Panoptic Scene Program Diffusion Transformer

Chika Maduabuchi
William & Mary

Abstract

Modern text-to-image models produce high-fidelity images but still struggle with compositional prompts that require instance identity, attribute ownership, counting, spatial ordering, and role-sensitive relations. We introduce *Panoptic Scene Program Diffusion Transformer* (PSP-DiT), a diffusion-transformer architecture that treats a panoptic scene program as a first-class latent variable rather than an external control signal or post-hoc parse. PSP-DiT jointly denoises image latents and scene-program latents through coupled transformer streams, while panoptic grounding and cycle-consistency objectives tie object instances, attributes, relations, and counts to visual support in the generated image. Under matched training and inference settings, PSP-DiT improves over a strong flat-text baseline across GenEval 2, SANEval-Simple, PSG-Score, and DetailMaster, with the largest gains on counting, attribute binding, role-sensitive relations, and long structured prompts. The method preserves image quality, adds modest inference overhead, and remains robust to imperfect scene programs.

1 Introduction

Text-to-image generation has advanced at extraordinary speed, evolving from latent diffusion and large language-conditioned cascades to diffusion transformers and rectified-flow systems that now deliver strong visual fidelity, prompt adherence, and typography at unprecedented scale [Rombach et al., 2022, Saharia et al., 2022, Betker et al., 2023, Peebles and Xie, 2023, Esser et al., 2024, Xie et al., 2025, Tong et al., 2026]. Yet the central scientific bottleneck has shifted rather than disappeared: the hardest failures are no longer low-level realism, but compositional faithfulness under multi-entity, detail-heavy prompts. Across benchmarks spanning object co-occurrence, counting, attribute binding, spatial relations, long prompts, spatial intelligence, and reasoning, state-of-the-art systems still degrade precisely when they must keep track of instance identity, ownership, ordering, and interaction structure [Ghosh et al., 2023, Huang et al., 2023, Kamath et al., 2025, Pramanik et al., 2026, Jiao et al., 2025, Wang et al., 2026b, Chen et al., 2025a, Zhang et al., 2025a, Gao et al., 2026]. This pattern suggests that the remaining barrier is not only stronger text encoders or more data, but the absence of an explicit scene state inside the generative process itself.

A broad line of work has tried to close this gap through controllable or structured generation. Training-time and inference-time guidance methods such as GLIGEN, ControlNet, Attend-and-Excite, and BoxDiff improve controllability by injecting boxes, masks, or attention constraints into pretrained diffusion models [Li et al., 2023, Zhang et al., 2023, Chefer et al., 2023, Xie et al., 2023]. Scene-graph and structure-centric approaches further demonstrate the value of relational representations for generation, evaluation, and data synthesis [Xu et al., 2025, Wang et al., 2024b, Chen et al., 2025b, Gao et al., 2026, Jayasekara et al., 2026]. In parallel, panoptic scene graph research has shown that instance-resolved, region-grounded relational representations are substantially richer than box-level graphs for visual understanding [Yang et al., 2023, Zhao et al., 2023, Zhou et al., 2025]. However, these directions remain fundamentally separated: control methods treat structure as an external condition, scene-graph methods typically factor graph generation from image generation or translate graphs into layouts and semantics used only as guidance, and panoptic scene graphs are studied

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

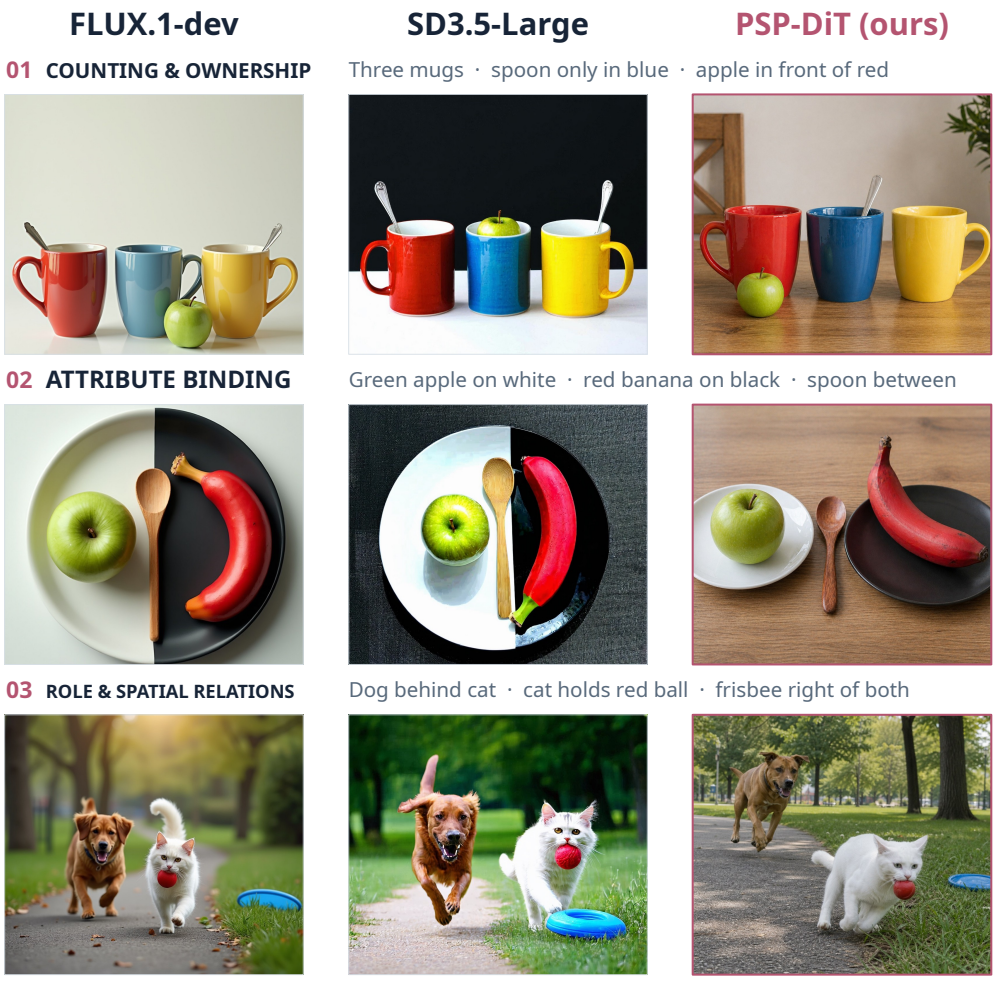


Figure 1: Qualitative comparisons of FLUX.1-dev, SD3.5-Large, and PSP-DiT on three shared compositional prompts. The rows illustrate counting and ownership, attribute binding, and role-sensitive spatial relations. Additional annotated comparisons are discussed in Section 4.9.

primarily for recognition rather than as first-class latent variables in modern text-to-image generators. We target this missing middle. Our method, *Panoptic Scene Program Diffusion Transformer* (PSP-DiT), makes an instance-indexed panoptic scene program a jointly denoised latent companion to the image stream, so that objects, attributes, relations, counts, and visual ownership are represented and refined throughout generation rather than inferred only after the image has already been produced.

Contributions. Our key contributions are:

- We propose *PSP-DiT*, a joint generative formulation $p_\theta(z, s \mid y)$ in which a panoptic scene program s is a first-class latent variable rather than an auxiliary control signal or a post-hoc parse.
- We design a coupled diffusion-transformer architecture that jointly refines image tokens and scene-program tokens, enabling explicit instance indexing, attribute ownership, ordered relations, and cardinality tracking inside the denoising trajectory.
- We introduce panoptic grounding and cycle-consistency objectives that require the generated image to remain explainable by the intended scene program, connecting structural faithfulness to spatial support instead of relying only on holistic alignment metrics.

- We position PSP-DiT for the failure modes that matter in current evaluation—counting, role-sensitive relations, multi-object attribute binding, and long structured prompts—making the method directly responsive to the shortcomings highlighted by the newest generation of compositional T2I benchmarks [Kamath et al., 2025, Pramanik et al., 2026, Jiao et al., 2025, Wang et al., 2026b, Chen et al., 2025a, Zhang et al., 2025a].

2 Related Works

Compositional evaluation and failure diagnosis. A large body of work has shown that strong perceptual quality in text-to-image (T2I) synthesis does not imply reliable compositional faithfulness. Benchmarks such as TIFA, GenEval, T2I-CompBench, VQAScore, and GenAI-Bench exposed persistent failures in counting, attribute binding, relative positioning, and relation understanding even in visually compelling samples [Hu et al., 2023, Ghosh et al., 2023, Huang et al., 2023, Lin et al., 2025, Li et al., 2024a]. Recent benchmarks have made this gap sharper rather than smaller: T2I-CompBench++ broadens compositional coverage, PSGEval emphasizes panoptic-scene-graph consistency, and recent analyses argue that current metrics still miss important aspects of structured grounding [Huang et al., 2025, Deng et al., 2025, Kasaei et al., 2025]. The newest evaluations further confirm that the problem remains unresolved in modern generators: GenEval 2 reports substantial benchmark drift and harder compositional stress tests, SANEval targets open-vocabulary spatial and attribute reasoning, DetailMaster studies long and detail-rich prompts, and *Everything in Its Place* shows that higher-order spatial composition is still a major failure mode [Kamath et al., 2025, Pramanik et al., 2026, Jiao et al., 2025, Wang et al., 2026b]. These works are essential, but they are primarily diagnostic: they score compositional errors after generation. Our method is motivated by the complementary question of representation—how to endow the generator with an explicit latent scene state in which object identity, ownership, count, and relations are represented during denoising rather than only measured afterward.

Inference-time guidance, alignment, and retrofit compositional fixes. A second line of work attempts to improve compositionality by steering pretrained diffusion models at inference time or by fine-tuning them with auxiliary objectives. Representative examples include compositional and region-wise guidance, cross-attention editing, grounded generation, box/layout guidance, and multi-crop or dense-attention fusion [Liu et al., 2022, Chefer et al., 2023, Li et al., 2023, Xie et al., 2023, Bar-Tal et al., 2023, Kim et al., 2023, Chen et al., 2024, Liu et al., 2024, Chen et al., 2026, Zhao et al., 2024, Wang et al., 2024a, Feng et al., 2023]. A related thread improves specific failure modes such as semantic binding or relation fidelity using VLM feedback, enriched text embeddings, syntax-aware attention, or seed/prompt optimization [Wen et al., 2023, Zarei et al., 2025, Trusca et al., 2024, Li et al., 2025b, Zhang et al., 2025b, Han et al., 2025, Li et al., 2026a, Jiang et al., 2024]. More recent alignment-based approaches adapt preference optimization, reward learning, or reinforcement learning to diffusion models, including Diffusion-DPO, Bi-DPO, DATE, InfSplign, LOOP, RePrompt, HiCoGen, and MCCD [Wallace et al., 2024, Pham et al., 2026, Xie and Gong, 2025, Rastegar et al., 2025, Gupta et al., 2026, Wu et al., 2026, Yang et al., 2025, Li et al., 2025a].

