SALVATION LIES WITHIN: ELICITING INHERENT STYLE TRANSFER IN STEP-DISTILLED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting step-distilled text-to-image (T2I) models through post-training incurs additional computational costs and affects native few-step generation behavior. This motivates a complementary route beyond style-specific adaptation: drawing on the visual knowledge already encoded in step-distilled T2I models to elicit stylistic capabilities through language. Pursuing this direction requires textual guidance that captures how visual attributes jointly define a style and remain applicable as the depicted content changes. To explore this approach, we introduce StyleForge, a fully automatic, training-free framework that expresses reference styles as reusable rendering instructions. By integrating overall rendering characteristics with local color and lighting behavior, StyleForge organizes visual evidence from reference images into a coherent specification of how the target style should be expressed. The specification is then compiled into textual guidance that can be reused across content prompts, enabling frozen step-distilled T2I models to render different subjects and scenes in the reference style while retaining native few-step generation. Extensive experiments show relative gains of up to 29.47% in generation quality scores over the strongest baseline, while Pareto analysis indicates that improved stylization is accompanied by strong adherence to the requested content.

1 INTRODUCTION

Text-to-image (T2I) diffusion models have advanced rapidly (Yang et al., 2022; Cao et al., 2024), enabling a wide range of applications in digital art (Gao et al., 2025), graphic design (Wang et al., 2025), and visual content creation (Brooks et al., 2023). However, conventional diffusion models typically require many sequential denoising evaluations, leading to high computational costs and inference latency. To make generation more efficient, step-distillation techniques have emerged (Lin et al., 2024; Ge et al., 2026; Luo et al., 2025), enabling high-quality synthesis in only a few denoising steps (Saharia et al., 2022).

Among the applications of step-distilled T2I models, style transfer is particularly demanding, for it requires a model to preserve the rendering characteristics of a reference while synthesizing new semantic content. Classical style transfer methods restyle a given content image through feature-statistics alignment or adaptive feature transformation (Gatys et al., 2016; Huang & Belongie, 2017). Recent style transfer approaches shift the focus from restyling existing images to synthesizing new subjects and scenes in a reference style with T2I models through learnable style representations (Sohn et al., 2023; Hertz et al., 2024; Wang et al., 2024). However, in the few-step setting, adapting these models to new styles requires preserving the generation behavior learned during distillation. Prior work has shown that conventional post-training methods, such as supervised fine-tuning (SFT), can disrupt this behavior and produce blurry outputs (Jiang et al., 2026a), motivating training strategies tailored to this setting. A representative effort in this direction is D-OPSD (Jiang et al., 2026a), which uses on-policy self-distillation to learn new styles while retaining few-step generation quality.

Despite recent progress, such adaptation-based approaches still incur additional training costs, and adaptation remains a delicate process that can compromise the model's original few-step generation capability (Jiang et al., 2026a). These limitations raise a natural question: *Can the inherent capabilities of step-distilled T2I models be elicited for style transfer?* This possibility is motivated by

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the remarkably broad visual vocabulary that modern step-distilled T2I models already command. Even simple textual descriptions can evoke recognizable stylistic characteristics, suggesting that the capabilities needed for style transfer may already exist within step-distilled models. The challenge is therefore to identify transferable style information and organize it into textual guidance that elicits the intended style across different subjects and scenes.

Visual style is characterized by recurring rendering attributes and the relationships that organize them into a coherent appearance. Successful style transfer requires both fidelity to these characteristics in the reference and adherence to the content specified by the target prompt. Meeting both requirements through textual guidance presents three challenges: (i) Style descriptions may conflate reusable rendering attributes with subjects and layout, introducing unwanted content constraints in new scenes. (ii) Hue (e.g., red or blue) alone does not specify a color's visual role or spatial extent: the same hue may cover a broad material region or appear as a local luminous accent. (iii) Independently listed visual attributes can leave local surface and color details disconnected from the overall rendering manner, resulting in fragmented guidance for the step-distilled T2I model.

To address these challenges, we propose **StyleForge**, a fully automatic framework that enables style transfer by eliciting the inherent capabilities of step-distilled T2I models. The framework organizes visual attributes into a structured style state according to their transfer behavior, specifying which information should be preserved, weakened, or excluded when content changes. Within this state, color and light are specified jointly with their visual roles, transferable surface types, spatial extent, and distribution. The state is then compiled into textual guidance that places local surface and color attributes within the overall rendering manner. The resulting style prompt can be combined with different content prompts to guide the frozen step-distilled T2I model in rendering new content in the specified style. Our contributions are summarized as follows:

- We offer a distinct perspective on style transfer in the step-distilled setting that centers on eliciting the model's existing visual knowledge. Specifically, we frame the style transfer task as constructing textual specifications of rendering behavior, enabling stylization across content without affecting the model's native few-step generation capability.
- Guided by the view that step-distilled T2I models already possess rich visual knowledge, we propose StyleForge, an agent-driven framework that expresses style as structured rendering instructions by jointly modeling rendering, color, and lighting at both global and local levels.
- Extensive experiments across diverse settings show that the StyleForge framework consistently outperforms a range of baselines in generation quality, with relative gains of up to 29.47% over the strongest baseline. Pareto analysis further demonstrates that StyleForge achieves a favorable balance between style fidelity and content adherence.

2 Related Work

Diffusion-Based Image Generation. Diffusion-based T2I generation has progressed from latent-space denoising (Rombach et al., 2022) to scalable multimodal rectified-flow transformers (Esser et al., 2024), improving image fidelity and prompt following. Alongside these quality improvements, inference efficiency has become an important goal for practical deployment, as also reflected in recent commercial models (OpenAI, 2026). Step distillation addresses this need by enabling image generation in one or a few sampling steps. Representative methods learn consistent mappings along sampling trajectories (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023), match generated and target distributions (Yin et al., 2024a;b), or incorporate adversarial objectives (Sauer et al., 2023). More recent work refines distribution matching through Decoupled-DMD (Liu et al., 2025) and integrates distillation with reinforcement learning in DMDR (Jiang et al., 2026b). Recent model releases further demonstrate the practical viability of few-step synthesis: the distilled FLUX.2 [klein] 4B and 9B variants (Black Forest Labs, 2026) support image generation and editing in four steps, while Z-Image-Turbo (Z-Image Team et al., 2025) combines a compact 6B backbone with eight-step generation. Their low sampling budgets make interactive content creation and deployment on consumer hardware increasingly practical. We therefore focus on step-distilled T2I models, whose strong visual capabilities and low sampling cost favor widespread adoption.

Style Transfer. Early style transfer relies on feature-statistic matching (Gatys et al., 2016; Huang & Belongie, 2017; He et al., 2024). With pretrained T2I models, reference-based customization becomes

possible through fine-tuning, as exemplified by DreamBooth for subject-driven generation (Ruiz et al., 2023). Further work on style learning and reference-guided generation can be found in Gal et al. (2022); Sohn et al. (2023); Hertz et al. (2024); Wang et al. (2024). However, extending style adaptation to step-distilled generators introduces an additional constraint: learning the target style must preserve the generation behavior acquired during distillation. Conventional denoising or flow-matching fine-tuning optimizes states derived from noised target images, which can differ from those visited by the model's few-step sampler; this mismatch can disrupt the distilled behavior and produce blurry outputs (Miao et al., 2025; Jiang et al., 2026a). To address this issue, PSO (Miao et al., 2025) uses pairwise sample optimization to directly tune timestep-distilled models. D-OPSD (Jiang et al., 2026a) adopts on-policy self-distillation, using a teacher with richer multimodal conditioning to supervise the student's own sampling trajectories and learn new styles while retaining few-step generation quality. These methods enable style adaptation in the few-step regime through additional training. StyleForge explores a complementary approach by converting reference images into reusable textual rendering instructions that elicit the frozen generator's existing style capabilities while preserving its native few-step sampling procedure.

3 METHOD

3.1 FRAMEWORK OVERVIEW: FROM MOTIVATION TO STYLE-GUIDE GENERATION

One established line of work approaches style transfer through post-training. Related work, such as D-OPSD (Jiang et al., 2026a), adapts model parameters to a given reference style, maintaining a style-specific learned state. Such adaptation incurs additional training costs for each target style and may disrupt the model's distilled generation behavior, degrading few-step image quality. This motivates a central question: can style transfer be achieved without per-style post-training? Our key insight is that modern step-distilled diffusion models already possess broad visual priors that can be elicited through precise textual descriptions of rendering attributes. Figure 1 illustrates two aspects of this control. In (a), the requested blue and yellow colors appear on the robot's body and scarf, respectively. In (b), the phrases "mostly" and "a small bottom layer" yield a dominant cyan region above a thin magenta layer in the glass. These examples suggest that textual instructions can specify visual attributes together with the surfaces they apply to, their relative extent, and their spatial arrangement. This motivates expressing style as a structured specification of rendering behavior.



Figure 1: Precise textual control of rendering attributes.

Motivated by this observation, we propose StyleForge, an agent-driven framework for style transfer, as illustrated in Fig. 2. Given reference images, the agent extracts a style state and compiles it into a generator-facing style prompt. For each generation prompt, the system reuses this style prompt, combines it with the content prompt, and passes the combined prompt to a step-distilled diffusion model to produce an image in the reference style. Formally, given a reference image set $\mathcal{R}$ and a generation prompt c, StyleForge computes as follows

$$\begin{cases} S_{\mathcal{R}} = \Psi\big(\Phi(\mathcal{R}; q),\, H(\mathcal{R})\big), & (1) \\ p_{\mathcal{R}} = C(S_{\mathcal{R}}; \pi), & (2) \\ x = G\big(p_{\mathcal{R}}\,; c\big), & (3) \end{cases}$$

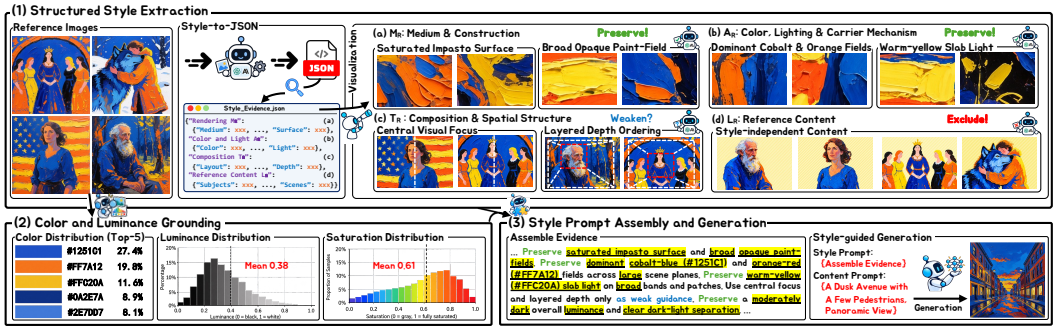

Figure 2: Overview of StyleForge. (1) The agent extracts structured style evidence from reference images. (2) Color and luminance statistics ground the extracted evidence. (3) The agent compiles a reusable style prompt and combines it with new content prompts to guide a step-distilled T2I model.

where $S_{\mathcal{R}}$ is the style state extracted from the reference images by the agent, $p_{\mathcal{R}}$ is the style prompt compiled from $S_{\mathcal{R}}$, and x is the image generated by the model. Specifically, the multimodal model Φ extracts structured style evidence from the reference images according to parsing protocol q. The color and luminance statistics module H supplements this evidence with information about global value, chroma, dominant hues, and related properties. The state integration function Ψ organizes these two sources of evidence into the style state $S_{\mathcal{R}}$. The prompt compiler C, following the fixed ordering π, first describes the overall rendering manner, then explains how surfaces and colors realize that manner, placing local attributes within the context of the global style, and converts the result into the style prompt $p_{\mathcal{R}}$. Finally, the style prompt $p_{\mathcal{R}}$ is concatenated with the generation prompt c and passed to the step-distilled generator G to obtain x. We detail this process in the following sections.

