Toward Semantically-Consistent Tuning-Free Customization for Rectified Flow Transformers

Jian Jin[1] **Kai Zhang**[2] **Zhenyong Fu**[1†] **Jian Yang**[1,2†]
[1]PCA Lab, Nanjing University of Science and Technology, [2]PCA Lab, Nanjing University
{jinj, z.fu, csjyang}@njust.edu.cn, kaizhang@nju.edu.cn

Abstract

Tuning-free customized generation has gained popularity for its efficiency, particularly with the rise of rectified flow transformers. However, it often suffers from semantic inconsistency, degrading both concept fidelity and prompt alignment. We attribute this failure to their tendency for inference-time superficial feature fusion, which is intrinsic to the tuning-free paradigm. In this paper, we propose SEMANTICFLOW for semantically consistent tuning-free customization, introducing explicit semantic control to jointly enhance concept fidelity and prompt alignment. SEMANTICFLOW operates through a cohesive "perceive-and-manipulate" mechanism for aligned fusion of visual and textual features at the early denoising stage. Specifically, we first devise lightweight semantic token streams as run-time probes to accurately perceive the spatial responses of the target semantics. We then formulate semantic manipulation as an energy-guided rectification of these semantic responses within the rectified flow framework, steering the generation process toward the desired outcome. Experiments demonstrate that SEMANTICFLOW improves the trade-off between concept fidelity and prompt alignment in previously challenging scenarios, effectively enabling semantically-consistent customization.

1 Introduction

Customized text-to-image (T2I) generation aims to synthesize images of user-specified visual concepts in novel textual contexts, which facilitates diverse applications spanning artistic creation to industrial design. Recently, the T2I models based on rectified flow transformers, such as Stable Diffusion 3 (SD3) [10] and FLUX [25], have emerged as powerful generative foundations, demonstrating superior scalability and performance in visual generation tasks. These advancements in foundation models offer promising opportunities for customized generation to evolve from concept-specific optimization [51, 11, 24] to the tuning-free paradigm [62, 28, 4, 57, 6, 60], which trains a generalized model capable of personalizing diverse new concepts.

Compared with tuning-based approaches [16, 12, 36], tuning-free methods [4, 53, 32, 57] exhibit superior computational efficiency, but often compromise on generation quality. Customization quality is evaluated in terms of concept fidelity and prompt adherence, which measure the visual and textual consistency of the generated images with the reference images and text prompts, respectively. Tuning-based methods update the model weights and internal conceptual space, effectively binding new concepts to existing general ones—thus preserving visual fidelity and leveraging generative priors. In contrast, tuning-free methods rely on the generalization of large-scale pre-tuned models to unseen concepts, directly injecting visual information at inference to steer the output toward the reference's likeness. Consequently, they tend to perform only superficial feature fusion rather than deep conceptual implantation, leading to degraded concept fidelity and prompt alignment. As shown

[†]corresponding author.

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



Figure 1: Semantically-consistent customization requires the generated images to exhibit both **visual consistency to the key identifying features of the subject references** and **textual consistency to the prompt**. SemanticFlow significantly enhances visual and textual consistency through explicit semantic control, achieving semantic consistency in tuning-free customization.

in Fig. 2, our extensive analysis confirms that even leading tuning-free methods [65, 35, 3] struggle with certain challenging scenarios.

In this paper, we propose SemanticFlow, a framework for aligned textual and visual feature fusion in tuning-free customization. A key contribution of SemanticFlow is that it introduces explicit semantic control to unify the enhancement of textual and visual consistency. Specifically, textual consistency can be improved by performing spatially aligned injection of textual semantics that would otherwise be omitted. Similarly, enhancing the concept semantics in the image latent strengthens their in-context correspondence to the reference, thereby refining visual feature fusion and improving visual consistency. However, achieving such precise control is nontrivial and fundamentally tied to the underlying model architecture.

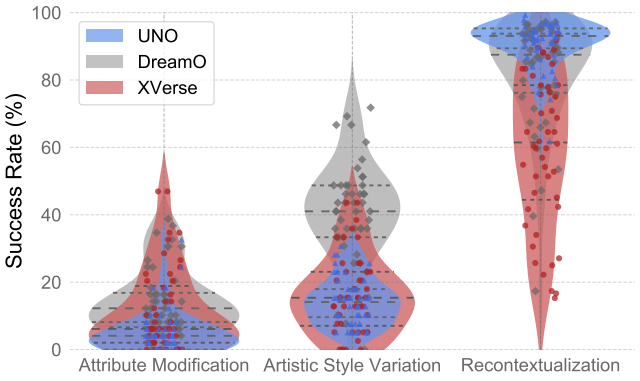


Figure 2: The extensive generations of leading tuning-free methods [65, 35, 3] are evaluated by users, where "success" refers to semantically-consistent customization with satisfactory concept fidelity and prompt alignment. These methods struggle in certain challenging scenarios and exhibit semantic inconsistency, particularly in attribute modification and artistic style variation. Further experimental details are provided in Appendix D.2.

While previous efforts primarily focused on U-Net-based models, where explicit cross-attention structures provide a natural gateway for spatial semantic manipulation, the shift to multimodal transformers poses challenges to directly applying these established practices. In models like SD3 and FLUX, textual and visual conditions are integrated into the image latent within a shared attention space. This leads to semantic leakage and entanglement, which obscure the clear correspondence between semantics and spatial regions. Therefore, precise semantic control in transformer-based models is fundamentally challenged, calling for a new mechanism tailored to their textual and visual conditioning structure.

SemanticFlow introduces a novel design for explicit semantic control in rectified flow transformers, comprising spatially-aware semantic perception and energy-guided semantic manipulation. Specifically, SemanticFlow creates lightweight semantic token streams that serve as run-time probes, tracking intermediate representations for semantic perception during inference. These streams produce semantic responses in the attention output space, accurately capturing the spatial distribution of target semantics. We then formulate spatially-aware semantic manipulation as the rectification

of these semantic responses, converting it into gradient guidance via energy functions within the rectified flow framework. The resulting guidance perturbs the velocity field, steering the inference process toward the desired injection and enhancement of specific semantics in target spatial regions. Extensive experiments demonstrate that SEMANTICFLOW significantly reduces semantic inconsistencies on previously challenging scenarios, jointly boosting concept fidelity and prompt alignment. Our contributions are summarized as follows:

- We reveal a critical yet unaddressed semantic inconsistency challenge in tuning-free customization for MM-DiT.
- We address diverse semantic inconsistency failures by introducing explicit semantic control, proposing a unified perceive-and-manipulate framework.
- Extensive experiments demonstrate that our method reduces semantic inconsistencies in challenging scenarios and can be plugged into other rectified-flow-based methods.

2 Related Work

Text-to-Image Generation. Text-to-image (T2I) generation seeks to translate natural language descriptions into semantically aligned and visually realistic images. Early works accomplish this task using GANs [48, 68] or autoregressive models [45, 8]. More recently, diffusion models [54, 15, 55] have emerged as the dominant framework for T2I generation. Diffusion models [37, 52, 49, 46, 41] generate images by iteratively denoising a random noise vector conditioned on the input text description. The latest advancements [25, 10] replace the conventional U-Net backbone [50] with a diffusion transformer (DiT) [40] and are formulated based on flow matching [29, 30]. These models demonstrate impressive scalability and quality in generation. Post-training can further improve generation quality by aligning models with human preferences [58, 1, 61].

Customized Text-to-Image Generation. Customized T2I generation aims to synthesize images of user-specific visual concepts from textual prompts, which can be broadly categorized into tuning-based and tuning-free paradigms. Tuning-based methods [11, 16, 51, 24, 36, 18, 31, 19] fine-tune pre-trained T2I models for customization. While this paradigm often exhibits superior generation quality, it falls short in efficiency. In contrast, tuning-free methods [62, 4, 28, 53] enable customization for diverse unseen subjects without concept-specific fine-tuning at test time. Recently, large-scale DiT-based models [25, 10] have marked a significant shift in the development of tuning-free customization methods. These methods [26, 17, 57, 5, 66, 65, 35] leverage the in-context learning capacity of DiT and adopt a unified conditioning strategy that jointly processes text embeddings, latent tokens, and VAE-encoded conditioning subjects.

Attention-based Semantic Localization and Control. In U-Net-based models, cross-attention maps provide explicit text-to-region correspondences, which are exploited for image manipulation [14, 47]. For MM-DiTs, recent studies show that joint attention still encodes spatial semantics: ConceptAttention [13] introduces a concept residual stream and derives interpretable saliency maps in the attention output space, while RefAM [22] and Seg4Diff [20] exploit attention maps for zero-shot segmentation. These methods read out semantics for interpretation or segmentation. In contrast, SEMANTICFLOW turns semantic perception into active run-time controllers for tuning-free customization.

3 Method

3.1 Preliminaries

Multi-Modal DiTs. Stable Diffusion 3 (SD3) [10] and FLUX [25] utilize a multi-modal DiT (MM-DiT) architecture for T2I generation. After converting the text and image inputs into token sequences with a common encoding dimension D, MM-DiT employs separate weights for the text and image processing streams, while enabling bidirectional information flow between the two modalities in a multi-modal attention (MM-Attention) module. Specifically, the text and image embeddings are separately projected into key, value, and query features, denoted as $\mathbf{K}_c$, $\mathbf{V}_c$, $\mathbf{Q}_c$ for text and $\mathbf{K}_x$, $\mathbf{V}_x$, $\mathbf{Q}_x$ for images. MM-Attention concatenates the features from two modalities to form unified sequences $\mathbf{K}$, $\mathbf{V}$, and $\mathbf{Q}$, where, for example, $\mathbf{K} = [\mathbf{K}_c, \mathbf{K}_x]$. A self-attention operation is then

applied across all multi-modal tokens:

$$[\mathbf{Z}_c, \mathbf{Z}_x] = \texttt{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})\mathbf{V}, \tag{1}$$

where $\mathbf{Z}_c$ and $\mathbf{Z}_x$ denote the attention outputs of the text and image streams. These outputs are then processed separately by subsequent layers with distinct weights.

Energy Function in Score-based Models. Generative models aim to approximate the data distribution p_0, and score-based approaches sample from it by following the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. When an external condition $\mathbf{y}$ is provided, the sampling process can be guided toward the desired goal by estimating the conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$. Using the Bayesian formula $p(\mathbf{x}_t|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y})}$, the conditional score can be decomposed as:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t), \tag{2}$$

where the first term can be directly estimated by a score-based model, while the second term can be interpreted as a conditional gradient derived from an energy function [27] $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \propto -\nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t; \mathbf{y})$. $\mathcal{E}_t(\mathbf{x}_t; \mathbf{y})$ measures the compatibility between $\mathbf{y}$ and $\mathbf{x}_t$, and minimizing it guides $\mathbf{x}_t$ toward better satisfying $\mathbf{y}$. The energy function in score-based models can be implemented using any differentiable measure of compatibility [9, 47, 34, 7, 37, 39].

3.2 Overview of SEMANTICFLOW

SEMANTICFLOW unifies the enhancement of textual and visual feature fusion in tuning-free customization under a single core mechanism, *i.e.*, spatially-aware semantic control. It consists of spatially-aware semantic perception (Sec. 3.3) and energy-guided semantic manipulation (Sec. 3.4). A conceptual illustration is shown in Fig. 3.

Why Not Perform Semantic Control on Attention Maps? U-Net–based models encode the text prompt once and inject it into model layers via the cross-attention mechanism, as shown in Fig. 3. This provides a consistent and clear semantic signal across different layers and produces explicit cross-attention maps. Notably, each token in the prompt is associated with a specific channel of the attention map, which determines the spatial layout of the corresponding textual semantics in the generated image [14]. Therefore, precise semantic localization and modification can be achieved by directly operating on the attention map.

MM-DiTs adopt a fundamentally different strategy from U-Net to incorporate textual conditions. As presented in Sec. 3.1 and Fig. 3, MM-DiTs forward image and text tokens in parallel and integrate them in the joint attention modules. Each textual token attends to both other textual tokens and image tokens during multi-modal integration, leading to semantic leakage and entanglement. Thus, the clear association between a semantic concept and its attention map becomes increasingly ambiguous with model depth. As shown in Fig. 5, the saliency responses in cross-attention maps fail to accurately capture the spatial distribution of the associated semantics. Therefore, MM-DiT–based models face inherent challenges in leveraging joint attention maps for precise semantic control.

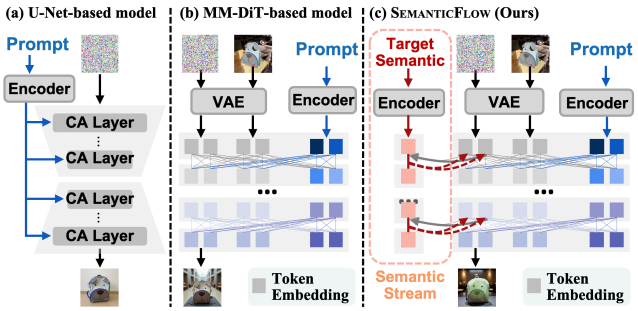


Figure 3: **Injection mechanism of semantic conditions in different models. (a)** Each layer of U-Net-based models receives identical and clear semantic signals from the text encoder and has an independent, explicit cross-attention structure. **(b)** MM-DiTs receive semantic conditions only in the model input space and integrate them with the image latent using a joint attention mechanism. **(c)** SEMANTICFLOW uses semantic streams to perform spatially-aware semantic perception and manipulation.