Distinction from recent approaches. CompGen [Wang et al., 2026a] uses scene graphs to construct difficulty-aware reinforcement-learning curricula, while Prompt Reinjection [Yao et al., 2026] mitigates semantic forgetting by reintroducing early text representations. GEMS [He et al., 2026] improves generation through an external iterative agent loop. In contrast, PSP-DiT jointly denoises image and instance-indexed scene-program latents with bidirectional refinement and panoptic grounding, without test-time search. These approaches are complementary rather than direct architectural equivalents.

Structured scene representations for controllable generation. The closest prior work introduces stronger structure into the conditioning signal, for example through scene graphs, layouts, blobs, layered decompositions, or synthetic structure-rich supervision. Scene-graph-based generators and disentanglement methods show that explicit object-relation structure can improve controllability beyond plain text [Zhang et al., 2025c, Wang et al., 2024b]. Dense Blob Representations and related layout-generation pipelines move further toward object-centric intermediate structure, while Generate Any Scene, LAION-Comp, and All-in-One Conditioning highlight the growing interest in richer structural annotations and unified controls for T2I synthesis [Nie et al., 2024, Liu et al., 2025, Gao et al., 2026, Li et al., 2026b, Jayasekara et al., 2026]. Other work explores instance-layer or

document-style decomposition, including RGBA layered generation, paragraph-to-image generation, and compositional prompt decomposition for long text [Fontanella et al., 2024, Wu et al., 2025, Huang et al., 2026]. In parallel, panoptic scene graph research has developed stronger object-relation representations and supervision sources that better capture "things," "stuff," and their interactions [Yang et al., 2023, Zhao et al., 2023]. These directions are highly relevant, but they typically use structure as external conditioning, preprocessing, supervision, or dataset annotation. PSP-DiT goes a step further by making the panoptic scene program a first-class latent variable of generation: it is sampled jointly with the image, maintained throughout denoising, and tied back to the image through explicit scene decoding and cycle consistency. This design directly targets the identity-binding mismatch that persists when structure is supplied only as side information.

Stronger backbones do not by themselves solve composition. Finally, recent progress in T2I quality has been driven by stronger generative backbones, latent spaces, and large-scale training, including DALL-E 2, Imagen, eDiff-I, latent diffusion, SDXL, Diffusion Transformers, and, more recently, representation autoencoders for DiT-style models [Ramesh et al., 2022, Saharia et al., 2022, Balaji et al., 2023, Rombach et al., 2022, Podell et al., 2024, Peebles and Xie, 2023, Zheng et al., 2026]. Yet the latest compositional evaluations continue to show that scaling image quality and scaling compositional reasoning are not the same problem [Ghosh et al., 2023, Huang et al., 2023, Kamath et al., 2025, Pramanik et al., 2026, Jiao et al., 2025, Wang et al., 2026b]. Our work is therefore orthogonal to backbone scaling: we instantiate PSP-DiT on a modern DiT+RAE stack, but the core contribution is a new generative primitive—a jointly denoised, panoptically grounded scene-program latent—that explicitly addresses the structured reasoning bottleneck left open by backbone advances alone.

3 Method

PSP-DiT augments text-to-image generation with a panoptic scene-program latent that is denoised jointly with the image latent. Section 3.1 defines the joint modeling goal and why flat text conditioning is insufficient; Section 3.2 introduces the scene-program latent; Section 3.3 describes the coupled image–program denoiser; Section 3.4 adds panoptic grounding and ownership constraints; Sections 3.5–3.6 give the training objective and inference procedure.

3.1 Problem Setup and Design Goal

Let y be a text prompt, $x \in \mathbb{R}^{H\times W\times 3}$ an image, and $z \in \mathcal{Z}$ its RAE latent. Standard text-to-image models generate $p_\theta(z \mid y)$ through flat prompt-token conditioning, which leaves instance identity, attribute ownership, ordered relations, cardinality, and visual support implicit. PSP-DiT instead introduces a structured scene latent $s \in \mathcal{S}$ and models the joint conditional state $p_\theta(z, s \mid y)$, where s encodes instances, attributes, relations, counts, and scene context. Thus a prompt such as "a green apple and a red banana on a wooden table" is treated as a structured scene to be generated, not only as a bag of text tokens. As shown in Figure 2, PSP-DiT jointly denoises the image and scene-program latents through bidirectional cross-attention.

This design imposes three constraints: instance indexing, joint refinement of (z_t, s_t), and grounding of s_t in visual ownership. Section 3.2 defines the scene program, Section 3.3 gives the coupled denoiser, and Section 3.4 enforces visual accountability.

3.2 Panoptic Scene Programs as Latent Variables

The panoptic scene program is the scene-state component of the joint model. We write $s = (\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{C}, \mathcal{B})$, where $\mathcal{O}$ are object-instance variables, $\mathcal{A}$ instance-attached attributes, $\mathcal{R}$ ordered relations, $\mathcal{C}$ cardinalities, and $\mathcal{B}$ scene-context variables. The object state is instance-indexed, $\mathcal{O} = \{o_i\}_{i=1}^{N}$; each object carries attributes $\mathcal{A}_i = \{a_{i\ell}\}_{\ell=1}^{L_i}$; and relations are ordered pairs $\mathcal{R} = \{r_{ij}\}_{(i,j)\in\mathcal{E}}$, preserving subject–object roles. Repeated categories therefore remain distinct even when labels match, and attributes/relations are resolved at the instance level rather than pooled over prompt tokens.

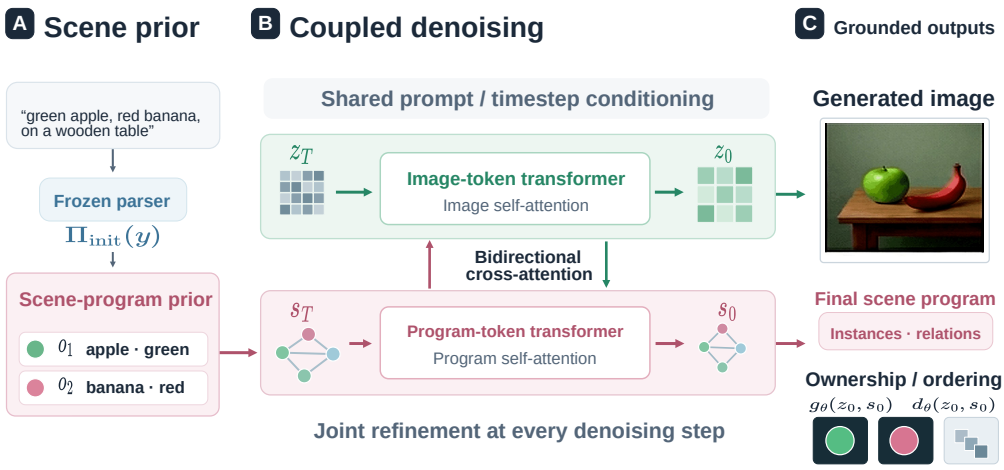


Figure 2: **Overview of PSP-DiT.** (A) A frozen prompt parser initializes an instance-indexed scene-program prior encoding objects, attributes, relations, and cardinalities. (B) Coupled transformer streams jointly denoise image and scene-program latents through bidirectional cross-attention, refining both states throughout generation. (C) The final latents produce a generated image, a structured scene program, and panoptic ownership and ordering predictions. During training, image/program denoising, grounding, alignment, and cycle-consistency losses provide supervision; the frozen cycle recognizer is not used at inference.

The program is *panoptic* because it is later tied to visual ownership and support. It is not a fixed control graph: it is a generated latent state refined jointly with the image latent and grounded spatially. Appendix A.9 gives the procedural overview.

3.3 PSP-DiT: Joint Program–Image Denoising

At denoising time t, PSP-DiT maintains an image latent z_t and a scene-program latent s_t and predicts coupled denoising targets,

$$(\hat{u}_t^z, \hat{u}_t^s) = f_\theta(z_t, s_t, y, t), \tag{1}$$

where $\hat{u}_t^z$ and $\hat{u}_t^s$ update the image and program streams. Let $Z_t = \{z_t^{(m)}\}_{m=1}^{M}$ and $S_t = \{s_t^{(n)}\}_{n=1}^{N_s}$ denote image-latent and scene-program tokens. The coupled denoiser applies within-stream self-attention, bidirectional cross-stream attention, and stream-specific heads:

$$S'_t = \text{ProgAttn}(S_t; y, t), \qquad Z'_t = \text{ImgAttn}(Z_t; y, t), \tag{2}$$

$$\tilde{Z}_t = \text{CrossAttn}_{z \leftarrow s}(Z'_t, S'_t), \qquad \tilde{S}_t = \text{CrossAttn}_{s \leftarrow z}(S'_t, Z'_t), \tag{3}$$

$$\hat{u}_t^z = h_z(\tilde{Z}_t, \tilde{S}_t), \qquad \hat{u}_t^s = h_s(\tilde{S}_t, \tilde{Z}_t). \tag{4}$$

Thus image tokens are updated using the current scene state, and scene tokens are updated using the current visual state. Unlike graph- or layout-conditioned generation, the structured state is sampled and refined rather than kept fixed during denoising. This is critical for repeated categories and ordered relations: instance identity remains explicit in s_t while visual realization is refined in z_t, keeping attributes and directional predicates attached to the correct entities. Appendix A.9 gives the full training and sampling loops.

The evolution of image predictions, program-token cross-attention, and instance ownership throughout denoising is visualized in Appendix A.14.

3.4 Panoptic Grounding and Scene Ownership

Joint denoising alone does not ensure that the scene-program latent explains the rendered image, so PSP-DiT predicts panoptic grounding variables from the coupled image–program state. For

object instance $o_i \in \mathcal{O}$, the soft ownership field $M_t = \{M_{t,i}\}_{i=1}^N$ assigns image-token support to object instances; the grounding head predicts $\hat{M}_t = g_\theta(Z_t, S_t)$; and an ordering head predicts $\hat{D}_t = d_\theta(Z_t, S_t)$ for coarse depth/occlusion structure:

$$M_t = \{M_{t,i}\}_{i=1}^N, \qquad \hat{M}_t = g_\theta(Z_t, S_t), \qquad \hat{D}_t = d_\theta(Z_t, S_t). \tag{5}$$

These variables enforce that repeated objects remain separated, attributes appear on regions owned by the correct instance, and spatial or interaction relations agree with induced geometry. This distinguishes PSP-DiT from structured conditioning methods: scene variables are not only supplied to the model but are part of the generative state and must own visual support. Once scene programs are tied to ownership, training can also require that the intended structure be recoverable from the generated image.