3.2 State Design: Identifying Transferable Information

In analyzing the reference images, the visual style agent first distinguishes rendering practices that can be reused across scenes from descriptions tied to the reference content. Lines and surface qualities are usually transferable. Composition, including the number, positions, and spatial relations of subjects, is tightly coupled to the content. Concrete subjects, scenes, and objects belong to the reference content and may carry that content into a new scene if copied directly. To make the role of each description explicit when content is replaced, StyleForge stores every description together with its transfer policy in an internal state:

$$S_{\mathcal{R}} = (M_{\mathcal{R}},\, A_{\mathcal{R}},\, T_{\mathcal{R}},\, L_{\mathcal{R}}), \tag{4}$$

where the rendering state $M_{\mathcal{R}}$ represents transferable rendering practices, the color-and-light state $A_{\mathcal{R}}$ represents colors and illumination, the composition state $T_{\mathcal{R}}$ represents compositional habits, and the reference-content record $L_{\mathcal{R}}$ represents the reference content. This partition is organized around two requirements under content replacement: preserve visual mechanisms that can be reused across contents, while preventing concrete content from the reference images from entering the new scene. The role of each state component is detailed below:

- **Rendering state $M_{\mathcal{R}}$.** The state describes line and edge rhythm, shape and proportion, texture and brushwork, detail density, ornamental contours, and visual flow, because these factors primarily determine how the image is rendered and generally do not depend on the identity of a subject.
- **Color-and-light state $A_{\mathcal{R}}$.** The state records the surfaces on which colors and light act, their spatial extent, and their distribution, because merely listing color names cannot distinguish a large-area material color from a local highlight or emissive accent, causing colors to detach from their carriers.
- **Composition state $T_{\mathcal{R}}$.** The state records spatial habits such as centered placement, shallow or layered depth, and negative space. These relations are tightly coupled to the number, position, and actions of the subjects, thus, the weak constraint on $T_{\mathcal{R}}$ retains only spatial tendencies that can transfer across scenes and does not fix the particular layout of the references.
- **Reference-content record $L_{\mathcal{R}}$.** The record stores the concrete subjects, props, scenes, and enclosing structures in the reference images. It is used only to check that the style prompt excludes reference content and does not participate in positive generation.

Table 1: Transfer policies for the four types of information in the style state $S_{\mathcal{R}}$.

Category	Description	Transfer Policy
Rendering $M_{\mathcal{R}}$	Medium, Surface, Line, Shape, Texture, Detail Density, etc.	Preserve ✓
Color and Light $A_{\mathcal{R}}$	Carrier Surfaces, Extent, Distribution of Colors, Light, etc.	Preserve ✓
Composition $T_{\mathcal{R}}$	Layout, Depth, Negative Space, etc.	Weaken ↓
Reference Content $L_{\mathcal{R}}$	Subjects, Props, Scenes, Enclosing Structures, etc.	Exclude ✗

Accordingly, the rendering state $M_{\mathcal{R}}$ and color-and-light state $A_{\mathcal{R}}$ are written into the style prompt, the composition state $T_{\mathcal{R}}$ is retained only as a weak constraint, and the reference-content record $L_{\mathcal{R}}$ is used solely for subsequent checks. The transfer policies for the four state components are summarized in Table 1.

3.3 Color Allocation: Preserving Carriers and Distribution

The previous section places the color-and-light state $A_{\mathcal{R}}$ among the transferable style mechanisms. To achieve this transfer, the agent must preserve not only hue but also how each color functions in the image: the same hue may form a large-area material color or merely a local highlight or emissive accent, and these roles differ in their carrier surfaces, spatial extent, and distribution. The visual style agent therefore first determines the visual role of a color, and then organizes that role together with its carrier, scale, and color anchor as

$$a = (\rho,\, h,\, \kappa,\, s,\, g,\, \ell) \in A_{\mathcal{R}}, \tag{5}$$

where a denotes one color/light allocation, ρ specifies the visual role and scale of the color (e.g., dominant field color, large-area material color, medium light, small accent), h provides the hexadecimal color anchor, κ identifies the carrier surface to which the color is attached, s specifies the spatial extent of the color, g specifies the color's spatial distribution, and ℓ specifies the color-attribute type (e.g., material color, material light, secondary illumination, shadow structure, accent).

Based on the color/light allocation records in $A_{\mathcal{R}}$, the agent abstracts concrete carriers into transferable surface types and generates corresponding extent constraints for each color. For example, a warm orange on a small luminous ornament in a reference image is described as a color accent within fine-scale luminous regions, while retaining its color anchor and restricting its coverage. When the subject or scene changes, the carrier may change, but the accent's role, scale, and distribution remain constrained by the same record. The organization of color can therefore be reused without importing the original ornament.

3.4 Execution Pipeline: Building the Style State and Generating Images

The preceding sections introduced the state design and the color-allocation rules. This section instantiates them as an auditable pipeline executed by the visual style agent. The agent first parses and integrates the reference image set $\mathcal{R}$ into a style state $S_{\mathcal{R}}$, then converts $S_{\mathcal{R}}$ into a style prompt $p_{\mathcal{R}}$, and finally combines $p_{\mathcal{R}}$ with a new generation prompt to produce an image.

- **Structured extraction (reference image set $\mathcal{R}$ → structured style evidence).** To make visual observations computable and checkable, the visual style agent invokes the multimodal model Φ with the reference-image parsing protocol q to analyze $\mathcal{R}$. Protocol q instructs the model to extract rendering practices shared across the images, together with colors and their carriers. The protocol also instructs the model to record subjects, props, and scenes that must not be transferred. Based on the extracted information, the model outputs style evidence in a structured format (e.g., JSON), organized into four categories: rendering $M_{\mathcal{R}}$, color and light $A_{\mathcal{R}}$, composition $T_{\mathcal{R}}$, and reference content $L_{\mathcal{R}}$. To ensure structural validity and prevent reference-content leakage, the agent validates the evidence before passing it to subsequent modules. All required fields must be present, and each color record must include both a color and its carrier surface. Rendering descriptions must exclude reference subjects, props, and scenes.
- **Color and luminance statistics (reference image set $\mathcal{R}$ → color and luminance distribution).** To supplement global tonal information that is difficult to assess reliably from language alone, the

visual style agent invokes the color and luminance statistics module H. It computes luminance, saturation, hue, and dominant color clusters over $\mathcal{R}$, and summarizes them as global value, chroma, dominant hues, and their coverage relations. For example, if the statistics show that the references are dominated by high-chroma blue, the compiler requires blue to appear on carrier regions covering a substantial area rather than only as a local accent.

- **State mapping (structured style evidence + color and luminance distribution $\rightarrow$ style state $S_{\mathcal{R}}$).** The visual style agent uses the state integration function Ψ to convert the evidence from the preceding two steps into a unified style state $S_{\mathcal{R}} = (M_{\mathcal{R}}, A_{\mathcal{R}}, T_{\mathcal{R}}, L_{\mathcal{R}})$:
  - **Categorization:** Rendering fields are written to $M_{\mathcal{R}}$. Color roles, carrier surfaces, and color/luminance statistics are written to $A_{\mathcal{R}}$. Transferable spatial tendencies are written to $T_{\mathcal{R}}$. Subjects, props, scenes, and concrete enclosing structures are written to $L_{\mathcal{R}}$.
  - **Abstraction:** Carrier descriptions tied to the reference content are rewritten as transferable visual mechanisms, while the original carriers are retained in $L_{\mathcal{R}}$. For example, "segmented curved illumination on a window" is abstracted as "segmented curved luminous organization".
  - **Fusion and policy annotation:** Color descriptions in the structured style evidence are merged with the color and luminance statistics, and each record is annotated as preserve, weaken, or exclude. For example, if the structured evidence describes a warm color as a local highlight and the color statistics also show that it is concentrated in a small region, the "local highlight" color record is preserved while its concrete carrier is marked as reference content and excluded.
- **Prompt compilation (style state $S_{\mathcal{R}} \rightarrow$ style prompt $p_{\mathcal{R}}$).** Compilation rewrites the structured fields of the style state into natural-language style instructions usable for image generation. The visual style agent invokes the prompt compiler C, which follows the fixed ordering π to compile the transferable state into $p_{\mathcal{R}}$: it first describes the overall rendering manner, then the scope in which that manner operates, and finally the colors and their carriers. The composition state $T_{\mathcal{R}}$ appears only as a weak hint. Before generation, the agent checks that $p_{\mathcal{R}}$ contains none of the subjects, props, scenes, or concrete enclosing structures in $L_{\mathcal{R}}$, and that all extracted rendering practices and color allocations have been written into the style prompt. Once the checks pass, the agent concatenates $p_{\mathcal{R}}$ with the generation prompt c and feeds the result to the generator G to obtain x.

4 Experiments

4.1 Experiment Setup

We evaluate StyleForge on MegaStyle-1.4M (Gao et al., 2026), a diverse testbed for style transfer. For efficiency, we randomly sample 31 style categories and generate 16 scenes per category for each method, averaging metrics over these scenes. We instantiate the few-step distilled regime with FLUX.2-Klein (4B and 9B), holding its native 4-step inference path and the target prompts fixed across methods. All methods use a resolution of 1024×1024 and the shared set of 16 content prompts. Additional visual results across generators are provided in Appendix A.3. We compare StyleForge with five baselines: Vanilla SFT, PSO (Miao et al., 2025), D-OPSD (Jiang et al., 2026a), Direct Style Extraction (DSE), and Reference-Guided Generation (RCG). For quantitative comparison, we use the VLM-based *Score* (Jiang et al., 2026a) and pairwise *Win Rate* (Jiang et al., 2024) to assess overall style-transfer quality. The former rates each generated image from 1 to 4 according to its adherence to the target content and fidelity to the reference style, while the latter measures the proportion of direct comparisons won by a method. Both metrics are obtained from judgments by Qwen3-VL-8B-Instruct, Qwen3-VL-32B-Instruct (Bai et al., 2025), and Qwen3.5-9B[1]. The content and style tradeoff is examined through four complementary metrics in the Pareto analysis. *CLIP-Text* (Radford et al., 2021) and *VLM-F Macro* (Lan et al., 2026) measure content fidelity, while *CLIP-Style* (Radford et al., 2021) and *VLM-S Macro* (Lan et al., 2026) measure style similarity. Higher values indicate better performance for all metrics. Detailed baseline descriptions and implementation details are provided in Appendix A.1.

