LCD^3: Layout-Conditioned Diffusion for Dataset Distillation in Object Detection

Yue Cao[1] **Mohsen Zardadi**[2] **Yu Hu**[1] **Yanshuo Fan**[1]
Jozsef Hamari[2] **Zheng Liu**[1†] **Jianyang Gu**[3†]

[1]The University of British Columbia, Canada [2]TerraSense Analytics, Canada
[3]The Ohio State University, USA

{yue.cao, yu.hu, yanshuo.fan, zheng.liu}@ubc.ca
{mohsen.zardadi, jozsef.hamari}@terrasense.ca, gu.1220@osu.edu

Abstract

Object detection data contains a structural asymmetry that is easy to overlook in dataset distillation (DD). Each detection image couples a spatial layout with a visual realization. Layouts define detection targets through object categories and bounding boxes, while pixels instantiate these targets with particular object appearances and backgrounds. Although both forms can be redundant across a dataset, they play different roles. Layouts define the task distribution, so synthesizing them freely risks altering the detection problem. Visual realizations, on the other hand, are conditional samples attached to layouts. This motivates a separation principle: preserve and recombine layouts from the original data, then use generative priors to diversify their visual realizations. Based on this principle, we propose LCD^3, a layout-conditioned diffusion framework for object detection DD. LCD^3 constructs composite layouts from diverse groups based on scene-level embeddings. A layout-conditioned diffusion model then generates new images from these layouts, enriching object appearance and scene realization without discarding the spatial structure needed for detection. The generation process is grounded with semantically verified object crops to retain the original visual style. Experiments show that this separation between layout and realization produces more effective distilled detection datasets. LCD^3 consistently outperforms the previous state-of-the-art, OD^3, across the MS COCO and PASCAL VOC benchmarks using fewer objects per distilled image, *e.g.*, a 12.5% mAP50 improvement at a 0.5% compression ratio on PASCAL VOC.

1 Introduction

Object detection requires models to learn object appearance together with spatial localization [1]. Training modern detectors on large-scale datasets is expensive in both computation and storage [2–4]. Dataset distillation (DD) aims to reduce this cost by condensing a large training set into a smaller set of informative samples. Although DD has made substantial progress in image classification [5–9], object detection remains more challenging [10–12]. Classification images are often centered around a dominant object and supervised by image-level labels. In contrast, detection images couple visual appearances with spatial layouts. As a result, distilled detection samples must preserve meaningful object placement rather than only emphasize class-discriminative visual patterns.

In this work, we examine object detection datasets through this lens and find recurring redundancy both within and across images. As shown in Figure 1a, multiple instances in the same image may

†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) (b)

Figure 1: (a) Real images in COCO have redundancy both within an image and across images (top row), limiting the information density for DD. Diffusion models can generate images that preserve plausible layout structure while introducing visual variety in object appearance, background, and viewpoint (bottom row). (b) Generated images significantly increase diversity across classes.

look highly similar, and different images may share comparable backgrounds and scene layouts. When dataset distillation aims to reduce redundancy for detection, however, the asymmetrical roles of spatial layout and visual realization are easily overlooked. The layout, defined by object categories and bounding boxes, specifies the detection targets. The pixels instantiate these targets through a particular appearance and scene context. Layouts define the task distribution, so synthesizing them freely may alter the detection problem. In comparison, visual realizations are conditional samples attached to layouts, making them a more natural source of diversity.

This motivates a separation in DD between the spatial layout of a detection image and its particular visual content. If original images are condensed as fixed pixel samples, useful detection structure remains tied to redundant visual realization. In Figure 1b, we quantitatively show that original images in the MS COCO dataset [13] contain substantial information overlap among instances of the same class. Intriguingly, conditioning a diffusion model [14] on the original layout can produce instances with more distinctive appearances. This suggests that the generative prior learned from large-scale pre-training can enrich visual realization while respecting detection structure. We therefore seek to preserve useful layout structure and use generative models to synthesize more diverse visual content.

