

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

Sanghyun Jo<sup>1,2,\*†</sup> , Wooyeol Lee<sup>2,\*</sup> , Ziseok Lee<sup>2,\*</sup> , Jonghyun Choi<sup>2</sup> ,  
Jaesik Park<sup>2</sup> , and Kyungsu Kim<sup>2†</sup> 

<sup>1</sup> OGQ, Seoul, Korea

<sup>2</sup> 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** (Instance-to-Semantic Attention Control), 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

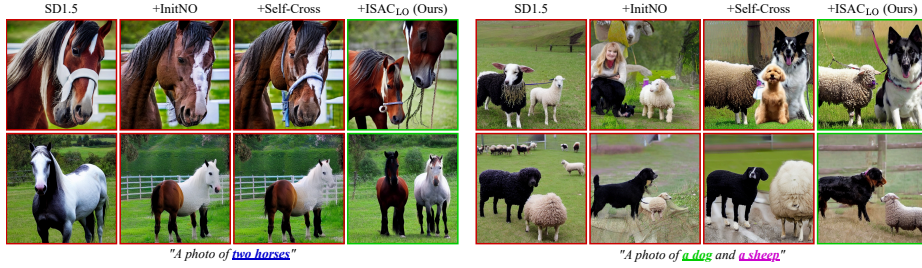
## 1 Introduction

Text-to-image (T2I) generative models [6, 15, 23, 47] now produce highly realistic images from text prompts. In particular, recent commercial systems (*e.g.*, Nano Banana 2 [23] and GPT-Image [40]) establish a strong empirical upper bound on multi-instance, multi-attribute prompts. Such performance is often supported by massive proprietary data pipelines [1, 21] and expensive MLLM-driven reasoning loops, such as verifying interim images and refining the composition [22]. Although few failure cases are observable in these systems (see Appendix C), their closed

---

\* Equal contribution.

† Corresponding authors: [shjo.april@gmail.com](mailto:shjo.april@gmail.com), [kyskim@snu.ac.kr](mailto:kyskim@snu.ac.kr)



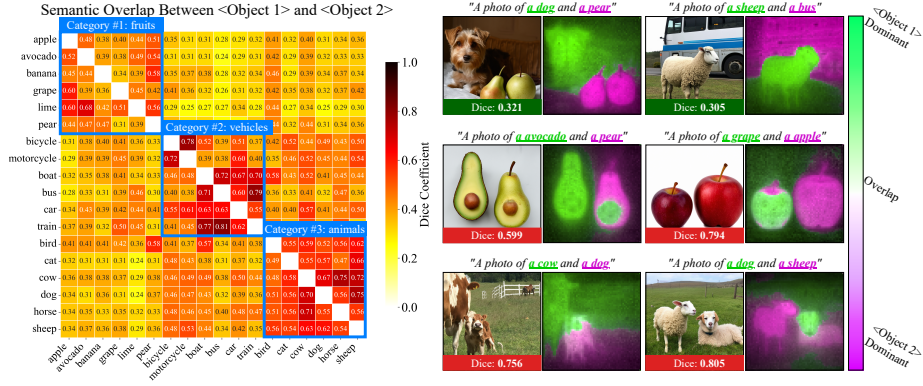
**Fig. 1: Qualitative comparison for representative multi-instance prompts.** Generated samples are compared across the baseline SD1.5 [47], prior methods (InitNO [24], Self-Cross [44]), and our ISAC<sub>LO</sub> (ISAC with latent optimization). Text prompts are provided below each group.

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 [6] and Qwen-Image [52]) 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 [10, 11, 24, 27, 44] 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 [19] 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.

