

# OverLayBench: A Benchmark for Layout-to-Image Generation with Dense Overlaps

Bingnan Li<sup>1\*</sup> Chen-Yu Wang<sup>1\*</sup> Haiyang Xu<sup>1\*</sup> Xiang Zhang<sup>1</sup> Ethan Armand<sup>1</sup>  
Divyansh Srivastava<sup>1</sup> Xiaojun Shan<sup>1</sup> Zeyuan Chen<sup>1</sup> Jianwen Xie<sup>2</sup> Zhuowen Tu<sup>1</sup>  
<sup>1</sup>UC San Diego <sup>2</sup>Lambda, Inc  
\*equal contribution

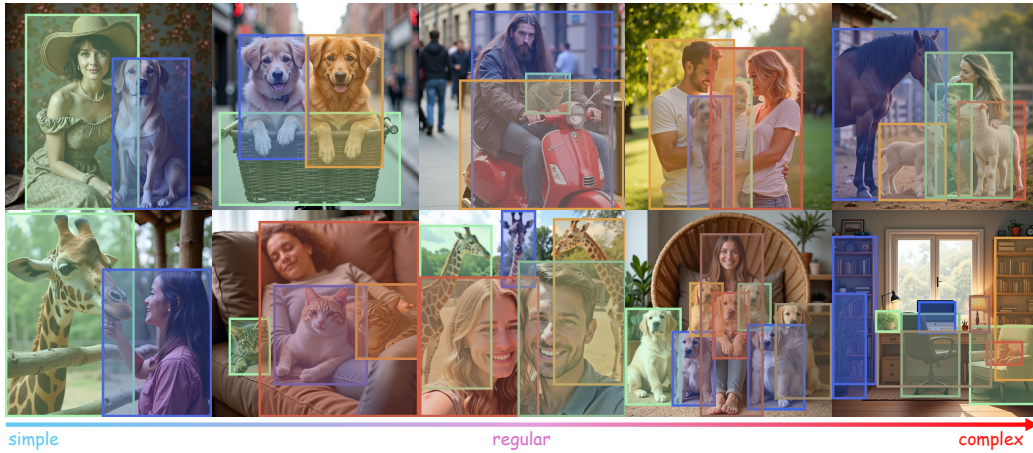


Figure 1: Examples from OverLayBench with difficulty increasing from left to right.

## Abstract

Despite steady progress in layout-to-image generation, current methods still struggle with layouts containing significant overlap between bounding boxes. We identify two primary challenges: (1) large overlapping regions and (2) overlapping instances with minimal semantic distinction. Through both qualitative examples and quantitative analysis, we demonstrate how these factors degrade generation quality. To systematically assess this issue, we introduce OverLayScore, a novel metric that quantifies the complexity of overlapping bounding boxes. Our analysis reveals that existing benchmarks are biased toward simpler cases with low OverLayScore values, limiting their effectiveness in evaluating model performance under more challenging conditions. To bridge this gap, we present OverLayBench, a new benchmark featuring high-quality annotations and a balanced distribution across different levels of OverLayScore. As an initial step toward improving performance on complex overlaps, we also propose CreatiLayout-AM, a model fine-tuned on a curated amodal mask dataset. Together, our contributions lay the groundwork for more robust layout-to-image generation under realistic and challenging scenarios. *Project link: <https://mlpc-ucsd.github.io/OverLayBench>.*

## 1 Introduction

With the advancement of text-to-image generative models [Ramesh et al., 2021, Nichol et al., 2021, Rombach et al., 2022, Chen et al., 2023, Xue et al., 2024], there has been growing interest



Figure 2: Failure cases from state-of-the-art layout-to-image models. Each row presents an example with overlapping instances, and image captions are shown below. More examples and more detailed failure descriptions can be found in Appendix C.

in *controllable image generation* [Li et al., 2023b, Zhang et al., 2023]. A recent line of work proposes generating images conditioned on layouts, commonly referred to as **Layout-to-Image (L2I)** generation, which allows users to directly specify spatial locations [Xie et al., 2023b, Wang et al., 2024b, Li et al., 2023b] and object counts [Binyamin et al., 2024, Yang et al., 2023] in the generated outputs. While existing frameworks [Xie et al., 2023b, Wang et al., 2024b, Li et al., 2023b] can achieve satisfactory spatial and numerical control over image generation, these approaches fail to generate distinct, coherent objects when multiple bounding boxes overlap in layout and their associated categories are semantically similar. As illustrated in Figure 2, such scenarios lead to artifacts including object blending, spatial ambiguity, and visual distortion.

To quantify the effect of bounding box overlap on generation, we introduce **OverLayScore** (Equation (1)), a simple metric that captures the difficulty of generation based on spatial and semantic overlap between bounding boxes. OverLayScore is computed as the sum of IoUs for all instance pairs, weighted by their semantic similarity, measured via the dot product of CLIP embeddings of instance annotations. We empirically demonstrate that generation quality degrades with higher OverLayScore, i.e., large overlap between bounding boxes and high category-level semantic similarity. Henceforth, we will interchangeably refer to layouts with high OverLayScore score as complex layouts and low OverLayScore score as simple layouts.

Existing benchmarks for Layout-to-Image (L2I) generation [Cheng et al., 2024, Zhang et al., 2024] primarily focus on image quality, offering limited evaluation of complex layouts or the accuracy of generated instance relationships. Our analysis reveals a strong bias towards simple layouts in these benchmarks, which restricts their utility in assessing model performance under more challenging, realistic scenarios. To address this gap, we introduce **OverLayBench**, a new benchmark specifically designed to evaluate L2I models on their ability to reconstruct complex layouts and instance-level relationships. OverLayBench features rich annotations such as detailed image and dense instance captions, enrichment of complex images with higher OverLayScore, improved semantic grounding of bounding boxes using Qwen [Bai et al., 2023], and quality instance relationships for evaluation. We conduct extensive evaluations of state-of-the-art L2I models [Li et al., 2023a, Wang et al., 2024a, Zhou et al., 2024a, Cheng et al., 2024, Zhou et al., 2024b, Zhang et al., 2025, 2024] on OverLayBench and verify their effectiveness in addressing layout-level challenges. These results provide strong baselines and highlight areas for improvement.

Finally, we demonstrate that fine-tuning CreatiLayout [Zhang et al., 2024] with amodal mask supervision on complex layouts helps mitigate generation artifacts caused by instance occlusion. This new baseline, **CreatiLayout-AM**, provides initial evidence that explicit mask-level guidance improves generation quality under high-overlap conditions, offering a promising direction for future research.

Our contributions are summarized as follows: (1) We propose **OverLayScore**, a novel metric that empirically quantifies the difficulty of L2I generation by measuring the IoU and semantic similarity between bounding boxes in layouts; (2) We introduce **OverLayBench**, a challenging benchmark with high-quality annotations and balanced difficulty distribution, designed to evaluate complex

relationships between instances in layouts; (3) We demonstrate that training with amodal masks helps alleviate generation artifacts in overlapping regions. Specifically, we present a simple yet effective baseline that aligns attention maps in diffusion models with amodal mask supervision.

## 2 Related Work

### 2.1 Layout-to-Image Generation

Generative models have recently become highly popular [Wang et al., 2024d, Zhao et al., 2025, Zeng et al., 2025, Srivastava et al., 2025, Xu et al., 2024, Wang et al., 2024c, Chen et al., 2025], and controllable generation is attracting growing interest. In particular, Layout-to-Image (L2I) generation has gained attention as it enables structured and spatially grounded image synthesis.

Prior work on L2I generation has primarily focused on fine-tuning foundational text-to-image (T2I) generative models [Labs, 2024, Esser et al., 2024, Podell et al., 2023], introducing various techniques to inject layout conditioning into pre-trained architectures. U-Net-based approaches [Li et al., 2023a, Zhou et al., 2024a, Wang et al., 2024a, Wu et al., 2024, Cheng et al., 2024, Zheng et al., 2023] typically incorporate layout information through layer insertions (either in series or parallel) or by manipulating attention maps and masks. While these methods have shown promising results, their generation quality is often constrained by the representational capacity of U-Net backbones. More recent methods [Zhang et al., 2025, 2024] leverage powerful diffusion Transformer (DiT) architectures [Peebles and Xie, 2023], fine-tuning models such as Flux [Labs, 2024] or SD3 [Esser et al., 2024], and achieve improved image fidelity. These layouts can be provided by the user or generated automatically from text using text-to-layout models [Feng et al., 2023a, Gani et al., 2023, Lian et al., 2023, Srivastava et al., 2025], forming a two-stage pipeline for controllable image synthesis. In parallel, a growing line of training-free approaches [Xie et al., 2023a, Gani et al., 2023, Yang et al., 2024b, Lee et al., 2024, Chen et al., 2024a,c] has emerged, which utilize guidance mechanisms within diffusion models to enforce spatial constraints without additional training. However, these models often treat layout information as a soft constraint, which limits their ability to strictly adhere to spatial specifications.