Based on this insight, we propose a layout-conditioned diffusion framework, named LCD^3, for object detection DD. Given object categories and bounding boxes, a diffusion model synthesizes images that follow the specific layouts while varying visual appearance. The resulting distilled dataset keeps the spatial supervision needed for detector training and reduces redundancy in the visual content. To improve layout coverage under a compact image budget, we construct scene-based composite layouts. We group original images using scene-level visual embeddings and assemble each synthetic layout from different scene groups. This design allows one distilled image to inherit layout structure from multiple source images while covering diverse scene types. The final image is rendered by diffusion, which turns the composite layout into a coherent visual sample. The generation process is further grounded with real object instances from the original dataset. For each object, we use a semantic consistency check to identify reliable crops. Verified crops serve as appearance references during generation, helping generated objects remain aligned with the original category distribution.

This formulation gives a clear distinction from existing dataset distillation methods [10, 12, 15]. Methods that select or reuse real images preserve the fidelity of original data, but they may also retain its redundancy. Methods based on direct region composition increase information density per image but disrupt the local spatial context that is necessary for detection training. In contrast, our approach is motivated by the empirical structure of detection data itself, which preserves the compact task-relevant layout and regenerates the redundant visual realization.

Empirically, LCD^3 produces more effective compact detection datasets. It outperforms the prior SOTA OD^3 by 5.7% mAP$_{50}$ on MS COCO [13] and by 12.5% on PASCAL VOC [16, 17], using much fewer objects per distilled image. These results support the view that detection dataset distillation benefits from moving beyond image-level compression toward layout-preserving visual regeneration. By separating layout from visual contents, our framework preserves reliable spatial supervision while leveraging generative priors to produce more informative training images.

2 Related Work

Dataset distillation aims to synthesize a compact dataset that preserves the training effectiveness of a much larger one [18, 19]. Most existing methods focus on image classification, through objectives such as gradient matching [20, 21], distribution matching [7, 22, 23, 22, 24], trajectory matching [8], or decoupled approaches that leverage pretrained models [15, 25, 9, 26, 27]. Extending dataset distillation to object detection is more challenging, as each image contains multiple spatially annotated objects and bi-level optimization becomes prohibitive at detection-scale resolutions. DCOD [11] decouples the process into a Fetch stage that stores detection knowledge in model parameters, followed by a Forge stage that reconstructs images via model inversion. UniDD [12] proposes a universal framework spanning classification, detection, and segmentation by training a task-specific proxy to guide diffusion synthesis. OD^3 [10] densely copy-pastes real object crops onto a canvas and uses an observer model to filter low-confidence candidates. Despite their progress, these methods treat each detection image as a monolithic sample, without leveraging the distinct roles of spatial layout and visual content. OD^3 increases object density through dense copy-paste, but this disrupts the local spatial context of each object, since crops from unrelated images are overlaid with mismatched scales and lighting, and retains the visual redundancy of the original crops. Our work addresses this by preserving and recombining original layouts while using layout-conditioned diffusion to regenerate diversified visual realizations that maintain coherent spatial relationships within each quadrant.

Layout-conditioned generation aims to synthesize images where the spatial arrangement of objects is specified by bounding boxes or segmentation masks. Earlier GAN-based approaches [28–30] explored generating multi-object scenes from layout or scene graph inputs but were limited in image quality and diversity. With the advent of diffusion models, controllable generation has advanced significantly. ControlNet [31] injects spatial conditions such as edges, depth, or segmentation maps into pretrained diffusion models. GLIGEN [14] enables bounding box-conditioned generation through gated self-attention layers. InstanceDiffusion [32] further improves instance-level control with per-object attribute specification. MS-Diffusion [33] builds on SDXL to achieve higher-resolution layout-conditioned generation with image reference support. These advances in controllable generation provide effective tools for regenerating visual realizations from preserved detection layouts.

