well as two variants of our Aid module with hidden dimensions of 512 and 2048. The results are summarized in Tab. 15.

Table 15: **Comparison of parameter efficiency and computational overhead between LoRA and our method.** Runtime is measured on $1,000$ FLUX.1-Dev images at 1024×1024 resolution and 28 sampling steps; preference scores use the 2,400-prompt HPDv3 subset. Our method achieves substantially better performance than LoRA at comparable parameter budgets.

Method	Additional Param (M)	Peak VRAM (GB)	Avg. Time (s)	HPSv2	HPSv3	ImgRwd	Aes.
FLUX.1-Dev	—	37.22	5.7483	28.53	10.42	0.89	6.66
+LoRA (rank=32)	37.35 (+0.31%)	37.67	5.7649	28.54	10.43	0.86	6.63
+LoRA (rank=128)	149.42 (+1.26%)	37.96	5.8164	28.55	10.44	0.88	6.68
+Ours (dim=512)	45.21 (+0.38%)	37.83	5.7897	28.80	10.82	0.95	6.75
+Ours (dim=2048)	179.77 (+1.51%)	38.08	5.8252	28.80	10.84	0.95	6.76

Parameter Efficiency. At comparable parameter budgets, our Aid module demonstrates substantially superior performance over LoRA-based methods. Specifically, when comparing LoRA (rank=128) with 149.42M additional parameters (+1.26%) against our method (dim=2048) with 179.77M parameters (+1.51%), our approach achieves significantly larger improvements: +0.42 on HPSv3 and +0.27 on HPSv2, while LoRA only gains +0.02 on both metrics. Even our lightweight variant (dim=512) with only 45.21M parameters (+0.38%) substantially outperforms both LoRA configurations, achieving +0.40 on HPSv3 and +0.27 on HPSv2. This demonstrates that the performance gains stem from our Aid module's token-adaptive modulation design rather than simply increasing parameter count.

Computational Overhead. Our method introduces modest computational overhead that is comparable to LoRA at similar parameter scales. The default configuration (dim=2048) adds only 0.86 GB of peak VRAM and incurs a wall-clock overhead of just +1.34% (0.077s per image) compared to the baseline. This overhead is comparable to LoRA (rank=128), which adds 0.74 GB VRAM and +1.18% inference time, yet our method delivers substantially larger performance gains. The lightweight variant (dim=512) further reduces the overhead to +0.61 GB VRAM and +0.72% inference time with

minimal performance degradation, demonstrating the flexibility of our approach to balance between efficiency and performance.

Architectural Advantages over LoRA. The superior performance of our Aid module over LoRA at comparable parameter budgets highlights the importance of where and how modulation is applied. Unlike LoRA, which uniformly adjusts backbone weights across all layers, our Aid module provides dynamic, input-dependent modulation of text features before the joint self-attention in MMDiT, enabling fine-grained control over text-image interactions at the token level. This architectural design allows our method to adaptively modulate text conditioning strength across blocks, timesteps, and tokens, addressing the token imbalance problem that LoRA cannot capture. The consistent improvements across all metrics (HPSv2, HPSv3, ImageReward, and Aesthetic Score) further validate that our Aid module's structured modulation capability is more effective than LoRA's uniform weight adaptation for enhancing text-image alignment.

E.4 Quantitative Results on Applications

In Sec. 3.4 of the main paper, we demonstrate the versatility of our proposed *Diff-Aid* by applying it to different conditional generation tasks. We visualize some qualitative results in Fig. 6a, Fig. 6b and Fig. 6c. Here, we further evaluate our method on two applications: 1) using "Canny"[3] and "Depth"[4] conditions. 2) using "Sketch"[5] and "Glass"[6] LoRA modules to adapt to new styles. The quantitative results are shown in Tab. 16.

Table 16: **Quantitative results of applying our *Diff-Aid* to style LoRAs and control conditions generation.** Our method can be flexibly applied to different applications while obtaining competitive performance improvements.