3.3 Spatially-Aware Semantic Perception

SEMANTICFLOW introduces lightweight semantic token streams for spatially aware semantic perception. Each semantic token stream Φ_s serves as a probe for a target semantic, uses the model weights

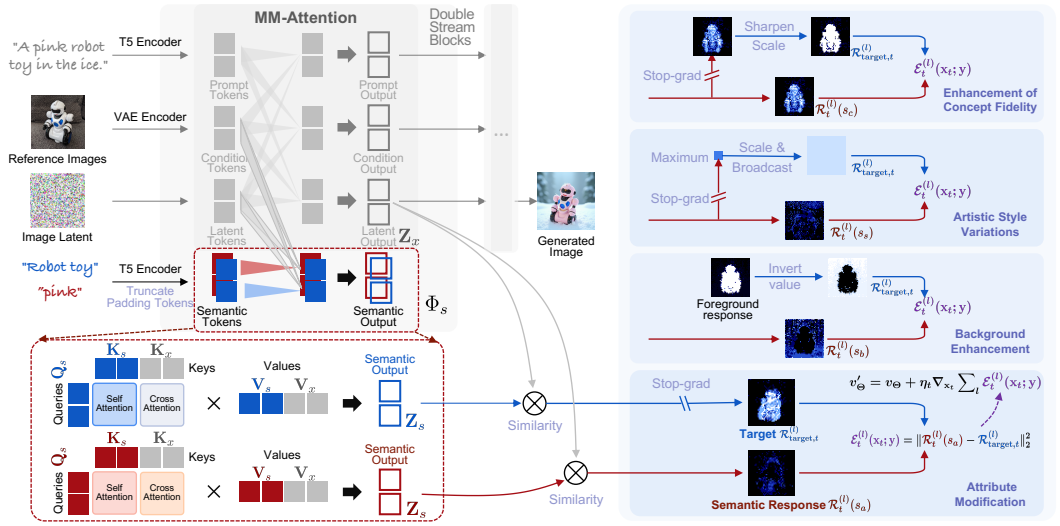


Figure 4: **An overview of SEMANTICFLOW.** The core idea of SEMANTICFLOW is to perform spatially aware semantic manipulation for semantically-consistent customization. The target semantic is specified by its textual description. Lightweight semantic token streams are introduced as run-time probes to accurately capture the spatial distribution of target semantics. Semantic manipulation is then formulated as an energy-guided rectification of these responses within the rectified flow framework, enabling versatile enhancement.

of the textual token stream, and requires no additional training. As depicted in Fig. 4, Φ_s evolves alongside the other token streams across successive transformer blocks, receiving unidirectional interaction [13] from the image token streams in the MM-Attention modules.

A target semantic s, specified by its textual description (*e.g.*, "pink"), is fed into the T5 encoder [44] and then projected to the common patch dimension D. We truncate the padding tokens of the obtained embedding, forming the initial token sequence $c_{\text{init}} \in \mathbb{R}^{d_c \times D}$ for Φ_s. In each MM-Attention module, the semantic tokens of Φ_s are projected into key, value, and query features, denoted as $\mathbf{K}_s$, $\mathbf{V}_s$, and $\mathbf{Q}_s$. We then concatenate $\mathbf{K}_s$ and $\mathbf{V}_s$ with the keys $\mathbf{K}_x$ and values $\mathbf{V}_x$ from the image streams (both latent and reference) to form:

$$\mathbf{K}_u = [\mathbf{K}_s, \mathbf{K}_x], \mathbf{V}_u = [\mathbf{V}_s, \mathbf{V}_x]. \tag{3}$$

We perform attention across semantic and image features:

$$\mathbf{Z}_s = \texttt{softmax}(\mathbf{Q}_s \mathbf{K}_u^\top / \sqrt{d}) \mathbf{V}_u, \tag{4}$$

where $\mathbf{Z}_s$ is the attention output of the semantic stream Φ_s. This interaction creates an information flow from the image streams to the semantic streams, allowing semantic tokens to "observe" the generated latent and reference images and update themselves. $\mathbf{Z}_s$ is then processed separately by subsequent layers. The semantic tokens sequentially evolve in parallel with the other streams. These semantic embeddings dynamically adapt from a generic textual representation into a specific visual descriptor tailored to the image features.

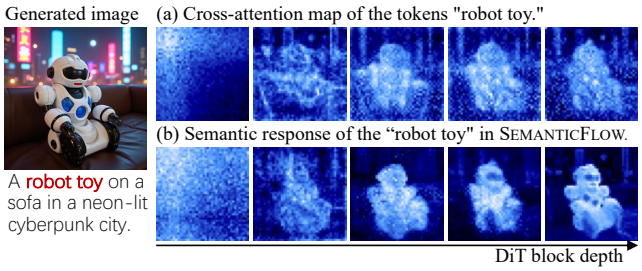


Figure 5: **Saliency visualization.** The saliency responses in cross-attention maps fail to capture the spatial distribution of associated semantics, whereas semantic responses accurately reflect the target semantics across different model layers.

In each MM-Attention layer indexed by l, we compute the similarity between the attention outputs of the semantic and image latent streams:

$$\mathcal{R}^{(l)}(s) = \texttt{softmax}(\mathbf{Z}_s^{(l)} \mathbf{Z}_x^{(l)\top}), \tag{5}$$

where the `softmax` is applied across all semantic streams, including a background semantic stream initialized from a textual description of the background. This operation queries the image latent $\mathbf{Z}_x$ with $\mathbf{Z}_s$, the visual descriptor of semantic s. Since each semantic stream encodes a single textual concept, it yields clear and disentangled semantic features. Moreover, semantic information is highly interpretable and separable in the attention output space [13]. Therefore, $\mathcal{R}^{(l)}(s)$ represents the **semantic response** of the target semantic s, capturing its precise spatial distribution in the layer-l latent features, as shown in Fig. 5. We use $\mathcal{R}^{(l)}$ as a run-time probe to track the latents within the model at inference.

3.4 Energy-Guided Semantic Manipulation

Inspired by the energy-guided mechanism in score-based models, SEMANTICFLOW constructs energy functions within the rectified flow framework to derive gradient guidance for unified semantic manipulation. To incorporate semantic guidance, an energy function $\mathcal{E}_t(\mathbf{x}_t; \mathbf{y})$ encoding the desired condition $\mathbf{y}$ defines an instantaneous energy-reweighted density:

$$q_t(\mathbf{x}_t|\mathbf{y}) \propto p_t(\mathbf{x}_t) \exp(-\mathcal{E}_t(\mathbf{x}_t; \mathbf{y})). \tag{6}$$

We use this density to motivate local energy reduction during sampling. Rectified flow-based models directly learn a parameterized velocity field $v_\Theta(\mathbf{x}_t, t)$ that governs the evolution of $\mathbf{x}_t$, generating samples approximately following $p_0(\mathbf{x}_0)$ at $t = 0$. The following lemma formalizes how $\mathcal{E}_t(\mathbf{x}_t; \mathbf{y})$ induces guidance on v_Θ in the rectified flow model.

Lemma 3.1 (Guidance as Velocity Field Perturbation)**.** *Under the shared-SDE assumption, the conditioned velocity field $v'_\Theta(\mathbf{x}_t, t)$ corresponding to the guided distribution above is obtained by perturbing the original velocity field with the gradient of $\mathcal{E}_t(\mathbf{x}_t; \mathbf{y})$:*

$$v'_\Theta(\mathbf{x}_t, t) = v_\Theta(\mathbf{x}_t, t) + \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t; \mathbf{y}), \tag{7}$$

where $g(t)$ is the diffusion coefficient of the underlying SDE. We provide the full proof in Appendix A.1.

We construct energy functions for semantic manipulation based on the semantic response $\mathcal{R}$, as it captures highly accurate and interpretable semantic information. Specifically, at time step t of the l-th MM-Attention layer, we extract $\mathcal{R}_t^{(l)}(s)$ of s and formulate the energy function as:

$$\mathcal{E}_t^{(l)}(\mathbf{x}_t; \mathbf{y}) = \|\mathcal{R}_t^{(l)}(s) - \mathcal{R}_{\text{goal},t}^{(l)}(s)\|_2^2, \tag{8}$$

where $\mathcal{R}_{\text{goal},t}^{(l)}(s)$ is the desired semantic response for s, guiding the distribution of s toward it. Motivated by Eq. (7), we add a scaled energy gradient to the learned velocity field to steer the reverse-time update toward lower semantic energy. After each forward pass, we collect layer-wise energy functions to calculate perturbations based on Eq. (7) for velocity adjustment:

$$v'_\Theta(\mathbf{x}_t, t) = v_\Theta(\mathbf{x}_t, t) + \eta_t \nabla_{\mathbf{x}_t} \sum\nolimits_l \mathcal{E}_t^{(l)}(\mathbf{x}_t; \mathbf{y}), \tag{9}$$

where η_t is the weighting factor. $v'_\Theta(\mathbf{x}_t, t)$ replaces $v_\Theta(\mathbf{x}_t, t)$ in Eq. (25) to update the latent. The gradient guidance steers the rectification of semantic responses in the early denoising stage, guiding specific semantics toward the correct spatial distribution. By constructing appropriate $\mathcal{R}_{\text{goal}}(s)$ for the semantic s needing rectification, SEMANTICFLOW can perform versatile semantic manipulation. Users can freely specify the target semantics and manipulation type. Below, we detail how to construct $\mathcal{R}_{\text{goal}}(s)$ for different types of semantic manipulation:

Attribute Modification. For a target attribute semantic s_a (*e.g.*, "pink" in the case "A pink robot toy"), we create a semantic token stream for its entity-noun s_c (*i.e.*, "toy") in parallel, obtaining the semantic response $\mathcal{R}_t^{(l)}(s_c)$ and performing `stop-grad` operation for it. Intuitively, the desired spatial distribution of s_a should be consistent with s_c. Thus, we set $\mathcal{R}_{\text{goal},t}^{(l)}(s_a) = \mathcal{R}_t^{(l)}(s_c)$ for the spatially aligned injection of the attribute s_a.

Enhancement of Concept Fidelity. We perform rectification on the concept semantic s_c to improve its fidelity. The goal is constructed from $\mathcal{R}_t^{(l)}(s_c)$ by sharpening its spatial distribution and then

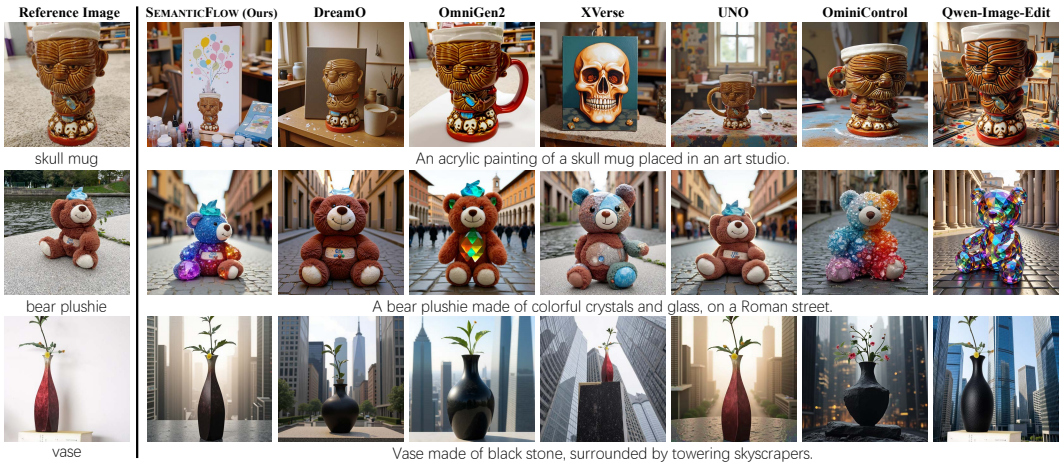


Figure 6: **Single-concept qualitative comparison.** We compare SEMANTICFLOW with competitive baselines. Baselines struggle to generate images that are both faithful to the concept and consistent with the prompt. In contrast, our method faithfully generates all the desired content specified in the prompt while preserving the key identifying features of the customized concepts.



Figure 7: **Comparison with editing methods.** The editing methods distort the key features of customized concepts. Our method preserves concept fidelity while adhering to the prompt.

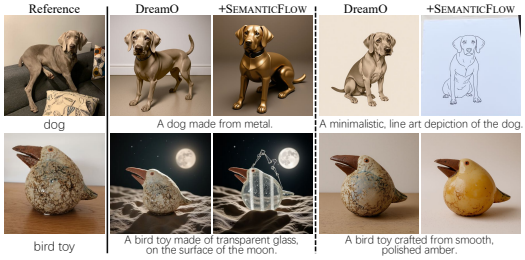


Figure 8: SEMANTICFLOW can be combined with methods that are based on the rectified flow transformer, such as DreamO [35], alleviating their semantic inconsistency issue.

scaling it by a factor of $\lambda = 1.5$. This goal steers the model to amplify the response of s_c in the latent, thereby strengthening their in-context correspondence to the reference and enhancing concept fidelity.

Artistic Style Variations. The responses of style-related semantics s_s are expected to manifest across the entire image. To construct the goal $\mathcal{R}^{(l)}_{\text{goal},t}(s_s)$, we first identify the maximum activation within the semantic response of s_s across spatial dimensions. This response, $\texttt{max}(\mathcal{R}^{(l)}_t(s_s))$, is then scaled by a factor of $\lambda = 1.5$ and broadcast over the full spatial extent of the latent, yielding a uniform goal map that guides the model to enhance the style consistently throughout the generated image.