3 Methodology

3.1 Preliminary

The objective of dataset distillation for object detection is to compress a training set $\mathcal{T} = \{(\boldsymbol{x}_i, \mathcal{A}_i)\}_{i=1}^{N}$ of N images, where $\boldsymbol{x}_i$ is an image and $\mathcal{A}_i$ is its associated set of bounding box and class label annotations, into a compact synthetic dataset $\mathcal{S}$ with $N_{\mathcal{S}} \ll N$ images [10, 11]. The goal is that a detector trained on $\mathcal{S}$ achieves comparable performance to one trained on $\mathcal{T}$. Following prior work [10], we define the compression ratio as IPD (Images Per Dataset), *i.e.*, $N_{\mathcal{S}}/N$. Formally, let f_θ denote a detector with parameters θ, and let $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{T}}$ denote the parameters obtained by training on $\mathcal{S}$ and $\mathcal{T}$, respectively. The distillation objective is to construct $\mathcal{S}$ such that:

$$\ell_{\text{det}}(f_{\theta_{\mathcal{S}}}, \mathcal{T}') \approx \ell_{\text{det}}(f_{\theta_{\mathcal{T}}}, \mathcal{T}'), \tag{1}$$

where $\mathcal{T}'$ is the validation set and ℓ_{det} is the detection loss.

As discussed in §1, each detection image can be decomposed into a spatial layout and a visual realization. We define the layout of image $\boldsymbol{x}_i$ as:

$$\mathcal{L}_i = \{(\boldsymbol{b}_{i,k}, c_{i,k})\}_{k=1}^{K_i}, \tag{2}$$

where $\boldsymbol{b}_{i,k} = (x_c, y_c, w, h)$ is the bounding box coordinates and $c_{i,k} \in \{1, \ldots, C\}$ is the class label of the k-th object. The layout $\mathcal{L}_i$ defines the detection targets and their spatial structure. Since layouts specify the task distribution, they should be preserved from the original data. Visual realizations, on the other hand, are conditional on layouts and can be redundant across the dataset, as shown quantitatively in Figure 1b. This asymmetry motivates our approach. As illustrated in Figure 2, LCD^3 first recombines layouts from multiple source images into scene-diverse composite layouts, increasing the detection task coverage per synthetic image. Meanwhile, a crop semantic check verifies each object instance via CLIP to filter unreliable crops. A layout-conditioned diffusion model then renders each composite layout into a visually consistent image, diversifying the visual realization while preserving the intra-quadrant spatial relationships inherited from the source images.

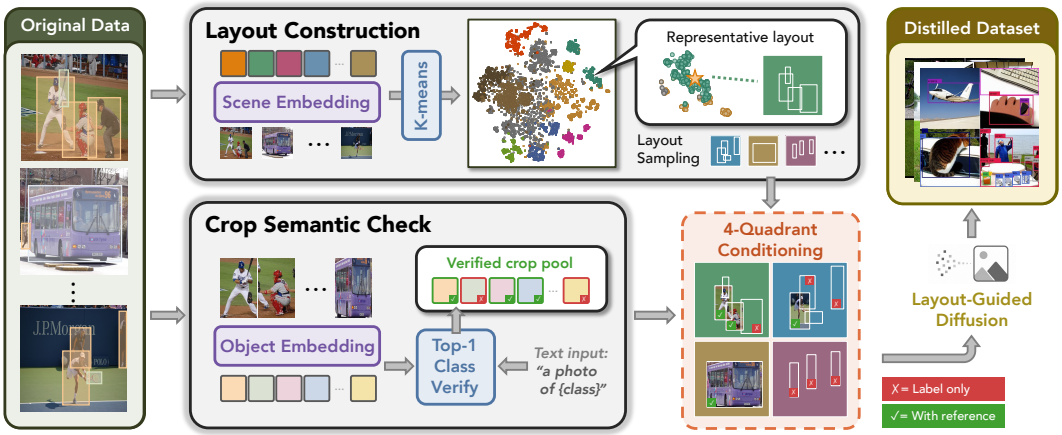


Figure 2: **Overview of the proposed pipeline**. Given original training data, the Layout Construction module extracts scene embeddings and clusters images via K-means to sample representative layouts. The Crop Semantic Check module verifies each object crop through CLIP top-1 class matching to build a clean instance pool. The selected layouts and verified crops are combined into a 4-quadrant conditioning and fed into a layout-guided diffusion model to generate the final distilled data.