Strategies	HPSv2	HPSv3	ImgRwd	Aes.
LoRA	28.50	10.38	0.87	6.64
LoRA + Ours	28.54	10.43	0.86	6.63
Control	28.57	10.48	0.90	6.67
Control + Ours	28.76	10.68	0.93	6.70

These results are obtained on the FLUX model on the 20% of the HPDv3 dataset. Our method not only seamlessly integrates with these models but also significantly enhances their performance in terms of prompt adherence and image quality. This demonstrates the versatility and effectiveness of our approach across different conditional generation tasks.

E.5 Heatmap of Alpha Values

Fig. 4(a) and (b) in the main paper show the distribution of α values across various transformer blocks and timesteps, respectively. Here, Fig. 9 provides a more detailed visualization of each block's α values at different timesteps for both models. Actually, the sharp changes described in Fig. 4(c) of the main paper can also be observed in the attention heatmap in Fig. 9b, where some blocks ($41st - 54th$) exhibit significant variations in α values at the timestep 11. In this way, we can clearly see how our *Diff-Aid* dynamically adjusts the block-wise interactions between token-level textual conditions across different timesteps.

E.6 Distribution of α across different timesteps

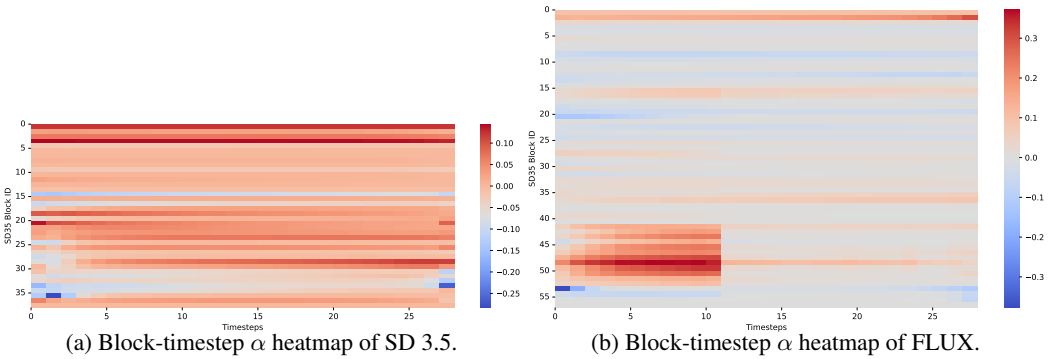
In Fig. 4(c) of the main paper, we plot the distribution of α values across timesteps ($T = 28$) for both SD 3.5 and FLUX. Here, we further visualize the distribution of α values across different timesteps ($T \in \{6, 8, 10, 20, 28, 40, 50, 60\}$) for both models in Fig. 10. As shown in Fig. 10, the distributions of α values at different timesteps exhibit similar patterns and do not vary significantly with the number of sampling steps. As the timestep increases, the curve shows a stable convergence trend,

[3]https://huggingface.co/black-forest-labs/FLUX.1-Canny-dev

[4]https://huggingface.co/black-forest-labs/FLUX.1-Depth-dev

[5]https://huggingface.co/Shakker-Labs/FLUX.1-dev-LoRA-Children-Simple-Sketch

[6]https://civitai.com/models/553811/stained-glass?modelVersionId=771546



(a) Block-timestep α heatmap of SD 3.5. (b) Block-timestep α heatmap of FLUX.

Figure 9: Visualization of the α across different timesteps and blocks for the models.

indicating that the modulation ability of our *Diff-Aid* stabilizes at higher timesteps. This indicates that our *Aid* effectively captures the timestep-aware interactions between token-level textual conditions, maintaining consistent behavior across various sampling step configurations.