Background Enhancement. We can enhance the background semantic s_b for generation tasks such as recontextualization. The goal is constructed from the inverted value of $\mathcal{R}^{(l)}_{\text{goal},t}(s_c)$ or a user-specified region mask.

4 Experiments

4.1 Experimental Setups

Dataset. Experiments are conducted on concept datasets proposed in three widely used prior works [51, 11, 24] on customized generation. These datasets encompass a diverse range of personalized concepts. The evaluation prompts are also drawn from these datasets and cover various generation types, including 144 prompts for recontextualization, 49 for attribute modification, and 39

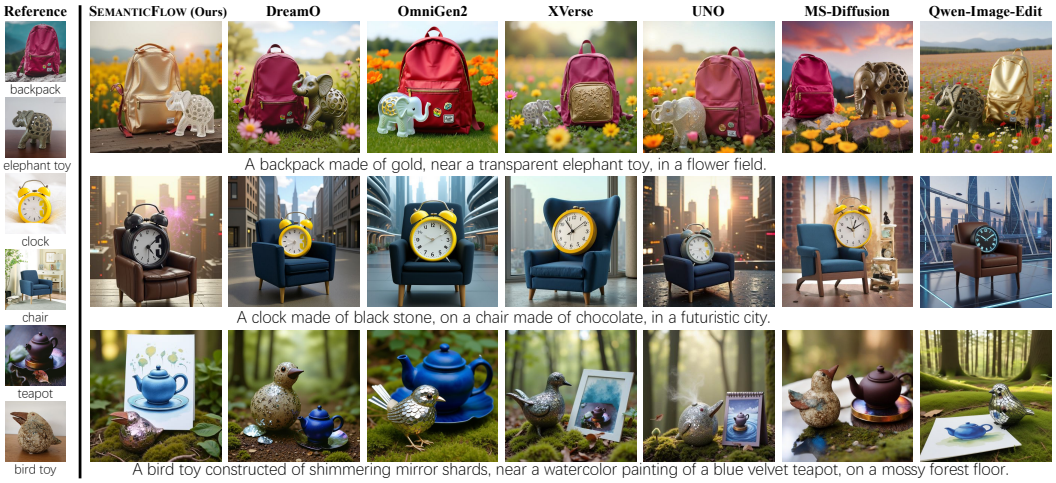


Figure 9: **Multi-concept qualitative comparison.** Our method enables multi-concept enhancements such as proper attribute binding and fidelity improvement, generating semantically-consistent images that demonstrate both high concept fidelity and prompt adherence.

for artistic style variation. For single-concept generation, each concept is evaluated with 30 prompts and 2 random seeds. For multi-concept generation, 15 concept pairs are each evaluated with 15 prompts and 2 random seeds. More details are provided in Appendix D.1.

Evaluation Metrics. Following previous studies, we evaluate customized generation along two dimensions: 1) Adherence to the prompt (Text alignment): measured by the average CLIP [43] similarity ($S^{\mathrm{T}}_{\mathrm{CLIP}}$) between the generated images and their textual prompts. 2) Consistency with the reference (Concept alignment): measured by CLIP and DINO [2] similarities ($S^{\mathrm{I}}_{\mathrm{CLIP}}$ and S_{DINO}) between the generated images and the reference images. For generations containing multiple subjects, we compute the similarity of each subject in the generated images to its corresponding reference individually, and then report the mean value.

Experimental Details. We compare SEMANTICFLOW with state-of-the-art tuning-free methods, including MS-Diffusion [59], OminiControl [57], UNO [65], DreamO [35], XVerse [3], and OmniGen2 [64]. We also compare our method with the editing methods [21, 23, 70, 63] for reference. UNO [65] is used as the base model by default. The pre-trained T2I model is FLUX.1-dev [25]. In the data constructed for our experiments, the target semantics are automatically determined in batch by parsing prompts with Qwen2.5-3B-Instruct [42]. In practical use, target semantics can be either automatically constructed via the lightweight LLM or manually specified by users. Manual specification grants users greater flexibility and controllability. Implementation details and more experimental settings are provided in Appendix D.3.

4.2 Main Results

Qualitative Comparison. We qualitatively compare SEMANTICFLOW with baselines for single-concept customized generation. The results are presented in Fig. 6. For generations involving subject attribute modification and artistic style variation, the baseline methods often face challenges in generating images with both high concept fidelity and prompt adherence. Some baselines, such as XVerse, generate images that align well with the textual prompt but lack concept consistency. Others miss specific contextual content and exhibit poor textual alignment. In contrast, our method faithfully generates all the desired content specified in the prompt while preserving the key identifying features of the customized concepts. Fig. 9 shows the comparison results for multi-subject generation. Our method effectively recovers missing semantics and achieves multi-concept generation enhancements, such as proper attribute binding. Moreover, various semantic manipulations can be flexibly combined to realize multiple effects simultaneously, such as attribute injection and style variation. We further compare our method with several recent image editing approaches [21, 23, 70] and present the results in Fig. 7. The editing methods typically distort the key identifying features of customized concepts.

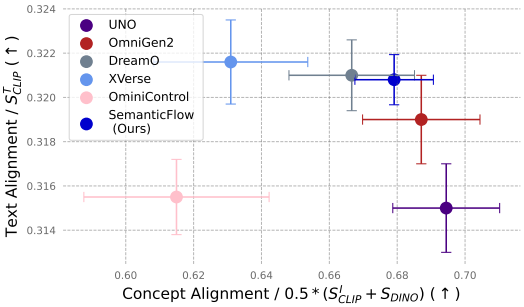


Figure 10: **Single-concept quantitative comparison.** SEMANTICFLOW shows superior performance among all methods in consideration of the fidelity–editability trade-off (see Appendix C.4)

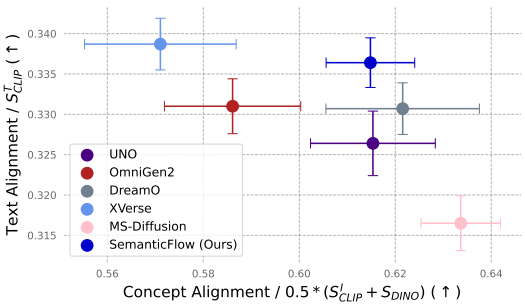


Figure 11: **Multi-concept quantitative comparison.** SEMANTICFLOW outperforms competing methods considering the fidelity–editability trade-off (see Appendix C.4).

SEMANTICFLOW can also be combined with other methods based on the rectified flow transformer, as shown in Fig. 8, alleviating their semantic inconsistency issue. We provide more visual comparisons in the Appendix.

Quantitative Comparison. Fig. 10 and Fig. 11 present the quantitative results for single-concept and multi-concept customized generation, respectively. Each point denotes the mean over all evaluation generations of each method, and the error bars denote the standard error of the mean across concepts. As demonstrated, all methods generally exhibit a trade-off [24, 11] between the concept alignment and textual alignment metrics. Methods such as XVerse struggle to preserve subject identity, resulting in low concept alignment. Baselines like MS-Diffusion tend to lose editability, generating images with high visual alignment but poor prompt adherence. These results fail to achieve the essential goals of semantically-consistent customized generation. In contrast, SEMANTICFLOW achieves a favorable trade-off between concept fidelity and textual alignment compared with the other methods in both scenarios.

Variant	$S^{\mathrm{T}}_{\mathrm{CLIP}}$ (↑)	$S^{\mathrm{I}}_{\mathrm{CLIP}}$ (↑)	S_{DINO} (↑)
CA-op (w/o SR-op)	0.3165	0.6908	0.6530
w/o sep. stream	0.3102	0.7265	0.6262
w/o velocity adjustment	0.3025	0.7026	0.6316
SEMANTICFLOW (Ours)	0.3313	0.7135	0.6460

Table 1: **Ablation study.** Two key components are necessary for accurate semantic control. The generations significantly degrade in both text and concept alignment without velocity adjustment.

User Study. We conduct a user study to further evaluate SEMANTICFLOW. Our method demonstrates a strong user preference for semantically-consistent customization. More details and results are provided in Appendix D.2.

4.3 Ablation Studies

Key Components in Spatially-Aware Semantic Perception. We ablate two components of spatially-aware semantic perception. 1) We perform semantic operations on the semantic response $\mathcal{R}$ rather than on cross-attention maps. The *"CA-op (w/o SR-op)"* variant instead constructs the energy functions directly on the cross-attention maps in MM-Attention. 2) We construct a separate semantic stream for each textual concept, such as customized concepts and subject attributes. The *"w/o sep. stream"* variant instead concatenates all semantic descriptions and processes them in a single stream. We present qualitative examples in Fig. 12 and quantitative results on a subset of the concepts in Table 1. Both variants show clearly lower text alignment, i.e., they fail to inject the target semantics, while their concept-alignment metrics diverge in ways that reflect their distinct failure modes. This divergence arises because $S^{\mathrm{I}}_{\mathrm{CLIP}}$ captures category-level semantics, whereas S_{DINO} captures instance-level structure and texture [51]. CA-op suffers from inaccurate spatial localization. Semantics are injected into wrong regions, which leaves the subject largely unmodified (higher S_{DINO}) but corrupts other image regions (lower $S^{\mathrm{I}}_{\mathrm{CLIP}}$). A single shared stream causes entanglement among semantics. The target semantics fail to be injected, so the output stays close to the reference at the category level

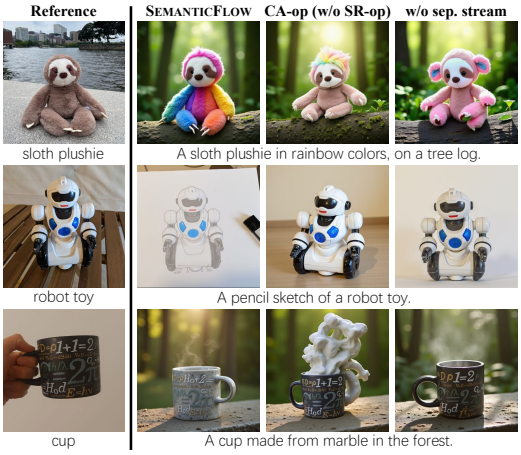


Figure 12: **Ablation of two key components.** Both constructing energy on semantic responses (SR) and processing semantics in separate streams are essential for accurate semantic control.

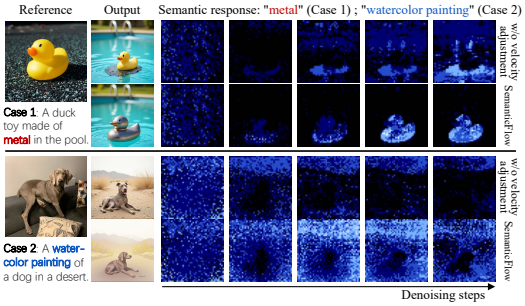


Figure 13: **Evolution of semantic response along denoising steps.** Without velocity adjustment, the target semantics exhibit weak and spatially misaligned responses, being absent from the generated images. Our method rectifies the spatial distribution of the target semantics and successfully generates them.

(higher $S^{\mathrm{I}}_{\mathrm{CLIP}}$), while cross-semantic interference distorts the fine-grained structure of the subject (lower S_{DINO}).

Energy-Guided Semantic Manipulation. The velocity perturbation derived from the energy function provides gradient guidance for semantic manipulation. In the *"w/o velocity adjustment"* variant shown in Fig. 13 and Table 1, we ablate the velocity adjustment in Eq. (9) by directly using the original velocity for latent updates. As observed, energy-guided velocity adjustment effectively recovers missing semantics and enhances concept fidelity in challenging scenarios, whereas without it, the results exhibit low textual and visual alignment. As shown in Fig. 14, performing adjustment for a few steps in the early denoising stage is sufficient for semantic manipulation. More ablation studies are provided in Appendix C.1.



Figure 14: N_a denotes the number of adjustment steps at the beginning of the denoising process out of a total of 25 steps. Performing a few adjustment steps in the early denoising stage is sufficient for semantic manipulation, and we set $N_a = 4$.

5 Conclusion

In this work, we introduce SEMANTICFLOW, a novel framework for joint precise visual and textual feature fusion in rectified flow transformers, addressing the key challenge of semantic inconsistency in tuning-free customization. It leverages a lightweight run-time probe to achieve accurate, spatially-aware semantic perception. This enables a targeted, energy-guided manipulation of the generation process, providing a unified solution for enhancing both concept fidelity and prompt alignment. Our method proves effective in achieving semantically consistent customization, even in challenging scenarios. This framework provides a powerful and generalizable tool for fine-grained control over generative models. We hope this work will inspire further research on semantic control mechanisms.

Acknowledgments and Disclosure of Funding

This work was supported by the NSFC under Grant Nos. U24A20330, 62276132, and 61876085.

References

[1] Akhil Agnihotri, Rahul Jain, Deepak Ramachandran, and Zheng Wen. Multi-objective preference optimization: Improving human alignment of generative models. *arXiv preprint arXiv:2505.10892*, 2025.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640, 2021.

[3] Bowen Chen, Mengyi Zhao, Haomiao Sun, Li Chen, Xu Wang, Kang Du, and Xinglong Wu. Xverse: Consistent multi-subject control of identity and semantic attributes via DiT modulation. In *NeurIPS*, 2025.