3.2 Scene-based layout construction

Since layouts encode the task-relevant detection structure, we preserve them from the original dataset and recombine them to increase layout coverage under a compact image budget. We first partition all training images into scene groups based on their scene-level embeddings. Each image $\boldsymbol{x}_i$ is encoded by a pretrained vision model E into a scene embedding $\boldsymbol{z}_i = E(\boldsymbol{x}_i) \in \mathbb{R}^d$, and we apply K-means clustering [34] on $\{\boldsymbol{z}_i\}_{i=1}^N$ to obtain M clusters $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$, where $M = N_{\mathcal{S}}$ equals the target number of synthetic images. Each cluster groups images that share similar scene characteristics, such as indoor dining scenes or outdoor street views.

Each synthetic image $\hat{\boldsymbol{x}}_j$ is then constructed by dividing the canvas into four quadrants and filling each quadrant with the layout from a source image belonging to a different cluster. The source image's bounding box annotations are rescaled to half of the original size and placed in the corresponding quadrant position. This allows one distilled image to inherit layout structure from four different source images while covering diverse scene types, and the spatial relationships within each quadrant are preserved from the original data.

Selecting which images fill each quadrant requires balancing scene diversity with class coverage. Since the distilled dataset contains far fewer images than the original, it is important to ensure that the class distribution in $\mathcal{S}$ remains as close as possible to that of $\mathcal{T}$, so that no category is severely under-represented in the training data. Within each cluster $\mathcal{G}_m$, we first identify its medoid [35], the image whose scene embedding is closest to all others in the cluster:

$$\boldsymbol{x}_m^* = \arg\min_{\boldsymbol{x}_i \in \mathcal{G}_m} \sum_{\boldsymbol{x}_j \in \mathcal{G}_m} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2. \tag{3}$$

This most representative image of each scene group is assigned to one quadrant slot first, ensuring that the core visual characteristic of each cluster is preserved. The remaining slots are then filled to align the class distribution of $\mathcal{S}$ with $\mathcal{T}$. Let $n_c^{\mathcal{T}}$ and $N^{\mathcal{T}}$ denote the number of instances of class c and the total number of instances in the original dataset, and $n_c^{\mathcal{S}}$ and $N^{\mathcal{S}}$ the corresponding counts accumulated across all currently selected images. For each unfilled slot, we greedily select the image that maximally reduces the total class deficit:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}_i \in \mathcal{G}_m} \sum_{c=1}^{C} n_c(\boldsymbol{x}_i) \cdot \max\left(0,\ \frac{n_c^{\mathcal{T}}}{N^{\mathcal{T}}} - \frac{n_c^{\mathcal{S}}}{N^{\mathcal{S}}}\right), \tag{4}$$

where $n_c(\boldsymbol{x}_i)$ is the number of instances of class c in image $\boldsymbol{x}_i$. The ratio $n_c^{\mathcal{T}}/N^{\mathcal{T}}$ represents the target class proportion, and the $\max$ operator ensures that classes already sufficiently represented

do not contribute to the score. Once the class distribution is sufficiently aligned, the remaining slots are filled by random sampling within each cluster. This formulation encourages the distilled dataset to approximate the class distribution of $\mathcal{T}$ while maintaining scene diversity across quadrants. Additional details of the layout construction are provided in §A.

3.3 Instance-Grounded Image Synthesis

Given a composite layout from §3.2, we need to generate a realistic image with objects faithfully placed at their designated locations. Purely layout-conditioned generation can produce coherent scenes but lacks control over individual object appearances. Without explicit guidance, the generated objects may deviate from the visual distribution of the original training data, introducing a domain gap that harms downstream detector training. To mitigate this, we use real crops from the original images as appearance references during generation.

To filter out low-quality crops in the original dataset, such as heavily occluded, truncated, or ambiguous instances, we introduce a semantic consistency filtering step. We encode each crop $o_{i,k}$ and all C class names using a frozen CLIP model [36], with $\phi_v(\cdot)$ and $\phi_t(\cdot)$ denoting the visual and text encoders, respectively. For each crop, we perform a zero-shot classification prediction:

$$p(o_{i,k}) = \arg\max_{c \in \{1,\dots,C\}} \ \mathrm{sim}\left(\phi_v(o_{i,k}),\ \phi_t(c)\right), \tag{5}$$

where $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity. A crop is accepted into a verified reference pool $\mathcal{R}$ only if $p(o_{i,k}) = c_{i,k}$, where $c_{i,k}$ is the ground-truth class label. This top-1 matching criterion is more robust than a fixed cosine similarity threshold, as raw similarity magnitudes vary substantially across classes. We include the analysis on this semantic verification step in §D.