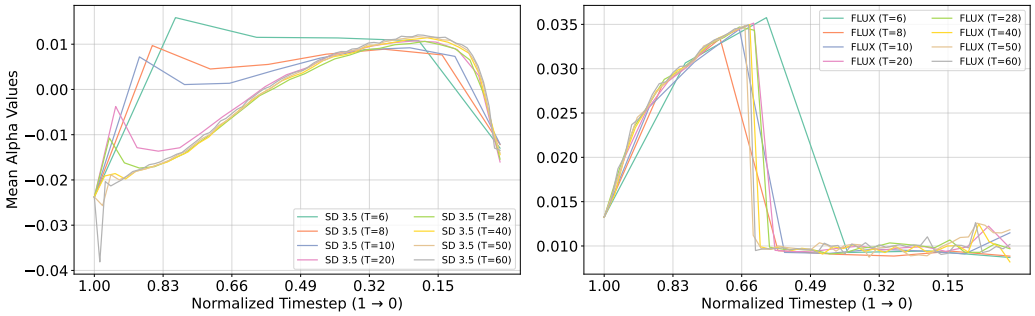


Figure 10: **Distribution of α across different timesteps for both SD 3.5 and FLUX.** The distributions of α values at different timesteps exhibit similar patterns and do not vary significantly with the number of sampling steps.

E.7 Isolating the Contribution of Timestep, Block, and Token Dimensions

To quantify the individual contribution of each dimension in our three-dimensional adaptive modulation design, we conduct controlled ablation studies by removing the adaptivity of one dimension while keeping the other two fully adaptive. Specifically, we evaluate three variants: **(1) Share Blocks**: A single *Aid* is shared across all transformer blocks, removing block-wise adaptivity. **(2) Share Tokens**: A scalar α is shared across all text tokens, removing token-level adaptivity. **(3) Timestep variants**: We test three strategies to remove timestep adaptivity: *mean*—averaging α across all timesteps; *1st hack*—fixing α to values from the 1st denoising step; *10th hack*—fixing α to values from the 10th denoising step.

As shown in Tab. 17 in the main paper, sharing the module across blocks causes the largest HPSv3 drop (-1.26). Sharing one coefficient across prompt tokens reduces HPSv3 by 0.56. Averaging α over timesteps reduces it by 0.38, leaving only a 0.04 gain over FLUX; fixing it to the first or tenth step lowers the score below FLUX. These controls show why the three adaptive dimensions are retained together.

E.8 Effect of L_2 Regularization on Magnitude and Stability

To demonstrate the importance of regularization in our *Aid* design, we analyze how the L_2-norm term $\mathcal{L}_{reg} = \mathbb{E}_{t,l}[\|\alpha_t^l\|_2]$ affects the learned α values. The role of $\mathcal{L}_{reg}$ is not merely to maximize performance metrics, but to keep α within a reasonable range and ensure training stability. As shown

Table 17: **Contribution of three dimension adaptivity on FLUX.**

Setting	HPSv2	HPSv3	ImgRwd	Aes.
FLUX baseline	28.53	10.42	0.89	6.66
Diff-Aid (full)	**28.80**	**10.84**	**0.95**	**6.76**
Shared across blocks	27.41	9.58	0.77	6.34
Shared across tokens	28.07	10.28	0.88	6.48
Time-averaged α	28.63	10.46	0.91	6.53
Fixed 1st-step α	28.56	10.35	0.82	6.33
Fixed 10th-step α	28.53	10.32	0.82	6.35

in Fig. 11, we compare the distribution of α values across transformer blocks with and without regularization.

Without regularization, α exhibits erratic fluctuations and extreme values across blocks, causing unstable modulation that undermines interpretable patterns. In contrast, with regularization ($\lambda_{reg} = 0.1$), the α magnitudes become smaller and more stable, maintaining meaningful token-level patterns while preventing excessive modulation that would distort text feature distributions. The gate separately encourages selective modulation. This demonstrates that regularization is essential for achieving both effective and interpretable modulation in our *Aid*.