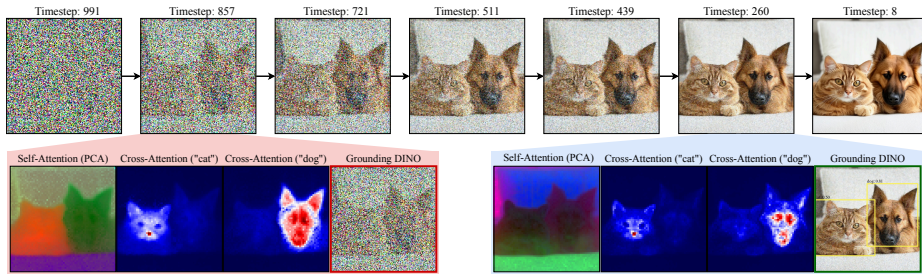
We instantiate this hierarchy as ISAC (**I**nstance-to-**S**emantic **A**ttention **C**ontrol), a training-free hierarchical objective for latent optimization or latent selection. ISAC separates structural formation from semantic binding along the denoising trajectory. Phase 1 ( $\mathcal{L}_{\text{ins}}$ ) exploits the observation that self-attention already exposes emerging instance layouts in early steps [30], 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 [2, 6–8, 12, 13, 19, 43, 47, 49, 52] and layout-to-image controllers [36]. 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 <small>NeurIPS’24</small> [11]         | ✗                                 | ✓                       | ✗                           | ✗                                    | ✓                                      |
| DOS <small>AAAI’26</small> [9]                | ✗                                 | ✓                       | ✗                           | ✗                                    | ✓                                      |
| A&E <small>SIGGRAPH’23</small> [10]           | ✗                                 | ✓                       | ✓                           | ✗                                    | ✓                                      |
| InitNO <small>CVPR’24</small> [24]            | ✗                                 | ✓                       | ✓                           | ✗                                    | ✓                                      |
| SynGen <small>NeurIPS’23</small> [46]         | ✗                                 | ✓                       | ✓                           | ✗                                    | ✓                                      |
| CONFORM <small>CVPR’24</small> [39]           | ✗                                 | ✓                       | ✓                           | ✗                                    | ✓                                      |
| Self-Cross <small>CVPR’25</small> [44]        | ✗                                 | ✓                       | ✓                           | ✗                                    | ✓                                      |
| CountGen <small>CVPR’25</small> [5]           | ✗                                 | ✓                       | ✗                           | ✓                                    | ✗                                      |
| Counting Guidance <small>WACV’25</small> [31] | ✗                                 | ✓                       | ✗                           | ✓                                    | ✗                                      |
| ISAC (Ours)                                   | ✓                                 | ✓                       | ✓                           | ✓                                    | ✓                                      |

**Fig. 3: Dynamics of text-to-image diffusion models.** In the early stages of diffusion, instance structures emerge [30] while semantics remain underdeveloped. In later diffusion steps, instance structures are stabilized, and semantic refinements happen. Because detection models (*e.g.*, [38]) 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 [19].

- 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 [28], HRS-Bench [3], and IntraCompBench, over multiple diffusion backbones and both text- and layout-to-image generation. ISAC improves multi-object metrics by at least  $1.9\times$  over the baseline [47] and outperforms prior counting-supervised methods [5, 31] by at least 10% accuracy without additional training.

## 2 Related Work

**Training-free text-to-image guidance.** While text embedding approaches [9, 11, 20, 27], such as TEBOpt [11] and DOS [9], 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 [45] architecture limits compatibility with state-of-the-art architectures (*e.g.*, Flux.2 [7], Qwen-Image [52])

