ISAC: Training-Free Instance-to-Semantic Attention Control for Multi-Instance Generation

Sanghyun Jo[1,2⋆†], Wooyeol Lee[2⋆], Ziseok Lee[2⋆], Jonghyun Choi[2], Jaesik Park[2], and Kyungsu Kim[2†]

[1] OGQ, Seoul, Korea
[2] Seoul National University, Seoul, Korea

Abstract. Recent open-weight text-to-image (T2I) diffusion models still struggle with multi-instance prompts, often omitting or merging instances and mixing semantics among similar objects. We trace these failures to early denoising steps, before instance boundaries are reliably stabilized. Existing training-free guidance is largely driven by cross-attention or other token-conditioned semantic signals. Such guidance can separate concepts at the token level, but largely assumes that distinct instance regions have already emerged. In early denoising steps, it cannot reliably carve out these regions, so count failures and semantic mixing persist. By contrast, self-attention exposes class-agnostic instance layouts during early denoising. To exploit this asymmetry, we propose **ISAC** (**I**nstance-to-**S**emantic **A**ttention **C**ontrol), a training-free, model-agnostic objective that first stabilizes self-attention layouts and then binds cross-attention semantics within them, without fine-tuning or external vision models. Across T2I-CompBench, HRS-Bench, and our newly curated IntraCompBench, ISAC consistently outperforms prior training-free methods. Furthermore, ISAC enhances layout-to-image controllers by refining coarse, overlapping bounding boxes into dense instance masks. Code and IntraCompBench are available at `https://shjo-april.github.io/ISAC`.

Keywords: hierarchical · text-to-image · diffusion · training-free · instance-to-semantic · attention

1 Introduction

Text-to-image (T2I) generative models [8, 18, 27, 66] now produce highly realistic images from text prompts. In particular, recent commercial systems (*e.g.*, Nano Banana 2 [27] and GPT-Image [56]) establish a strong empirical upper bound on multi-instance, multi-attribute prompts. Such performance is often supported by massive proprietary data pipelines [1, 25] and expensive MLLM-driven reasoning loops, such as verifying interim images and refining the composition [26]. Although few failure cases are observable in these systems (see Appendix C), their closed

⋆ Equal contribution.
† Corresponding authors: `shjo.april@gmail.com`, `kyskim@snu.ac.kr`

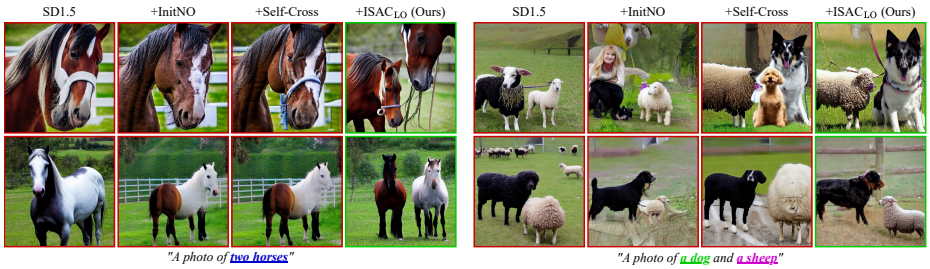


Fig. 1: Qualitative comparison for representative multi-instance prompts. Generated samples are compared across the baseline SD1.5 [66], prior methods (InitNO [28], Self-Cross [62]), and our ISAC$_{\text{LO}}$ (ISAC with latent optimization). Text prompts are provided below each group.

nature makes them difficult to analyze or reproduce. This limits the community's ability to diagnose failure modes and develop transferable solutions.

By contrast, recent open-weight diffusion backbones (*e.g.*, Flux.1-dev [8] and Qwen-Image [77]) improve accessibility and enable reproducible inference-level analysis by releasing model weights, architectures, and inference pipelines. Yet multi-instance compositionality remains unreliable: models often omit or merge requested objects (*count failures*) and leak attributes across instances (*semantic mixing*). These failures are especially pronounced when several instances are semantically similar, which is common in real-world scenarios. Moreover, their training data and full training recipes often remain unavailable, making retraining-based improvements hard to reproduce or transfer across backbones. This motivates training-free, inference-time guidance that improves instance separation without additional data, fine-tuning or external vision models.

Recent training-free T2I guidance methods [13, 14, 28, 33, 62] have aimed to improve compositional generation without retraining. At a high level, they mainly correct the semantic side of generation, for example by reducing interference among prompt tokens. Such correction can be effective when the model has already formed separate object regions. However, semantic correction alone does not explicitly guarantee that each requested instance is formed as a distinct region, especially for same-class or semantically similar objects. Figure 1 illustrates this limitation. For "two horses," SD1.5 and prior semantic-driven methods often omit or merge instances. For "a dog and a sheep," they often confuse the animal identities or fail to keep the requested objects spatially distinct.

To assess this limitation more systematically, we analyze semantic overlap across class pairs grouped by supercategory in Fig. 2. We construct instance-aware semantic masks by injecting structural cues from self-attention into token-level semantics via Eq. 4. We then quantify semantic mixing as the Dice overlap between the two masks for a requested pair of classes. The overlap is consistently higher for pairs within the same supercategory, indicating that token-level semantic footprints tend to cover multiple semantically similar instances simultaneously.

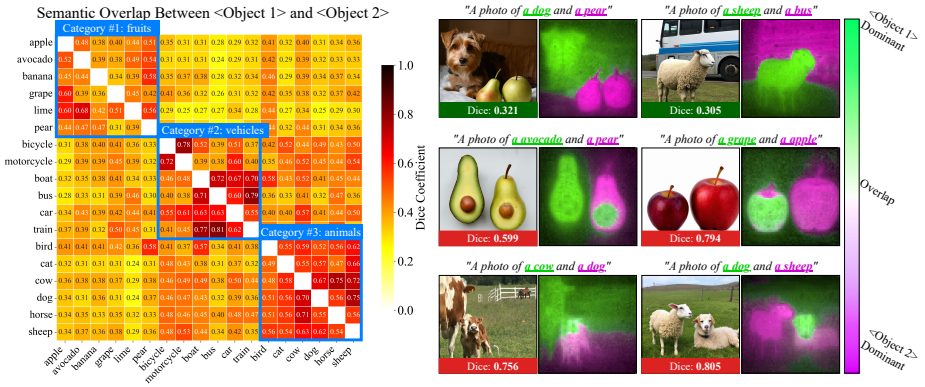


Fig. 2: Overlap of instance-aware semantic masks. For images generated by SD3.5-M [23] using prompts of the format "A photo of a <object1> and a <object2>", instance-aware semantic masks are computed via Eq. (4) and averaged across all denoising timesteps. The Dice coefficient measures the overlap between the two object-specific masks. *Left:* Pairwise overlap heatmap, where blue boxes denote supercategories. *Right:* Example generations and signed, normalized mask-difference maps, where color denotes the dominant mask and intensity indicates the normalized magnitude.

These overlaps show that token-level semantic separation may fail to determine which pixels belong to which instance when boundaries are ambiguous, as semantic cues can spill across multiple similar objects. This motivates an instance-first hierarchy that first stabilizes instance regions from structure rather than class-token semantics, and then binds semantics within each region.

We instantiate this hierarchy as ISAC (**I**nstance-to-**S**emantic **A**ttention **C**ontrol), a training-free hierarchical objective for latent optimization or latent selection. ISAC separates structural formation from semantic binding along the denoising trajectory. Phase 1 ($\mathcal{L}_{\text{ins}}$) exploits the observation that self-attention already exposes emerging instance layouts in early steps [39], when token-conditioned semantics are still coarse and entangled. It derives N class-agnostic instance layouts from structural cues and penalizes their overlap, encouraging the requested number of separated regions before assigning class labels. Phase 2 ($\mathcal{L}_{\text{sem}}$) injects these stabilized structural signals into cross-attention maps to form instance-aware semantic masks, so that classes and attributes are bound within each instance without leaking across regions. This plug-and-play formulation supports both text-to-image backbones [3,8–10,15,16,23,61,66,70,77] and layout-to-image controllers [48]. To evaluate the challenging regime where these failures are most frequent, we also introduce IntraCompBench, a benchmark designed to stress-test intra-supercategory compositions.

In summary, our main contributions are:

– We provide a quantitative diagnosis of semantic mixing via Dice overlap between instance-aware semantic masks, revealing substantially higher overlap for class pairs within the same supercategory.

Table 1: Conceptual comparison of methods for multi-instance text-to-image generation.

Method	First preserve instance structure	Separate semantic masks	Diffusion-backbone agnostic	Require instance counts at inference	No fine-tuning or extra counting model
TEBOpt NeurIPS'24 [14]	✗	✓	✗	✗	✓
DOS AAAI'26 [12]	✗	✓	✗	✗	✓
A&E SIGGRAPH'23 [13]	✗	✓	✓	✗	✓
InitNO CVPR'24 [28]	✗	✓	✓	✗	✓
SynGen NeurIPS'23 [64]	✗	✓	✓	✗	✓
CONFORM CVPR'24 [55]	✗	✓	✓	✗	✓
Self-Cross CVPR'25 [62]	✗	✓	✓	✗	✓
CountGen CVPR'25 [7]	✗	✓	✗	✓	✗
Counting Guidance WACV'25 [40]	✗	✓	✗	✓	✗
ISAC (Ours)	✓	✓	✓	✓	✓

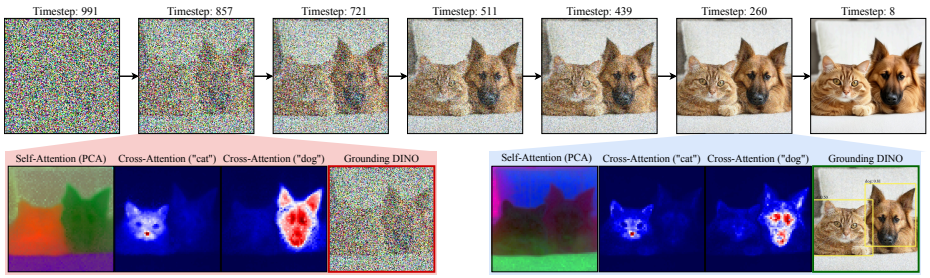


Fig. 3: Dynamics of text-to-image diffusion models. In the early stages of diffusion, instance structures emerge [39] while semantics remain underdeveloped. In later diffusion steps, instance structures are stabilized, and semantic refinements happen. Because detection models (*e.g.*, [53]) rely on strong semantic cues, they are effective only in later steps. We use a prompt of "A photo of a cat and a dog" on SD3.5-M [23].

- We introduce **ISAC**, a training-free, model-agnostic objective that enforces an instance-to-semantic hierarchy by separating instance formation from semantic assignment.
- We propose **IntraCompBench**, a benchmark for explicit 2–5-instance counting and intra-supercategory multi-class compositions.
- We demonstrate broad gains across T2I-CompBench [34], HRS-Bench [4], and IntraCompBench, over multiple diffusion backbones and both text- and layout-to-image generation. ISAC improves multi-object metrics by at least 1.9× over the baseline [66] and outperforms prior counting-supervised methods [7, 40] by at least 10% accuracy without additional training.

2 Related Work

Training-free text-to-image guidance. While text embedding approaches [12, 14, 24, 33], such as TEBOpt [14] and DOS [12], mitigate semantic bias via embedding optimization or token reordering, they still face two key limitations. First, without explicit spatial information, text embeddings hinder accurate instance formation. Second, reliance on the CLIP [63] architecture limits compatibility with state-of-the-art architectures (*e.g.*, Flux.2 [9], Qwen-Image [77])

that leverage modern advances in large language models (LLMs). Alternatively, cross-attention (CA) reveals the spatial footprint of textual semantics [29], enabling spatial guidance. Attend-and-Excite [13] boosts attention peaks to recover neglected objects, while SynGen [64] and CONFORM [55] separate and bind CA maps using contrastive objectives and parser-derived relations. InitNO [28] and Self-Cross [62] additionally add structural cues from self-attention (SA). However, these methods use CA token semantics to select, aggregate, or guide structural SA maps. Consequently, their structural grouping remains conditioned on CA semantics. They can separate SA maps associated with different CA tokens, but are not designed to disambiguate multiple instances that share the same semantic token. Thus, when early CA maps already merge same-class instances or activate only on object parts, the resulting SA-based structures inherit this ambiguity. In contrast, ISAC is designed to explicitly establish the instance structure first and subsequently bind semantics. This strategy ensures reliable instance discrimination with minimal cues (see Tab. 1).

Layout-to-image methods. Token-level semantics in text prompts lack instance discriminative cues, making additional layout conditions (*e.g.*, bounding boxes) appealing. Box layouts are lightweight and, combined with the spatial understanding of large language models (LLMs), support a two-stage pipeline that first generates a bounding-box layout from text and then conditions image synthesis on that layout [49, 82]. While state-of-the-art controllers provide dense layout control [11, 35, 48, 74, 81, 84, 85], they often struggle with overlapping layouts [46, 80]. Training-free layout-to-image methods [6, 17, 21, 41, 45, 60, 69, 78, 79] help, yet they do not ensure instance discrimination. Constraining attention within each box either by excluding background [60] or other instances' layouts [21] only separates semantic regions rather than instance structures. In contrast, ISAC first evaluates instance structures from internal attention without layout priors, then assigns semantics. This addresses controllers' failures under overlapping boxes by carving coarse box layouts into dense instance masks.

Count-supervised text-to-image methods. Given the limits of box layouts, recent work pursues instance-level control with minimal supervision via instance counts. One approach leverages a pretrained vision model to enforce counts [40], but because such models rely on strong semantic cues, they are ineffective in early diffusion steps when instances form (see Fig. 3). Another replaces boxes with automatically generated dense masks to guide instance separation [7, 20], yet this depends on a fine-tuned mask generator and extensive training to cover broad vocabularies and compositions.

Evaluation and benchmarks. To evaluate multi-object generation, T2I-CompBench [34] and HRS-Bench [4] are widely adopted. However, they do not isolate intra-category cases where (i) *count failures* and (ii) *semantic mixing* are most prevalent. Evaluation set from [13] and SSD [62] target this issue but

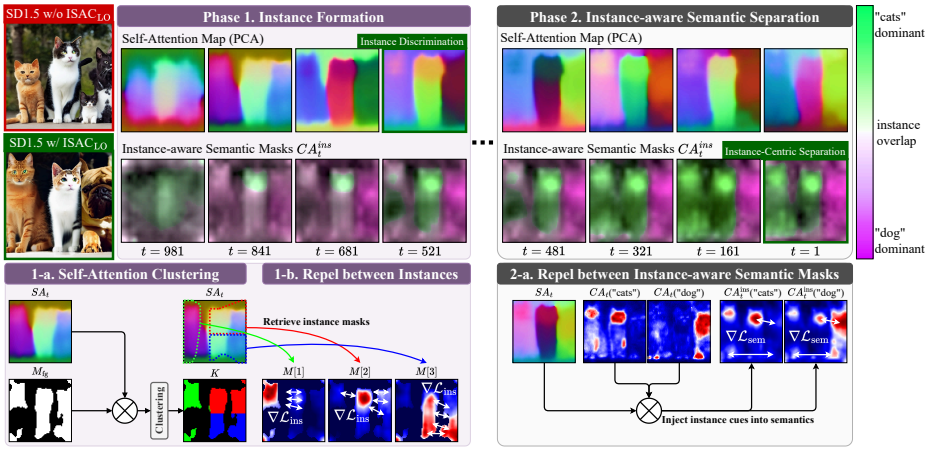


Fig. 4: Overview of ISAC. Guided by diffusion dynamics, ISAC computes a hierarchical objective in two phases. Phase 1 (Sec. 3.2) clusters self-attention to shape N class-agnostic instance layouts, repelling overlaps to establish clean boundaries early in the trajectory. Phase 2 (Sec. 3.3) then inject these reliable instance structures into cross-attention to align semantic evidence, using a repel-and-bind loss to prevent cross-instance semantic mixing. An instance-to-semantic schedule (Sec. 3.4) seamlessly transitions the objective from Phase 1 to Phase 2.

benchmark only simple 2–3-object compositions, with SSD limited to a small prompt set (31 two-object and 21 three-object prompts). We address this gap with IntraCompBench, a comprehensive benchmark that stress-tests similar-object generation across 2–5-object compositions with diverse prompt combinations.

