

APPENDIX

We first provide LLM usage statement in Appx. A. We provide preliminaries in Appx. B. In Appx. C, we further analyze the domain gap and structure preservation of diffusion features. Then we elaborate on the implementation details of our proposed method in Appx. D and the experimental setups in Appx. E. We show additional experimental results in Appx. F.

A THE USE OF LARGE LANGUAGE MODELS (LLMs)

We used large language models (LLMs) to assist in refining the paper’s writing and producing the appearance prompt in the ARP module. We used LLMs to enable ARP in the experiments. LLMs played no significant role in the research ideation of this paper.

B PRELIMINARIES

Diffusion Models. Diffusion models are a family of probabilistic generative models characterized by two processes.

The *forward process* iteratively adds Gaussian noise to a clean image \mathbf{x}_0 to obtain \mathbf{x}_t for time step $t \sim [1, T]$, which can be reparameterized in terms of a noise schedule α_t where

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon, \quad (2)$$

for $\epsilon \sim \mathcal{N}(0, \mathbb{I})$.

The *backward process* generates images by iteratively denoising an initial Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbb{I})$, also known as diffusion sampling (Ho et al., 2020). This process uses a parameterized denoising network ϵ_θ conditioned on a text prompt \mathcal{P} , where at time step t we obtain a cleaner \mathbf{x}_{t-1} as

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \hat{\mathbf{x}}_t + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t | t, \mathcal{P}), \quad (3)$$

$$\hat{\mathbf{x}}_t = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta(\mathbf{x}_t | t, \mathcal{P})}{\sqrt{\alpha_t}}. \quad (4)$$

Intuitively, $\hat{\mathbf{x}}_t$ approximates the initial clean image, which is subsequently perturbed with an appropriate amount of noise to produce the input for the following timestep.

Guidance. The iterative inference of diffusion enables people to guide the sampling process on auxiliary information. *Guidance* modifies Eq. (3) to compose additional score functions that point toward richer and specifically conditioned distributions (Bansal et al., 2023; Epstein et al., 2023), expressed as

$$\hat{\epsilon}_\theta(\mathbf{x}_t | t, \mathcal{P}) = \epsilon(\mathbf{x}_t | t, \mathcal{P}) - s \mathbf{g}(\mathbf{x}_t | t, y), \quad (5)$$

where \mathbf{g} is an energy function and s is the guidance strength. In practice, \mathbf{g} can range from classifier-free guidance (where $\mathbf{g} = \epsilon$ and $y = \emptyset$, *i.e.* the empty prompt) to improve image quality and prompt adherence for T2I diffusion (Ho & Salimans, 2021; Rombach et al., 2022), to arbitrary gradients computed from auxiliary models or diffusion features common to guidance-based controllable generation (Bansal et al., 2023; Epstein et al., 2023; Mo et al., 2024). Thus, guidance provides the customizability on the type and variety of conditioning for controllable generation, as it merely requires a differentiable loss with respect to \mathbf{x}_t . However, the need for backpropagation during inference often leads to increased memory consumption and slower inference speed. Moreover, guidance-based methods often fail to capture fine structural details in controllable generation tasks.

Diffusion U-Net architecture. Many pretrained T2I diffusion models are text-conditioned U-Nets, which contain an encoder and a decoder that downsample and then upsample the input \mathbf{x}_t to predict ϵ , with long skip connections between matching encoder and decoder resolutions (Ho et al., 2020; Rombach et al., 2022; Podell et al., 2024). Each encoder/decoder block contains convolution layers, self-attention layers, and cross-attention layers: The first two control both structure and appearance, and the last injects textual information. Thus, many training-free controllable generation methods utilize these layers, through direct manipulation (Hertz et al., 2023; Tumanyan et al., 2023; Kim et al., 2023a; Alaluf et al., 2024; Xu et al., 2024a) or for computing guidance losses (Epstein et al., 2023;

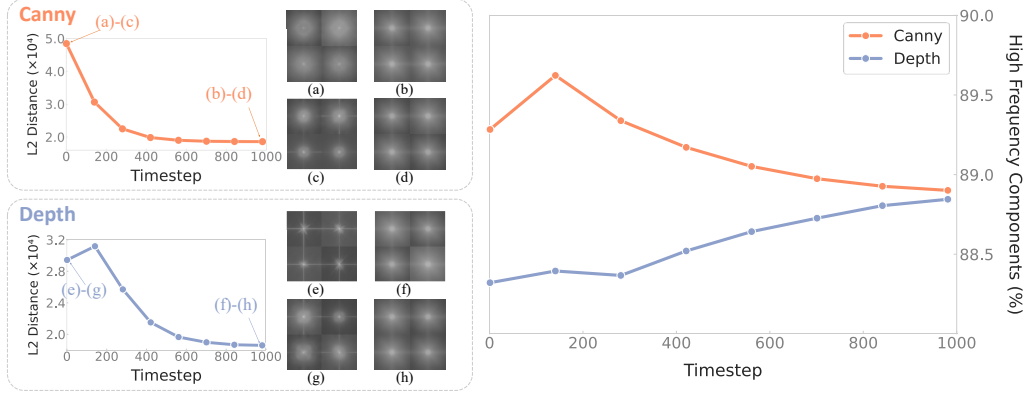


Figure 10: **Fourier analysis of noisy latents under canny edge and depth map conditions.** (Left) Average L2 distance between natural and condition image DFT spectra over timesteps. Subfigures (a)–(d) and (e)–(h) show the DFT spectra of four randomly selected images for both conditions at different timesteps. In each group, (a/e) and (b/f) correspond to condition latents at t_{low} and t_{high} , while (c/g) and (d/h) correspond to natural latents at t_{low} and t_{high} , respectively. (Right) Average high-frequency component ratio over timesteps.