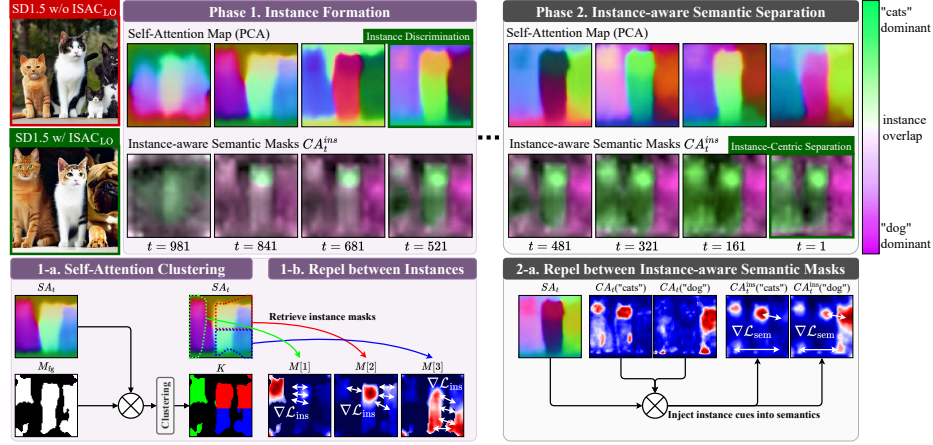


that leverage modern advances in large language models (LLMs). Alternatively, cross-attention (CA) reveals the spatial footprint of textual semantics [25], enabling spatial guidance. Attend-and-Excite [10] boosts attention peaks to recover neglected objects, while SynGen [46] and CONFORM [39] separate and bind CA maps using contrastive objectives and parser-derived relations. InitNO [24] and Self-Cross [44] 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 [37, 57]. While state-of-the-art controllers provide dense layout control [16, 29, 36, 51, 56, 58, 59], they often struggle with overlapping layouts [35, 55]. Training-free layout-to-image methods [4, 14, 18, 32, 34, 42, 48, 53, 54] help, yet they do not ensure instance discrimination. Constraining attention within each box either by excluding background [42] or other instances’ layouts [18] 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 [31], 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 [5, 17], 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 [28] and HRS-Bench [3] 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 [10] and SSD [44] 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 injects 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 [25, 33], 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  $\epsilon_\theta$  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 [5, 31], 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  $\epsilon_\theta$  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

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

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

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:

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

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 [30], 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  $M_{\text{fg}}$  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:

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

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

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

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

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

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

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  $M[1], \dots, M[N]$  over foreground pixels. To separate instances, we penalize their worst local overlap using the *maximum pixel-wise overlap* (MPO):

$$\text{MPO}(A, B) = \max_{p \in \{1, \dots, F\}} A[p] \cdot B[p], \quad (8)$$

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

$$\mathcal{L}_{\text{ins}}(X_t) = \max_{1 \leq i < j \leq N} \text{MPO}(M[i], M[j]). \quad (9)$$

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  $P_{\text{repel}}$  denote pairs of token indices that should remain distinct (*e.g.*, different classes/instances), and  $P_{\text{bind}}$  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:

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

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

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

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

While using such token relations is common [39, 46], 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

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

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 [10, 39, 44, 46], we primarily use ISAC for latent optimization (ISAC<sub>LO</sub>) as summarized in Algorithm 1. ISAC is also compatible with other guidance schemes, such as latent selection (ISAC<sub>LS</sub>; see Sec. 4.3).

---

#### Algorithm 1: ISAC with Latent Optimization (ISAC<sub>LO</sub>)

---

**Input:** Prompt  $\mathcal{T}$ , Model  $\epsilon_\theta$ , decoder  $\mathcal{D}$ , step size  $\eta$   
**Output:** Image  $I_0$

- 1  $X_T \sim \mathcal{N}(0, I)$
- 2 **for**  $t = T, T-1, \dots, 1$  **do**
- 3     Call  $\text{Denoise}(X_t, \mathcal{T}, \epsilon_\theta, t)$  with **Hooks**  $\rightarrow SA_t, CA_t$
- 4      $CA_t^{\text{ins}} \leftarrow SA_t \cdot CA_t$
- 5     Build foreground gate and instance masks (Eqs. 5, 6, 7)
- 6     Compute  $\mathcal{L}_{\text{ins}}, \mathcal{L}_{\text{sem}}$  (Eqs. 9, 12)
- 7      $\mathcal{L}_{\text{ISAC}}(X_t, t) \leftarrow \lambda_{\text{ins}}(t) \mathcal{L}_{\text{ins}}(X_t) + \lambda_{\text{sem}}(t) \mathcal{L}_{\text{sem}}(X_t)$
- 8      $\tilde{X}_t \leftarrow X_t - \eta \cdot \nabla_{X_t} \mathcal{L}_{\text{ISAC}}(X_t, t)$
- 9      $X_{t-1} \leftarrow \text{Denoise}(\tilde{X}_t, \mathcal{T}, \epsilon_\theta, t)$
- 10  $I_0 \leftarrow \mathcal{D}(X_0)$  // 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 [28], HRS-Bench [3], and our IntraCompBench. T2I-CompBench [28] and HRS-Bench [3] 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 [38, 50] and a prompt-specific matching procedure between detections and requested classes (see Appendix A for details of our IntraCompBench).