3 Method

ISAC targets two core failure modes of multi-object generation, *count failures* and *semantic mixing*, which stem from vague instance boundaries. Guided by evidence that coarse structure emerges before fine-grained semantics in diffusion [29, 44], we design ISAC as a hierarchical, instance-first, training-free objective function. Figure 4 illustrates how Phase 1 (Sec. 3.2) shapes class-agnostic instance layouts from self-attention and how Phase 2 (Sec. 3.3) aligns semantic evidence to these layouts, yielding instance-aware semantics that prevent cross-instance mixing.

3.1 Background

Text-to-image diffusion models. Diffusion models learn to reverse a forward noising process that gradually corrupts a sample X_0 into noise X_T, conditioned on a text embedding $\mathcal{T} \in \mathbb{R}^{L \times d}$. At inference time, a neural network ϵ_θ iteratively denoises $X_T \sim \mathcal{N}(0, I)$ to obtain X_0. In text-to-image models, denoising is typically performed in latent space; a VAE decoder $\mathcal{D}$ then maps X_0 to pixel

space, yielding the final image $I_0 = \mathcal{D}(X_0)$. ISAC is model-agnostic: it only requires access to X_t and the model's attention maps.

Prompt notation. Following [7, 40], ISAC targets the multi-instance setting in which each prompt explicitly specifies instance counts. For each prompt, we parse class tokens $\{\tau_i\}_{i=1}^{k}$, per-class instance counts $\{n_i\}_{i=1}^{k}$, and optional attributes $\{\chi_{i,j}\}$. Ambiguous-count prompts, where the requested number of instances cannot be determined from the prompt, are outside the scope of this work. The total number of requested instances is then given by $N = \sum_i n_i$. We use an LLM-based parser to automatically obtain $(\{\tau_i\}, \{n_i\}, \{\chi_{i,j}\})$. Additional details and examples are provided in Appendix A.

Attention. At each denoising timestep t, the denoiser ϵ_θ uses self-attention (SA) to capture spatial relations within the latent $X_t \in \mathbb{R}^{HW \times d}$ and cross-attention (CA) to align X_t with text embeddings $\mathcal{T}$. Both compute maps from queries and keys obtained via learned projections (W_Q, W_K). For SA, $Q_t^{\text{self}} = X_t W_Q^{\text{self}}$ and $K_t^{\text{self}} = X_t W_K^{\text{self}}$. For CA, $Q_t^{\text{cross}} = X_t W_Q^{\text{cross}}$ and $K_t^{\text{cross}} = \mathcal{T} W_K^{\text{cross}}$. For a head of width d_h at layer l, the per-head attention maps are

$$SA_l^h(X_t) = \text{softmax}(Q_t^{\text{self}} {K_t^{\text{self}}}^\top / \sqrt{d_h}) \in [0,1]^{HW \times HW}, \tag{1}$$

$$CA_l^h(X_t, \mathcal{T}) = \text{softmax}(Q_t^{\text{cross}} {K_t^{\text{cross}}}^\top / \sqrt{d_h}) \in [0,1]^{HW \times L}. \tag{2}$$

We register hooks on all attention layers to read out SA and CA without altering computation. Let M be the number of attention layers and h_l the number of heads at layer l. For each timestep t, we obtain a single SA/CA pair by averaging the attention maps over all layers and heads:

$$SA_t = \frac{1}{\mathcal{N}} \sum_{l,h} SA_l^h(X_t),\ CA_t = \frac{1}{\mathcal{N}} \sum_{l,h} CA_l^h(X_t, \mathcal{T}) \tag{3}$$

where $\mathcal{N} = \sum_{l=1}^{M} h_l$. For U-Nets where attention maps vary in spatial resolution, we bilinearly upsample each map to the highest resolution and take the average (See Appendix A).

3.2 Phase 1: Instance Formation

Motivated by [39], we observe that pixels belonging to the same instance exhibit higher mutual attention, whereas pixels from different instances attend to each other less. This suggests that the N most discriminative pixel clusters reveal N disjoint instance layouts. We translate this property into a guidance objective by (i) clustering self-attention into N groups and (ii) penalizing overlap between the resulting masks. This loss strengthens attention within each instance and suppresses attention outside its boundary.

Self-attention to N clusters. Self-attention (SA) encodes instance structure, but also assigns non-trivial mass to background regions, so clustering it over the whole image can split or merge instances. We therefore first build a foreground gate M_{fg} from semantic maps and then cluster SA only within this foreground. Given accumulated maps SA_t and CA_t from Eq. (3), we leverage the instance structure encoded in SA_t to form instance-aware semantic masks:

$$CA_t^{\text{ins}} \;=\; SA_t\, CA_t \;\in\; [0,1]^{HW\times L}, \tag{4}$$

where column j highlights the region most responsive to token $\mathcal{T}[j]$. We binarize each column by its mean μ_j and define M_{fg} as the union over class tokens,

$$CA_t^{\text{bin}} \leftarrow \texttt{Binarize}(CA_t^{\text{ins}}), \tag{5}$$

$$M_{\text{fg}} = \bigcup_{\mathcal{T}[i]\in\{\tau_j\}_{j=1}^{k}} CA_t^{\text{bin}}[:,i] \in \{0,1\}^{HW}. \tag{6}$$

Let $\mathcal{I}{=}\{p : M_{\text{fg}}[p]{=}1\}$ and $F{:=}|\mathcal{I}|$. We restrict SA to these foreground positions and cluster its rows into N components (*e.g.*, K-means on SA features, concatenated with normalized coordinates $(x,y)\in[-1,1]^2$ for spatial coherence). With one-hot assignments $K\in\{0,1\}^{F\times N}$ (no gradient through K), the resulting instance masks are

$$M \;=\; SA_t[\mathcal{I},\mathcal{I}]\,\texttt{stopgrad}(K) \;\in\; [0,1]^{F\times N}. \tag{7}$$

This yields instance masks that highlight pixels attending strongly within the same cluster and weakly outside, leading to sharper instance boundaries. Further details of the instance clustering procedure are provided in Appendix A.

Repel guidance between instance masks. From Eq. (7) we obtain instance masks $M[1],\ldots,M[N]$ over foreground pixels. To separate instances, we penalize their worst local overlap using the *maximum pixel-wise overlap* (MPO):

$$\texttt{MPO}(A,B) \;=\; \max_{p\in\{1,\ldots,F\}} A[p]\cdot B[p], \tag{8}$$

and define the instance separation loss as the maximum MPO over all mask pairs,

$$\mathcal{L}_{\text{ins}}(X_t) \;=\; \max_{1\le i<j\le N} \texttt{MPO}\big(M[i],\,M[j]\big). \tag{9}$$

Within each step we treat K as `stopgrad`, so gradients flow only through SA_t.

3.3 Phase 2: Instance-aware Semantic Separation

After Phase 1 stabilizes sharp instance boundaries in SA_t, the propagated semantic maps $CA_t^{\text{ins}} = SA_t\, CA_t$ in Eq. (4) are activated within spatially partitioned instances. Building on this structure, in Phase 2, we apply a repel-and-bind loss $\mathcal{L}_{\text{sem}}$: tokens referring to different instances are pushed apart, while tokens

describing the same instance are pulled together, reinforcing clean semantic separation per instance.

Let P_{repel} denote pairs of token indices that should remain distinct (*e.g.*,, different classes/instances), and P_{bind} denote pairs that should co-activate (*e.g.*, class/attribute within the same instance). We use MPO as a sharp, local measure of overlap between the corresponding semantic maps:

$$\mathcal{L}_{\text{repel}}(X_t) = \max_{(a,b)\in P_{\text{repel}}} \left[+ \texttt{MPO}\big(CA_t^{\text{ins}}[:,a],\, CA_t^{\text{ins}}[:,b]\big)\right] \tag{10}$$

$$\mathcal{L}_{\text{bind}}(X_t) = \max_{(a,b)\in P_{\text{bind}}} \left[1 - \texttt{MPO}\big(CA_t^{\text{ins}}[:,a],\, CA_t^{\text{ins}}[:,b]\big)\right] \tag{11}$$

We combine them as a repel-and-bind objective:

$$\mathcal{L}_{\text{sem}}(X_t) \;=\; \mathcal{L}_{\text{repel}}(X_t) + \mathcal{L}_{\text{bind}}(X_t) \tag{12}$$

While using such token relations is common [55, 64], our contribution lies in binding these relations to the *instance-aware* masks formed in Phase 1 (Sec. 3.2).

3.4 Instance-to-Semantic Loss Schedule

Combining both phases, we define the per-step ISAC objective as

$$\mathcal{L}_{\text{ISAC}}(X_t, t) \;=\; \lambda_{\text{ins}}(t)\,\mathcal{L}_{\text{ins}}(X_t) \;+\; \lambda_{\text{sem}}(t)\,\mathcal{L}_{\text{sem}}(X_t), \tag{13}$$

where $\lambda_{\text{ins}}(t)$ and $\lambda_{\text{sem}}(t)$ control the relative emphasis on instance layout versus semantics over the diffusion trajectory; in practice, we simply set $\lambda_{\text{ins}}(t) = t/T$ and $\lambda_{\text{sem}}(t) = 1 - t/T$ so that early steps focus on instance formation and later steps focus on semantic refinement.

Following prior work [13, 55, 62, 64], we primarily use ISAC for latent optimization (ISAC_{LO}) as summarized in Algorithm 1. ISAC is also compatible with other guidance schemes, such as latent selection (ISAC_{LS}; see Sec. 4.3).

Algorithm 1: ISAC with Latent Optimization (ISAC_{LO})

Input: Prompt $\mathcal{T}$, Model ϵ_θ, decoder $\mathcal{D}$, step size η
Output: Image I_0

$X_T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \ldots, 1$ **do**
  `Call` Denoise$(X_t, \mathcal{T}, \epsilon_\theta, t)$ `with Hooks` $\rightarrow SA_t, CA_t$
  $CA_t^{\text{ins}} \leftarrow SA_t \cdot CA_t$
  `Build` foreground gate and instance masks (Eqs. 5, 6, 7)
  `Compute` $\mathcal{L}_{\text{ins}}, \mathcal{L}_{\text{sem}}$ (Eqs. 9, 12)
  $\mathcal{L}_{\text{ISAC}}(X_t, t) \leftarrow \lambda_{\text{ins}}(t)\mathcal{L}_{\text{ins}}(X_t) + \lambda_{\text{sem}}(t)\mathcal{L}_{\text{sem}}(X_t)$
  $\tilde{X}_t \leftarrow X_t - \eta \cdot \nabla_{X_t}\mathcal{L}_{\text{ISAC}}(X_t, t)$
  $X_{t-1} \leftarrow \text{Denoise}(\tilde{X}_t, \mathcal{T}, \epsilon_\theta, t)$
$I_0 \leftarrow \mathcal{D}(X_0)$ `// Decode to pixel space`

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate ISAC on three benchmarks that cover complementary aspects of multi-instance generation: T2I-CompBench [34], HRS-Bench [4], and our IntraCompBench. T2I-CompBench [34] and HRS-Bench [4] are standard for compositional text-to-image evaluation but do not isolate the intra-category regime where *count failures* and *semantic mixing* are most severe, so we introduce IntraCompBench to explicitly target two failure modes: multi-instance accuracy (prompts such as "three dogs"), which asks the model to generate a specified count N of a single class and measures how often the predicted instance count matches N, and multi-class accuracy (prompts such as "a dog, a cat, and a horse"), which asks for one instance of each of k different classes ($N = k$) and measures how often all k classes appear as distinct instances. For both settings, we evaluate generated images using open-vocabulary detection models [53, 73] and a prompt-specific matching procedure between detections and requested classes (see Appendix A for details of our IntraCompBench).

Implementation details. We strictly follow the official inference configurations for all diffusion models [8, 23, 61, 66, 77]. We apply the ISAC objective (Eq. (13)) in latent optimization and latent selection, two common training-free approaches for multi-instance generation. Both variants share identical schedules $\lambda_{\text{ins}}(t)$ and $\lambda_{\text{sem}}(t)$, which are fixed by design as in Eq. (13) and are not tuned per model or benchmark. ISAC with latent optimization (ISAC_{LO}) requires a single tuned hyperparameter, $\eta = 0.01$, shared across all models and benchmarks. Sensitivity analysis for η is provided in Appendix A. For all experiments involving ISAC with latent selection (ISAC_{LS}), we choose the best-out-of-10. For each text prompt, we automatically extract class tag–count pairs (τ_i, n_i) and simple token relations (P_{repel} and P_{bind}) using the open-sourced GPT-OSS [57] to ensure reproducibility. More details (*e.g.*, GPT instructions) are described in Appendix A.

4.2 Comparison with State of the Arts

Table 2 summarizes quantitative results for two popular diffusion models [23, 66]. With similar inference cost compared to attention control methods [13, 28, 62, 64], ISAC_{LO} clearly outperforms them on every metric of HRS-Bench, T2I-CompBench, and the multi-class setting of IntraCompBench. Text embedding approaches [12,14] show lower computational overhead, yet their performance falls behind ISAC due to spatially unaware optimization. The largest improvements appear in the intra-category regime: on SD1.5 [66], the average multi-class accuracy on IntraCompBench increases from 20% for the strongest baseline [12,28] to 36% with ISAC, and the HRS-Bench spatial score is roughly doubled (0.135 to 0.263). On the stronger SD3.5-M backbone, ISAC still provides a clear margin with especially large gains in the more crowded #4 and #5 cases. These trends

Table 2: Quantitative results on HRS-Bench [4], T2I-CompBench [34], and IntraCompBench. "Class" denotes the multi-class subset of IntraCompBench. Efficiency metrics are averaged across these multi-class task runs.