[4] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. In *NeurIPS*, pages 30286–30305, 2023.

[5] Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *CVPR*, pages 12501–12511, 2025.

[6] Yufan Deng, Yuanyang Yin, Xun Guo, Yizhi Wang, Jacob Zhiyuan Fang, Shenghai Yuan, Yiding Yang, Angtian Wang, Bo Liu, Haibin Huang, et al. Magref: Masked guidance for any-reference video generation with subject disentanglement. *arXiv preprint arXiv:2505.23742*, 2025.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, volume 34, pages 8780–8794, 2021.

[8] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, pages 19822–19835, 2021.

[9] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, volume 36, pages 16222–16239, 2023.

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, pages 12606–12633, 2024.

[11] 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 *ICLR*, 2023.

[12] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *ICCV*, pages 7323–7334, 2023.

[13] Alec Helbling, Tuna Han Salih Meral, Ben Hoover, Pinar Yanardag, and Duen Horng Chau. ConceptAttention: Diffusion transformers learn highly interpretable features. In *ICML*, 2025.

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

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

[18] Jian Jin, Yang Shen, Zhenyong Fu, and Jian Yang. Customized generation reimagined: Fidelity and editability harmonized. In *ECCV*, pages 410–426, 2024.

[19] Jian Jin, Zhenbo Yu, Yang Shen, Zhenyong Fu, and Jian Yang. Latexblend: Scaling multi-concept customized generation with latent textual blending. In *CVPR*, pages 23585–23594, 2025.

[20] Chaehyun Kim, Heeseong Shin, Eunbeen Hong, Heeji Yoon, Anurag Arnab, Paul Hongsuck Seo, Sunghwan Hong, and Seungryong Kim. Seg4diff: Unveiling open-vocabulary semantic segmentation in text-to-image diffusion transformers. In *NeurIPS*, volume 38, pages 71685–71724, 2025.

[21] Jimyeong Kim, Jungwon Park, Yeji Song, Nojun Kwak, and Wonjong Rhee. Reflex: Text-guided editing of real images in rectified flow via mid-step feature extraction and attention adaptation. In *ICCV*, pages 15939–15948, 2025.

[22] Anna Kukleva, Enis Simsar, Alessio Tonioni, Muhammad Ferjad Naeem, Federico Tombari, Jan Eric Lenssen, and Bernt Schiele. Refam: Attention magnets for zero-shot referral segmentation. *arXiv preprint arXiv:2509.22650*, 2025.

[23] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, pages 19721–19730, 2025.

[24] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, pages 1931–1941, 2023.

[25] Black Forest Labs. Flux. black-forest-labs/flux, 2024.

[26] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`.

[27] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

[28] Dongxu Li, Junnan Li, and Steven Hoi. BLIP-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *NeurIPS*, pages 30146–30166, 2023.

[29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[30] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[31] Zhiheng Liu, Yifei Zhang, Yujun Shen, Kecheng Zheng, Kai Zhu, Ruili Feng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones 2: Customizable image synthesis with multiple subjects. In *NeurIPS*, pages 57500–57519, 2023.

[32] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. In *SIGGRAPH*, pages 1–12, 2024.

[33] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM computing surveys (CSUR)*, 54(1):1–41, 2021.

[34] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *ICLR*, 2024.

[35] Chong Mou, Yanze Wu, Wenxu Wu, Zinan Guo, Pengze Zhang, Yufeng Cheng, Yiming Luo, Fei Ding, Shiwen Zhang, Xinghui Li, et al. Dreamo: A unified framework for image customization. In *SIGGRAPH Asia*, 2025.

[36] Jisu Nam, Heesu Kim, DongJae Lee, Siyoon Jin, Seungryong Kim, and Seunggyu Chang. Dreammatcher: Appearance matching self-attention for semantically-consistent text-to-image personalization. In *CVPR*, pages 8100–8110, 2024.

[37] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, pages 16784–16804, 2022.

[38] Lianyu Pang, Jian Yin, Baoquan Zhao, Feize Wu, Fu Lee Wang, Qing Li, and Xudong Mao. Attndreambooth: Towards text-aligned personalized text-to-image generation. In *NeurIPS*, pages 39869–39900, 2024.

[39] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH*, pages 1–11, 2023.

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.

[41] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

[42] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

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

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021.

[46] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

[47] Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. In *NeurIPS*, volume 36, pages 3536–3559, 2023.

[48] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016.

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[51] 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 *CVPR*, pages 22500–22510, 2023.

[52] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, pages 36479–36494, 2022.

[53] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *CVPR*, pages 8543–8552, 2024.

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015.

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[57] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024.

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

[59] Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. Ms-diffusion: Multi-subject zero-shot image personalization with layout guidance. In *ICLR*, 2025.

[60] Yuanzhi Wang, Xuhua Ren, Jiaxiang Cheng, Bing Ma, Kai Yu, Sen Liang, Wenyue Li, Tianxiang Zheng, Qinglin Lu, and Zhen Cui. Faithfulfaces: Pose-faithful facial identity preservation for text-to-video generation. *arXiv preprint arXiv:2605.04702*, 2026.

[61] Yuanzhi Wang, Xuhua Ren, Jiaxiang Cheng, Bing Ma, Kai Yu, Tianxiang Zheng, Qinglin Lu, and Zhen Cui. Implicit preference alignment for human image animation. In *ICML*, volume 306, pages 129632–129651, 2026.

[62] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *ICCV*, pages 15943–15953, 2023.

[63] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL `https://arxiv.org/abs/2508.02324`.

[64] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[65] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. In *ICCV*, 2025.

[66] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *CVPR*, pages 13294–13304, 2025.

[67] Zebin Yao, Lei Ren, Huixing Jiang, Wei Chen, Xiaojie Wang, Ruifan Li, and Fangxiang Feng. Freegraftor: Training-free cross-image feature grafting for subject-driven text-to-image generation. *arXiv preprint arXiv:2504.15958*, 2025.

[68] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5907–5915, 2017.

[69] Yuxuan Zhang, Yiren Song, Jiaming Liu, Rui Wang, Jinpeng Yu, Hao Tang, Huaxia Li, Xu Tang, Yao Hu, Han Pan, et al. SSR-Encoder: Encoding selective subject representation for subject-driven generation. In *CVPR*, pages 8069–8078, 2024.

[70] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. In *NeurIPS*, 2025.

Appendix

The appendix is organized as follows. In Section A, we provide the full proof of the lemma. Section B reviews rectified flow. In Section C, we present additional experimental results, including further ablation studies, more analysis of the proposed method, more visual comparisons, cost analysis, and a discussion of limitations. In Section D, we provide the implementation details of our method and the baselines.

A Technical Proofs

A.1 Proof of Lemma 3.1

Guidance in Generative Models. Let $p_t(\mathbf{x}_t)$ denote the marginal density of the unguided model at time t, and let $q_t(\mathbf{x}_t \mid \mathbf{y})$ denote the guided density that incorporates desired constraints or conditions $\mathbf{y}$. According to Bayes' theorem, this posterior can be expressed as

$$q_t(\mathbf{x}_t|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}_t) \cdot p_t(\mathbf{x}_t), \tag{10}$$

where $p(\mathbf{y}|\mathbf{x}_t)$ is a likelihood function quantifying how well a sample $\mathbf{x}_t$ satisfies the guidance $\mathbf{y}$.

For computational convenience, we work with log-probabilities:

$$\log q_t(\mathbf{x}_t|\mathbf{y}) = \log p(\mathbf{y}|\mathbf{x}_t) + \log p_t(\mathbf{x}_t) + C, \tag{11}$$

where C is a constant.

Following the energy-based modeling paradigm [27], we define an energy function as the negative log-likelihood of the guidance:

$$\mathcal{E}_t(\mathbf{x}_t; \mathbf{y}) = -\log p(\mathbf{y} \mid \mathbf{x}_t). \tag{12}$$

This energy measures the compatibility between the condition $\mathbf{y}$ and the sample $\mathbf{x}_t$. Intuitively, a lower energy corresponds to a sample that better satisfies the guidance. At each fixed time t, any differentiable energy $\mathcal{E}_t$ defines a valid energy-reweighted density, provided that the resulting density can be normalized.

Substituting Eq. (12) into Eq. (11) gives

$$\log q_t(\mathbf{x}_t \mid \mathbf{y}) = \log p_t(\mathbf{x}_t) - \mathcal{E}_t(\mathbf{x}_t; \mathbf{y}) + C. \tag{13}$$

Taking the gradient of Eq. (13) with respect to $\mathbf{x}_t$ gives the posterior score:

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t; \mathbf{y}). \tag{14}$$

This equation forms the basis of all guidance methods. The direction towards a guided sample is given by the sum of the original model's score and the negative gradient of the energy function, which pushes the sample towards satisfying the desired guidance.

The Relationship between Velocity Field and Score Function. The connection between the velocity field and the score function is formalized through the Probability Flow Ordinary Differential Equation (ODE) [56]. For any stochastic differential equation (SDE) defining a diffusion process, there exists a corresponding deterministic ODE that evolves samples while preserving the marginal distributions of the SDE. The general form of the Probability Flow ODE is:

$$\frac{d\mathbf{x}_t}{dt} = f(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), \tag{15}$$

where $f(\mathbf{x}_t, t)$ is the drift and $g(t)$ is the scalar diffusion coefficient of the underlying SDE, and

$$s(\mathbf{x}_t, t) \triangleq \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \tag{16}$$

denotes the score function of the marginal distribution at time t. The *velocity field*, $v(\mathbf{x}_t, t)$, is defined as the total deterministic drift of the process, *i.e.*, the time derivative of $\mathbf{x}_t$:

$$v(\mathbf{x}_t, t) \triangleq \frac{d\mathbf{x}_t}{dt} = f(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 s(\mathbf{x}_t, t). \tag{17}$$

This formulation establishes that the evolution of $\mathbf{x}_t$ under the deterministic flow is fully determined by the combination of the SDE drift and the scaled score function, linking the probabilistic gradient information directly to the velocity of the diffusion process.

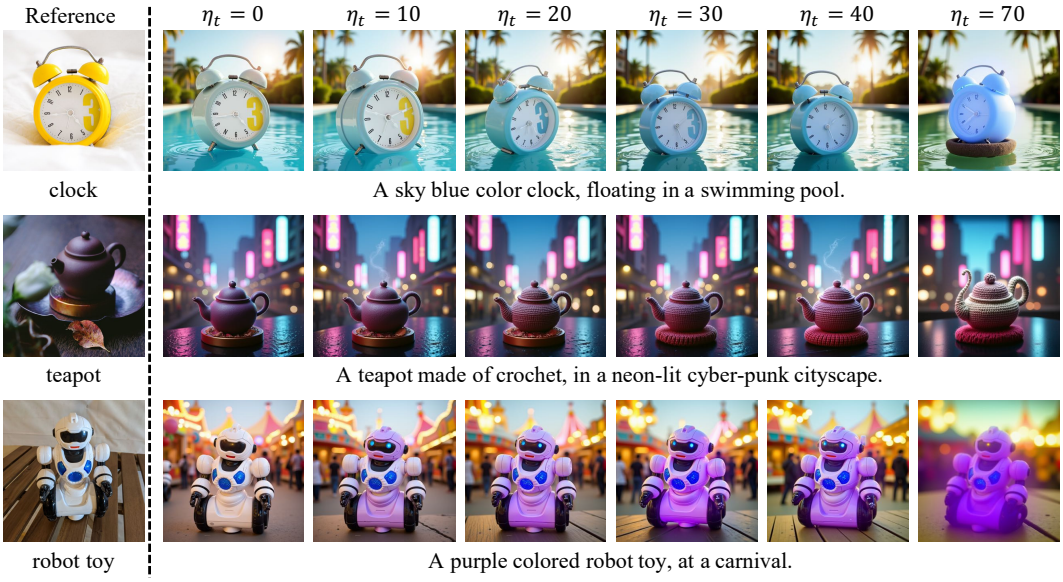


Figure 15: **Effect of varying η_t in velocity adjustment.** The model is generally robust across a wide range of η_t, injecting the target semantics to achieve high prompt alignment while preserving the key identifying features of the customized subjects.

Derivation of the Guided Velocity Update Rule. For this derivation, suppose that the original density p_t and the guided density $q_t \propto p_t \exp(-\mathcal{E}_t(\cdot;\mathbf{y}))$ are marginal density families of SDEs sharing the same drift $f(\mathbf{x},t)$ and diffusion coefficient $g(t)$. Under this assumption, their probability-flow velocity fields can be expressed using the same f and g. From the Bayesian perspective, the score function of the guided distribution is given by:

$$\begin{aligned} s'(\mathbf{x}_t,t) &= \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{y}) \\ &= \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t;\mathbf{y}). \end{aligned} \tag{18}$$

Let $s_\Theta(\mathbf{x}_t,t)$ denote the score of the original (unguided) model distribution, *i.e.*, $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. Then,

$$s'(\mathbf{x}_t,t) = s_\Theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t;\mathbf{y}). \tag{19}$$

Under the shared-SDE assumption stated above, the probability-flow velocity field associated with the guided density is:

$$v'(\mathbf{x}_t,t) = f(\mathbf{x}_t,t) - \frac{1}{2} g(t)^2 s'(\mathbf{x}_t,t). \tag{20}$$

Substituting Eq. (19) into the score term of Eq. (20) yields:

$$\begin{aligned} v'(\mathbf{x}_t,t) &= f(\mathbf{x}_t,t) - \tfrac{1}{2} g(t)^2 \left[s_\Theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t;\mathbf{y}) \right] \\ &= \left[f(\mathbf{x}_t,t) - \tfrac{1}{2} g(t)^2 s_\Theta(\mathbf{x}_t,t) \right] + \tfrac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t;\mathbf{y}). \end{aligned} \tag{21}$$

According to Eq. (17), the expression in brackets corresponds to the velocity field of the unguided model:

$$v_\Theta(\mathbf{x}_t,t) = f(\mathbf{x}_t,t) - \frac{1}{2} g(t)^2 s_\Theta(\mathbf{x}_t,t). \tag{22}$$

Thus, the guided velocity field can be written as:

$$v'_\Theta(\mathbf{x}_t,t) = v_\Theta(\mathbf{x}_t,t) + \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t} \mathcal{E}_t(\mathbf{x}_t;\mathbf{y}). \tag{23}$$

This shows that the guided velocity v'_Θ equals the original velocity v_Θ plus a correction term proportional to $\frac{1}{2} g(t)^2$ times the gradient of the energy function.