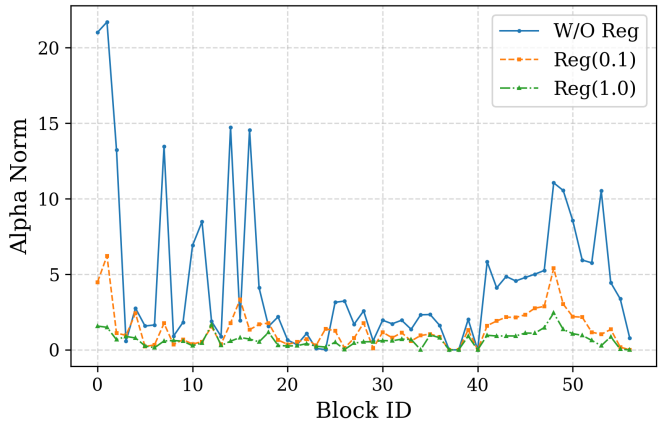


Figure 11: **Effect of regularization on α distribution across transformer blocks.** Without regularization (left), α values show erratic fluctuations and extreme values. With regularization (right), α magnitudes become smaller and more stable, maintaining interpretable modulation patterns.

E.9 More Ablation of dropout probability.

We further investigate the impact of different dropout probabilities in our *Aid* on the performance of FLUX. As shown in Tab. 18, we experiment with dropout probabilities ranging from 0.0 to 0.7. The results indicate that a dropout probability of 0.1 yields the best performance across all metrics. When the dropout probability grows to 0.2, the performance slightly decreases but remains competitive. Big probabilities like 0.5 and 0.7 lead to significant drops in all metrics. These results suggest that incorporating a small amount of dropout helps regularize the model, preventing overfitting while still allowing it to learn effective modulation patterns. In contrast, big p values may cause excessive information loss during training and inference mismatch, leading to degraded performance.

Table 18: More ablation studies on different dropout probabilities.

Categories	Settings	HPSv2	HPSv3	ImgRwd	Aes.
FLUX		28.53	10.42	0.89	6.66
Dropout Probability	p=0.0	28.71	10.74	<u>0.94</u>	6.72
	p=0.1	**28.80**	**10.84**	**0.95**	**6.76**
	p=0.2	<u>28.73</u>	<u>10.77</u>	<u>0.94</u>	<u>6.73</u>
	p=0.3	28.47	10.44	0.87	6.64
	p=0.5	28.21	10.19	0.77	6.59
	p=0.7	27.03	9.42	0.35	6.54

F More Qualitative Results

F.1 Visualization of Adaptive Modulation Patterns: A Case Study

To provide fine-grained interpretability of our Aid module, we present a concrete case study tracing per-token α dynamics for a complex prompt: *"a red cat sitting on a blue chair in a sunny garden"*.

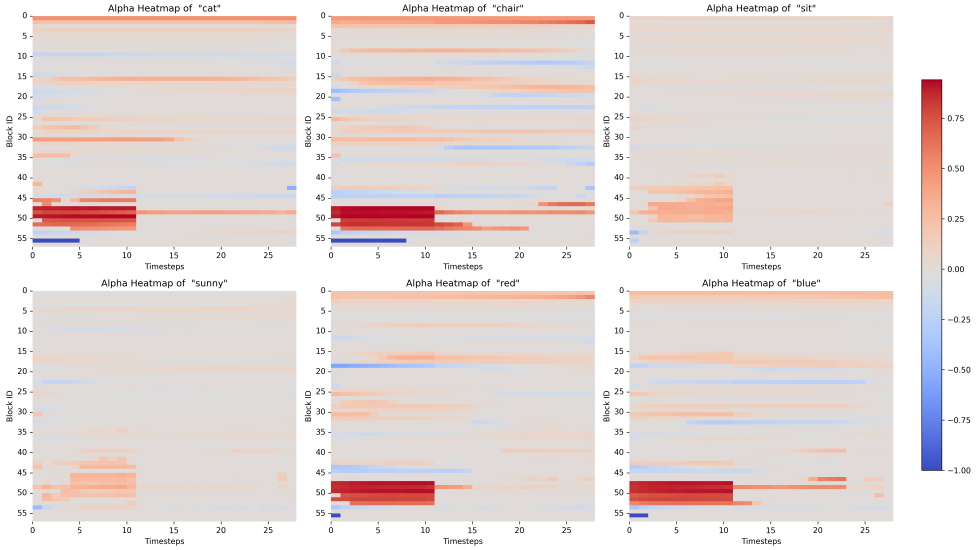


Figure 12: **Per-token α heatmap across blocks and timesteps.** We visualize the adaptive weights α for each token in the prompt *"a red cat sitting on a blue chair in a sunny garden"* across all 57 transformer blocks and 28 denoising timesteps. Subject tokens (*"cat"*, *"chair"*) show consistently high α values, while attribute tokens (*"red"*, *"blue"*) peak in early and late blocks. Relation tokens maintain near-zero values throughout.