Mo et al., 2024), with self-attention most commonly used. Let $\mathbf{f}_{l,t} \in \mathbb{R}^{HW \times c}$ be the diffusion feature with height H , width W , and channel size c at time step t right before attention layer l . Then, the self-attention operation is

$$\begin{aligned} \mathbf{Q} &= \mathbf{f}_{l,t} \mathbf{W}_l^Q, \quad \mathbf{K} = \mathbf{f}_{l,t} \mathbf{W}_l^K, \quad \mathbf{V} = \mathbf{f}_{l,t} \mathbf{W}_l^V, \\ \mathbf{f}_{l,t} &\leftarrow \mathbf{A} \mathbf{V}, \quad \mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q} \mathbf{K}^\top}{\sqrt{d}} \right), \end{aligned} \quad (6)$$

where $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{c \times d}$ are linear transformations which produce the query \mathbf{Q} , key \mathbf{K} , and value \mathbf{V} , respectively, and d is the dimensionality of the attention space. The softmax operation is applied across the second (HW) -dimension (typically, $c = d$ in diffusion models). Intuitively, the attention map $\mathbf{A} \in \mathbb{R}^{(HW) \times (HW)}$ encodes how each pixel in \mathbf{Q} corresponds to each in \mathbf{K} , which then rearranges and weighs \mathbf{V} . The rich structural information embedded in U-Net features lays the foundation for extensive training-free controllable generation approaches, and, together with the common issues of training-free methods, motivates us to study the temporal dynamics of diffusion features.

C ADDITIONAL ANALYSES

C.1 KL DIVERGENCE

To analyze the domain gap between natural images and condition images, we collect 20 natural images from the *ImageNet-T2IR* dataset from (Tumanyan et al., 2023). Then we use the ControlNet processor (Zhang et al., 2023) to convert these natural images into 5 conditions (canny edge, depth map, normal map, HED edge, and scribble drawing), resulting in 100 natural-condition image pairs.

To quantify the distributional difference, we extract diffusion features at a fixed timestep for each image, flatten them into feature maps (size $(HW) \times F$), and concatenate all features from each domain. We then apply PCA to the combined feature set and retain only the first principal component. Each image is thus projected into a 1-dimensional vector of HW values along this dominant component.

We estimate a probability distribution over these projections for each domain using Gaussian KDE. Specifically, we sample 1000 evenly spaced points between the minimum and maximum values observed in the two distributions. We then compute the KL divergence between the estimated densities of condition and natural images:

$$\text{KL}(P||Q) = \sum_{i=1}^{1000} p(x_i) \log \frac{p(x_i)}{q(x_i)}, \quad (7)$$

where $p(x)$ and $q(x)$ denote the normalized KDE densities of condition and natural images, respectively. We repeat this computation across timesteps to observe how the domain gap evolves during the diffusion process.

C.2 SELF-SIMILARITY

Following (Tumanyan et al., 2022), we adopt the DINO self-similarity distance Caron et al. (2021) to quantify structural similarity between images. In Vision Transformer (ViT) (Dosovitskiy et al., 2021), an image is first divided into a sequence of non-overlapping patches, which are then linearly embedded and processed as tokens. In each Transformer layer, the tokens are projected into queries, keys, and values as follows:

$$\mathbf{Q}_l = \mathbf{T}_{l-1} \mathbf{W}_l^Q, \mathbf{K}_l = \mathbf{T}_{l-1} \mathbf{W}_l^K, \mathbf{V}_l = \mathbf{T}_{l-1} \mathbf{W}_l^V, \quad (8)$$

where $\mathbf{T}_l(\mathbf{I})$ denotes the output tokens for layer l for image \mathbf{I} , and \mathbf{W}_l^Q , \mathbf{W}_l^K , and \mathbf{W}_l^V are the corresponding query, key, and value weight matrices, respectively.

To capture an image’s internal structure, we compute its DINO self-similarity matrix at the final Transformer layer L :

$$S_L(\mathbf{I})_{ij} = \text{cos_sim}(k_L(\mathbf{I})_i, k_L(\mathbf{I})_j), \quad (9)$$

where $\mathbf{K}_L(\mathbf{I}) = [k_L(\mathbf{I})_{cls}, k_L(\mathbf{I})_1, \dots, k_L(\mathbf{I})_n]$ are the key embeddings from the last layer for image \mathbf{I} (n denotes the number of patch tokens), and cos_sim denotes cosine similarity.

As shown in (Tumanyan et al., 2022), this self-similarity-based descriptor can effectively capture the structure of an image while ignoring appearance details. Given two images \mathbf{I}_1 and \mathbf{I}_2 , their structural distance is computed as the ℓ_2 distance between their self-similarity matrices:

$$\mathcal{L}^{\text{struct}} = \|\mathbf{S}_L(\mathbf{I}_1) - \mathbf{S}_L(\mathbf{I}_2)\|_2, \quad (10)$$

where $S_L(\mathbf{I})$ is defined in Eq. (9).

C.3 DISCRETE FOURIER TRANSFORMATION (DFT)

As an alternative to quantifying the domain gap between natural and condition images, we employ the Discrete Fourier Transformation (DFT) to analyze differences in their frequency components. Specifically, we begin by extracting diffusion feature maps for natural and condition images at a fixed timestep, following the method described in Appx. C.1. Since DFT typically operates on spatial images rather than high-dimensional feature tensors, we use the diffusion decoder to transform these feature maps back into RGB images. We then apply DFT to the decoded images to obtain their frequency spectra and compute the L2 distance between the spectra of natural and condition image pairs.

This process is repeated across all diffusion timesteps, and the resulting distances are averaged over the same 100 natural-condition image pairs as described in Appx. C.1. As shown in Fig. 11, the average L2 distance between the frequency spectra decreases progressively as the diffusion timestep increases. This trend indicates that the diffusion process gradually reduces the frequency-domain gap between natural and condition images—consistent with our findings in Sec. 3.

To further investigate how frequency components evolve through the diffusion process, we conduct a detailed analysis on two representative conditions: canny edge and depth map. Intuitively, canny edges, characterized by sharp edges and detailed contours, are expected to exhibit a higher proportion of high-frequency components in their DFT spectrum. In contrast, depth maps tend to be dominated by smooth gradients, suggesting a stronger presence of low-frequency components.

