FROM TOKENS TO CANVASES: RETHINKING DENSE LAYOUT-TO-IMAGE GENERATION IN REMOTE SENSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Layout-to-image generation controls image synthesis through object categories and bounding boxes, yet existing instance-centric conditioning becomes difficult to scale in dense scenes, particularly in remote-sensing imagery where numerous objects often coexist. Practical capacity limits force excess object conditions to be truncated, creating a condition–image mismatch when omitted objects remain visible in the training images, which later manifests as missing or spurious objects during inference. We revisit this representation by rasterizing all bounding boxes onto a fixed-resolution global canvas, avoiding object-wise growth in the spatial conditioning input. Surprisingly, this simple representation achieves comparable or even better layout consistency than richer edge- and shape-based conditions. Building on this finding, we propose DenseGen, a category-centric layered conditioning framework that decouples category-level semantic conditioning from instance-level spatial control. Category-Level Layout Mask Attention injects shared category semantics into corresponding spatial regions, while Instance-Aware Spatial Adaptive Layering separates overlapping boxes to reduce geometric ambiguity. Experiments on DIOR-R, DOTA-v1.0, and SODA-A demonstrate improved layout consistency and semantic accuracy in dense scenes, as well as greater utility of generated images for downstream object detection, without requiring additional structural annotations.

1 INTRODUCTION

Layout-to-image (L2I) generation enables controllable image synthesis from object categories and bounding boxes. Existing methods (Tang et al., 2025; Zhang et al., 2024; Wang et al., 2025) commonly represent objects as individual semantic–spatial tokens and inject them into the generative model through attention. Such instance-centric conditioning provides explicit object-level control, but its representation grows with the number of instances. This sensitivity to object count poses a critical challenge for remote-sensing image (RSI) generation, where a single image often contains tens or even hundreds of densely distributed objects, as illustrated in Fig. 1.

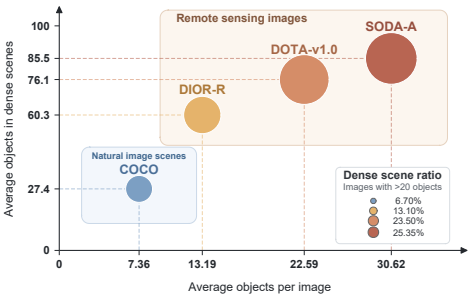

Figure 1: Object density comparison across natural and remote sensing scenes.

In practice, instance-centric methods often impose an upper limit on the number of input instances to bound memory and computation, as shown in Fig. 2(a). Objects exceeding this limit are omitted from the conditioning sequence, yet remain visible in the corresponding training images. The resulting condition–image mismatch can weaken the learned correspondence between layouts and images, contributing to missing or spurious objects at inference.

Increasing the instance capacity can alleviate truncation, but at the cost of greater computational and memory overhead. The more fundamental issue lies in the representation itself: layouts are inherently spatial structures, yet instance-centric methods serialize them into sequences whose length grows with the number of instances. This motivates a basic question: **must dense layouts be represented as long sequences of instance tokens?**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

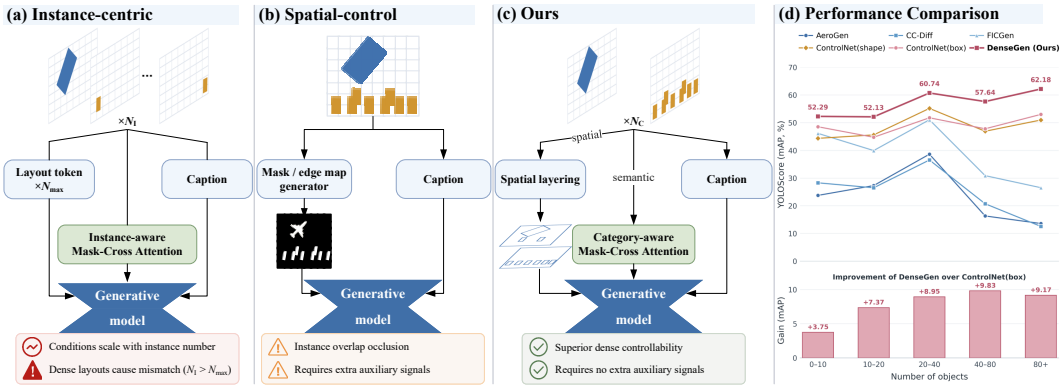

Figure 2: Comparison of mainstream L2I paradigms and performance across different object-density ranges on the DOTA-v1.0 dataset. N_I, N_C, and N_{max} denote the number of instances, the number of categories, and the maximum number of input instances, respectively.

We revisit layout representation from a spatial perspective by rasterizing all annotated bounding boxes onto a fixed-resolution global bounding-box canvas. This converts a variable-length instance sequence into a fixed-size spatial condition that directly encodes object locations, scales, and orientations in image coordinates. Every annotated box contributes to the spatial condition without being subject to an object-token budget, while the canvas dimensions remain unchanged as the layout becomes denser.

Existing spatial-guidance methods (Zhang et al., 2023; Ye et al., 2026) also use fixed-resolution conditions to guide generation, but commonly rely on signals such as edges or shape masks. These signals require additional annotations or preprocessing beyond the layout conditions. Moreover, this additional structure is not always beneficial for dense-layout control. Under the same ControlNet model (Zhang et al., 2023) and training protocol, a simple box canvas achieves comparable or even better controllability than richer edge- and shape-based conditions. This finding suggests that richer spatial information is not necessarily more effective for layout control, and that a simple task-aligned representation can be sufficient for dense layouts.

Despite its simplicity and effectiveness, a single box canvas does not explicitly bind categories to regions, and overlapping boxes can produce ambiguous geometric patterns. We address these two limitations with DenseGen, a category-centric layered conditioning framework. Category-Level Layout Mask Attention (C-LMA) shares semantic embeddings across instances of the same category and injects them into the corresponding layout regions, thereby reducing the complexity of semantic embeddings from the number of instances to the number of categories. Instance-Aware Spatial Adaptive Layering (I-SL) distributes overlapping boxes across spatial layers to reduce geometric ambiguity and better preserve instance-level spatial cues. Together, these components decouple category-level semantic conditioning from instance-level spatial control. We evaluate DenseGen on three representative remote-sensing datasets (Cheng et al., 2022; Xia et al., 2018; Cheng et al., 2023), which contain numerous densely distributed objects and provide challenging testbeds for layout-controlled generation. DenseGen achieves stronger layout consistency and semantic accuracy, while also improving the utility of generated images for downstream object detection.