Despite these advancements, existing methods struggle to generate coherent and distinguishable objects when multiple bounding boxes overlap, particularly when the associated categories are semantically similar. Such scenarios frequently lead to confusion, ambiguity, visual distortion, or artifacts in the generated images. To address the gap, we study Layout-to-Image (L2I) generation under bounding box overlap in layouts and highlight it as a novel and underexplored challenge that demands dedicated investigation.

### 2.2 Layout-to-Image Benchmarks

The most widely used benchmark for L2I generation is COCO [Lin et al., 2014], which provides image-text pairs annotated with entity bounding boxes (bboxes) and simple category labels. Although COCO and its variants are commonly used for evaluating L2I models, they lack detailed image and instance-level captions, which limits their utility for training and evaluating in more semantically rich generation tasks [Feng et al., 2023b]. To address this shortcoming, many works manually augment COCO with additional instance-level captions to support more comprehensive evaluation.

Recent efforts [Cheng et al., 2024, Zhang et al., 2024] have introduced new layout benchmarks to offer more robust and holistic assessments. HiCo [Cheng et al., 2024] proposes HiCo-7k, a benchmark containing 7,000 carefully curated samples from GriT-20M [Peng et al., 2023], while CreatiLayout [Zhang et al., 2024] introduces LayoutSAM, comprising 5,000 samples selected from the SAM-1B [Kirillov et al., 2023] dataset. Both benchmarks employ image/entity filtering and generate detailed annotations. However, they rely on GroundingDINO [Liu et al., 2023] for bounding box extraction, which, according to its authors, exhibits limited recognition performance and frequently produces false positives in detection outputs (as discussed in Section 4.1). HiCo-7k partially addresses these issues through manual human curation, whereas LayoutSAM is generated without human intervention.

In contrast, OverLayBench leverages Qwen2.5-VL-32B [Bai et al., 2025], which surpasses GroundingDINO in benchmark performance and our own qualitative evaluation. Additionally, we incorporate human curation to further ensure data quality. Most importantly, unlike previous benchmarks that

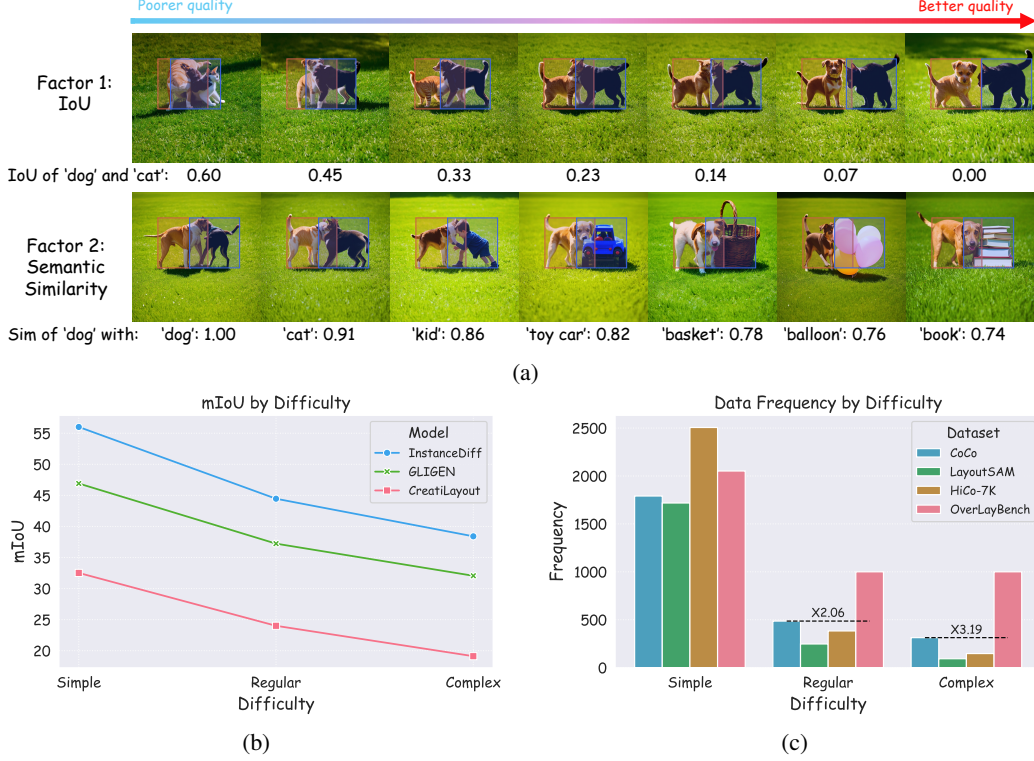


Figure 3: (a) Image quality of CreatiLayout across varying levels of bounding box IoU and semantic similarity, from poorer to better; (b) Performance comparison of three L2I models on toy COCO samples, grouped by layout difficulty using OverLayScore—higher difficulty consistently leads to lower mIoU; (c) Distribution of layout difficulty across COCO, LayoutSAM, HiCo-7k, and our proposed benchmark, OverLayBench (introduced in detail in Section 4.1).

primarily feature simple or regular layouts, our benchmark includes significantly more complex and challenging layouts, providing a more balanced distribution of layout difficulty and enabling more rigorous evaluation of L2I models.

### 3 OverLayScore

We evaluate the impact of overlap on L2I generation performance from two perspectives: spatial and semantic. Figure 3 (a) presents the L2I generation results in a simplified two object setting, from which we derive two key observations: (1) as the Intersection-over-Union (IoU) between bounding boxes increases, the output quality of state-of-the-art L2I models deteriorates; and (2) given the same IoU, a higher semantic similarity between the instance captions of overlapping boxes further degrades the generation quality. These findings indicate that both spatial and semantic overlaps introduce complexities that negatively affect L2I generation.

Building on our observations, we propose OverLayScore, a metric designed to quantify the difficulty of L2I generation arising from overlapping elements within a layout. Formally, given a layout with  $K$  objects, let  $p_k$  and  $B_k$  denote the instance caption and normalized bounding box for the  $k$ -th object, respectively. We define the metric as:

$$\text{OverLayScore} = \sum_{(i,j): \text{IoU}(B_i, B_j) > 0} \text{IoU}(B_i, B_j) \cdot \cos(\langle p_i, p_j \rangle), \quad (1)$$

where  $\cos(\langle p_i, p_j \rangle)$  is the CLIP-based cosine similarity between instance caption  $p_i$  and  $p_j$ . The summation captures the spatial-semantic entanglement of all overlapping object pairs in the layout. A higher OverLayScore indicates greater expected difficulty for an L2I model to faithfully generate an image that conforms to both the layout and the associated instance-level semantics.



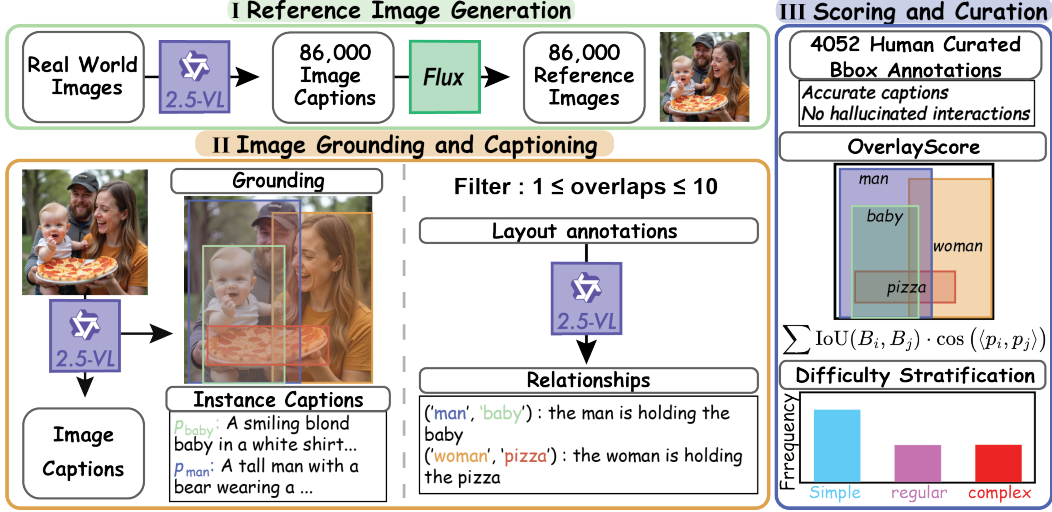


Figure 4: An overview of the data curation pipeline for OverLayBench.