As illustrated in Fig. 10, the average L2 distance between the DFT spectra of natural and condition latents decreases over time for both conditions, consistent with the trend shown in Fig. 11. We also

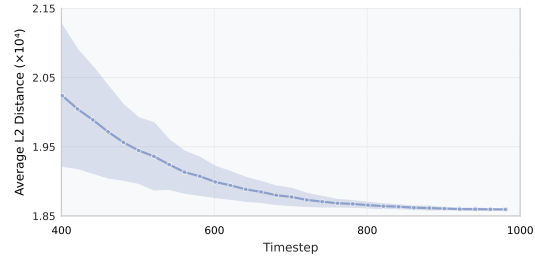


Figure 11: **Average L2 distance between natural and condition image DFT spectra over diffusion timesteps .** Results are averaged over all five conditions.

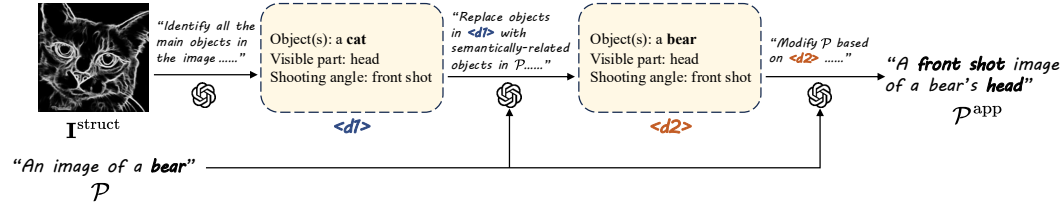


Figure 12: **Illustration of the Appearance-Rich Prompting (ARP) module.** Given the original text prompt \mathcal{P} , our module derives an appearance-rich prompt \mathcal{P}^{app} by integrating semantic information from the condition image $\mathbf{I}^{\text{struct}}$.

visualize DFT spectra of both image types at two representative timesteps of the denoising trajectory, denoted as t_{low} and t_{high} . In practice, we set $t_{\text{low}} = 1$ and $t_{\text{high}} = 981$. Since SDXL inference performs 50 denoising steps, steps 1 and 981 correspond to the lowest and highest noise levels in the denoising process. At t_{low} , canny edge spectra exhibit dispersed high-activation regions, indicative of prominent high-frequency composition. In contrast, depth map spectra show energy concentrated near the center, reflecting low-frequency dominance. Both differ markedly from the corresponding spectra of natural images. At t_{high} , due to accumulated noise, the DFT spectra for all images become visually similar.

This pattern is further confirmed by the right panel of Fig. 10, which plots the ratio of high-frequency components—defined as the proportion of DFT energy outside a centered circle with a radius equal to one-sixth of the image size—over timesteps. Initially, canny features are dominated by high-frequency content, while depth exhibits more low-frequency patterns. These differences gradually converge, reflecting a narrowing frequency-domain gap between different conditions.

D METHOD DETAILS

D.1 SPATIALLY-AWARE APPEARANCE TRANSFER

We build our method on top of the spatially-aware appearance transfer mechanism proposed in Ctrl-X (Lin et al., 2024). Specifically, given diffusion features $\mathbf{f}_{l,t}^{\text{out}}$ and $\mathbf{f}_{l,t}^{\text{app}}$ from the output and appearance branches respectively, Ctrl-X (Lin et al., 2024) computes a cross-image attention map as follows:

$$\mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q}^{\text{out}} \mathbf{K}^{\text{app}^\top}}{\sqrt{d}} \right), \quad (11)$$

$$\mathbf{Q}^{\text{out}} = \text{norm}(\mathbf{f}_{l,t}^{\text{out}}) \mathbf{W}_l^Q, \quad \mathbf{K}^{\text{app}} = \text{norm}(\mathbf{f}_{l,t}^{\text{app}}) \mathbf{W}_l^K,$$

where $\text{norm}(\cdot)$ is applied across the spatial dimension (HW) and removes global statistics across spatial dimensions to isolate structural correspondence.

Subsequently, attention-weighted statistics are computed from the appearance features:

$$\begin{aligned} \mathbf{M} &= \mathbf{A} \mathbf{f}_{l,t}^{\text{app}}, \\ \mathbf{S} &= \sqrt{\mathbf{A}(\mathbf{f}_{l,t}^{\text{app}} \odot \mathbf{f}_{l,t}^{\text{app}}) - (\mathbf{M} \odot \mathbf{M})}, \end{aligned} \quad (12)$$

which are then used to modulate the output features:

$$\mathbf{f}_{l,t}^{\text{out}} \leftarrow \mathbf{S} \odot \mathbf{f}_{l,t}^{\text{out}} + \mathbf{M}. \quad (13)$$

D.2 APPEARANCE-RICH PROMPTING

Directly using the original prompt for appearance transfer may lead to artifacts in the generated image, since such prompts tend to be brief and lacking in semantic correspondence with the condition image (see Sec. 4.3 for details). To overcome this limitation, we propose a pipeline that enriches the original text prompt \mathcal{P} with semantic information extracted from the structure condition image $\mathbf{I}^{\text{struct}}$, yielding a more appearance-rich prompt \mathcal{P}^{app} for generating the final appearance image \mathbf{I}^{app} . As illustrated in Fig. 12, we first utilize GPT-4o Achiam et al. (2023) to extract key semantic entities from the

condition image to produce dictionary $\langle d1 \rangle$. To facilitate semantic alignment between the condition image and the text prompt, we further employ GPT-4o Achiam et al. (2023) to identify and associate these extracted entities with semantically-related elements in the original text, modifying $\langle d1 \rangle$ to produce $\langle d2 \rangle$. Finally, we revise the original prompt \mathcal{P} using the extracted semantic information, producing an enhanced appearance prompt \mathcal{P}^{app} . To help the multimodal LLM correctly follow instructions and mitigate erroneous semantic transfer, our pipeline stores intermediate information in structured dictionaries, enabling more controlled and interpretable prompt editing. More examples of the Appearance-Rich Prompting (ARP) module are provided in Fig. 15. For the full prompt used with GPT-4o Achiam et al. (2023) for Appearance-Rich Prompting, see the accompanying .txt file in the supplementary material.