3.5 Training Objectives

Appendix A.9 gives the explicit training procedure. We optimize

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \lambda_{\text{prog}}\mathcal{L}_{\text{prog}} + \lambda_{\text{pan}}\mathcal{L}_{\text{pan}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{cycle}}\mathcal{L}_{\text{cycle}}, \tag{6}$$

where the terms make the image denoisable, the scene program generative, the program spatially grounded, and the generated image recoverable as the intended scene. The component losses are

$$\mathcal{L}_{\text{img}} = \mathbb{E}_{t,z_0,\epsilon}[\ell_z(\hat{u}_t^z, u_t^z)], \qquad \mathcal{L}_{\text{prog}} = \mathbb{E}_{t,s_0,\eta}[\ell_s(\hat{u}_t^s, u_t^s)], \tag{7}$$

$$\mathcal{L}_{\text{pan}} = \mathcal{L}_{\text{own}}(\hat{M}, M^\star) + \lambda_D\mathcal{L}_{\text{depth}}(\hat{D}, D^\star), \qquad \mathcal{L}_{\text{align}} = \sum_{i=1}^N \ell_{\text{align}}(q_i, v_i), \tag{8}$$

$$\mathcal{L}_{\text{cycle}} = \ell_{\text{cycle}}(\Pi(\hat{x}), s^\star). \tag{9}$$

Here u_t^z and u_t^s are denoising targets, $\hat{M}$ and $\hat{D}$ are ownership and ordering predictions, q_i is the latent representation of instance o_i, v_i is pooled image evidence over its support, and Π is a frozen recognizer/parser. Relative to token- or graph-conditioned baselines, PSP-DiT optimizes the scene program as part of the generative state itself rather than as an external control signal.

3.6 Inference and Relation to Prior Conditioning Methods

Appendix A.9 gives the explicit sampling procedure. Given a prompt y, PSP-DiT initializes noisy latents (z_T, s_T), with s_T seeded by a prompt-to-program prior, then jointly samples

$$(z_{t-1}, s_{t-1}) = \Phi_\theta(z_t, s_t, y, t), \qquad t = T, \ldots, 1, \tag{10}$$

where Φ_θ uses the joint predictions $(\hat{u}_t^z, \hat{u}_t^s)$. The final image is decoded from z_0, while the scene explanation is decoded from s_0 and its ownership/ordering outputs.

The key distinction is that s_t is sampled, not fixed: PSP-DiT models $p_\theta(z, s \mid y)$ rather than only $p_\theta(z \mid y)$,

$$p_\theta(z, s \mid y) \qquad \text{vs.} \qquad p_\theta(z \mid y). \tag{11}$$

Flat text conditioning leaves repeated entities, ordered relations, and ownership assignments implicit; graph/layout conditioning keeps structure fixed; and training-free guidance modifies trajectories externally. PSP-DiT instead internalizes scene structure into the generative state and refines it jointly with the image latent, while inference remains a direct forward sampler with no test-time search, reranking, or external optimization.

4 Experiments

We evaluate PSP-DiT along four questions: whether explicit scene-program latents improve compositional faithfulness, whether the gains preserve image quality, which components are responsible, and whether the method remains robust and scalable. Section 4.1 defines the unified evaluation protocol and matched flat-text baseline; Sections 4.2–4.4 report the main benchmark, quality, and skill-level results; Sections 4.5–4.8 analyze components, scaling, efficiency, and robustness; and Section 4.9 provides matched qualitative comparisons.

4.1 Experimental Setup

Unless otherwise stated, all quantitative results are reproduced under a unified evaluation pipeline rather than copied from heterogeneous prior reports. We evaluate compositional faithfulness using GenEval 2 on the Rewritten prompt set [Kamath et al., 2025], SANEval-Simple [Pramanik et al., 2026], PSG-Score [Deng et al., 2025], and DetailMaster Avg. [Jiao et al., 2025]; GenEval 2 and SANEval-Simple are reported on a $\times 100$ scale for readability, while PSG-Score remains on its native $[0, 1]$ scale. For GenEval 2, we sample 4 images per prompt using 4 fixed random seeds and report single-sample prompt-level correctness with no best-of-N selection.

For image quality and alignment, we report FID-30K [Heusel et al., 2017], HPS v2.1 [Wu et al., 2023, HPSv2, 2024], ImageReward [Xu et al., 2023], and a blinded pairwise human-preference study over 200 held-out prompts. Within each comparison block, we use a single fixed inference recipe and keep backbone, training data, optimization budget, and inference settings matched whenever a table is intended to isolate the contribution of the scene-program latent. We use OURS-FLATTEXT to denote the matched baseline with the same backbone, training data, and inference recipe as PSP-DiT, but conditioned only on flat text tokens and without the scene-program latent. Additional evaluator, dataset, algorithmic, and skill-aggregation details are provided in Appendices A.5, A.9, and A.11.

Table 1: **Main quantitative evidence.** Panel (a) reports the main compositional benchmark comparison; panel (b) checks image quality and prompt alignment; panel (c) breaks down gains by compositional skill; panel (d) tests scaling and resolution robustness; and panel (e) reports efficiency for the matched 1B setting. GE2 denotes GenEval 2 on the Rewritten prompt set reported on a $\times 100$ scale, SANE denotes SANEval-Simple on a $\times 100$ scale, DM denotes DetailMaster Avg., IR denotes ImageReward, and Pref. denotes human preference win rate. All panels use the unified evaluation pipeline described in Section 4.1. Human preference denotes model-specific win rates against the sampled opponent pool; values are not complementary head-to-head percentages.

(a) Compositional benchmarks

Model	GE2↑	SANE↑	PSG↑	DM↑	Rank↓
FLUX.1-dev	29.5	61.8	0.62	63.6	3.25
SD3.5-Large	19.0	55.4	0.60	59.6	5.00
SG-Adapter + SD3.5-Large	24.7	59.8	0.66	62.1	3.50
Ours-FlatText	32.8	65.9	0.65	66.8	2.25
PSP-DiT	**38.2**	**73.6**	**0.73**	**72.4**	**1.00**

(b) Quality and alignment

Model	FID-30K↓	HPS↑	IR↑	Pref. (%)↑
FLUX.1-dev	8.7	31.4	1.31	45.2
Ours-FlatText	8.1	31.9	1.35	48.7
PSP-DiT	**7.9**	**32.6**	**1.42**	**56.8**

(c) Compositional skill breakdown

Model	Count↑	Attr.↑	Spatial↑	Role↑	Long↑	Avg.↑
FLUX.1-dev	58.4	63.1	60.2	54.8	61.5	59.6
Ours-FlatText	62.7	67.4	64.1	58.9	65.8	63.8
PSP-DiT	**74.1**	**75.3**	**72.8**	**69.7**	**71.6**	**72.7**

(d) Scaling and resolution

Model	Size	Res.	GE2↑	SANE↑	DM↑	HPS↑
Ours-FlatText	0.5B	512	28.9	61.7	63.2	30.8
PSP-DiT	**0.5B**	**512**	**33.6**	**68.4**	**67.9**	**31.3**
Ours-FlatText	1.0B	512	32.8	65.9	66.8	31.9
PSP-DiT	**1.0B**	**512**	**38.2**	**73.6**	**72.4**	**32.6**
Ours-FlatText	3.0B	1024	34.7	68.1	69.9	32.8
PSP-DiT	**3.0B**	**1024**	**41.5**	**76.8**	**75.6**	**33.4**

(e) Efficiency and overhead

Model	Params	VRAM↓	Lat.↓	Rel.↓	GE2↑
Ours-FlatText	12.8B	29.6	21.4	1.00×	32.8
PSP-DiT	**13.4B**	**32.1**	**23.9**	**1.12×**	**38.2**

4.2 Main Compositional Benchmark Comparison

Table 1(a) isolates the main claim of the paper. While our matched flat-text baseline is already competitive, PSP-DiT delivers a further decisive improvement on GenEval 2, SANEval-Simple, PSG-Score, and DetailMaster. This gap cleanly attributes the compositional gains to explicit scene-program modeling rather than to backbone strength alone.

Comparisons with recent direct-generation systems and GEMS-based agentic generation are provided in Appendix A.10.1.

4.3 Image Quality and Prompt Alignment

Table 1(b) addresses the most immediate alternative explanation for Table 1(a): that compositional gains may come from sacrificing realism or overall visual quality. This is not the case. Under a unified

Table 2: **Method-component and robustness analyses.** Panel (a) ablates the structured scene-program latent. Simply appending scene-program tokens yields modest gains; the full improvement requires instance indexing, joint program–image denoising, panoptic grounding, and cycle consistency. Panel (b) perturbs the scene program at inference time and shows that PSP-DiT remains robust to moderate program noise without finetuning.

(a) Scene-program latent ablation

Variant	Prog.	Inst.	Joint	Pan.	Cyc.	GE2↑	SANE↑	PSG↑	DM↑
Ours-FlatText						32.8	65.9	0.65	66.8
+ scene-program tokens	✓					33.7	66.8	0.66	67.4
+ instance-indexed program	✓	✓				35.2	68.9	0.68	68.9
+ joint program denoising	✓	✓	✓			36.8	71.1	0.70	70.4
+ panoptic grounding	✓	✓	✓	✓		37.6	72.6	0.72	71.6
PSP-DiT	✓	✓	✓	✓	✓	**38.2**	**73.6**	**0.73**	**72.4**

(b) Program-noise robustness

Condition	GE2↑	SANE↑	PSG↑	DM↑
Ours-FlatText	32.8	65.9	0.65	66.8
Exact program	**38.2**	**73.6**	**0.73**	**72.4**
Paraphrase parse	37.6	72.8	0.72	71.9
10% node/edge drop	36.9	71.5	0.71	71.0
20% node/edge drop	35.8	69.9	0.69	69.6
10% relation corrupt.	35.4	69.1	0.69	69.0

evaluation protocol, PSP-DiT matches or improves fidelity, improves prompt-alignment metrics, and yields the strongest human preference, indicating that the scene-program latent improves structured faithfulness without inducing the usual compositionality–quality trade-off.

4.4 Breakdown by Compositional Skill

The skill columns in Table 1(c) are reproduced benchmark-aggregated scores. Each skill is computed by averaging benchmark-native submetrics and rescaling them to a common 0–100 range, with weights proportional to the number of prompts in each contributing subset: Counting from numeracy-focused evaluations, Attr. Binding from attribute-binding evaluations, Spatial Rel. from spatial-relation evaluations, Role Rel. from a reproduced role-sensitive relation subset, and Long Prompts from long-prompt / complex-composition evaluations. Concretely, SANEval-Simple contributes the numeracy, attribute-binding, and spatial-relation components; GenEval 2 (Rewritten) contributes prompt-level compositional and role-sensitive relation subsets; and DetailMaster contributes the long-prompt component. We provide the exact sub-benchmark mapping, prompt counts, and weights in Appendix A.11.

Table 1(c) shows that the gains of PSP-DiT are concentrated exactly where current text-to-image systems remain most fragile. The largest improvements appear on counting, attribute binding, and role-sensitive relations, indicating that the panoptic scene-program latent addresses the intended compositional bottlenecks rather than merely improving overall image quality.

