CONTROLLABLE MULTI-OBJECT INSERTION FOR DETECTION DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Controllable object insertion provides a targeted approach to augmenting object detection datasets that require category–count fidelity, annotation alignment, and source-content preservation beyond visual realism. Several approaches have been proposed to address this grand challenge, but most suffer from different constraints in precise category–count requests and visual consistency, limited coordination across categories and instances, and poor localization of multiple objects of specific category and count. We design SceneP2F, a two-stage trainable framework that connects scene-aware joint planning with layout-aligned generation leveraging explicit instance-level layouts. SceneP2F consists of two novel designs. The first is SCENEPLAN which allocates one class-labeled query per requested instance, jointly predicts scene-compatible object centers, and separately estimates box sizes conditioned on the planned centers and scene context. The second is SCENEALIGN-FILL which combines geometry- and appearance-aware conditioning, box–class-guided regional attention, and an annotation-region-weighted flow-matching objective to render the planned objects while limiting unintended changes to source content. Extensive experiments on three challenging benchmarks UAVDT, VisDrone, and BDD100K show that SceneP2F outperforms representative general-purpose editors and object-insertion methods consistently, demonstrating the value of coordinating instance-level planning and generation for category-targeted detection augmentation.

1 INTRODUCTION

Object-detection data often suffer from a long-tail class distribution, motivating controllable object insertion as a targeted approach to adding selected instances to real images. Rather than collecting and annotating entirely new scenes, object insertion preserve existing backgrounds and annotations while enriching the training distribution with selected instances. To make such augmentation effective, the inserted objects must be visually consistent with the source scene and occupy plausible locations, while content outside the intended editing regions is preserved. Reliable control over object categories and counts is equally important for achieving reliable augmentation benefits.

Existing methods have demonstrated the feasibility of object insertion and its utility for downstream tasks. General-purpose image editors, such as GPT-Image-2.5, FLUX.2 (Labs, 2025), and Qwen-Image-Edit (Wu et al., 2025), provide strong generative priors for instruction-guided editing. More directly relevant to our task, task-specific object-insertion methods (Yang et al., 2024; Zhao et al., 2024; Tewel et al., 2025) improve controllability through specialized training or inference-time guidance. However, general-purpose editors may transfer poorly to specialized domains, limiting their utility for detection augmentation, while existing object-insertion methods often rely on free-form requests without instance-level constraints for coordinating new objects with the scene, resulting in unpredictable placement and unreliable category–count control. Some methods (Yeh et al., 2026; Hou et al., 2026; Petersen et al., 2025) explicitly predict or sample the spatial layout before image synthesis, improving the plausibility of candidate insertion locations. But they still not able to jointly achieve precise multi-object control, visual consistency with the source scene, plausible placement, and background preservation, potentially limiting the benefits of downstream data augmentation.

We propose SceneP2F, a two-stage planning–generation framework for *scene-conditioned, category- and count-controlled multi-object insertion* in challenging downstream domains. Given a source

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.

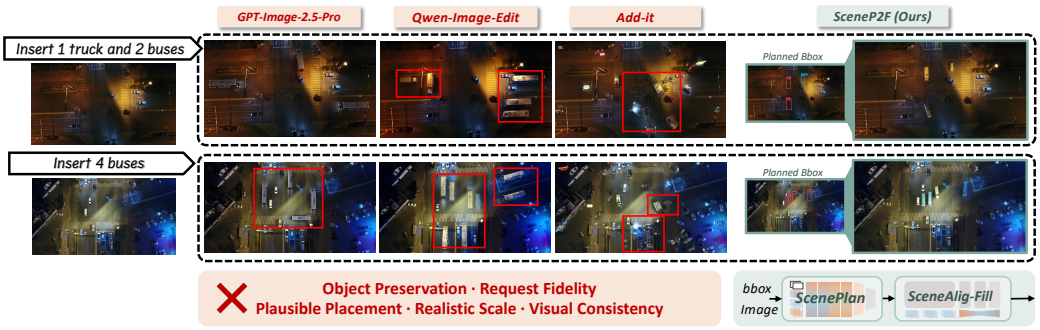

Figure 1: Given a source image and an object insertion request, representative image editing and object insertion approaches may fail to satisfy the requested categories or counts, alter existing objects, or generate implausible placements, scales, and appearances, as highlighted by the red boxes. SceneP2F, a two-stage framework comprising ScenePlan and SceneAlign-Fill, first predicts a scene-compatible, class-labeled box layout and then jointly synthesizes the requested instances within the planned regions, aiming to improve those questions in multi-object insertion task.

scene, the objective of SceneP2F is to add the specified number of *new* instances at plausible locations and scales while maintaining visual consistency and preserving existing content, with the ultimate goal of enabling effective object-level augmentation for downstream detection.

SceneP2F comprises two key designs, namely, **ScenePlan** and **SceneAlign-Fill**. **ScenePlan** decomposes box prediction into size and position components while retaining their conditional dependence, conditioned on the source image, existing annotations, and admissible insertion regions. It jointly models dependencies among the requested instances and incorporates scene context and geometric feasibility constraints to produce a scene-compatible layout with the requested cardinality. Given this layout, **SceneAlign-Fill** jointly renders the planned objects using geometry and appearance conditioning. Regional attention guided by box–class tokens associates each planned region with its target category, while an annotation-region-weighted flow-matching objective emphasizes generation within the designated regions. These mechanisms encourage the synthesized instances to follow the planned boxes and remain visually compatible with the source scene, while limiting unintended changes outside the editing regions. Together, the two stages connect explicit category–count constraints at the layout level with instance-level control during image synthesis.

We evaluate our framework on two challenging UAV object detection benchmarks, UAVDT (Du et al., 2018) and VisDrone (Zhu et al., 2021), with BDD100K (Yu et al., 2020) providing a complementary driving-scene benchmark. Our task-aligned evaluation suite covers two dimensions: *request fidelity*, which measures adherence to the requested categories and counts of newly inserted instances, and *scene fidelity*, which assesses insertion naturalness and preservation of the original scene and existing objects. Comparisons against FLUX.2-dev, Qwen-Image-Edit-2511, Add-SD, and Add-it suggest that SceneP2F improves category–count adherence and provides competitive scene consistency under our evaluation protocol. SceneP2F achieves the lowest mTCE (Mean Total Count Error) across all three datasets, with particularly large reductions over the best competing baseline on UAVDT (2.744 to 0.817) and BDD100K (5.523 to 1.924). It also improves Outsides PSNR by 5.49, 3.94, and 3.33 dB on UAVDT, VisDrone, and BDD100K, respectively, while achieving the highest DINO similarity and CLIP-I across all three benchmarks. Adding the resulting synthetic images to detector training improves bus AP_{50} from 22.9 to 31.0 and truck AP_{50} from 7.2 to 12.2, demonstrating the potential value of category-targeted insertion for detection augmentation.

Our main contributions are summarized as follows: (1) We introduce SceneP2F, a trainable framework for annotated detection data generation that integrates category–count control, object-size modeling, appearance alignment, and source-content preservation through explicit instance-level layouts; (2) We propose SCENEPLAN to jointly plan scene-compatible layouts through coupled size and position modeling. SCENEALIGN-FILL uses box-class-guided regional attention and an annotation-region-weighted flow-matching objective to align generated objects with the planned instances while preserving source content; (3) We show that SceneP2F-generated images and corresponding annotations improve detection of long-tail categories on UAVDT, demonstrating the potential utility of our framework for targeted data augmentation.

2 Related Work

General-Purpose Image Editing and Object Addition. General-purpose Image Editing models, including Qwen-Image-Edit (Wu et al., 2025), FLUX.2 [dev] (Labs, 2025), GLM-Image (Z.ai, 2026), BAGEL (Deng et al., 2025), and OmniGen (Xiao et al., 2025), support flexible text- and image-conditioned generation and editing. Object-addition methods can be organized by their dominant control interface. *Mask-free editing* leaves placement to the model: InstructPix2Pix (Brooks et al., 2023) establishes instruction-guided editing; Diffree (Zhao et al., 2024) and Paint by Inpaint (Wasserman et al., 2025) learn insertion from removal-based supervision; and Add-it (Tewel et al., 2025) uses training-free attention control. *Spatially grounded editing* uses user-specified regions in SmartBrush (Xie et al., 2023) and ObjectAdd (Zhang et al., 2025b), or instruction-derived plans in X-Planner (Yeh et al., 2026) and MIRAGE (Liu & Alaniz, 2026), with the latter supporting multi-instance editing. *Reference-conditioned composition* transfers exemplar content: Paint by Example (Yang et al., 2023) and ObjectStitch (Song et al., 2023) perform diffusion compositing; AnyDoor (Chen et al., 2024b) emphasizes identity and detail preservation; and Insert Anything (Song et al., 2026), FreeInsert (Zhang et al., 2025a), and Multitwine (Tarrés et al., 2025) extend prompting flexibility, geometric and style control, and joint multi-object composition, respectively.