Per-token α heatmaps in Fig. 12 reveal distinct modulation patterns: subject tokens (*"cat"*, *"chair"*) exhibit higher and more temporally stable α values across timesteps, while attribute tokens (*"red"*, *"blue"*) show α peaking in early blocks and later blocks (blocks 0-3, 47-53) and early timesteps.

Spatial image-text attention maps further validate this interpretability. Block-wise visualization in Fig. 13 at timestep $t = 10$ shows that the averaged attention across all text tokens is strong in early blocks (0-5) and final blocks (47-53), corresponding to regions of high α norm in the heatmap. Early blocks establish spatial structure, while late blocks refine details. Timestep-wise visualization in Fig. 14 at block 4 shows that the averaged attention map evolves from diffuse (early timesteps) to spatially focused (later timesteps), reflecting the transition from structure generation to semantic refinement. These fine-grained traces demonstrate that Aid's modulation is not only interpretable at the overall distribution level, but also semantically meaningful at the individual token level.

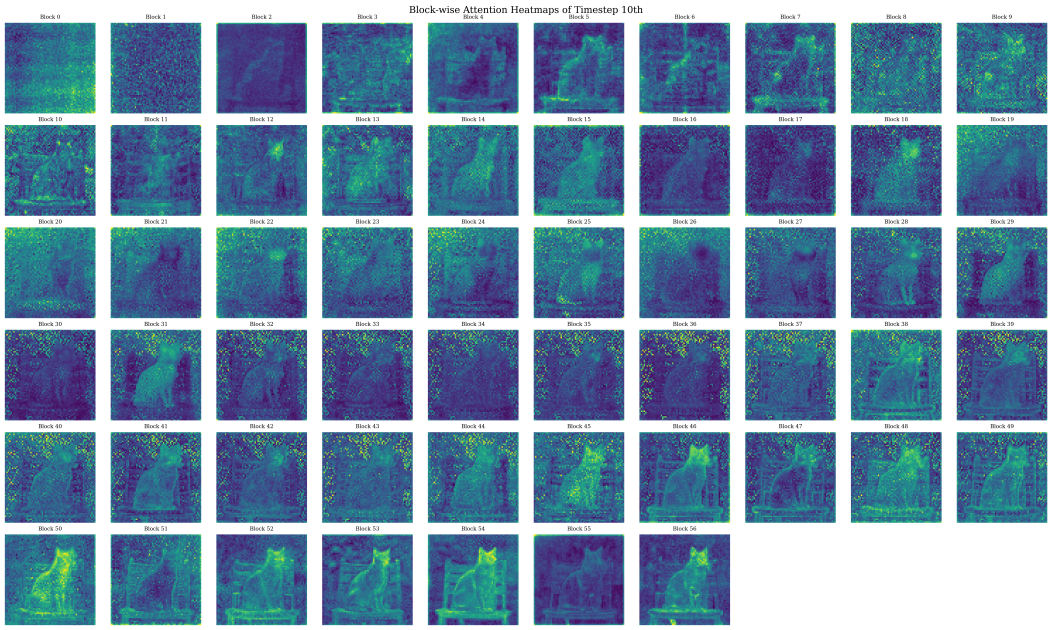


Figure 13: **Block-wise spatial image-text attention maps at timestep $t = 10$.** We visualize the spatial attention averaged across all text tokens in the prompt at different transformer blocks. The averaged attention is strong in early blocks (0-5) and final blocks (47-53), corresponding to regions of high α norm in the heatmap.