Our main contributions are summarized as follows:

- We identify the object-wise condition growth and the mismatch between conditions and images caused by truncation as key bottlenecks in dense L2I generation, and revisit layout conditioning using a global box canvas with fixed spatial resolution.
- Through controlled comparisons, we show that a simple, task-aligned box canvas can match or outperform richer structural conditions in dense layouts.
- We propose DenseGen, which combines category-level semantic injection with instance-aware spatial layering to improve dense-layout control and downstream detection utility without additional structural annotations.

2 Related Work

2.1 Text-to-Image Generation

Text-to-image (T2I) generation synthesizes high-fidelity images from textual descriptions. Early GAN-based methods (Reed et al., 2016; Zhang et al., 2017; Lv et al., 2021) showed promising results but were limited by unstable training and scalability. Autoregressive models (Ding et al., 2021; Yu et al., 2022) improved generation quality through Transformer-based sequence modeling. More recently, diffusion models (Nichol et al., 2022; Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022) have become the dominant paradigm due to their strong fidelity, stability, and diversity. However, text prompts alone provide limited control over object categories and spatial locations.

2.2 Layout-to-Image Generation

Layout-to-image generation synthesizes images conditioned on object bounding boxes and category labels. For general image generation, existing methods (Li et al., 2023; Zheng et al., 2023; Wu et al., 2025) encode each object as a spatial–semantic condition and inject these conditions into the denoising network to improve layout controllability. Remote-sensing approaches extend this paradigm to complex aerial scenes: AeroGen (Tang et al., 2025) supports both horizontal and oriented bounding boxes, CC-Diff (Zhang et al., 2024) improves foreground–background contextual coherence, and FICGen (Wang et al., 2025) introduces frequency-inspired contextual disentanglement for degraded image synthesis. However, representing each object independently makes the conditioning length grow with the object count, limiting scalability in dense scenes containing a large number of objects.

2.3 Spatial Guidance for Controllable Generation

Another line of work improves controllability through spatial signals. ControlNet (Zhang et al., 2023) and T2I-Adapter (Mou et al., 2024) guide generation with edges, depth maps, or poses. LAMA (Li et al., 2021) predicts object masks from category embeddings and bounding-box dimensions, then refines overlapping masks to guide image synthesis. Remote-sensing methods such as CRS-Diff (Tang et al., 2024) and OF-Diff (Ye et al., 2026) further exploit shape masks or edges. Since these conditions are represented as global spatial maps, their complexity is less sensitive to object count. However, they typically require additional annotations or pre-processing and are often difficult to obtain in practice.

3 Method

3.1 Problem Formulation and Overview

Given a layout condition $\mathcal{L} = \{(b_i, c_i)\}_{i=1}^{N}$, where c_i is the category label and $b_i = \{(x_i^j, y_i^j)\}_{j=1}^{4}$ denotes the normalized corner coordinates of the i-th bounding box, our goal is to generate an image that follows all specified categories and spatial layouts. This representation supports both horizontal and oriented bounding boxes. Figure 3 provides an overview of DenseGen. DenseGen decouples spatial geometry from category semantics. We first rasterize all bounding boxes into a fixed-resolution global canvas and inject it through a spatial control branch. This basic design already forms a complete layout-conditioned generation model. Instance-Aware Spatial Adaptive Layering (I-SL) further reduces overlap-induced ambiguity, while Category-Level Layout Mask Attention (C-LMA) injects category semantics into their corresponding regions.

3.2 Global Bounding-Box Canvas Conditioning

Bounding-box layouts are inherently spatial, yet existing methods commonly convert them into variable-length object conditions. We instead retain their spatial form by rasterizing all bounding-box outlines into a fixed-resolution global canvas.

Let $\mathcal{P}(b_i)$ denote the polygon enclosed by b_i. The global bounding-box canvas is defined as

$$S^{\text{box}}(p) = \max_{i=1,\dots,N} \mathbb{I}[p \in \mathcal{R}(\partial P(b_i))], \tag{1}$$

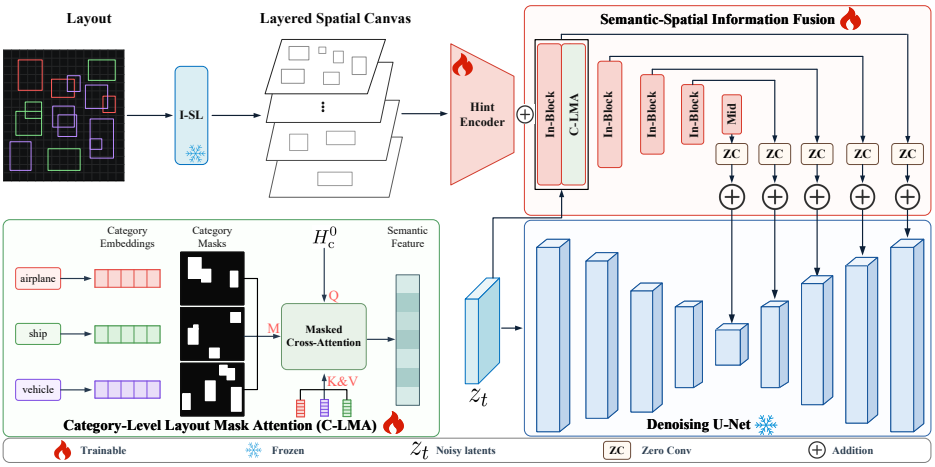

Figure 3: Given a dense layout, I-SL constructs a layered spatial canvas to reduce overlap-induced ambiguity, while C-LMA injects category semantics through masked cross-attention. Their fused features are propagated through the control branch.

where p denotes a spatial position and $\mathbb{I}[\cdot]$ is the indicator function, $\mathcal{R}(\partial P(b_i))$ denotes the rasterized outline of the i-th bounding box. The resulting canvas encodes the spatial extent of all input boxes while maintaining a fixed input size regardless of the number of objects.

A hint encoder transforms the canvas into a spatial guidance feature. Let $H_c^0 = \Phi_{\text{in}}^0(z_t, e_t, T)$ denote the initial feature of the trainable control branch, where z_t, e_t, and T are the noisy latent, timestep embedding, and global text condition, respectively. The initial controlled feature is

$$G_{\text{hint}} = E_{\text{hint}}(S^{\text{box}}, e_t, T), \qquad F_c^0 = H_c^0 + G_{\text{hint}}. \tag{2}$$

The resulting features are propagated through the control branch and injected into the frozen denoising U-Net via zero-initialized convolutions at corresponding layers.

This basic model already provides effective spatial control using only bounding boxes. However, overlapping boxes may merge on a single canvas, and the binary canvas does not explicitly encode category semantics. We address these two limitations in the following sections.