4.5 Ablation of the Scene-Program Latent

Table 2(a) rules out the strongest alternative explanation for our results: that they can be recovered by adding structured prompt tokens to an otherwise standard model. They cannot. The token-only variant yields only small gains, while the full improvement emerges only when the scene program is jointly denoised as an instance-indexed, panoptically grounded latent with cycle consistency. Thus, the benefit of PSP-DiT is not extra conditioning, but elevating scene structure to a first-class generative state.

Further evidence from controlled fixed-versus-joint refinement, stage-wise program probing, and targeted latent interventions is provided in Appendix A.13.

4.6 Scaling and Resolution Robustness

Table 1(d) shows that the gains of PSP-DiT are not confined to a single model size or image resolution. Across 0.5B–3B trainable backbones and from 512px to 1024px generation, PSP-DiT consistently improves compositional fidelity while maintaining strong alignment quality. This indicates that scene-program latents remain useful as capacity grows, rather than acting as a compensatory mechanism only for smaller models.

4.7 Efficiency and Overhead

Table 1(e) reports efficiency under a fully matched H200 setup for the 1B model setting. This measurement is intentionally reported only against the matched flat-text baseline: unlike quality benchmarks, systems measurements are highly sensitive to hardware, precision, attention backend, and compilation stack, making cross-paper comparisons substantially less informative. The result is favorable: promoting scene structure to a first-class latent variable adds only modest overhead in memory and latency, while yielding a large improvement in compositional fidelity.

4.8 Robustness to Program Noise

Table 2(b) addresses a natural concern for structured generation methods: sensitivity to imperfect scene programs. The answer is favorable. PSP-DiT degrades gracefully under paraphrase-induced parsing variation, node/edge dropout, and relation corruption, and it remains clearly stronger than the matched flat-text baseline under moderate perturbations. This indicates that the gains of scene-program latents do not rely on an unrealistically perfect parser.

4.9 Qualitative Results

Figure 3 provides a visual counterpart to the benchmark results. Across matched prompts, random seeds, and inference budgets, PSP-DiT better preserves the compositional constraints that define the scene: object counts, attribute ownership, subject–object roles, spatial ordering, and grounded visual support. These examples support the central claim that explicit panoptic scene-program latents improve structured faithfulness beyond what is obtained from stronger flat text conditioning alone.

5 Conclusion

We introduced PSP-DiT, a text-to-image generative model that treats a panoptic scene program as a jointly denoised latent state rather than an external control signal or post-hoc parse. By coupling image and scene-program streams, and grounding scene variables through ownership and cycle-consistency objectives, PSP-DiT makes object identity, attribute ownership, relations, counts, and visual support explicit during generation. Across compositional benchmarks, quality metrics, ablations, scaling tests, and robustness analyses, PSP-DiT improves structured faithfulness while preserving image quality and adding only modest inference overhead. These results suggest that explicit scene-state generation is a promising direction for resolving the remaining compositional failures of modern text-to-image systems.

Limitations This work focuses on text-to-image generation. Extending panoptic scene-program latents to text-to-video generation, where scene state must remain temporally consistent across frames, is left to future work.

Acknowledgments and Disclosure of Funding

The author acknowledges computational support from the NSF-supported Delta and DeltaAI systems.

Competing Interests. The author declares no competing interests relevant to this work.



Figure 3: **Qualitative compositional stress cases**. We compare FLUX.1-dev, SD3.5-Large, and PSP-DiT using the same prompts, random seeds, and matched inference budget. Rows test: (1) counting and attribute ownership, with three mugs, a spoon only in the blue mug, and an apple in front of the red mug; (2) attribute binding, with a green apple on a white plate, a red banana on a black plate, and a spoon between them; (3) role-sensitive spatial reasoning, with a dog behind a cat, the cat carrying a red ball, and a frisbee to the right; and (4) long structured composition, with classroom roles, notebook ownership, a yellow-shirt student, and exactly two pencils. The baselines often produce visually plausible images while losing instance-level constraints. PSP-DiT more consistently preserves instance identity, attribute ownership, ordered relations, and visual support, qualitatively corroborating the trends in Table 1(a,c). Full prompts are provided in Appendix A.12.

References

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2023. URL https://arxiv.org/abs/2211.01324.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1737–1752. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/bar-tal23a.html.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. *OpenAI Technical Report*, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Black Forest Labs. FLUX.1 [dev]. Hugging Face model card, 2024. URL https://huggingface.co/black-forest-labs/FLUX.1-dev.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592116. URL https://doi.org/10.1145/3592116.

Huancheng Chen, Jingtao Li, Weiming Zhuang, Haris Vikalo, and Lingjuan Lyu. Training-free layout-to-image generation with marginal attention constraints, 2026. URL https://arxiv.org/abs/2411.10495.

Kaijie Chen, Zihao Lin, Zhiyang Xu, Ying Shen, Yuguang Yao, Joy Rimchala, Jiaxin Zhang, and Lifu Huang. R2I-bench: Benchmarking reasoning-driven text-to-image generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12595–12630, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.636. URL https://aclanthology.org/2025.emnlp-main.636/.

Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5331–5341, 2024. doi: 10.1109/WACV57701.2024.00526.

Zuyao Chen, Jinlin Wu, Zhen Lei, and Chang Wen Chen. What makes a scene ? scene graph-based evaluation and feedback for controllable generation, 2025b. URL https://arxiv.org/abs/2411.15435.

Xueqing Deng, Linjie Yang, Qihang Yu, Chenglin Yang, and Liang-Chieh Chen. Leveraging panoptic scene graph for evaluating fine-grained text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15107–15116, October 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12606–12633. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PUIqjT4rzq7.

Alessandro Fontanella, Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, and Sarah Parisot. Generating compositional scenes via text-to-image rgba instance generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 43864–43893. Curran Associates, Inc., 2024. doi: 10.52202/079017-1392. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4d90363e96781894b1327b87d1ade17e-Paper-Conference.pdf.

Ziqi Gao, Weikai Huang, Jieyu Zhang, Aniruddha Kembhavi, and Ranjay Krishna. Generate any scene: Scene graph driven data synthesis for visual generation training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EwdWR6lfvW.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=Wbr51vK331.

Shashank Gupta, Chaitanya Ahuja, Tsung-Yu Lin, Sreya Dutta Roy, Harrie Oosterhuis, Maarten de Rijke, and Satya Narayan Shukla. A simple and effective reinforcement learning method for text-to-image diffusion fine-tuning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=i8WJhKn455.

Xu Han, Linghao Jin, Xiaofeng Liu, and Paul Pu Liang. Progressive compositionality in text-to-image generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=S85PP4xjFD.

Zefeng He, Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Yu Cheng, and Yang Yang. Gems: Agent-native multimodal generation with memory and skills, 2026. URL https://arxiv.org/abs/2603.28088.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

HPSv2. HPS v2: Benchmarking Text-to-Image Generative Models. GitHub repository, 2024. URL https://github.com/tgxs002/HPSv2. Official repository; includes the HPS v2.1 release.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and Interpretable Text-to-Image Faithfulness Evaluation with Question Answering . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20349–20360, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.01866. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01866.

Jen-Yuan Huang, Tong Lin, and Yilun Du. Long-text-to-image generation via compositional prompt decomposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jxyEci13Dd.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=weHBzTLXpH.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An Enhanced and Comprehensive Benchmark for Compositional Text-to-Image Generation . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 47(05):3563–3579, May 2025. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3531907. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2025.3531907.

Hirunima Jayasekara, Chuong Huynh, Yixuan Ren, Christabel Acquaye, and Abhinav Shrivastava. All-in-one conditioning for text-to-image synthesis, 2026. URL https://arxiv.org/abs/2602.09165.

Dongzhi Jiang, Guanglu Song, Xiaoshi Wu, Renrui Zhang, Dazhong Shen, Zhuofan Zong, Yu Liu, and Hongsheng Li. Comat: Aligning text-to-image diffusion model with image-to-text concept matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=OW1ldvMNJ6.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Xika Lin, Ying Shen, and Yaliang Li. Detailmaster: Can your text-to-image model handle long prompts?, 2025. URL https://arxiv.org/abs/2505.16915.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation, 2025. URL https://arxiv.org/abs/2512.16853.

Seyed Amir Kasaei, Ali Aghayari, Arash Marioriyad, Niki Sepasian, MohammadAmin Fazli, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. Evaluating the evaluators: Metrics for compositional text-to-image generation, 2025. URL https://arxiv.org/abs/2509.21227.

Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7667–7677, 2023. doi: 10.1109/ICCV51070.2023.00708.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Evaluating and improving compositional text-to-visual generation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5290–5301, 2024a. URL https://api.semanticscholar.org/CorpusID:272915959.

Chunhan Li, Qifeng Wu, Jia-Hui Pan, Ka-Hei Hui, Jingyu Hu, Yuming Jiang, Bin Sheng, Xihui Liu, Wenjuan Gong, and Zhengzhe Liu. codrawagents: A multi-agent dialogue framework for compositional image generation, 2026a. URL https://arxiv.org/abs/2603.12829.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024b. URL https://arxiv.org/abs/2402.17245.

Mingcheng Li, Xiaolu Hou, Ziyang Liu, Dingkang Yang, Ziyun Qian, Jiawei Chen, Jinjie Wei, Yue Jiang, Qingyao Xu, and Lihua Zhang. MCCD: Multi-Agent Collaboration-based Compositional Diffusion for Complex Text-to-Image Generation . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13263–13272, Los Alamitos, CA, USA, June 2025a. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.01238. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.01238.

Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. Enhancing compositional text-to-image generation with reliable random seeds. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=5BSlakturs.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22511–22521, 2023. doi: 10.1109/CVPR52729.2023.02156.

Zejian Li, Chenye Meng, Yize Li, Ling Yang, Zhongni Liu, Changle Xie, Jiarui Ma, Shengyuan Zhang, Jiayi Li, Guang Yang, Changyuan Yang, Zhiyuan Yang, Jinxiong Chang, and Lingyun Sun. LAION-comp: Unlocking controllable and compositional generation with structural annotations, 2026b. URL https://openreview.net/forum?id=PFhrOUJZ5o.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors,

Computer Vision – ECCV 2024, pages 366–384, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72673-6.

Jiaqi Liu, Tao Huang, and Chang Xu. Training-free composite scene generation for layout-to-image synthesis. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXVIII*, page 37–53, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73112-9. doi: 10.1007/978-3-031-73113-6_3. URL https://doi.org/10.1007/978-3-031-73113-6_3.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 423–439, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19790-1.

Sifei Liu, Weili Nie, An-Chieh Cheng, Morteza Mardani, Chao Liu, Benjamin Eckart, and Arash Vahdat. Compositional text-to-image generation with feedforward layout generation. In Alessio Del Bue, Cristian Canton, Jordi Pont-Tuset, and Tatiana Tommasi, editors, *Computer Vision – ECCV 2024 Workshops*, pages 23–33, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-91979-4.