Method	HRSBench(↑)			T2I-CompBench(↑)			IntraCompBench(Class)(↑)					Efficiency(↓)	
	Color	Spatial	Size	Color	Texture	Complex	#2	#3	#4	#5	Avg.	Latency	VRAM
SD1.5 [66]	0.136	0.094	0.091	0.356	0.406	0.306	28%	2%	1%	0%	8%	**8s**	**4.9 GB**
+ A&E SIGGRAPH'23 [13]	0.149	0.104	0.101	0.392	0.447	0.290	48%	10%	5%	2%	16%	17s	9.2 GB
+ SynGen NeurIPS'23 [64]	0.159	0.111	0.107	0.420	0.479	0.311	50%	9%	4%	2%	16%	19s	9.3 GB
+ InitNO CVPR'24 [28]	0.175	0.120	0.116	0.456	0.520	0.338	55%	12%	7%	5%	20%	20s	9.6 GB
+ TEBOpt NeurIPS'24 [14]	0.181	0.127	0.123	0.461	0.544	0.353	52%	11%	8%	3%	18%	10s	6.4 GB
+ Self-Cross CVPR'25 [62]	0.170	0.118	0.114	0.445	0.508	0.324	48%	8%	4%	2%	15%	21s	10 GB
+ DOS AAAI'26 [12]	0.191	0.135	0.121	0.468	0.531	0.341	56%	14%	7%	4%	20%	11s	6.5GB
+ **ISAC$_{LO}$ (Ours)**	**0.318**	**0.263**	**0.252**	**0.683**	**0.631**	**0.354**	**65%**	**31%**	**29%**	**18%**	**36%**	21s	9.7 GB
SD3.5-M [23]	0.425	0.264	0.209	0.796	0.726	0.377	62%	23%	12%	3%	25%	**40s**	**22.9 GB**
+ A&E SIGGRAPH'23 [13]	0.427	0.263	0.215	0.798	0.726	0.378	65%	29%	16%	5%	28%	124s	73.8 GB
+ SynGen NeurIPS'23 [64]	0.425	0.260	0.211	0.801	0.718	0.365	66%	28%	15%	6%	28%	131s	74.3 GB
+ InitNO CVPR'24 [28]	0.443	0.275	0.228	0.810	0.728	0.378	77%	31%	17%	7%	33%	138s	74.6 GB
+ TEBOpt NeurIPS'24 [14]	0.438	0.279	0.220	0.805	0.730	0.381	78%	31%	19%	8%	34%	42s	28.8 GB
+ Self-Cross CVPR'25 [62]	0.431	0.268	0.219	0.795	0.720	0.371	78%	38%	19%	3%	34%	147s	76.4 GB
+ DOS AAAI'26 [12]	0.440	0.273	0.231	0.808	0.729	0.380	79%	33%	18%	7%	34%	44s	29.1GB
+ **ISAC$_{LO}$ (Ours)**	**0.473**	**0.350**	**0.258**	**0.838**	**0.739**	**0.388**	**98%**	**51%**	**40%**	**20%**	**52%**	140s	74.8 GB

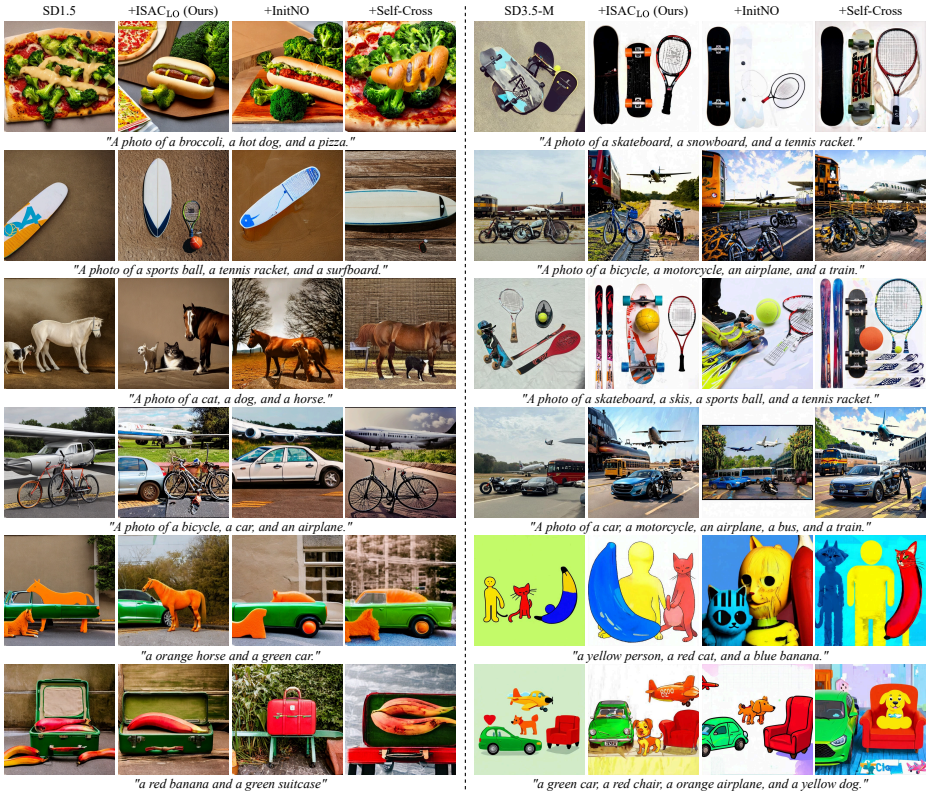


Fig. 5: Qualitative comparison of attention control methods on SD1.5 [66] and SD3.5-M [23].

Table 3: Comparison of ISAC_{LO} and count-supervised methods [7, 40] on IntraComp-Bench.

Method	IntraCompBench(Instance)(↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	Latency	VRAM
SD1.4 [66]	94%	74%	28%	22%	55%	**8s**	**4.9GB**
+ CG WACV'25 [40]	79%	67%	32%	19%	49%	14s	17.5GB
+ **ISAC_{LO} (Ours)**	**100%**	**90%**	**51%**	**40%**	**70%**	21s	9.7GB
SDXL [61]	90%	71%	49%	32%	61%	**48s**	**12.8GB**
+ CountGen CVPR'25 [7]	**97%**	83%	52%	44%	69%	100s	55.3GB
+ **ISAC_{LO} (Ours)**	96%	**89%**	**71%**	**47%**	**76%**	101s	29.8GB

Fig. 6: Qualitative results of ISAC_{LO} for exact-instance-count prompts on SD1.5 [66] and SD3.5-M [23].

Table 4: Application of ISAC_{LO} to layout-to-image generation.

Method	HRSBench (↑)			
	Counting (F1)	Color (Acc.)	Spatial (Acc.)	Size (Acc.)
GLIGEN [48]	0.666	0.307	0.268	0.188
+ CAR&SAR [60]	0.675	0.402	0.277	0.263
+ **ISAC_{LO} (Ours)**	**0.713**	**0.452**	**0.281**	**0.275**

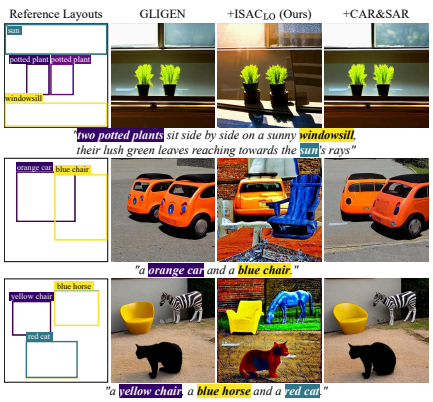


Fig. 7: Qualitative results of ISAC_{LO} with the GLIGEN [48] layout-to-image controller.

indicate that our instance-first attention control is most beneficial exactly where multiple similar objects must be separated and counted.

As shown in Fig. 5, previous state-of-the-art methods [28, 62] frequently blur boundaries between objects or merge categories into hybrid shapes when several related objects appear in a scene. By contrast, ISAC allocates distinct, spatially coherent instances to each requested class while maintaining their attributes. This behavior is consistent across prompts (from simple color–shape compositions to scenes with multiple objects) and across two diffusion models [23, 66]. Additional quantitative and qualitative results are provided in Appendices F and G.

4.3 Discussion

ISAC vs. prior count-supervised methods. As shown in Tab. 3 and Fig. 6, ISAC_{LO} achieves 70% and 76% instance-counting accuracy on SD1.4 [66] and SDXL [61], surpassing count-supervised methods (49% for Counting Guidance [40] and 69% for CountGen [7]) even though all methods are given the same instance-count supervision and ISAC_{LO} uses no fine-tuning or extra training data. Counting-based supervision for auxiliary models can only be exploited once

semantic evidence is sufficiently clear, so its effect is limited to later diffusion steps and to samples where object categories are already well formed. By contrast, ISAC_{LO} strengthens instance structure at early timesteps, before semantics have fully emerged, by reshaping attention to separate and stabilize instance-level regions; this enables it to reach the highest counting accuracy without auxiliary networks [32] or extra labels such as instance-level mask annotations.

Task flexibility of ISAC. Beyond the text-to-image setting in Tabs. 2 and 3, Tab. 4 shows that ISAC_{LO} also improves layout-to-image controllers [48]. On HRS-Bench, ISAC_{LO} yields the largest gains compared with layout refinements [60]. This advantage comes from enforcing instance separation for adjacent boxes early in the diffusion trajectory, rather than only constraining attention within each box. By carving out dense instance masks from initial coarse box layouts, ISAC_{LO} prevents neighboring objects from being merged and delivers more reliable counting in crowded layouts as shown in Fig. 7. Figure 25 in Appendix G provides more results on layout tasks.

Scalability via latent selection. In terms of computational cost, ISAC_{LO} lies in the same latency increase as prior latent-optimization attention-control methods, while incurring the expected additional VRAM usage from backpropagation (Tab. 2). For larger backbones, such as Flux [8, 9] and Qwen-Image [77], ISAC_{LS} (Algorithm 2) provides a gradient-free alternative that avoids backpropagation and uses the ISAC objective only as a verifier. Table 5 shows the effectiveness of ISAC_{LS} on multi-object generation. Figure 8 further shows that the ISAC_{LS} score distinguishes samples with missing instances or semantic mixing from better candidates.

Algorithm 2: ISAC with Latent Selection (ISAC_{LS})

Input: Prompt $\mathcal{T}$, Model ϵ_θ, decoder $\mathcal{D}$, Batch size B
Output: Image I_0

$X_T^{(i)} \sim \mathcal{N}(0, I)$, $S[i] = 0$, $\forall i = 1, \ldots, B$
for $i = 1, \ldots, B$ **do**
  for $t = T, T-1, \ldots, 1$ **do**
    $X_{t-1}^{(i)} \leftarrow \text{Denoise}(X_t^{(i)}, \mathcal{T}, \epsilon_\theta, t)$
      `with Hooks` $\rightarrow SA_t^{(i)}, CA_t^{(i)}$
    $CA_t^{(i),\text{ins}} \leftarrow SA_t^{(i)} \cdot CA_t^{(i)}$
    `Build` foreground gate and instance masks (Eqs. 5, 6, 7)
    `Compute` $\mathcal{L}_{\text{ins}}, \mathcal{L}_{\text{sem}}$ (Eqs. 9, 12)
    $\mathcal{L}_{\text{ISAC}}(X_t^{(i)}, t) \leftarrow \lambda_{\text{ins}}(t)\mathcal{L}_{\text{ins}}(X_t^{(i)}) + \lambda_{\text{sem}}(t)\mathcal{L}_{\text{sem}}(X_t^{(i)})$
    `Score Update:` $S[i] \leftarrow S[i] + \mathcal{L}_{\text{ISAC}}(X_t^{(i)}, t)$
$i^\star = \arg\min_i S[i]$ `// Best scored latent`
$I_0 \leftarrow \mathcal{D}(X_0^{(i^\star)})$ `// Decode to pixel space`

Table 5: Best-of-10 latent selection on IntraCompBench. The strategy is applied with ISAC using the base models [8, 9, 77].

Method	Multi-Class Accuracy (↑)					Multi-Instance Accuracy (↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	#2	#3	#4	#5	Avg.	Latency	VRAM
Flux.1-dev [8]	84%	37%	3%	2%	31%	97%	89%	82%	66%	83%	**50s**	**37.2GB**
+ **ISAC$_{LS}$ (Ours)**	**97%**	**48%**	**38%**	**19%**	**51%**	**99%**	**94%**	**85%**	**72%**	**88%**	85s	40.8GB
Qwen-Image [77]	91%	45%	33%	10%	48%	98%	92%	84%	70%	86%	**140s**	**60.1GB**
+ **ISAC$_{LS}$ (Ours)**	**99%**	**58%**	**42%**	**25%**	**56%**	**99%**	**96%**	**89%**	**78%**	**91%**	210s	65.3GB
Flux.2-dev [9]	97%	95%	84%	78%	88%	**100%**	93%	81%	75%	87%	**205s**	**74.2GB**
+ **ISAC$_{LS}$ (Ours)**	**99%**	**98%**	**89%**	**83%**	**92%**	**100%**	**98%**	**88%**	**81%**	**92%**	305s	79.8GB

(a) Latent Selection on Flux.1-dev (b) Latent Selection on Qwen-Image

Fig. 8: Qualitative results of latent selection with ISAC$_{LS}$ scoring.

Table 6: Application of ISAC$_{LO}$ on few-step text-to-image diffusion models.

Method	Multi-Class Accuracy (↑)					Multi-Instance Accuracy (↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	#2	#3	#4	#5	Avg.	Latency	VRAM
Z-Image-Turbo [70]	67%	58%	38%	32%	48%	**100%**	92%	45%	37%	68%	**6s**	**33GB**
+ **ISAC$_{LO}$ (Ours)**	**87%**	**74%**	**61%**	**49%**	**68%**	**100%**	**97%**	**62%**	**49%**	**77%**	13s	78GB
Flux.2-klein-4B [10]	79%	45%	12%	1%	34%	98%	81%	75%	49%	75%	**4s**	**29GB**
+ **ISAC$_{LO}$ (Ours)**	**89%**	**63%**	**43%**	**19%**	**54%**	**100%**	**93%**	**86%**	**65%**	**86%**	8s	64GB

Applicability to few-step models. A key concern is whether ISAC$_{LO}$ remains effective under the limited inference budgets of few-step models. We validate our method on Z-Image-Turbo [70] and Flux.2-klein-4B [10], which utilize only 8 and 4 steps, respectively. The results in Tab. 6 demonstrate that ISAC$_{LO}$ effectively boosts multi-object synthesis even with fewer opportunities for diffusion guidance. This confirms ISAC's suitability for low-latency applications.

Importance of instance-to-semantic schedule. Table 7 shows that both losses and their ordering are critical. Optimizing only the instance term (A) yields strong multi-instance accuracy but almost no gain on multi-class prompts, indicating that it can form the right number of structures but cannot reliably assign semantics. Using only the semantic term or a fixed balance (B–C) is also suboptimal, since semantic separation without prior boundary stabilization is unstable. The reverse schedule that goes from semantic to instance (D) further

Table 7: Effect of the loss schedule on $\mathbf{ISAC_{LO}}$. "Class" and "Instance" denote multi-class and multi-instance accuracy on IntraCompBench, respectively.