[1] https://qwen.ai/blog?id=qwen3.5

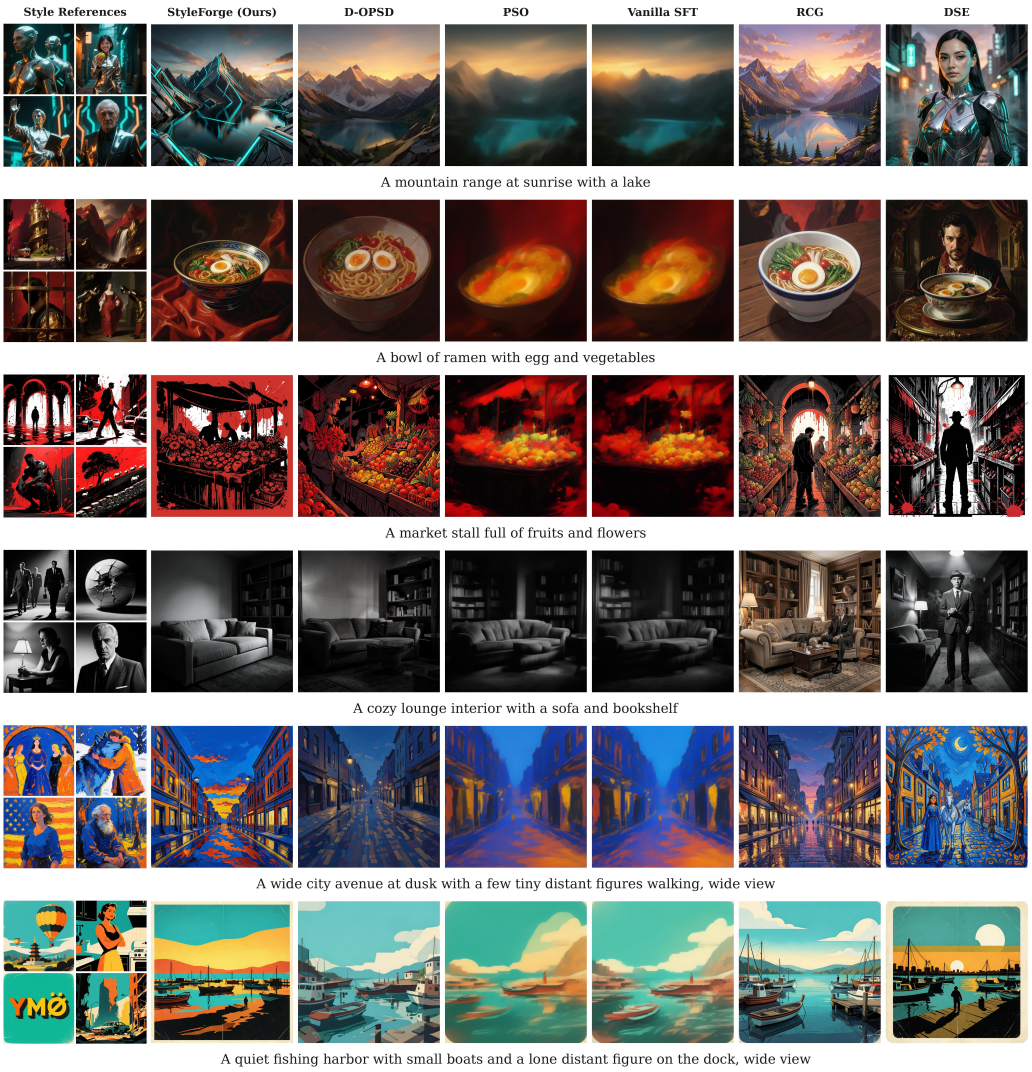

Figure 3: Qualitative comparison with baselines on FLUX.2-Klein-4B.

4.2 Results and Analysis

4.2.1 Qualitative Comparison

Figures 3 and 4 present qualitative comparisons of StyleForge with baselines on FLUX.2-Klein-4B and FLUX.2-Klein-9B, respectively. In Figure 3, StyleForge transfers the reference style more clearly than the baselines: the mountain scene (row 1, column 2) takes on the neon-lit metallic look of the cyberpunk references, and the market scene (row 3, column 2) takes on the bold red-and-black graphic look of its references, including the ink-splash accents. Figure 4 shows a similar conclusion on the 9B backbone: the forest trail (row 1, column 2) keeps the soft golden haze of its references, and the vintage car (row 6, column 2) keeps the soft ink-wash look of its references.

Compared with StyleForge, the qualitative results highlight three issues observed in the baseline outputs. (1) **Reference content leakage.** DSE may introduce objects from the references that are not requested by the target prompt. For example, in Figure 3 (row 1, column 7), it generates a cyberpunk woman even though the prompt asks only for a mountain range at sunrise with a lake, clearly carrying reference content into the output. (2) **Weak style preservation.** Baselines often fail to retain distinctive visual cues from the references. For example, in Figure 3 (row 1, columns 3–6), D-OPSD, PSO, Vanilla SFT, and RCG render a generic sunrise landscape instead of the neon-lit metallic look

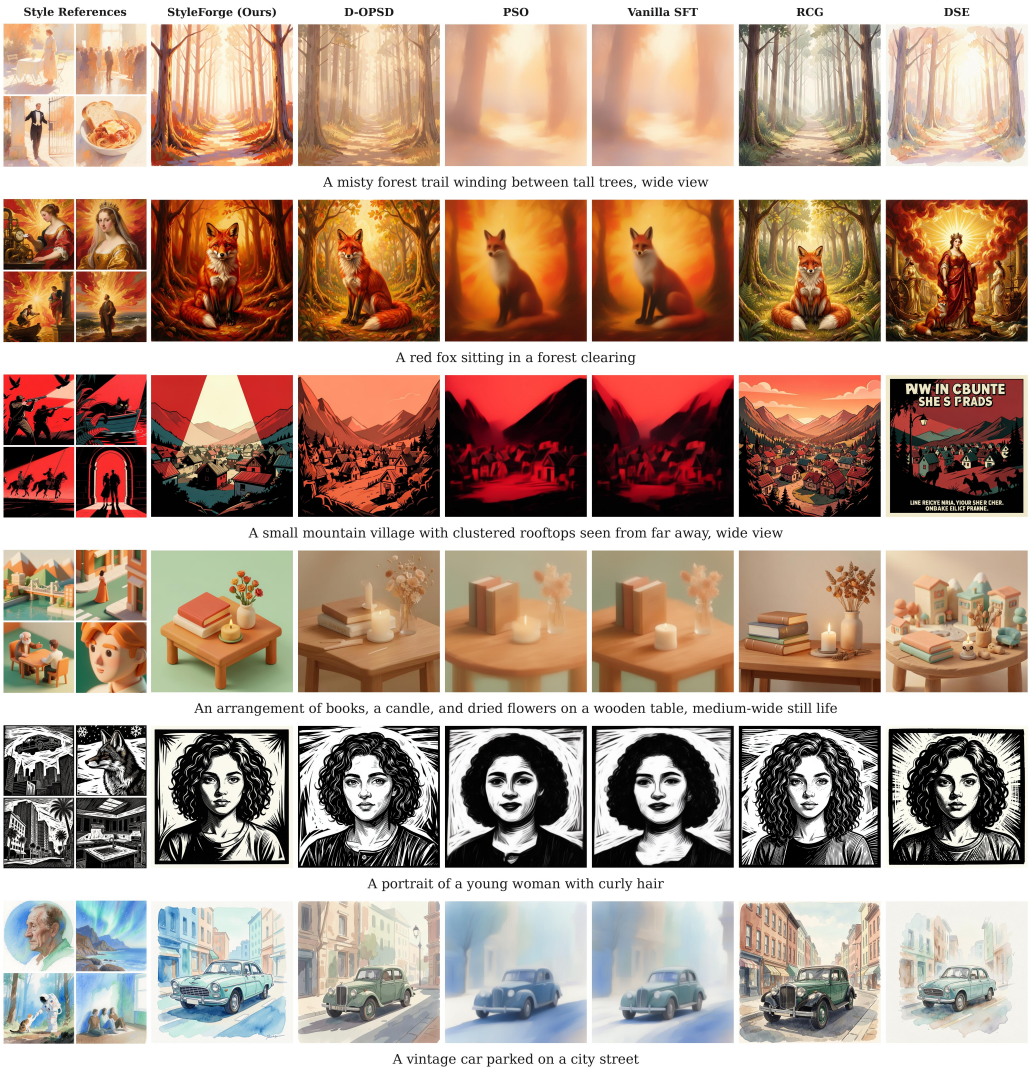

Figure 4: Qualitative comparison with baselines on FLUX.2-Klein-9B.

of the cyberpunk references. (3) **Degraded few-step generation.** Adaptation-based baselines often blur outputs or lose high-frequency detail under few-step generation. For example, in Figure 3 (row 2, columns 3–5), D-OPSD, PSO, and Vanilla SFT reduce the ramen scene to diffuse orange blobs, erasing bowl edges and noodle structure. Across these cases, StyleForge better preserves reference style while keeping prompt-aligned content and clear detail under few-step generation.

4.2.2 Quantitative Comparison

Table 2 and Figure 5 provide complementary quantitative views of style transfer. Table 2 summarizes overall transfer quality through Score and pairwise Win Rate across VLM judges and FLUX.2-Klein backbones, whereas Figure 5 evaluates the content and style tradeoff by plotting content fidelity against style similarity. The Pareto plot in Figure 5 uses the VLM-F Macro and VLM-S Macro scores judged by Qwen3-VL-32B-Instruct; corresponding Pareto plots for Qwen3-VL-8B-Instruct and Qwen3.5-9B are provided in Appendix A.3.6. StyleForge delivers the strongest and most balanced performance, attaining rank 1 across all settings with maximum gains over the second-best method of 29.47% in Score and 26.29% in Win Rate. Averaged across judges, the gains for the 4B backbone are 13.40% in Score and 20.77% in Win Rate, compared with 14.24% and 24.81%, respectively, for the 9B backbone. The Pareto analysis in Figure 5 places StyleForge at the Pareto frontier knee for

both backbones. DSE lies toward the upper-left because reference-content leakage increases style similarity while reducing fidelity to the target content. D-OPSD lies toward the lower-right because post-training degradation blurs the background and leaves the main subject overly salient, inflating apparent content fidelity while weakening style similarity. StyleForge instead lies in the high-content, high-style region, preserving target-prompt content while transferring reference style.

Table 2: VLM evaluation of content and style transfer on FLUX.2-Klein backbones (**Best** , Second Best). Gain is relative to the strongest baseline.

Method	Qwen3-VL-8B-Instruct		Qwen3-VL-32B-Instruct		Qwen3.5-9B		Avg. Rank
	Score	Win Rate (%)	Score	Win Rate (%)	Score	Win Rate (%)	
FLUX.2-Klein-4B Backbone							
DSE	3.226	52.67%	3.187	46.75%	2.077	46.06%	3.667
RCG	2.689	41.84%	2.950	47.53%	1.698	44.12%	4.500
Vanilla SFT	3.769	40.54%	3.300	29.43%	1.506	34.86%	4.167
PSO	3.757	37.82%	3.255	27.49%	1.496	32.65%	5.167
D-OPSD	3.558	57.42%	3.548	66.74%	2.016	65.29%	2.500
StyleForge	**3.908**	**69.70%**	**3.798**	**82.06%**	**2.689**	**77.02%**	**1.000**
Gain	*3.69%↑*	*21.39%↑*	*7.05%↑*	*22.95%↑*	*29.47%↑*	*17.97%↑*	–
FLUX.2-Klein-9B Backbone							
DSE	3.205	53.55%	3.251	45.42%	2.230	46.26%	3.667
RCG	3.126	52.25%	3.358	59.99%	1.983	55.34%	3.833
Vanilla SFT	3.689	35.69%	3.228	27.44%	1.524	30.67%	4.500
PSO	3.541	29.18%	3.073	19.03%	1.394	22.09%	5.667
D-OPSD	3.588	57.68%	3.576	65.45%	2.220	65.04%	2.333
StyleForge	**3.909**	**71.65%**	**3.856**	**82.66%**	**2.875**	**80.60%**	**1.000**
Gain	*5.96%↑*	*24.22%↑*	*7.83%↑*	*26.29%↑*	*28.92%↑*	*23.92%↑*	–

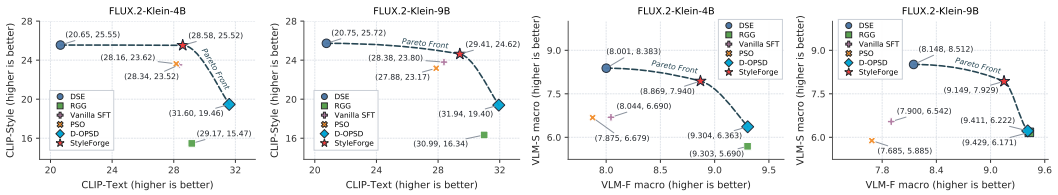

Figure 5: Pareto comparison of content and style across FLUX.2-Klein backbones.