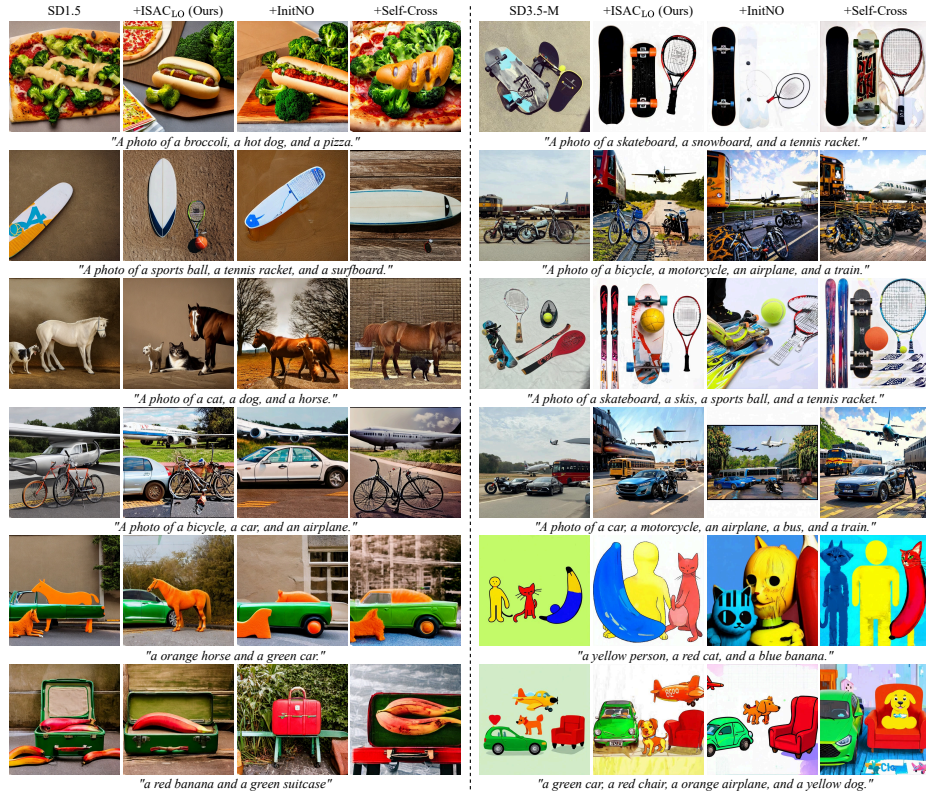
**Implementation details.** We strictly follow the official inference configurations for all diffusion models [6, 19, 43, 47, 52]. We apply the ISAC objective (Eq. (13)) in latent optimization and latent selection, two common training-free approaches for multi-instance generation. Both variants share 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<sub>LO</sub>) requires a single tuned hyperparameter,  $\eta = 0.01$ , shared across all models and benchmarks. Sensitivity analysis for  $\eta$  is provided in Appendix A. For all experiments involving ISAC with latent selection (ISAC<sub>LS</sub>), we choose the best-out-of-10. For each text prompt, we automatically extract class tag-count pairs  $(\tau_i, n_i)$  and simple token relations ( $P_{\text{repe}}l$  and  $P_{\text{bind}}$ ) using the open-sourced GPT-OSS [41] 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 [19, 47]. With similar inference cost compared to attention control methods [10, 24, 44, 46], ISAC<sub>LO</sub> clearly outperforms them on every metric of HRS-Bench, T2I-CompBench, and the multi-class setting of IntraCompBench. Text embedding approaches [9, 11] 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 [47], the average multi-class accuracy on IntraCompBench increases from 20% for the strongest baseline [9, 24] 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 [3], T2I-CompBench [28], 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 [47]                        | 0.136        | 0.094        | 0.091        | 0.356            | 0.406        | 0.306        | 28%                      | 2%         | 1%         | 0%         | 8%         | <b>8s</b>     | <b>4.9 GB</b>  |
| + A&E SIGGRAPH’23 [10]            | 0.149        | 0.104        | 0.101        | 0.392            | 0.447        | 0.290        | 48%                      | 10%        | 5%         | 2%         | 16%        | 17s           | 9.2 GB         |
| + SynGen NeurIPS’23 [46]          | 0.159        | 0.111        | 0.107        | 0.420            | 0.479        | 0.311        | 50%                      | 9%         | 4%         | 2%         | 16%        | 19s           | 9.3 GB         |
| + InitNO CVPR’24 [24]             | 0.175        | 0.120        | 0.116        | 0.456            | 0.520        | 0.338        | 55%                      | 12%        | 7%         | 5%         | 20%        | 20s           | 9.6 GB         |