E EXPERIMENT SETUP DETAILS

E.1 IMPLEMENTATION DETAILS

We implement our method with Diffusers (von Platen et al., 2022) on SDXL 1.0 (Podell et al., 2024) and adopt the same injection layers following previous work (Lin et al., 2024). We sample \mathbf{I} with 50 steps of DDIM sampling and set $\eta = 1$ (Song et al., 2021). For structure-rich injection, we set $\tau = 400$ and $C = 600$.

For restart refinement, we set $\sigma_{t_{\min}} = 1.0$, $\sigma_{t_{\max}} = 2.0$, $N = 3$, $S = 5$, where S is the total number of timesteps in the restart backward process. **For the restart backward process, we adopt the same noise schedule as the base model, SDXL (Podell et al., 2024), which is:**

$$\sigma_{\min} = \sqrt{\frac{\beta_{\min}}{1 - \beta_{\min}}}, \quad \sigma_{\max} = \sqrt{\frac{\beta_{\max}}{1 - \beta_{\max}}}, \quad (14)$$

$$\sigma_t = \sigma_{\min} - (\sigma_{\max} - \sigma_{\min}) \frac{t}{T - 1}, \quad \alpha_t = \frac{1}{1 + \sigma_t^2}, \quad \beta_t = 1 - \alpha_t, \quad (15)$$

where $\beta_{\min} = 0.00085$ and $\beta_{\max} = 0.012$.

For self-recurrence, we set $t'_{\min} = 500$, $t'_{\max} = 900$, $N' = 2$, where t'_{\max} is the self-recurrence starting point, t'_{\min} is the self-recurrence end point, and N' is the number of self-recurrence (Lin et al., 2024). We run most experiments on NVIDIA Tesla V100 GPUs. For FreeControl (Mo et al., 2024), InfEdit (Xu et al., 2024a), and computational efficiency comparisons, we run the experiments on A800 GPUs.

For any input condition image $\mathbf{I}^{\text{struct}}$, we preprocess it with a dilation and unsharp masking operation. Specifically, we binarize the image, perform a distance transform operation to detect the minimum line width w . If $w_{\min} \leq w \leq w_{\max}$, we dilate $\mathbf{I}^{\text{struct}}$ with kernel size k^e . On the other hand, if the inverted image meets the standard, we erode $\mathbf{I}^{\text{struct}}$. We set $w_{\min} = 25$, $w_{\max} = 50$ and $k^e = 10$. Then we perform unsharp masking $(1 + \gamma) \cdot \mathbf{I}^{\text{dilate}} - \gamma \cdot B$ to modify the dilated (eroded) image, where $\mathbf{I}^{\text{dilate}}$ denotes the dilated (eroded) input condition image, $\gamma = 50$, and B is the Gaussian blur operation with blur radius $r = 3$. We empirically find the two operations beneficial for highlighting object boundaries and improving structure preservation.

E.2 DATASET DETAILS

We construct our dataset based on the conditional generation datasets from Ctrl-X (Lin et al., 2024) and FreeControl (Mo et al., 2024). Specifically, for conditions canny edge, depth map, normal map, HED edge, and scribble drawing, we select condition-prompt pairs from both datasets and merge them. We collect a total of 15 condition images per condition and form 22 condition-prompt pairs for canny edge and 21 pairs for each of the remaining four conditions.

For human pose and segmentation map, since both original datasets contain limited examples, we supplement them by collecting additional human pose images from the web and segmentation masks from the ADE20K (Zhou et al., 2017) dataset. We obtain 15 images for each of these two conditions and pair them with text prompts using a combination of templates and hand annotation, resulting in 21 image-prompt pairs for human pose and 23 for segmentation mask.

(11/30) Please select the best image, considering its structure alignment with the Input Image, semantic consistency with the Input Text, and visual quality.

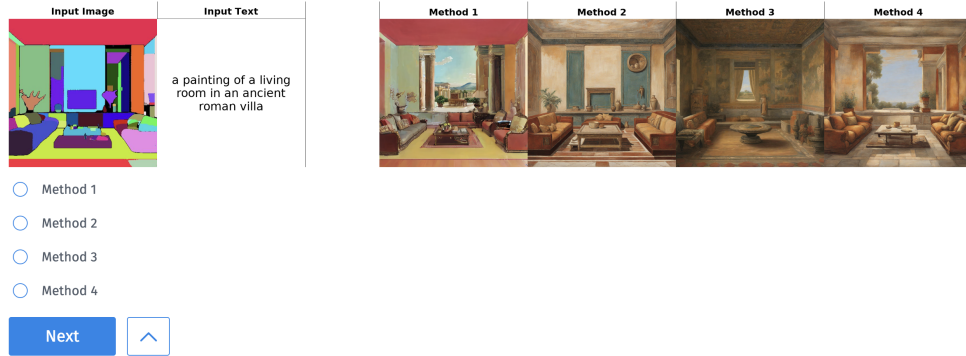


Figure 13: **Screenshot of the user study interface.** Participants are presented with the inputs and asked to select the best result from four randomly shuffled candidates.

E.3 USER STUDY DETAILS

We hereby provide the detailed protocol used for our subjective user study. For each case, participants with relevant expertise are asked to select the best image from four anonymous, randomly shuffled results according to a holistic criterion. The instruction provided in the questionnaire is: *Please select the best image, considering its structural alignment with the Input Image, semantic consistency with the Input Text, and visual quality.* Fig. 13 shows the interface of the user study.

E.4 COMPUTATIONAL EFFICIENCY EXPERIMENT DETAILS

We evaluate the baselines on our dataset to compare their average inference time and memory usage. Specifically, we implement FreeControl (Mo et al., 2024), Ctrl-X (Lin et al., 2024), and our method using SDXL 1.0 (Podell et al., 2024) checkpoints. For InfEdit (Xu et al., 2024a), we utilize the LCM Dreamshaper v7 (Luo et al., 2023) checkpoint (based on SD1.5), as it is the only model provided in their official codebase. To ensure a fair comparison, we generate 1024×1024 images using 50 sampling steps for all methods.