	Description	$\lambda_{\text{ins}}(t)$	$\lambda_{\text{sem}}(t)$	Class	Instance
A	Instance Only	1	0	10%	65%
B	Semantic Only	0	1	28%	54%
C	Fixed Balance	0.5	0.5	25%	60%
D	Semantic-to-Instance	$1 - t/T$	t/T	21%	55%
E	Instance-to-Semantic	t/T	$1 - t/T$	**36%**	**69%**

degrades multi-class accuracy. Our instance-to-semantic schedule (E) achieves the best performance on both metrics, supporting the hypothesis that instance structure should be established first and then refined semantically.

5 Conclusion

In this work, we study multi-instance generation in diffusion models through the interaction between instance structure and token-conditioned semantics. We show that count failures and semantic mixing arise when semantic binding proceeds before instance boundaries are reliably stabilized. Building on this diagnosis, we introduce ISAC, a training-free, model-agnostic objective that first stabilizes instance regions from structural cues and then binds classes and attributes within each region. Across three benchmarks and multiple diffusion backbones, ISAC improves both text-to-image and layout-to-image generation. On IntraCompBench, it achieves higher counting and compositional accuracy than prior count-supervised approaches without additional data, fine-tuning, or external vision models. Overall, instance-first control provides a practical, reproducible path toward narrowing the multi-instance reliability gap of open-weight diffusion models. It further suggests a promising direction for decoupled structure–semantics control in video and other structured generative settings.

Acknowledgments

This work was partly supported by the KHIDI grant funded by the Korean government (MOHW) [No.RS-2025-02307233], the NRF or IITP grants funded by the Korean government (MSIT) [No.RS-2026-25472075, No.RS-2025-02305581, No.RS-2025-25442338 (AI Star Fellowship-SNU), and No.RS-2021II211343 (SNU AI)], the ITIP grant funded by the Korean government (MOTIR) [No.RS-2026-25549946], the Advanced GPU Utilization and AI Computing Infrastructure Enhancement User Support Programs funded by the Korean government (MSIT) [No.05-26-04-0094], the Research grant from SNU, the Strategic Hub grant for International Research Collaboration of SNU, and the AI Seoul Tech Research Support Program of the Seoul Future Foundation.

Kyungsu Kim is affiliated with the School of Transdisciplinary Innovations, Department of Biomedical Science, Interdisciplinary Program in Artificial Intelligence (IPAI), Medical Research Center, and AI Institute at SNU.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ahn, D., Cho, H., Min, J., Jang, W., Kim, J., Kim, S., Park, H.H., Jin, K.H., Kim, S.: Self-rectifying diffusion sampling with perturbed-attention guidance. In: ECCV. pp. 1–17 (2024)
3. Arkhipkin, V., Korviakov, V., Gerasimenko, N., Parkhomenko, D., Vasilev, V., Letunovskiy, A., Vaulin, N., Kovaleva, M., Kirillov, I., Novitskiy, L., Koposov, D., Kiselev, N., Varlamov, A., Mikhailov, D., Polovnikov, V., Shutkin, A., Agafonova, J., Vasiliev, I., Kargapoltseva, A., Dmitrienko, A., Maltseva, A., Averchenkova, A., Kim, O., Nikulina, T., Dimitrov, D.: Kandinsky 5.0: A family of foundation models for image and video generation. arXiv preprint arXiv:2511.14993 (2025)
4. Bakr, E.M., Sun, P., Shen, X., Khan, F.F., Li, L.E., Elhoseiny, M.: HRS-Bench: Holistic, reliable and scalable benchmark for text-to-image models. In: ICCV. pp. 20041–20053 (2023)
5. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, R., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: ICLR. pp. 51304–51323 (2024)
6. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: MultiDiffusion: Fusing diffusion paths for controlled image generation. In: ICML. pp. 1737–1752 (2023)
7. Binyamin, L., Tewel, Y., Segev, H., Hirsch, E., Rassin, R., Chechik, G.: Make It Count: Text-to-image generation with an accurate number of objects. In: CVPR. pp. 13242–13251 (2025)
8. Black Forest Labs: FLUX. `https://github.com/black-forest-labs/flux` (2024), accessed 2026-03-05
9. Black Forest Labs: FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2` (2025), accessed 2026-03-05
10. Black Forest Labs: FLUX.2-klein-4B. `https://huggingface.co/black-forest-labs/FLUX.2-klein-4B` (2026), accessed 2026-03-05
11. Bocheng, YuhangMa, wuliebucha, Liu, S., Ma, A., Wu, X., Leng, D., Yin, Y.: HiCo: Hierarchical controllable diffusion model for layout-to-image generation. In: NeurIPS. pp. 128886–128910 (2024)
12. Byun, D., Park, J., Ko, J., Choi, C., Rhee, W.: DOS: Directional object separation in text embeddings for multi-object image generation. In: AAAI. pp. 2490–2497 (2026)
13. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-Excite: Attention-based semantic guidance for text-to-image diffusion models. ACM transactions on Graphics (TOG) **42**(4), 1–10 (2023)
14. Chen, C.Y., Tseng, C., Tsao, L.W., Shuai, H.H.: A cat is a cat (not a dog!): Unraveling information mix-ups in text-to-image encoders through causal analysis and embedding optimization. In: NeurIPS. pp. 57944–57969 (2024)
15. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: PixArt-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In: ECCV. pp. 74–91 (2024)
16. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In: ICLR. pp. 57611–57640 (2024)

17. Chen, M., Laina, I., Vedaldi, A.: Training-free layout control with cross-attention guidance. In: WACV. pp. 5343–5353 (2024)
18. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-Pro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811 (2025)
19. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE TPAMI **24**(5), 603–619 (2002)
20. Dahary, O., Cohen, Y., Patashnik, O., Aberman, K., Cohen-Or, D.: Be decisive: Noise-induced layouts for multi-subject generation. In: ACM SIGGRAPH. pp. 1–12 (2025)
21. Dahary, O., Patashnik, O., Aberman, K., Cohen-Or, D.: Be yourself: Bounded attention for multi-subject text-to-image generation. In: ECCV. pp. 432–448 (2024)
22. Dutta, D., Chen, J., RAJAGOPALAN, R., Wei, Y.L., Choudhury, R.R.: Steer away from mode collisions: Improving composition in diffusion models. In: ICLR (2026)
23. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML. pp. 12606–12633 (2024)
24. Feng, W., He, X., Fu, T.J., Jampani, V., Akula, A.R., Narayana, P., Basu, S., Wang, X.E., Wang, W.Y.: Training-free structured diffusion guidance for compositional text-to-image synthesis. In: ICLR (2023)
25. Gemini Team: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. `https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf` (2025), accessed 2026-03-05
26. Google AI for Developers: Gemini api documentation: Nano banana image generation. `https://ai.google.dev/gemini-api/docs/image-generation#thinking-process`, accessed 2026-03-05
27. Google DeepMind: Nano banana 2: Combining pro capabilities with lightning-fast speed (2026), `https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/`, accessed 2026-03-05
28. Guo, X., Liu, J., Cui, M., Li, J., Yang, H., Huang, D.: InitNO: Boosting text-to-image diffusion models via initial noise optimization. In: CVPR. pp. 9380–9389 (2024)
29. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross-attention control. In: ICLR (2023)
30. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS. p. 6629–6640 (2017)
31. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop on Deep Generative Models and Downstream Applications (2021)
32. Hobley, M., Prisacariu, V.: Learning to count anything: Reference-less class-agnostic counting with weak supervision. In: CVPR Workshop on Transformers for Vision (2023)
33. Hu, T., Li, L., van de Weijer, J., Gao, H., Shahbaz Khan, F., Yang, J., Cheng, M.M., Wang, K., Wang, Y.: Token merging for training-free semantic binding in text-to-image synthesis. In: NeurIPS. pp. 137646–137672 (2024)
34. Huang, K., Duan, C., Sun, K., Xie, E., Li, Z., Liu, X.: T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. IEEE TPAMI **47**(5), 3563–3579 (2025)

35. Huang, S., Huang, S., Luo, P., Zhang, H.: Laytrol: Preserving pretrained knowledge in layout control for multimodal diffusion transformers. In: AAAI. pp. 5113–5121 (2026)
36. Hwang, J.J., Yu, S.X., Shi, J., Collins, M.D., Yang, T.J., Zhang, X., Chen, L.C.: SegSort: Segmentation by discriminative sorting of segments. In: ICCV. pp. 7334–7344 (2019)
37. Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., Kumar, S.: Rethinking FID: Towards a better evaluation metric for image generation. In: CVPR. pp. 9307–9315 (2024)
38. Jiang, D., Song, G., Wu, X., Zhang, R., Shen, D., Zong, Z., Liu, Y., Li, H.: CoMat: Aligning text-to-image diffusion model with image-to-text concept matching. In: NeurIPS. pp. 76177–76209 (2024)
39. Jo, S., Lee, Z., Lee, W., Choi, J., Park, J., Kim, K.: TRACE: Your diffusion model is secretly an instance edge detector. In: ICLR (2026)
40. Kang, W., Galim, K., Koo, H.I., Cho, N.I.: Counting guidance for high fidelity text-to-image synthesis. In: WACV. pp. 899–908 (2025)
41. Kim, D., Lee, J., Park, J.: Improving editability in image generation with layer-wise memory. In: CVPR. pp. 7889–7898 (2025)
42. Kim, K., Ye, J.C.: Noise2Score: tweedie's approach to self-supervised image denoising without clean images. In: NeurIPS. pp. 864–874 (2021)
43. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In: NeurIPS. pp. 36652–36663 (2023)
44. Lee, H., Lee, H., Gye, S., Kim, J.: Beta sampling is all you need: Efficient image generation strategy for diffusion models using stepwise spectral analysis. In: WACV. pp. 4215–4224 (2025)
45. Lee, Y., Yoon, T., Sung, M.: GrounDiT: Grounding diffusion transformers via noisy patch transplantation. In: NeurIPS. pp. 58610–58636 (2024)
46. Li, B., Wang, C.Y., Xu, H., Zhang, X., Armand, E.J., Srivastava, D., Shan, X., Chen, Z., Xie, J., Tu, Z.: OverLayBench: A benchmark for layout-to-image generation with dense overlaps. In: NeurIPS Datasets and Benchmarks Track (2025)
47. Li, S., Le, H., Xu, J., Salzmann, M.: Enhancing compositional text-to-image generation with reliable random seeds. In: ICLR. pp. 45444–45465 (2025)
48. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: GLIGEN: Open-set grounded text-to-image generation. In: CVPR. pp. 22511–22521 (2023)
49. Lian, L., Li, B., Yala, A., Darrell, T.: LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. TMLR (2024)
50. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014)
51. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
52. Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the CoordConv solution. In: NeurIPS. p. 9628–9639 (2018)
53. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. In: ECCV. pp. 38–55 (2024)
54. Ma, Y., Wu, X., Sun, K., Li, H.: HPSv3: Towards wide-spectrum human preference score. In: ICCV. pp. 15086–15095 (2025)

55. Meral, T.H.S., Simsar, E., Tombari, F., Yanardag, P.: CONFORM: Contrast is all you need for high-fidelity text-to-image diffusion models. In: CVPR. pp. 9005–9014 (2024)
56. OpenAI: GPT-Image-1.5 (2025), `https://platform.openai.com/docs/models/gpt-image-1.5`, accessed 2026-03-05
57. OpenAI: gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925 (2025)
58. OpenAI: GPT-5.5 (2026), `https://platform.openai.com/docs/models/gpt-5.5`, accessed 2026-03-05
59. Park, D., Kim, S., Moon, T., Kim, M., Lee, K., Cho, J.: Rare-to-Frequent: Unlocking compositional generation power of diffusion models on rare concepts with LLM guidance. In: ICLR. pp. 14650–14676 (2025)
60. Phung, Q., Ge, S., Huang, J.B.: Grounded text-to-image synthesis with attention refocusing. In: CVPR. pp. 7932–7942 (2024)
61. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: ICLR. pp. 1862–1874 (2024)
62. Qiu, W., Wang, J., Tang, M.: Self-cross diffusion guidance for text-to-image synthesis of similar subjects. In: CVPR. pp. 23528–23538 (2025)
63. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021)
64. Rassin, R., Hirsch, E., Glickman, D., Ravfogel, S., Goldberg, Y., Chechik, G.: Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. In: NeurIPS. pp. 3536–3559 (2023)
65. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded SAM: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)
66. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
67. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
68. Shen, D., Song, G., Xue, Z., Wang, F.Y., Liu, Y.: Rethinking the spatial inconsistency in classifier-free diffusion guidance. In: CVPR. pp. 9370–9379 (2024)
69. Shirakawa, T., Uchida, S.: NoiseCollage: A layout-aware text-to-image diffusion model based on noise cropping and merging. In: CVPR. pp. 8921–8930 (2024)
70. Team, I., Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., Li, Z., Li, Z.Y., Liu, D., Liu, D., Shi, J., Wu, Q., Yu, F., Zhang, C., Zhang, S., Zhou, S.: Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699 (2025)
71. Tian, Y., Ye, Q., Doermann, D.: YOLOv12: Attention-centric real-time object detectors. In: NeurIPS. pp. 78433–78457 (2025)
72. Ventura, M., Toker, M., Patashnik, O., Belinkov, Y., Reichart, R.: DeLeaker: Dynamic inference-time reweighting for semantic leakage mitigation in text-to-image models. In: ICLR (2026)
73. Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., Ding, G.: YOLOE: Real-time seeing anything. In: ICCV. pp. 24591–24602 (2025)
74. Wang, X., Darrell, T., Rambhatla, S.S., Girdhar, R., Misra, I.: InstanceDiffusion: Instance-level control for image generation. In: CVPR. pp. 6232–6242 (2024)
75. Wang, Z., Sha, Z., Ding, Z., Wang, Y., Tu, Z.: TokenCompose: Text-to-image diffusion with token-level supervision. In: CVPR. pp. 8553–8564 (2024)

76. Wang, Z., Peng, D., Chen, F., Yang, Y., Lei, Y.: Training-free dense-aligned diffusion guidance for modular conditional image synthesis. In: CVPR. pp. 13135–13145 (2025)
77. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-Image technical report. arXiv preprint 2508.02324 (2025)
78. Xiao, J., Lv, H., Li, L., Wang, S., Huang, Q.: R&B: Region and boundary aware zero-shot grounded text-to-image generation. In: ICLR. pp. 26088–26116 (2024)
79. Xie, J., Li, Y., Huang, Y., Liu, H., Zhang, W., Zheng, Y., Shou, M.Z.: BoxDiff: Text-to-image synthesis with training-free box-constrained diffusion. In: ICCV. pp. 7452–7461 (2023)
80. Yang, L., Yu, Z., Meng, C., Xu, M., Ermon, S., Cui, B.: Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In: ICML. pp. 56704–56721 (2024)
81. Zhang, H., Hong, D., Wang, Y., Shao, J., Wu, X., Wu, Z., Jiang, Y.G.: CreatiLayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In: ICCV. pp. 18487–18497 (2025)
82. Zhang, X., Yang, L., Cai, Y., Yu, Z., Wang, K.N., Tian, Y., Xu, M., Tang, Y., Yang, Y., Cui, B., et al.: RealCompo: Balancing realism and compositionality improves text-to-image diffusion models. In: NeurIPS. pp. 96963–96992 (2024)
83. Zhang, X., Yang, L., Li, G., Cai, Y., xie jiake, Tang, Y., Yang, Y., Wang, M., CUI, B.: IterComp: Iterative composition-aware feedback learning from model gallery for text-to-image generation. In: ICLR. pp. 31968–31988 (2025)
84. dewei Zhou, Xie, J., Yang, Z., Yang, Y.: 3DIS: Depth-driven decoupled image synthesis for universal multi-instance generation. In: ICLR (2025)
85. Zhou, D., Li, Y., Ma, F., Zhang, X., Yang, Y.: MIGC: Multi-instance generation controller for text-to-image synthesis. In: CVPR. pp. 6818–6828 (2024)

A Implementation Details

A.1 Method Details

Attention accumulation. We accumulate self- and cross-attention maps from all attention layers. When self-attention (SA) and cross-attention (CA) maps have different spatial resolutions across layers, in case of U-Net [67] architectures, we bilinearly upsample each map to the highest spatial resolution ($H \times W$) and average over layers and heads to obtain a single SA/CA pair *per step*:

$$SA_t = \frac{1}{\mathcal{N}} \sum_{l=1}^{M} \sum_{h=1}^{h_l} \texttt{Upsample}(SA_l^h(X_t), \delta_l) \tag{14}$$

$$CA_t = \frac{1}{\mathcal{N}} \sum_{l=1}^{M} \sum_{h=1}^{h_l} \texttt{Upsample}(CA_l^h(X_t, \mathcal{T}), \delta_l) \tag{15}$$

where $\delta_l = H/H_l = W/W_l$ is the upsampling factor for layer l, and $\mathcal{N} = \sum_{l=1}^{M} h_l$ is the total number of heads. Then, we apply `min-max` normalization to the accumulated maps as defined in Eq. (17). For a fair comparison, we use the same attention accumulation scheme across all baseline methods [8, 15, 16, 23, 61, 66, 77].