Detection-Oriented Synthetic Data Generation. *Compositional augmentation* constructs training samples by transferring foreground instances, as in Copy-Paste (Dwibedi et al., 2017; Dvornik et al., 2018) and Simple Copy-Paste (Ghiasi et al., 2021). X-Paste (Zhao et al., 2023) and *Stable Diffusion for Aerial Object Detection* (Jian et al., 2023) extend this strategy with diffusion-generated foregrounds. *Layout-conditioned synthesis* generates images from categories and spatial constraints, as in LayoutDiffusion (Zheng et al., 2023) and GLIGEN (Li et al., 2023). GeoDiffusion (Chen et al., 2024a), ODGEN (Zhu et al., 2024), AeroGen (Tang et al., 2024), and UAVGen (Li et al., 2026) extend this paradigm to detection-oriented data generation, while PlanGen (He et al., 2025) incorporates global layout planning. DatasetDM (Wu et al., 2023) instead generates images and predicts their perception annotations. *Task-aware synthesis* improves training utility through data-distribution modeling or detector feedback. Add-SD (Yang et al., 2024) uses frequency-aware category sampling; AeroGen (Tang et al., 2024) models and filters remote-sensing layouts; and UAVGen (Li et al., 2026) combines class prototypes and focal regions with detector-based annotation refinement. DetDiffusion (Wang et al., 2024) uses detector-derived difficulty attributes, whereas *Unbiased Object Detection Beyond Frequency* (Cai et al., 2026) adapts generation to representation gaps using detector feedback. These strategies address synthetic-data composition and utility, complementing our focus on per-image category–count control.

Scene-Conditioned Object Insertion and Layout Planning. Scene-conditioned insertion ranges from implicit placement to explicit geometric planning. Add-SD (Yang et al., 2024) learns insertion from object-removal pairs; *Imagining the Unseen* (Yun et al., 2024) predicts class-conditioned 2D boxes for inpainting; Scene-Aware Location Modeling (Petersen et al., 2025) samples placements using scene geometry and dataset statistics; MonoPlace3D (Parihar et al., 2025) learns distributions over plausible 3D boxes; and PDA (Hou et al., 2026) combines physics-grounded planning with appearance adaptation. Joint layout modeling also appears in traffic simulation: SceneGen places actors autoregressively, while Scenario Diffusion and SceneControl use diffusion to generate actor configurations under scene constraints (Tan et al., 2021; Pronovost et al., 2023; Lu et al., 2024). Scenario Dreamer jointly generates lane graphs and actor boxes, and ScenarioControl supports text- or image-conditioned vectorized scenes (Rowe et al., 2025; Gao et al., 2026). For detection-oriented insertion, the challenge extends beyond plausible placement to maintaining instance-level consistency between a complete category–count request and the generated image. Our framework uses an explicit, class-labeled 2D layout to connect scene-aware joint planning with layout-aligned generation, promoting request fidelity, image-annotation alignment, and source-content preservation.

3 Methodology

3.1 Framework Overview

Given an aerial image I, its existing object configuration $\mathcal{C}$, and a structured category–count request $\mathcal{R} = \{(c_k, n_k)\}_{k=1}^{K}$, our goal is to generate an edited image $\hat{I}$ and insertion annotations b^* containing exactly n_k new instances of each category c_k.

We address these requirements in a two-stage **ScenePlan** method. As shown in Fig. 2. the framework consists of two modules: ScenePlan and SceneAlign-FIll. Firstly, **ScenePlan** expands $\mathcal{R}$ into category-specific queries, jointly denoises their centers conditioned on I, $\mathcal{C}$, and the validity mask, and employs a scene-aware size predictor to realize them as complete boxes. Finally, **SceneAlign-Fill** inserts requested objects by rendering the planned objects with scene-aware conditioning and box-restricted regional attention, ensuring appearance consistency, and source-scene preservation. Details of SceneP2F training among three datasets are provided in Appendix E.1.

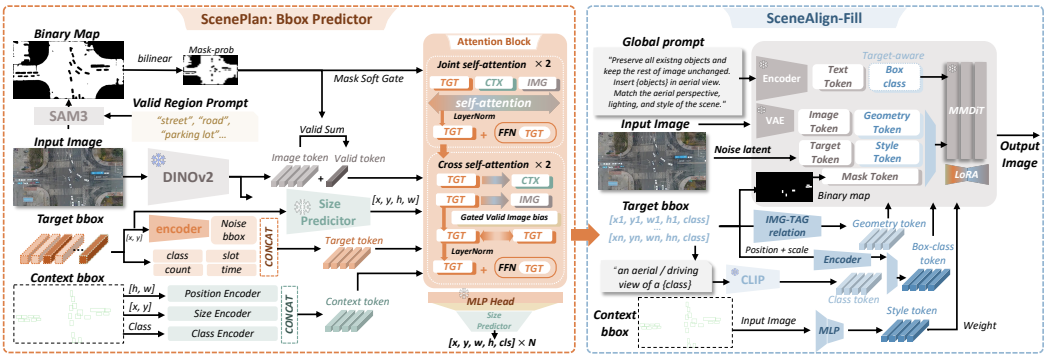

Figure 2: Overview of the SceneP2F framework, comprising ScenePlan (left) and SceneAlign-Fill (right). ScenePlan first jointly predicts object centers from image features, existing boxes, and category–count queries, while a size predictor estimates the corresponding box extents. Given the resulting class-labeled layout, SceneAlign-Fill uses geometric, categorical, and scene-style conditioning to synthesize the requested objects while limiting changes outside the target regions.

3.2 SCENEPLAN: PLANNED SCENE-CONDITIONED BOX PLANNING

We formulate insertion-layout planning as category and count constrained set prediction: given a background image I, existing context annotations $\mathcal{C} = \{(\mathbf{b}_j^{\text{ctx}}, c_j^{\text{ctx}})\}_{j=1}^{N_{\text{ctx}}}$, and a request $\mathcal{R} = \{(c_k, n_k)\}_{k=1}^{K}$, our ScenePlan predicts a set of target boxes:

$$\widehat{\mathcal{L}} = \left\{ \left(\widehat{\mathbf{b}}_i^*, c_i^{\text{tgt}} \right) \right\}_{i=1}^{N_{\text{tgt}}}, \quad \sum_{i=1}^{N_{\text{tgt}}} \left[c_i^{\text{tgt}} = c_k \right] = n_k, \quad k = 1, \ldots, K. \tag{1}$$

Each box is parameterized as $\widehat{\mathbf{b}}_i^* = (\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$, where $(\hat{x}_i, \hat{y}_i) \in [0,1]^2$ and $(\hat{w}_i, \hat{h}_i) \in (0,1]^2$ denote its normalized center and extent, respectively. We expand the category–count request into $N_{\text{tgt}} = \sum_{k=1}^{K} n_k$ active queries with fixed category assignments, such that the category multiplicities and output cardinality are determined by construction at the layout-representation level.

Removal Dataset Construction. ScenePlan is trained on our task-specific dataset. Inspired by works (Canberk et al., 2024; Wasserman et al., 2025; Winter et al., 2024; Li et al., 2025; Winter et al., 2025; Yu et al., 2025a) on object insertion and spatial prediction methods trained by their removal-pair datasets, we construct ours by mapping entire annotations to a unified label space and splitting valid instances into target to remove and context boxes to retain, and further sampling and filtering

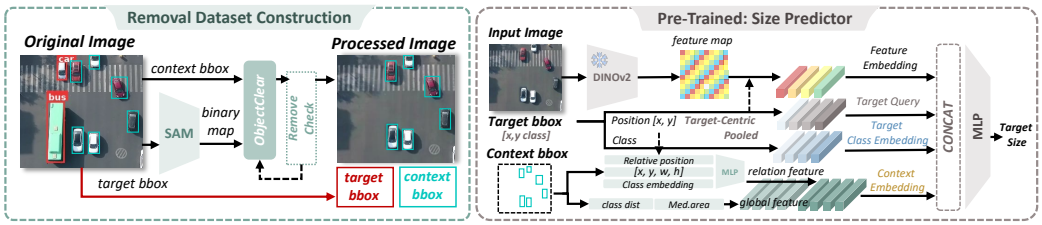

Figure 3: Removal dataset construction (left) and pretrained size predictor (right). The removal-data construction pipeline uses SAM2 to generate masks and ObjectClear to remove the target objects, while a verification module triggers iterative reprocessing when the check fails. Size predictor submodel estimates target-box dimensions from image features, context boxes, and input center.

details are provided in Sec. A.2. As shown in Fig. 3, we generate slightly dilated target masks with SAM2 (Ravi et al., 2024), protect annotated non-target regions, and apply ObjectClear (Zhao et al., 2026), optionally with a second refinement pass to filter unsuccessful edits by using edge-density and image-difference criteria.

Decoupled size-coords prediction. Before training the main ScenePlan model, we first design and pre-train a sub-model G_ϕ to estimate the width and height of each target box, which is illustrated in Fig. 3. It takes the image features, context-box tokens, and the planned center query as input and uses lightweight feature-fusion layers followed by an MLP regressor to predict

$$\widehat{\mathbf{s}}_i = G_\phi\left(I, \mathcal{C}, c_i, \widehat{\mathbf{z}}_{0,i}\right), \qquad \widehat{\mathbf{b}}_i = \left(\widehat{\mathbf{z}}_{0,i}, \widehat{\mathbf{s}}_i\right), \tag{2}$$

where c_i is the request class, $\widehat{\mathbf{z}}_{0,i} = (\hat{x}_i, \hat{y}_i)$ and $\widehat{\mathbf{s}}_i = (\hat{w}_i, \hat{h}_i)$. We initially attempted to predict complete bounding boxes jointly, but found the predictions unstable, particularly for width and height, which often deviated substantially from the ground-truth extents. This observation motivated us to decouple center prediction from size estimation. We provide ablation of the model in Appendix.C.1.