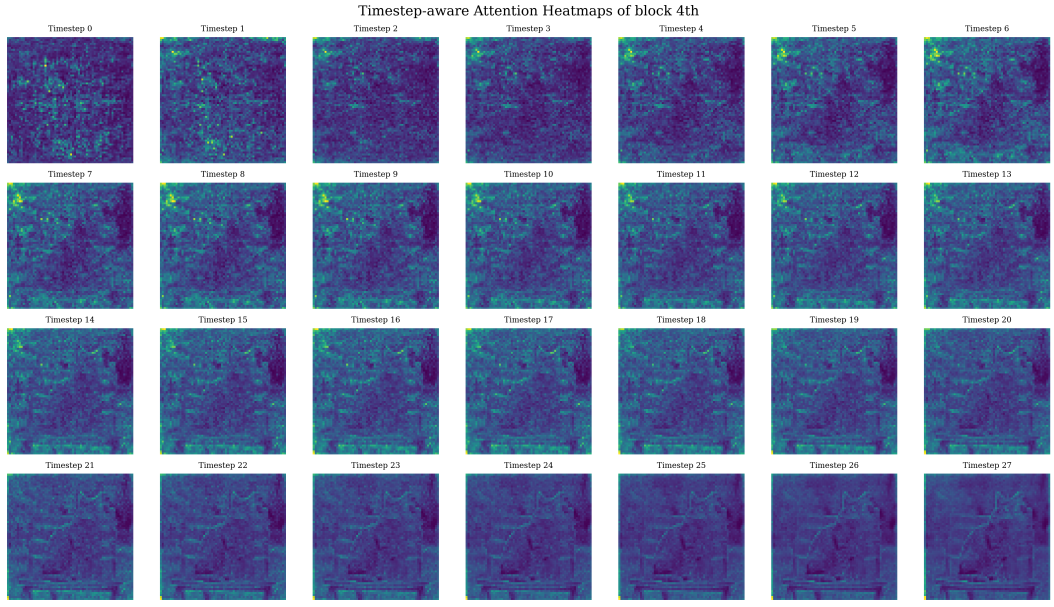


Figure 14: **Timestep-wise spatial image-text attention evolution at block 4.** We visualize the spatial attention averaged across all text tokens in the prompt at different denoising timesteps. The averaged attention map evolves from diffuse (early timesteps) to spatially focused (later timesteps).

F.2 More Results on FLUX and SD 3.5

The quantitative results below each image show the scores of HPSv3, Image Reward, and Aesthetic Score. Our method demonstrates superior prompt adherence and image quality.