5 Conclusion

This work investigated how the existing visual knowledge of step-distilled diffusion models can support reference-based style transfer without further training. To connect this knowledge with the visual characteristics of a target style, we introduced StyleForge, which expresses the interplay of rendering, color, and light as coherent guidance for image synthesis. By separating these relationships from the particular subjects and scenes in the references, the framework makes the guidance reusable as the requested content changes. This capacity to carry style across content is reflected in our evaluation, where StyleForge improves transfer quality while maintaining a strong balance between stylistic fidelity and content adherence, with the generator retaining its few-step synthesis capability. Taken together, these results highlight the importance of how style is represented when drawing on pretrained visual knowledge, and support explicit language-based guidance as a practical complement to model adaptation.

AI USE STATEMENT

During manuscript preparation, we used large language models solely for language polishing, including improvements to grammar, clarity, and readability. Large language models are also studied as experimental components in this work, but were not used to design the algorithm or derive the scientific contributions. The authors take full responsibility for the final manuscript, including all AI-assisted revisions and the accuracy, originality, and integrity of its content.

ETHICS STATEMENT

This work studies style transfer using publicly available datasets. All models used were accessed through publicly released checkpoints or publicly accessible APIs and used in accordance with their licenses and terms of use. We do not collect human-subject data or private personal information as part of this study. The experiments are conducted to evaluate image-generation behavior and do not make claims about real individuals or events. We acknowledge the original creators and providers of these resources. The generated images are presented only as research examples, and the authors are responsible for the accuracy and context of the results reported in this paper.

REPRODUCIBILITY STATEMENT

Section 3 describes the style representation, color and luminance analysis, and generation pipeline, with pseudocode provided in Algorithm 1 in Appendix A.2. Section 4.1 specifies the dataset, numbers of evaluated styles and content prompts, generator backbones, and shared inference settings. Appendix A.1.1 describes the comparison methods, while Appendix A.1.2 reports the model configurations and training hyperparameters. Appendix A.1.3 provides the metric definitions, score aggregation procedures, and prompts used for VLM-based evaluation. The complete codebase for reproducing our experimental results will be made publicly available upon paper acceptance.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. In *arXiv:2511.21631*, 2025.

Black Forest Labs. FLUX.2 [klein]. Black Forest Labs model documentation, 2026. URL `https://docs.bfl.ai/flux_2/flux2_overview`.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Pu Cao, Feng Zhou, Qing Song, and Lu Yang. Controllable generation with text-to-image diffusion models: A survey. In *arXiv:2403.04279*, 2024.

Zhongjie Duan and Yingda Chen. Compressing image style training into a single model forward. In *arXiv:2606.13809*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the International Conference on Machine Learning*, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *Proceedings of the International Conference on Learning Representations*, 2022.

Junyao Gao, Yanchen Liu, Yanan Sun, Yinhao Tang, Yanhong Zeng, Kai Chen, and Cairong Zhao. StyleShot: A snapshot on any style. In *arXiv:2407.01414*, 2025.

Junyao Gao, Sibo Liu, Jiaxing Li, Yanan Sun, Yuanpeng Tu, Fei Shen, Weidong Zhang, Cairong Zhao, and Jun Zhang. MegaStyle: Constructing diverse and scalable style dataset via consistent text-to-image style mapping. In *arXiv:2604.08364*, 2026.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. SenseFlow: Scaling distribution matching for flow-based text-to-image distillation. In *Proceedings of the International Conference on Learning Representations*, 2026.

Feihong He, Gang Li, Fuhui Sun, Mengyuan Zhang, Lingyu Si, Xiaoyan Wang, and Li Shen. FreeStyle: Free lunch for text-guided style transfer using diffusion models. In *arXiv:2401.15636*, 2024.

Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven Hoi. D-OPSD: On-policy self-distillation for continuously tuning step-distilled diffusion models. In *arXiv:2605.05204*, 2026a.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Changsheng Lu, Zhen Li, Bo Zhang, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning. In *Proceedings of the European Conference on Computer Vision*, 2026b.

Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. GenAI Arena: An open evaluation platform for generative models. In *arXiv:2406.04485*, 2024.

Jinghong Lan, Wei Cheng, Yunuo Chen, Ziqi Ye, Peng Xing, Yixiao Fang, Rui Wang, Yufeng Yang, Xuanyang Zhang, Xianfang Zeng, Difan Zou, Gang Yu, and Chi Zhang. FreeStyle: Free control of style-content dual-reference generation from community lora mining. In *arXiv:2606.20506*, 2026.

Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive adversarial diffusion distillation. In *arXiv:2402.13929*, 2024.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Hongsheng Li, and Steven Hoi. Decoupled DMD: CFG augmentation as the spear, distribution matching as the shield. In *arXiv:2511.22677*, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. In *arXiv:2310.04378*, 2023.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. In *Proceedings of the International Conference on Learning Representations*, 2025.

OpenAI. GPT Image 2.5 Flare. OpenAI API model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-image-2.5-flare.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *Proceedings of the International Conference on Learning Representations*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *arXiv:2311.17042*, 2023.

Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, Yuan Hao, Irfan Essa, Michael Rubinstein, and Dilip Krishnan. StyleDrop: Text-to-image generation in any style. In *Advances in Neural Information Processing Systems*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning*, 2023.

Haofan Wang, Matteo Spinelli, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. InstantStyle: Free lunch towards style-preserving in text-to-image generation. In *arXiv:2404.02733*, 2024.

Zhendong Wang, Jianmin Bao, Shuyang Gu, Dong Chen, Wengang Zhou, and Houqiang Li. Design-Diffusion: High-quality text-to-design image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. In *arXiv:2209.00796*, 2022.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. In *arXiv:2511.22699*, 2025.

A Appendix

A.1 Additional Experimental Details

A.1.1 Detailed Baseline Descriptions

We provide detailed descriptions of the baselines evaluated in this paper. All baselines are evaluated with a shared protocol, including the reference image set, target content prompts, and image resolution. They differ in how the reference style enters generation, as follows:

- **Direct Style Extraction (DSE).** A vision-language model summarizes the reference images in a natural-language style description, which is combined with the target content prompt and supplied to the generator as text.
- **Reference-Guided Generation (RCG).** The generator receives the reference images through its native multi-reference interface along with a text instruction to render the target content in their shared style. The reference style is conveyed jointly by image conditioning and textual guidance.
- **Vanilla SFT.** The reference style enters through style-specific training examples, whose supervised objective updates the generator's parameters.
- **Pairwise Sample Optimization (PSO)**[2]. PSO (Miao et al., 2025) uses pairwise preferences over generated samples to indicate which outputs better match the reference style, and updates the generator accordingly.
- **D-OPSD**[3]. D-OPSD (Jiang et al., 2026a) introduces the reference style through a multimodally conditioned teacher that supervises the generator's own few-step sampling trajectories. On-policy self-distillation transfers guidance to the student while preserving few-step sampling behavior.

A.1.2 Implementation Details

All experiments were conducted on a server equipped with eight GPUs. We use the 4B[4] and 9B[5] variants of the FLUX.2-Klein text-to-image generator with their native four-step sampling procedure at a resolution of 1024×1024, using the same content prompts across methods. The visual style agent uses GPT-5.5[6] through an OpenAI-compatible API to extract reference-style information. It should be noted that StyleForge is model-agnostic and can use other compatible vision-language models. For the trainable baselines, we train a separate LoRA adapter for each target style from the reference images on both FLUX.2-Klein backbones. D-OPSD, PSO, and vanilla SFT use the same adapter configuration, with rank $r = 64$ and scaling factor $\alpha = 128$. Reference images are resized and center-cropped to 1024×1024. For each training example, the caption is randomly selected from eight English/Chinese variants, with a maximum text-sequence length of 512 tokens. We optimize the adapters with AdamW using a constant learning rate of 1×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 10^{-8}$. Each adapter is trained for 1,000 optimizer updates and the total batch size is four. For D-OPSD, the student and teacher share the same frozen backbone but maintain separate LoRA adapters. The teacher adapter is initialized from the student adapter and updated after every optimizer step with an exponential moving average (EMA) decay of 0.9999. Training follows a four-step on-policy student trajectory and minimizes the mean squared error between the student and stop-gradient teacher predictions of the clean latent at each step. Vanilla SFT minimizes the latent flow-matching velocity MSE, with one timestep sampled uniformly from the model's prescribed training timestep schedule for each example. PSO uses the same timestep sampling strategy and pairs each positive reference image with a negative image drawn from a fixed pool of 20 base-model samples per process. Positive and negative examples share the same noise, timestep, and training text condition. We use the PSO-DB objective with $\beta = 5.0$, a negative-loss factor of 0.1, and a prior-loss weight of 0.5.

[2] https://github.com/ZichenMiao/Pairwise_Sample_Optimization
[3] https://github.com/vvvvvjdy/D-OPSD
[4] https://huggingface.co/black-forest-labs/FLUX.2-klein-4B
[5] https://huggingface.co/black-forest-labs/FLUX.2-klein-9B
[6] https://platform.openai.com/docs/models/gpt-5.5

A.1.3 Evaluation Metrics

We evaluate each generated image from two complementary perspectives. The first is an overall assessment of whether the image jointly satisfies the target content prompt and transfers the reference style. These results form the Score and pairwise Win Rate reported in Table 2. The second decomposes this objective into content fidelity and style similarity for the Pareto analysis in Figure 5. Unless stated otherwise, all statistics are computed over the same set of styles and content prompts, and larger values indicate better performance. The evaluation metrics are detailed as follows:

• **Score** (Jiang et al., 2026a). We use Qwen3-VL-8B-Instruct, Qwen3-VL-32B-Instruct, and Qwen3.5-9B as independent judges. For each generated image, a judge receives the target content prompt and evaluates the generated image against each available reference image, using each reference only for style. The judge must assess the integrated style-transfer result rather than matching the reference content. It outputs one of four ordered labels, mapped to 1–4: *not consistent*, *slightly consistent*, *mostly consistent*, and *highly consistent*. The image score is the mean of the valid per-reference scores, and the reported value is the mean over all generated images for each method and backbone. The full prompt for evaluation is given below.

Score Prompt

```
[System Prompt]
You are a strict visual style-transfer evaluator for text-guided image generation.
You will receive:
  - A target content prompt describing what the candidate image should depict.
  - A first image that is a reference for visual style only.
  - A second image that is the generated candidate image.
Judge the candidate as a complete style-transfer result. The candidate must simulta-
neously satisfy the target content prompt and express the reference image's visual
style. The reference image content is not the target content: do not require the
candidate to copy the reference subject, objects, scene, layout, or composition.
Use the first image only to judge style.
Content criterion:
  - Check whether the candidate visibly depicts the requested subject, scene, action,
    objects, scale, and viewpoint from the target content prompt.
  - A generic, missing, wrong, or materially changed main subject should lower the
    judgment even if the style looks plausible.
Style criterion:
  - Check whether the candidate transfers the reference's visual style through
    medium, rendering method, palette behavior, lighting treatment, texture, line
    and edge quality, abstraction level, graphic effects, and atmosphere.
  - Do not give high credit for generic style-category words alone when the concrete
    visual style cues are weak or absent.
Joint criterion:
  - Give a high label only when the requested content and the reference style are
    both present and naturally integrated in the same image.
  - Judge the overall semantic success directly. Do not compute a mechanical minimum,
    average, or checklist score between content and style.
Scoring labels:
  - highly consistent: the candidate clearly depicts the target content and strongly
    transfers the reference visual style as an integrated result.
  - mostly consistent: the candidate depicts the main target content and transfers
    most important style cues, with only minor omissions or drift.
  - slightly consistent: the candidate has partial target content or partial style
    transfer, but one side is weak, incomplete, generic, or poorly integrated.
  - not consistent: the candidate misses the target content, misses the reference
    style, or looks like a different/generic result.
Output exactly one label from this list:
  - highly consistent, mostly consistent, slightly consistent, not consistent.
[User Prompt]
```

[Reference Image] [Generated Image] [Content Prompt]

The first image is a style reference only. The second image is the generated candidate. Judge whether the second image both depicts the target content prompt and transfers the first image's visual style as one integrated result. Output one label only. Output exactly one digit and nothing else: 4 = highly consistent, 3 = mostly consistent, 2 = slightly consistent, 1 = not consistent.