F ADDITIONAL EXPERIMENTAL RESULTS

F.1 COMPUTATIONAL EFFICIENCY

Complementing the overall efficiency comparison against prior works in Tab. 1, we further analyze the specific runtime contribution of each module within our pipeline in Tab. 3. The SRI module, which dominates the computation (85.1%), represents the core injection framework responsible for handling both condition and appearance image features. Notably, the additional latency introduced by the RR and ARP modules is marginal, accounting for only 6.8% and 8.1% of the total inference time, respectively. Despite their low computational overhead, these components play a critical role in significantly enhancing image quality, as evidenced by the ablation study in Tab. 2.

Table 3: Proportion of inference time consumed by each module of our method.

Module	Percentage of Inference Time
SRI	85.1%
RR	6.8%
ARP	8.1%

Table 4: **Additional quantitative comparison of controllable T2I.** Our method consistently surpasses all training-free baselines in structure preservation, image-text alignment, and visual diversity. The best results are in **bold**, and the second best are underlined.

Method	Self-sim ↓	CLIP ↑	LPIPS ↑	Dream-Sim ↓	Image-Reward ↑	HPSv2 ↑
ControlNet (Zhang et al., 2023)	0.067	0.309	0.701	0.509	0.298	0.285
T2I-Adapter (Mou et al., 2024)	0.116	0.287	0.728	0.636	-0.050	0.261
SDEdit (Meng et al., 2022)	0.154	0.259	0.315	0.734	-1.374	0.189
P2P (Hertz et al., 2023)	0.197	0.251	0.266	0.724	-1.786	0.168
PnP (Tumanyan et al., 2023)	0.157	0.256	0.151	0.724	-1.789	0.168
InfEdit (Xu et al., 2024a)	0.135	0.296	0.357	0.636	-0.202	0.244
FreeControl (Mo et al., 2024)	0.116	<u>0.320</u>	0.667	0.626	<u>0.554</u>	<u>0.285</u>
Ctrl-X (Lin et al., 2024)	<u>0.104</u>	0.315	0.650	<u>0.579</u>	0.291	0.283
Ours	0.096	0.322	<u>0.662</u>	0.558	0.897	0.313

F.2 ADDITIONAL QUALITATIVE RESULTS

We provide additional qualitative comparisons with baselines in Fig. 19 and additional qualitative results for a broader range of condition types in Fig. 20. Our method demonstrates strong generation performance across both common and challenging conditions. It also handles diverse and complex text prompts effectively. As a training-free approach, it generalizes effortlessly to various in-the-wild conditions without any additional training cost, producing high-quality outputs. This level of zero-shot generalization is often unattainable for training-based methods.

F.3 ADDITIONAL QUANTITATIVE RESULTS

Since T2I-Adapter-SDXL (Mou et al., 2024) supports only four (canny, depth, normal, and pose) out of the seven condition types in our dataset, we further conduct a quantitative comparison limited to these four types. As shown in Tab. 4, our method outperforms all baselines across almost every metric. Notably, these metrics jointly assess both structure preservation (e.g., DINO self-similarity (Tumanyan et al., 2022), DreamSim (Fu et al., 2023)) and generation quality (e.g., ImageReward (Xu et al., 2023a), HPSv2 (Wu et al., 2023)), highlighting the effectiveness of our approach.

F.4 ADDITIONAL ABLATION STUDY

We present additional ablation studies on key components of our proposed method to validate our design choices. The results are shown in Fig. 14, Fig. 15, and Fig. 16.

Structure-Rich Injection. As a complementary study to the SRI ablation presented in Sec. 5.3, we further investigate the choice of constants in the case of constant injection. Specifically, we evaluate the effects of the injection schedule $g(t) = C$ across various C values. As shown in Fig. 14, lower C values result in severe conditional leakage due to a pronounced domain gap (e.g. $C = 0$). In contrast, higher values of C (e.g. $C = 800$) produce more natural appearances with higher fidelity but compromise structural control. Empirically, $C = 600$ achieves the best balance between appearance fidelity and structure control, significantly outperforming the synchronous injection baseline by enhancing structural alignment, suppressing condition leakage and reducing visual artifacts simultaneously.

Appearance-Rich Prompting. Fig. 15 demonstrates the effectiveness of appearance-rich prompting in enhancing semantic alignment between the structural condition and the appearance image. This strategy helps recover missing semantic elements (e.g., “building” in row 1 and “hand” in row 3), significantly reducing visual artifacts and improving the overall quality of the generated images.

Restart Refinement. Fig. 16 illustrates the efficacy of restart refinement in mitigating visual artifacts (e.g., the duplicated eye on the rabbit’s body and the incorrect eyes in the husky’s background). Additionally, it alleviates condition leakage under abstract conditions (e.g., pose), further improving generation fidelity.

The number of restart iterations N . We also conduct an ablation study of restart iterations N . As shown in Fig. 17, setting $N = 1$ is not adequate for suppressing visual artifacts, and both $N = 3$ and $N = 5$ yield high-quality outputs. Consequently, we set $N = 3$ for optimal visual quality and computational efficiency.