3.3 Instance-Aware Spatial Adaptive Layering

To reduce geometric ambiguity caused by overlapping boxes, I-SL distributes intersecting instances across spatial layers. Instances are sorted by bounding-box area in descending order before layer assignment. The overlap relation between two instances is defined as

$$\Omega(i,j) = \mathbb{I}\left[\text{Area}\left(\mathcal{P}(b_i) \cap \mathcal{P}(b_j)\right) > 0\right]. \tag{3}$$

Following this order, each box is assigned to the first available layer containing no overlapping instance:

$$\ell_i = \begin{cases} \min \mathcal{A}_i, & \mathcal{A}_i \neq \emptyset, \\ L_{\max}, & \mathcal{A}_i = \emptyset, \end{cases} \tag{4}$$

$$\mathcal{A}_i = \{l \in \{1, \ldots, L_{\max}\} \mid \Omega(i,j) = 0,\ \forall j \in \mathcal{G}_l\}, \tag{5}$$

where $\mathcal{G}_l$ denotes the instances assigned to the l-th layer. When no non-overlapping layer is available, the instance is assigned to the last layer. Bounding-box outlines are rasterized independently in each layer, with unused layers zero-padded. The resulting maps are stacked as

$$S_l(p) = \max_{i:\ell_i=l} \mathbb{I}[p \in \mathcal{R}(\partial P(b_i))], \quad S = \text{Stack}(S_1, \ldots, S_{L_{\max}}). \tag{6}$$

I-SL replaces S^{box} in Sec. 3.2 with $\mathbf{S}$:

$$G_{\text{hint}} = E_{\text{hint}}(\mathbf{S}, e_t, T). \tag{7}$$

3.4 CATEGORY-LEVEL LAYOUT MASK ATTENTION

The layered canvas encodes instance geometry but does not explicitly associate spatial regions with object categories. Rather than introducing one semantic condition per instance, C-LMA shares semantics among objects of the same category.

Let $\mathcal{C}(L) = \{u_k\}_{k=1}^{K}$ denote the categories appearing in the layout. For each category, we extract its token-level semantic features and construct the union of its bounding-box regions:

$$e_k = E_{\text{text}}(u_k), \quad M_k(p) = \max_{i:c_i=u_k} \mathbb{I}[p \in P(b_i)], \tag{8}$$

where E_{text} is a frozen CLIP text encoder (Radford et al., 2021), and $e_k \in \mathbb{R}^{L\times d_t}$ denotes the complete text-token sequence.

Using H_c^0 as the query feature, C-LMA independently performs masked cross-attention for each category and aggregates the resulting features:

$$A_{\text{cat}}(H_c^0, E, M) = \text{Reshape}\left[\sum_{k=1}^{K} \widetilde{M_k} \odot \text{Softmax}\left(\frac{QK_k^\top}{\sqrt{d_k}}\right) V_k\right], \tag{9}$$

where $E = \{e_k\}_{k=1}^{K}$ and $M = \{M_k\}_{k=1}^{K}$. Q is projected from H_c^0, while K_k and V_k are projected from e_k, and d_k denotes the key dimension. The Softmax is applied over the text-token dimension of each category, and $\widetilde{M_k}$ denotes the spatial mask resized and flattened to match the query resolution.

C-LMA replaces the original feature H_c^0 in Sec. 3.2. The initial feature of the complete model is therefore

$$F_c^0 = A_{\text{cat}}(H_c^0, E, M) + G_{\text{hint}}. \tag{10}$$

Subsequent control propagation follows Sec. 3.2. Since each category is represented by one shared text-token sequence, semantic conditioning scales with the number of categories rather than the number of instances.

3.5 TRAINING OBJECTIVE

Given a training image x, its latent representation $z_0 = E_{\text{VAE}}(x)$ is perturbed at timestep t as

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{11}$$

DenseGen predicts the added noise conditioned on the global text description T, category embeddings E, category masks M, and layered spatial canvas $\mathbf{S}$:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x,t,\epsilon}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, T, E, M, \mathbf{S}\right)\|_2^2\right]. \tag{12}$$

4 EXPERIMENTS

4.1 DATASETS

We evaluate DenseGen on DIOR-R (Cheng et al., 2022), DOTA-v1.0 (Xia et al., 2018), and SODA-A (Cheng et al., 2023), all of which contain remote-sensing or aerial imagery with diverse object scales, orientations, and dense distributions. Their high density and scale variation also result in frequent object overlaps, providing challenging benchmarks for layout-conditioned generation.

4.2 EVALUATION METRICS

We evaluate image quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) for distributional similarity and Category Accuracy Score (CAS) for semantic accuracy using a pretrained classifier, following OF-Diff (Ye et al., 2026). Layout consistency is measured by YOLOScore (Li et al., 2021), using detections from a pretrained YOLOv11-OBB model (Khanam & Hussain, 2024) to report mAP, mAP_{50}, and mAP_{75}. Following the official data splits, we further evaluate downstream trainability under two settings: synthetic-only, training detectors on images generated from test-set layouts and evaluating on real training images; and real + synthetic, augmenting real training images with images generated from validation-set layouts and evaluating on real test images. Detection performance is measured using mAP_{50}.

Table 1: Image quality and layout consistency. YOLOScore reports mAP/mAP$_{50}$/mAP$_{75}$. Bold and underline indicate the best and second-best results for each metric.

Method	DIOR-R			DOTA-v1.0			SODA-A		
	FID↓	CAS↑	YOLOScore↑	FID↓	CAS↑	YOLOScore↑	FID↓	CAS↑	YOLOScore↑
Instance-Centric Methods									
LayoutDiffusion	43.11	53.65	7.85/10.73/8.34	<u>42.31</u>	54.24	16.26/31.45/17.39	52.86	43.72	9.14/26.43/5.62
GLIGEN	41.83	58.94	8.37/10.73/8.95	57.22	52.54	15.28/28.04/15.83	53.98	44.22	8.83/25.38/5.20
AeroGen	32.11	59.32	36.72/52.47/39.13	50.63	58.47	24.45/38.47/25.10	<u>48.02</u>	46.37	14.86/31.69/12.51
CC-Diff	52.45	67.83	44.59/61.30/48.58	64.25	70.17	24.60/39.81/26.32	61.41	42.88	11.13/28.83/6.70
FICGen	52.80	69.69	53.02/<u>68.02</u>/**56.96**	55.64	**78.52**	39.69/57.95/42.58	53.27	48.82	30.77/59.98/29.16
Spatial-Guidance Methods									
OF-Diff	**27.09**	57.61	44.79/59.48/47.61	72.26	56.43	37.61/51.14/41.87	49.28	51.39	37.16/58.23/41.14
ControlNet (shape)	47.75	**76.94**	48.36/65.65/51.44	54.18	70.11	<u>46.88</u>/65.96/<u>51.41</u>	49.34	<u>52.05</u>	37.86/59.02/42.90
ControlNet (edge)	59.33	64.47	41.86/55.37/43.94	54.33	72.81	45.59/63.02/49.39	49.56	51.49	36.71/57.68/40.41
ControlNet (box)	51.18	70.01	<u>53.12</u>/67.57/56.46	49.11	71.60	45.33/<u>68.10</u>/48.43	56.11	50.13	<u>41.09</u>/<u>67.65</u>/<u>51.08</u>
DenseGen	<u>28.57</u>	<u>73.24</u>	**53.64/69.71**/<u>56.68</u>	**40.37**	<u>75.75</u>	**56.05/72.73/63.35**	**45.03**	**52.66**	**49.37/73.43/56.81**