• **Pairwise Win Rate** (Jiang et al., 2024). The same VLM judges compare two candidates generated for the same target prompt. Each comparison provides a style-reference board, Candidate A, and Candidate B, and asks the judge to select the better complete style-transfer result by jointly considering content correctness and reference-style fidelity. Every unordered method pair is evaluated in both candidate orders. A win, tie, and loss contribute 1, 0.5, and 0, respectively. We first average the ordered comparisons for each method–opponent matchup and then average equally over all other methods. The resulting average Pairwise Win Rate is reported as a percentage. The comparison prompt is shown below.

Pairwise Preference Prompt

[System Prompt]

You are a strict pairwise evaluator of text-guided visual style transfer.

You will receive:
- a target content prompt;
- a style-reference board containing examples of the target visual style;
- Candidate A and Candidate B, which were generated for the same target.

Choose which candidate is the better complete style-transfer result. Judge the two candidates jointly on both criteria below.

Content correctness:
- The candidate should visibly depict the requested subject, scene, action, objects, scale, and viewpoint.
- Missing or materially changing an important requested element is a serious failure even when the candidate looks attractive.

Reference-style fidelity:
- The candidate should transfer the reference examples' visual style through medium and rendering method, palette behavior, lighting, texture, line and edge quality, abstraction level, geometry, graphic effects, and atmosphere.
- The reference-board content is not the requested content. Do not reward a candidate for copying the reference subjects, objects, scene, layout, or composition.
- Do not reward a generic match to a broad style name when the concrete visual mechanisms shown in the references are weak or absent.

Prefer the candidate in which correct target content and faithful reference style are more successfully integrated. Do not assign independent numerical scores. Use a tie only when neither candidate is clearly better overall, such as an effectively indistinguishable result or a genuine balanced tradeoff.

The final answer must be exactly one uppercase letter and nothing else:
- A = Candidate A is better
- B = Candidate B is better
- C = tie

[User Prompt]

[Style Reference Board] [Candidate A Image] [Candidate B Image] [Content Prompt]

The first image is the STYLE REFERENCE BOARD only; its depicted content is not the target. The second image is CANDIDATE A. The third image is CANDIDATE B. Select the better joint content-and-style result. Output exactly A, B, or C.

• **CLIP-Text and CLIP-Style** (Radford et al., 2021). CLIP-Text measures alignment between a generated image x and its target content prompt c, whereas CLIP-Style measures alignment between x and the canonical style prompt s provided by MegaStyle-1.4M. It should be noted that CLIP-Text

and CLIP-Style use a CLIP ViT-H/14[7] encoder. For an image x and a text prompt t, the CLIP score is defined as

$$\mathrm{CLIP}(x,t) = \exp(\ell)\,\frac{f_I(x)^\top f_T(t)}{\|f_I(x)\|_2\,\|f_T(t)\|_2},$$

where f_I and f_T are the image and text embeddings and ℓ is the learned logit scale. We average each score over all generated images for a method and backbone. Higher CLIP-Text and CLIP-Style indicate stronger content and style alignment, respectively.

• **VLM-F and VLM-S** (Lan et al., 2026). VLM-F (instruction following) scores how well a generated image satisfies the target content prompt on a 0–10 scale. The evaluator checks the requested subjects, attributes, actions, and spatial constraints. VLM-S (style similarity) compares the generated image with the first style-reference image on a 0–10 integer scale, considering brushstroke and edge behavior, texture, color system and distribution, shape language, and recurring patterns. For each method and backbone, we first average scores within each style and then average the per-style means, yielding VLM-F Macro and VLM-S Macro. VLM-F allows scores in increments of 0.5, whereas VLM-S requires integer scores.

VLM-F Prompt

```
[User Prompt]
Final image: [Generated Image] Instruction: [Target Content Prompt]
You will be given:
  1) An image (the final generated result).
  2) An instruction (the target content prompt).
Task:
Score how well the final image follows the instruction, from 0 to 10. Give PARTIAL
CREDIT whenever the image implements some parts correctly. Do NOT assume anything
that is not clearly visible in the image.
Important evaluation rule:
  - Ignore style/aesthetic differences (rendering style, brushwork, resolution,
    noise, overall beauty) ONLY IF the instruction does NOT explicitly ask for
    them.
  - If the instruction explicitly mentions color, lighting, mood, style, lens,
    texture, etc., then you MUST evaluate those aspects (do NOT ignore them).
How to score (internal steps; do not output these steps):
  1) Break the instruction into atomic, checkable requirements:
      - main subject(s) / objects
      - key attributes (color, material, count, size, identity if specified)
      - actions / interactions
      - spatial/layout constraints (left/right, background, position)
      - any required text/logos (exact wording matters)
     Mark each requirement as MUST-HAVE or NICE-TO-HAVE.
  2) For each requirement, judge: satisfied / partially satisfied / not satisfied
     / contradicted.
  3) Combine into a final score using this guidance:
      - Start from 10 and subtract penalties:
         * Missing a MUST-HAVE requirement: -2 to -4 (bigger miss = bigger penalty)
         * Partially meeting a MUST-HAVE: -1 to -2
         * Minor detail wrong (NICE-TO-HAVE): -0.5 to -1
         * Direct contradiction (e.g., asked to add X but it removes X): -3 to -5
         * Wrong or missing required text/logo: -3 to -6
     Clamp the result to [0, 10].
Score anchors (use these to avoid extreme scores):
  - 9-10: All MUST-HAVEs satisfied; at most tiny minor issues. Use 10 ONLY if
    nothing important is missing or wrong.
  - 7-8.5: Most MUST-HAVEs satisfied; 1-2 notable misses or several small issues.
  - 4-6.5: Some key parts are implemented, but major omissions/misapplied edits
    exist.
  - 1-3.5: Slight relation, but most MUST-HAVEs are missing/wrong.
```

[7] https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K

```
  - 0–0.5: Completely unrelated or clearly opposite to the instruction. Use 0 ONLY
    when it is truly a total failure.
Output rules (very important):
  * Output ONLY one line in the format: score@reason
  * score must be a number from 0 to 10, allowing .5 steps (e.g., 7.5). Do NOT
    always use 0 or 10.
  * reason must be 1–2 short sentences, specific and directly observable (mention
    1–2 key satisfied/missed requirements).
```

VLM-S Prompt

[User Prompt]

```
Image A: [Reference Image] Image B: [Generated Image]
You will see two images: A and B. Please assess their consistency solely in terms
of visual/artistic style. Do not judge based on whether the depicted subjects are
the same (people/objects/scenes may be completely different). Assign an overall
similarity score from 0 to 10 by considering all dimensions together.
Dimension definitions (judge strictly according to the following meanings):
  1) BRUSHSTROKE (brushstrokes/lines/edge treatment)
      - Line thickness, whether lines are shaky, outlining methods, and how visible
        brush marks are.
      - Whether edges are hard or soft, whether outlines are present, and stroke
        characteristics such as smearing/splattering/pencil/pen marks.
  2) TEXTURE (texture/material rendering/grain and noise/canvas and paper texture)
      - Microscopic surface grain: e.g., "sandpaper grain/film noise/watercolor
        paper fibers/oil-painting canvas weave".
      - How fine material structures are rendered: e.g., "built-up paint tex-
        ture/spray dots/halftone dots/relief texture".
      - Note: TEXTURE is neither color palette nor shape contours.
  3) COLOR (color palette and distribution/color temperature/saturation/contrast).
      - Dominant colors and color-temperature tendencies (cool/warm), saturation
        levels, and the strength of light-dark contrast.
      - Similarity in color distribution: e.g., "whether the proportions/coverage of
        major colors are similar (large background regions, proportions of dominant
        color blocks)".
      - Note: COLOR is neither texture grain nor brushstroke lines.
  4) SHAPE (shape language/characteristic forms).
      - The degree of geometric or realistic stylization, whether proportions are
        exaggerated, and whether contours are sharper or more rounded.
      - How structures are simplified (e.g., flat cartoons, chibi, minimalist ge-
        ometry, realistic structures).
  5) PATTERN (patterns/motifs/decorative patterning).
      - Whether there are repeated patterns, decorative elements, or recurring
        symbolic motifs (e.g., fixed patterns, recurring decorative lines, patterned
        backgrounds).
      - Whether pattern density, repetition rules, and the arrangement of decorative
        elements are similar.
Scoring criteria (0–10; score strictly):
  * 0: The styles are completely inconsistent, with virtually no basis for consid-
    ering them the same along this dimension.
  * 1–3: Mostly inconsistent, with only very weak/incidental similarities.
  * 4–6: Some similarity, but clear omissions or key differences remain.
  * 7–9: Largely consistent, with only a few minor issues.
  * 10: Fully consistent, with a high degree of agreement in the key characteristics.
Important notes (must be followed):
  * Score strictly. Do not give high scores unless the images clearly and accurately
    match along the dimension being assessed.
  * Only assess consistency in style dimensions; do not consider whether the people
    have the same identity, whether the objects are the same, or whether the
    backgrounds/scenes match.
```

```
    * Do not judge aesthetic quality or whether the image looks attractive.
  Output rules (very important):
    * score must be an integer from 0 to 10.
    * reason must be 1-2 sentences stating specific, observable evidence.
    * Output strictly in the format score@reason (output only this single line; do
      not output any additional characters/punctuation/line breaks).
```

A.2 Algorithm Overview

Algorithm 1 details the StyleForge pipeline for constructing reusable style instructions. Stages 1–4 expand the state integration in Eq. 1, grounding color allocations in reference statistics, abstracting their carriers, and assigning transfer policies. Stage 5 compiles the state into a style prompt and checks its coverage and exclusion of reference content. These stages are executed once per reference set. Stage 6 reuses the validated prompt across content prompts while keeping the parameters and native few-step sampling procedure of the step-distilled T2I model fixed. This preserves the model's native few-step generation capability.