Plausible Region Control. ScenePlan incorporates scene-conditioned positional validity through spatial-validity mask $M_{valid} \in [0, 1]$ generated by SAM3 (Carion et al., 2025). Dataset-specific prompts cover plausible support regions while excluding overly generic terms that may merge semantically distinct surfaces, and details are described in Sec.B. We incorporate the mask in three complementary ways: (1) construct a valid summary token by mask-weighted pooling of image features followed by an MLP: (2) transform the mask into a logarithmic soft bias on target-to-image attention logits; (3) include a region-validity penalty loss illustrated in the next part.

ScenePlan Diffusion Model ScenePlan encodes I input image using DINOv2 and then projected to imgae token τ_{u_k}. The context object token τ_{c_j} and target token $\tau_{z_i^0}$ are encoded as

$$\tau_{c_j} = \mathrm{MLP}_{xy}(b^{ctx}_{j,xy}) + \mathrm{MLP}_{size}(b^{ctx}_{j,\log wh}) + e_{\mathrm{class}}(c^{ctx}_j),$$
$$\tau_{z_i^0} = \mathrm{MLP}_{xy}(z^{tgt}_{t,i}) + e_t(t) + e_{\mathrm{slot}}(i) + e_{\mathrm{count}}(N_{tgt}) + e_{\mathrm{class}}(c^{tgt}_i)$$

, where $\mathbf{z}_{t,i}$ is the noisy center of target query i at timestep t, and $e(\cdot)$ denotes a learnable embedding. The tokens are processed by a four-layer relation-aware transformer. The first two layers perform joint self-attention over the target, context, and image tokens, whereas the remaining two refine the target tokens through target to-target self-attention and cross-attention to the context and image tokens. The frozen size predictor G_ϕ provides provisional geometry for relative cross-attention biases and subsequently predicts the final box extents from the refined centers and scene context, decoupling center planning from size estimation while preserving their conditional dependence.

ScenePlan applies diffusion only to the centers of the N_{tgt} active target queries. Given their noisy center set $\mathbf{Z}_t = [\mathbf{z}_{t,i}]_{i=1}^{N_{\mathrm{tgt}}}$ at timestep t and the conditioning information $\Omega = (I, \mathcal{C}, \mathcal{R}, \mathbf{M}_{\mathbf{valid}})$, the denoiser jointly predicts the corresponding velocity set using a $T = 100$-step DDPM:

$$\widehat{\mathbf{V}}_t = f_\theta(\mathbf{Z}_t, t, \Omega), \qquad \mathbf{v}_{t,i} = \sqrt{\bar{\alpha}_t}\boldsymbol{\epsilon}_i - \sqrt{1-\bar{\alpha}_t}\mathbf{z}_{0,i}, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2), \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. Starting from Gaussian center noise, the standard DDPM reverse process progressively refines the joint center state and outputs the planned center set $\widehat{\mathbf{Z}}_0$, obtaining the corresponding size by G_ϕ.

For training objective, we supervise ScenePlan using a velocity-prediction loss, a timestep-weighted set-level coordinate loss, and a validity loss that encourages the recovered centers to lie within the shared admissible region:

$$\mathcal{L}_{\mathrm{coord}} = \mathbb{E}_{t,\epsilon}\left[w_t\, \mathcal{D}_{\mathrm{set}}\left(\widehat{\mathcal{Z}}_{0|t}, \mathcal{Z}_0\right)\right], \quad \mathcal{L}_{\mathrm{mask}} = \mathbb{E}_t\left[\frac{1}{N_{\mathrm{tgt}}}\sum_i\left(1 - \rho\left(\mathbf{M}_{valid}, \widehat{\mathbf{b}}_{0|t,i}\right)\right)\right]. \tag{4}$$

Here, w_t is an SNR-based timestep weight, $\mathcal{D}_{\mathrm{set}}$ denotes the average Smooth-L1 coordinate loss after class-constrained Hungarian matching between the recovered and ground-truth centers. $\widehat{\mathcal{Z}}_{0|t} = \{\widehat{\mathbf{z}}_{0|t,i}\}_{i=1}^{N_{\mathrm{tgt}}}$ denotes the recovered clean-center set and $\rho(\mathbf{M}_{\mathbf{valid}}, \widehat{\mathbf{b}}_{0|t,i})$ denotes the differentiable fraction of the predicted box covered by the valid-region mask. The final objective $L_{\mathrm{ScenePlan}}$ is

$$\mathcal{L}_{\mathrm{ScenePlan}} = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\frac{1}{N_{\mathrm{tgt}}}\sum_i \|\widehat{\mathbf{v}}_{t,i} - \mathbf{v}_{t,i}\|_2^2\right] + \lambda_{\mathrm{coord}}\mathcal{L}_{\mathrm{coord}} + \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}}. \tag{5}$$

In the inference, we apply a deterministic post-process to locally adjust predictions that violate basic boundary or overlap constraints to obtain the final $\hat{b}_i^*$. The requested categories and instance count remain unchanged. Implementations details for each dataset are shown in Appendix C.2.

3.3 SceneAlign-Fill: Annotation-Aligned Scene Refinement

Given an input scene I, a target layout $\hat{\mathcal{L}} = \{(b_i^*, c_i)\}_{i=1}^{N_{\text{tgt}}}$, and a text prompt T, SceneAlign-Fill synthesizes the requested objects while enforcing category–box alignment, preserving unedited content, and maintaining scene consistency. Since FLUX.1-Fill relies mainly on a shared inpainting mask and a global prompt, it provides limited instance-level control over multi-category, multi-instance insertion. We address this limitation through scene and annotation image conditioning, box-token regional attention, and an annotation-weighted flow-matching objective.

Scene-conditioned image conditioning. Let z_{cond} denote the VAE latent representation of x_{safe}. For each image token p, we construct the image-side condition

$$C_p = \left[z_{\text{cond},p}; \tau_{m_p}; \tau_g; \tau_s\right], \tag{6}$$

where τ_{m_p} is the packed inpainting-mask feature, τ_g is a geometry token, and τ_s is a scene-level style token. The geometry token encodes the relative position of image token with respect to its associated target box with the its scale. The style token is obtained by aggregating color and latent texture features from visible context objects and is shared across all image tokens. Together, these conditions enable spatially controlled multi-instance synthesis while encouraging the inserted objects to follow the visual consistency. Details of token process are shown in Appendix D.

Box-token regional attention. To provide region-specific semantic control, we represent each target annotation with a box token:

$$u_i = f_{\text{box}}\left([e_{c_i}; \phi(b_i)]\right). \tag{7}$$

Here, e_{c_i} is obtained by encoding a local prompt for category c_i with the frozen CLIP text encoder, while $\phi(b_i)$ represents the normalized position and scale of box b_i. The resulting box tokens are appended to the FLUX text sequence and used for regional attention in the late Transformer blocks: image tokens within b_i attend only to the corresponding box token u_i, whereas global text tokens remain accessible to all regions. This restricts category-specific information to the intended boxes, reducing cross-instance semantic leakage and improving image–annotation alignment.

4 Experiments

4.1 Multi-Object Insertion

Settings. We train dataset-specific models on UAVDT (Du et al., 2018), VisDrone (Zhu et al., 2021), and BDD100K (Yu et al., 2020). We use the official training splits for VisDrone and BDD100K and a random 40/10 train/validation fixed sequence split for UAVDT. All methods are evaluated with the same predefined request schedules on all 1,609 VisDrone test images and 1,000 sampled images each from BDD100K and UAVDT. We use category-balanced single- and multi-class requests with $N \in \{1, \ldots, 5\}$ and an equal budget for each N. All methods use input prompt *"Preserve all existing objects and keep the rest of the image unchanged. Insert {objects} into the {scene type} driving scene while matching its perspective, lighting, and style,"* where *scene type* is set to *aerial* or *street-view* according to the dataset. Further details are provided in Appendix E.3.

Baseline Methods. We compare our method with two groups of baselines. General-purpose editors include FLUX.2 [dev] (Labs, 2025), evaluated using only its official checkpoint, and Qwen-Images-Edit-2511 (Wu et al., 2025), evaluated both with its official checkpoint and after fine-tuning on each aerial dataset. Object-insertion baselines include Add-it (Tewel et al., 2025), a training-free method built on FLUX.1-dev, and Add-SD (Yang et al., 2024), which is closely related to our task but was originally developed for a different image domain. For each aerial benchmark, we fine-tune Add-it (Tewel et al., 2025) from pretrained FLUX.1-dev weights and adapt Add-SD (Yang et al., 2024) using the corresponding training set. We do not fine-tune the baseline methods on BDD100K (Yu et al., 2020), as its street-view imagery exhibits relatively little domain shift. Appendix. E.2 illustrates the fine-tuned details. To ensure a fair comparison under the same supervision setting, we also evaluate our model without access to ground-truth bounding-box annotations.



Figure 4: Qualitative comparison of SceneP2F with image-editing models (FLUX.2-dev and Qwen-Image-Edit) and object-insertion baselines (Add-SD and Add-it) on the joint multi-object insertion task. The first three rows present results on UAVDT, the subsequent two rows on VisDrone, and the final row on BDD100K. The leftmost column shows the requested categories and counts.

Metrics. We use dataset-specific GroundingDINO detectors (Liu et al., 2023) to identify inserted objects, and we present details of the detector in Appendix E.3.2. We report three groups of metrics: (1) **Request Fidelity**, measured by Mean Total Count Error (mTCE)↓ and Mean Class Count Error (mCCE)↓, which quantify total and class-wise deviations from the requested counts; (2) **Visual Consistency**, measured by LPIPS↓, DINO_{Sim}↑, and CLIP-I↑ for perceptual distance, semantic similarity, and visual-semantic similarity, respectively; and (3) **Outside Preservation**, measured by PSNR↑ and SSIM↑ over unedited regions. Detected and ground-truth boxes define the generated and reference crops, respectively. LPIPS is averaged over crop pairs and images, whereas DINO_{Sim} and CLIP-I are computed by comparing each generated-crop feature with the image-level mean reference feature and averaging the cosine similarities. More details are presented in Appendix.E.3.3.