To validate the effectiveness of OverLayScore, we evaluate common L2I models on a subset of the COCO dataset [Lin et al., 2014]. Layouts are extracted using the dataset’s bounding box and category annotations, and filtered to include scenes with 2 to 10 objects. Based on their OverLayScore, these layouts are categorized into three difficulty levels – simple, regular, and complex. From each category, we randomly sample 100 layouts for evaluation. We then assess the performance of three representative L2I models, i.e., GLIGEN [Li et al., 2023a], InstanceDiffusion [Wang et al., 2024a], and CreatiLayout [Zhang et al., 2024]. As illustrated in Figure 3 (b), the performance of all models consistently declines as OverLayScore increases, demonstrating that OverLayScore effectively reflects the difficulty of generating images from overlapping layouts.

We further apply OverLayScore to several widely used L2I benchmarks—COCO [Lin et al., 2014], HiCo [Cheng et al., 2024], and LayoutSAM [Zhang et al., 2024], and visualize the score distribution in Figure 3 (c). We observe that the majority of samples fall within the low-difficulty regime, indicating a strong imbalance in existing benchmarks. This skew limits their ability to evaluate model performance in more complex overlapping scenarios.

## 4 OverLayBench

The analysis in the previous section highlights a key limitation of existing L2I benchmarks—they are heavily skewed toward low-OverLayScore samples, which restricts their ability to evaluate model performance under challenging layout conditions. To overcome this limitation, we introduce OverLayBench—a new benchmark specifically curated to assess L2I models on complex and overlapping layouts. By carefully selecting images across a broad range of OverLayScore values, OverLayBench provides a more balanced and comprehensive evaluation set, enabling rigorous evaluation of model robustness in spatially and semantically complex layouts.

### 4.1 Dataset Curation

An overview of our data curation pipeline is presented in Figure 4, comprising three key stages. In Stage I, we use Flux to generate reference images based on captions extracted from real-world images. Stage II leverages a vision-language model (VLM) to extract both image and instance-level descriptions, along with inter-instance relationships. Finally, Stage III involves a human curation process to filter out unrealistic generation and balance the distribution across difficulty levels.

#### 4.1.1 Reference Image Generation

We begin by extracting image captions from the COCO [Lin et al., 2014] training set using Qwen2.5-VL-7B [Bai et al., 2025]. These captions are then used to generate a diverse set of image candidates

with Flux.1-dev [Labs, 2024]. By leveraging captions derived from real-world images, we ensure that the generated content and corresponding layouts are both natural and realistic. In total, we collect approximately 86,000 generated images paired with their corresponding captions.

#### 4.1.2 Image Grounding and Captioning

**Step 1: Image Caption Refinement** Although Flux demonstrates strong image generation capabilities, the generated images do not always perfectly align with the input captions. To improve semantic consistency, we perform an additional captioning pass on all generated images using Qwen-2.5-VL-7B to produce refined global image captions.

**Step 2: Instance Grounding** Qwen [Bai et al., 2023] has demonstrated superior grounding performance compared to models commonly used in existing L2I benchmarks, such as GroundingDINO [Liu et al., 2023]. We leverage Qwen to detect and describe all foreground objects in each image. Based on the grounding results, we retain only images that contain one to ten valid overlapping bounding box pairs. A bounding box pair is considered valid if it satisfies both: (1) an IoU greater than 5%, and (2) an intersection area exceeding 1% of the total image area. After this step, each image contains a global description and local descriptions for all detected instances.

**Step 3: Relationship Extraction** In addition to image-level and instance-level captions and bounding boxes, we further prompt Qwen to generate pairwise relationship phrases between overlapping instances. These phrases capture both spatial and semantic relationships, providing a richer annotation signal crucial for evaluating model performance on inter-instance relationship in complex layouts.

#### 4.1.3 Scoring and Curation

To ensure high-quality and reliable annotations while minimizing hallucinations from VLMs, we perform thorough manual verification and discard all invalid cases. Specifically, we assess the accuracy of each bounding box, the alignment between image content and both global and instance-level captions, and the validity of relationship descriptions based on available textual inputs. This rigorous process ensures that OverLayBench remains free from hallucinations. After validation, we compute the OverLayScore score for each example and retain a curated dataset of 2,052 simple, 1,000 regular, and 1,000 complex layouts.

#### 4.2 Benchmark Metrics

We introduce two novel metrics tailored to overlapping scenarios: **O-mIoU** (Overlap-mIoU), which computes the mIoU within the ground-truth overlap regions and the corresponding predicted regions. By isolating shared areas between instances, O-mIoU provides a more sensitive and discriminative measure of fidelity in occluded or entangled regions than standard global mIoU; **SR<sub>R</sub>** (Success Rate of Relationship), which reports the percentage of object pairs whose predicted spatial relationships match the ground truth. It offers an interpretable, relationship-level measure of success.

In addition to these two metrics, we adopt commonly used evaluation measures from prior work [Zhang et al., 2025, 2024], including mIoU, CLIP [Radford et al., 2021], SR<sub>E</sub> (Success Rate of Entity), and FID [Heusel et al., 2017].

### 5 CreatiLayout-AM: Amodal Masks Improve Generation of Occluded Objects

To address the challenges posed by complex overlaps in state-of-the-art L2I models, we incorporate amodal mask supervision during training, which provide complete object shape information even under occlusion. We construct a custom L2I training set annotated with amodal masks and train a new model to validate the effectiveness of this approach.

**Training Dataset** We begin by synthesizing occlusions on top of FLUX-generated images. For each image, we employ Segment Anything Model v2 [Ravi et al., 2024] to extract amodal object masks. These masks are then used to crop individual objects, forming a pool of object-mask pairs denoted as  $\mathcal{O}$ . To simulate occlusions, we randomly select an object from  $\mathcal{O}$  and paste it onto a

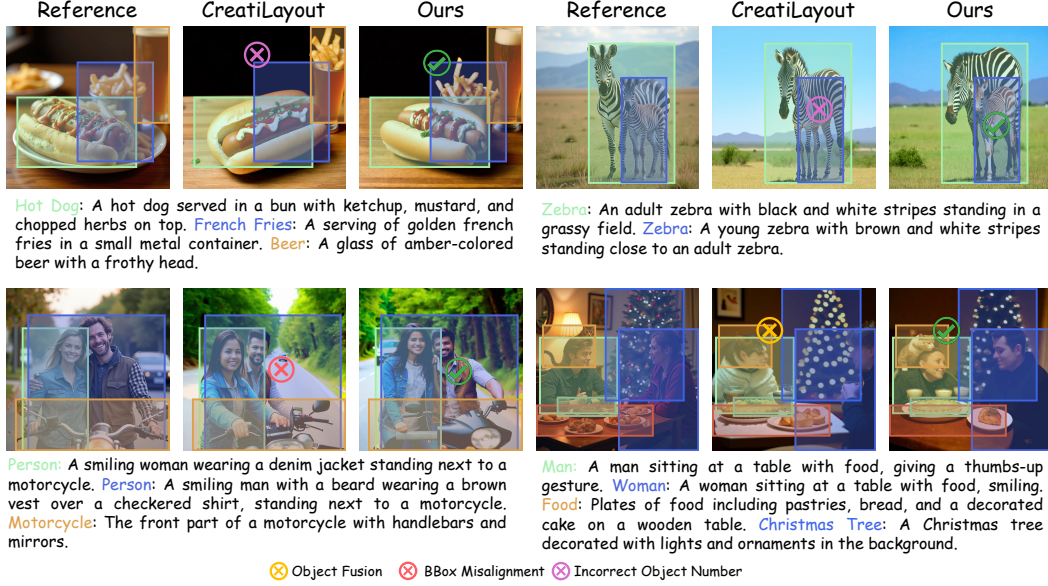


Figure 5: Comparison of generated images from CreatiLayout and CreatiLayout-AM. The CreatiLayout-AM handles overlapping instances more effectively, producing more coherent and realistic images.

target image at a location that creates overlap with an existing object. This method enables controlled occlusion synthesis while preserving the original scene context.