Figure 16: **Effect of varying λ.** The model is not sensitive to λ and remains generally robust within the range of $1.0 \sim 2.0$. In all of our experiments, we set $\lambda = 1.5$ for both concept-fidelity enhancement and artistic style variation.

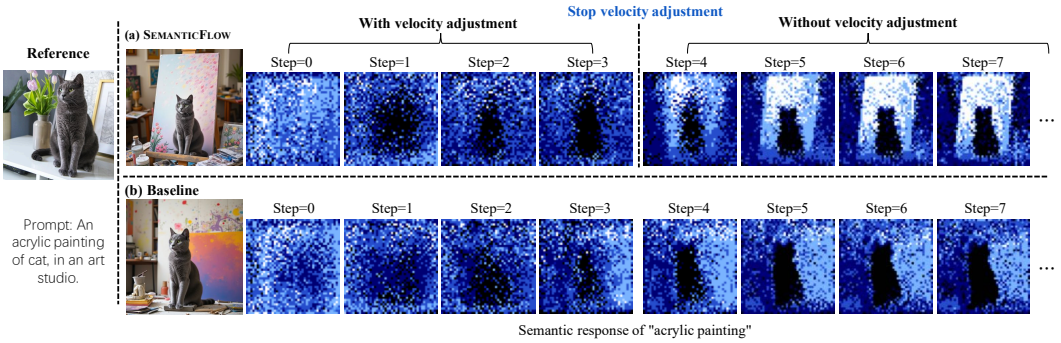


Figure 17: **Visualization of semantic response.** In this case, at the beginning of the denoising process, velocity adjustment steers the response of the style semantic across the entire image and provides a good initialization, allowing the style semantic to emerge in the appropriate regions (a local region in this case) according to the prompt during subsequent denoising steps. In contrast, without velocity adjustment, the style semantic remains weak throughout the process, causing the target semantic to fail to appear in the final output.

B Rectified Flow

Flow-based generative models learn to transport samples from a noise distribution p_1 to a data distribution p_0. Rectified flow [30] trains a velocity field using straight-line interpolations between data samples $\mathbf{x}_0 \sim p_0$ and noise samples $\mathbf{x}_1 \sim p_1$:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \quad t \in [0,1], \quad \mathbf{x}_1 \sim p_1 = \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{24}$$

The outputs of FLUX and SD3 parameterize the velocity field v_Θ that defines the governing ODE. At inference, images are generated by solving this ODE, transporting samples from p_1 at time $t = 1$ to p_0 at time $t = 0$ along the learned trajectory over N discrete time steps. The iterative update rule for a single step using the Euler method is given by:

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - v_\Theta(\mathbf{x}_t, t) \cdot \Delta t, \tag{25}$$

where Δt denotes the step size (*e.g.*, $1/N$).

C More Experimental Results

C.1 More Ablation Studies

Weighting Factor η_t in Velocity Adjustment. The weighting factor η_t in Eq. (9) controls the guidance strength of the velocity adjustment. Lemma 3.1 motivates the form of the velocity correction and the probability-flow ODE derivation in Eq. (7) motivates a coefficient of $\frac{1}{2}g(t)^2$ for the energy gradient. In practice, we replace this coefficient with a tunable guidance strength η_t, which is held constant across the active guidance steps for simplicity and controllability. We examine the effect of varying η_t on customized generation and provide visual examples in Fig. 15. Specifically, the samples exhibit semantic inconsistency when η_t is small. As η_t increases, more target content is injected into the generated images. The results indicate that the model is generally robust across a wide range of η_t, injecting the target semantics for high prompt alignment while preserving key identifying features of the customized subjects. However, when η_t becomes too large, concept fidelity begins to decline. Overall, the model's performance is not sensitive to η_t and functions reliably across a broad parameter interval. We use the same η_t for each type of semantic manipulation across all our experiments. Specifically, we set $\eta_t = 30$ for attribute modification and background enhancement, $\eta_t = 10$ for artistic style variation, and $\eta_t = 15$ for enhancement of concept fidelity.

Scaling Factor λ. We use the scaling factor λ in goal construction for the tasks of enhancing concept fidelity and applying artistic style variations. We examine the effect of varying λ and present visual examples in Fig. 16. In these examples, we apply velocity adjustment for both concept-fidelity enhancement and artistic style variation under different values of λ. As shown, the model is not sensitive to λ and remains generally robust within the range of $1.0 \sim 2.0$. In all of our experiments, we set $\lambda = 1.5$ for both concept-fidelity enhancement and artistic style variation.

C.2 Extended Analysis of Velocity Adjustment

Visualization. Velocity adjustment is applied only during the early steps of the denoising process, providing a good initialization and correcting the denoising direction for semantically consistent customization. We present a visual example in Fig. 17 for illustration. In this case, at the beginning of the denoising process, velocity adjustment steers the response of the style semantic across the entire image and provides a correct initialization, allowing the style semantic to emerge in the appropriate regions (a local region in this case) according to the prompt during subsequent denoising steps. In contrast, without velocity adjustment, the style semantic remains weak throughout the process, causing the target semantic to fail to appear in the final output.

Effects on the Baseline's Semantically Consistent Generation. For some cases, the baseline already achieves semantically consistent customization. We investigate the effect of velocity adjustment in these cases and present the corresponding samples of attribute modification and style transfer in Fig. 18. As observed, velocity adjustment shows little impact in these scenarios and preserves the semantically consistent generations. A possible explanation is that the mismatch between the semantic response $\mathcal{R}(s)$ and the desired goal $\mathcal{R}_{\text{goal}}(s)$ is relatively small and that velocity adjustments have only a marginal effect. The guidance goals are compatible with the existing semantic structure in these cases, so velocity adjustment primarily reinforces already-correct semantics without substantially altering the generation.

Robustness to Incorrect Specification. Target semantics are specified either by a lightweight LLM or directly by the user. We conduct a preliminary study to evaluate how SEMANTICFLOW degrades under incorrect semantic specifications. Specifically, we deliberately corrupt the specifications in multi-semantic cases in two ways: (a) pairing attributes with incorrect entities and (b) omitting one target semantic. As shown in Table 2, corruption (a) leads to localized mis-injection that follows the incorrect instruction—itself evidence of precise spatial control—while other semantics remain correctly controlled. Corruption (b) simply causes the omitted semantic to revert to baseline behavior.



Figure 18: For some cases, the baseline already achieves semantically consistent customization. Velocity adjustment shows negligible negative impact in these scenarios and preserves the semantically consistent generations produced by the baseline.

Table 2: **Robustness to incorrect specification of target semantics.**

Variant	$S^{\mathrm{T}}_{\mathrm{CLIP}}$ (↑)	$S^{\mathrm{I}}_{\mathrm{CLIP}}$ (↑)	S_{DINO} (↑)
Baseline	0.2986	0.6812	0.6034
(a) Wrong pairing	0.3087	0.6778	0.5898
(b) Omitted semantic	0.3110	**0.7189**	0.6121
SEMANTICFLOW (correct)	**0.3295**	0.7061	**0.6208**

C.3 Scaling to More Concepts

As the number of semantic targets increases, SEMANTICFLOW adds parallel streams, each of which independently processes one target. The layer-wise energy functions of all semantics are summed to compute multi-concept perturbations. To empirically verify the generation behavior beyond the 2-concept cases, we conduct preliminary experiments with 3 and 4 concepts involving different combinations of target semantics. As shown in Table 3, the baselines exhibit lower textual alignment, indicating that they struggle to preserve the target semantics. In contrast, our method improves textual alignment, demonstrating that SEMANTICFLOW still achieves effective semantic injection, and its advantage over baselines is maintained. Due to the inherent fidelity–editability trade-off (see Appendix C.4), a decrease in visual alignment may occur in customized generation. Considering both text alignment and visual alignment, our method exhibits a better trade-off balance.

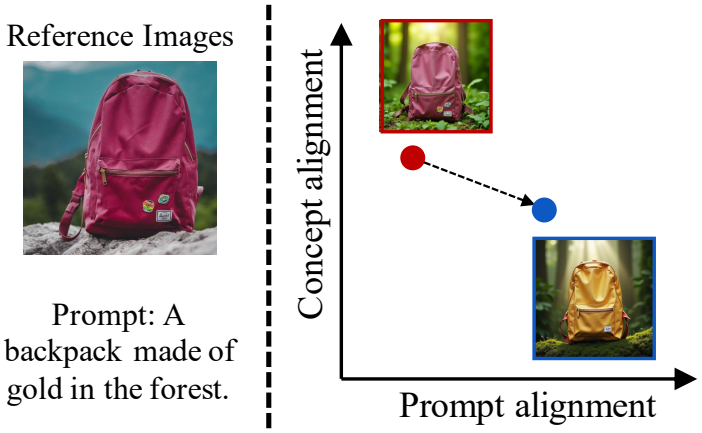


Figure 19: **Fidelity–editability Trade-off in Customization.** There is an inherent fidelity–editability trade-off in customized generation [11, 24, 18, 38]. This issue becomes particularly pronounced in tasks involving attribute modification of customized subjects or artist-style variations. In such scenarios, **faithfully adhering to the prompt and generating all instructed semantics** inevitably alters certain aspects of the subject's original visual appearance (*e.g.*, changing the material of a "backpack" when generating "a backpack made of gold"), leading to **a decrease in concept alignment**.

We note that as the number of concepts grows further, the overall generation quality is upper-bounded by the compositional capacity of the base model itself. Pre-trained models often struggle with lengthy and complex prompts that describe intricate scenes involving multiple objects. This issue arises from limited model capacity, stemming from factors such as the scale of pre-training, semantic competition, and spatial bottlenecks. This inherent limitation of pre-trained model capacity hinders the further scaling of concept numbers in multi-concept inference.

Table 3: **Generation with more concepts.**

Number of concepts	$S^{\mathrm{T}}_{\mathrm{CLIP}}$ (↑)	$S^{\mathrm{I}}_{\mathrm{CLIP}}$ (↑)	S_{DINO} (↑)
3 (Baseline)	0.3034	**0.6887**	**0.6073**
3 (SEMANTICFLOW)	**0.3242**	0.6820	0.5951
4 (Baseline)	0.2925	0.6245	0.5398
4 (SEMANTICFLOW)	**0.3189**	**0.6392**	**0.5656**

C.4 Trade-off in Customization

There is an inherent fidelity–editability trade-off in customized generation [11, 24, 18, 38]. This issue becomes particularly pronounced in tasks involving attribute modification of customized subjects or artistic style variations. In such scenarios, **faithfully adhering to the prompt and generating all instructed semantics** inevitably alters certain aspects of the subject's original visual appearance (*e.g.*, changing the material of a "backpack" when generating "a backpack made of gold"), leading to **a decrease in concept alignment**. For these tasks, it is essential to modify only the attributes specified in the prompt while preserving the remaining aspects of the subject's appearance (*e.g.*, modifying only the material of the "backpack" to "gold" while keeping the remaining features, such as its geometric structure, unchanged), thereby achieving semantically consistent customization. The semantic consistency in these cases leads to a balance between prompt alignment and concept alignment, as shown in Fig. 19.

C.5 Trade-off Analysis under Varying Control Strength

Because of the fidelity–editability trade-off (Appendix C.4), single-point comparisons may reflect different balancing choices rather than intrinsic differences between methods. We therefore compare

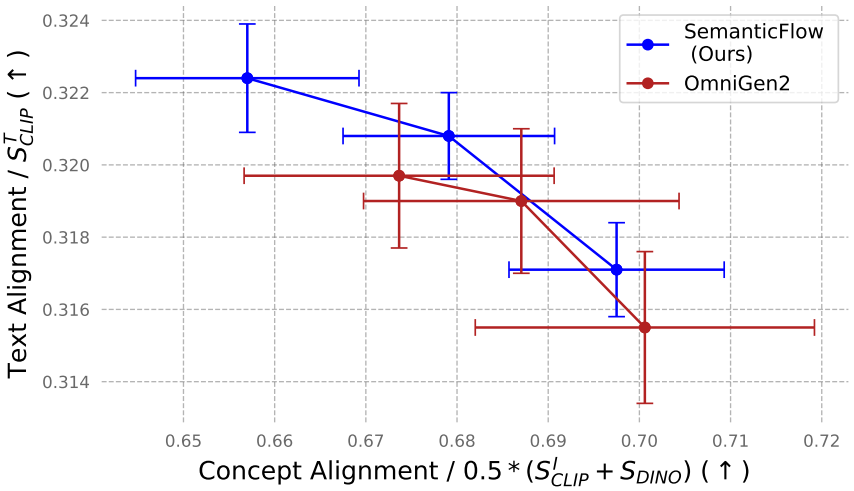


Figure 20: **Fidelity–editability trade-off.** We compare OmniGen2 and SEMANTICFLOW by varying their inference-time control parameters.

trade-off curves for methods that provide inference-time control over this trade-off. We inspect the official implementations of five baselines: OminiControl, UNO, XVerse, DreamO, and OmniGen2. OminiControl provides no such parameter. The adjustable parameters of UNO, XVerse, and DreamO govern other trade-offs, such as subject consistency or prompt adherence versus image naturalness. OmniGen2 introduces an `image_guidance_scale` parameter that directly scales the difference between the reference-conditioned and unconditional predictions, controlling the contribution of reference image features to the final output. This parameter offers a degree of control over the fidelity–editability trade-off.