Results. We present visualization comparison in Fig. 4. Table. 1 shows that our method achieves the strongest overall performance. It obtains the lowest mTCE and mCCE among all baselines on the three datasets, with clear gains on UAVDT (Du et al., 2018) ($0.817/1.189$ vs. $2.744/2.105$) and BDD100K ($1.924/1.648$ vs. $5.523/2.576$), demonstrating more accurate category–count control. Our method also achieves the highest CLIP-I scores on all datasets and consistently lower LPIPS values than the baselines, indicating that the inserted objects are natural and visually consistent with the original scenes. The results show that our method demonstrates good overall performance regardless of whether GT annotations are provided. We also evaluate the accuracy of insertion count shown in Fig6, which is calculated the rate of successfully inserting request objects into the image.

4.2 Detection Improvement

To evaluate downstream utility, we compare SceneP2F with the unaugmented training baseline, Copy-Paste (Dwibedi et al., 2017), and X-Paste (Zhao et al., 2023) on UAVDT (Du et al., 2018), where cars account for over 90% of the instances. In separate bus- and truck-augmentation settings, each method augments 40% of the training images by inserting two instances per image. The augmented images are added to the original training set, and a pretrained YOLO11-L (Jocher & Qiu, 2024) is fine-tuned for 100 epochs with an image size of 1024, a batch size of 8. As shown in Table 2, SceneP2F improves bus/truck recall by 29.5%/28.4% and inserted-class mAP_{50} by 35.4%/69.9%, respectively, demonstrating the effectiveness of targeted augmentation for rare-class detection on UAVDT. Additional details and visualizations results are provided in Appendix F.

Table 1: Quantitative comparison of general image-editing and object-insertion methods on UAVDT, VisDrone, and BDD100K. The best results are shown in bold, and the second-best are underlined.

Dataset	Method Category	Method	Fine-tuned	Request Fidelity		Visual Consistency			Outside Preservation	
				mTCE ↓	mCCE ↓	LPIPS ↓	DINO_{Sim} ↑	CLIP-I ↑	PSNR ↑	SSIM ↑
UAVDT (Du et al., 2018)	Image Editing	FLUX.2-dev (Labs, 2025)		7.001	3.076	0.5561	0.4753	0.8131	18.9585	0.8405
		Qwen-Image-Edit (Wu et al., 2025)		5.596	2.458	0.5588	0.4635	0.8188	20.6050	0.8973
			✓	14.959	7.603	0.5051	0.4499	0.8452	19.3728	0.8408
	Object Insertion	Add-it (Tewel et al., 2025)	✓	8.529	4.594	0.5164	0.4508	0.8410	21.5469	0.8987
		Add-SD (Yang et al., 2024)	✓	2.744	2.105	0.5751	0.4295	0.8356	22.2962	0.9010
	Ours (w/o context bbox)			0.916	1.243	**0.4727**	0.4901	0.8532	27.7671	**0.9635**
	Ours			**0.817**	**1.189**	0.4822	**0.4941**	**0.8566**	**27.7819**	0.9634
VisDrone (Zhu et al., 2021)	Image Editing	FLUX.2-dev (Labs, 2025)		25.557	10.661	0.5564	0.5098	0.8582	22.3622	0.9181
		Qwen-Image-Edit (Wu et al., 2025)		19.864	9.841	0.5558	0.5370	0.8671	20.8987	0.8861
			✓	39.184	19.323	0.5602	0.5282	0.8595	18.5221	0.7800
	Object Insertion	Add-it (Tewel et al., 2025)	✓	44.760	20.926	0.5629	0.5305	0.8623	19.8011	0.8273
		Add-SD (Yang et al., 2024)	✓	7.776	5.249	0.5656	0.5304	0.8655	22.6756	0.9148
	Ours (w/o context bbox)			11.605	6.187	0.5557	0.5364	0.8682	23.5125	0.9221
	Ours			**7.510**	**5.018**	**0.5355**	**0.5611**	**0.8784**	**26.6133**	**0.9669**
BDD100K (Yu et al., 2020)	Image Editing	FLUX.2-dev (Labs, 2025)		7.773	3.420	0.6178	0.4711	0.8215	27.8150	0.9428
		Qwen-Image-Edit (Wu et al., 2025)		5.523	2.576	0.6297	0.4701	0.8150	24.1381	0.9325
	Object Insertion	Add-it (Tewel et al., 2025)		12.346	5.880	0.6162	0.4730	0.8234	23.8459	0.9101
	Ours (w/o context bbox)			2.433	**1.516**	**0.4839**	0.4924	0.8493	19.5459	0.5573
	Ours			**1.924**	1.648	0.5913	**0.5400**	**0.8626**	**31.1420**	**0.9820**

Table 2: Downstream detection performance of rare-class augmentation methods on UAVDT using YOLO11-L. In two separate settings, each method augments 40% of the training images with either two buses or two trucks per image, and the resulting samples are added to the original training set.

Dataset	Inserted Objects	Method	Inserted Class			All Classes	
			Recall	mAP_{50}	$\text{mAP}_{50:95}$	mAP_{50}	$\text{mAP}_{50:95}$
UAVDT (Du et al., 2018)	2 buses	Original dataset	0.403	0.229	0.168	0.341	0.208
		Copy-and-Paste	0.408	0.268	0.190	0.358	0.211
		X-Paste	0.464	0.283	0.210	0.351	0.212
		Ours (SceneP2F)	**0.522**	**0.310**	**0.226**	**0.371**	**0.224**
	2 trucks	Original dataset	0.246	0.0718	0.045	0.341	0.208
		Copy-and-Paste	0.152	0.0659	0.0428	0.343	0.207
		X-Paste	0.202	0.0938	0.0590	0.347	**0.209**
		Ours (SceneP2F)	**0.316**	**0.122**	**0.0775**	**0.351**	0.208

4.3 Ablations

ScenePlan: Module Ablations We ablate ScenePlan using seven variants covering size prediction, region-mask conditioning, and attention design. SZ-xywh jointly predicts box coordinates and sizes, while SZ-mean uses mean sizes; tok0, loss0, gate0 remove the mask token, loss, and attention bias, respectively; and ATT-L8 and ATT-Joint use eight attention layers and joint self-attention throughout. Tab. 5 shows that the full model achieves the best overall placement validity and collision avoidance. For model selection and ablation only, we evaluate predicted layouts using valid regions obtained from SAM3 and four metrics: VRC (mean valid-region box coverage), Valid@0.9 (boxes with $\geq$90% coverage), ETC (existing–target collisions), and TTC (target–target collisions).

ScenePlan: Box Prediction Comparison. Visualization in Fig. 7 compares ScenePlan with Random and Dataset-Prior. Random uniformly samples centers and draws box sizes from class-specific training ranges, whereas Dataset-Prior follows ODGEN (Zhu et al., 2024) and samples position, area, and aspect ratio from class-wise Gaussian priors. Across object categories and insertion counts, ScenePlan produces more plausible locations and scales, indicating better adaptation to scene.

SceneAlign-Fill: Methods Comparison We compares SceneAlign-Fill with PowerPaint (Zhuang et al., 2023), BrushNet (Ju et al., 2024), FLUX.1-Fill, and its LoRA-fine-tuned variant on 100 UAVDT removal-test images (10 per sequence), with all specified boxes filled in a single pass. Results in Tab. 3 show that SceneAlign-Fill performs better across all metrics, indicating improved category–count control, local appearance consistency, and semantic alignment rendering ability.

SceneAlign-Fill: Module Ablations. To validate the components of SceneAlign-Fill, we also conduct ablation experiments on 100 images from the UAVDT (Du et al., 2018) removal test set. We separately remove the geometry token (G0) for target-position encoding, the style token (ST0) for

Figure 5: ScenePlan ablation. VRC denotes the mean valid-region coverage, Valid@0.9 the proportion of boxes with at least 90% coverage, and ETC and TTC the exist–target and target–target collision rates.

Group	Method	VRC ↑	Valid@0.9 ↑	1-ETC ↑	1-TTC ↑	Average ↑
Full Model	**Ours**	**0.7671**	**0.6339**	0.9057	**0.9218**	**0.8071**
Size	SZ-xywh	0.7520	0.6050	0.8796	0.9150	0.7879
	SZ-mean	0.7538	0.6163	0.9049	0.9188	0.7985
Mask	MK-tok0	0.7489	0.6073	0.9074	0.8976	0.7903
	MK-loss0	0.6895	0.5600	0.9021	0.9165	0.7670
	MK-gate0	0.6887	0.5590	**0.9076**	0.9119	0.7668
Attention	ATT-L8	0.7543	0.6128	0.9028	0.8971	0.7918
	ATT-Joint	0.6869	0.5429	0.8937	0.9129	0.7591

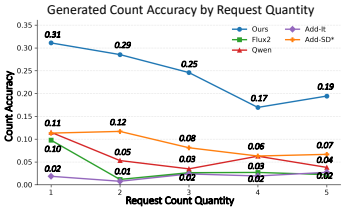

Figure 6: Mean count accuracy of the requested object count. SceneP2F achieves the highest accuracy across request quantities. Add-SD* is averaged over UAVDT and VisDrone only.