Choice of the step size η. We evaluate the sensitivity of ISAC_{LO} to the gradient step size η in Algorithm 1 under our instance-to-semantic schedule. When η is too small, the optimization loss barely decreases and has negligible impact on the generated images. In contrast, excessively large values of η cause latent collapse and noticeably degrade image quality. This trade-off is consistent with other latent-optimization methods [13], and ISAC_{LO} is subject to the same limitation.

Empirically, we find that a step size of $\eta = 10^{-2}$ achieves stable improvement without visible artifacts across backbones and datasets (Fig. 9). The gradient-descent update $\tilde{X}_t \leftarrow X_t - \eta \nabla_{X_t} \mathcal{L}_t(X_t)$ in Algorithm 1 is applied once per

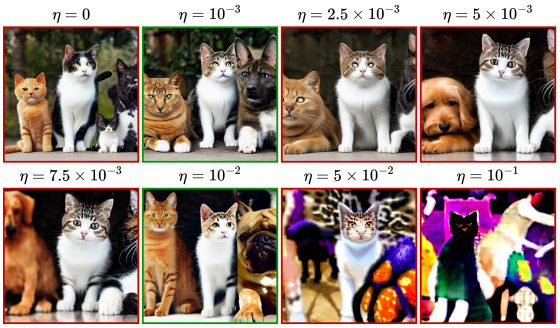


Fig. 9: Qualitative comparison of ISAC_{LO} with various step sizes η. For all cases, we provide *"A photo of two cats and a dog"* as the input prompt and use SD1.5 [66] as the baseline diffusion model.

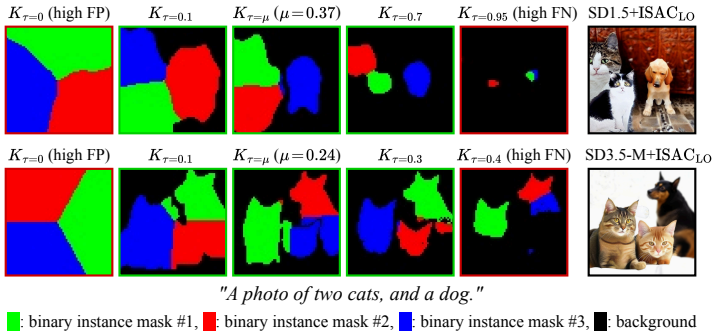


Fig. 10: Foreground-gate threshold analysis. Binary instance masks K are extracted at $t = T/2$ as in Fig. 4 under different thresholds τ on SD1.5 (top) and SD3.5-M (bottom). The rightmost images are generated by ISAC$_{\text{LO}}$ with mean thresholding ($\tau = \mu$).

diffusion timestep, and η is the only additional hyperparameter introduced by ISAC$_{\text{LO}}$; all other settings follow the default configuration of each backbone.

Foreground-gate. We restrict self-attention clustering to foreground regions to reduce potential clustering errors caused by background pixels. We use an adaptive foreground threshold, defined as the mean value of CA_t^{ins} (Eq. (5)), rather than a fixed threshold. Figure 10 supports the design choice. Low threshold τ admits background (FP), while high τ drops foreground (FN). Our adaptive mean threshold ($\tau = \mu$) scales with per-timestep attention magnitude, avoiding both. Still, small or thin objects remain challenging.

Details of clustering with coordinates. We visualize how and why normalized coordinates $(x, y) \in [-1, 1]^2$ are concatenated to the feature vectors in Eq. (7). When adjacent instances are highly semantically similar, their foreground positions can have similar SA features, and weak boundary cues are easily overwhelmed (Fig. 11, top/red). By forming a joint space of SA features and spatial coordinates and performing clustering in this augmented space, the clustering better respects spatial connectivity within each instance, making boundaries more separable and making subtle boundary cues more salient (Fig. 11, bottom/green).

This spatial augmentation is also quantitatively important. Clustering with SA features alone yields 21% multi-class accuracy, whereas the default SA+coordinate clustering achieves 36%, indicating that coordinates act as a spatial regularizer for separating adjacent objects.

Choice of clustering algorithm. We compare alternative clustering algorithms in Fig. 12 and Tab. 8. While K-means, spectral clustering, and Gaussian mixture models (GMMs) achieve similar performance on both *Multi-Class* and

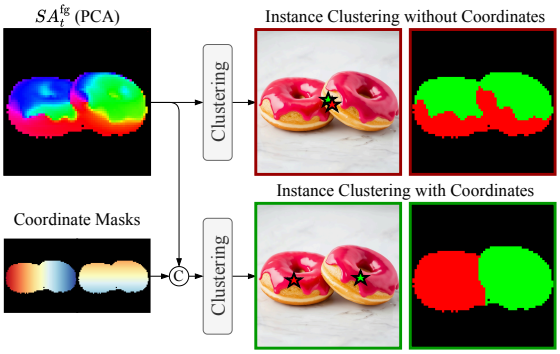


Fig. 11: Concatenating normalized coordinates $(x, y) \in [-1, 1]^2$ to SA features stabilizes clustering, reducing erroneous merges and making subtle boundary cues more salient. The image is generated with the prompt, *"a photo of two donuts"* on SD3.5-M [23]. We averaged all accumulated self/cross-attention maps over timesteps into a single (SA, CA) pair. Then, clustering algorithms are applied to those maps with or without concatenation of spatial coordinates. Note that this is commonly used in previous literature [19, 36, 52].

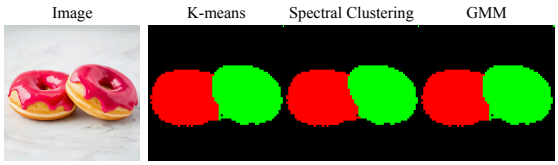


Fig. 12: Qualitative comparison of clustering algorithms. The image is generated with the prompt, *"a photo of two donuts"*. We averaged all accumulated self/cross-attention maps over timesteps into a single (SA, CA) pair. Then, clustering algorithms are applied to those maps with concatenation of spatial coordinates. This is an extension of Fig. 11.

Table 8: Comparison of clustering algorithms in terms of accuracy (%) and latency (ms). Latency is defined as the execution time for a one-time application of each clustering algorithm to a self-attention map. This table can be seen as an extension of Tab. 7.

Clustering Algorithm	Multi-Class (↑)	Multi-Instance (↑)	Latency (↓)
K-means	**36%**	**69%**	**505 ms**
Spectral Clustering	35%	**69%**	1,630 ms
GMM	**36%**	**69%**	8,313 ms

Multi-Instance accuracy, K-means, which we adopt as our default choice, is approximately 3× to 16× faster. Given this favorable speed–accuracy trade-off, we use K-means for ISAC.

(a) LLM instruction for prompt parsing (b) Prompt parsing examples

Fig. 13: Details on LLM-guided automatic prompt parsing.

Automatic prompt parsing. We use an LLM-based parser to automatically extract class tokens τ_i, counts n_i, attributes $\{\chi_{i,j}\}$, and token relations for P_{repel} and P_{bind} from natural-language prompts. For P_{repel} and P_{bind}, the parser resolves only object-local attribute pairs. If an attribute is global or cannot be uniquely assigned to a noun/class, we conservatively omit the bind pair. Fig. 13(a) shows the instruction used for this parsing procedure and Fig. 13(b) provides example outputs on prompts from T2I-CompBench and HRS-Bench. In all experiments, we use GPT-OSS-20B [57], which produces reliable and consistent parses in practice. For a moderately complex prompt such as *"The fluffy pillow and glass lampshade rest on the wooden nightstand by the metallic bed."*, a single forward pass of the parser takes about 10 seconds and requires roughly 40 GB of VRAM on a single A100 GPU.

While the LLM parser produces highly accurate outputs overall, our current rules do not resolve ambiguous count expressions such as "a few", "several", "a couple of", or plurals without explicit numerals (*e.g.*, "blades"). Handling these ambiguous counts is beyond the scope of our experiments, but it could be addressed by sampling counts from a small candidate set, for example $\mathcal{N} = \{3, 4\}$, or by providing targeted few-shot examples to the LLM.

Cross-attention normalization. Baseline methods, Attend-and-Excite [13], InitNO [28], and TEBOpt [14], use the `softmax` normalization technique on cross-attention maps. It operates by applying the `softmax` function to the cross-

attention maps along the token dimension, excluding the `SOT` token at index 0. This sharpens the attention, emphasizing foreground objects while suppressing background noise. The formulation is given by:

$$CA_t^{\text{softmax}} = \texttt{softmax}(\beta \cdot CA_t[1:]). \tag{16}$$

Following the official implementation, we set $\beta = 100$ for SD1.4, SD1.5, SD2.1 [66] and SD3.5-M [23] across baseline methods through our experiments (see Secs. 4.2 and F). In contrast, ISAC adopts a simple element-wise `min-max` normalization to rescale attention maps:

$$CA_t^{\text{minmax}} = \frac{CA_t - \min(CA_t)}{\max(CA_t) - \min(CA_t)}. \tag{17}$$

Although both `softmax` and `min-max` normalization are common, we opt for the `min-max` approach due to its practicality as a parameter-free method.

A.2 IntraCompBench Details

Standard benchmarks like T2I-CompBench [34], HRS-Bench [4] and MultiGen benchmark [75] do not isolate the *intra-category* setting where failure on instance discrimination becomes most severe. IntraCompBench is designed to stress this regime and to separately probe the two symptoms we target: (1) *count failures* and (2) *semantic mixing*.

Task 1 – multi-instance accuracy (%). This task isolates count failures. We sample a single class A from a super-category in Tab. 9 and specify an integer $n \in \{2, 3, 4, 5\}$. Now the prompt is formatted as "A photo of [n] [class A]s" (*e.g.*, *"A photo of five cats"*). Success requires producing exactly $N = n$ instances of A. This primarily evaluates instance formation.

Task 2 – multi-class accuracy (%). This task stresses semantic mixing. We sample $k \in \{2, 3, 4, 5\}$ distinct classes within the same super-category and format into "A photo of a [class A], a [class B], ..., and a [class E]" (*e.g.*, *"A photo of a dog, a cat, a horse, a cow, and a sheep"*). Success requires (i) forming $N{=}k$ instances and (ii) assigning the correct semantics to each layout (preventing cross-object leakage). The intra-category constraint makes this substantially harder than prior multi-class settings (*e.g.*, MultiGen [75]), where inter-category separability reduces confusion (see Fig. 2).

Class distribution. For reliable automatic evaluation we use a subset of countable object classes from the 80 COCO categories[3] [50], grouped into four

[3] Most detection models are trained on COCO classes, so we focus our evaluation on COCO, where the pretrained models are most reliable, rather than experimenting with new classes from datasets like ADE. In practice, MultiGen results in TokenCompose [75] have shown minimal differences in performance trends between COCO and ADE.

Table 9: Countable classes from COCO [50] dataset used in the evaluation.

Category	Classes
Animal	(9 classes) cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
Vehicle	(8 classes) bicycle, car, motorcycle, airplane, bus, train, truck, boat
Sports	(10 classes) skateboard, snowboard, skis, sports ball, baseball bat, baseball glove, tennis racket, surfboard, kite, frisbee
Food	(10 classes) banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake

Table 10: Possible combinations of classes for multi-class evaluation.

Category	#2	#3	#4	#5
Animal (9 classes)	36	84	126	126
Vehicle (8 classes)	28	56	70	56
Sports (10 classes)	45	120	210	252
Food (10 classes)	45	120	210	252
Total	154	380	616	686

super-categories: animals, vehicles, sports, and food (Tab. 9). We exclude classes such as *"person"*, very small objects (*e.g.*, *"fork"*), ambiguous items (*e.g.*, *"book"*), or objects whose duplication is ill-defined (*e.g.*, *"bench"*), due to difficulties in reliable detection or instance differentiation.

Evaluation via ensemble. To evaluate accuracy, we use an ensemble of three detectors: Grounding DINO [53], YOLOE [73], and YOLOv12 [71]. We adopt a 2-of-3 agreement rule, in other words, an instance is considered correctly detected only if captured by at least two detectors. Accuracy is the ratio of correctly detected instances to the target number of instances (k for multi-class, n for multi-instance). For example, if "cat" and "dog" are detected in the image from the text prompt "A photo of a cat, a dog, a horse, a cow and a sheep", then the accuracy is calculated as $2/5 = 40\%$ for multi-class accuracy. For multi-instance accuracy, if 3 instances of "cat" are detected in the image from the text prompt "A photo of five cats", then the accuracy is calculated as $3/5 = 60\%$.

Prompt sampling. For multi-class evaluation, we sample 20% of all combinations in Tab. 10 for each $k \in \{2, 3, 4, 5\}$ and generate 10 images per prompt. For example, in #5-class evaluation, we randomly sample 25 animal, 11 vehicle, 50 sports, 50 food combinations; 136 prompts total. For multi-instance evaluation, we generate 10 images for each class in the four super-categories for every $n \in \{2, 3, 4, 5\}$, yielding 37 unique prompts per n.

A.3 Evaluation Details

For the layout-to-image evaluation in Tab. 4 and Fig. 7, we use HRS-Bench [4] and follow the two-stage pipeline established by prior work [60]: (i) generate text-to-box layouts with an LLM, then (ii) synthesize images conditioned on those layouts. To ensure comparability, we directly reuse the LLM-produced box layouts released in Attention-Refocusing [60] and run only the layout-conditioned image generation step.