We then feed the assembled layout to a layout-guided diffusion model G. For each bounding box $\hat{\boldsymbol{b}}_k$ in the layout, if its corresponding source crop passes the filtering above (*i.e.*, $o_{i,k} \in \mathcal{R}$), the crop serves as an appearance reference; otherwise, the object is generated using only its class name and bounding box as conditioning. The generation process can be expressed as:

$$\hat{\boldsymbol{x}}_j = G(\mathcal{L}_j, \mathcal{R}_j, \boldsymbol{t}_j), \tag{6}$$

where $\mathcal{L}_j = \{(\hat{\boldsymbol{b}}_k, \hat{c}_k)\}$ is the composite layout containing bounding boxes and class labels, $\mathcal{R}_j = \{o_{i,k} \mid o_{i,k} \in \mathcal{R}\}$ is the set of verified reference crops, and $\boldsymbol{t}_j$ is the text prompt derived from the class names in $\mathcal{L}_j$.

Since diffusion models do not perfectly comply with every spatial constraint, following OD^3 [10], we apply a pretrained object detection observer model D_{obs} (More detailed in §E). We retain all detections whose confidence score exceeds a threshold $\tau_c = 0.2$, yielding the final annotations:

$$\hat{\mathcal{A}}_j = \{(\boldsymbol{b}_l^{\text{det}}, c_l^{\text{det}}) \mid s_l^{\text{det}} \geq \tau_c\}. \tag{7}$$

This ensures that the annotations faithfully reflect the actual visual content of the synthesized images. We provide a sensitivity analysis of τ_c in §C.2.

4 Experiments

4.1 Experimental Setup

Following the literature [10, 12, 11], we evaluate LCD^3 on two detection benchmarks: MS COCO [13] and PASCAL VOC [16, 17]. On COCO, we distill from the train2017 split and evaluate on val2017 using mAP, mAP$_{50}$, and mAP$_{75}$. On VOC, we distill from the combined train-val splits of VOC 2007 and 2012 and evaluate on the VOC 2007 test set using mAP50. We consider compression ratios (IPD) of 0.25%, 0.5%, and 1.0% for COCO, and 0.5%, 1.0%, and 2.0% for VOC.

For image generation, we experiment with two layout-guided diffusion models: GLIGEN based on Stable Diffusion 1.5 [14, 37], generating images at 512×512 resolution, and MS-Diffusion based on SDXL [33, 38], generating at 768×768 resolution. For each model, we report results with two scene embedding variants for clustering: CLIP and DINOv2 [36, 39]. For the semantic consistency filtering, we use CLIP. We use a DINO observer model D_{obs} [2] to perform quality checks. We adopt the same training configuration as OD^3 [10], including the data augmentation pipeline from

Table 1: **Performance comparison on MS COCO**. The compression ratios range from 0.25% to 1.0%. The evaluation model is Faster R-CNN-R50, the soft-label teacher is Faster R-CNN-R101, and the observer model for annotation filtering is a DINO detector. -C and -D denote CLIP and DINOv2 as the scene embedding for clustering, respectively.