Figure 7: Qualitative comparison of Random, Dataset-Prior, and ScenePlan object placements on VisDrone and UAVDT, illustrating differences in predicted locations and scales.

Table 3: Quantitative performance comparison of representative object-inpainting methods on the UAVDT removal test set in terms of request fidelity and within-box visual consistency.

Method	mTCE ↓	mCCE ↓	LPIPS ↓	DINO_{Sim} ↑	CLIP-I ↑
PowerPaint (Zhuang et al., 2023)	2.03	0.9403	0.6901	0.2598	0.7537
BrushNet (Ju et al., 2024)	1.68	0.7262	0.6551	0.2487	0.7346
FLUX-Fill-Joint	2.12	0.6798	0.5495	0.3318	0.7872
FLUX-Fill-Joint-FT	2.38	1.0731	0.5347	0.2940	0.7654
Ours (SceneAlign-Fill)	**0.77**	**0.3388**	**0.4229**	**0.4128**	**0.8252**

Table 4: SceneAlign-Fill Modules Ablation on UAVDT. G0 for Geometry Token, ST0 for Style Token and TX-noR for regional attention.

Method	LPIPS ↓	DINO_{Sim} ↑	CLIP-I ↑
G0	0.4305	0.3972	0.8195
ST0	0.4335	0.3951	0.8183
TX-noR	0.4339	0.4046	0.8201
Ours(SceneAlign-Fill)	**0.4229**	**0.4128**	**0.8252**

color and style guidance, and the box-token regional attention (TX-noR) for region-specific semantic control. As shown in Tab. 4, the full model outperforms all ablated variants, indicating that these components jointly improve the geometric, stylistic, and semantic consistency of inserted objects.

5 Analysis and Conclusion

Limitation Analysis. Despite enabling scene-conditioned, category- and count-controlled multi-object insertion in aerial and road scenes, SceneP2F has several limitations. ScenePlan estimates object centers more reliably than box sizes, whose errors can propagate to SceneAlign-Fill and cause scale distortion, misalignment, or incomplete rendering. SceneAlign-Fill also struggles with extremely small objects and rare categories, sometimes producing fragmented or missing instances. Moreover, although SceneP2F improves category–count adherence, its absolute accuracy remains limited. Finally, qualitative observations suggest that GPT-Image-2.5-Pro can perform better on several aspects of the task. Since it was not included in the same controlled evaluation as the open-source baselines, we do not claim superiority over it; a unified comparison is left for future work.

Conclusion. We presented SceneP2F, a two-stage framework for scene-conditioned, category- and count-controlled multi-object insertion. By connecting ScenePlan and SceneAlign-Fill through explicit class-labeled layouts, the framework coordinates scene-compatible placement with annotation-aligned rendering while promoting source-content preservation. Experiments on challenging aerial and road scenes demonstrate improved request and scene fidelity, while targeted augmentation on UAVDT (Du et al., 2018) improves detection of underrepresented categories. These findings highlight the importance of maintaining instance-level consistency across requests, layouts, and generated objects to turn visually plausible insertions into useful annotated training data.

AI USE STATEMENT

We used generative AI tools for language polishing, improving manuscript readability, discussing research methodology and experimental design, and assisting with minor code utilities and debugging. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for all contents, claims, and artifacts in this work.

ETHICS STATEMENT

This work investigates controllable object insertion as a means of reducing the cost of constructing and annotating object-detection datasets, particularly for underrepresented categories and challenging conditions. Synthetic data may inherit biases or artifacts from source datasets and pretrained models, and their associated annotations may contain errors. Models trained with such data should therefore be evaluated on real-world data and should not be deployed in safety-critical settings without task-specific validation.

REPRODUCIBILITY STATEMENT

The main paper and appendix describe the proposed method, data-processing pipeline, training and inference settings, evaluation protocol, and downstream detection experiments. To support independent verification, we intend to make the associated implementation artifacts available after publication where permitted by applicable dataset and model licenses.

REFERENCES

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

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June 2023.

Xinhao Cai, Liulei Li, Gensheng Pei, Tao Chen, Jinshan Pan, Yazhou Yao, and Wenguan Wang. Unbiased object detection beyond frequency with visually prompted image synthesis. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 37332–37355, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3edde8c8a05871c0172a6186a7d4850d-Paper-Conference.pdf.

Alper Canberk, Maksym Bondarenko, Ege Ozguroglu, Ruoshi Liu, and Carl Vondrick. EraseDraw: Learning to insert objects by erasing them from images. In *European Conference on Computer Vision*, pp. 144–160. Springer, 2024. doi: 10.1007/978-3-031-72986-7_9.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Kai Chen, Enze Xie, Zhe Chen, Yibo Wang, Lanqing HONG, Zhenguo Li, and Dit-Yan Yeung. Geodiffusion: Text-prompted geometric control for object detection data generation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 8979–9001, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25886d7a7cf4e33fd44072a0cd81bf30-Paper-Conference.pdf.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6593–6602, June 2024b.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking, 2018. URL https://arxiv.org/abs/1804.00518.

Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Lili Gao, Yanbo Xu, William Koch, Samuele Ruffino, Luke Rowe, Behdad Chalaki, Dmitriy Rivkin, Julian Ost, Roger Girgis, Mario Bijelic, and Felix Heide. Scenariocontrol: Vision-language controllable vectorized latent scenario generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2918–2928, June 2021.

Runze He, Bo Cheng, Yuhang Ma, Qingxiang Jia, Shanyuan Liu, Ao Ma, Xiaoyu Wu, Liebucha Wu, Dawei Leng, and Yuhui Yin. Plangen: Towards unified layout planning and image generation in auto-regressive vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18143–18154, October 2025.

Yingyan Hou, Xianchi Dong, Chao Ren, Wanxuan Lu, Zihan Wei, Hongfeng Yu, Yixiao Wang, and Xian Sun. Plan, decouple, assimilate: Physics-aware object insertion in remote sensing imagery. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ojx0CyGXHJ.

Yueru Jia, Aosong Cheng, Yuhui Yuan, Chuke Wang, Ji Li, Huizhu Jia, and Shanghang Zhang. DesignEdit: Unify spatial-aware image editing via training-free inpainting with a multi-layered latent diffusion framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3958–3966, 2025. doi: 10.1609/aaai.v39i4.32414.

Yanan Jian, Fuxun Yu, Simranjit Singh, and Dimitrios Stamoulis. Stable diffusion for aerial object detection, 2023. URL https://arxiv.org/abs/2311.12345.

Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL https://github.com/ultralytics/ultralytics.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion, 2024.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Ruibin Li, Tao Yang, Song Guo, and Lei Zhang. RORem: Training a robust object remover with human-in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14024–14035, 2025.

Wenhao Li, Zimeng Wu, Yu Wu, Zehua Fu, and Jiaxin Chen. Visual prototype conditioned focal region generation for uav-based object detection, 2026. URL https://arxiv.org/abs/2604.02966.

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 (CVPR)*, pp. 22511–22521, June 2023.

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.

Ziqian Liu and Stephan Alaniz. Mirage: Benchmarking and aligning multi-instance image editing, 2026. URL https://arxiv.org/abs/2604.05180.

Jack Lu, Kelvin Wong, Chris Zhang, Simon Suo, and Raquel Urtasun. Scenecontrol: Diffusion for controllable traffic scene generation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Rishubh Parihar, Srinjay Sarkar, Sarthak Vora, Jogendra Nath Kundu, and R. Venkatesh Babu. Monoplace3d: Learning 3d-aware object placement for 3d monocular detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6531–6541, June 2025.

Jens Petersen, Davide Abati, Amirhossein Habibian, and Auke Wiggers. Scene-aware location modeling for data augmentation in automotive object detection, April 2025. URL https://arxiv.org/abs/2504.17076.

Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68873–68894. Curran Associates, Inc., 2023. doi: 10.52202/075280-3014. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d95cb79a3421e6d9b6c9a9008c4d07c5-Paper-Conference.pdf.

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. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17207–17218, June 2025.

Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in dit. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(11):9097–9105, 2026. doi: 10.1609/aaai.v40i11.37866. URL https://doi.org/10.1609/aaai.v40i11.37866.

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18310–18319, June 2023.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022. doi: 10.1109/WACV51458.2022.00323.

Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 892–901, June 2021.

Datao Tang, Xiangyong Cao, Xuan Wu, Jialin Li, Jing Yao, Xueru Bai, and Deyu Meng. Aerogen: Enhancing remote sensing object detection with diffusion-driven data generation. *arXiv preprint arXiv:2411.15497*, 2024.

Gemma Canet Tarrés, Zhe Lin, Zhifei Zhang, He Zhang, Andrew Gilbert, John Collomosse, and Soo Ye Kim. Multitwine: Multi-object compositing with text and layout control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8094–8104, June 2025.

Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZeaTvXw080.

Yibo Wang, Ruiyuan Gao, Kai Chen, Kaiqiang Zhou, Yingjie Cai, Lanqing Hong, Zhenguo Li, Lihui Jiang, Dit-Yan Yeung, Qiang Xu, and Kai Zhang. Detdiffusion: Synergizing generative and perceptive models for enhanced data generation and perception, 2024. URL https://arxiv.org/abs/2403.13304.

Navve Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. Paint by inpaint: Learning to add image objects by removing them first. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18313–18324, June 2025.

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*, pp. 112–129. Springer, 2024. doi: 10.1007/978-3-031-72980-5_7.

Daniel Winter, Asaf Shul, Matan Cohen, Dana Berman, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectMate: A recurrence prior for object insertion and subject-driven generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16281–16291, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304, June 2025.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22428–22437, June 2023.

Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18381–18391, June 2023.

Lingfeng Yang, Xinyu Zhang, Xiang Li, Jinwen Chen, Kun Yao, Gang Zhang, Lingqiao Liu, Jingdong Wang, and Jian Yang. Add-sd: Rational generation without manual reference. *arXiv preprint arXiv*, 2024.

Chun-Hsiao Yeh, Yilin Wang, Nanxuan Zhao, Richard Zhang, Yuheng Li, Yi Ma, and Krishna Kumar Singh. Beyond simple edits: X-planner for complex instruction-based image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11991–11999, 2026.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020. URL https://arxiv.org/abs/1805.04687.

Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. ObjectMover: Generative object movement with video prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17682–17691, 2025a.

Yongsheng Yu, Ziyun Zeng, Haitian Zheng, and Jiebo Luo. OmniPaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17324–17334, 2025b.

Jooyeol Yun, Davide Abati, Mohamed Omran, Jaegul Choo, Amirhossein Habibian, and Auke Wiggers. Imagining the unseen: Generative location modeling for object placement, October 2024. URL https://arxiv.org/abs/2410.13564. Accepted by the ICCV 2025 DRL4Real Workshop.

Z.ai. GLM-Image: Auto-regressive for dense-knowledge and high-fidelity image generation. Technical blog, January 2026. URL https://z.ai/blog/glm-image. Accessed: 2026-09-16.

Yuhong Zhang, Han Wang, Yiwen Wang, Rong Xie, and Li Song. Freeinsert: Personalized object insertion with geometric and style control, 2025a. URL https://arxiv.org/abs/2509.20756.

Ziyue Zhang, Mingbao Lin, Quanjian Song, Yuxin Zhang, and Rongrong Ji. Objectadd: Adding objects into image via a training-free diffusion modification fashion, 2025b. URL https://arxiv.org/abs/2404.17230.

Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, Weiming Zhang, and Nenghai Yu. X-paste: Revisiting scalable copy-paste for instance segmentation using CLIP and StableDiffusion. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42098–42109. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhao23f.html.

Jixin Zhao, Zhouxia Wang, Peiqing Yang, and Shangchen Zhou. Precise object and effect removal with adaptive target-aware attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19370–19380, 2026.

Lirui Zhao, Tianshuo Yang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Rongrong Ji. Diffree: Text-guided shape free object inpainting with diffusion model. *arXiv preprint arXiv:2407.16982*, 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 (CVPR)*, pp. 22490–22499, June 2023.

Jingyuan Zhu, Shiyu Li, Yuxuan Liu, Jian Yuan, Ping Huang, Jiulong Shan, and Huimin Ma. Odgen: Domain-specific object detection data generation with diffusion models, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/743771397cef2aa0ef497c428c3a46b7-Paper-Conference.pdf.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting, 2023.

A Appendix: removal dataset preparation

A.1 removal methods comparison

We compared and analysis four different removal methods. First, LaMa employs a single-stage inpainting network with fast Fourier convolutions and a high-receptive-field perceptual loss, providing efficient reconstruction of large masks and strong generalization to high-resolution images, but without explicitly modeling object-associated effects such as shadows and reflections (Suvorov et al., 2022). DesignEdit instead adopts a training-free latent-diffusion framework that decomposes an image into object and background layers, performs background completion through key-masked self-attention, and suppresses artifacts during latent fusion; it supports diverse spatial edits but is less specialized for precise object-effect removal (Jia et al., 2025). OmniPaint jointly formulates insertion and removal using a pretrained diffusion prior and a progressive training strategy that combines paired supervision with large-scale unpaired refinement through CycleFlow, yielding improved foreground elimination and scene consistency at the cost of iterative diffusion inference (Yu et al., 2025b). ObjectClear (Zhao et al., 2026) explicitly decouples foreground removal from background reconstruction through adaptive target-aware attention and further protects unmasked content using attention-guided fusion (Zhao et al., 2026). Consequently, LaMa offers the best efficiency and resolution robustness, DesignEdit provides training-free editing flexibility, OmniPaint balances insertion and removal within a unified model, whereas ObjectClear achieves the strongest reported object-effect removal and background fidelity, particularly in scenes containing shadows, reflections, or multiple interacting objects. The final visualization results are shown in Fig. 8.

A.2 implement details

UAVDT (Du et al., 2018). We split the 50 UAVDT sequences into training and test sets at a 4:1 ratio. The test set comprised the following 10 sequences: m0101, m0202, m0207, m0208, m0209, m0210, m0704, m0901, m1102, and m1202. We use UAVDT frames and GT bbox annotations to construct the datasets. Cars, trucks, and buses are eligible for removal if their bounding boxes contain at least 8^2 pixels, do not touch image boundaries, and have zero overlap with other instances; all ineligible instances are retained as context. We sample the removal count k from $\{1, 2, 3, 4, 6, 8, 12, 16, 18\}$ with probabilities $\{0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.1\}$ and set

$$k \leftarrow \min\left(k, 18, \lfloor 0.5 n_{\text{valid}} \rfloor\right).$$

VisDrone (Zhu et al., 2021). We process the VisDrone (Zhu et al., 2021) validation split, with support for the train and test-dev splits. Categories 1–10 are removable, whereas ignored instances, non-positive-score boxes, and categories 0 and 11 are excluded from target selection but retained as context when applicable. Candidates must contain at least 64 pixels, be non-truncated, and have zero overlap with other boxes. We sample k from $\{1, 2, 3, 4, 6, 8, 12, 16, 18, 24\}$ with probabilities $\{0.09, 0.09, 0.09, 0.09, 0.09, 0.18, 0.09, 0.09, 0.09, 0.10\}$ and clip it by

$$k \leftarrow \min\left(k, \begin{cases} 24, & n_{\text{target}} \geq 38, \\ 18, & \text{otherwise}, \end{cases} \lfloor 0.5 n_{\text{target}} \rfloor\right).$$

Targets are selected using dataset-level category deficits and within-frame category round-robin sampling to improve class balance.

BDD100K (Yu et al., 2020). Only rectangular instances from nine classes—person, rider, bike, car, truck, bus, motor, traffic light, and traffic sign—are eligible for removal; lanes, drivable areas, trains, non-rectangular objects, and out-of-vocabulary categories are used only as context. Candidates must contain at least 64 pixels, be non-truncated and boundary-separated, and have a maximum IoU below 0.2 with other training instances. We sample k from $\{1, 2, 3, 4, 6, 8, 10, 12\}$ with probabilities $\{0.12, 0.12, 0.12, 0.12, 0.14, 0.16, 0.10, 0.12\}$ and set

$$k \leftarrow \min\left(k, 12, \lfloor 0.5 n_{\text{target}} \rfloor, n_{\text{target}}\right).$$

To mitigate class imbalance, target sampling allocates 65% of samples to major classes (car, traffic sign, traffic light, person, and truck) and 35% to tail classes (bus, bike, rider, and motor), followed by category balancing within each group.



Figure 8: Visualization of Removal Dataset Processed images. The left figure is the original image with target removed bbox and corresponding class. while the right is processed image.

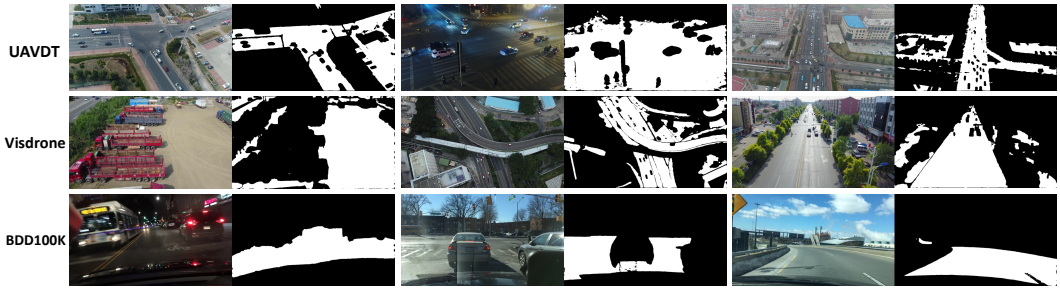

Figure 9: Visualization of plausible region mask among three datasets generated by SAM3. The black and white image on the right of each RGB image is the binary map of the result segmentation.

B Appendix: Plausible region mask construction

All datasets use SAM3 (Carion et al., 2025) masks with a four-pixel dilation, no expansion of protected regions, FP16 inference, and quality filtering with an edge-density threshold of 0.05 and an image-difference threshold of 15.0. UAVDT (Du et al., 2018) and VisDrone (Zhu et al., 2021) use at most five batches per sample and enable a second removal pass, with at most four and five boxes per group, respectively; BDD100K(Yu et al., 2020) uses four batches, three boxes per group, and a single pass. We present visualization of image and corresponding binary mask in Fig. 9.

VisDrone. We use the SAM3 (Carion et al., 2025) prompt configuration to separately segment roadway, pedestrian/footway, and open-ground support regions. Generic terms such as *pavement*, *ground*, *street*, and *concrete* are omitted to avoid merging roads with sidewalks. Four-wheel vehicles are constrained to roadway masks, light vehicles to roadway and footway masks excluding pedestrian-open regions, and pedestrians to the union of all support channels.

BDD100K. We generate a single-channel road-support mask using the prompt set: {*road*, *road surface*, *drivable road*, *traffic lane*, *street*, *intersection*, *parking lot*, *road shoulder*, *driveway*, *crosswalk on road*}. Masks are generated at resolution 1008, thresholded at 0.4; traffic signs and lights are excluded from ground-based placement.