| + TEBOpt NeurIPS’24 [11]          | 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 [44]         | 0.170        | 0.118        | 0.114        | 0.445            | 0.508        | 0.324        | 48%                      | 8%         | 4%         | 2%         | 15%        | 21s           | 10 GB          |
| + DOS AAAI’26 [9]                 | 0.191        | 0.135        | 0.121        | 0.468            | 0.531        | 0.341        | 56%                      | 14%        | 7%         | 4%         | 20%        | 11s           | 6.5 GB         |
| <b>+ ISAC<sub>LO</sub> (Ours)</b> | <b>0.318</b> | <b>0.263</b> | <b>0.252</b> | <b>0.683</b>     | <b>0.631</b> | <b>0.354</b> | <b>65%</b>               | <b>31%</b> | <b>29%</b> | <b>18%</b> | <b>36%</b> | 21s           | 9.7 GB         |
| SD3.5-M [19]                      | 0.425        | 0.264        | 0.209        | 0.796            | 0.726        | 0.377        | 62%                      | 23%        | 12%        | 3%         | 25%        | <b>40s</b>    | <b>22.9 GB</b> |
| + A&E SIGGRAPH’23 [10]            | 0.427        | 0.263        | 0.215        | 0.798            | 0.726        | 0.378        | 65%                      | 29%        | 16%        | 5%         | 28%        | 124s          | 73.8 GB        |
| + SynGen NeurIPS’23 [46]          | 0.425        | 0.260        | 0.211        | 0.801            | 0.718        | 0.365        | 66%                      | 28%        | 15%        | 6%         | 28%        | 131s          | 74.3 GB        |
| + InitNO CVPR’24 [24]             | 0.443        | 0.275        | 0.228        | 0.810            | 0.728        | 0.378        | 77%                      | 31%        | 17%        | 7%         | 33%        | 138s          | 74.6 GB        |
| + TEBOpt NeurIPS’24 [11]          | 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 [44]         | 0.431        | 0.268        | 0.219        | 0.795            | 0.720        | 0.371        | 78%                      | 38%        | 19%        | 3%         | 34%        | 147s          | 76.4 GB        |
| + DOS AAAI’26 [9]                 | 0.440        | 0.273        | 0.231        | 0.808            | 0.729        | 0.380        | 79%                      | 33%        | 18%        | 7%         | 34%        | 44s           | 29.1GB         |
| <b>+ ISAC<sub>LO</sub> (Ours)</b> | <b>0.473</b> | <b>0.350</b> | <b>0.258</b> | <b>0.838</b>     | <b>0.739</b> | <b>0.388</b> | <b>98%</b>               | <b>51%</b> | <b>40%</b> | <b>20%</b> | <b>52%</b> | 140s          | 74.8 GB        |