Label	Method	IPD = 0.25%			IPD = 0.5%			IPD = 1.0%		
		mAP%	mAP_{50}%	mAP_{75}%	mAP%	mAP_{50}%	mAP_{75}%	mAP%	mAP_{50}%	mAP_{75}%
Hard	Random	$0.5_{\pm 0.1}$	$1.7_{\pm 0.3}$	–	$3.7_{\pm 0.2}$	$10.1_{\pm 0.3}$	–	$7.2_{\pm 0.8}$	$17.3_{\pm 0.9}$	–
	Uniform	$0.8_{\pm 0.2}$	$2.4_{\pm 0.5}$	–	$3.4_{\pm 0.4}$	$9.5_{\pm 0.6}$	–	$7.4_{\pm 0.5}$	$17.6_{\pm 0.5}$	–
	K-Center	$0.4_{\pm 0.1}$	$1.5_{\pm 0.2}$	–	$3.2_{\pm 0.5}$	$9.3_{\pm 0.5}$	–	$6.1_{\pm 0.3}$	$15.4_{\pm 0.6}$	–
	Herding	$0.5_{\pm 0.1}$	$1.8_{\pm 0.4}$	–	$3.5_{\pm 0.3}$	$9.7_{\pm 0.3}$	–	$6.7_{\pm 0.4}$	$16.3_{\pm 0.7}$	–
	DCOD	–	–	–	$3.7_{\pm 0.3}$	$8.4_{\pm 0.2}$	–	$6.1_{\pm 0.4}$	$13.5_{\pm 0.3}$	–
	UniDD	$4.5_{\pm 0.3}$	$10.3_{\pm 0.4}$	–	$7.1_{\pm 0.4}$	$16.9_{\pm 0.3}$	–	$10.8_{\pm 0.4}$	$22.5_{\pm 0.5}$	–
	OD^3	$5.5_{\pm 0.3}$	$12.9_{\pm 0.5}$	$3.9_{\pm 0.2}$	$8.8_{\pm 0.1}$	$19.3_{\pm 0.3}$	$7.0_{\pm 0.3}$	$11.2_{\pm 0.3}$	$23.4_{\pm 0.3}$	$9.6_{\pm 0.3}$
	LCD^3 (GLIGEN-C)	$6.4_{\pm 0.5}$	$13.9_{\pm 0.9}$	$5.2_{\pm 0.4}$	$9.2_{\pm 0.4}$	$18.3_{\pm 0.7}$	$8.3_{\pm 0.4}$	$11.2_{\pm 0.5}$	$20.2_{\pm 0.7}$	$11.1_{\pm 0.5}$
	LCD^3 (GLIGEN-D)	$6.6_{\pm 0.7}$	$14.3_{\pm 1.5}$	$5.5_{\pm 0.6}$	$9.5_{\pm 0.3}$	$19.0_{\pm 0.6}$	$8.6_{\pm 0.4}$	$10.8_{\pm 0.3}$	$20.4_{\pm 0.6}$	$10.2_{\pm 0.4}$
	LCD^3 (MS-Diff-C)	$6.8_{\pm 0.3}$	$15.8_{\pm 0.6}$	$4.9_{\pm 0.3}$	$9.5_{\pm 0.2}$	$20.0_{\pm 0.2}$	$7.8_{\pm 0.3}$	$12.5_{\pm 0.1}$	$\mathbf{24.8}_{\pm 0.3}$	$\mathbf{11.5}_{\pm 0.2}$
	LCD^3 (MS-Diff-D)	$\mathbf{8.7}_{\pm 0.3}$	$\mathbf{16.8}_{\pm 0.6}$	$\mathbf{6.8}_{\pm 0.3}$	$\mathbf{10.4}_{\pm 0.3}$	$\mathbf{20.6}_{\pm 0.4}$	$\mathbf{9.6}_{\pm 0.4}$	$\mathbf{12.6}_{\pm 0.1}$	$24.7_{\pm 0.1}$	$11.3_{\pm 0.2}$
Soft	OD^3	$11.9_{\pm 0.3}$	$23.4_{\pm 0.8}$	$10.8_{\pm 0.3}$	$17.8_{\pm 0.2}$	$33.2_{\pm 0.2}$	$17.1_{\pm 0.1}$	$21.5_{\pm 0.3}$	$38.9_{\pm 0.3}$	$21.3_{\pm 0.3}$
	LCD^3 (GLIGEN-C)	$14.2_{\pm 0.3}$	$26.2_{\pm 0.4}$	$14.0_{\pm 0.3}$	$18.6_{\pm 0.2}$	$32.2_{\pm 0.4}$	$19.0_{\pm 0.2}$	$23.6_{\pm 0.1}$	$38.0_{\pm 0.2}$	$25.0_{\pm 0.2}$
	LCD^3 (GLIGEN-D)	$14.0_{\pm 0.2}$	$25.7_{\pm 0.4}$	$13.7_{\pm 0.2}$	$18.8_{\pm 0.2}$	$32.8_{\pm 0.5}$	$19.1_{\pm 0.3}$	$23.7_{\pm 0.1}$	$38.1_{\pm 0.1}$	$\mathbf{25.2}_{\pm 0.1}$
	LCD^3 (MS-Diff-C)	$14.7_{\pm 0.1}$	$27.5_{\pm 0.2}$	$14.2_{\pm 0.1}$	$19.3_{\pm 0.3}$	$34.1_{\pm 0.4}$	$19.6_{\pm 0.1}$	$23.6_{\pm 0.2}$	$39.7_{\pm 0.2}$	$24.8_{\pm 0.2}$
	LCD^3 (MS-Diff-D)	$\mathbf{16.3}_{\pm 0.2}$	$\mathbf{29.1}_{\pm 0.3}$	$\mathbf{16.5}_{\pm 0.4}$	$\mathbf{21.1}_{\pm 0.3}$	$\mathbf{35.9}_{\pm 0.4}$	$\mathbf{22.1}_{\pm 0.3}$	$\mathbf{23.8}_{\pm 0.2}$	$\mathbf{40.1}_{\pm 0.3}$	$24.9_{\pm 0.1}$
Whole Dataset (Hard)		38.4	59.0	42.0	38.4	59.0	42.0	38.4	59.0	42.0