4.3 Implementation Details

Data Preparation. We follow the official data splits, training each generative model only on its corresponding training set. DIOR-R images are directly resized to 512×512. DOTA-v1.0 images are cropped into 1024×1024 patches, while SODA-A images are cropped into 512×512 patches. All cropped patches are then resized to 512×512. Images without any objects are removed. For spatial-guidance methods, object-shape masks are extracted using RemoteSAM (Yao et al., 2025), and object edges using SketchKeras (lllyasviel, 2017) on foreground-only images.

Training and Inference. DenseGen is initialized from Stable Diffusion v1.5. During fine-tuning, we update the ControlNet branch and C-LMA module while keeping the remaining components frozen. Training is performed on two NVIDIA L20 GPUs using AdamW with a learning rate of 1×10^{-5} and a batch size of 8 per GPU. All generative models are trained for 110 epochs. To mitigate condition truncation in dense scenes, we increase the object limits of all instance-centric methods to 20 during training and 60 during inference, balancing layout coverage with GPU memory and time costs. We set $L_{\max} = 3$, and zero-pad unused channels.

4.4 Quantitative Comparison

Layout Control and Image Quality. Table 1 shows that DenseGen achieves the highest layout mAP and mAP$_{50}$ on all three datasets, indicating stronger agreement with the input layouts. It also produces high-quality images with accurate object semantics, ranking among the top two on FID and CAS. Its advantages in layout consistency are particularly pronounced on the denser datasets, with larger gains over ControlNet (box) and wider margins over instance-centric baselines. These results highlight the benefits of preserving dense box layouts and explicitly associating category semantics with spatial regions for controllable remote sensing image generation.

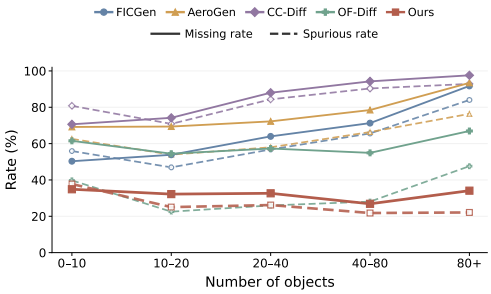

Figure 4: Object error rates by layout density on DOTA-v1.0: missing (solid) and spurious (dashed).

Robustness to Layout Density. Figure 4 examines missing and spurious object rates across different layout densities on DOTA-v1.0. DenseGen maintains relatively low error rates across all density ranges, even for layouts containing more than 80 objects. In contrast, instance-centric baselines exhibit increasingly severe object omissions and spurious generations as layout density increases. These results are consistent with our analysis that truncation-induced condition–image mismatch contributes to both missing and spurious objects.

Table 3: Downstream detection with synthetic-only or real + synthetic training (mAP_{50}). Ranking is among generation methods; Δ compares DenseGen with the baseline for each protocol.

Method	Synthetic-only			Real + Synthetic		
	DIOR-R	DOTA-v1.0	SODA-A	DIOR-R	DOTA-v1.0	SODA-A
Baseline	89.56	70.45	84.71	77.23	80.17	84.68
AeroGen	79.86	43.10	61.10	76.14	80.29	84.62
CC-Diff	60.62	32.81	34.84	77.92	80.08	84.66
FICGen	60.78	43.05	58.37	78.37	80.91	83.23
OF-Diff	59.82	62.80	68.58	78.92	81.95	85.85
ControlNet (shape)	75.71	57.06	63.36	**79.27**	81.73	85.63
ControlNet (edge)	74.18	57.83	65.61	79.11	81.65	85.28
ControlNet (box)	72.31	57.80	69.36	78.02	81.06	85.99
DenseGen	**84.29**	**69.62**	**82.24**	79.20	**82.36**	**87.42**
Δ vs. baseline	−5.27	−0.83	−2.47	+1.97	+2.19	+2.74

Effect of Training-Time Truncation. To examine the impact of condition truncation, we select training and test images containing 5 to 20 objects from DOTA-v1.0. AeroGen, CC-Diff, and FICGen are trained with instance limits of 10 or 20 and evaluated on identical, untruncated test layouts. As shown in Table 2, increasing the training limit consistently reduces missing and spurious object rates while improving YOLOScore. These results support our hypothesis that truncation-induced condition–image mismatch degrades layout controllability, highlighting the importance of preserving complete layout conditions during training.

Table 2: Effect of training-time condition truncation on DOTA-v1.0.

Method	Train Limit	Missing↓	Spurious↓	YOLOScore↑
AeroGen	10	69.76%	62.14%	17.66/31.12/17.37
	20	**68.19%**	**60.23%**	**21.16/33.54/21.75**
CC-Diff	10	82.97%	81.51%	16.75/28.40/18.28
	20	**78.23%**	**72.47%**	**20.05/33.87/22.60**
FICGen	10	59.08%	50.09%	30.62/45.82/31.81
	20	**56.77%**	**48.21%**	**32.61/47.15/35.06**

Generated-Data Trainability. Table 3 compares the utility of images generated by different methods for downstream object detection under the same protocol (detailed in the Appendix). Under the synthetic-only setting, DenseGen achieves the highest mAP_50 across all three datasets, approaching the performance of detectors trained on real images (baseline). Under the real + synthetic setting, real training images are augmented with synthetic data. DenseGen improves the real-only baseline mAP_50 by 1.97, 2.19, and 2.74 points on DIOR-R, DOTA-v1.0, and SODA-A, respectively, achieving the largest gains on the latter two datasets.