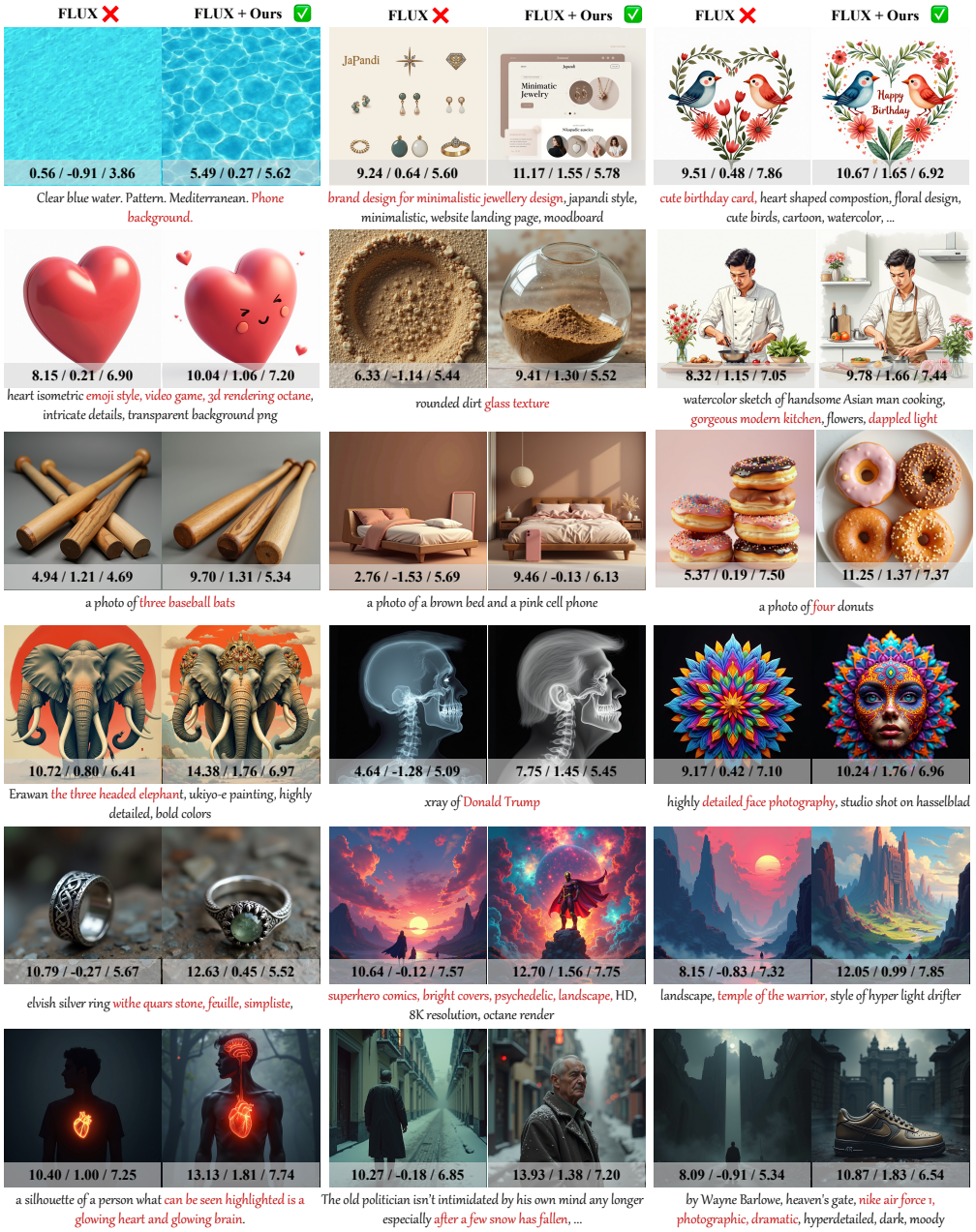


Figure 15: **More Qualitative comparisons of images generated by baseline FLUX with and without our proposed *Diff-Aid*.** The quantitative results below each image show the scores of HPSv3, Image Reward, and Aesthetic Score. Our method demonstrates superior prompt adherence and image quality.



Figure 16: **More Qualitative comparisons of images generated by baseline SD 3.5 with and without our proposed *Diff-Aid*.** The quantitative results below each image show the scores of HPSv3, Image Reward, and Aesthetic Score. Our method demonstrates superior prompt adherence and image quality.

F.3 Failure Cases

Fig. 17 groups the remaining errors by what textual modulation cannot reliably supply. Counting and spatial binding remain difficult: neither method renders exactly five stop signs, and the dog, teddy bear, horse, and vase are not consistently arranged as requested. Abstract or highly constrained descriptions, such as the "soul emerges from body" scene and the detailed dancer prompt, leave substantial room for an incorrect visual interpretation. Rare or ambiguous concepts also expose limitations of the frozen backbone, illustrated by the yarrow flower and "hot dog" examples. In several cases *Diff-Aid* recovers a requested object, but the complete relation or visual style still fails. These examples limit the claim to improved average alignment rather than guaranteed prompt satisfaction.