We vary `image_guidance_scale` in OmniGen2 across 1.5, 2.0, and 3.0, and multiply η_t in SEMANTICFLOW by factors of 0.5, 1, and 2. As shown in Fig. 20, increasing `image_guidance_scale` improves fidelity to the reference image but reduces text alignment, whereas decreasing it has the opposite effect. The SEMANTICFLOW curve lies above that of OmniGen2. OmniGen2 offers relatively coarse control: a single scalar uniformly scales the entire reference-conditioned prediction, creating a direct competition between concept fidelity and text alignment. SEMANTICFLOW enables fine-grained control through spatially aware velocity-field rectification, which can improve both concept fidelity and text alignment through semantic control. Furthermore, the adjustment capability of `image_guidance_scale` is limited.

C.6 Comparison with More Tuning-Free Methods

We compare SEMANTICFLOW with several previous tuning-free methods, including SSR-Encoder [69], FreeGraftor [67], and MS-Diffusion [59], using their official implementations. SSR-Encoder and MS-Diffusion are based on the U-Net architecture. The qualitative comparison results for single-concept and multi-concept customized generation are shown in Fig. 21 and Fig. 22, respectively. As shown, the primary drawback of these tuning-free methods is their difficulty in preserving the key identifying and fine-grained features of customized concepts. The generated concepts often exhibit low subject fidelity, especially for complex subjects. Moreover, in challenging scenarios such as attribute modification and artistic style variations, the baselines struggle to accurately follow the textual prompts. They also face difficulties in correctly binding attributes for multiple subjects.

C.7 More Comparison with Editing Methods

We further compare our method with several recent image editing approaches, including ReFlex [21], FlowEdit [23], and IC-Edit [70], using their official implementations. The editing methods are applied to the outputs of the customization baselines, with editing instructions designed to inject the semantics that were omitted in the baseline outputs. We present the qualitative results for single-concept and multi-concept customized generation in Fig. 23 and Fig. 24, respectively. As observed, the editing

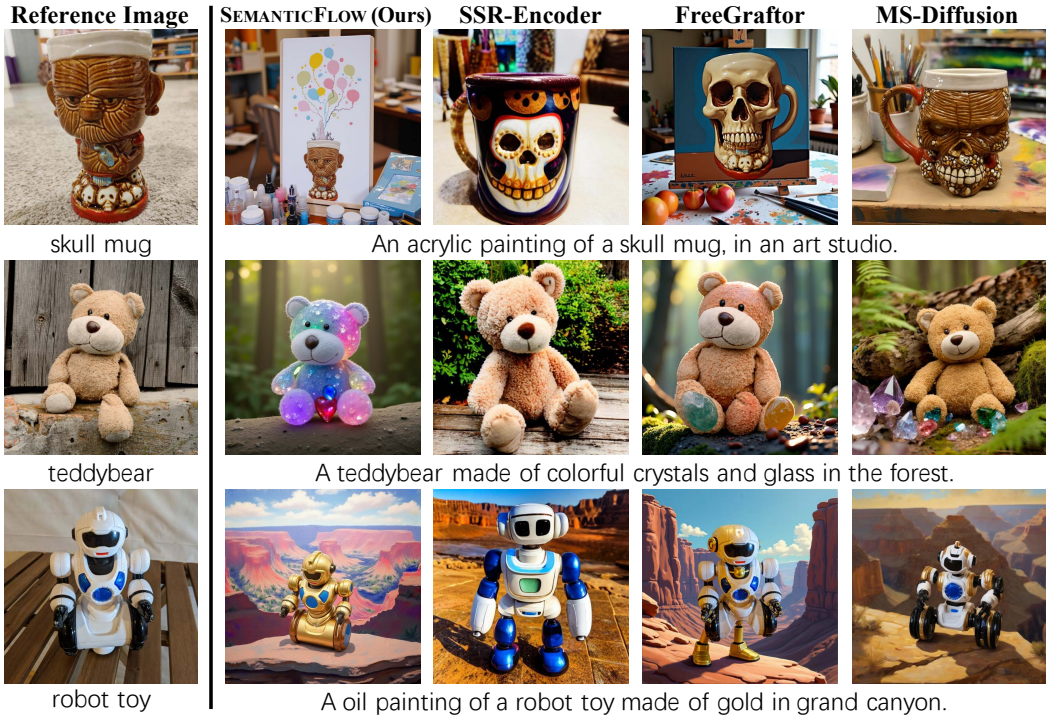


Figure 21: **Single-concept qualitative comparison with more tuning-free methods.** The concepts generated by baselines often exhibit low subject fidelity, especially for complex subjects. In scenarios such as attribute modification and artistic style variations, the baselines struggle to accurately follow the textual prompts.

methods are ineffective for large-scale modifications. Most importantly, these methods are unable to enhance concept fidelity in the edited images and typically further distort the key identifying features of the customized concepts during the editing process. Finally, the editing methods exhibit attribute fusion in some cases involving multiple subjects.

C.8 Comparison with Tuning-based Methods

Tuning-based customized generation adapts a large, pre-trained text-to-image model to customized visual concepts provided by the user. This is achieved by fine-tuning the model's weights using a small set of reference images. This optimization process embeds the new concept deep within the model. We compare SEMANTICFLOW with several representative tuning-based customized generation methods, including DreamBooth [51], Custom Diffusion [24], LoRA [16] (with rank $r = 4$), and DreamMatcher [36]. We present sample generations in Fig. 25. As observed, tuning-based methods typically exhibit high concept fidelity due to their concept-specific fine-tuning and deep conceptual integration. However, these methods often sacrifice editability on the customized subjects, making it difficult to accurately render these subjects within the query context.

C.9 More Results of Visual Comparison

We provide additional visual comparisons between the proposed SEMANTICFLOW and the baseline methods, including OminiControl [57], UNO [65], DreamO [35], XVerse [3], and OmniGen2 [64], as shown in Fig. 31, Fig. 32, Fig. 33, and Fig. 34. As we can see, the images generated by the baselines typically omit one or more types of contextual content specified by the textual prompts, such as the concept property and background scene. In some generation cases, the customized concepts exhibit low concept fidelity.



Figure 22: **Multi-concept qualitative comparison with more tuning-free methods.** The baselines generally suffer from poor concept alignment and weak prompt adherence. They also struggle to correctly bind attributes when multiple subjects are present.

C.10 Memory Usage and Computation Cost

SEMANTICFLOW introduces lightweight semantic token streams. These semantic streams repurpose the model weights of the textual token stream and require no additional training. We truncate the padding tokens in the input token sequence of the semantic stream, resulting in a short sequence length (typically $1 \sim 3$). Furthermore, we apply gradient checkpointing to the calculation of velocity adjustment. We use UNO as the baseline to analyze memory usage and computation cost. Without cost optimization, our method incurs a peak GPU memory usage of 42 GB for one subject–attribute pair, compared to 37 GB for the baseline. Peak memory increases by approximately 5 GB for each additional subject–attribute pair. As discussed in Sec. 4.3 of the main paper, SEMANTICFLOW performs velocity adjustment only during the first 4 steps of the 25-step denoising process, leading to a small additional time cost. SEMANTICFLOW requires 28 seconds to generate an image on a single NVIDIA RTX A6000 GPU, whereas the baseline takes 23 seconds.

As discussed in the Introduction of the main paper, our extensive investigation reveals that leading tuning-free approaches fail dramatically in some scenarios and exhibit semantic inconsistency, particularly in attribute modification and artistic style variation. Existing methods struggle in these cases and may require repeated attempts to obtain satisfactory results, incurring higher computational costs.

C.11 Limitations

SEMANTICFLOW has limitations in certain generation cases. We list several typical failure cases and present sample generations in Fig. 26. For reference, we also include the outputs of pre-trained text-to-image model (FLUX) on these cases. Some contexts rarely occur in real data. Such cases are



Figure 23: **Single-concept qualitative comparison with editing methods.** The editing methods are ineffective for large-scale modifications. Most importantly, these methods are unable to enhance concept fidelity in the edited images and typically further distort the key identifying features of the customized concepts during the editing process.

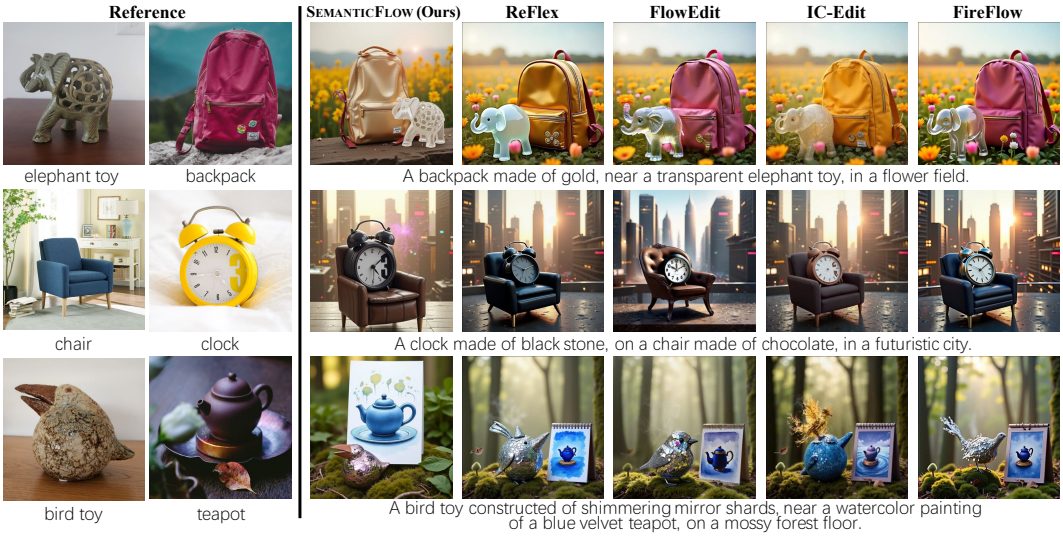


Figure 24: **Multi-concept qualitative comparison with editing methods.** The editing methods are unable to enhance concept fidelity in the edited images and typically further distort the key identifying features of the customized concepts during the editing process.

underrepresented in large-scale training datasets and thus have weak generative priors. Additionally, some prompts are counterfactual or contain semantic conflicts, such as the second case involving contradictory object materials. In these situations, the target semantics are inherently difficult to inject. The pre-trained model also struggles with these generation tasks.



Figure 25: **Qualitative comparison with tuning-based methods.** Tuning-based methods typically exhibit high concept fidelity but show poor prompt alignment.

D Implementation Details

D.1 Subjects and Prompts

Experiments are conducted on DreamBench [51] (30 subjects), TI Dataset [11] (9 subjects), and CustomConcept [24] (9 subjects), which are among the most widely used datasets for customized generation. These datasets encompass a diverse range of personalized concepts, each represented by several reference images. The evaluation prompts are also drawn from these datasets, covering various generation types such as recontextualization, attribute modification, and style transfer. As for multi-concept customized generation, the subject combinations and corresponding query prompts are automatically generated and subsequently reviewed and curated manually. Specifically, we create a `concept list` containing the subjects mentioned above, then generate 2-concept subject combinations and their query prompts with the following several steps: 1. Select 2 subjects from the `concept list`. 2. Use the selected 2 nouns to construct sentences based on the following requirements: 1) Apart from the 2 selected nouns, avoid using other nouns from the concept list in the sentence. 2) Ensure the sentences are logical. 3) Maintain diversity in the sentence structures. Some multi-concept prompts are also inspired by other previous works [51, 24].

D.2 Additional Details on User Study

Details of the Experiment in Fig. 2 of the Main Paper. The experiment in Fig. 2 of the main paper is conducted on DreamBench [51] with 30 concepts, the TI Dataset [11] with 9 concepts, and CustomConcept [24] with 9 concepts. The evaluation prompts are also drawn from these datasets. Specifically, we collect a total of 144 prompts for recontextualization, 49 prompts for attribute modification, and 39 prompts for artistic style variation in our investigation. We evaluate three leading tuning-free customized generation methods [65, 35, 3] that are based on rectified flow transformers. All images are generated using the same initial seed. We collect the images generated by these methods and evaluate them by users. In each evaluation case, users are given a textual prompt, reference images of the customized subject, and the corresponding generation. The users are asked to evaluate whether the generation is satisfactory—that is, whether it is semantically consistent with both the reference images and the textual prompt. Generations that are semantically consistent and exhibit satisfactory concept fidelity and prompt alignment are marked as “success”. We provide examples of "successful" and "unsuccessful" generations in Fig. 27 for reference. The "unsuccessful"



Figure 26: **Failure cases.** Some contexts rarely occur in real data, and some are counterfactual or contain semantic conflicts. The pre-trained model (*i.e.*, FLUX) often fails on such tasks. In these situations, the target semantics are difficult to inject.

cases exhibit poor concept fidelity and prompt alignment. Finally, we collect the evaluation results and report the success rate of each method for each subject across different generation types using violin plots.