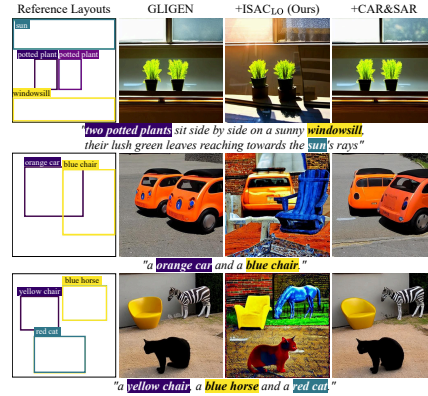
**Fig. 5: Qualitative comparison of attention control methods on SD1.5 [47] and SD3.5-M [19].**

**Table 3:** Comparison of ISAC<sub>LO</sub> and count-supervised methods [5, 31] on IntraCompBench.

| Method                           | IntraCompBench(Instance)(↑) |            |            |            |            | Efficiency (↓) |        |
|----------------------------------|-----------------------------|------------|------------|------------|------------|----------------|--------|
|                                  | #2                          | #3         | #4         | #5         | Avg.       | Latency        | VRAM   |
| SD1.4 [47]                       | 94%                         | 74%        | 28%        | 22%        | 55%        | 8s             | 4.9GB  |
| + CG <sub>WACV22</sub> [31]      | 79%                         | 67%        | 32%        | 19%        | 49%        | 14s            | 17.5GB |
| + ISAC <sub>LO</sub> (Ours)      | <b>100%</b>                 | <b>90%</b> | <b>51%</b> | <b>40%</b> | <b>70%</b> | 21s            | 9.7GB  |
| SDXL [43]                        | 90%                         | 71%        | 49%        | 32%        | 61%        | 48s            | 12.8GB |
| + CountGen <sub>CVPR23</sub> [5] | <b>97%</b>                  | 83%        | 52%        | 44%        | 69%        | 100s           | 55.3GB |
| + ISAC <sub>LO</sub> (Ours)      | 96%                         | <b>89%</b> | <b>71%</b> | <b>47%</b> | <b>76%</b> | 101s           | 29.8GB |

**Fig. 6:** Qualitative results of ISAC<sub>LO</sub> for exact-instance-count prompts on SD1.5 [47] and SD3.5-M [19].**Table 4:** Application of ISAC<sub>LO</sub> to layout-to-image generation.

| Method                      | HRSBench (↑)  |              |                |              |
|-----------------------------|---------------|--------------|----------------|--------------|
|                             | Counting (F1) | Color (Acc.) | Spatial (Acc.) | Size (Acc.)  |
| GLIGEN [36]                 | 0.666         | 0.307        | 0.268          | 0.188        |
| + CAR&SAR [42]              | 0.675         | 0.402        | 0.277          | 0.263        |
| + ISAC <sub>LO</sub> (Ours) | <b>0.713</b>  | <b>0.452</b> | <b>0.281</b>   | <b>0.275</b> |

**Fig. 7:** Qualitative results of ISAC<sub>LO</sub> with the GLIGEN [36] 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 [24, 44] 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 [19, 47]. 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<sub>LO</sub> achieves 70% and 76% instance-counting accuracy on SD1.4 [47] and SDXL [43], surpassing count-supervised methods (49% for Counting Guidance [31] and 69% for CountGen [5]) even though all methods are given the same instance-count supervision and ISAC<sub>LO</sub> 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<sub>LO</sub> 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 [26] 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<sub>LO</sub> also improves layout-to-image controllers [36]. On HRS-Bench, ISAC<sub>LO</sub> yields the largest gains compared with layout refinements [42]. 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<sub>LO</sub> 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<sub>LO</sub> 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 [6, 7] and Qwen-Image [52], ISAC<sub>LS</sub> (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<sub>LS</sub> on multi-object generation. Figure 8 further shows that the ISAC<sub>LS</sub> score distinguishes samples with missing instances or semantic mixing from better candidates.