Weili Nie, Sifei Liu, Morteza Mardani, Chao Liu, Benjamin Eckart, and Arash Vahdat. Compositional text-to-image generation with dense blob representations. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38091–38116. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nie24b.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Khiem Pham, Quang Nguyen, Tung Nguyen, Jingsen Zhu, Michele Santacatterina, Dimitris Metaxas, and Ramin Zabih. Dedpo: Debiased direct preference optimization for diffusion models, 2026. URL https://arxiv.org/abs/2602.06195.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Rishav Pramanik, Ian E. Nielsen, Jeff Smith, Saurav Pandit, Ravi P. Ramachandran, and Zhaozheng Yin. Saneval: Open-vocabulary compositional benchmarks with failure-mode diagnosis, 2026. URL https://arxiv.org/abs/2602.00249.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Sarah Rastegar, Violeta Chatalbasheva, Sieger Falkena, Anuj Singh, Yanbo Wang, Tejas Gokhale, Hamid Palangi, and Hadi Jamali-Rad. Infsplign: Inference-time spatial alignment of text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2512.17851.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022. doi: 10.1109/CVPR52688.2022.01042.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ec795aeadae0b7d230fa35cbaf04c041-Paper-Conference.pdf.

Stability AI. Stable Diffusion 3.5 Large. Hugging Face model card, 2024. URL https://huggingface.co/stabilityai/stable-diffusion-3.5-large.

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

Maria Mihaela Trusca, Wolf Nuyts, Jonathan Thomm, Robert Honig, Thomas Hofmann, Tinne Tuytelaars, and Marie-Francine Moens. Object-attribute binding in text-to-image generation: Evaluation and control, 2024. URL https://arxiv.org/abs/2404.13766.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8238, 2024. doi: 10.1109/CVPR52733.2024.00786.

Shijian Wang, Runhao Fu, Siyi Zhao, Qingqin Zhan, Xingjian Wang, Jiarui Jin, Yuan Lu, Hanqian Wu, and Cunjian Chen. Synthetic curriculum reinforces compositional text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21953–21963, June 2026a.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6232–6242, 2024a. doi: 10.1109/CVPR52733.2024.00596.

Yunnan Wang, Ziqiang Li, Wenyao Zhang, Zequn Zhang, Baao Xie, Xihui Liu, Wenjun Zeng, and Xin Jin. Scene graph disentanglement and composition for generalizable complex image generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 98478–98504. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3125. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b288470688e72f58c02031304ad6339f-Paper-Conference.pdf.

Zengbin Wang, Xuecai Hu, Yong Wang, Feng Xiong, Man Zhang, and Xiangxiang Chu. Everything in its place: Benchmarking spatial intelligence of text-to-image models. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=ddFN3lWpIr.

Song Wen, Guian Fang, Renrui Zhang, Peng Gao, Hao Dong, and Dimitris Metaxas. Improving compositional text-to-image generation with large vision-language models, 2023. URL https://arxiv.org/abs/2310.06311.

Mingrui Wu, Lu Wang, Pu Zhao, Fangkai Yang, Jianjin Zhang, Jianfeng Liu, Yuefeng Zhan, Weihao Han, Hao Sun, Jiayi Ji, Xiaoshuai Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, and Rongrong Ji. Reprompt: Reasoning-augmented reprompting for text-to-image generation via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HJ3vgg7TYQ.

Weijia Wu, Zhuang Li, Yefei He, Mike Zheng Shou, Chunhua Shen, Lele Cheng, Yan Li, Tingting Gao, and Di Zhang. Paragraph-to-Image Generation with Information-Enriched Diffusion Model. *International Journal of Computer Vision*, 133(8):5413–5434, August 2025. ISSN 1573-1405. doi: 10.1007/s11263-025-02435-1. URL https://doi.org/10.1007/s11263-025-02435-1.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, 2023. URL https://arxiv.org/abs/2306.09341.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=N8Oj1XhtYZ.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7418–7427, 2023. doi: 10.1109/ICCV51070.2023.00685.

Xin Xie and Dong Gong. DyMO: Training-Free Diffusion Model Alignment with Dynamic Multi-Objective Scheduling . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13220–13230, Los Alamitos, CA, USA, June 2025. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.01234. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.01234.

Bicheng Xu, Qi Yan, Renjie Liao, Lele Wang, and Leonid Sigal. Joint generative modeling of grounded scene graphs and images via diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=2cxxZI2LOL.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15903–15935. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/33646ef0ed554145eab65f6250fab0c9-Paper-Conference.pdf.

Hongji Yang, Yucheng Zhou, Wencheng Han, Runzhou Tao, Zhongying Qiu, Jianfei Yang, and Jianbing Shen. Hicogen: Hierarchical compositional text-to-image generation in diffusion models via reinforcement learning, 2025. URL https://arxiv.org/abs/2511.19965.

Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, and Ziwei Liu. Panoptic video scene graph generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18675–18685, 2023. doi: 10.1109/CVPR52729.2023.01791.

Yuxuan Yao, Yuxuan Chen, Hui Li, Kaihui Cheng, Qipeng Guo, Yuwei Sun, Zilong Dong, Jingdong Wang, and Siyu Zhu. Prompt reinjection: Alleviating prompt forgetting in multimodal diffusion transformers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=P3b1028tNC.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=AFDcYJKhND. Featured Certification.

Arman Zarei, Keivan Rezaei, Samyadeep Basu, Mehrdad Saberi, Mazda Moayeri, Priyatham Kattakinda, and Soheil Feizi. Improving compositional attribute binding in text-to-image generative models via enhanced text embeddings, 2025. URL https://arxiv.org/abs/2406.07844.

Daoan Zhang, Che Jiang, Ruoshi Xu, Biaoxiang Chen, Zijian Jin, Yutian Lu, Jianguo Zhang, Liang Yong, Jiebo Luo, and Shengda Luo. Worldgenbench: A world-knowledge-integrated benchmark for reasoning-driven text-to-image generation, 2025a. URL https://arxiv.org/abs/2505.01490.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023. doi: 10.1109/ICCV51070.2023.00355.

Xinchen Zhang, Ling Yang, Guohao Li, YaQi Cai, xie jiake, Yong Tang, Yujiu Yang, Mengdi Wang, and Bin CUI. Itercomp: Iterative composition-aware feedback learning from model gallery for text-to-image generation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=4w99NAikOE.

Xu Zhang, Jin Yuan, Hanwang Zhang, Guojin Zhong, Yongsheng Zang, Jiacheng Lin, and Zhiyong Li. Sgdiff: scene graph guided diffusion model for image collaborative segcaptioning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference*

on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025c. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i10.33113. URL https://doi.org/10.1609/aaai.v39i10.33113.

Chengyang Zhao, Yikang Shen, Zhenfang Chen, Mingyu Ding, and Chuang Gan. Textpsg: Panoptic scene graph generation from textual descriptions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2827–2838, 2023. doi: 10.1109/ICCV51070.2023.00266.

Peiang Zhao, Han Li, Ruiyang Jin, and S. Kevin Zhou. Loco: Locally constrained training-free layout-to-image synthesis, 2024. URL https://arxiv.org/abs/2311.12342.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0u1LigJaab.

Zijian Zhou, Zheng Zhu, Holger Caesar, and Miaojing Shi. Openpsg: Open-set panoptic scene graph generation via large multimodal models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 199–215, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72684-2.

A Supplementary Material

This appendix is organized as an implementation- and reproduction-oriented supplement. We first document the data construction, architecture, optimization, compute, and evaluation protocol used by PSP-DiT, then provide parser/recognizer details, a compact reproducibility checklist, procedural algorithms, contextual baselines, skill aggregation, and qualitative stress cases.

Appendix Contents

Appendix entry	Page
A. Supplementary Material	18
A.1. Training data and scene-program construction	18
A.2. Architecture and hyperparameters	19
A.3. Optimization and training schedule	19
A.4. Compute, hardware, and runtime accounting	19
A.5. Evaluation protocol and metric details	20
A.6. Human preference study	20
Table 3: human preference protocol	21
A.7. Parser, recognizer, and program-noise protocol	20
A.8. Reproducibility checklist	21
A.9. Algorithmic details	21
Algorithm 1: PSP-DiT overview	22
Algorithm 2: PSP-DiT training	22
Algorithm 3: PSP-DiT inference	22
A.10. Evaluation references and systems context	23
A.10.1. Comparison with recent generative systems	23
Table 4: recent generative systems	23
Table 5: public baseline positioning	23
Table 6: external systems anchor	24
A.11. Skill aggregation details	23
Table 7: skill-to-benchmark mapping	24
A.12. Qualitative prompts and additional stress cases	24
Table 8: prompts for main qualitative cases	24
Figures 4–9: additional qualitative pages	25–26
A.13. Program-latent diagnostics and controlled refinement	27
Table 9: program-latent diagnostics	27
A.14. Timestep-Wise Scene-Program Visualization	28
Figure 10: Timestep-wise scene-program evolution	28

A.1 Training Data and Scene-Program Construction

PSP-DiT is trained on a mixture of captioned image data and structure-enriched image–text examples. For all matched comparisons, Ours-FlatText and PSP-DiT use the same image–caption pool, filtering, resolution schedule, and optimization budget; the only difference is that PSP-DiT additionally receives the panoptic scene-program latent.

We construct scene programs in three stages. First, prompts are parsed into candidate object instances, attributes, relations, cardinalities, and scene context using a frozen text-to-program parser. Second, image-side grounding targets are obtained using a frozen panoptic segmentation / open-vocabulary grounding stack, producing coarse ownership maps $M^\star$ and ordering targets $D^\star$. Third, we reconcile text-derived and image-derived structure by removing unsupported objects, merging duplicate instance mentions, and retaining only relations whose endpoints are both grounded.

Each resulting training tuple has the form

$$(x, y, s^\star, M^\star, D^\star),$$

where x is the image, y is the caption or prompt, $s^\star = (\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{C}, \mathcal{B})$ is the target panoptic scene program, $M^\star$ is the ownership target, and $D^\star$ is the ordering target. The same images and prompts are used for Ours-FlatText; the scene-program components are removed from that baseline.

We discard training examples whose program parser produces no foreground instance or whose grounding stack cannot assign any object support. This prevents the program stream from being

trained on empty or visually unsupported scene structure. We also cap the maximum number of foreground object instances at $N_{\max} = 16$ and keep a fixed set of register/background tokens for scene context.

A.2 Architecture and Hyperparameters

All PSP-DiT models use the same RAE latent interface as the matched Ours-FlatText baseline. The image stream is a DiT-style transformer over RAE image tokens, and the program stream is a transformer over scene-program tokens. For a backbone size of 1B, we use hidden width 2048, 24 transformer blocks, 16 attention heads, and MLP expansion ratio 4. The 0.5B and 3B variants are obtained by scaling width/depth while keeping the program-token interface and loss definitions fixed.