Algorithm 1 StyleForge: Style Transfer with Reusable Rendering Instructions

Require: Reference image set $\mathcal{R}$; N target content prompts $\{c_j\}_{j=1}^{N}$; multimodal parser Φ with parsing protocol q; color/luminance analyzer H; prompt compiler C with order $\pi = (\text{rendering}, \text{scope}, \text{color})$; frozen step-distilled generator G

Ensure: Reusable style prompt $p_{\mathcal{R}}$ and stylized image x_j for each content prompt c_j

// Stage 1: Structured reference parsing
// $M_{\mathcal{R}}$: rendering practices; $A_{\mathcal{R}}$: color/light allocation records
// $T_{\mathcal{R}}$: compositional tendencies; $L_{\mathcal{R}}$: reference-content records
$E_{\mathcal{R}} \leftarrow \Phi(\mathcal{R}; q)$ ▷ *structured style evidence*
assert $E_{\mathcal{R}}$ contains all required fields and color–carrier pairs
$(M_{\mathcal{R}}, A_{\mathcal{R}}, T_{\mathcal{R}}, L_{\mathcal{R}}) \leftarrow E_{\mathcal{R}}$
assert $M_{\mathcal{R}}$ excludes reference subjects, props, and scenes
// Stage 2: Color and luminance grounding
$D_{\mathcal{R}} \leftarrow H(\mathcal{R})$ ▷ *reference color and luminance statistics*
Refine color anchors and coverage in $A_{\mathcal{R}}$ using $D_{\mathcal{R}}$
// Stage 3: Transferable color carriers
// ρ: visual role and scale; h: hex color anchor; κ: carrier surface
// s: spatial extent; g: spatial distribution; ℓ: color/light attribute type
for $i = 1, \ldots, |A_{\mathcal{R}}|$ **do** ▷ *$|A_{\mathcal{R}}|$: number of color records*
  $(\rho, h, \kappa, s, g, \ell) \leftarrow A_{\mathcal{R}}[i]$ ▷ *the i-th color allocation*
  if κ is tied to reference content **then**
    $L_{\mathcal{R}} \leftarrow L_{\mathcal{R}} \cup \{\kappa\}$ ▷ *retain the original carrier for auditing*
    $\kappa' \leftarrow$ transferable surface description of κ
    $A_{\mathcal{R}}[i] \leftarrow (\rho, h, \kappa', s, g, \ell)$ ▷ *preserve color, role, and extent*
  end if
end for
// Stage 4: Style-state assembly
Preserve $M_{\mathcal{R}}, A_{\mathcal{R}}$; retain $T_{\mathcal{R}}$ only as weak spatial tendencies
Keep $L_{\mathcal{R}}$ solely for reference-content exclusion checks
$S_{\mathcal{R}} \leftarrow (M_{\mathcal{R}}, A_{\mathcal{R}}, T_{\mathcal{R}}, L_{\mathcal{R}})$ ▷ *integrated style state*
// Stage 5: Prompt compilation and validation
$p_{\mathcal{R}} \leftarrow C(S_{\mathcal{R}}; \pi)$ ▷ *rendering → scope → color*
assert $p_{\mathcal{R}}$ covers all rendering and color records in $M_{\mathcal{R}}, A_{\mathcal{R}}$
assert $p_{\mathcal{R}}$ contains no reference content recorded in $L_{\mathcal{R}}$
// Stage 6: Reusable few-step generation
for $j = 1, \ldots, N$ **do**
  $x_j \leftarrow G(p_{\mathcal{R}}\,; c_j)$ ▷ *concatenate $p_{\mathcal{R}}$ and c_j; native sampler*
end for
return $p_{\mathcal{R}}$, $\{x_j\}_{j=1}^{N}$

A.3 Further Analysis

A.3.1 Additional Qualitative Results on the FLUX.2-Klein Series

Figures 6 and 7 supplement the qualitative comparisons in Section 4.2.1 by showing the same style–content pairs on the other FLUX.2-Klein backbone. Specifically, Figure 6 revisits the examples in Figure 4 using the 4B model, while Figure 7 revisits those in Figure 3 using the 9B model. These paired comparisons help assess whether the observed behavior persists across model sizes.

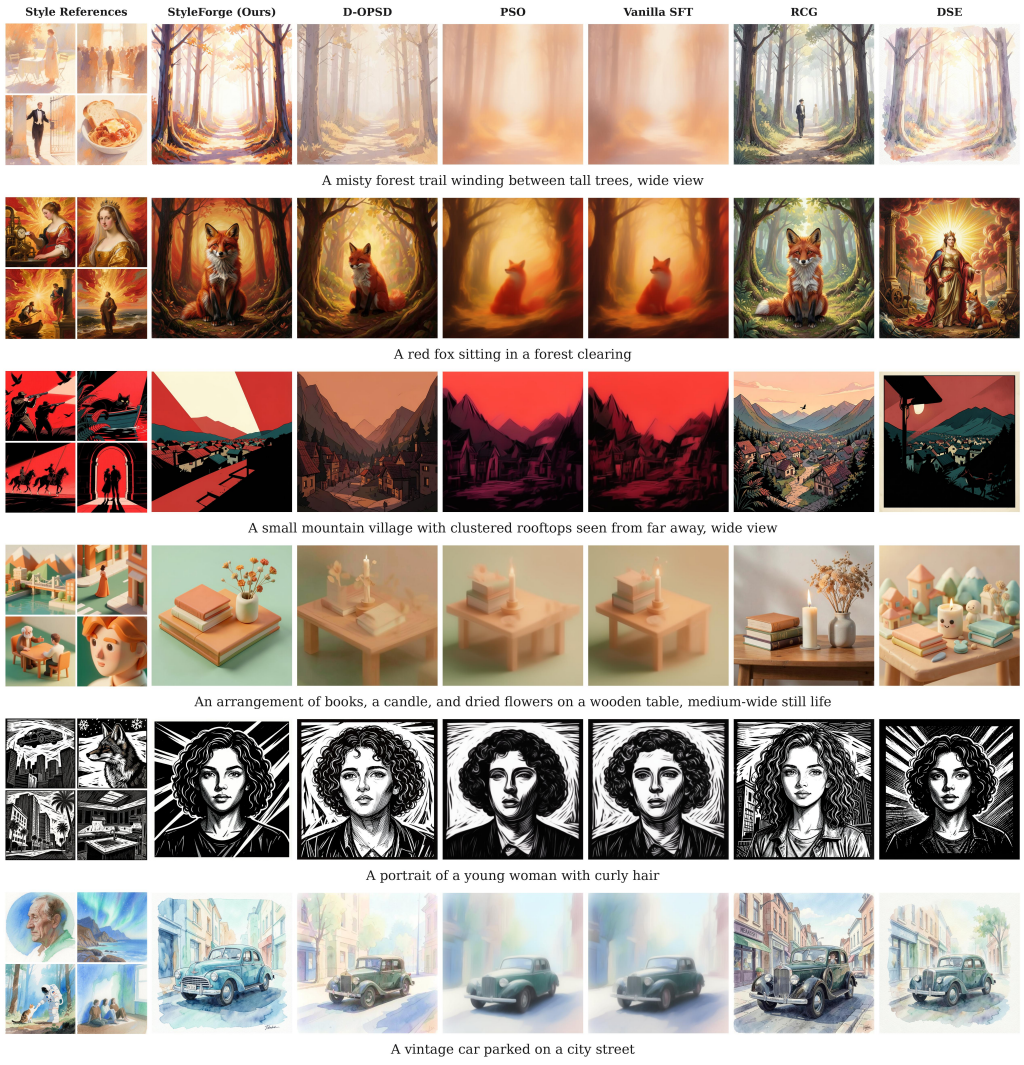

Figure 6: Additional qualitative comparison with baselines on FLUX.2-Klein-4B.

StyleForge retains distinctive rendering and color characteristics across these additional cases. In Figure 6, the fox (row 2, column 2) combines the warm red-and-gold palette of the references with strong light–dark contrast, while the vintage car (row 6, column 2) is rendered with translucent blue washes and a paper-like texture. In Figure 7, the city avenue (row 5, column 2) carries over the saturated blue–orange palette and pronounced brushwork of the references. The fishing harbor (row 6, column 2) instead adopts flat teal and yellow-orange shapes characteristic of the reference posters. These examples illustrate how reference-style characteristics can be expressed through different requested subjects and scenes on both backbones.

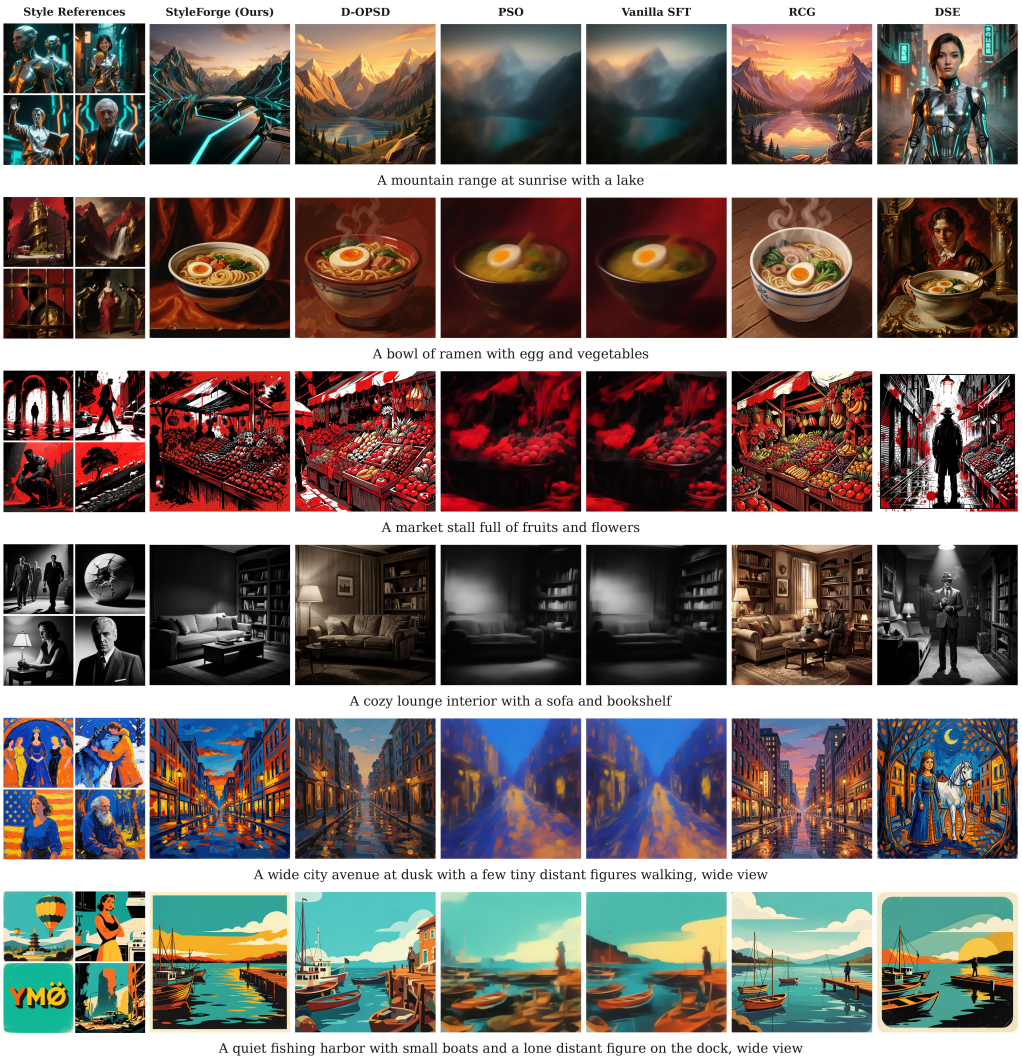

Figure 7: Additional qualitative comparison with baselines on FLUX.2-Klein-9B.

The additional comparisons also show the baseline failure modes discussed in Section 4.2.1. DSE introduces reference-associated figures into unrelated prompts: it replaces the fox with a crowned human figure in Figure 6 (row 2, column 7) and adds a standing man to the lounge interior in Figure 7 (row 4, column 7). RCG preserves the lounge content but renders it in warm colors (row 4, column 6), whereas StyleForge retains the monochrome lighting of the film-noir references. PSO and Vanilla SFT produce blurred structures in the forest trail of Figure 6 (row 1, columns 4–5) and the city avenue of Figure 7 (row 5, columns 4–5). Together, these observations reinforce the finding that StyleForge better balances reference-style fidelity, content adherence, and visual clarity under native few-step generation on both FLUX.2-Klein backbones.