Although OverLayBench [46] is tailored for evaluating overlapping layouts, its publicly available annotations are not directly compatible with training-free layout-to-image guidance methods [17, 21, 45, 60, 78, 79]. OverLayBench provides a global prompt together with (local prompt, bounding box) pairs, where each local prompt specifies the semantics within its box. Training-based controllers [11,48,74,81,84,85] can ingest an *arbitrary* local prompt via dedicated layout adapters. In contrast, existing training-free layout-to-image guidance methods [17,21,45,60,78,79] require per-layout class tokens $\{\tau_i\}$ to be present in the *global* prompt. Because this condition is not guaranteed in the public OverLayBench data, it precludes a fair comparison with ISAC. We leave a comprehensive quantitative study on OverLayBench—after reconciling prompt formats—to future work.

B Broader Related Work Comparisons

S-CFG [68] highlights spatial inconsistencies in global Classifier-Free Guidance (CFG) [31] and proposes region-level guidance that leverages self- and cross-attention maps. However, S-CFG primarily *amplifies* semantic evidence within semantically segmented patches. It neither disentangles competing semantic signals nor establishes instance boundaries from structural cues.

Contemporary training-free methods also address compositional failures from different perspectives. CO3 [22] steers sampling away from mode collisions and improves multi-class composition, but it does not explicitly build spatial instance partitions. DeLeaker [72] dynamically reweights attention to mitigate semantic leakage, but, like other token-conditioned SA-guided methods [28, 62], its structural cues remain tied to CA token semantics. As a result, these methods can reduce semantic leakage across different tokens, but they are not designed to separate multiple instances that share the same semantic token. In contrast, ISAC uses CA only to obtain a class-agnostic foreground gate and derives instance partitions directly from SA affinities, decoupling instance discovery from token semantics. These differences among CO3 [22], DeLeaker [72], and ISAC are reflected in the IntraCompBench performance results presented in Tab. 11.

Table 11: Comparison with contemporary training-free baselines on IntraCompBench. Multi-Class and Multi-Instance denote average accuracy over the #2–#5 settings.

Method	IntraCompBench (↑)		Efficiency (↓)	
	Class	Instance	Latency	VRAM
SD1.5 [66]	8%	54%	8 s	4.9 GB
+ CO3† [22]	18%	55%	23 s	6.0 GB
+ ISAC_{LO}	**36%**	**69%**	21 s	9.7 GB
SD3.5-M [23]	25%	64%	40 s	22.9 GB
+ DeLeaker‡ [72]	30%	62%	60 s	32.0 GB
+ ISAC_{LO}	**52%**	**83%**	140 s	74.8 GB

†DDIM-only; not directly applicable to flow-matching backbones.
‡DiT-only; not directly applicable to U-Net backbones.

Beyond Counting Guidance [40], several works [5, 76] steer generation by querying pretrained vision models as external perceptual signals. Dense Geometry Alignment [76] aligns segmentation masks predicted from intermediate images with the target object layouts. Yet, because these vision models [53] are trained on clean, semantically rich images, their predictions on noisy diffusion states are unreliable, yielding weak guidance (see Fig. 3). Moreover, many approaches apply predictions on Tweedie-denoised intermediates [42]; Tweedie's correction is not applicable to flow-matching models [51] such as SD3.5-M [23], limiting the practicality of using external models in this regime.

Specifically, CountGen [7] is a two-stage approach that first establishes instance-mask layouts with a fine-tuned `ReLayout` module and then applies guidance conditioned on the resulting masks. We find its gains stem largely from the quality of the mask proposals; without explicit instance-separation guidance, it lags behind ISAC_{LO} on our intra-category settings. ISAC_{LO} also complements CountGen: replacing CountGen's guidance with ISAC_{LO} further improves accuracy, indicating that ISAC_{LO}'s instance discrimination and CountGen's mask proposals address different parts of the problem (see Tab. 12).

Table 12: Extended results for CountGen [7] and ISAC_{LO}.

Method	IntraCompBench (Instance) (↑)				
	#2	#3	#4	#5	Avg.
SDXL [61]	90%	71%	49%	32%	61%
+ CountGen CVPR'25 [7]	97%	83%	52%	44%	69%
+ **ISAC_{LO} (Ours)**	96%	89%	71%	47%	76%
+ CountGen + **ISAC_{LO} (Ours)**	**98%**	**91%**	**74%**	**50%**	**78%**

Among training-free layout guidance methods [6, 17, 21, 45, 49, 59, 69, 78, 79], Bounded Attention [21] is the only approach that explicitly separates self- and cross-attention across box-level masks to mitigate semantic mixing. Because it only partitions semantics by boxes, it cannot guarantee instance discrimination and often fails to count correctly in crowded scenes (see Fig. 14).

Moreover, its mutual-exclusivity assumption—each pixel belongs to at most one box —breaks under overlapping or tightly packed layouts. Thus, the resulting suppression of attention in shared regions degrades spatial control and semantic fidelity (Fig. 15). Since performance can also depend on the arbitrary ownership order under overlap, we exclude Bounded Attention from our layout comparisons.

Fine-tuned models. Several methods ensure object-level separation in attention maps by adapting the base model with external segmentation signals. TokenCompose [75] and CoMat [38] fine-tune the UNet so that cross-attention aligns with masks predicted by a pretrained model [65]. A key limitation is the reduced effective vocabulary compared with general-purpose diffusion backbones, which restricts applicability. ISAC is complementary to these methods and can be applied at inference time without retraining (Tab. 17).

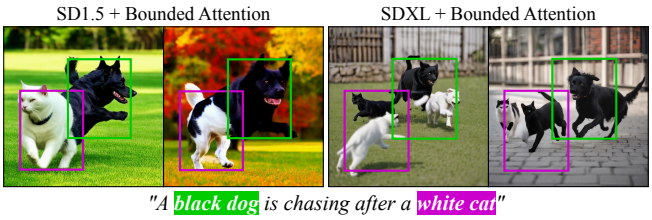


Fig. 14: Semantic separation of layout guidance methods (*e.g.*, Bounded Attention [21]) do not ensure instance discrimination.

Fig. 15: Limitation of Bounded Attention masking. Bounded Attention [21] enforces exclusive pixel ownership among bounding boxes on self-attention maps. Each box can only attend to its owned pixels and the background. We adopt a "smaller-box-first" ownership rule to build self-attention masks. Each mask visualizes pixels attendable from the corresponding layout. As shown, exclusivity suppresses attention in shared regions and degrades control.

C Additional Discussions

C.1 Multi-Instance Generation of Commercial Models

While commercial models such as GPT-Image-1.5 [56] and Nano Banana 2 [27] establish a high upper bound for multi-instance compositionality, a notable performance disparity remains between these closed-source systems and base open-weight architectures like Qwen-Image [77] and Flux.2-dev [9]. As demonstrated in Tab. 13 and Fig. 16, applying the ISAC objective consistently shifts the performance of open-weight models toward this commercial upper bound. By utilizing explicit attention manipulation, ISAC effectively bridges the empirical gap between accessible open-weight models and resource-intensive closed-source systems without the need for architectural modifications or fine-tuning.

Table 13: Performance comparison between open-weight models equipped with ISAC and commercial models on IntraCompBench. ISAC can be seamlessly applied to state-of-the-art open-weight baselines, such as Qwen-Image [77] and Flux.2-dev [9]. ISAC effectively narrows the performance gap with commercial counterparts [27, 56].

Method	Multi-Class Accuracy (↑)					Multi-Instance Accuracy (↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	#2	#3	#4	#5	Avg.	Latency	VRAM
Qwen-Image [77]	91%	45%	33%	10%	48%	98%	92%	84%	70%	86%	**140s**	**60.1GB**
+ **ISAC$_{LS}$ (Ours)**	**99%**	**58%**	**42%**	**25%**	**56%**	**99%**	**96%**	**89%**	**78%**	**91%**	210s	65.3GB
Flux.2-dev [9]	97%	95%	84%	78%	88%	**100%**	93%	81%	75%	87%	**205s**	**74.2GB**
+ **ISAC$_{LS}$ (Ours)**	**99%**	**98%**	**89%**	**83%**	**92%**	**100%**	**98%**	**88%**	**81%**	**92%**	305s	79.8GB
GPT-Image-1.5 [56]	99%	99%	98%	95%	97%	100%	100%	99%	94%	98%	N/A	N/A
Nano Banana 2 [27]	99%	97%	93%	92%	95%	100%	100%	100%	95%	99%	N/A	N/A

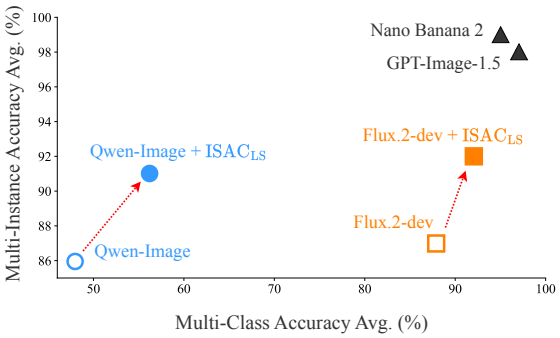


Fig. 16: Two-dimensional comparison of average performance on IntraCompBench. Across both Qwen-Image [77] and Flux.2-dev [9], ISAC moves open-weight models toward the upper-right, showing consistent gains on both metrics and narrowing the gap to commercial models.

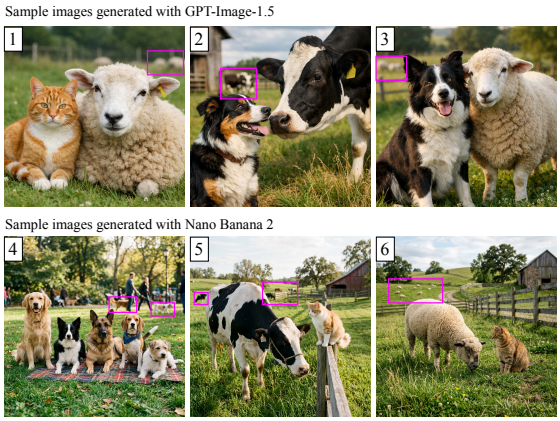


Fig. 17: Multi-instance generation samples from GPT-Image-1.5 [56] and Nano Banana 2 [27]. Used input prompts are: (1)*"A photo of a cat and a sheep"*, (2)*"A photo of a dog and a cow"*, (3)*"A photo of a dog and a sheep"*, (4)*"A photo of 5 dogs"*, (5)*"A photo of a cat and a cow"*, (6)*"A photo of a cat and a sheep"*.

Despite their robust performance on standardized benchmarks, qualitative assessments indicate that multi-instance generation remains a nuanced challenge even for advanced commercial systems [27, 56]; for example, they occasionally generate unwanted background instances when processing semantically adjacent classes (Fig. 17). These minor artifacts suggest that implicit text guidance does not perfectly resolve all boundary ambiguities, highlighting the continued scientific relevance of explicit structural interventions like ISAC.

C.2 Detector-Agnostic Evaluation of ISAC

In Tab. 14, we report a vision-language-model (VLM)-based evaluation of multi-instance generation, adopting the protocol of [62]. On the IntraCompBench suite, GPT-5.5 [58] independently reproduces ISAC's gains in both magnitude and correlation. These results confirm that ISAC's improvements are not driven by detector-specific biases.

Table 14: Detector and VLM evaluation on IntraCompBench.

Method	Detector Ensemble (↑)	GPT-5.5 [58] (↑)
SD1.5 [66]	8%	14%
+ISAC$_{\text{LO}}$	**36%**	**42%**
Pearson(r) vs. Detectors	-	0.73
SD3.5-M [23]	25%	32%
+ISAC$_{\text{LO}}$	**52%**	**58%**
Pearson(r) vs. Detectors	-	0.75

Table 15: Performance-aesthetic trade-off analysis on SDXL. We evaluate three aspects: Alignment (Class/Instance accuracy on IntraCompBench), Distribution (FID [30]/CMMD [37]), and Preference (PickScore [43]/HPSv3 [54]). For distribution and preference metrics, we generate images using the T2I-CompBench [34] Numeracy task. FID and CMMD measure the distance between the generated distribution and the native SDXL [61] baseline; lower values indicate less deviation from the original aesthetic manifold. Note that PickScore represents the win rate against the baseline.

Method	Alignment Metrics		Distribution Metrics		Preference Metrics	
	Class (↑)	Instance (↑)	FID (↓)	CMMD (↓)	PickScore (↑)	HPSv3 (↑)
SDXL [61]	7%	61%	0.0000	0.0000	0.4208	6.7230
+ **ISAC$_{\text{LO}}$ (Ours)**	34%	76%	16.402	0.0026	0.5792	7.1463

C.3 Effect on Perceptual Quality and Human Preference

We analyze whether ISAC$_{\text{LO}}$'s structural interventions affect the perceptual quality of generated images. In Tab. 15, we report *Distribution Metrics* (Frechet Inception Distance (FID) [30] and CLIP [63]-based Maximum Mean Discrepancy (CMMD) [37]) alongside *Human Preference Metrics* (PickScore [43], HPSv3 [54]).

It is important to note that preference models like PickScore [43] and HPSv3 [54] evaluate generation quality holistically, considering both text-image alignment and aesthetic fidelity. As shown in Tab. 15, ISAC$_{\text{LO}}$ demonstrates a clear superiority in human preference, achieving a significantly higher PickScore win rate ($0.42 < 0.58$) and improved HPSv3 score ($6.72 \rightarrow 7.15$) compared to the baseline SDXL [61]. Given that ISAC$_{\text{LO}}$ dramatically improves instance accuracy (as seen in earlier evaluations), this preference gain confirms that our method does not trade off visual quality for controllability. Instead, it achieves a superior balance, generating images that are not only structurally accurate but also more aligned with human visual preferences.

Regarding distribution shifts, while FID [30] shows a slight increase, the CMMD [37] remains negligible (0.0026). Since CMMD [37] is more robust to outliers and better correlates with human perception than FID [37], this result suggests ISAC keeps the native generative manifold without noticeable artifacts.

C.4 Sensitivity to Internal Instance Count

ISAC uses the parsed instance count N to form N self-attention clusters. To examine the effect of count mismatch, we perturb only the internal count used by ISAC to $N-1$ or $N+1$, while keeping the prompt and evaluation target unchanged. Relative to the default N setting reported in Tab. 2, multi-class accuracy drops by 22/16 percentage points for $N-1/N+1$ on SD1.5 [66] and by 25/18 percentage points on SD3.5-M [23]. This confirms that ISAC benefits from an accurate count prior. Using too few clusters forces distinct target instances to share a layout, weakening instance separation. On the other hand, using too many clusters fragments a single object into artificial sub-instances and disrupts semantic binding.

C.5 Effect of Self-Attention Clustering Frequency

ISAC clusters self-attention maps at every denoising step. As this is time-consuming, Fig. 18 analyzes the trade-off between recomputation frequency and accuracy. Together with the clustering algorithm comparison in Tab. 8, and the coordinate-augmentation analysis in Fig. 11, this result shows that ISAC's clustering is robust to algorithmic choices but benefits from frequent recomputation to track evolving attention layouts.