The scene-program stream uses one token per object instance, one token per relation edge, one token per cardinality group, and a fixed bank of background/context tokens. We use $N_{\max} = 16$ object-instance tokens, $R_{\max} = 32$ relation tokens, $C_{\max} = 8$ cardinality tokens, and $B_{\max} = 8$ scene-context tokens. Unused slots are masked and do not contribute to program loss or grounding loss.

The coupled transformer contains three operations per block:

$$S'_t = \mathrm{ProgAttn}(S_t; y, t), \qquad Z'_t = \mathrm{ImgAttn}(Z_t; y, t),$$
$$\widetilde{Z}_t = \mathrm{CrossAttn}_{z \leftarrow s}(Z'_t, S'_t), \qquad \widetilde{S}_t = \mathrm{CrossAttn}_{s \leftarrow z}(S'_t, Z'_t).$$

The final image and program denoising heads are linear projection heads applied to the coupled stream outputs.

The grounding head g_θ predicts low-resolution ownership logits over the RAE latent grid. The ordering head d_θ predicts pairwise or coarse ordinal depth scores among object instances. Both heads are trained only through the structural losses and are used at inference to decode the final scene explanation.

A.3 Optimization, Loss Weights, and Training Schedule

All matched internal comparisons use the same optimizer, learning-rate schedule, batch size, training data, and number of optimization steps. We train with AdamW, weight decay 0.03, $\beta_1 = 0.9$, $\beta_2 = 0.95$, gradient clipping at 1.0, bf16 mixed precision, and exponential moving average of model weights with decay 0.9999. The base learning rate is 1.0×10^{-4} with linear warmup over the first 10k steps followed by cosine decay.

Unless otherwise stated, the objective weights are

$$\lambda_{\mathrm{prog}} = 1.0, \qquad \lambda_{\mathrm{pan}} = 1.0, \qquad \lambda_{\mathrm{align}} = 0.25, \qquad \lambda_{\mathrm{cycle}} = 0.5, \qquad \lambda_D = 0.1.$$

These weights are fixed across all PSP-DiT variants and are not tuned per benchmark. The ablations in Table 2 remove architectural components while keeping the remaining loss weights and optimization schedule unchanged.

For the 1B setting, both Ours-FlatText and PSP-DiT are trained for 1.2M optimization steps at 512px. The 0.5B scaling experiment is trained for 800k steps at 512px, and the 3B model is trained for 1.5M steps with a final 1024px resolution phase. All models use the same 50-step inference recipe for the main benchmark tables unless otherwise stated.

A.4 Compute, Hardware, and Runtime Accounting

All internal training runs were performed on NVIDIA H200 GPUs using bf16 mixed precision. Efficiency measurements in Table 1 are reported on a single H200 with batch size 1, 1024×1024 resolution, 50 denoising steps, FlashAttention enabled, `torch.compile` enabled, and end-to-end timing including VAE decode. Latency is averaged over 100 timed runs after 10 warmup runs.

The 1B Ours-FlatText and PSP-DiT models use total deployed pipeline sizes of 12.8B and 13.4B parameters, respectively. Backbone Size in Table 1 denotes only the trainable generative backbone, excluding frozen text encoders and auxiliary pipeline components. Total Pipeline Params in Table 1 includes the full loaded inference pipeline.

The main 1B PSP-DiT training run used 64 H200 GPUs for 1.2M optimization steps. The 0.5B scaling run used 32 H200 GPUs, and the 3B scaling run used 128 H200 GPUs. Including ablations, reproduced baselines, scaling runs, and failed preliminary runs, the full project used approximately 1.1M H200 GPU-hours. The experiments reported in the main paper account for approximately 0.72M H200 GPU-hours.

For external context only, Table 6 reports a published FLUX.1-dev systems anchor on RTX A6000. We do not compare that latency directly against our H200 measurements because systems numbers depend strongly on hardware, precision, attention kernels, compilation stack, and implementation.

A.5 Evaluation Protocol and Metric Details

All reported benchmark numbers are reproduced under a unified evaluation pipeline unless explicitly marked as a public reference. For GenEval 2, we evaluate the Rewritten prompt set [Kamath et al., 2025], report prompt-level correctness multiplied by 100, sample 4 images per prompt using 4 fixed random seeds, and average over all generated images. We do not use best-of-N selection, reranking, or prompt-specific cherry-picking. This is also the configuration under which the public FLUX.1-DEV [Black Forest Labs, 2024] and SD3.5-LARGE [Stability AI, 2024, Esser et al., 2024] prompt-level scores match our anchor values most closely.

For SANEval, we report SANEval-Simple averaged score multiplied by 100 [Pramanik et al., 2026]. For PSG-Score, we use the native $[0, 1]$ scale [Deng et al., 2025]. For DetailMaster Avg., we compute the mean over the eight benchmark task accuracies [Jiao et al., 2025]. FID-30K is computed using Fréchet Inception Distance [Heusel et al., 2017] on MJHQ-30K [Li et al., 2024b]. HPS uses the v2.1 scorer [Wu et al., 2023, HPSv2, 2024], and ImageReward uses the ImageReward-v1.0 checkpoint [Xu et al., 2023] on PartiPrompts [Yu et al., 2022]. Human preference is reported as pairwise win rate over 200 held-out PartiPrompts prompts, with ties disallowed and 95% confidence intervals computed by bootstrap over prompts.

All baseline models are evaluated with the same resolution, number of denoising steps, and prompt set within each comparison block. OURS-FLATTEXT denotes the matched internal baseline with the same backbone, data, optimization schedule, and inference recipe as PSP-DiT, but with the scene-program latent removed and standard flat-text conditioning used instead. As a calibration check, our reproduced FLUX.1-DEV values are consistent with the public benchmark and model references used to define the evaluation settings.

A.6 Human Preference Study

We conduct blinded pairwise evaluations on 200 held-out PartiPrompts prompts. For each prompt, each evaluated model is compared against an image sampled from the other two models (FLUX.1-dev, Ours-FlatText, and PSP-DiT). Each pair receives three independent forced-choice judgments, with randomized presentation order and no ties. Raters were instructed: "Choose the image that better satisfies the prompt while preserving overall visual quality."

Human Pref. is the percentage of individual judgments favoring each evaluated model against its sampled opponent pool, without majority-vote aggregation. These model-specific win rates are not complementary shares of a single head-to-head comparison. We estimate 95% confidence intervals using 10,000 prompt-level bootstrap resamples, retaining the associated rater judgments. Participation was voluntary, and no compensation was provided. The study involved only low-risk preference judgments over generated images and no potential risks were incurred by participants.

A.7 Scene-Program Parser and Cycle Recognizer

The prompt-to-program prior Π_{init} is a frozen parser that converts a prompt into an initial scene-program graph with object instances, attributes, ordered relations, cardinality groups, and scene-context tokens. The parser is used to initialize the program latent and to construct training targets; its output is not treated as an oracle final scene. During sampling, PSP-DiT updates the scene program jointly with the image latent.

The cycle recognizer Π_{cyc} is a frozen image-to-scene recognizer that predicts object instances, attributes, relations, and coarse ownership from a generated image. It is used only to compute $\mathcal{L}_{\text{cycle}}$

Table 3: **Human preference protocol.** We use a blinded forced-choice study on held-out PartiPrompts prompts to evaluate prompt satisfaction and visual quality under matched inference settings.

Study component	Protocol
Prompt source	PartiPrompts held-out subset
Number of prompts	200
Raters per pair	3
Judgments per evaluated model	600
Comparison type	Blinded pairwise forced choice
Displayed instruction	Raters were instructed: "Choose the image that better satisfies the prompt while preserving overall visual quality."
Tie policy	Ties disallowed
Aggregation	Model-specific individual-rater win rate
Comparison pool	Each model versus a sampled image from the other two models
Order randomization	Left/right randomized per prompt and rater
Inference budget	Matched across models
Confidence interval	Prompt-level bootstrap; 10,000 resamples
Qualitative-prompt overlap	Excluded
Participation	Voluntary
Compensation	None
Participant risk	No potential risks incurred

during training. Gradients do not update Π_{cyc}. At inference time, PSP-DiT does not require an external cycle recognizer, reranker, or search loop.

For robustness analysis, we perturb the scene program only at inference time. Node/edge dropout removes object-instance nodes together with their incident attribute and relation edges; dropped edges are masked and orphaned attributes/relations are discarded so that the residual program remains syntactically valid. Relation corruption randomly replaces relation labels while preserving endpoints. Paraphrase noise reparses a paraphrased prompt before sampling.

A.8 Reproducibility Checklist

We summarize the implementation details needed to reproduce the main results.

- Image latent interface: RAE encoder/decoder shared by Ours-FlatText and PSP-DiT.
- Main training objective: Eq. (6).
- Main sampler: Eq. (10).
- Inference steps: 50 steps for all main comparisons.
- Main resolution: 512px for 0.5B and 1B models; 1024px for the 3B high-resolution model.
- Precision: bf16.
- Attention backend: FlashAttention.
- Compilation: `torch.compile` enabled for timing.
- GenEval 2: Rewritten split, 4 fixed seeds, 4 images per prompt, no best-of-N.
- SANEval: SANEval-Simple.
- HPS: v2.1 scorer.
- ImageReward: ImageReward-v1.0.
- Human preference: 200 held-out PartiPrompts prompts, blinded forced choice, ties disallowed, 10,000 bootstrap resamples; see Appendix A.6.
- Efficiency timing: batch size 1, includes VAE decode, averaged over 100 runs after 10 warmup runs.

A.9 Algorithmic Details

This section collects the procedural details for PSP-DiT. The main paper presents the modeling components and objectives; the algorithms here spell out the corresponding overview, training loop, and inference loop without interrupting the main method narrative.