For each synthesized image, we use Qwen-2.5-VL-32B to generate both a global image caption and local instance descriptions, covering both original and newly pasted overlapping objects. The final training set contains approximately 67.8k images.

**CreatiLayout-AM** Building on our curated training dataset, we introduce CreatiLayout-AM, a modified version of CreatiLayout-SD3 [Zhang et al., 2024] designed to enhance generation quality in the presence of occluded bounding boxes by incorporating amodal masks during training. Inspired by TokenCompose [Wang et al., 2024e], we fine-tune CreatiLayout by introducing two additional loss terms that explicitly encourage alignment between the model’s attention maps and ground truth amodal masks.

Specifically, we compute two auxiliary losses,  $\mathcal{L}_{\text{token}}$  and  $\mathcal{L}_{\text{pixel}}$ , in addition to the original training objective. Let  $\mathcal{A}^i$  denote the attention map between image tokens and the layout token corresponding to the  $i^{\text{th}}$  instance, and let  $m^i$  be the ground-truth amodal mask for that instance. The token-level alignment loss is defined as:

$$\mathcal{L}_{\text{token}} = \frac{1}{n} \sum_{i=1}^n \left( 1 - \frac{\sum_u \mathcal{A}_u^i \cdot m_u^i}{\sum_u \mathcal{A}_u^i} \right) \quad (2)$$

where  $n$  is the total number of instances and  $u$  indexes each pixel coordinate. The pixel-level alignment loss is defined using a cross-entropy function:

$$\mathcal{L}_{\text{pixel}} = \sum_u \text{CE}(\mathcal{A}_u^i, m_u^i) \quad (3)$$

where CE is the cross entropy loss. The final training objective is given by:

$$\mathcal{L} = \mathcal{L}_{\text{LDM}} + \lambda \mathcal{L}_{\text{token}} + \beta \mathcal{L}_{\text{pixel}} \quad (4)$$

where  $\mathcal{L}_{\text{LDM}}$  is the original denoising loss used in the Latent Diffusion Models (LDM).

Beyond CreatiLayout [Zhang et al., 2024], we also implemented an EliGen [Zhang et al., 2025] based AM method, please find the detail in Appendix A.2.1.

Table 1: Comprehensive comparisons between training-based methods on OverLayBench, including the newly-released model. **Bold** and underline denote the best and the second best methods. Methods above the dashed line are U-Net-based, while those below are DiT-based. † means the model takes additional depth map as input.

Method	mIoU(%) ↑	O-mIoU(%) ↑	SR <sub>E</sub> (%) ↑	SR <sub>R</sub> (%) ↑	CLIP <sub>Global</sub> ↑	CLIP <sub>Local</sub> ↑	FID ↓
<b>OverLayBench-Simple</b>							
GLIGEN	60.54±1.82	36.22±0.13	49.99±0.43	78.72±0.50	34.17±0.02	24.75±0.02	31.27±0.38
InstanceDiff	<b>71.21</b> ±0.11	<b>49.99</b> ±0.06	77.71±0.18	87.49±0.23	34.25±0.05	27.69±0.02	36.17±0.23
MIGC	58.64±0.18	32.15±0.31	63.41±0.53	81.60±0.19	33.07±0.05	26.49±0.10	31.64±0.11
HiCo	69.47±0.26	47.23±0.44	67.75±0.18	86.08±0.76	35.25±0.10	27.04±0.09	29.21±0.16
3DIS	65.75±0.07	38.38±0.23	86.24±0.17	86.98±0.24	35.85±0.05	<u>29.67</u> ±0.03	29.18±0.29
CreatiLayout-SD3	58.78±0.44	32.52±0.61	72.34±0.61	84.45±0.04	37.29±0.04	27.49±0.03	27.51±0.15
CreatiLayout-FLUX	<u>71.17</u> ±0.23	<u>49.80</u> ±0.52	84.35±0.49	<b>90.87</b> ±0.27	<b>37.40</b> ±0.15	20.18±0.11	<b>23.79</b> ±0.17
EliGen	68.17±0.41	43.72±0.76	<u>86.50</u> ±0.59	89.67±0.26	36.65±0.06	28.29±0.13	28.87±0.88
DreamRender <sup>†</sup>	67.60±0.36	43.45±0.78	<b>88.80</b> ±0.44	<u>90.07</u> ±0.12	<u>37.29</u> ±0.03	<b>30.11</b> ±0.18	<u>24.91</u> ±0.65
<b>OverLayBench-Regular</b>							
GLIGEN	52.46±0.29	26.53±0.06	44.88±0.31	77.46±0.49	33.93±0.07	23.42±0.02	52.22±0.43
InstanceDiff	<b>60.08</b> ±0.24	<u>34.15</u> ±0.16	72.51±0.29	83.36±0.30	33.09±0.10	26.19±0.03	59.73±0.95
MIGC	47.42±0.08	20.06±0.18	56.67±0.75	77.85±0.59	32.72±0.11	24.99±0.02	54.24±0.52
HiCo	55.02±0.33	29.60±0.53	58.24±1.01	79.89±0.66	33.91±0.14	25.34±0.04	49.07±0.45
3DIS	55.66±0.22	27.29±0.18	<u>80.80</u> ±0.47	83.69±0.09	35.42±0.05	<u>28.12</u> ±0.01	48.56±0.48
CreatiLayout-SD3	47.04±0.13	20.67±0.28	62.60±0.77	78.31±0.38	36.67±0.05	25.55±0.08	45.57±0.20
CreatiLayout-FLUX	<u>59.72</u> ±0.29	<b>35.51</b> ±0.44	77.20±0.57	<b>86.39</b> ±0.31	<u>36.73</u> ±0.08	26.21±0.07	<b>41.51</b> ±0.39
EliGen	58.56±0.38	32.62±0.52	80.85±0.20	84.42±0.36	36.27±0.09	27.05±0.12	45.65±1.09
DreamRender <sup>†</sup>	58.08±0.36	33.00±0.50	<b>83.52</b> ±0.18	84.95±0.41	<b>36.85</b> ±0.11	<b>28.74</b> ±0.12	<u>42.66</u> ±0.56
<b>OverLayBench-Complex</b>							
GLIGEN	50.79±0.75	23.85±0.52	41.70±0.91	79.93±0.58	33.92±0.06	22.75±0.06	57.32±0.11
InstanceDiff	<u>53.68</u> ±0.56	25.63±0.34	66.02±0.47	80.34±0.25	32.33±0.05	25.53±0.01	66.32±0.29
MIGC	40.04±0.31	13.26±0.05	47.80±0.67	74.48±0.99	31.93±0.05	24.20±0.04	66.52±0.33
HiCo	46.56±0.31	20.35±0.38	48.88±0.32	75.19±0.48	33.15±0.18	24.41±0.05	55.78±0.35
3DIS	50.65±0.61	21.75±0.31	74.31±1.24	81.57±0.89	35.11±0.09	<u>27.35</u> ±0.07	54.90±0.29
CreatiLayout-SD3	44.24±0.55	18.05±0.39	52.10±0.53	79.98±0.30	36.55±0.08	24.76±0.03	53.29±0.80
CreatiLayout-FLUX	<b>54.50</b> ±0.50	<b>28.97</b> ±0.54	69.72±0.39	<b>86.45</b> ±0.45	<u>36.72</u> ±0.07	24.85±0.09	<b>45.66</b> ±0.75
EliGen	52.53±0.17	<u>26.19</u> ±0.27	74.03±0.66	84.09±0.58	36.18±0.11	25.92±0.13	50.41±0.74
DreamRender <sup>†</sup>	52.47±0.14	26.13±0.36	<b>77.87</b> ±0.45	<u>84.93</u> ±0.55	<b>36.75</b> ±0.10	<b>27.54</b> ±0.11	<u>48.11</u> ±0.89

Table 2: BaseModel v.s. Ours AM method on OverLayBench.