A.3.2 Comparison with Image-to-LoRA Style Transfer

To complement the comparisons in Section 4.2.1, Figure 8 presents a qualitative comparison with i2L (Duan & Chen, 2026), a feed-forward image-to-LoRA style-transfer method that predicts LoRA adapter weights directly from reference images. We use its released KleinBase4B-i2L-v2[8] model to examine how StyleForge's outputs compare with this distinct reference-to-adapter paradigm. The two methods use different generator checkpoints within the FLUX.2-Klein-4B series. The i2L is

[8] https://modelscope.cn/models/DiffSynth-Studio/KleinBase4B-i2L-v2

designed for the base model and is evaluated here with classifier-free guidance (CFG) scales of 1 and 4. StyleForge uses the step-distilled model with native few-step sampling, which does not support adjustable CFG. Since both the generator checkpoint and inference configuration differ, these results provide a supplementary qualitative comparison rather than a controlled comparison under identical generation conditions.

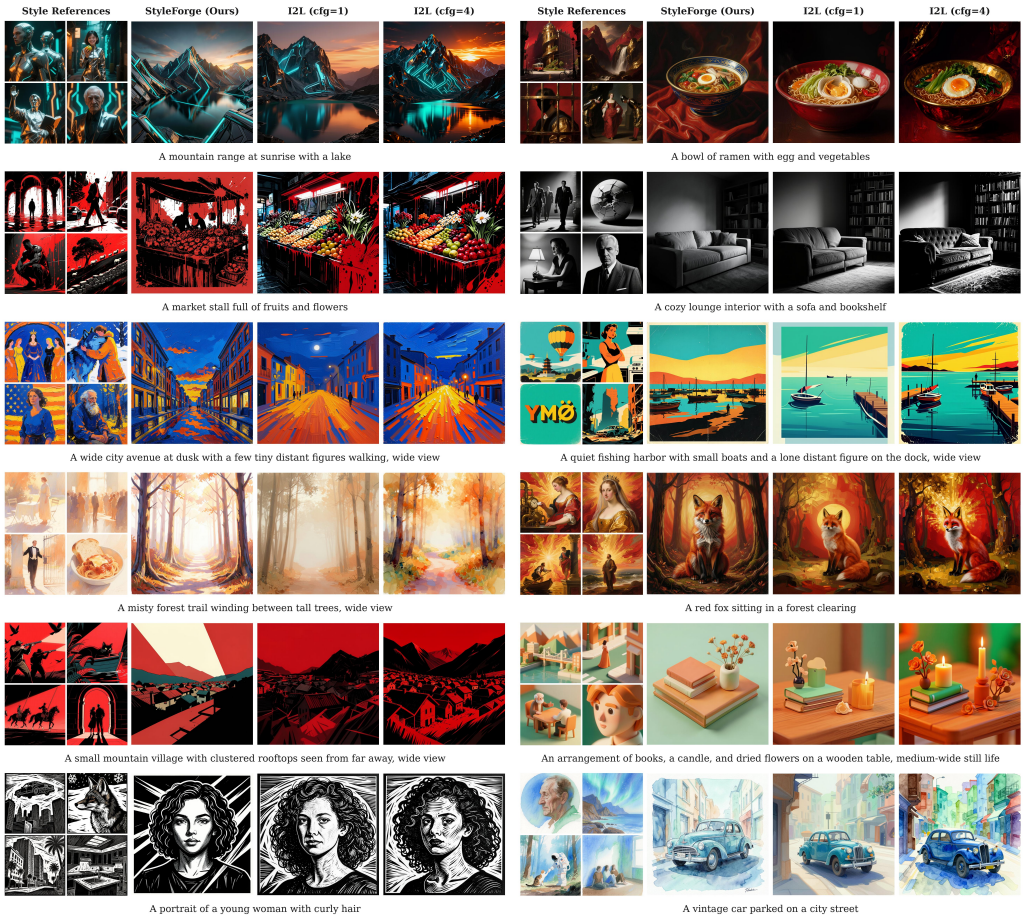

Figure 8: Supplementary qualitative comparison with i2L on the FLUX.2-Klein-4B series. Each group shows style references, StyleForge using the step-distilled model, and i2L using the base model with CFG scales of 1 and 4.

As shown in Figure 8, StyleForge is visually competitive with i2L across diverse styles and performs favorably on several style–content pairs. Both methods capture the cyan-lit metallic appearance of the mountain scene (row 1, left group), the monochrome lighting of the lounge (row 2, right group), and the carved black-and-white marks of the portrait (row 6, left group). In some cases, StyleForge more closely preserves specific reference characteristics. For the market scene (row 2, left group), it retains the restricted red-and-black palette, while both i2L configurations introduce multicolored fruit and flowers. For the vintage car (row 6, right group), StyleForge preserves pale watercolor washes and soft transitions, while i2L at CFG=4 produces more saturated colors and stronger outlines. These examples illustrate that StyleForge can achieve competitive reference-style fidelity through textual guidance while retaining few-step generation. More generally, the side-by-side examples provide a convenient visual check of palette, illumination, surface treatment, and edge rendering across different content prompts. By placing each reference beside outputs for the same content prompt, the comparison exposes whether a method transfers these reference-specific properties while preserving the requested content, including distinctions that aggregate quantitative scores may obscure.

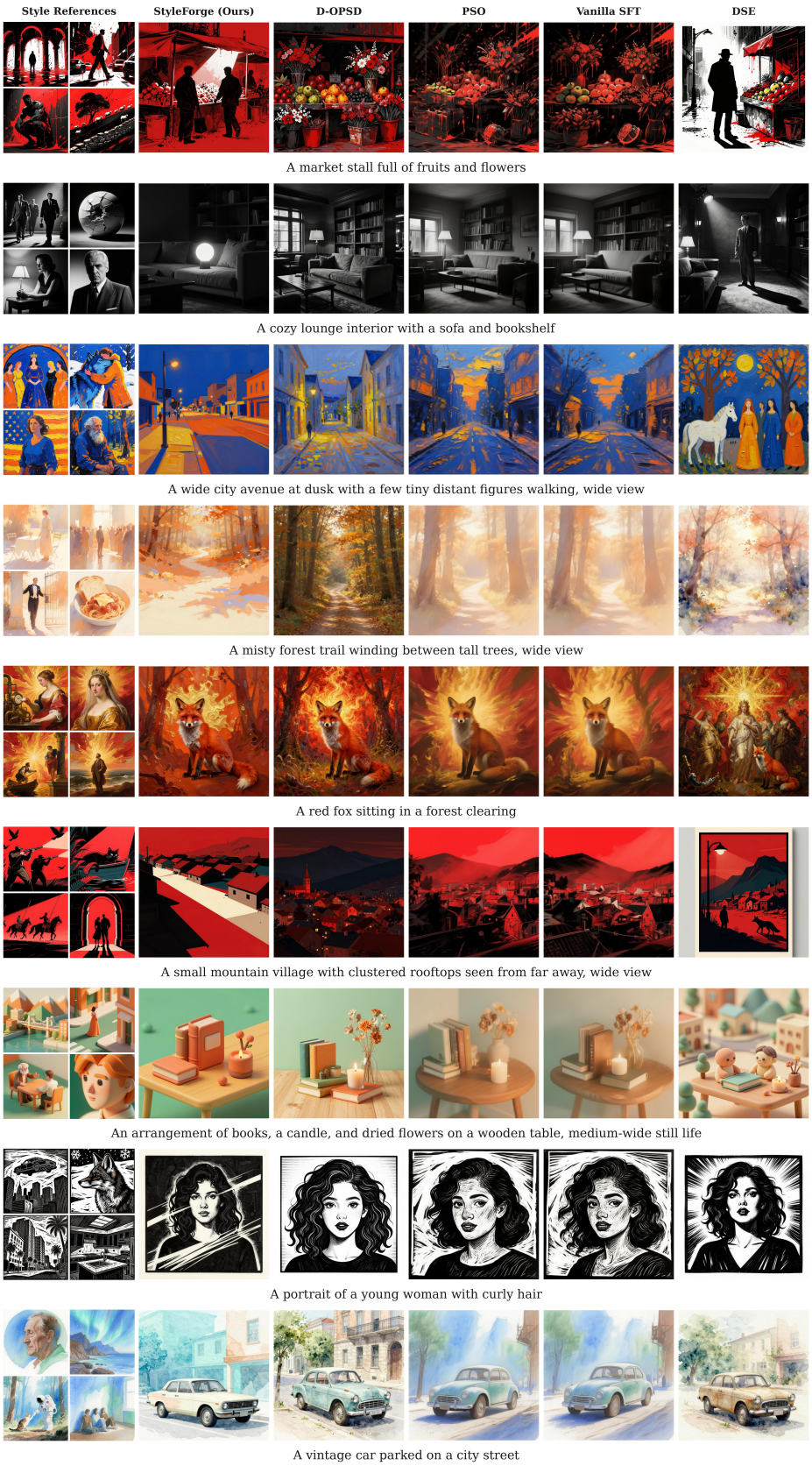

Figure 9: Qualitative comparison with baselines on Z-Image-Turbo 6B.



Figure 10: Qualitative style-transfer results with GPT-Image-2.5-Flare. The first column shows style references, and the remaining columns show generated images for different content prompts.

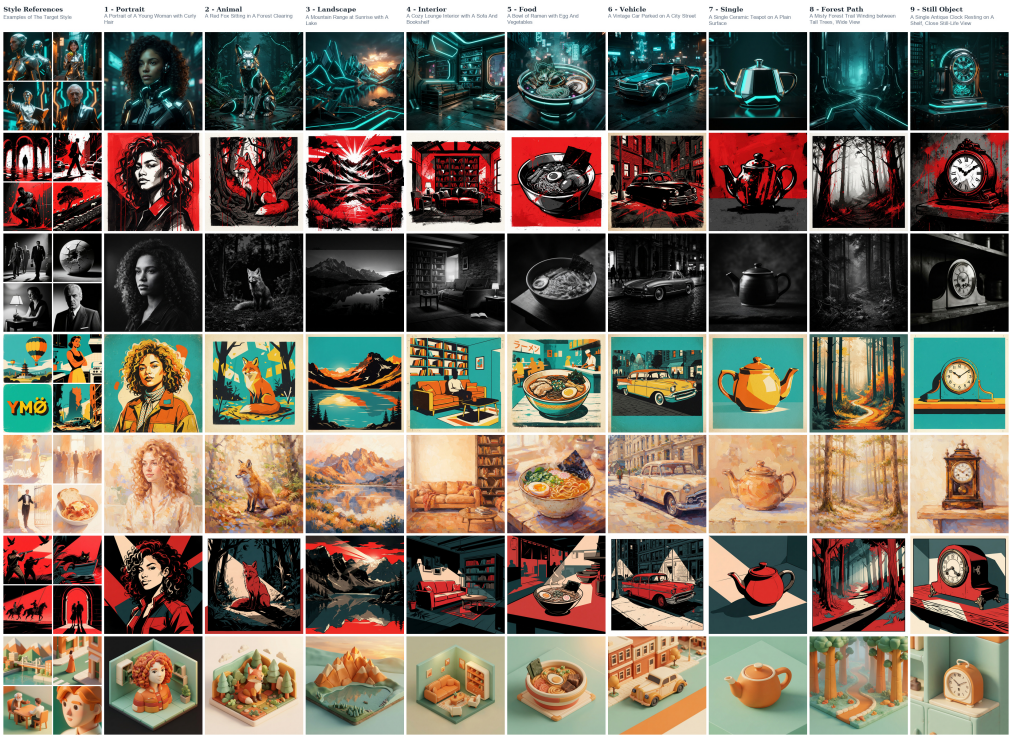

Figure 11: Qualitative style-transfer results with Nano-Banana-2-Lite. The first column shows style references, and the remaining columns show generated images for different content prompts.