Table 5: Ablation study of the ScenePlan size predictor. We evaluate alternative global-feature and relative-feature designs, MLP head architectures, and training settings using class-balanced log-area and aspect-ratio L1 errors. The average is the arithmetic mean of the two errors.

Group	ID	Class-balanced log-area L1 ↓	Class-balanced aspect-ratio L1 ↓	Average ↓	Description
Ours		0.4651	0.2730	0.3691	Selected configuration
Global feature	GF0	0.5057	0.2732	0.3895	All global features
	GF1	0.4621	0.2933	0.3777	Distance feature
	GF2	0.4774	0.2876	0.3825	Median-area feature
Relative feature	RF0	0.5149	0.3550	0.4350	All relative features
	RF1	0.4834	0.2715	0.3775	Class embedding
	RF2	0.4853	0.3170	0.4012	Relative geometry
MLP head	HM-d1	0.4759	0.3165	0.3962	One hidden layer
	HM-d4	0.4877	0.3819	0.4348	Four hidden layers
	HM-w128	0.4913	0.2719	0.3816	Hidden width of 128
	HM-w512	0.5014	0.5018	0.5016	Hidden width of 512
Training settings	TS-lr↓	0.4626	0.2799	0.3713	$\mathrm{lr} = 3 \times 10^{-4}$
	TS-lr↑	0.4612	0.3403	0.4008	$\mathrm{lr} = 3 \times 10^{-3}$

UAVDT. For each UAVDT frame, we generate a vehicle-support mask using the SAM3 prompts {*roads*, *street*, *parking areas*}. Inference uses a confidence threshold of 0.4, skips completed outputs, and stores one binary mask per frame.

C Appendix: Details of ScenePlan

C.1 Pre-Trained Size Predictor

Ablations. As shown in Tab.5, we evaluate the ScenePlan size predictor on 100 images from the UAVDT (Du et al., 2018) removal test set, sampling 10 images from each sequence. For each removed object, the center coordinates (x, y) of its ground-truth bounding box are provided as input, and the predictor estimates its width and height (w, h). We compute class-balanced L1 errors between the predicted and ground-truth sizes in the log-area and aspect-ratio spaces. As shown in Table 5, the complete model achieves an average error of 0.3691. Removing the entire global-feature branch (GF0) increases the error to 0.3895, while removing the box-distance cue (GF1) or the median area of neighboring context boxes (GF2) results in errors of 0.3777 and 0.3825, respectively. The relative-feature branch has a greater effect: removing the entire branch (RF0) increases the error to 0.4350, while removing the context-class embeddings (RF1) or relative-position features (RF2) yields 0.3775 and 0.4012, respectively. These results indicate that relative geometry is particularly important for scene-conditioned size estimation. For the prediction head, our two-layer architecture ($352 \rightarrow 256 \rightarrow 256 \rightarrow 2$) outperforms the shallower, deeper, narrower, and wider alternatives, suggesting that moderate model capacity provides the best trade-off for this task. Finally, reducing the learning rate from 10^{-3} to 3×10^{-4} produces a comparable error of 0.3713, whereas increasing it to 3×10^{-3} degrades the error to 0.4008, supporting the use of 10^{-3} as the default learning rate.

C.2 Box Planner Post-Process

The module repairs each box onto a feasible set defined by image bounds, scene-valid masks, and existing object layouts, while preserving the planner's semantic class assignment and insertion count. When a predicted box violates feasibility, we repair it using constrained translations and, when necessary, mild scale or size re-estimation, yielding final boxes that are both spatially valid and less likely to collide with observed objects.

For VisDrone(Zhu et al., 2021) and UAVDT(Du et al., 2018) aerial datasets, we first clamps each raw box to image bounds and size limits, and then applies an iterative procedure against the union plausible region mask mentioned above, context boxes, and other target boxes; this procedure allows

bounded mask shifts, context/target separation shifts, and small scaling when overlap or invalid-mask coverage remains. For BDD100K(Yu et al., 2020), we use a rule-based repair tailored to street-view geometry: if a predicted insertion overlaps an existing vehicle, we translate it only along the horizontal axis until the collision is removed, keeping its predicted size fixed; otherwise, we snap its center vertically into the road/ground valid mask, re-estimate its size with the frozen size predictor, and optionally match its area to the nearest upper vehicle in the same horizontal direction. Thus, both pipelines enforce validity after layout prediction and before inpainting, but BDD100K(Yu et al., 2020) uses dataset-specific street-scene heuristics with size predictor.

D Appendix: Details of SceneAlign-Fill

To equip FLUX.1-Fill (Black Forest Labs, 2024) with instance-level layout control and source-scene appearance awareness, we introduce three complementary conditioning signals: dense geometry features, region-level box tokens, and a context-derived appearance code.

Dense geometry conditioning. For each packed image token at normalized coordinate $\boldsymbol{\xi}_n$, we identify its nearest valid insertion box $\hat{b}_i^*$ and encode the token's relative position and box scale as

$$\mathbf{g}_n = \left[(\boldsymbol{\xi}_n - \boldsymbol{\mu}_{\hat{b}_i^*})^\top, \log \max(w_{\hat{b}_i^*}, \delta), \log \max(h_{\hat{b}_i^*}, \delta) \right]^\top . \tag{8}$$

Here, $i = i(n)$ indexes the nearest valid box, $\boldsymbol{\mu}_{\hat{b}_i^*}$, $w_{\hat{b}_i^*}$, and $h_{\hat{b}_i^*}$ denote its normalized center, width, and height, respectively, and $\delta > 0$ ensures numerical stability. The resulting dense geometry field provides each image token with explicit spatial information relative to the planned layout.

Box-token regional attention. For each valid insertion box, we combine its normalized geometry with a frozen CLIP class embedding to construct a box token. These tokens are appended to the global text-conditioning sequence. In the final four FLUX blocks, an image token can attend to a box token only when its spatial location lies inside the corresponding box, whereas the global text tokens remain accessible to all image regions. This region-restricted attention associates category semantics with their designated insertion regions and reduces interference between instances.

Context-derived appearance conditioning. To align the inserted objects with the source scene, we extract color statistics and ROI-pooled VAE features exclusively from existing context objects. The two descriptors are projected, fused through layer normalization, and modulated by a learned scalar gate to produce a global appearance code τ_s. When no valid context object is available, a learned null-style embedding is used instead. The appearance is then broadcast to all image-token positions.

Finally, the native FLUX.1-Fill condition is augmented with the dense geometry and appearance features:

$$C_p = \left[z_{\mathrm{cond},p}; \tau_{m_p}; \tau_g; \tau_s \right], \tag{9}$$

Together, these conditions provide instance-level geometric control, region-specific category guidance, and scene-consistent appearance conditioning for multi-object insertion.

Training and inference. We train SceneAlign-Fill using rectified flow matching. Given the target-image latent z_0, we sample a timestep $t \sim \mathcal{U}(0,1)$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ of the same dimensionality, and construct

$$z_t = (1-t)z_0 + t\epsilon, \qquad v^\star = \epsilon - z_0, \tag{10}$$

where z_t is the interpolated noisy latent and $v^\star$ is its target velocity. The model predicts the conditional velocity field as

$$\hat{v}_\theta = F_\theta \left(z_t, C, \mathcal{U}_{\mathrm{box}}, y, t \right), \tag{11}$$

where F_θ is the conditional velocity predictor, $C = \{C_p\}_{p=1}^N$ denotes the image-side conditioning tokens, $\mathcal{U}_{\mathrm{box}} = \{u_i\}_{i=1}^{N_{\mathrm{tgt}}}$ is the set of box tokens, and y is the global text condition.

To emphasize annotation-aligned synthesis while preserving the source scene, we use the spatially weighted flow-matching objective

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\epsilon} \left[\frac{1}{N} \sum_{p=1}^{N} \omega_p \left\| \hat{v}_{\theta,p} - v_p^\star \right\|_2^2 \right], \qquad \omega_p = \begin{cases} \lambda_{\mathrm{box}}, & p \in \bigcup_i b_i, \\ 1, & \text{otherwise}, \end{cases} \tag{12}$$

where N is the number of packed latent tokens, p indexes their spatial positions, and b_i denotes the region covered by the i-th target box. The weight $\lambda_{\text{box}} > 1$ emphasizes velocity prediction within the annotated regions, while the unit weight elsewhere penalizes unintended changes to the preserved scene.

We freeze the VAE, text encoders, and pretrained FLUX.1-Fill backbone weights, and optimize only the newly introduced conditioning modules, expanded input projection, and LoRA adapters. At inference time, predicted or ground-truth layouts are converted into the same image-side conditions and box tokens. The conditional rectified-flow trajectory is then integrated from Gaussian noise, and the resulting latent is decoded by the frozen VAE to produce the output image.

E Model training and Evaluation

We train a separate model for each dataset using the same two-stage pipeline. We first train a size predictor G_ϕ to estimate box width and height from a target center and its scene context. After freezing G_ϕ, we train ScenePlan to jointly predict the centers of the requested instances. The predicted centers and sizes are combined into complete boxes and adjusted by the post-processing procedure. The resulting layout is then passed to SceneAlign-Fill for image synthesis.