Method	Split	mIoU(%) ↑	O-mIoU(%) ↑	SR <sub>E</sub> (%) ↑	SR <sub>R</sub> (%) ↑	CLIP <sub>Global</sub> ↑	CLIP <sub>Local</sub> ↑	FID ↓
CreatiLayout	Simple	58.78	32.52	72.34	84.45	37.29	27.49	27.51
CreatiLayout-AM		61.16	37.69	73.33	84.84	37.17	27.44	27.76
vs. BaseModel		+4.05%	+15.90%	+1.37%	+0.46%	-0.32%	-0.18%	+0.91%
EliGen	Simple	68.17	43.72	86.50	89.67	36.65	28.29	28.87
EliGen-AM		69.70	46.43	86.83	90.07	36.84	28.58	26.43
vs. BaseModel		+2.24%	+6.20%	+0.38%	+0.45%	+0.52%	+1.03%	-8.45%
CreatiLayout	Regular	47.04	20.67	62.60	78.31	36.67	25.55	45.57
CreatiLayout-AM		47.38	21.79	63.13	78.71	36.49	25.46	46.34
vs. BaseModel		+0.72%	+5.42%	+0.85%	+0.51%	-0.49%	-0.35%	+1.68%
EliGen	Regular	58.56	32.62	80.85	84.42	36.27	27.05	45.65
EliGen-AM		59.44	33.85	81.18	85.47	36.46	27.37	43.52
vs. BaseModel		+1.50%	+3.74%	+0.41%	+2.43%	+0.52%	+1.18%	-4.67%
CreatiLayout	Complex	44.24	18.05	52.10	79.98	36.55	24.76	53.29
Ours		43.97	18.07	52.49	79.77	36.32	24.72	53.48
vs. BaseModel		-0.61%	+0.11%	+0.75%	-0.26%	-0.63%	-0.16%	+0.36%
EliGen	Complex	52.53	26.19	74.03	84.09	36.18	25.92	50.41
EliGen-AM		53.28	26.69	76.32	84.31	36.39	26.15	49.42
vs. BaseModel		+1.43%	+1.91%	+3.09%	+0.27%	+0.58%	+0.89%	-2.00%



## 6 Evaluation

### 6.1 Implementation Details

For data curation, we use Flux-1-dev for image generation with 28 sampling steps. Qwen2.5-VL-7B is employed for image captioning, while Qwen2.5-VL-32B is used to extract bounding bboxes, instance-level captions, and relationship captions. Additionally, we apply RealVisXL\_V5.0\_Lightning<sup>1</sup> for object removal during training data construction.

For CreatiLayout-AM training, we fine-tune the model for 3,500 steps on 8 NVIDIA RTX A6000 (48GB) GPUs with a batch size of 16, and a learning rate of  $10^{-5}$ . We use the AdamW [Loshchilov and Hutter, 2017] optimizer with bf16 precision. Please refer to the Appendix A for more details.

### 6.2 Quantitative Results

Table 1 presents the quantitative evaluation of multiple layout-to-image (L2I) generation methods [Li et al., 2023a, Wang et al., 2024a, Zhou et al., 2024a, Cheng et al., 2024, Zhang et al., 2024, Zhou et al., 2024b, Zhang et al., 2025, Zhou et al., 2025] across varying difficulty levels in the OverLayBench benchmark: Simple, Regular, and Complex.

As task difficulty increases, all models show a noticeable decline in spatial metrics (particularly O-mIoU), highlighting the inherent challenges posed by highly overlap and semantically similar layouts. Despite this, DiT-based models demonstrate more stable visual quality and stronger semantic alignment, underscoring their robustness and scalability in handling complex generation scenarios.

### 6.3 CreatiLayout-AM Comparison

Table 2 demonstrates that CreatiLayout-AM outperforms the original CreatiLayout on the **Simple** and **Regular** splits, with particularly notable gains in O-mIoU (+15.90% and +5.42%, respectively). These improvements are consistent across other spatial and relational metrics, including mIoU, SR<sub>E</sub>, and SR<sub>R</sub>, indicating enhanced spatial and relational alignment despite minor drops in CLIP scores. On the **Complex** split, where a distribution shift from the training set is more pronounced, performance remains competitive, exhibiting only slight declines in mIoU and CLIP. Overall, these results validate the effectiveness of amodal mask supervision in improving L2I generation under bbox overlap, presenting a promising direction for future explorations.

### 6.4 Qualitative Analysis

For qualitative analysis, we provide a comprehensive visualization of state-of-the-art L2I models on OverLayBench, showcasing a diverse set of examples with varying levels of layout overlap complexity in Figure 6. We annotate each row with gold, silver, and bronze icons representing the top-3 performing models.

## 7 Conclusion and Future Works

In this work, we present a comprehensive study on the often-overlooked challenge of object occlusion in Layout-to-Image (L2I) generation. We introduce **OverLayScore**, a principal difficulty metric that captures both spatial overlap and semantic similarity, and show that higher OverLayScore values strongly correlate with degraded generation quality. To support rigorous evaluation, we propose **OverLayBench**, a balanced benchmark spanning the full spectrum of layout difficulty. It features high-fidelity images and dense captions, enabling in-depth assessment of instance interactions in densely overlapping scenes. Additionally, we demonstrate that amodal mask supervision mitigates collusion artifacts, enhancing generation quality in complex layouts. Our baseline model, CreatiLayout-AM, outperforms existing methods under OverLayScore.

Together, our metric, benchmark, and baseline establish a unified testbed for advancing occlusion-aware, controllable image generation, and aim to inspire future methods with stronger spatial reasoning and compositional understanding.

---

<sup>1</sup>[https://huggingface.co/SG161222/RealVisXL\\_V5.0\\_Lightning](https://huggingface.co/SG161222/RealVisXL_V5.0_Lightning)



Figure 6: Comparison of generated images from different models on our OverLayBench.

**Acknowledgment.** This work is supported by NSF award IIS-2127544 and NSF award IIS-2433768. We thank Lambda, Inc. for their compute resource help on this project.

## References

- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL <https://arxiv.org/abs/2308.12966>.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. *arXiv preprint arXiv:2406.10210*, 2024.
- Anthony Chen, Jianjin Xu, Wenzhao Zheng, Gaole Dai, Yida Wang, Renrui Zhang, Haofan Wang, and Shanghang Zhang. Training-free regional prompting for diffusion transformers. *arXiv preprint arXiv:2411.02395*, 2024a.
- Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5343–5353, 2024b.
- Zeyuan Chen, Hongyi Xu, Guoxian Song, You Xie, Chenxu Zhang, Xin Chen, Chao Wang, Di Chang, and Linjie Luo. X-dancer: Expressive music to human dance video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.
- Zhennan Chen, Yajie Li, Haofan Wang, Zhibo Chen, Zhengkai Jiang, Jun Li, Qian Wang, Jian Yang, and Ying Tai. Region-aware text-to-image generation via hard binding and soft refinement. *arXiv preprint arXiv:2411.06558*, 2024c.
- Bo Cheng, Yuhang Ma, Liebucha Wu, Shanyuan Liu, Ao Ma, Xiaoyu Wu, Dawei Leng, and Yuhui Yin. Hico: Hierarchical controllable diffusion model for layout-to-image generation. *arXiv preprint arXiv:2410.14324*, 2024.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023a.
- Yutong Feng, Biao Gong, Di Chen, Yujun Shen, Yu Liu, and Jingren Zhou. Ranni: Taming text-to-image diffusion for accurate instruction following. *arXiv preprint arXiv:2311.17002*, 2023b.
- Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. Llm blueprint: Enabling text-to-image generation with complex and detailed prompts. *arXiv preprint arXiv:2310.10640*, 2023.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.
- Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.

- Yuseung Lee, Taehoon Yoon, and Minhyuk Sung. Groundit: Grounding diffusion transformers via noisy patch transplantation. *Advances in Neural Information Processing Systems*, 37:58610–58636, 2024.
- Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22511–22521, 2023a.
- Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation, 2023b. URL <https://arxiv.org/abs/2301.07093>.
- Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.
- Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7932–7942, 2024.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR, 2021.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL <https://arxiv.org/abs/2408.00714>.



- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- Divyansh Srivastava, Xiang Zhang, He Wen, Chenru Wen, and Zhuowen Tu. Lay-your-scene: Natural scene layout generation with diffusion transformers. *arXiv preprint arXiv:2505.04718*, 2025.
- Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6232–6242, 2024a.
- Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6232–6242, 2024b.
- Yilin Wang, Zeyuan Chen, Liangjun Zhong, Zheng Ding, and Zhuowen Tu. Dolfn: Diffusion layout transformers without autoencoder. In *European Conference on Computer Vision*, pages 326–343. Springer, 2024c.
- Yilin Wang, Haiyang Xu, Xiang Zhang, Zeyuan Chen, Zhizhou Sha, Zirui Wang, and Zhuowen Tu. Omnicontrolnet: Dual-stage integration for conditional image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7436–7448, 2024d.
- Zirui Wang, Zhizhou Sha, Zheng Ding, Yilin Wang, and Zhuowen Tu. Tokencompose: Text-to-image diffusion with token-level supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8553–8564, 2024e.
- Yinwei Wu, Xianpan Zhou, Bing Ma, Xuefeng Su, Kai Ma, and Xinchao Wang. Ifadapter: Instance feature control for grounded text-to-image generation. *arXiv preprint arXiv:2409.08240*, 2024.
- Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023a.
- Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023b.
- Haiyang Xu, Yu Lei, Zeyuan Chen, Xiang Zhang, Yue Zhao, Yilin Wang, and Zhuowen Tu. Bayesian diffusion models for 3d shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10628–10638, 2024.
- Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36, 2024.
- Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024a.
- Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*, 2024b.
- Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255, 2023.
- Guanning Zeng, Xiang Zhang, Zirui Wang, Haiyang Xu, Zeyuan Chen, Bingnan Li, and Zhuowen Tu. Yolo-count: Differentiable object counting for text-to-image generation. *arXiv preprint arXiv:2508.00728*, 2025.

- Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, and Yu Zhang. Eligen: Entity-level controlled image generation with regional attention. *arXiv preprint arXiv:2501.01097*, 2025.
- Hui Zhang, Dexiang Hong, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. *arXiv preprint arXiv:2412.03859*, 2024.
- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. Depr: Depth guided single-view scene reconstruction with instance-level diffusion. *arXiv preprint arXiv:2507.22825*, 2025.
- Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499, 2023.
- Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6818–6828, 2024a.
- Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis: Depth-driven decoupled instance synthesis for text-to-image generation. *arXiv preprint arXiv:2410.12669*, 2024b.
- Dewei Zhou, Mingwei Li, Zongxin Yang, and Yi Yang. Dreamrenderer: Taming multi-instance attribute control in large-scale text-to-image models. *arXiv preprint arXiv:2503.12885*, 2025.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

### 1. Claims

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

Answer: [Yes]

Justification:

Guidelines:

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

### 2. Limitations

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

Answer: [Yes]

Justification:

Guidelines:

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

### 3. Theory assumptions and proofs

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

Answer: [NA]

Justification:

Guidelines:

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

### 4. Experimental result reproducibility

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

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not include experiments.



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

## 5. Open access to data and code

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

Answer: [\[Yes\]](#)

Justification:

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification:

Guidelines:

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

## 7. Experiment statistical significance

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

Answer: [\[No\]](#)

Justification: Experiments are not evaluated for statistical significance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

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

Answer: [\[Yes\]](#)

Justification:

Guidelines:

- The answer NA means that the paper does not include experiments.

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

#### 9. Code of ethics

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

Answer: [Yes]

Justification:

Guidelines:

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

#### 10. Broader impacts

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

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

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

Answer: [NA]

Justification:

Guidelines:

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

## 12. Licenses for existing assets

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

Answer: [\[Yes\]](#)

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

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

Answer: [\[Yes\]](#)

Justification:

Guidelines:

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

## 14. Crowdsourcing and research with human subjects

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



Answer: [Yes]

Justification:

Guidelines:

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

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

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

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

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

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

# Appendix

## A Implementation Details

We begin by outlining the implementation details for the following stages: (1) data curation, (2) training, (3) inference and evaluation.

### A.1 Data Curation

#### A.1.1 Prompts Construction

We provide the prompt templates used with Qwen-2.5-VL-7B and Qwen-2.5-VL-32B during the dataset curation process. Qwen-2.5-VL is employed for three distinct tasks: (1) Image Captioning, (2) Grounding, and (3) Relationship Extraction. The specific prompts for each task are shown below.

##### Image Captioning

```
Give a detailed caption of this image.
```

##### Grounding

```
You are required to detect the main foreground instances in the image and describe them.
Response in json format:
```

```
{
  instance_category_1: {
    bbox: [x1, y1, x2, y2],
    local_prompts: description of this instance
  },
  instance_category_2: {
    bbox: [x1, y1, x2, y2],
    local_prompts: description of this instance
  },
  ...
}
```

```
Each bounding box must correspond to a single, distinct individual object - never a group
or collection. Do not merge multiple instances into one. Strictly follow this instruction
without exceptions or interpretation. Strictly follow the format in English, without any
irrelevant words.
```

##### Relationship Extraction

```
You are required to extract the relation between two instances in a given annotation,
based on their bounding boxes and local descriptions. Only describe the relationship for
the provided valid instance pairs.
Here is the image caption: {Caption}.
Here are the instance annotations: {Bbox and Instance-Caption}.
Here are the list of valid instance pairs to describe: {Valid Overlapping Pairs}.
Response in the following json format:
```

```
{
  (Instance_1, Instance_2): Instance_1 {relationship} Instance_2,
  ...
}
```

```
where {relationship} are words in the caption and local prompts that explicitly describe
the interaction or spatial relation. Each key must be a tuple from the list of valid
instance pairs. If the relationship is not explicitly specified, please respond with ‘
None’. Never invent any relationship that is not specified in the annotation. Strictly
respond the relationship description in one short sentence, without any irrelevant words.
```

#### A.1.2 Web-UI Construction

As part of our data curation pipeline, we incorporate human auditing to ensure data quality. To facilitate this process, we develop a custom Web-UI that displays key annotations, including image captions, bounding boxes, instance captions, and relationship captions. Annotators are instructed to verify the accuracy of each annotation to uphold high data quality standards. A screenshot of the Web-UI is provided in Figure 7.

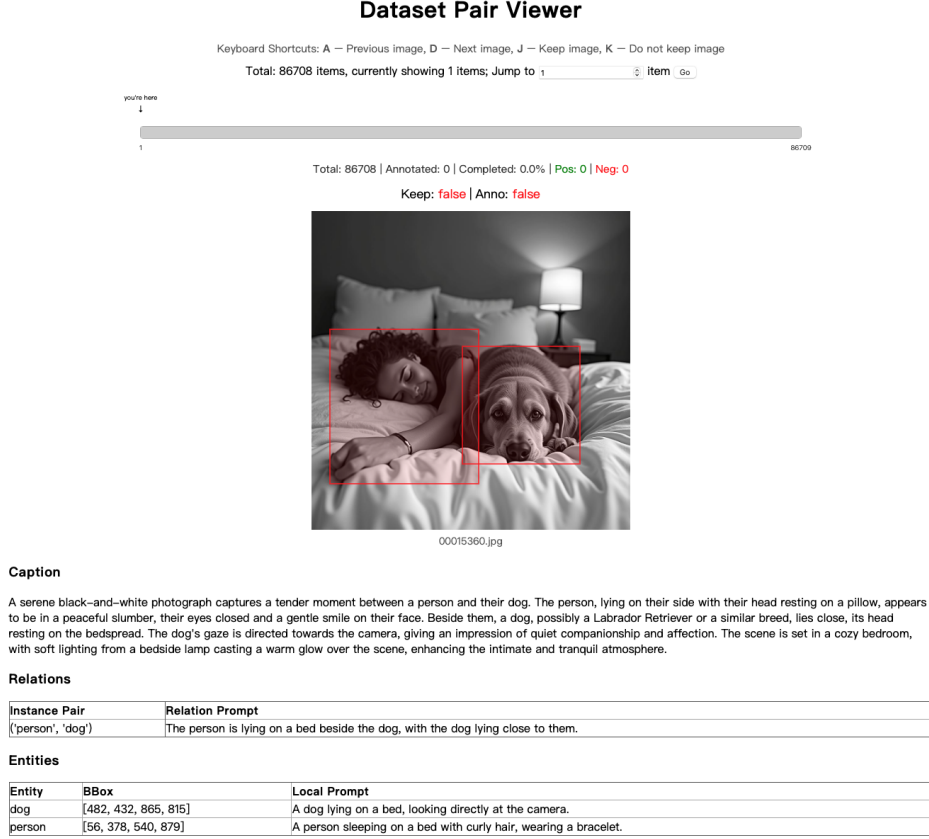


Figure 7: Web-UI used for human auditing.

## A.2 Training

The complete set of hyper-parameters used during training is listed in Table 3.

Table 3: Training hyperparameters of CreatiLayout-AM.