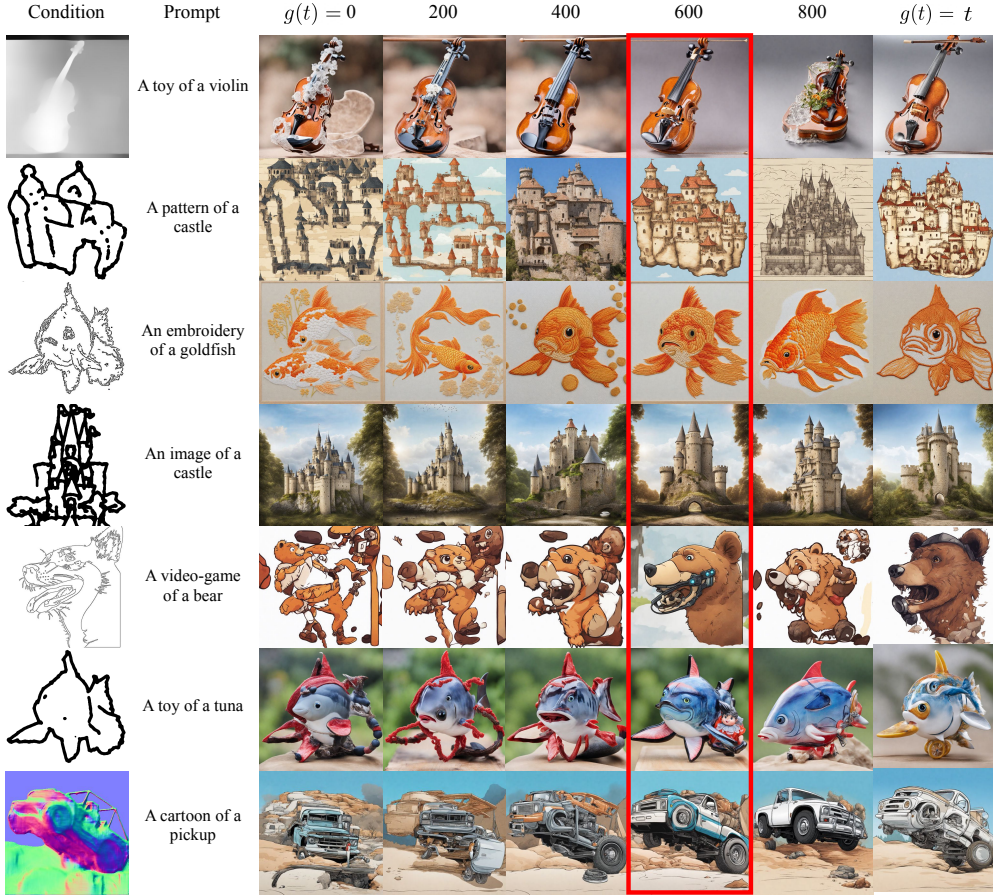


Figure 14: **Additional ablation of Structure-Rich Injection (SRI).** For asynchronous injection $g(t) = C$, lower C suffers from conditional leakage, while higher values improve appearance fidelity at the cost of structural control. The optimal trade-off is achieved at $C = 600$, outperforming the synchronous schedule ($g(t) = t$).

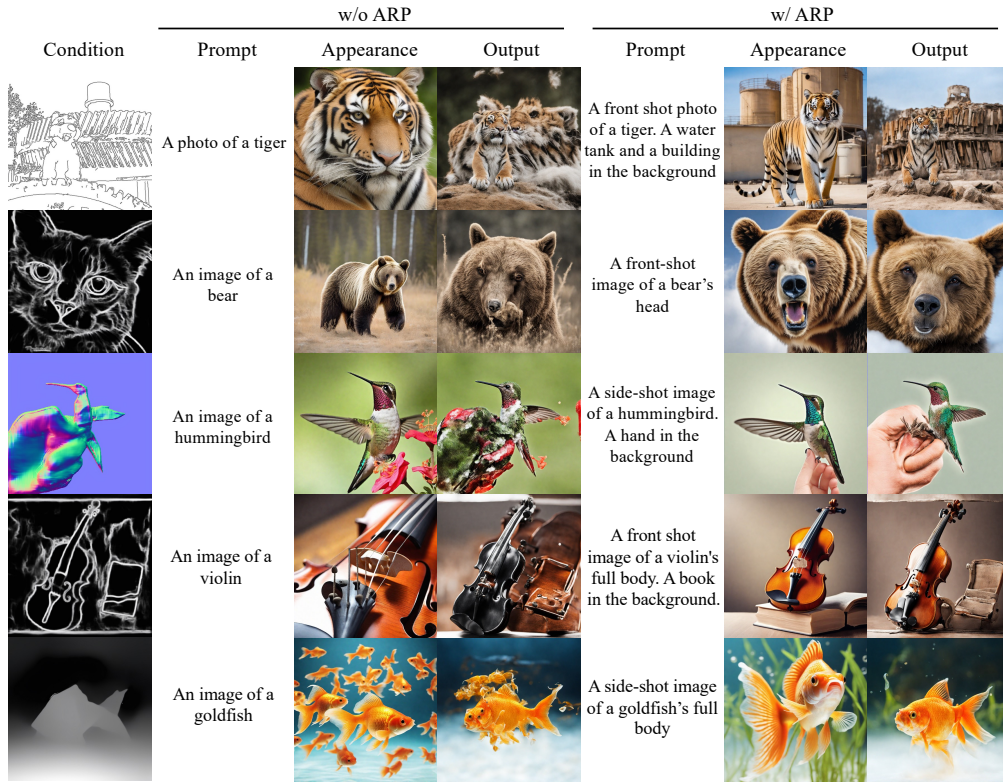


Figure 15: **Additional ablation of Appearance-Rich Prompting (ARP)**. This module improves semantic alignment with the condition image by adapting prompts to better capture key visual attributes, thereby mitigating incorrect appearance transfers and reducing artifacts.

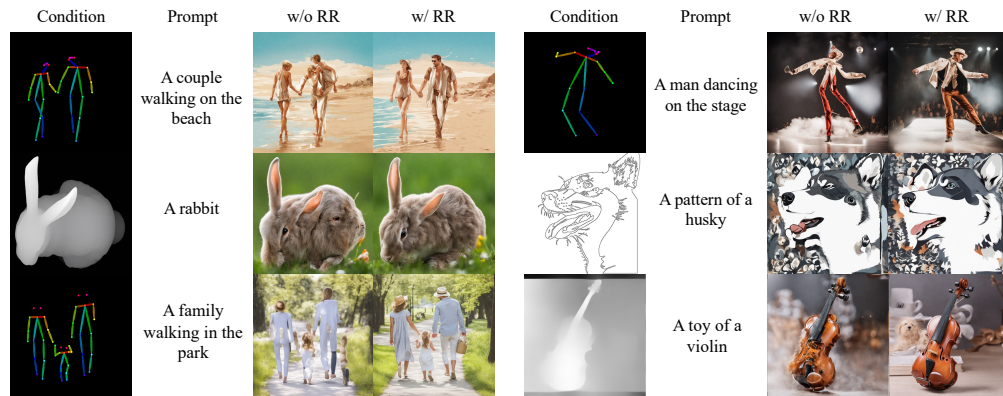


Figure 16: **Additional ablation of Restart Refinement (RR)**. This strategy significantly mitigates condition leakage and appearance artifacts, improving generation quality while maintaining structural alignment.