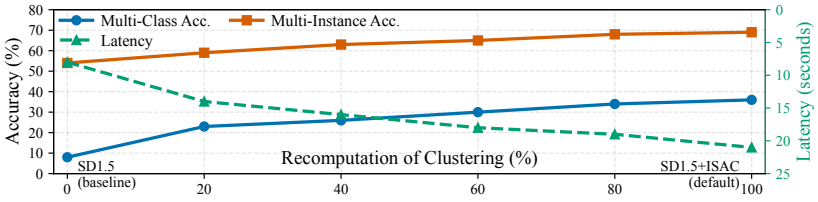


Fig. 18: Recomputation trade-off (SD1.5). Per-step (100%; right) maximizes accuracy; lower rate yields a smooth trade-off.

C.6 Choice of Loss Design

Table 16 evaluates alternative similarity metrics in place of MPO. Replacing MPO with MAE, KL, IoU or Dice consistently degrades performance, reducing multi-class accuracy from 36% to at most 21% and multi-instance accuracy from 69% to at most 61%. This gap indicates that MPO's margin-based separation between class-specific masks is better suited to enforcing instance-aware semantic decoupling than generic overlap measures.

Top-k% MPO alternatives, which average the largest k% overlaps instead of taking the maximum, are smoother but less effective than MPO. This gap indicates that focusing on peak overlap reduction is an effective design choice.

Table 16: Analysis of alternative similarity metrics in Eq. (8).

Method	Metric	IntraCompBench (↑)	
		Class	Instance
SD1.5 [66]	N/A	8%	54%
+ ISAC_{LO}	MAE	9% (+1%)	55% (+1%)
+ ISAC_{LO}	KL	16% (+8%)	60% (+6%)
+ ISAC_{LO}	IoU	20% (+12%)	61% (+7%)
+ ISAC_{LO}	Dice	21% (+13%)	61% (+7%)
+ ISAC_{LO}	top-10% MPO	28% (+19%)	63% (+9%)
+ ISAC_{LO}	top-5% MPO	31% (+23%)	65% (+11%)
+ ISAC_{LO}	MPO	**36%** (+28%)	**69%** (+15%)

C.7 Performance Stability of ISAC

Initial random seeds (X_T) significantly affect compositional generation [47]. Thus, we report seed-wise stability on the T2I-CompBench Numeracy task in Tab. 20 (IntraCompBench already averages 10 seeds/prompt). ISAC_{LO} reduces standard deviation by ∼3-4×, yielding qualitatively stable generations (Fig. 26). This demonstrates that the sharp MPO loss does not induce generation instability.

C.8 Integration with Fine-tuned Models

To evaluate complementarity, we apply ISAC to two externally supervised fine-tuned methods: TokenCompose [75], using Grounded SAM masks [65] for cross-attention alignment, and IterComp [83], employing preference-guided refinement with RPG [80] and InstanceDiffusion [74]. ISAC further improves multi-class and multi-instance accuracy (Tab. 17). Incidentally, TokenCompose requires `float32` weights, increasing latency and VRAM over standard SD1.4 [66].

Table 17: IntraCompBench evaluation of ISAC_{LO} on fine-tuned models [75, 83]

Method	Multi-Class Accuracy (↑)					Multi-Instance Accuracy (↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	#2	#3	#4	#5	Avg.	Latency	VRAM
$\text{TokenCompose}_{\text{SD1.4}}$ [75]	27%	4%	1%	0%	8%	77%	65%	41%	13%	49%	**12s**	**8.5GB**
+ **ISAC_{LO} (Ours)**	**62%**	**36%**	**28%**	**17%**	**36%**	**84%**	**80%**	**60%**	**30%**	**63%**	33s	17.9GB
$\text{IterComp}_{\text{SDXL}}$ [83]	11%	5%	4%	0%	5%	95%	73%	64%	37%	67%	**49s**	**11.8GB**
+ **ISAC_{LO} (Ours)**	**46%**	**28%**	**26%**	**21%**	**30%**	**99%**	**93%**	**85%**	**55%**	**83%**	100s	29.9GB

C.9 Limitation

ISAC lacks explicit 3D understanding, which can fail for prompts requiring depth ordering through transparent materials, as shown in Fig. 19. Future work will explore 3D- or physics-aware extensions.

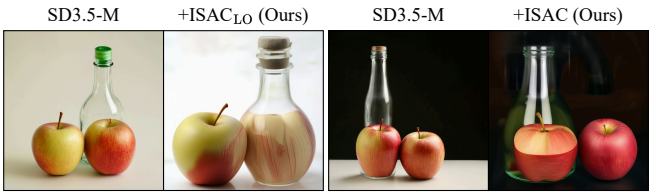


Fig. 19: Limitation of ISAC.

D Future Works

Most of ISAC's additional computational cost comes from applying backpropagation through *all* attention layers in the diffusion model. A natural way to reduce this overhead is to identify informative attention layers and apply backpropagation only to those. Ahn *et al.* [2] explore such informative layers and empirically find that specific attention layers in the mid-block of the U-Net [67] are particularly important for guidance. Fig. 20 and Tab. 18 show that ISAC exhibits a similar trend, suggesting that layer-wise selection is a promising direction for improving ISAC's efficiency.

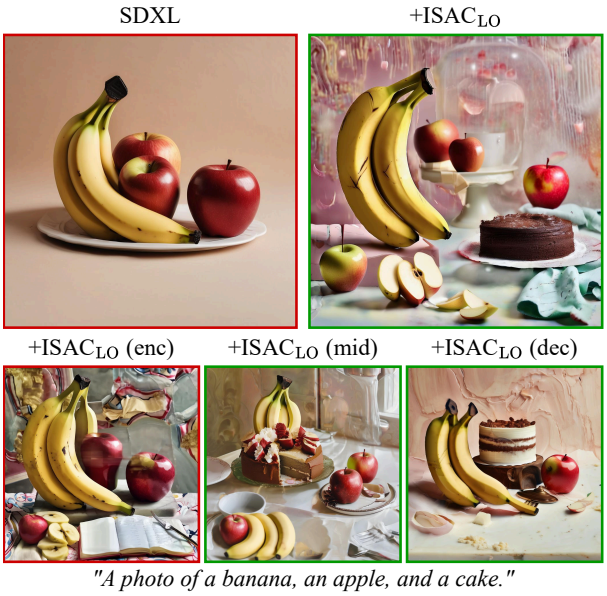


Fig. 20: ISAC$_{LO}$ applied to different subsets of SDXL [61] attention layers.

Table 18: Complexity analysis of SDXL [61] and ISAC$_{LO}$ on various layers.

Method	Latency (s) (↓)	VRAM (GB) (↓)
SDXL [61] w/o attention	**10.7**	**18.0**
+ ISAC$_{LO}$	111.4	30.7
+ ISAC$_{LO}$ (enc)	60.9	<u>21.2</u>
+ ISAC$_{LO}$ (mid)	<u>31.7</u>	**18.0**
+ ISAC$_{LO}$ (dec)	60.4	25.0

E Additional Diffusion Dynamics Visualization

In Fig. 21, we show the extended visualization of Fig. 4 to popular diffusion models [23, 66] and ISAC_{LO}'s dynamics on SD3.5-M [23]. The two-step instance formation and instance-centric semantic separation achieves highly reliable multi-object generation.

F Additional Quantitative Results

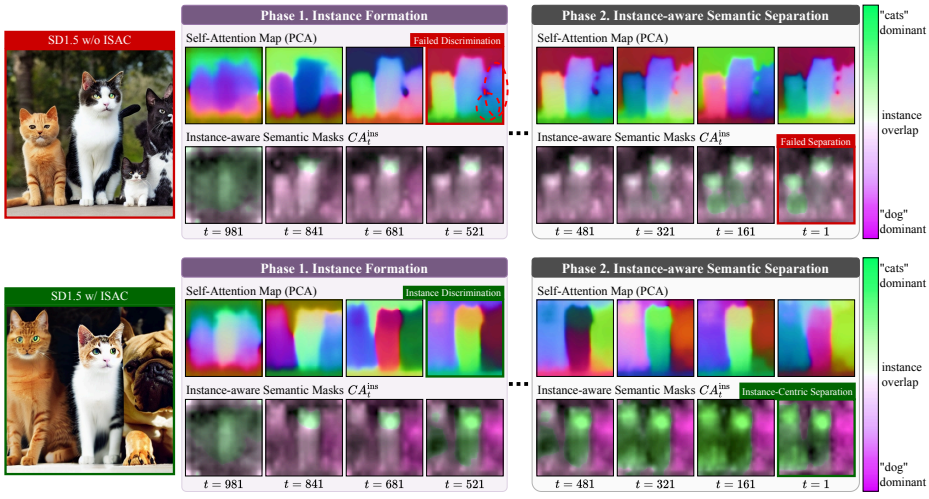
We provide full quantitative results on IntraCompBench metrics in Table 19. The results are obtained by applying ISAC_{LO} to various models, including SD1.4, SD1.5, SD2.1 [66], SDXL [61], SD3.5-M, SD3.5-L [23], PixArt-α [16], PixArt-Σ [15] and Flux.1-dev [8].

Also in Tab. 20, we provide extended quantitative results of ISAC_{LO} on IntraCompBench and public counting benchmark T2I-CompBench [34], where prompts specify exact counts. It is worth note that SD1.5 [66] with ISAC_{LO} surpasses baseline performance of plain SDXL [61] and ISAC_{LO} reduces performance variance from initial seeds (see Sec. C for details).

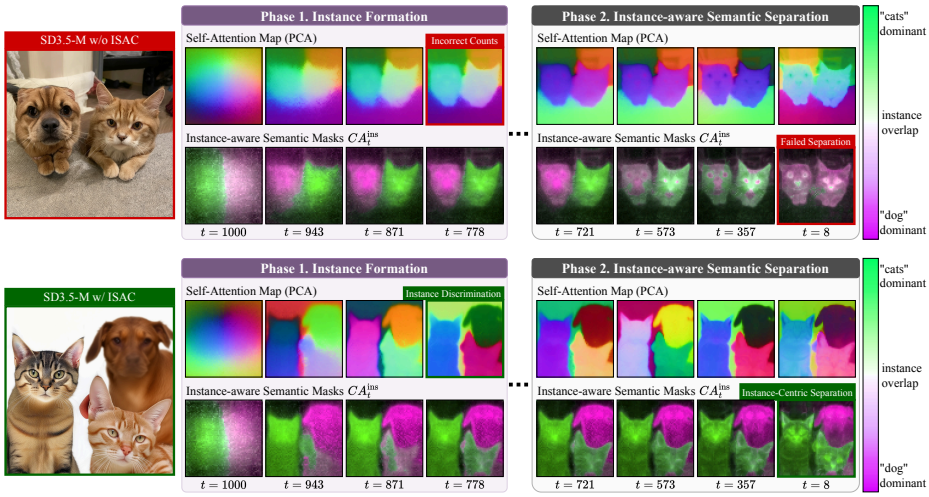
G Additional Qualitative Results

Generality beyond simple prompts. Figures 22 to 25 show additional qualitative comparisons of ISAC_{LO} with other attention control methods, InitNO [28], Self-Cross [62] and Attention-Refocusing [60]. The results show that ISAC_{LO} not only improves drawing multiple instances, but also improves assigning correct attributes to each instance. Beyond simple color attributes, ISAC_{LO} is also effective to shape, texture, positioning and the combination of them. This highlights ISAC's broader applicability.

Robust multi-instance generation across seeds. We provide qualitative results when generating 5 images with a fixed prompt, *"three cats"*, on SD1.5 [66] and SD3.5-M [23]. When the baseline output already contains the correct number of instances, ISAC_{LO} minimally alters the result. However, when the baseline produces too few or too many instances, ISAC_{LO} effectively corrects the output (see Figs. 26a and 26b). These results show that ISAC reliably helps correct instance counts across diverse seeds for a fixed prompt, while leaving correct samples mostly unchanged.



(a) Dynamics of SD1.5 with or without ISAC



(b) Dynamics of SD3.5-M with or without ISAC

Fig. 21: Diffusion dynamics visualization with or without $\mathbf{ISAC_{LO}}$. This is an extended version of Fig. 4 to SD1.5 [66] and SD3.5-M [23].

Table 19: Additional quantitative results of latent optimization methods on IntraCompBench.

Method	Multi-Class Accuracy (↑)					Multi-Instance Accuracy (↑)					Efficiency (↓)	
	#2	#3	#4	#5	Avg.	#2	#3	#4	#5	Avg.	Latency	VRAM
SD1.4 [66]	30%	2%	1%	0%	8%	94%	74%	28%	22%	55%	**8s**	**4.9GB**
+ A&E SIGGRAPH'23 [13]	50%	9%	8%	2%	17%	97%	79%	26%	23%	56%	17s	9.2GB
+ SynGen NeurIPS'23 [64]	54%	11%	6%	2%	18%	90%	69%	25%	19%	51%	19s	9.3GB
+ InitNO CVPR'24 [28]	58%	10%	7%	4%	20%	94%	79%	31%	20%	56%	20s	9.6GB
+ TEBOpt NeurIPS'24 [14]	55%	13%	8%	2%	19%	91%	73%	31%	20%	54%	17s	9.3GB
+ **ISAC$_{\text{LO}}$ (Ours)**	**66%**	**34%**	**29%**	**16%**	**36%**	**100%**	**90%**	**51%**	**40%**	**70%**	21s	9.7GB
SD1.5 [66]	28%	2%	1%	0%	8%	88%	65%	36%	26%	54%	**12s**	**4.4 GB**
+ A&E SIGGRAPH'23 [13]	48%	10%	5%	2%	16%	91%	68%	34%	24%	54%	24s	9.1 GB
+ SynGen NeurIPS'23 [64]	50%	9%	4%	2%	16%	84%	61%	38%	22%	51%	27s	9.2 GB
+ InitNO CVPR'24 [28]	55%	12%	7%	5%	20%	90%	68%	40%	29%	57%	29s	9.5 GB
+ Self-Cross CVPR'25 [62]	48%	8%	4%	2%	15%	89%	67%	38%	27%	55%	21s	10 GB
+ TEBOpt NeurIPS'24 [14]	52%	11%	8%	3%	18%	87%	65%	36%	27%	54%	25s	9.2 GB
+ **ISAC$_{\text{LO}}$ (Ours)**	**65%**	**31%**	**29%**	**18%**	**36%**	**95%**	**82%**	**56%**	**44%**	**69%**	30s	9.6 GB
SD2.1 [66]	31%	6%	3%	0%	10%	91%	74%	41%	28%	58%	**13s**	**4.8 GB**
+ A&E SIGGRAPH'23 [13]	53%	12%	4%	1%	18%	94%	79%	39%	29%	60%	26s	9.3 GB
+ SynGen NeurIPS'23 [64]	55%	10%	7%	3%	19%	87%	69%	38%	25%	55%	29s	9.4 GB
+ InitNO CVPR'24 [28]	59%	13%	11%	5%	22%	91%	79%	44%	26%	60%	31s	9.7 GB
+ TEBOpt NeurIPS'24 [14]	56%	14%	7%	6%	21%	88%	75%	44%	27%	58%	27s	9.4 GB
+ **ISAC$_{\text{LO}}$ (Ours)**	**67%**	**35%**	**34%**	**20%**	**39%**	**98%**	**88%**	**64%**	**42%**	**73%**	32s	9.8 GB
SDXL [61]	20%	4%	3%	0%	7%	90%	71%	49%	32%	61%	**48s**	**12.8 GB**
+ **ISAC$_{\text{LO}}$ (Ours)**	**57%**	**32%**	**29%**	**17%**	**34%**	**96%**	**89%**	**71%**	**47%**	**76%**	101s	29.8 GB
PixArt-α [16]	27%	3%	1%	0%	8%	99%	93%	33%	15%	60%	**17s**	**19.9 GB**
+ **ISAC$_{\text{LO}}$ (Ours)**	**63%**	**30%**	**29%**	**21%**	**36%**	**100%**	**100%**	**56%**	**31%**	**72%**	40s	53.7 GB
PixArt-Σ [15]	39%	8%	0%	0%	12%	98%	98%	30%	16%	60%	**18s**	**19.9 GB**
+ **ISAC$_{\text{LO}}$ (Ours)**	**78%**	**39%**	**31%**	**20%**	**42%**	**100%**	**100%**	**48%**	**31%**	**70%**	41s	53.8 GB
SD3.5-M [23]	62%	23%	12%	3%	25%	84%	71%	51%	51%	64%	**40s**	**22.9 GB**
+ A&E SIGGRAPH'23 [13]	65%	29%	16%	5%	28%	86%	72%	52%	50%	65%	124s	73.8 GB
+ SynGen NeurIPS'23 [64]	66%	28%	15%	6%	28%	82%	68%	50%	48%	62%	131s	74.3 GB
+ InitNO CVPR'24 [28]	77%	31%	17%	7%	33%	84%	73%	52%	49%	65%	138s	74.6 GB
+ Self-Cross CVPR'25 [62]	78%	38%	19%	3%	34%	86%	72%	51%	50%	65%	147s	76.4 GB
+ TEBOpt NeurIPS'24 [14]	78%	31%	19%	8%	34%	85%	71%	52%	52%	65%	139s	74.5 GB
+ **ISAC$_{\text{LO}}$ (Ours)**	**98%**	**51%**	**40%**	**20%**	**52%**	**98%**	**91%**	**72%**	**69%**	**83%**	140s	74.8 GB