Computational Scalability. For this scalability benchmark, instance limits are adjusted to the evaluated object counts (20, 40, and 80), rather than the fixed limits used in the main experiments. Figure 5 shows that DenseGen maintains nearly constant computational costs during both training

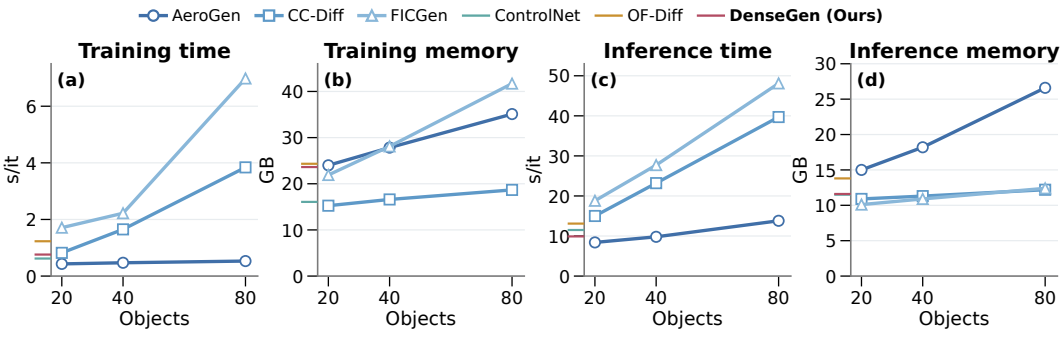

Figure 5: Training/inference time and peak GPU memory usage across varying object counts (batch size: 1). Colored axis ticks indicate the nearly constant resource costs of spatial-guidance methods.



Figure 6: Qualitative comparison on DIOR-R and DOTA-v1.0. DenseGen more closely follows the specified object groups, orientations, and spatial arrangements in the selected scenes.

and inference across the evaluated object-density ranges, with resource efficiency competitive with other spatial-guidance methods. In contrast, instance-centric methods exhibit substantial growth in time or memory as object count increases, making it impractical to address condition truncation by indefinitely increasing their instance limits during training and inference. The instance limits of 20 during training and 60 during inference adopted for instance-centric methods in our experimental setup already approach the practical memory and runtime limits of our available hardware.

4.5 Qualitative Evaluation

Comparative Results. Figure 6 compares images generated by different methods under the same layouts. Instance-centric methods often produce low-quality objects and exhibit substantial discrepancies between the specified layouts and generated objects, even when the number of input objects is within their inference limits. Spatial-guidance methods largely preserve the spatial arrangement, but often produce objects with indistinct boundaries and degraded image quality. In contrast, DenseGen more closely follows the specified layouts while maintaining high image quality and recognizable object structures, even in densely packed scenes with substantial box overlap.

Extreme-Density Generation. Figure 7 demonstrates DenseGen's ability to generate scenes containing 84 to 486 objects, far exceeding the instance limits of instance-centric baselines. Despite substantial box overlap in the first, second, and fourth examples, DenseGen generates recognizable objects while closely following the specified layouts.

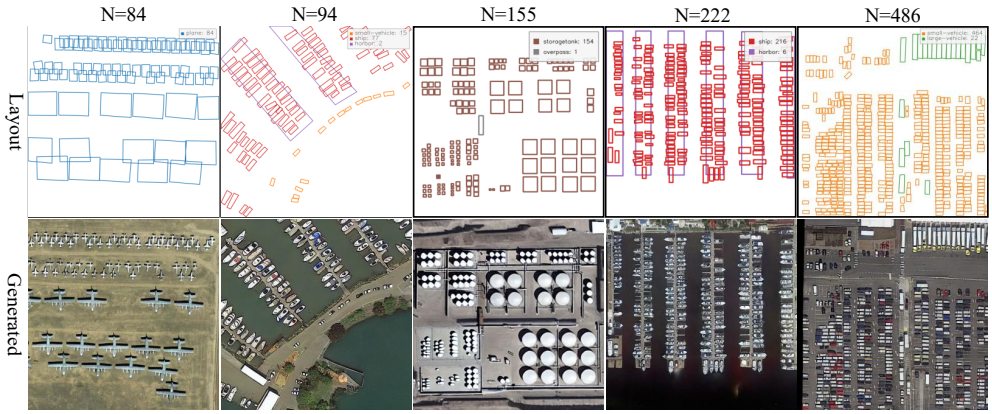

Figure 7: DenseGen generation from layouts with up to 486 input boxes, including scenes with extensive box overlap.

Table 4: Component ablations and the effect of the value of $L_{\max}$. G_{hint} denotes spatial guidance from the single bounding-box canvas. FSI (Fully Separated Images) measures the percentage of images with all instances separated into non-overlapping layers.

Method	C-LMA	G_{hint}	I-SL	DOTA-v1.0				SODA-A			
				FSI↑	CAS↑	YOLOScore↑	Trainability↑	FSI↑	CAS↑	YOLOScore↑	Trainability↑
Semantic-only	✓			–	64.21	20.86/36.08/20.82	50.43/80.64	–	46.96	13.47/29.14/10.33	38.27/82.84
Spatial-only		✓		49.21	71.60	45.33/68.10/48.43	57.80/81.06	59.03	50.13	41.09/67.65/51.08	69.36/85.99
DenseGen (w/o I-SL)	✓	✓		49.21	75.04	54.91/71.67/62.26	67.85/81.93	59.03	52.18	48.16/72.12/55.42	79.22/86.95
DenseGen ($L_{\max}=3$)	✓	✓	✓	87.68	**75.75**	56.05/72.73/63.35	**69.62/82.36**	94.56	**52.66**	**49.37/73.43/56.81**	**82.24/87.42**
DenseGen ($L_{\max}=4$)	✓	✓	✓	93.71	74.82	55.92/72.46/63.08	68.84/81.92	98.01	51.08	48.74/72.66/56.33	81.25/86.67
DenseGen ($L_{\max}=5$)	✓	✓	✓	**97.33**	75.64	**56.17/73.08/63.54**	69.55/82.19	**99.28**	51.95	49.05/73.17/56.53	81.94/87.13

4.6 Ablation Study

Table 4 evaluates the contributions of each component on DOTA-v1.0 and SODA-A. Semantic-only conditioning specifies object categories within designated regions but lacks explicit control over object count and individual positions. Spatial-only conditioning achieves good layout consistency but lacks explicit category semantics, often resulting in simplistic and uniform object appearances and difficulty handling overlapping boxes. Adding C-LMA to the box canvas complements spatial guidance with category semantics, while incorporating I-SL further improves performance by reducing overlap-induced ambiguity. Figure 8 further illustrates these findings.