A.3.3 Generalization to Other Efficient Image Generators

To examine whether StyleForge's extracted style representation transfers beyond the FLUX.2-Klein family, we first reuse the same rendering instructions with Z-Image-Turbo 6B[9], a few-step distilled generator. Z-Image-Turbo does not support reference-image conditioning, so RCG is not applicable and is omitted from Figure 9. StyleForge is therefore evaluated through its extracted textual instructions. The results preserve recognizable palette structure, contrast, and lighting cues across the requested subjects, indicating that the extracted description remains effective when the generator changes. Z-Image-Turbo appears somewhat less consistent in following explicit hexadecimal color specifications than the FLUX.2-Klein series, so its overall color control is slightly less precise. This difference is best understood as a property of the generator's conditioning behavior and does not prevent the broader rendering style from transferring. We then apply the same instructions to the efficient closed-source systems GPT-Image-2.5-Flare[10] and Nano-Banana-2-Lite[11]. Figure 10 shows that GPT-Image-2.5-Flare follows the extracted style cues across portraits, objects, and scenes while adapting them to different content prompts. The results in Figure 11 provide a similar observation for Nano-Banana-2-Lite. Consistent style transfer on a distilled generator and two efficient closed-source systems suggests that StyleForge captures reusable textual style descriptions rather than generator-specific visual features.



Figure 12: Qualitative illustration of subject driven generation with a step-distilled model. Three reference subjects, a can, a toy car, and a berry bowl, are placed in a new cinematic scene. The reference-conditioned result preserves their colors, graphics, shapes, and lettering, while the text-only sample uses the same prompt and seed without access to the reference images.

A.3.4 Discussion of the Subject Driven Generation Task

Prior work such as D-OPSD (Jiang et al., 2026a) studies subject driven generation together with style transfer. In this work, we focus on transferring visual style to new content and treat subject driven generation as a related but separate setting. This scope is motivated by the capabilities of modern step-distilled T2I models: subject driven generation is already a strong native capability. Given a small set of reference images, these models can preserve the identities and fine visual details of multiple subjects while composing them in a new scene. We include the following example to illustrate this native capability and to clarify the distinction between these two uses of reference images. Figure 12 gives a qualitative illustration. We provide three reference images, showing a

[9] https://huggingface.co/Tongyi-MAI/Z-Image-Turbo
[10] https://developers.openai.com/api/docs/models/gpt-image-2.5-flare
[11] https://ai.google.dev/gemini-api/docs/image-generation

can, a toy car, and a berry bowl, and ask the model to place all three subjects in a new cinematic mountain-road scene. With the references, the generated image retains the identities and salient details of all three objects: the can's blue color, graphics, and lettering, the toy car's red-and-green body, driver, and antenna, and the bowl's shape, grid pattern, and handwritten text. The generated scene also composes these objects into the requested interaction, including the collision, trailer, and flying berries. For comparison, the text-only sample produced with the same prompt and seed contains different instances of the objects and does not preserve their reference-specific appearance. The example shows how reference images can preserve subject identity and fine details, while our main setting uses references to transfer rendering behavior across different content.

A.3.5 Case Study

This section presents selected case studies of the StyleForge extraction process. For each selected style, we show the reference images together with the final style prompt produced by the visual style agent. These examples illustrate the connection between visual evidence and the reusable textual rendering instructions without listing all extracted style prompts.

Case Study: Cyberpunk

[Style References]



[Style Prompt]

```
Render the requested content in glossy cyberpunk cinematic realism: recognizable
forms rebuilt with polished chrome or black lacquer surfaces, angular segmented
seams, crisp beveled edges, deep teal-black shadow fields, shaped cyan-teal neon
strips; Visual distribution lock: cyan hue families occupy broad visible carrier
regions; large planes become high-chroma carriers, not small accents or white wash;
Dominant chromatic field: cyan colored light islands stay visibly chromatic; high-
value color stays chromatic; Scene carrier map: cyan distribution maps to broad
scene/material planes; Medium construction: foreground objects/background expose
units; Geometry grammar: one grammar across objects/background; Mechanism transfer:
Preserve requested object identities/actions through extracted; Material process
evidence: object/background process evidence; Evidence carriers: process cues on
object contours, internal seams; Apply extracted shape/proportion and edge rules
across subjects, objects, and background; Color carriers: role discipline: colored
material/light anchors stay carrier-bound before shadow material/base; no generic
object color or whole-image tint; primary material/color anchors: cyan teal neon
bands: radiating material bands (#00E1D0/#20F7E3), mirror chrome silver: material
surfaces, object (#C7D8D5/#E1EFEA); secondary light/highlight anchors: warm amber
rim reflection (#FF8A25/#F06A16); shadow material anchors: gloss black lacquer
(#050807/#111A17), support chromatic carriers without whole-scene fill; carrier
arbitration: selected/accent/local roles stay on small surfaces, bands, patches;
exclude whole subject body, whole clothing mass, whole tabletop, whole background;
do not recolor entire semantic objects; Color/value distribution: preserve ma-
jor dark/black fields; keep selective light islands readable with visible area;
keep cyan chromatic material fields as repeated carrier-area coverage, not tiny ac-
cents/brown/gray/white wash; dominant saturated fields need broad repeated coverage
on compatible patches/fields before shadows/highlights; selected/accent roles may
repeat on patches/bands, not whole objects; preserve dark-light separation.
```

Case Study: Folk Art

[Style References]



[Style Prompt]

Render the requested content as saturated impasto folk art: broad cobalt-blue, hot-orange, and warm-yellow opaque paint fields; Visual distribution lock: blue and orange-red hue families occupy broad visible carrier regions; large planes become high-chroma; Dominant chromatic field: blue and orange-red colored light islands stay visibly chromatic; high-value color stays chromatic; Scene carrier map: dominant hue distribution maps to broad scene/material planes; Medium construction: foreground objects/background expose units; Geometry grammar: one grammar across objects/background; Mechanism transfer: Preserve requested object identities/actions through extracted; Material process evidence: object/background process evidence; Evidence carriers: process cues on object contours, internal seams; Color carriers: role discipline: keep extracted scales; bind roles only to extracted carriers; material/light/local anchors precede dark/base fields; no generic object color or whole-image tint; primary material/color anchors: warm yellow slab light: bands, patches [broad field], hot orange raised paint: bands, patches [selected carrier only] (#FFD92F/#F6C51E, #F56A18/#E65313); carrier arbitration: selected/accent/local roles allowed only on small surfaces, bands, inserts, facets, patches; exclude whole subject body, whole clothing mass, whole tabletop, whole background; do not recolor entire semantic objects; Color/value distribution: preserve dark/black structure; keep selective light islands readable with visible area; keep blue and orange-red chromatic material fields as repeated carrier-area coverage, not tiny accents/brown/gray/white wash; dominant saturated fields need broad repeated coverage on compatible patches/fields before shadows/highlights; selected/accent roles may repeat on patches/bands, not whole objects; preserve dark-light separation; preserve wide value range with dark anchors and bright islands; base fields: deep cobalt flat field [broad field] (#0646B8/#07358F).

A.3.6 Additional Pareto Analyses

To complement the Qwen3-VL-32B-Instruct results in Figure 5, this section reports the corresponding Pareto analyses obtained with Qwen3-VL-8B-Instruct and Qwen3.5-9B. The plots use the same content-fidelity and style-similarity metrics and the same FLUX.2-Klein settings, allowing the position of each method to be compared across the three language-model judges. Consistent with the observation in Figure 5, StyleForge lies near the knee of the Pareto frontier for both backbones, indicating a favorable balance between content fidelity and style similarity.

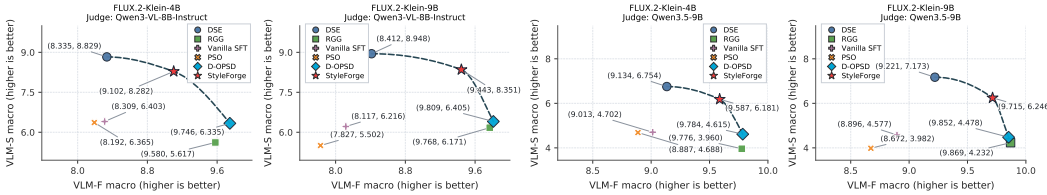

Figure 13: Additional Pareto comparisons using Qwen3-VL-8B-Instruct and Qwen3.5-9B as judges. The two left panels show the FLUX.2-Klein-4B and 9B results for Qwen3-VL-8B-Instruct, while the two right panels show the corresponding results for Qwen3.5-9B.

A.3.7 STRUCTURED STYLE EXTRACTION PROMPT

This section presents the structured prompt used by StyleForge for the VLM-facing extraction stage, which converts multiple reference images into an initial reusable style descriptor. The prompt directs the vision-language model to identify only shared, content-agnostic rendering traits, organize them by the enabled style axes, and return a strict JSON descriptor together with a generator-facing style fragment. Reference-specific subjects, objects, and one-off scene details are explicitly excluded so that the extracted style can be applied to new content. The returned descriptor is subsequently integrated with color and luminance statistics and mapped to the rendering, color-and-light, composition, and reference-content components of the StyleForge state.

StyleForge Structured Style Extraction Prompt

[System Prompt]

You extract only shared style traits across all reference images. Ignore subject identity and one-off content. Return strict JSON.

[Instruction]

Summarize the intersection of the reference image style across these axes:

- color_palette
- composition
- brushwork
- lighting
- texture
- subject_treatment
- mood

Return JSON with `global_summary`, `per_axis`, and `rendered_prompt_fragment`.

[Required JSON Schema]

```
{
    "global_summary": "style-only summary shared by all references",
    "per_axis": {"axis_name": "axis-specific style description for
    every enabled axis"},
    "rendered_prompt_fragment": "the final, vivid, generator-facing
    style description"
}
```

[Rendered Prompt Fragment Specification]

Write `rendered_prompt_fragment` as one vivid, concrete English style paragraph (about 110 to 180 words) that a text-to-image model can follow with no reference image. Derive every claim from repeated visual relations across the reference set. Do not name or describe any concrete subject, object, character, scene, logo, or identity from the reference images; describe only content-adaptive style rules that generalize to any future content. Do not force the style into a canned family; only name a medium or genre when it is visually necessary and supported by the references. Cover these in order:

1) **STYLE IDENTITY**: give a compact style identity based on the shared rendering behavior, not on reference subject matter or a memorized preset label.
2) **PALETTE RELATIONS**: describe the repeated color system with specific color names, saturation, value range, dominant/accent/neutral roles, and how color roles behave across highlights, shadows, edges, backgrounds, and focal forms. State opponent, analogous, monochrome, pastel, low-key, high-key, multi-hue, or sparse-accent relations only when shown by the references. Do not reduce the style to a flat list of colors or impose an unsupported warm/cool relation.
3) **SIGNATURE STRUCTURE**: identify repeated structural rules: edge hardness, line presence or absence, shape simplification, region size, pattern repetition, texture frequency, surface finish, artifacts, grain, blockiness, contour behavior, spatial layering, and other visible marks. Name observed mechanisms only.
4) **LIGHTING AND VALUE ROLES**: describe the organization of light, shadow, highlight, rim/edge light, ambient fill, contrast, bloom/haze, and black/white levels. Tie color to lighting roles when the references do so, while separating style lighting from local object color.

Continuation: StyleForge Structured Style Extraction Prompt

5) **RENDERING AND FORM**: describe dimensionality, shading model, material abstraction, texture/grain, edge treatment, detail density, proportions, and how forms remain readable across new content.
6) **MOOD AND DESIGN DISCIPLINE**: describe shared emotional tone and restraint/energy level as visual rules, not story content.
7) **AVOID CLAUSE**: end with a short "Avoid: ..." listing only visual looks that contradict the observed reference relations or common base-model drifts. Keep it style-only and content-agnostic.

Be concrete and evocative; avoid vague analytical jargon. Use no bullet lists and no JSON words inside `rendered_prompt_fragment`. Do not use raw hex colors, numeric scores, or reference-image subject nouns. In `per_axis`, give `color_palette`, `lighting`, `texture`, and `brushwork` the same reference-derived relation treatment.