---

**Algorithm 2:** ISAC with Latent Selection (ISAC<sub>LS</sub>)

---

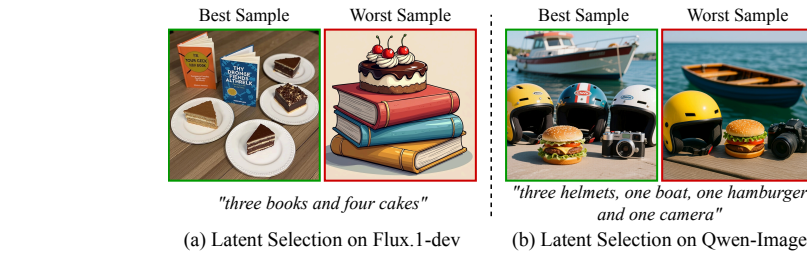
**Input:** Prompt  $\mathcal{T}$ , Model  $\epsilon_\theta$ , decoder  $\mathcal{D}$ , Batch size  $B$   
**Output:** Image  $I_0$

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

---

**Table 5: Best-of-10 latent selection on IntraCompBench.** The strategy is applied with ISAC using the base models [6, 7, 52].

| Method                      | Multi-Class Accuracy ( $\uparrow$ ) |            |            |            |            | Multi-Instance Accuracy ( $\uparrow$ ) |            |            |            |            | Efficiency ( $\downarrow$ ) |               |
|-----------------------------|-------------------------------------|------------|------------|------------|------------|----------------------------------------|------------|------------|------------|------------|-----------------------------|---------------|
|                             | #2                                  | #3         | #4         | #5         | Avg.       | #2                                     | #3         | #4         | #5         | Avg.       | Latency                     | VRAM          |
| Flux.1-dev [6]              | 84%                                 | 37%        | 3%         | 2%         | 31%        | 97%                                    | 89%        | 82%        | 66%        | 83%        | <b>50s</b>                  | <b>37.2GB</b> |
| + ISAC <sub>LS</sub> (Ours) | <b>97%</b>                          | <b>48%</b> | <b>38%</b> | <b>19%</b> | <b>51%</b> | <b>99%</b>                             | <b>94%</b> | <b>85%</b> | <b>72%</b> | <b>88%</b> | 85s                         | 40.8GB        |
| Qwen-Image [52]             | 91%                                 | 45%        | 33%        | 10%        | 48%        | 98%                                    | 92%        | 84%        | 70%        | 86%        | <b>140s</b>                 | <b>60.1GB</b> |
| + ISAC <sub>LS</sub> (Ours) | <b>99%</b>                          | <b>58%</b> | <b>42%</b> | <b>25%</b> | <b>56%</b> | <b>99%</b>                             | <b>96%</b> | <b>89%</b> | <b>78%</b> | <b>91%</b> | 210s                        | 65.3GB        |
| Flux.2-dev [7]              | 97%                                 | 95%        | 84%        | 78%        | 88%        | <b>100%</b>                            | 93%        | 81%        | 75%        | 87%        | <b>205s</b>                 | <b>74.2GB</b> |
| + ISAC <sub>LS</sub> (Ours) | <b>99%</b>                          | <b>98%</b> | <b>89%</b> | <b>83%</b> | <b>92%</b> | <b>100%</b>                            | <b>98%</b> | <b>88%</b> | <b>81%</b> | <b>92%</b> | 305s                        | 79.8GB        |

**Fig. 8:** Qualitative results of latent selection with ISAC<sub>LS</sub> scoring.**Table 6:** Application of ISAC<sub>LO</sub> on few-step text-to-image diffusion models.