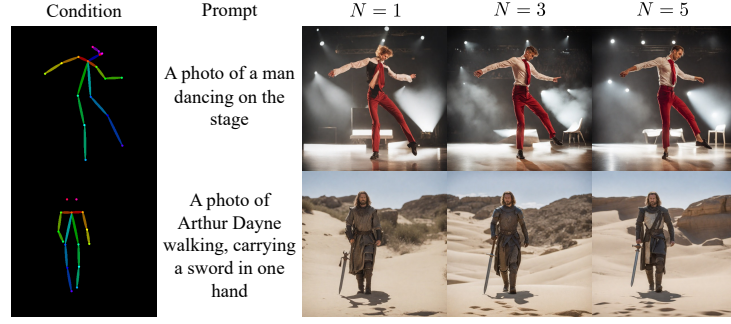


Figure 17: **Additional ablation of restart iterations N .** Setting $N = 1$ is not adequate for suppressing visual artifacts, and both $N = 3$ and $N = 5$ yield high-quality outputs. Consequently, we set $N = 3$ for optimal visual quality and computational efficiency.

F.5 EXPERIMENTS ON DiT-BASED ARCHITECTURES

While our main experiments focused on U-Net-based models (*e.g.*, SDXL, SD1.5) for conditional T2I generation, we conducted exploratory experiments to extend our feature injection paradigm to the Diffusion Transformer (DiT) architecture, specifically FLUX (Labs, 2024). We adopted two distinct strategies to select effective injection layers. First, following the recent analysis by Avrahami et al. (2025), we injected condition features exclusively into the identified “vital layers”¹ of FLUX. As shown in Fig. 18, this strategy resulted in negligible structural alignment, indicating that the control signal was insufficient to modulate the deep multimodal attention layers of DiT. To strengthen the control, we subsequently attempted feature injection across all 56 layers. However, this setting resulted in severe condition leakage: the model over-prioritized the condition features, reproducing only the coarse structure of the input while failing to generate coherent textures, leading to significantly degraded appearance quality.

Identifying the optimal injection layers for structure and appearance control within FLUX is a non-trivial task, given the combinatorial search space of 2^{56} possible subsets and the complex text–image interactions inherent in its multimodal attention layers. Designing a robust, training-free structure-and-appearance control framework for DiT requires extensive investigation that lies beyond the scope of this work. We therefore leave this exploration to future work.

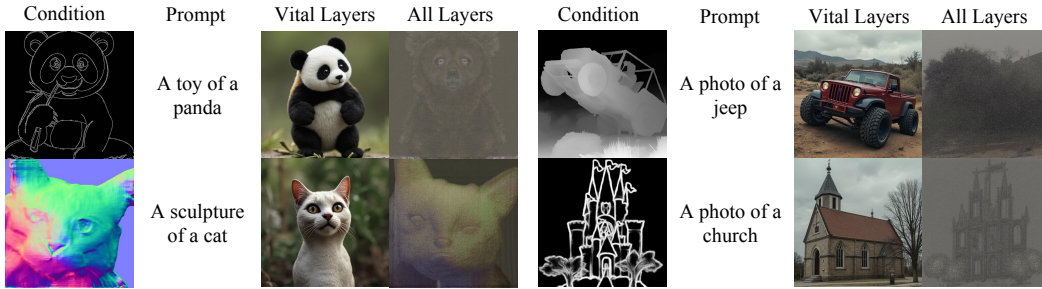


Figure 18: **Results of feature injection for structure control in DiT.** Injecting vital layers (Avrahami et al., 2025) results in inadequate structure alignment, whereas injecting all layers leads to severe condition leakage.

¹The authors of StableFlow (Avrahami et al., 2025) identified nine vital layers for training-free image editing in FLUX through removal-influence analysis. They are layers 0, 1, 17, 18, 25, 28, 53, 54, 56.