E.1 Training Detail Settings

Pre-Trained Size predictor. All size predictors are trained independently for 50 epochs using AdamW with weight decay 10^{-4} and an effective batch size of 64. They are subsequently frozen and incorporated into ScenePlan as geometric conditions. UAVDT and BDD100K use a learning rate of 10^{-3}, 500 warm-up steps, and gradient clipping at 1.0. For Visdrone, We use a $352 \to 512 \to 512 \to 2$ MLP with a learning rate of 3×10^{-3}. The training, validation, and test-dev removal splits are used for training, model selection, and final evaluation, respectively.For UAVDT, the predictor directly regresses log width and height. To mitigate class imbalance, the losses for cars, trucks, and buses are weighted by 0.4, 0.4, and 0.2, respectively. For BDD100K, we use a $352 \to 256 \to 256 \to 2$ MLP trained on the training removal set, with the validation removal set used for model selection. DINOv2-Large features are limited to 256 tokens and 128 context boxes; center and validation-box jitter are set to 0.03 and 0.05, respectively, and the loss is macro-averaged across classes.

ScenePlan training. All planner variants use DINOv2-Large as the image encoder. The planner contains four Transformer layers with a hidden dimension of 512 and eight attention heads. It supports at most 18 target queries and 128 context boxes, and is trained using a 100-step diffusion process with velocity prediction.

For UAVDT, we use a learning rate of 5×10^{-5}, weight decay 0.01, and a batch size of 64; the selected checkpoint is obtained at step $20{,}000$. Both aerial planners use precomputed SAM3 support-region masks. The masks are encoded into valid-region summary tokens and used to softly modulate image attention. A center-validity loss with weight 0.5 further encourages planned boxes to lie in plausible regions. For VisDrone2019, we train the planner for $20{,}000$ optimization steps using AdamW with a learning rate of 2×10^{-5} and an effective batch size of 64.

The BDD100K planner is trained without scene-validity masks, mask-guided attention, or the center-validity loss. Its active insertion categories are *person*, *rider*, *bike*, *car*, *truck*, *bus*, *motor*, *traffic light*, and *traffic sign*; *lane* and *drivable area* are excluded. Although the model reserves 11 class slots, only these nine object categories are used as insertion targets.

SceneAlign-Fill training. All dataset-specific models are initialized from FLUX.1-Fill-dev, with the pretrained VAE, text encoders, and base-model parameters frozen. We optimize only the LoRA adapters, expanded image-input projection, and new conditioning modules at a resolution of 768, using a per-device batch size of 1, four gradient-accumulation steps, `bfloat16` precision, and seed 2024. Training is limited to $50{,}000$ steps with an early-stopping patience of $5{,}000$ steps. For UAVDT, we use learning rates of 5×10^{-5} and 10^{-4} for the conditioning modules and LoRA adapters, respectively, and select the checkpoint at step $11{,}000$. Class-balanced sampling is enabled, with insertion classes weighted by the square root of inverse class frequency. For Visdrone, the

conditioning modules and LoRA adapters use learning rates of 10^{-5} and 3×10^{-5}, respectively. The selected checkpoint is obtained at step $4{,}000$. For BDD100K, the conditioning modules and LoRA adapters use learning rates of 5×10^{-5} and 10^{-4}, respectively; the LoRA adapters have rank 16, scaling factor 16, and dropout 0.05.

E.2 Details of Baseline Model Fine-tuned

For a controlled comparison, we adapt Add-it and Qwen separately to UAVDT and VisDrone using the same object-removed original image pairs and insertion prompt template as our method. Training uses 32,297 pairs from the designated 40 UAVDT training sequences and 6,404 pairs from the official VisDrone training split. We freeze the FLUX.1-dev backbone and train dataset-specific LoRA adapters on its attention projections, with rank and scaling factor both set to 16. Images are resized while preserving their aspect ratios and padded to 1024×1024. Each adapter is trained for 2,000 optimization steps using AdamW, a learning rate of 5×10^{-5}, weight decay of 0.01, 100 warm-up steps, and an effective batch size of 4, with BF16 precision on a single NVIDIA H200 GPU. We use an object-region-weighted flow-matching loss ($\lambda_{\text{obj}} = 10$) together with an L1 background-preservation loss ($\lambda_{\text{bg}} = 1$), deriving supervision masks from the removed-object bounding boxes. The final-step adapters are used for evaluation without test-set-based checkpoint selection.

E.3 Appendix: Evaluation Methods

E.3.1 Dataset-Specific Insertion Test Schedule.

All three schedules combine equal quotas, category balancing, and GT-admissible or scene-consistency constraints to produce reproducible and auditable insertion requests. By reducing the influence of natural category frequencies and scene densities, these schedules more directly evaluate the planner's controllability over object count, category composition, placement, and scale.

UAVDT. For UAVDT, test requests are constructed at the scene level using three vehicle classes: `car`, `truck`, and `bus`. The Common Track assigns equal quotas to $N \in \{1, \ldots, 5\}$. For $N = 1$, only single-class requests are used; for $N \geq 2$, the schedule contains 60% single-class and 40% mixed-class requests. The three classes are balanced symmetrically across both request types, preventing the naturally dominant `car` class from suppressing the evaluation of `truck` and `bus`.

Visdrone. We evaluate Visdrone on the official test-dev split, covering 10 active categories of pedestrians, vehicles, and lightweight transportation objects. The Common Track assigns equal quotas to $N \in \{1, \ldots, 5\}$. Each image may request only categories present in its original ground-truth annotations, which we refer to as the *GT-admissibility* constraint. Subject to this constraint, we maintain global category balance and a target ratio of 60% single-class to 40% mixed-class requests whenever feasible. This design accommodates the dataset's diverse, long-tailed categories and varying scene densities while avoiding semantically implausible insertions.

BDD100K. For BDD100K, we sample a fixed set of 1,000 independent images from the official validation split and allocate 200 images to each $N \in \{1, \ldots, 5\}$. We use the same nine detection classes as in training: `person`, `rider`, `bike`, `car`, `truck`, `bus`, `motor`, `traffic light`, and `traffic sign`. Non-generative categories, such as `train`, `drivable area`, and `lane`, are explicitly excluded. Requests are restricted to categories visible in the original ground truth. For $N = 1$, admissible categories are sampled uniformly; for $N \geq 2$, we target a 60% single-class and 40% mixed-class ratio and apply global rebalancing to increase the exposure of long-tailed categories.

E.3.2 Detect Added Objects

We fine-tune GroundingDINO with a Swin-B backbone, using a batch size of 2 for 12 epochs across three datasets. To identify inserted objects, we apply the trained detector to both the source and generated images, using the concatenated dataset class names as the caption. Same-class detections with an IoU of at least 0.50 are greedily matched, and unmatched detections in the generated image are treated as inserted-object boxes. We use score thresholds of 0.25, followed by class-wise NMS with an IoU threshold of 0.70. As visualization shown in Fig.10, green boxes denote matched objects, red boxes denote unmatched source objects treated as unpreserved, and blue boxes denote

Table 6: Evaluation results of GroundingDINO on BDD100K, UAVDT and Visdrone test dataset

Dataset	Recall	mAP50	mAP50–95
BDD100K (Yu et al., 2020)	94.91	65.54	38.04
UAVDT (Du et al., 2018)	96.64	91.71	70.75
VisDrone (Zhu et al., 2021)	78.88	48.29	28.59



Figure 10: Visualization of inserted object detected by GroundingDINO. The figure in the left in original image and right is the generated. Green, red, and blue boxes indicate matched, unpreserved, and newly inserted objects, respectively; inserted-object metrics are computed on the blue boxes.

newly inserted objects. All subsequent inserted-object metrics are computed using the blue boxes. We present the accuracy of our trained GroundingDINO in Tab.6.

E.3.3 Metrics Calculation Details

Request Fidelity We calculate Mean Total Count Error (mTCE) and Mean Class Count Error (mCCE):

$$\mathrm{mTCE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{T}_i - T_i\right|, \qquad \mathrm{mCCE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{c\in\mathcal{C}}|\hat{n}_{i,c} - n_{i,c}|}{\sum_{c\in\mathcal{C}} n_{i,c}}.$$

Here, T_i and $\hat{T}_i$ denote the requested and detected total counts, while $n_{i,c}$ and $\hat{n}_{i,c}$ denote their category-wise counterparts; lower values indicate better request fidelity.

F Datasets Augmentation for Detection

For Copy–Paste, we build a bank of ground-truth `truck` and `bus` crops with a minimum side length of 8 pixels, a boundary margin of at least 2 pixels, and an aspect ratio in $[0.3, 4.0]$. We augment 40% of frames sampled per sequence (seed 0) by pasting two same-class instances, avoiding same-frame sources whenever possible. Each crop is scaled to the median vehicle height with jitter $[0.85, 1.15]$, horizontal flipping ($p = 0.5$), and brightness adjustment ($p = 0.3$), and is placed in the lower image region with IoU < 0.10 within 50 attempts. For X-Paste, we construct an instance bank with up to 512 instances per class, using CLIP and segmentation thresholds of 0.28 and 0.45, respectively. We retain masks with fill ratios between 0.20 and 0.85 and restrict the bounding-box area ratio to at most 0.55. During placement, we use an IoU threshold of 0.10, apply scale jittering between 0.85 and



Figure 11: Qualitative results for the two-bus and two-truck insertion settings on UAVDT.

1.15, and perform horizontal flipping and brightness adjustment with probabilities of 0.5 and 0.3, respectively. For our method, the 40% subset is sampled by object count: 50%, 25%, and 15% are assigned to images with zero, one, and multiple objects, respectively, while the remaining 10% is reserved for hard examples with high object density, small objects, or objects near image boundaries. Some visualizations of generated images for datasets augmentation are shown is Fig.11.