Hyperparameter	CreatiLayout-AM	EliGen-AM
Number of GPUs	8 Nvidia A6000 48GB	8 Nvidia H100 80GB
Batch Size (per GPU)	1	1
Gradient Accumulation Steps	2	1
Gradient Checkpointing	False	True
Learning Rate	1e-5	1e-4
LR Scheduler	Linear	Linear
Warm-up Steps	500	500
Training Steps	3500	2000
LoRA Rank	32	64
Parallel	DDP	FSDP
$\lambda$	0.5	1
$\beta$	1	1

### A.2.1 EliGen-AM

To further validate the effectiveness of our training set and the proposed AM method, we conduct experiments on an additional baseline, EliGen [Zhang et al., 2025]. Unlike CreatiLayout, which

represents each entity with explicit layout tokens—thereby allowing direct extraction of entity-specific attention maps for applying our losses in Equation (4), EliGen constrains image tokens within a bounding box to only attend to the global description tokens and their corresponding local description tokens. This design makes it non-trivial to define the instance-level attention map required in Equation (4). To address this, we approximate  $\mathcal{A}^i$  as the average attention map across all text tokens in the  $i^{th}$  local description.

Formally, we define the attention map of  $i^{th}$  instance as:

$$\bar{\mathcal{A}}^i = \frac{1}{L^i} \sum_{j=0}^{L^i} \mathcal{A}^j \quad (5)$$

where  $L^i$  is the number of tokens in the  $i^{th}$  local description,  $\mathcal{A}^j$  refers to the attention map between images tokens and the  $j^{th}$  local description token.

We replace the  $\mathcal{A}^i$  with  $\bar{\mathcal{A}}^i$  in Equation (2) and Equation (3) and train the model with the hyperparameters in Table 3. The quantitative results is shown in Table 2.

### A.3 Inference and Evaluation

For evaluation, we generate three images per method for each layout using a fixed random seed (20251202, 20251203 and 20251204) to ensure a fair comparison.

To compute the standard **mIoU**, we match each ground truth bounding box to its corresponding predicted box using the Hungarian algorithm.

For our proposed metric, **O-mIoU**, we compute the mIoU over the cropped intersection region between two instances involved in a specified relationship. We argue that this metric more effectively captures the fidelity and realism of object rendering, particularly in densely overlapping scenarios.

The CLIP score is computed using the pretrained CLIP model "ViT-B/32" [Radford et al., 2021].

During the inference and evaluation stages, Qwen-2.5-VL-32B is employed for **object detection** (to generate predicted bounding boxes for mIoU computation) and for **question answering**, which is used to determine the instance-level success rate  $\text{SR}_E$  and relationship-level success rate  $\text{SR}_R$ . We provide prompts for each task below.

#### Object Detection

You are required to detect all the instances of the following categories {Categories} in the image.  
Response in json format:

```
{
  category_1: [[x1, y1, x2, y2], [x1, y1, x2, y2], ...],
  category_2: [[x1, y1, x2, y2], ...]
  ...
}
```

For each category, provide a list of bounding boxes of all its instances in the image. Each bounding box must correspond to a single, distinct individual object - never a group or collection.  
Strictly follow this instruction without exceptions or interpretation.  
Strictly follow the format in English, without any irrelevant words.

#### $\text{SR}_E$

You are required to answer whether the instances in an image match the corresponding descriptions, based on their bounding boxes.  
Here are the instance names, the corresponding bboxes and the instance description: {Bbox and Instance Captions}  
Please follow these rules:  
Check if the generated instance visually matches its local\_prompt description.  
If the instance is clearly generated and not corrupted, and its key attributes described in the local\_prompt are present, answer Yes.  
If the instance is missing, corrupted, or the key attributes are not present, answer No.  
Response in the following format:

```
{
  Instance_name: Yes/No,
```

```
} ...
```

Each key must be from the dict of the instance name, the corresponding bbox and the instance description.  
Each value must be Yes or No.  
If the instance name is not in the image, the answer should be No.  
Strictly follow the format in English, without any irrelevant words.

## SR<sub>R</sub>

You are required to answer whether the relationship between two instances in an image matches the description.  
Here are the instance name and the bbox: {Bbox and Instance Name}.  
Here are dict of the instance pair and the ground truth relationship descriptions: {Relationships}.  
Please follow these rules:  
For proximity relations like near, beside, close to, next to, if the two instances are generated well (not corrupted or fused into one) and their bounding boxes are close, you can consider the description as matched.  
For directional or positional relations like behind, in front of, you must strictly check if the spatial arrangement in the image actually matches the description, because bounding boxes alone are not enough.  
Response in the following format:

```
{
  (Instance_1, Instance_2): Yes/No,
  ...
}
```

Each key must be a tuple from the dict of the instance pair and the ground truth relationship descriptions.  
Each value must be Yes or No.  
Yes means the action/spatial relationship between the two instances matches the description.  
You shouldn't pay too much attention on how well the bounding boxes are aligned.  
Strictly follow the format in English, without any irrelevant words.

## A.4 Baseline Clarification

We include several recent training-based L2I methods in our benchmark evaluation, some of which provide multiple variants depending on the underlying base model. To avoid ambiguity, we clarify our choices here. For HiCo [Cheng et al., 2024], we use the HiCo-SD1.5 model, as the SDXL version was not publicly available at the time of evaluation. For 3DIS [Zhou et al., 2024b] and DreamRender [Zhou et al., 2025], we adopt the FLUX-based versions. In addition, since DreamRender requires auxiliary modalities such as canny edge maps and depth maps, we utilize DepthAnything v2 [Yang et al., 2024a] to extract depth maps from reference images, which are then provided together with the layout as inputs to the model.

## B Quantitative Results Analysis on Training-Free Methods

In addition to the training-based approaches, we evaluate several training-free methods on OverLayBench [Xie et al., 2023a, Chen et al., 2024b, Phung et al., 2024, Lee et al., 2024, Chen et al., 2024a], as shown in Table 4.

Specifically, on the **OverLayBench-Simple** split, RegionalPrompting achieves the highest overall performance, with the best mIoU (42.54%), O-mIoU (20.10%), SR<sub>E</sub> (73.49%), SR<sub>R</sub> (75.81%), CLIP<sub>Local</sub> (27.40), and FID (23.94). On the **Regular** split, it continues to lead in mIoU (32.72%), O-mIoU (12.29%), SR<sub>E</sub> (63.74%), and CLIP<sub>Local</sub> (25.82), again achieving the lowest FID (43.13), despite a slight drop in SR<sub>R</sub> (67.08%). For the more challenging **Complex** split, RegionalPrompting still attains the highest mIoU (28.35%), O-mIoU (9.05%), SR<sub>E</sub> (53.56%), and CLIP<sub>Local</sub> (25.29), alongside a competitive FID (49.41), further demonstrating its robustness in dense and overlap-heavy scenarios.

Table 4: Comprehensive comparison between training-free methods on OverLayBench. **Bold and underline** denote the best and the second best methods. Methods above the dashed line are U-Net-based, while those below are DiT-based.