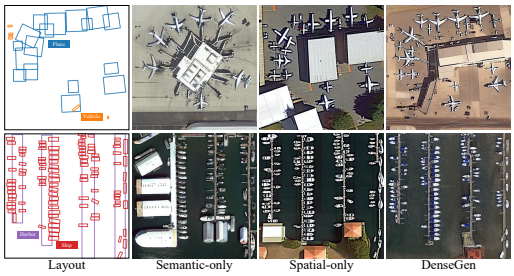

Figure 8: Conditioning ablations for aircraft and harbor scenes.

Increasing $L_{\max}$ improves instance separation but does not consistently improve generation performance, partly due to reduced compatibility with pretrained ControlNet weights. Based on these results, we adopt $L_{max}=3$ to balance instance separation and generation performance.

5 Conclusion

We revisit dense layout-to-image generation in remote sensing by replacing instance-wise conditions with a fixed-resolution box canvas to avoid condition truncation. Building on this representation, DenseGen combines category-level semantic conditioning with spatial layering to improve dense-layout control. Experiments on DIOR-R, DOTA-v1.0, and SODA-A demonstrate improved layout consistency, semantic accuracy, and downstream detection utility, while maintaining favorable computational scalability without additional structural annotations.

AI USE STATEMENT

Generative AI tools were used for language editing and LaTeX formatting. All AI-assisted content was reviewed by the authors, who take responsibility for the final manuscript.

REFERENCES

Gong Cheng, Jiabao Wang, Ke Li, Xingxing Xie, Chunbo Lang, Yanqing Yao, and Junwei Han. Anchor-free oriented proposal generator for object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19822–19835, 2021.

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.

Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 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*, pp. 22511–22521, 2023.

Zejian Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. Image synthesis from layout with locality-aware mask adaption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13819–13828, 2021.

lllyasviel. SketchKeras. `https://github.com/lllyasviel/sketchKeras`, 2017.

Ning Lv, Hongxiang Ma, Chen Chen, Qingqi Pei, Yang Zhou, Fenglin Xiao, and Ji Li. Remote sensing data augmentation through adversarial training. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:9318–9333, 2021.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 16784–16804, 2022.

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*, pp. 8748–8763, 2021.

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

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pp. 1060–1069. PMLR, 2016.

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*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.

Datao Tang, Xiangyong Cao, Xingsong Hou, Zhongyuan Jiang, Junmin Liu, and Deyu Meng. Crs-diff: Controllable remote sensing image generation with diffusion model. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Datao Tang, Xiangyong Cao, Xuan Wu, Jialin Li, Jing Yao, Xueru Bai, Dongsheng Jiang, Yin Li, and Deyu Meng. Aerogen: Enhancing remote sensing object detection with diffusion-driven data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3614–3624, June 2025.

Wenzhuang Wang, Yifan Zhao, Mingcan Ma, Ming Liu, Zhonglin Jiang, Yong Chen, and Jia Li. Ficgen: Frequency-inspired contextual disentanglement for layout-driven degraded image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19097–19107, 2025.

Yinwei Wu, Xianpan Zhou, Bing Ma, Xuefeng Su, Kai Ma, and Xinchao Wang. IFAdapter: Instance feature control for grounded text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15949–15959, October 2025.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

Liang Yao, Fan Liu, Delong Chen, Chuanyi Zhang, Yijun Wang, Ziyun Chen, Wei Xu, Shimin Di, and Yuhui Zheng. Remotesam: Towards segment anything for earth observation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3027–3036, 2025.

Ziqi Ye, Shuran Ma, Jie Yang, Xiaoyi Yang, Yi Yang, Ziyang Gong, Xue Yang, and Haipeng Wang. Object fidelity diffusion for remote sensing image generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 3258–3278, 2026.

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

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

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*, pp. 3836–3847, 2023.

Mu Zhang, Yunfan Liu, Yue Liu, Yuzhong Zhao, and Qixiang Ye. Cc-diff: enhancing contextual coherence in remote sensing image synthesis. *arXiv preprint arXiv:2412.08464*, 2024.

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*, pp. 22490–22499, 2023.

A Implementation Details

A.1 Dataset Preprocessing

We follow the official data splits for all three datasets. DIOR-R images are originally 800×800 and are directly resized to 512×512. For DOTA-v1.0, we adopt sliding-window cropping with a window size of 1024, a gap of 200, and a scale rate of 1.0. For SODA-A, we use a window size of 512 and a gap of 100, with the same scale rate. Both datasets use an image coverage threshold of 0.6, an object-window IoF threshold of 0.7, and boundary padding with a value of [104, 116, 124]. All cropped patches are resized to 512×512, and patches without annotated objects are discarded.

A.2 Generation Settings

For image generation, we employ the DDIM sampler with 50 sampling steps and $\eta = 0.2$. The classifier-free guidance scale is set to 7.5, and all images are generated at a resolution of 512×512. We use a batch size of 1 and generate one sample per layout. During sampling, the unconditional branch uses an empty text prompt while retaining the same spatial and category conditions as the conditional branch.

A.3 Text Conditioning

For each layout, we construct a global text prompt from its unique object categories. The prompt begins with "an aerial image with", followed by category-specific descriptions joined using commas and "and". Repeated categories are included only once. During training, the global text prompt is randomly replaced with an empty string according to the configured condition dropout probability.

A.4 Downstream Detection Protocol

Unless otherwise specified, the training, validation, and test splits in this section refer to the official object detection dataset partitions, rather than splits constructed for generative model evaluation. All generative methods are trained on the official real training splits. For downstream detector training, each synthetic image is paired with the layout used to generate it. The same layout source and evaluation protocol are used for all compared generative methods.

DIOR-R and SODA-A. We use separate training, validation, and test splits. In the synthetic-only setting, detectors are trained on images generated from test-set layouts and evaluated on real training images. The corresponding real-image Baseline is trained on real test images and their annotations, and evaluated on the same real training images. In the real + synthetic setting, detectors are trained on real training images augmented with synthetic images generated from validation-set layouts, and evaluated on real test images. Its Baseline is trained only on real training images and evaluated on the same real test images.

DOTA-v1.0. Because annotations for the official test split are not publicly available, we use the official validation split as the local test split for downstream detection evaluation. In the synthetic-only setting, detectors are trained on images generated from perturbed validation-set layouts and evaluated on real training images. The corresponding real-image Baseline is trained on real validation images and evaluated on the same real training images. In the real + synthetic setting, detectors are trained on real training images augmented with synthetic images generated from perturbed validation-set layouts and evaluated on real validation images. Its Baseline uses only real training images.