Algorithm 1 PSP-DiT overview

Require: Prompt y, image x during training

Encode image into RAE latent $z_0 \leftarrow E_{\mathrm{RAE}}(x)$
Construct scene program $s^\star = (\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{C}, \mathcal{B})$
Corrupt image and program latents to obtain (z_t, s_t)
Jointly denoise (z_t, s_t) with coupled transformer f_θ
Predict panoptic ownership $\hat{M}$ and ordering $\hat{D}$
Enforce image denoising, program denoising, grounding, alignment, and cycle consistency losses
At inference, initialize (z_T, s_T) from noise and a prompt-derived scene prior
Jointly sample (z_0, s_0), then decode the final image and scene outputs

Algorithm 2 PSP-DiT training

Require: Training set $\mathcal{D} = \{(x, y, s^\star, M^\star, D^\star)\}$, encoder/decoder $(E_{\mathrm{RAE}}, D_{\mathrm{RAE}})$, coupled denoiser f_θ, grounding head g_θ, ordering head d_θ, frozen recognizer Π_{cyc}

for each minibatch $(x, y, s^\star, M^\star, D^\star) \sim \mathcal{D}$ **do**
  Encode image latent: $z_0 \leftarrow E_{\mathrm{RAE}}(x)$
  Encode scene-program tokens: $S_0 \leftarrow \mathrm{Tok}_{\mathrm{prog}}(s^\star)$
  Sample timestep t
  Corrupt image and program latents: $z_t \sim q_z(\cdot \mid z_0, t)$, $s_t \sim q_s(\cdot \mid S_0, t)$
  Compute prompt tokens $Y \leftarrow \mathrm{Tok}_{\mathrm{text}}(y)$
  Predict joint denoising targets $(\hat{u}_t^z, \hat{u}_t^s) \leftarrow f_\theta(z_t, s_t, Y, t)$
  Predict ownership and ordering $\hat{M}_t \leftarrow g_\theta(z_t, s_t)$, $\hat{D}_t \leftarrow d_\theta(z_t, s_t)$
  Pool instance-level image evidence and compute alignment features
  Decode provisional image $\hat{x}$ from predicted clean latent
  Recover scene structure $\hat{s}_{\mathrm{cyc}} \leftarrow \Pi_{\mathrm{cyc}}(\hat{x})$
  Compute total loss $\mathcal{L}$ from Eq. (6)
  Update θ by gradient descent on $\mathcal{L}$
end for

Algorithm 3 PSP-DiT inference

Require: Prompt y, prompt-to-program prior Π_{init}, coupled denoiser f_θ, grounding head g_θ, ordering head d_θ, RAE decoder D_{RAE}

Construct coarse scene prior: $s_{\mathrm{prior}} \leftarrow \Pi_{\mathrm{init}}(y)$
Initialize noisy image latent $z_T \sim \mathcal{N}(0, I)$
Initialize noisy scene-program latent $s_T \sim \mathcal{N}(s_{\mathrm{prior}}, \Sigma)$
for $t = T, T-1, \ldots, 1$ **do**
  Predict joint denoising targets $(\hat{u}_t^z, \hat{u}_t^s) \leftarrow f_\theta(z_t, s_t, y, t)$
  Update both latent states: $z_{t-1} \leftarrow \Phi_z(z_t, \hat{u}_t^z, t)$ and $s_{t-1} \leftarrow \Phi_s(s_t, \hat{u}_t^s, t)$
  Predict ownership and ordering: $\hat{M}_{t-1} \leftarrow g_\theta(z_{t-1}, s_{t-1})$ and $\hat{D}_{t-1} \leftarrow d_\theta(z_{t-1}, s_{t-1})$
end for
Decode final image $\hat{x} \leftarrow D_{\mathrm{RAE}}(z_0)$
Decode scene outputs from $(s_0, \hat{M}_0, \hat{D}_0)$
return generated image $\hat{x}$, final scene program s_0, ownership maps $\hat{M}_0$, ordering $\hat{D}_0$

A.10 Evaluation References and Systems Context

A.10.1 Comparison with Recent Generative Systems

We extend the evaluation to recent direct-generation systems and GEMS-based agentic generation [He et al., 2026]. All results use the GenEval 2 Rewritten protocol with four fixed seeds, four images per prompt, prompt-level averaging, and no best-of-N selection. For agentic comparisons, both generators are evaluated within the same GEMS multi-round loop.

Table 4: **Comparison with recent generative systems.** GenEval 2 Rewritten scores ($\times 100$). Direct generation and GEMS-based multi-round generation are reported separately; their inference budgets are not equivalent.

Model	Generation procedure	GE2 ↑
Z-Image-Turbo	Direct	31.2
Qwen-Image-2512	Direct	33.4
Ours-FlatText 3B	Direct	34.7
PSP-DiT 3B	Direct	**41.5**
GEMS + Z-Image-Turbo	Multi-round	63.5
GEMS + PSP-DiT 3B	Multi-round	**69.1**

PSP-DiT 3B improves over the strongest tested recent direct-generation baseline, Qwen-Image-2512, by 8.1 GE2 points. Within the matched GEMS agent loop, replacing Z-Image-Turbo with PSP-DiT 3B improves GE2 from 63.5 to 69.1 (+5.6 points). These comparisons distinguish improvements to the underlying generator from gains due to external multi-round search.

A.10.2 Public baseline positioning

Table 5 reports literature-provided public reference numbers used only for positioning. These values are not the primary matched comparisons in the main paper, but they help contextualize the reproduced benchmark ranges.

Table 5: **Appendix positioning tables using literature-reported public numbers. Top:** overlapping public open-baseline subset with matched coverage across GenEval 2, PSG-Score, and DetailMaster. **Bottom:** frontier public-reference systems on SANEval (reported SANEval-Simple averaged score).

Model	GenEval 2 ↑	PSG-Score ↑	DetailMaster Avg. ↑
FLUX.1-dev	29.5	0.62	63.6
SD3.5-Large	19.0	0.60	59.6

Model	SANEval (Averaged) ↑
Imagen 3.0	0.4371
Imagen 4.0	0.4210
Imagen 4.0 Ultra	0.4992
Nano Banana	0.4743
Seedream 3.0	0.4929
GPT Image 1	0.4763

A.10.3 Systems anchor for FLUX.1-DEV

To contextualize the efficiency measurements in Table 1, we report a published external systems anchor for FLUX.1-DEV in Table 6. A recent benchmark reports 49.7 s/image and 32.8 GB peak VRAM on RTX A6000 for 1024×1024 generation with 50 denoising steps. We include this reference only as a calibration point: unlike our H200 measurements, it is not directly comparable because latency and memory depend strongly on hardware, precision, attention backend, and compilation stack.

A.11 Skill Aggregation Details

Table 7 gives the exact mapping used to construct the compositional-skill breakdown in Table 1. Each reported skill corresponds to a single benchmark-native subset and is linearly rescaled to $[0, 100]$

Table 6: **Published external systems anchor for FLUX.1-DEV.** This reference is provided only for context; unlike our H200 measurements, it is not directly comparable because latency and VRAM depend strongly on hardware and software stack.

Setup	Latency (s/img) ↓	Peak VRAM (GB) ↓
FLUX.1-DEV RTX A6000, 1024×1024, 50 steps	49.7	32.8

using the benchmark's documented range. Accordingly, no nontrivial within-skill weighting is required in the current paper. The Avg. column in Table 1 is the unweighted mean over the five skill columns.

For SANEval-Simple and the GenEval 2 role-sensitive subset, exact prompt counts are taken from the released evaluation splits used in our runs. For public context, GenEval 2 contains 800 templated prompts overall and discusses 553 rewritten prompts, while DetailMaster contains 4,116 prompts.

Table 7: **Skill-to-benchmark mapping used for Table 1.** Each skill score is computed from the listed benchmark-native subset and linearly rescaled to $[0, 100]$ using the benchmark's documented range.

Skill	Contributing sub-benchmark(s)	Prompt-count source
Counting	SANEval-Simple: Numeracy	released SANEval-Simple numeracy split
Attr. Binding	SANEval-Simple: Attribute Binding	released SANEval-Simple attribute-binding split
Spatial Rel.	SANEval-Simple: Spatial Relations	released SANEval-Simple spatial-relations split
Role Rel.	GenEval 2 (Rewritten): transitive-verb / role-sensitive subset	released GenEval 2 rewritten role-sensitive subset
Long Prompts	DetailMaster: full benchmark	public benchmark total: 4,116 prompts

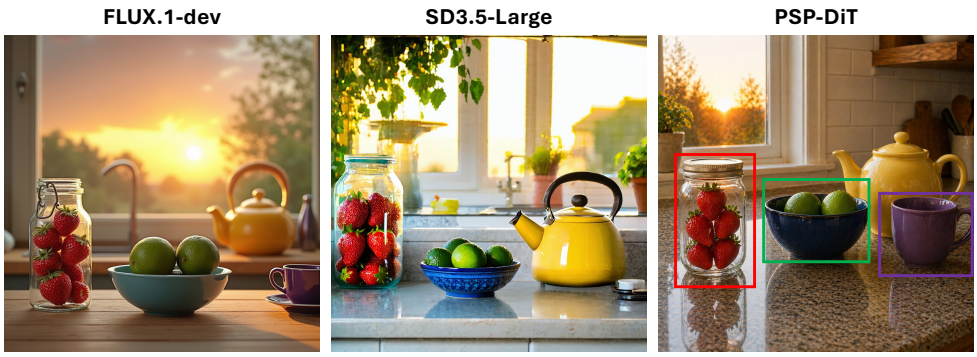
A.12 Qualitative Prompts and Additional Stress Cases

Table 8 lists the full prompts used for the main qualitative stress cases in Figure 3.

Table 8: **Full prompts used for the main qualitative stress cases in Figure 3.** The same prompt, random seed, and matched inference budget are used across FLUX.1-dev, SD3.5-Large, and PSP-DiT for each row.

Row	Skill stress	Full prompt
1	Counting; attribute ownership; left–center–right ordering	A photorealistic tabletop scene, viewed from the front, with exactly three ceramic mugs in a straight row: a red mug on the left, a blue mug in the center, and a yellow mug on the right. A silver spoon is inside only the blue mug, and a small green apple sits directly in front of the red mug.
2	Attribute binding; object–support assignment	A photorealistic close-up of a green apple on a white plate to the left of a red banana on a black plate, with a wooden spoon placed between the two plates.
3	Role-sensitive relation; spatial ordering; object ownership	A photorealistic park scene where a brown dog is chasing a white cat across a path; the dog is visibly behind the cat, the cat is carrying a red ball in its mouth, and a blue frisbee lies on the grass to the right of both animals.
4	Long structured prompt; roles; counting; attribute ownership	A photorealistic classroom: a teacher in a gray sweater points at a square on the blackboard while a student in a red shirt holds a blue notebook; another student in a yellow shirt sits behind the desk, and exactly two pencils lie on the desk.

The following pages extend Figure 3 with additional matched qualitative prompts covering counting, attribute binding, spatial relations, object interaction, and long structured descriptions. Each row uses the same prompt, random seed, and matched inference budget across FLUX.1-dev, SD3.5-Large, and PSP-DiT. Colored boxes, where shown, are visual highlights.



*"A photorealistic kitchen counter at sunset: on the left, a transparent jar contains **exactly five red strawberries**; in the center, a blue ceramic bowl holds **exactly two green limes**; on the right, a yellow teapot sits behind **a purple cup**."*

Figure 4: **Additional qualitative compositional stress cases.** These examples extend Figure 3 to additional prompts covering counting, attribute binding, spatial relations, object interaction, and long structured descriptions.



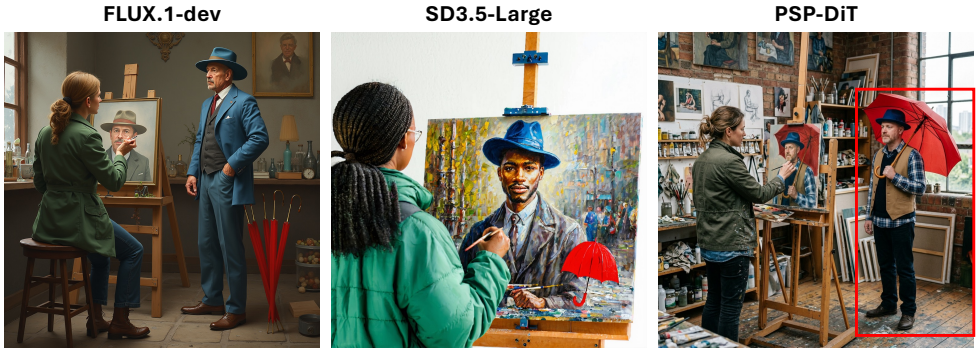
*"A photorealistic shelf scene with four objects arranged left to right: a cactus in a red pot, **a clock with a blue frame**, **a teddy bear wearing a green scarf**, and a white vase holding exactly two yellow flowers."*

Figure 5: Additional qualitative compositional stress cases (continued).