Method	mIoU(%) $\uparrow$	O-mIoU(%) $\uparrow$	SR <sub>E</sub> (%) $\uparrow$	SR <sub>R</sub> (%) $\uparrow$	CLIP <sub>Global</sub> $\uparrow$	CLIP <sub>Local</sub> $\uparrow$	FID $\downarrow$
<b>OverLayBench-Simple</b>							
BoxDiff	24.48 $\pm$ 0.08	7.71 $\pm$ 0.43	42.03 $\pm$ 0.89	69.94 $\pm$ 0.52	<b>36.78</b> $\pm$ 0.06	21.33 $\pm$ 0.08	<u>34.65</u> $\pm$ 0.20
LayoutGuidance	23.12 $\pm$ 0.27	7.92 $\pm$ 0.13	45.78 $\pm$ 0.65	70.83 $\pm$ 0.70	33.47 $\pm$ 0.10	21.22 $\pm$ 0.01	74.40 $\pm$ 1.01
R&B	27.78 $\pm$ 1.90	9.70 $\pm$ 1.16	36.98 $\pm$ 0.52	64.05 $\pm$ 0.45	34.64 $\pm$ 0.18	21.32 $\pm$ 0.04	36.57 $\pm$ 2.30
GroundDiT	<u>31.92</u> $\pm$ 0.34	<u>10.93</u> $\pm$ 0.36	<u>48.57</u> $\pm$ 0.89	<u>75.26</u> $\pm$ 0.40	<u>36.10</u> $\pm$ 0.12	<u>22.73</u> $\pm$ 0.03	34.98 $\pm$ 0.63
RegionalPrompting	<b>42.54</b> $\pm$ 0.17	<b>20.10</b> $\pm$ 0.27	<b>73.49</b> $\pm$ 0.53	<b>75.81</b> $\pm$ 0.32	35.45 $\pm$ 0.08	<b>27.40</b> $\pm$ 0.06	<b>23.94</b> $\pm$ 0.10
<b>OverLayBench-Regular</b>							
BoxDiff	19.40 $\pm$ 0.30	5.33 $\pm$ 0.24	37.58 $\pm$ 0.36	71.81 $\pm$ 0.44	<b>36.50</b> $\pm$ 0.05	19.97 $\pm$ 0.03	55.41 $\pm$ 0.28
LayoutGuidance	15.81 $\pm$ 0.20	4.51 $\pm$ 0.19	<u>44.94</u> $\pm$ 0.07	<u>72.76</u> $\pm$ 0.86	31.76 $\pm$ 0.01	20.05 $\pm$ 0.02	128.16 $\pm$ 0.35
R&B	20.35 $\pm$ 0.74	5.54 $\pm$ 0.61	32.85 $\pm$ 0.42	65.01 $\pm$ 0.06	34.49 $\pm$ 0.14	19.88 $\pm$ 0.04	58.55 $\pm$ 1.79
GroundDiT	<u>24.03</u> $\pm$ 0.20	<u>6.70</u> $\pm$ 0.31	42.24 $\pm$ 0.66	<b>74.02</b> $\pm$ 0.46	<u>36.00</u> $\pm$ 0.12	<u>21.10</u> $\pm$ 0.05	<u>54.94</u> $\pm$ 0.49
RegionalPrompting	<b>32.72</b> $\pm$ 0.30	<b>12.29</b> $\pm$ 0.23	<b>63.74</b> $\pm$ 0.73	67.08 $\pm$ 0.69	34.46 $\pm$ 0.10	<b>25.82</b> $\pm$ 0.10	<b>43.13</b> $\pm$ 0.31
<b>OverLayBench-Complex</b>							
BoxDiff	20.02 $\pm$ 0.32	5.21 $\pm$ 0.15	33.52 $\pm$ 0.85	<u>76.41</u> $\pm$ 0.51	<b>36.92</b> $\pm$ 0.02	19.91 $\pm$ 0.03	59.70 $\pm$ 0.67
LayoutGuidance	16.34 $\pm$ 0.33	4.01 $\pm$ 0.14	37.76 $\pm$ 0.91	75.53 $\pm$ 1.41	32.75 $\pm$ 0.04	19.72 $\pm$ 0.05	119.32 $\pm$ 1.19
R&B	19.80 $\pm$ 0.67	4.85 $\pm$ 0.36	28.38 $\pm$ 1.05	69.97 $\pm$ 1.31	34.57 $\pm$ 0.19	19.47 $\pm$ 0.06	63.71 $\pm$ 2.82
GroundDiT	<u>24.77</u> $\pm$ 0.88	<u>6.58</u> $\pm$ 0.31	<u>37.85</u> $\pm$ 1.27	<b>77.03</b> $\pm$ 1.81	<u>36.20</u> $\pm$ 0.24	<u>20.55</u> $\pm$ 0.06	<u>55.59</u> $\pm$ 1.39
RegionalPrompting	<b>28.35</b> $\pm$ 0.94	<b>9.05</b> $\pm$ 0.59	<b>53.56</b> $\pm$ 1.58	60.37 $\pm$ 1.56	33.34 $\pm$ 0.13	<b>25.29</b> $\pm$ 0.05	<b>49.41</b> $\pm$ 1.25

## C Error Pattern Analysis of Existing Methods

We identify and categorize the common failure patterns observed in existing methods into five major classes: 1) **Incorrect Object Number**, where models either hallucinate additional undesired objects or fail to generate required instances within the specified bounding box regions; 2) **Object Fusion**, where models struggle to generate distinct instances for overlapping bounding boxes, instead producing a single merged or entangled object; 3) **Object Distortion**, where the generated instance lacks realism, often exhibiting severe deformation or artifacts that degrade perceptual quality; 4) **Incorrect Category**, where the generated object does not match the intended category, undermining semantic correctness; 5) **BBox Misalignment**, where the object does not properly align with its designated bounding box, either overflowing beyond the box or failing to fully occupy the allocated region, thus breaking spatial consistency. Please refer to Figures 8 and 9 for additional visual examples and explanations of each error pattern.

## D Comparison Between GroundingDINO and Qwen

We compare the grounding capabilities of GroundingDINO v1.0 Liu et al. [2023] and Qwen-2.5-VL-32B Bai et al. [2023] using an example from GroundingDINO’s official demo<sup>2</sup>. As illustrated in Figure 10, GroundingDINO produces multiple false positives, including a misclassification of a shark as a butterfly. In contrast, Qwen demonstrates more accurate and robust detection performance.

## E User Study

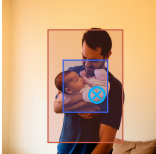
We conducted a user study with 15 participants over 60 image pairs (see Table 5), comparing CreatiLayout and CreatiLayout-AM across three difficulty levels. Excluding “No Preference” cases, our method achieved winning rates of 55.2%, 51.9%, and 46.8% on Simple, Regular, and Complex settings, respectively, showing moderate preference in simpler scenarios.

Table 5: User study results across three difficulty levels.

Ours vs. CreatiLayout	Simple	Regular	Complex
Winning Rate (%)	55.2	51.9	46.8

<sup>2</sup>[https://cloud.deepdataspace.com/playground/grounding\\_dino](https://cloud.deepdataspace.com/playground/grounding_dino)



	Image	Error Type & Explanation	Image	Error Type & Explanation
GLIGEN		<b>Object Fusion</b> The cat head and dog body are fused together.		<b>Object Distortion</b> The father's hand is distorted.
InstDiff		<b>BBox Misalignment</b> The cat only occupies the right part of the BBox.		<b>Incorrect Object Number</b> Two fathers
MIGC		<b>Object Fusion</b> The cat body and the dog body are fused together.		<b>BBox Misalignment</b> Baby's legs exceed the BBox.
HiCo		<b>Incorrect Object Number</b> Missing the dog.		<b>Incorrect Object Number</b> Additional Person
ELIGEN		<b>BBox Misalignment</b> The cat only occupies the right part of the BBox.		<b>BBox Misalignment</b> The baby occurs in the father's BBox and the father is generated in the wrong place.
3DIS		<b>Incorrect Category</b> Model generated a cat in the dog's BBox and a dog in the cat's BBox.		<b>Object Distortion</b> The hand is distorted.
CreatiLayout		<b>BBox Misalignment</b> The cat only occupies the right part of the BBox.		<b>Object Distortion</b> The baby's body is distorted.

**Image Caption:**  
A playful scene where a dog and a cat engage in a friendly tussle on a sunny lawn.

**Instance Captions:**  
(1) A playful dog with a joyful expression and wagging tail.  
(2) A nimble cat with alert eyes and a poised stance.

**Image Caption:**  
A heartwarming scene of a father holding his baby in a cozy room, with a gentle overlap where the father's arm supports the baby.

**Instance Captions:**  
(1) A caring father with a gentle smile, holding his baby tenderly.  
(2) A precious baby with joyful eyes and a warm expression.

⊗ Object Fusion   ⊗ BBox Misalignment   ⊗ Object Distortion   ⊗ Incorrect Object Number   ⊗ Incorrect Category

Figure 8: Error pattern analysis and explanation for existing models.

	Image	Error Type & Explanation	Image	Error Type & Explanation
3DIS		Incorrect Category ✓ Person ✗ Donut	MIGC	Incorrect Object Number ✓ One Woman ✗ Two Women
MIGC		Incorrect Category ✓ Person ✗ Elephant ✓ Person ✗ Elephant	ELIGEN	Incorrect Object Number Missing Child and Child. BBox Misalignment Child and Child do not align with their BBoxes.
CreatiLayout		Incorrect Category ✓ Giraffe ✗ Tree	3DIS	Incorrect Object Number ✓ One Woman ✗ Two Women
MIGC		Object Fusion Child and Dog are fused together.	HiCo	Object Distortion Skateboard is distorted.
ELIGEN		Object Fusion Sheep and Sheep are fused together.	GLIGEN	Object Distortion The Zebra is distorted. Incorrect Object Number ✓ One Zebra ✗ Four Zebra
3DIS		Incorrect Object Number ✓ One Girl ✗ Two Girls Object Fusion Man and Sofa are fused together.	CreatiLayout	Object Distortion The hand is distorted. Object Fusion Woman and Woman are fused together.

✗ Object Fusion ✗ BBox Misalignment ✗ Object Distortion ✗ Incorrect Object Number ✗ Incorrect Category

Figure 9: More examples of error patterns for existing methods.



Figure 10: Comparison of grounding performance between GroundingDINO v1.0 and Qwen-2.5-VL-32B.