To mitigate potential label leakage caused by reusing validation-set layouts, we randomly remove 10% of objects and apply horizontal flipping before generation. The perturbed layouts serve as annotations for the corresponding synthetic images. This perturbation reduces direct correspondence with the original validation annotations but does not completely eliminate information overlap. All generative methods follow the same protocol.

Detector Implementation. We adopt YOLO11L-OBB as the downstream detector for all datasets. Under the synthetic-only setting, detectors are trained from scratch without pretrained weights.

Algorithm 1 Instance-Aware Spatial Adaptive Layering

Require: Bounding boxes $\mathcal{B} = \{b_i\}_{i=1}^{N}$, maximum layers $L_{\max}$
Ensure: Layered spatial canvas S
Sort instances by descending bounding-box area
Initialize $G_l \leftarrow \emptyset$ and $S_l \leftarrow 0$ for all layers
for each instance i in sorted order **do**
  $A_i \leftarrow \{l \mid \Omega(i, j) = 0,\ \forall j \in G_l\}$
  if $A_i \neq \emptyset$ **then**
    $\ell_i \leftarrow \min A_i$
  else
    $\ell_i \leftarrow L_{\max}$
  end if
  $G_{\ell_i} \leftarrow G_{\ell_i} \cup \{i\}$
  Rasterize the outline of b_i onto S_{ℓ_i}
end for
$S \leftarrow \mathrm{Stack}(S_1, \ldots, S_{L_{\max}})$
Normalize S to $[0, 1]$
return S

Under the real + synthetic setting, detectors are initialized with pretrained YOLO11L weights (`yolo11l.pt`). All detectors are trained for 200 epochs with an image resolution of 512×512 and a batch size of 32. During evaluation, we use a batch size of 64, a confidence threshold of 0.25, and an NMS IoU threshold of 0.6. Detection performance is reported using mAP, mAP_{50}, mAP_{75}.

A.5 I-SL Implementation

I-SL distributes overlapping instances across a fixed number of spatial layers. We set $L_{\max} = 3$ in all experiments. Instances are processed in descending order of bounding-box area. Following this order, each instance is greedily assigned to the first available layer containing no overlapping instance. If no such layer exists, the instance is assigned to the last layer, where overlap is permitted. The complete procedure is summarized in Algorithm 1.

After layer assignment, the original bounding-box outlines are rasterized into their corresponding layers with a line thickness of two pixels. The resulting maps are stacked into a three-channel spatial canvas of size $512 \times 512 \times 3$ and normalized to $[0, 1]$. Unused layers remain zero-filled.

A.6 Evaluation Metrics

Fréchet Inception Distance (FID). FID measures the distributional similarity between generated and real images. We generate one image per evaluation layout and compute FID against the corresponding real images. The evaluation sets contain 11,738 images for DIOR-R, 3,066 for DOTA-v1.0, and 4,511 for SODA-A. We use the official test splits for DIOR-R and SODA-A, and the official validation split for DOTA-v1.0, whose test annotations are not publicly available.

Category Accuracy Score (CAS). Following OF-Diff (Ye et al., 2026), we evaluate semantic accuracy using a ResNet-101 classifier implemented in MMPreTrain. The classifier is trained for 100 epochs using object crops extracted from real training images according to their bounding-box annotations and category labels. During evaluation, object crops are extracted from generated images using their input layouts and assigned the corresponding category labels. To ensure a fair comparison with instance-centric methods and reduce the influence of different instance capacities, we retain at most the first 60 annotated objects per image for all methods. CAS is reported as the Top-1 classification accuracy over the resulting object crops.

YOLOScore. We evaluate layout consistency using a YOLO11l-OBB detector trained on real training images and their annotations. The detector is initialized with the official pretrained YOLO11l weights (`yolo11l.pt`) and trained for 200 epochs with an image resolution of 512×512 and a batch size of 32. During evaluation, we use a batch size of 64, a confidence threshold of 0.25, and an

NMS IoU threshold of 0.6. Following the same protocol across all generative methods, detections on generated images are evaluated against their input layouts. We report mAP, mAP_{50}, and mAP_{75} as layout consistency metrics.

Missing and Spurious Object Rates. We evaluate object-level generation errors by matching detections from the pretrained YOLO11l-OBB detector against the input layouts. A prediction is considered a valid match only if it shares the same category with a layout object and achieves a rotated IoU of at least 0.5. Candidate matches are sorted by IoU in descending order, and greedy one-to-one matching is applied.

Unmatched layout objects are counted as false negatives (FN), indicating missing objects, while unmatched predictions are counted as false positives (FP), indicating spurious objects. The missing and spurious rates are defined as

$$R_{\text{miss}} = \frac{FN}{TP + FN}, \quad R_{\text{spur}} = \frac{FP}{TP + FP}, \tag{13}$$

where TP denotes the number of successfully matched objects.

For detection, we use a confidence threshold of 0.25, an NMS IoU threshold of 0.6, and a maximum of 1,000 detections per image. To analyze robustness across layout densities, images are grouped according to their number of annotated objects into five ranges: 0–10, 11–20, 21–40, 41–80, and above 80. Both rates are computed from the aggregated TP, FP, and FN counts within each group, rather than averaging per-image rates.

A.7 Generalization to Dense Natural Scenes

To further evaluate the applicability of DenseGen beyond conventional remote-sensing benchmarks, we conduct additional experiments on VisDrone, where all generative models are retrained on the corresponding dataset. VisDrone features drone-captured scenes that are closer to natural imagery while retaining high object density, substantial scale variations, and frequent instance overlaps, providing a challenging testbed for dense layout-to-image generation.

Quantitative Comparison. As shown in Table 5, DenseGen achieves the best FID (56.12) and YOLOScore across all three metrics, while obtaining the second-highest CAS (54.67). These results demonstrate its effectiveness in dense natural scenes using only bounding-box annotations, without requiring additional structural priors.

Qualitative Comparison. Figure 9 further compares different methods on VisDrone, where extensive bounding-box overlap makes instance separation particularly challenging. We omit shape- and edge-based spatial-guidance methods, as their required fine-grained structural priors are difficult to obtain reliably in these heavily overlapping scenes, limiting their practical applicability. Existing instance-centric methods often exhibit missing objects, inaccurate placement, or category confusion. In contrast, DenseGen better preserves object groups and spatial arrangements, generating more complete and recognizable objects in heavily overlapping regions.

B Additional Visualization Results

Figure 10 presents additional generation results across diverse remote-sensing scenes. The results illustrate that DenseGen can generate images consistent with the input layouts while preserving recognizable object structures and spatial arrangements.