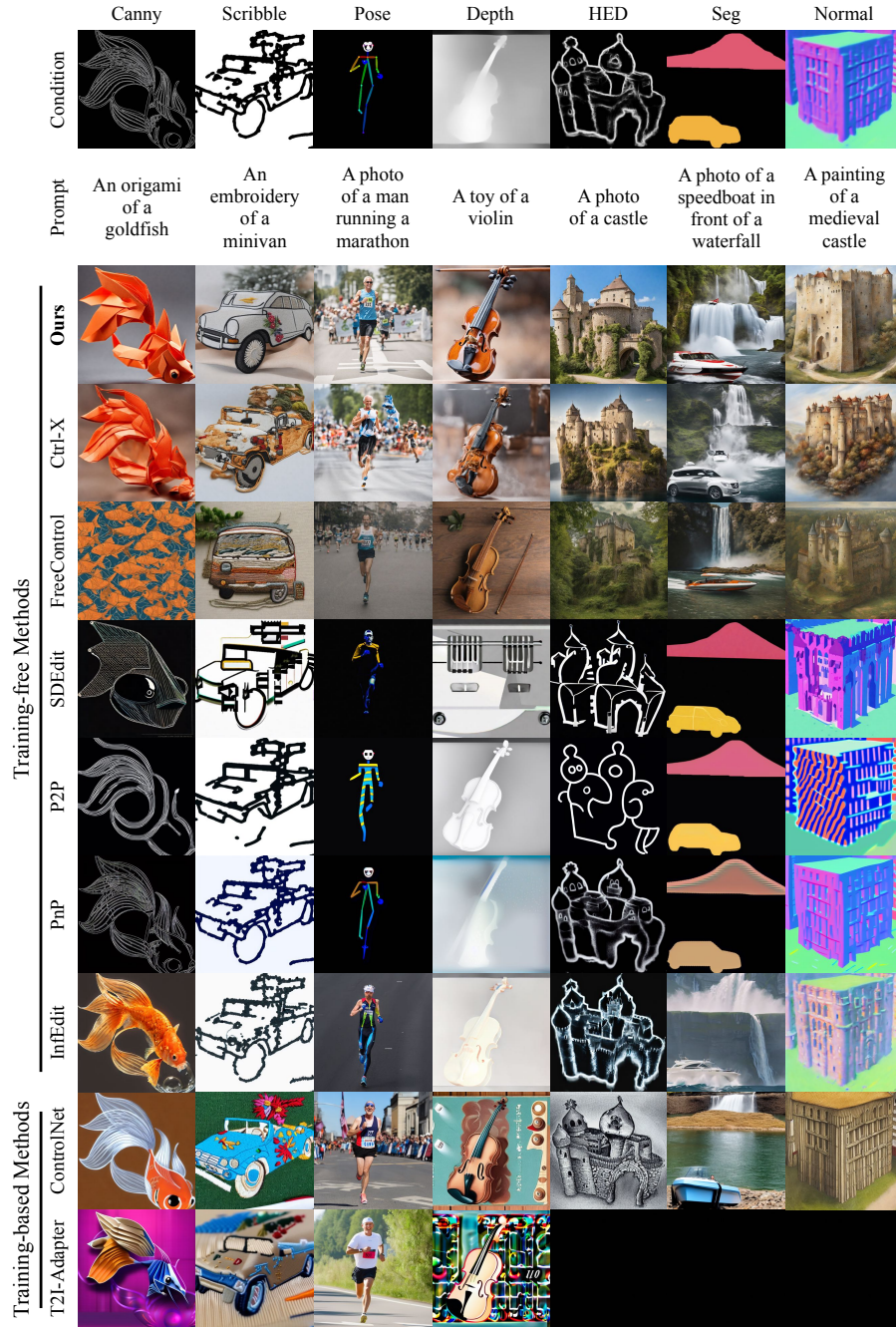


Figure 19: Qualitative comparison with existing methods.

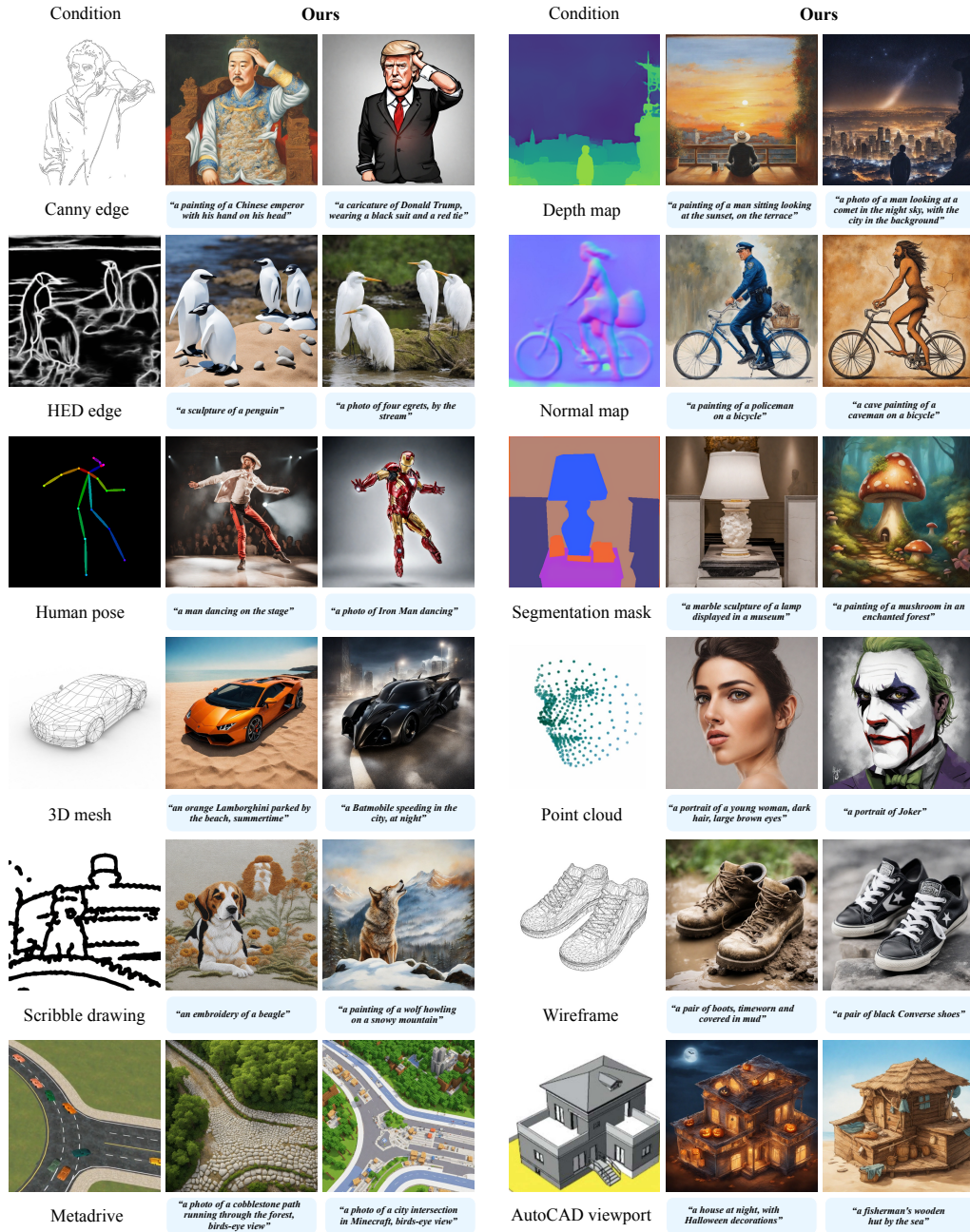


Figure 20: Qualitative results for more control conditions.