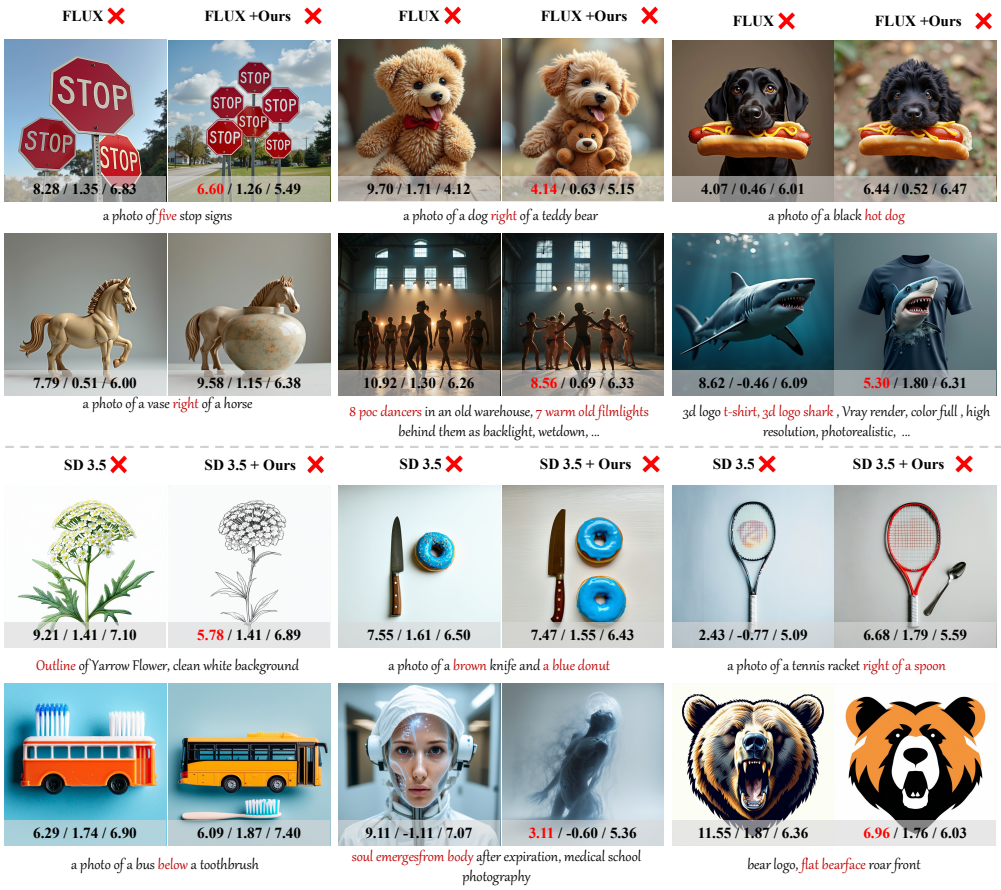


Figure 17: **Some failure cases of baseline models with and without our proposed *Diff-Aid*.** Examples include counting and spatial relations, abstract descriptions, and rare or ambiguous concepts. Both baseline and *Diff-Aid* can fail to satisfy the complete prompt.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We introduce our *Diff-Aid* in the abstract and introduction, provide a detailed description in Section 2, and present experimental results in Section 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The architecture of our module is introduced in Section 2, and the implementation details are presented at the beginning of Section 3 and supplementary sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release our code upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The implementation details are presented at the beginning of Section 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Although we report multiple evaluation metrics (e.g., HPSv2, HPSv3, ImageReward) showing significant improvements over baselines, we did not include error bars, standard deviations, confidence intervals, or statistical significance tests across multiple runs because it would be too computationally expensive. As such, our results do not fully meet the criteria outlined in this question for statistical rigor and uncertainty quantification.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We present the computer resources needed for the experiments at the beginning of Section 3, and time of execution in supplementary sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research were conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact in section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We use the existing excellent SD3.5, Flux.1 and their safeguards. Our model will not bring more risk than SD3.5 or Flux.1.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We work with open-source models that are publicly available, and we cited them properly. Especially, FLUX.1-Dev uses the Non-Commercial License.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The dataset defined for the model training will be made publicly available, in case of acceptance, together with the documentation required for reproducing the experiments. Moreover, in case of acceptance, we will also release the source code of our model.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We recruited 8 volunteers for the blinded pairwise evaluation; the voting protocol is described in the main text.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.