Table 5: Quantitative comparison on VisDrone. Bold and underline indicate the best and second-best results among evaluated methods, respectively. "–" indicates unavailable results. YOLOScore reports mAP/mAP_{50}/mAP_{75}.

Method	VisDrone		
	FID↓	CAS↑	YOLOScore↑
Instance-Centric Methods			
LayoutDiffusion	68.24	37.35	3.08/5.26/2.97
GLIGEN	79.67	39.96	3.24/5.58/3.67
AeroGen	68.13	40.50	3.59/6.03/3.73
CC-Diff	78.19	54.10	5.44/10.51/4.96
FICGen	76.55	**54.78**	<u>9.81</u>/15.07/<u>10.27</u>
Spatial-Guidance Methods			
ControlNet (shape)	–	–	–
ControlNet (edge)	–	–	–
ControlNet (box)	<u>58.64</u>	48.87	8.76/<u>15.68</u>/9.09
OF-Diff	–	–	–
DenseGen (Ours)	**56.12**	<u>54.67</u>	**13.01/20.47/13.08**

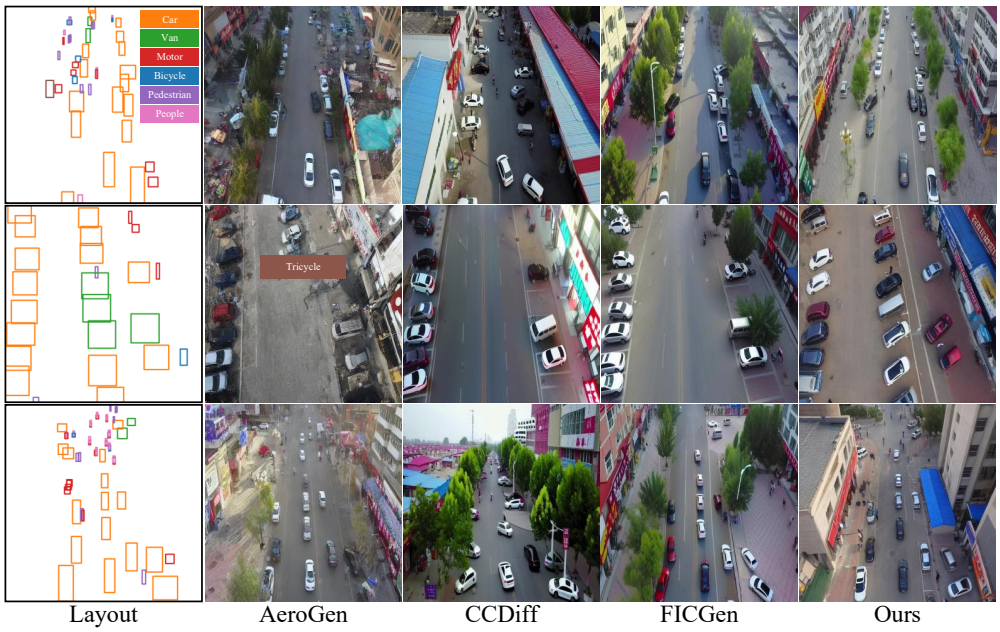

Figure 9: Qualitative comparison on VisDrone. DenseGen better preserves object completeness, spatial arrangements, and category consistency in densely distributed scenes with substantial bounding-box overlap.

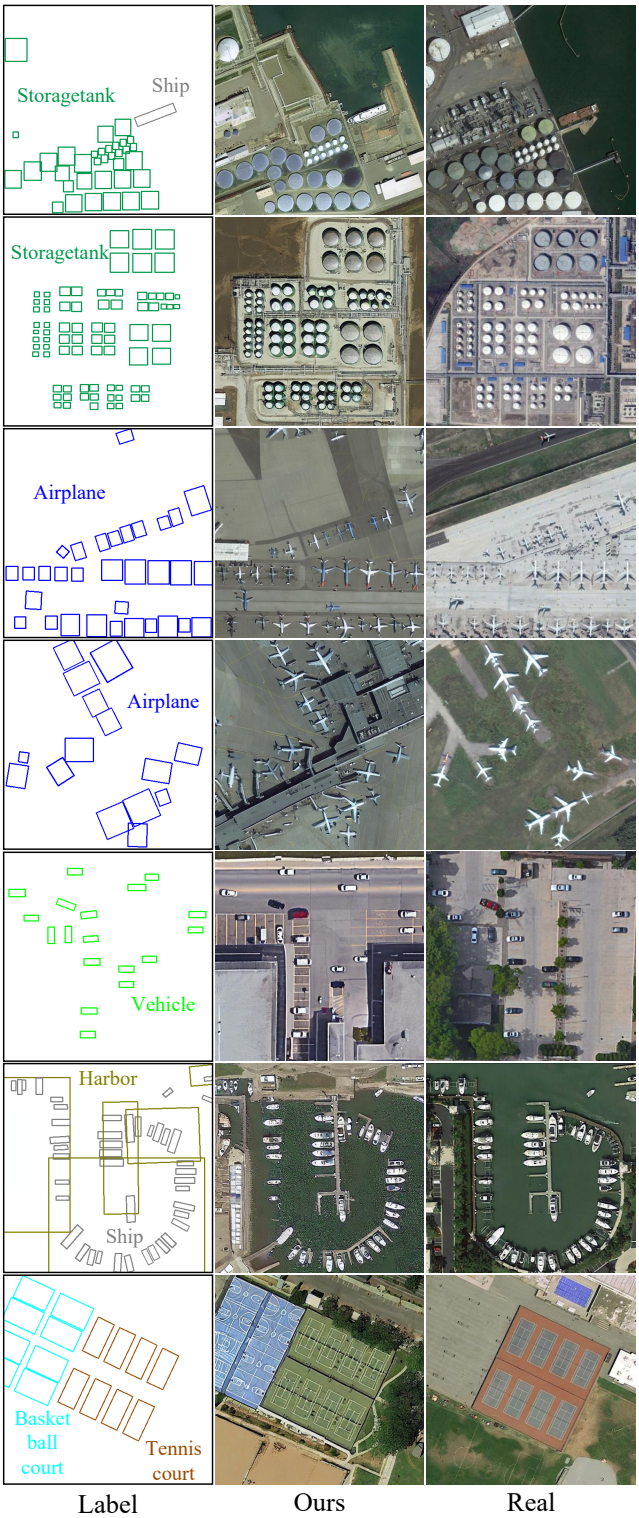

Figure 10: Additional visualization results of DenseGen. Each example presents the input layout, the generated image (Ours), and the corresponding ground-truth image (GT). DenseGen preserves the specified object categories, orientations, and spatial arrangements across diverse remote-sensing scenes.