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

**Applicability to few-step models.** A key concern is whether ISAC<sub>LO</sub> remains effective under the limited inference budgets of few-step models. We validate our method on Z-Image-Turbo [49] and Flux.2-klein-4B [8], which utilize only 8 and 4 steps, respectively. The results in Tab. 6 demonstrate that ISAC<sub>LO</sub> 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 ISAC<sub>LO</sub>.** “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 | $t/T$                     | $1 - t/T$                 | <b>36%</b> | <b>69%</b> |

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) [RS-2026-25507282(5%), RS-2026-25518317(5%), No.RS-2024-00405857(10%), No.05-26-04-0094, No.RS-2026-25472075, No.RS-2026-25483206, No.RS-2025-02305581, No.RS-2025-25442338, and No.RS-2021-II211343(5%)], the ITIP grant funded by the Korean government (MOTIR) [No.RS-2026-25549946], the Research grant from SNU, and the Strategic Hub grant for International Research Collaboration of SNU.

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. 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., Kargapol'tseva, 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)
3. 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)
4. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: MultiDiffusion: Fusing diffusion paths for controlled image generation. In: ICML. pp. 1737–1752 (2023)
5. 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)
6. Black Forest Labs: FLUX. <https://github.com/black-forest-labs/flux> (2024), accessed 2026-03-05
7. Black Forest Labs: FLUX.2: Frontier Visual Intelligence. <https://bfl.ai/blog/flux-2> (2025), accessed 2026-03-05
8. Black Forest Labs: FLUX.2-klein-4B. <https://huggingface.co/black-forest-labs/FLUX.2-klein-4B> (2026), accessed 2026-03-05
9. 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)
10. 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)
11. 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)
12. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: PixArt- $\Sigma$ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In: ECCV. pp. 74–91 (2024)
13. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: PixArt- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. In: ICLR. pp. 57611–57640 (2024)
14. Chen, M., Laina, I., Vedaldi, A.: Training-free layout control with cross-attention guidance. In: WACV. pp. 5343–5353 (2024)
15. 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)
16. Cheng, B., Ma, Y., Wu, L., 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)
17. 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)

18. 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)
19. 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)
20. 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)
21. Gemini Team: Gemini 2.5: Pushing the frontier with advanced reasoning, multi-modality, long context, and next generation agentic capabilities. [https://storage.googleapis.com/deepmind-media/gemini/gemini\\_v2\\_5\\_report.pdf](https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf) (2025), accessed 2026-03-05
22. 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
23. 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
24. 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)
25. 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)
26. Hobley, M., Prisacariu, V.: Learning to count anything: Reference-less class-agnostic counting with weak supervision. In: CVPR Workshop on Transformers for Vision (2023)
27. 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)
28. 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)
29. 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)
30. 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)
31. 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)
32. Kim, D., Lee, J., Park, J.: Improving editability in image generation with layer-wise memory. In: CVPR. pp. 7889–7898 (2025)
33. 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)
34. Lee, Y., Yoon, T., Sung, M.: GrounDiT: Grounding diffusion transformers via noisy patch transplantation. In: NeurIPS. pp. 58610–58636 (2024)
35. 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)



36. 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)
37. 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)
38. 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)
39. 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)
40. OpenAI: GPT-Image-1.5 (2025), <https://platform.openai.com/docs/models/gpt-image-1.5>, accessed 2026-03-05
41. OpenAI: gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925 (2025)
42. Phung, Q., Ge, S., Huang, J.B.: Grounded text-to-image synthesis with attention refocusing. In: CVPR. pp. 7932–7942 (2024)
43. 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)
44. Qiu, W., Wang, J., Tang, M.: Self-cross diffusion guidance for text-to-image synthesis of similar subjects. In: CVPR. pp. 23528–23538 (2025)
45. 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)
46. 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)
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
48. 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)
49. 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)
50. Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., Ding, G.: YOLOE: Real-time seeing anything. In: ICCV. pp. 24591–24602 (2025)
51. Wang, X., Darrell, T., Rambhatla, S.S., Girdhar, R., Misra, I.: InstanceDiffusion: Instance-level control for image generation. In: CVPR. pp. 6232–6242 (2024)
52. 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)
53. 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)
54. 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)

- 55. 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)
- 56. 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)
- 57. 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)
- 58. dewei Zhou, Xie, J., Yang, Z., Yang, Y.: 3DIS: Depth-driven decoupled image synthesis for universal multi-instance generation. In: ICLR (2025)
- 59. 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)