User Study. We conduct a user study to further evaluate SEMANTICFLOW. Before the evaluation, we use SEMANTICFLOW and the baseline methods to generate images of various customized concepts and different types of query prompts, creating an image candidate pool. A total of 29 participants take part in the user study. We provide detailed instructions containing scoring rules and examples for participants, with a screenshot shown in Fig. 29. Participants are thoroughly briefed on these instructions before the user study and can also access them during the evaluation. In each comparison, participants are shown reference images of the target subject and a query prompt, along with the corresponding images generated by each competing method. The generated images are randomly selected from the image candidate pool and presented side-by-side in a random order to the participants. Each generation is scored by participants on a scale from 0 to 5 (with 0 being the worst and 5 the best) based on the following two criteria: 1) Concept alignment: how well the key identifying features of the customized subject in the generated image align with the target subject in the reference images, 2) Prompt alignment: how closely the image content in the generated images adheres to the context described by the textual prompt. We provide a screenshot of an evaluation case in Fig. 30. We present the aggregated results in Fig.28. Our method demonstrates a greater user preference for semantically-consistent customization.

D.3 Implementation Details

Our implementation of SEMANTICFLOW uses FLUX.1-dev as its pretrained text-to-image backbone. For the baseline comparisons, we use the official implementations and their default inference configu-

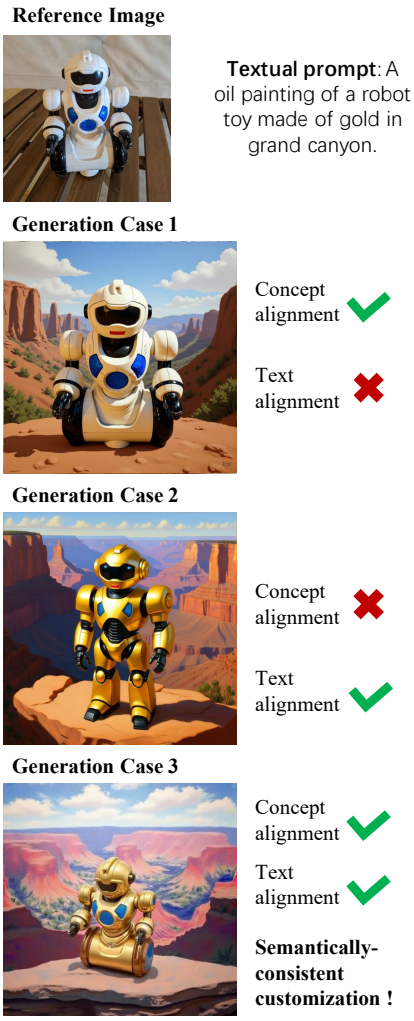


Figure 27: **An illustration of the human evaluation criteria in Fig. 2 of the main paper.** Case 1 preserves the key identifying features of the customized subject but exhibits low text alignment, *i.e.*, it fails to modify the subject's attribute to "golden." Case 2 aligns with the textual prompt but loses the key features of the customized subject. Case 3 both maintains the identifying features of the customized subject and faithfully adheres to the textual prompt (i.e., modifying the subject's attribute to "golden" while preserving its geometric structure), thereby achieving semantically consistent customization. Therefore, Case 3 is considered "successful," whereas Case 1 and Case 2 are "unsuccessful."

rations. MS-Diffusion uses SDXL as its backbone, and we evaluate OmniGen2 using its officially released checkpoint, which includes a separately trained diffusion transformer.

We compare SEMANTICFLOW with state-of-the-art tuning-free methods, including MS-Diffusion [59], OminiControl [57], UNO [65], DreamO [35], XVerse [3], and OmniGen2 [64]. We also compare our method with editing methods, including ReFlex [21], FlowEdit [23], and IC-Edit [70]. We use the official implementations for these baseline methods. All baselines perform inference using the default hyperparameters and configurations provided in their official code. We use UNO [65] as the baseline by default. SEMANTICFLOW can also be combined with methods that are based on the rectified flow transformer, such as DreamO [35]. In all of our experiments, we perform four adjustment steps in the early denoising stage, *i.e.*, we set $N_a = 4$. We use the same η_t for each type of semantic manipulation across all our experiments. Specifically, we set $\eta_t = 30$ for attribute modification and background enhancement, $\eta_t = 10$ for artistic style variation, and $\eta_t = 15$ for

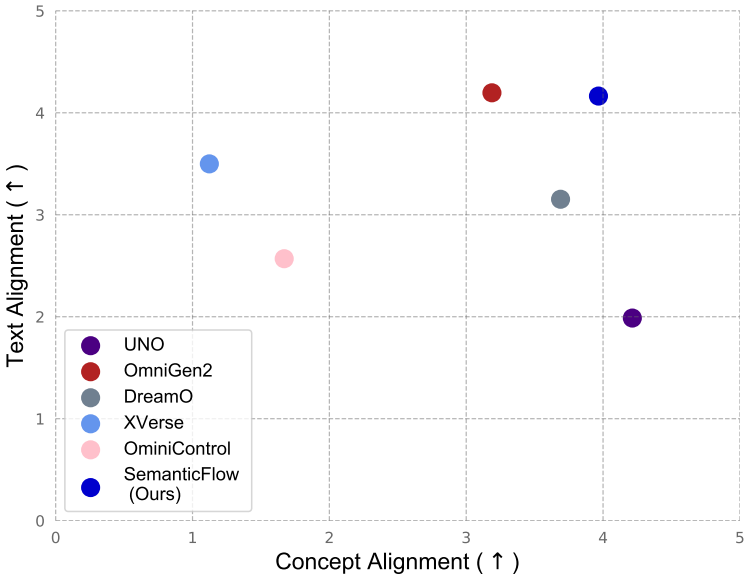
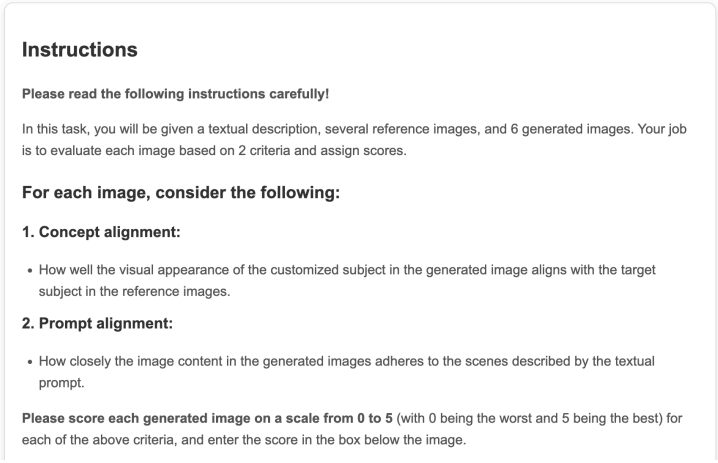


Figure 28: **User Study.** SEMANTICFLOW demonstrates a strong user preference for semantically-consistent customization.



Instructions

Please read the following instructions carefully!

In this task, you will be given a textual description, several reference images, and 6 generated images. Your job is to evaluate each image based on 2 criteria and assign scores.

For each image, consider the following:

1. Concept alignment:

- How well the visual appearance of the customized subject in the generated image aligns with the target subject in the reference images.

2. Prompt alignment:

- How closely the image content in the generated images adheres to the scenes described by the textual prompt.

Please score each generated image on a scale from 0 to 5 (with 0 being the worst and 5 being the best) for each of the above criteria, and enter the score in the box below the image.

Figure 29: **The instructions that were given to the participants.**

enhancement of concept fidelity. More detailed analysis of these parameters is provided in Sec. C.1 and Sec. 4.3 of the main paper.

E Impact Statement

This paper presents work whose goal is to advance the field of Customized Text-to-image Generation, offering greater flexibility and personalization for customized content creation. However, the technology's potential for misuse—such as producing deceptive or harmful content—could raise ethical issues. Recent research on safeguards [33] provides a promising approach to mitigating these potential negative impacts.

F Licenses for Pre-trained Models and Datasets

Our implementation is based on the publicly available FLUX.1-dev model [25], which is released under a non-commercial license. The datasets used for evaluation are from DB [51], TI [11], and

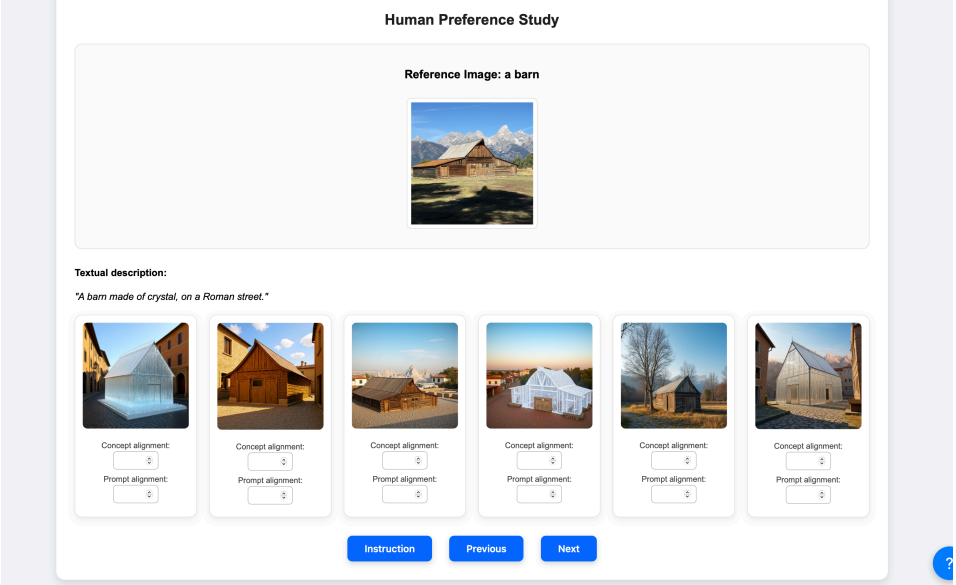


Figure 30: **A screenshot of an evaluation case from the user study.**

custom diffusion [24]. The data from DB and custom diffusion is under the Unsplash license, while the license information for the data from TI is not available online.