Table 20: Quantitative results on T2I-CompBench [34] Numeracy and IntraCompBench multi-instance tasks. Performance stability across initial random seeds is measured on T2I-CompBench, with 30 seeds. **Bold** shows the best and underline shows the second best performance.

Method	# Parameters	T2I-CompBench	IntraCompBench (Multi-Instance) (↑)				
		Numeracy (↑)	#2	#3	#4	#5	Avg.
SD1.5 [66]	0.8B	46.5% ± 5.1%	88%	65%	36%	26%	54%
+ **ISAC$_{\text{LO}}$ (Ours)**	0.8B	<u>55.0%</u> ± 1.4%	<u>95%</u>	<u>82%</u>	<u>56%</u>	<u>44%</u>	<u>69%</u>
SDXL [61]	2.6B	51.1% ± 4.8%	90%	71%	49%	32%	61%
+ **ISAC$_{\text{LO}}$ (Ours)**	2.6B	**64.0%** ± 1.2%	**96%**	**89%**	**71%**	**47%**	**76%**

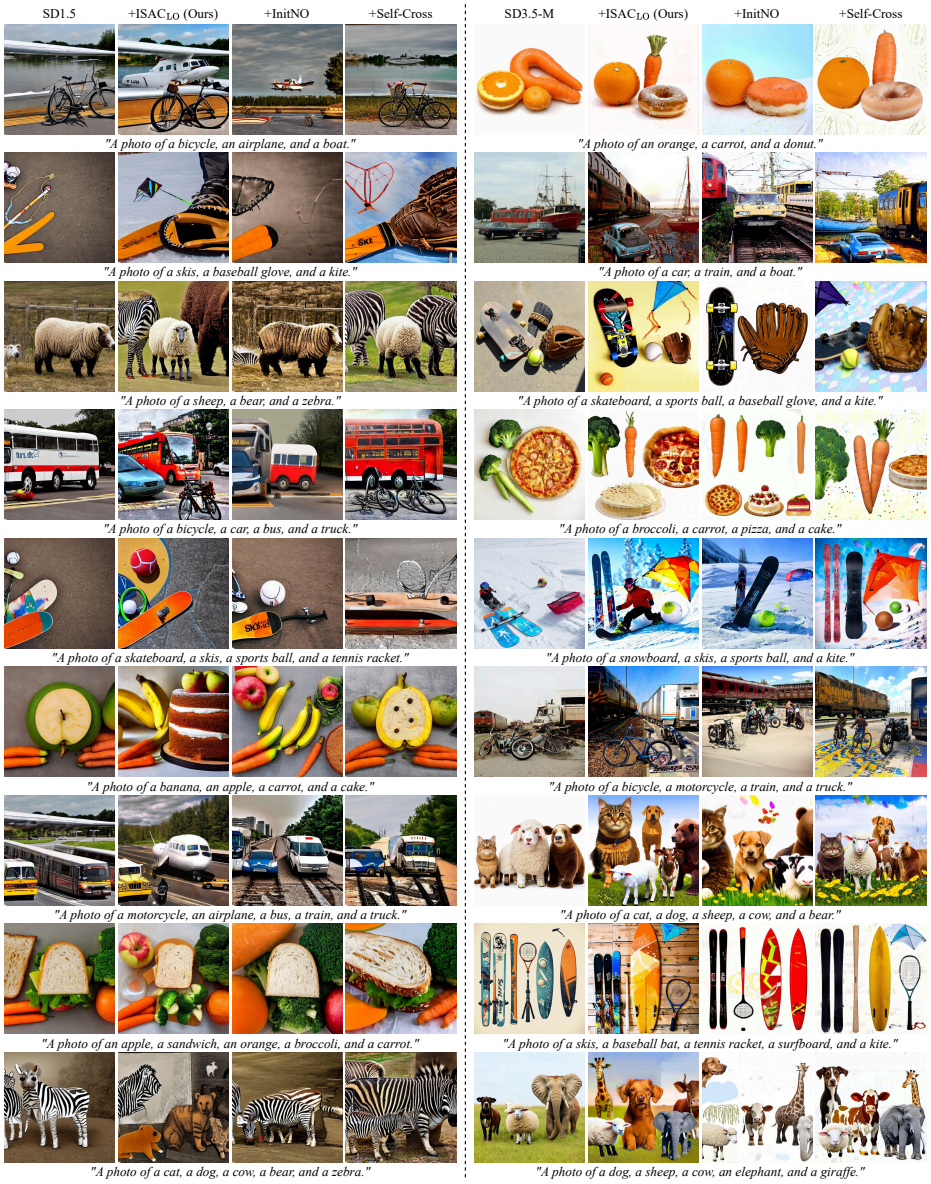


Fig. 22: Qualitative comparisons of attention control methods, InitNO [28], Self-Cross [62] and ISAC$_{LO}$, in 3 to 5 intra-category generation. All prompts are drawn from IntraCompBench.

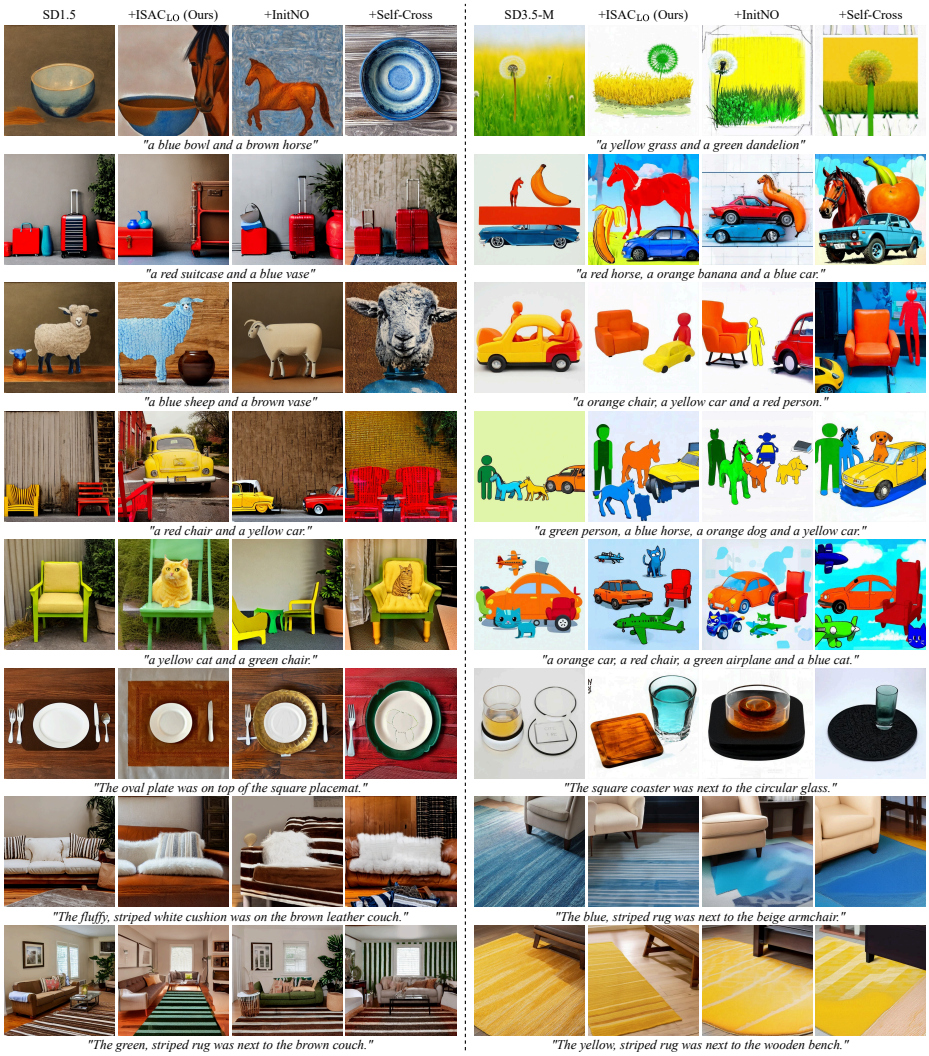


Fig. 23: Qualitative comparisons of attention control methods, InitNO [28], Self-Cross [62] and ISAC$_{LO}$, in complex scene generation. It requires the model to correctly bind attributes such as color, texture, spatial, and shape. All prompts are drawn from HRS-Bench [4] and T2I-CompBench [34].

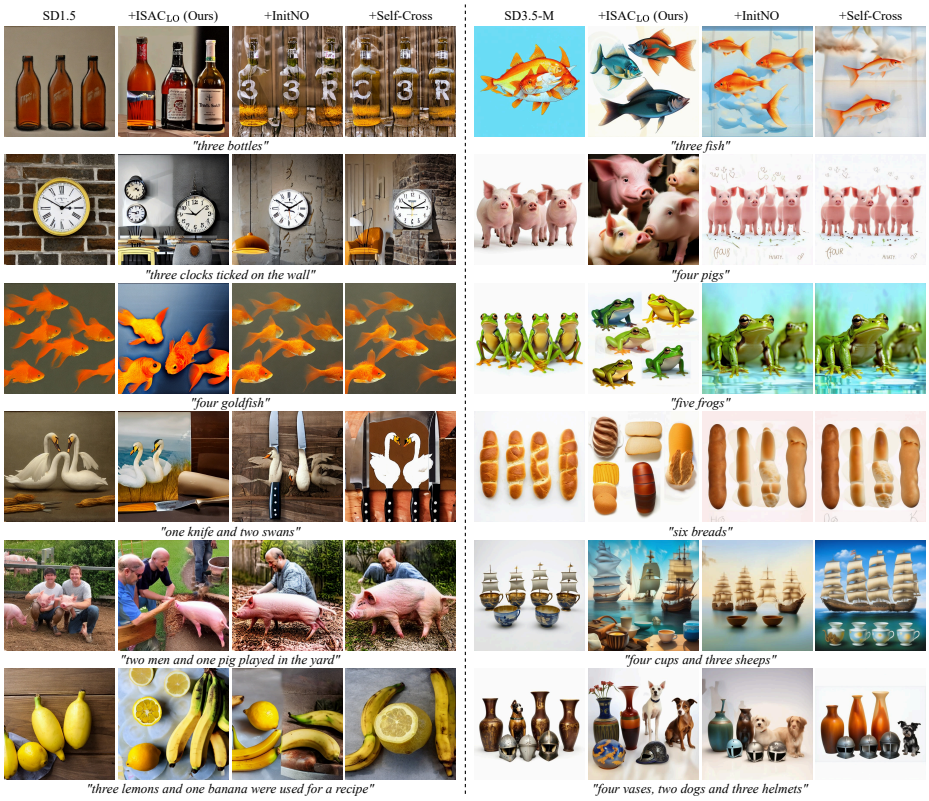
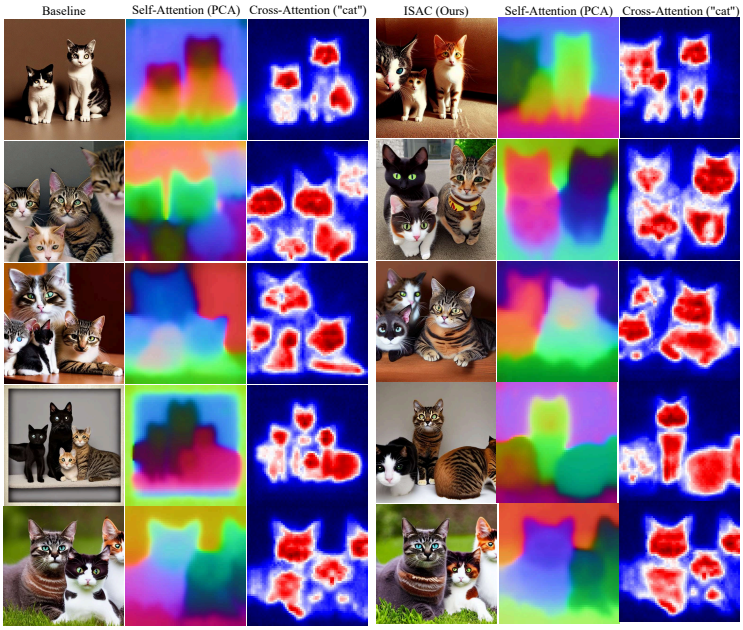


Fig. 24: Qualitative comparisons of attention control methods, InitNO [28], Self-Cross [62] and ISAC$_{LO}$, in general multi-instance scenarios. It requires the model to correctly generate multiple instances from one or more classes. All prompts are drawn from T2I-CompBench [34] Numeracy task.

Fig. 25: Qualitative comparisons of ISAC$_{LO}$ with layout guidance method, Attention Refocusing (CAR&SAR) [60] on top of finetuned layout model GLIGEN [48]. All prompts are drawn from HRS-Bench [34].



(a) Result from Stable Diffusion v1.5 [66].

(b) Result from Stable Diffusion v3.5-M [23].

Fig. 26: Qualitative results across multiple seeds for the prompt *"three cats"*, which involves multiple instances of the same class. Our method (right) consistently generates images with the correct instance count, sharper object boundaries, and improved separation compared to the baseline (left).