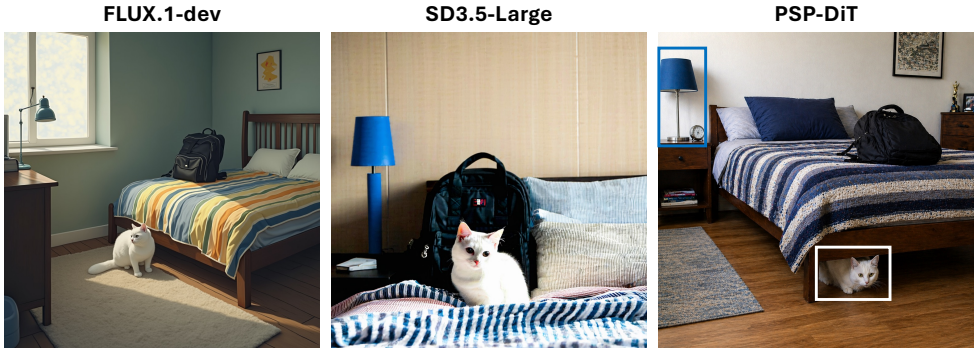
*"A photorealistic city intersection viewed from the sidewalk: a yellow taxi is beside a blue bus, **a red bicycle leans against a lamppost to the left of the taxi**, and exactly **three orange traffic cones form a triangle behind the bus**."*

Figure 6: Additional qualitative compositional stress cases (continued).



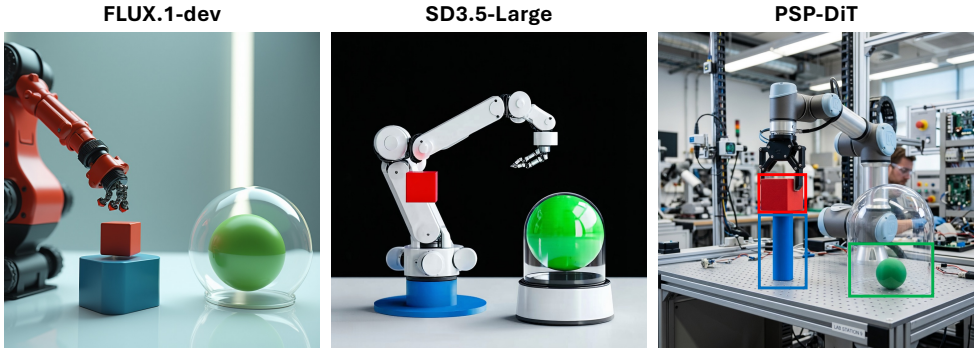
*"A photorealistic art studio: a woman in a green jacket is painting a portrait of a man in a blue hat; **the man stands to the right of the easel and holds a red umbrella**."*

Figure 7: Additional qualitative compositional stress cases (continued).



*"A photorealistic bedroom scene: a striped blanket is on the bed, **a white cat is under the bed**, a black backpack is on top of the bed, and **a blue lamp stands to the left of the bed**."*

Figure 8: Additional qualitative compositional stress cases (continued).



*"A photorealistic robotics lab: a small robot arm places **a red cube** on top of a **blue cylinder**, while **a green sphere sits under a transparent glass dome to the right of the robot arm**."*

Figure 9: Additional qualitative compositional stress cases (continued).

A.13 Program-Latent Diagnostics and Controlled Refinement

We examine whether the scene-program latent progressively encodes compositional structure and enables targeted control. Linear probes are trained on frozen PSP-DiT representations using disjoint training and held-out prompt sets; probe gradients do not update the generator. We additionally evaluate direct program-latent interventions and compare fixed conditioning against joint refinement under identical backbone, data, losses, seeds, and inference budgets.

Table 9: **Program-latent diagnostics and controlled refinement.** (a) Held-out linear-probe accuracy across denoising stages. (b) Targeted intervention success and preservation of unrelated image content. (c) Controlled fixed-versus-joint program refinement under exact and corrupted initialization. All scores are percentages except GE2, reported on the benchmark's $\times 100$ scale.

(a) Progressive program decoding

Decoded property	Early	Middle	Late
Object identity	62.1	84.7	96.0
Attribute ownership	48.5	77.9	91.8
Ordered relations	41.2	72.4	87.6
Cardinality	59.6	86.1	94.2

(b) Program-latent interventions

Intervention	Target success	Preserved
Red $\leftrightarrow$ blue attribute	89.3	93.1
Left-of $\leftrightarrow$ right-of	84.7	91.8
Count 2 $\leftrightarrow$ 3	87.9	92.6

(c) Fixed conditioning vs. joint refinement

Initial program	GE2 ↑		Gain	Final program F1 ↑	
	Fixed	Joint		Fixed	Joint
Exact	36.6	38.2	+1.6	94.7	96.1
10% node/edge drop	32.8	36.9	+4.1	80.5	91.4
10% relation corrupt.	31.9	35.4	+3.5	78.2	88.9

The probes show progressive recovery of object identity, attribute ownership, ordered relations, and cardinality. Interventions selectively change the targeted property while largely preserving unrelated content. In the controlled comparison, joint refinement improves GE2 by 1.6 points with exact initialization and by 3.5–4.1 points under corrupted programs, alongside improvements in final program F1. Unlike fixed conditioning, the evolving latent can revise imperfect initial structure.

A.14 Timestep-Wise Scene-Program Visualization

Figure 10 illustrates the evolution of image predictions, cross-attention, and instance ownership during joint denoising. We examine a single sampling trajectory at steps 10, 25, and 45 of a 50-step schedule.

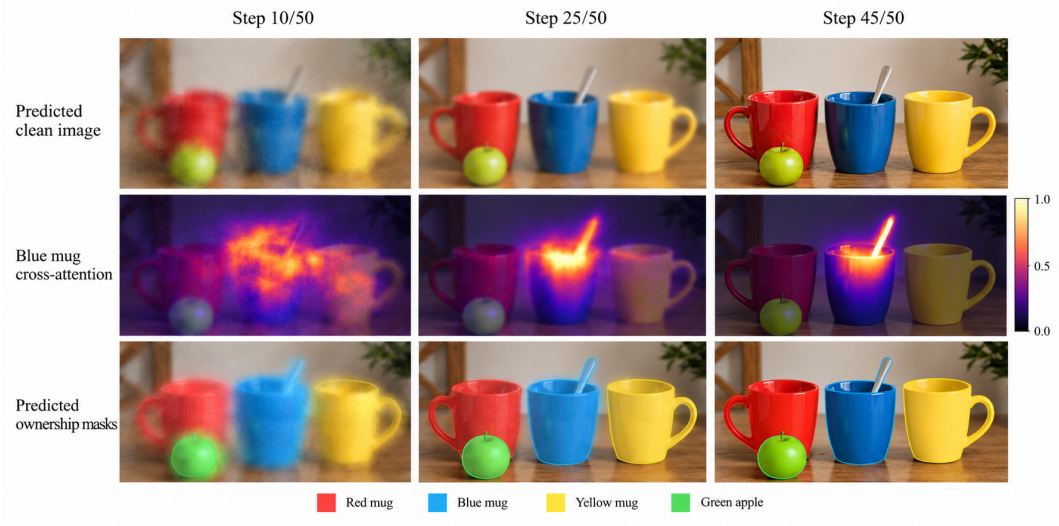


Figure 10: **Timestep-wise scene-program evolution.** PSP-DiT jointly refines image and scene-program representations across a 50-step sampling trajectory. Top: predicted clean-image estimates at steps 10, 25, and 45. Middle: image-to-program cross-attention associated with the blue-mug instance token, displayed using a shared normalization to $[0, 1]$ across the three stages. Bottom: predicted instance-ownership maps for the red, blue, and yellow mugs and the green apple. Consistent instance colors are maintained across timesteps. The visualization illustrates the progressive resolution of scene appearance, attention localization, and visual ownership.

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 abstract and introduction state the core contribution, scope, and matched empirical claims; these are supported by the method in Section 3 and the results in Section 4.

   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 includes a limitations paragraph after the conclusion, noting that the current study focuses on text-to-image generation and leaves text-to-video extension to future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results or proofs; the contribution is architectural and empirical.

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 main evaluation protocol is described in Section 4.1; evaluator, dataset, algorithmic, and skill-aggregation details are provided in Appendices A.5, A.9, and A.11.

   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: The submission does not include public code or checkpoints. The paper provides architecture, algorithmic, evaluation, and prompt details in Section 3, Section 4.1, and the appendix to support reproducibility.

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: Training/evaluation settings are summarized in Section 4.1, with algorithmic details in Appendix A.9 and evaluator/dataset details in Appendix A.5.

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: Human preference results are reported with confidence intervals computed by nonparametric bootstrap over prompts with 10,000 resamples; see Appendix A.6 and Table 3. Other quantitative tables report deterministic benchmark scores under the matched evaluation protocol described in Section 4.1.

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: Appendix A.4 reports the compute hardware, precision, runtime setup, GPU counts for the main training runs, optimization steps, total project GPU-hours, and GPU-hours for the experiments reported in the main paper. Table 1 also reports measured latency and VRAM under the matched H200 inference setup.

   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 work uses standard generative-model evaluation protocols and public benchmarks/models, preserves anonymity, and does not involve releasing unsafe datasets or deployed systems.

   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: The work advances text-to-image generation and discusses scope/limitations; as with stronger generative models, potential misuse risks are those generally associated with image synthesis.

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: [N/A]

Justification: The paper does not release a model, dataset, or deployment interface requiring additional release safeguards.

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: The paper cites the benchmark, model, dataset, and evaluator assets used in the evaluation, including model cards and repositories where applicable; no new redistributed assets are introduced.

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: The paper does not release a new dataset, codebase, benchmark, or model asset as part of the submission.

    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: Appendix A.6 describes the blinded pairwise human preference study, including prompt source, number of prompts, raters per pair, displayed instruction, randomization, individual-rater win-rate reporting, compensation status, and participant-risk statement.

    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: Appendix A.6 states that the preference study involved only low-risk judgments over generated images, participation was voluntary, no compensation was provided, and no potential risks were incurred by participants. The study did not collect sensitive personal information and did not require formal IRB review under the applicable institutional policy.

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 does not use LLMs as an important, original, or non-standard component; any routine writing or editing assistance does not affect the scientific method.

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.