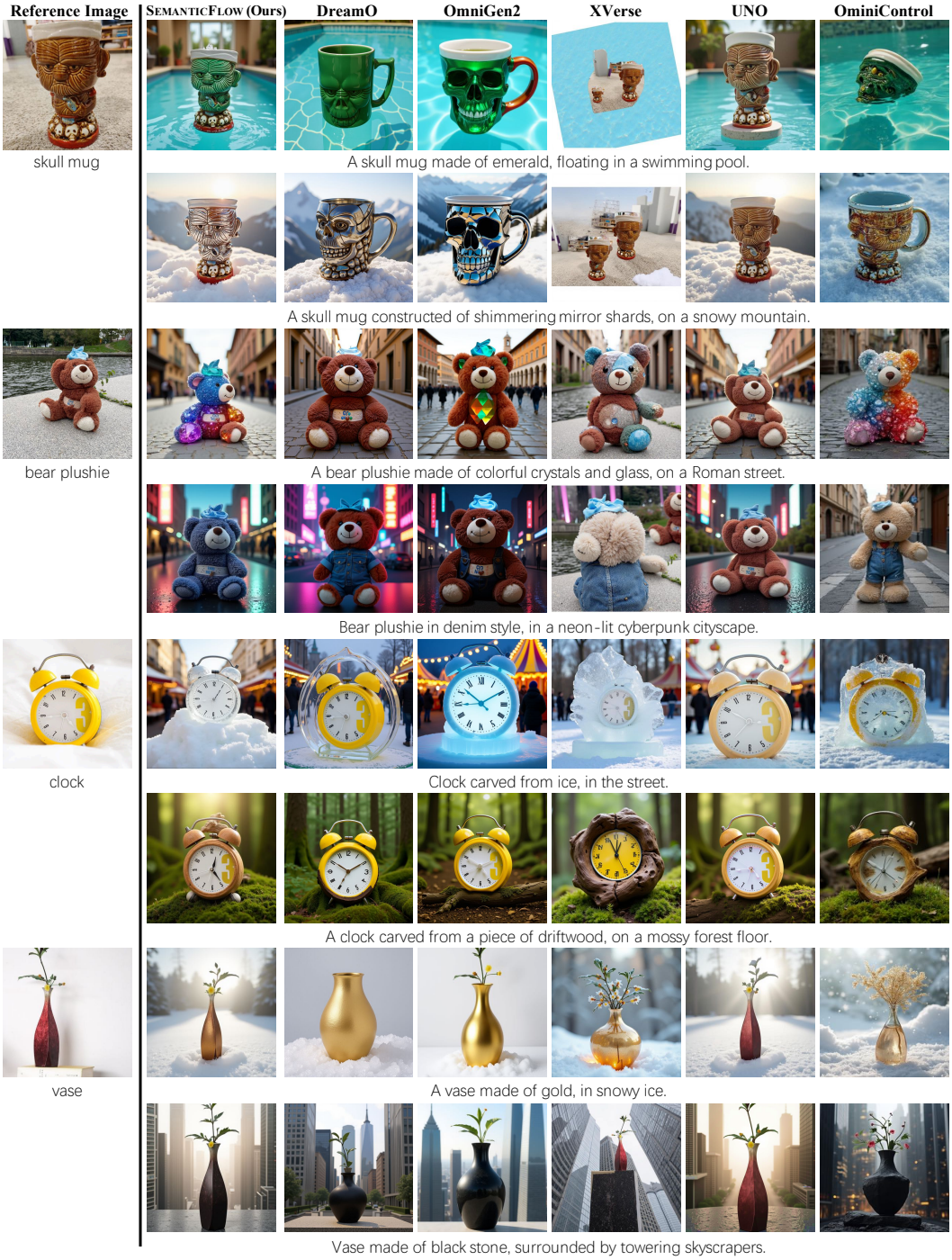


Figure 31: **More Results of Visual Comparison.**

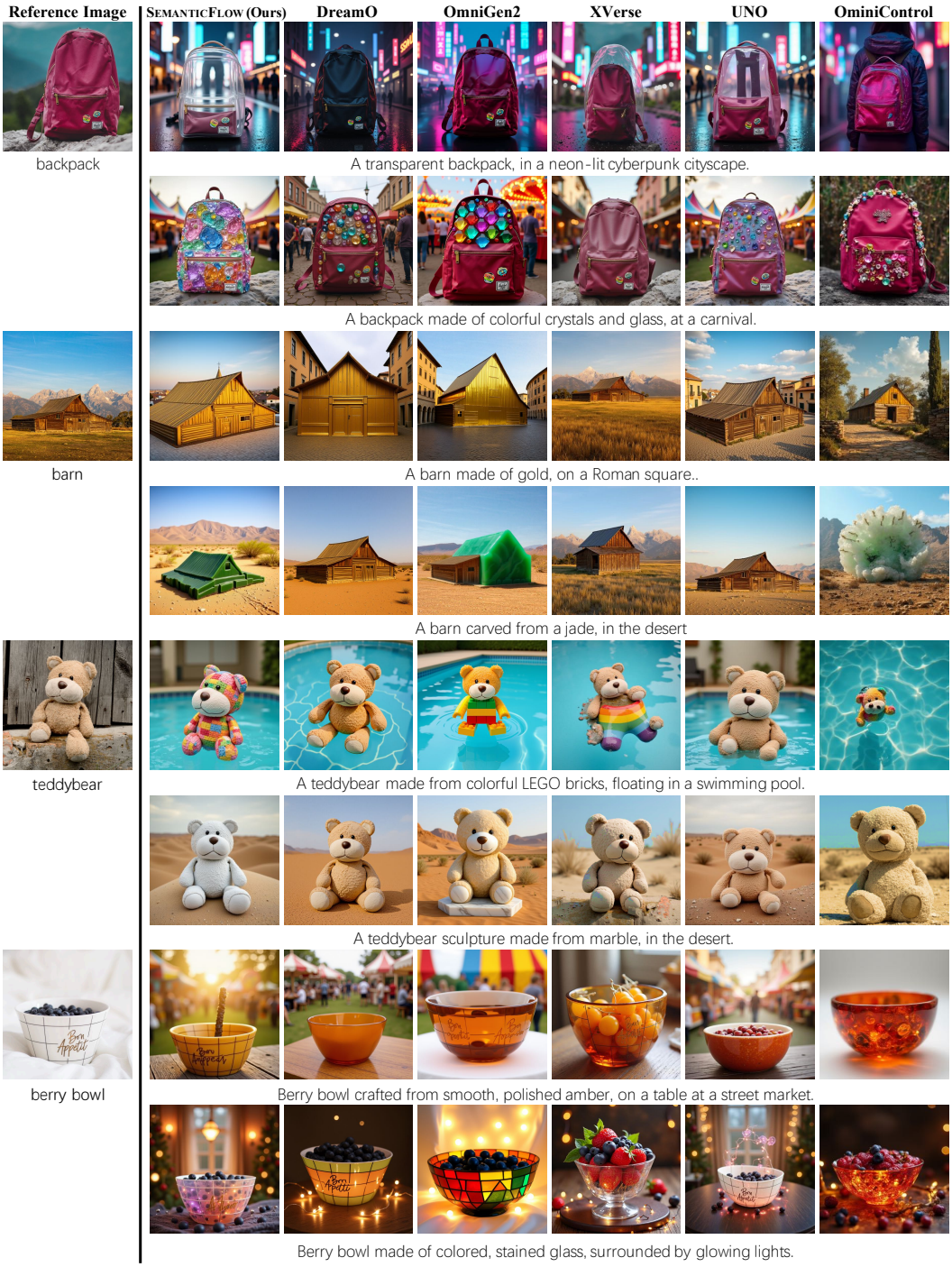


Figure 32: **More Results of Visual Comparison.**

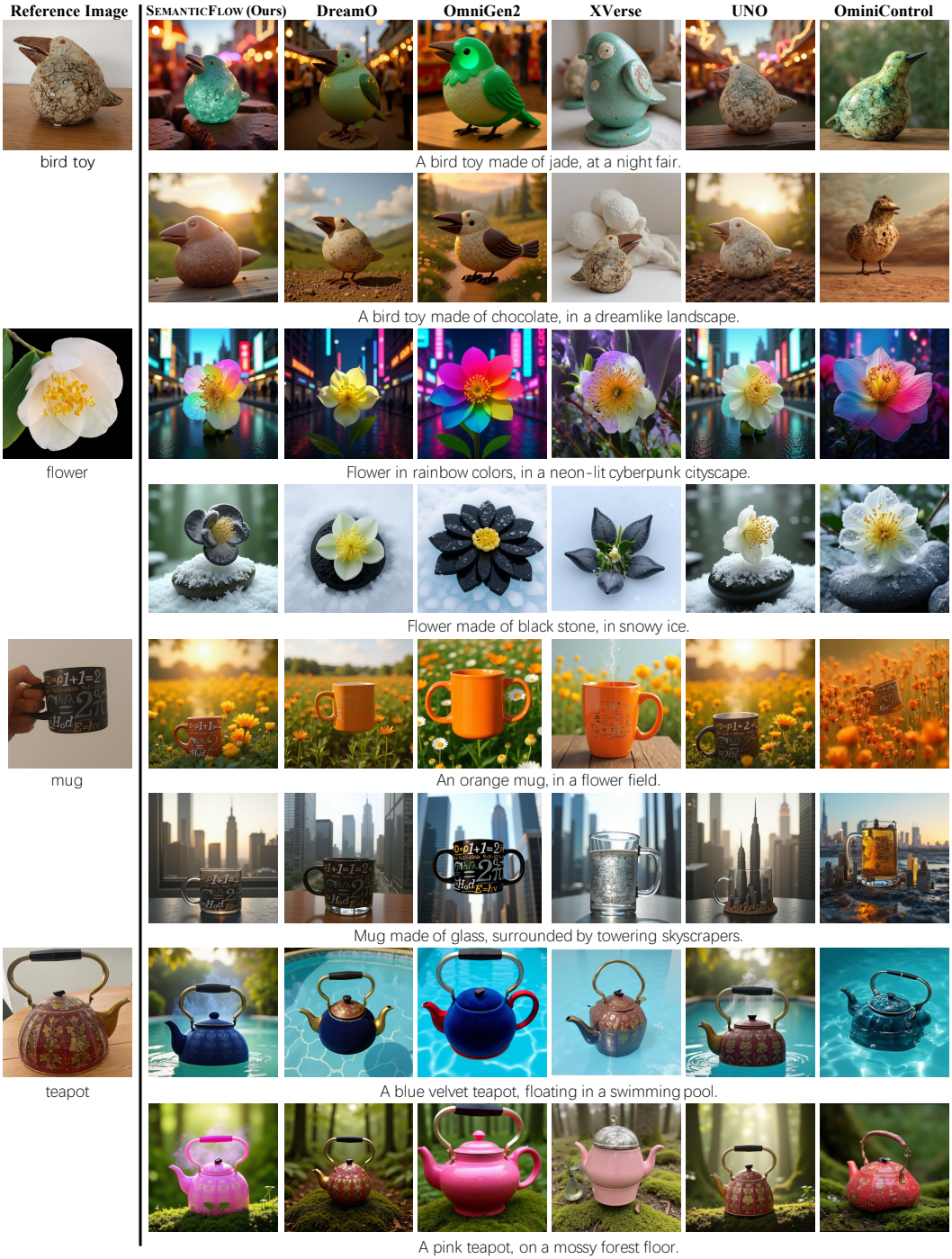


Figure 33: **More Results of Visual Comparison.**

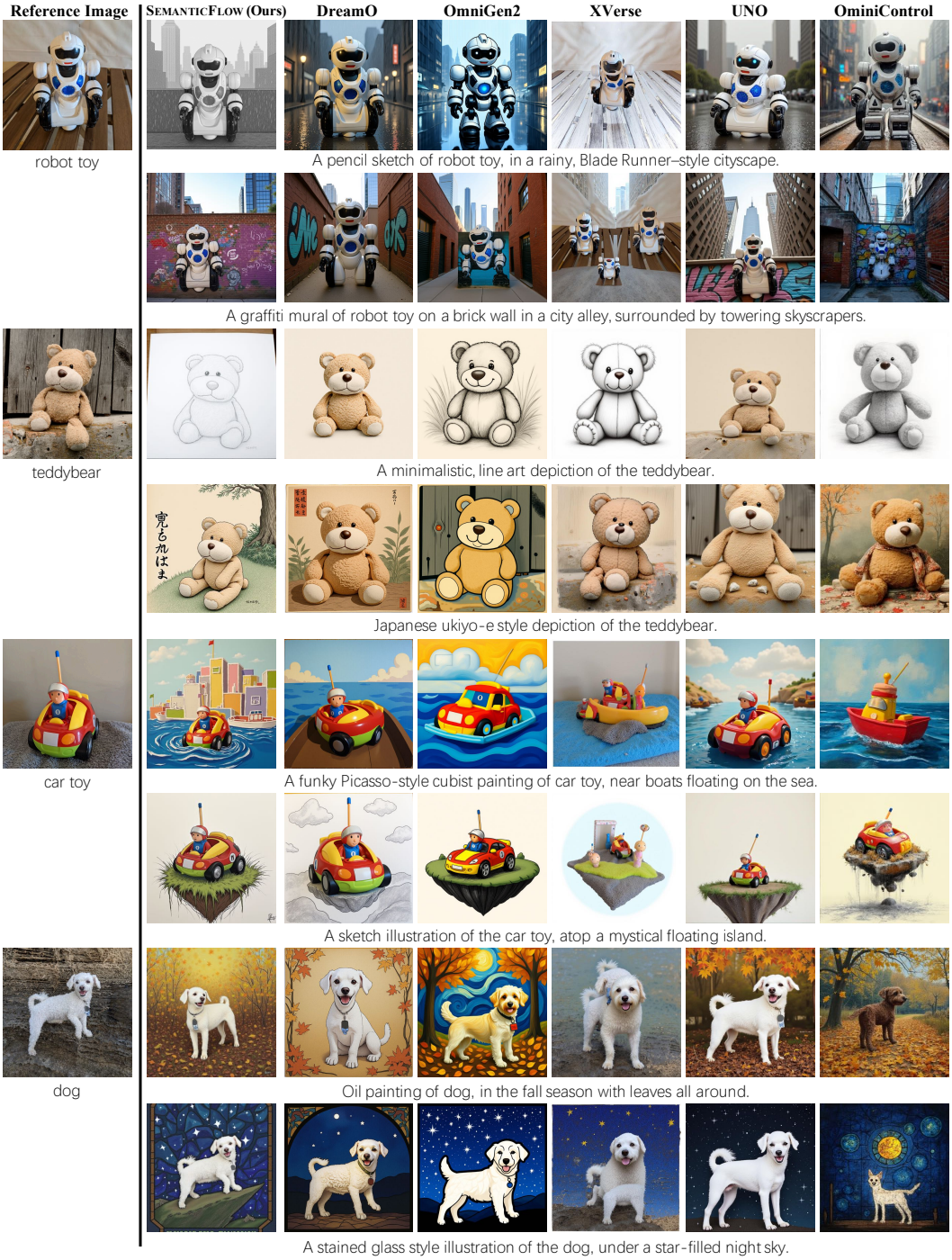


Figure 34: **More Results of Visual Comparison.**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations in Appendix C.11.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete (and correct) proof in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper fully discloses all information needed to reproduce the main experimental results, as detailed in the experimental setup and appendix sections.

   Guidelines:

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

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

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

Answer: [No]

Justification: We have made a detailed description of the implementation details to ensure that they are repeatable and use publicly available datasets. We will make our code publicly available following the paper's acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Please see Appendix D.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: This paper reports error bars in the main experiments.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Please see Appendix D.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

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

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have discussed the potential impacts of our work in Appendix E.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [No]

    Justification: We will personally evaluate the risks and decide on the publication of the model files.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have discussed the licenses of existing assets in Appendix F.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: We have provided the full text of instructions given to participants and screenshots in Appendix D.2.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: All human evaluations were conducted in accordance with the host institution's ethical guidelines.

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

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

    Guidelines:

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