Table 2: **Performance comparison on PASCAL VOC** (mAP_{50}%). The evaluation model and soft-label teacher are both Faster R-CNN-R50. The observer model for annotation filtering is a DINO detector. -C and -D denote CLIP and DINOv2 as the scene embedding for clustering, respectively.

Label	Method	IPD=0.5%	IPD=1.0%	IPD=2.0%
Hard	Random	$3.1_{\pm 0.4}$	$13.7_{\pm 0.6}$	$34.3_{\pm 0.5}$
	Uniform	$3.4_{\pm 0.3}$	$17.7_{\pm 0.4}$	$36.2_{\pm 0.4}$
	K-Center	$2.1_{\pm 0.3}$	$12.3_{\pm 0.3}$	$29.3_{\pm 0.6}$
	Herding	$2.4_{\pm 0.2}$	$11.9_{\pm 0.5}$	$28.7_{\pm 0.7}$
	UniDD	$22.3_{\pm 0.6}$	$38.9_{\pm 0.7}$	$48.5_{\pm 0.6}$
	OD^3	$21.7_{\pm 0.4}$	$36.2_{\pm 0.4}$	$48.4_{\pm 0.9}$
	LCD^3 (GLIGEN-C)	$31.0_{\pm 2.1}$	$44.0_{\pm 0.8}$	$50.7_{\pm 0.9}$
	LCD^3 (GLIGEN-D)	$34.0_{\pm 1.7}$	$44.1_{\pm 0.7}$	$50.0_{\pm 1.1}$
	LCD^3 (MS-Diff-C)	$\mathbf{34.2}_{\pm 0.6}$	$45.6_{\pm 1.8}$	$\mathbf{55.1}_{\pm 1.4}$
	LCD^3 (MS-Diff-D)	$33.5_{\pm 2.4}$	$\mathbf{46.1}_{\pm 1.7}$	$54.8_{\pm 0.5}$
Soft	OD^3	$54.1_{\pm 0.5}$	$61.0_{\pm 0.3}$	$65.3_{\pm 0.3}$
	LCD^3 (GLIGEN-C)	$59.4_{\pm 1.2}$	$67.8_{\pm 0.3}$	$72.1_{\pm 0.4}$
	LCD^3 (GLIGEN-D)	$59.6_{\pm 1.1}$	$67.3_{\pm 0.2}$	$71.4_{\pm 0.5}$
	LCD^3 (MS-Diff-C)	$61.5_{\pm 1.2}$	$68.2_{\pm 0.8}$	$\mathbf{74.1}_{\pm 0.2}$
	LCD^3 (MS-Diff-D)	$\mathbf{61.6}_{\pm 0.9}$	$\mathbf{68.8}_{\pm 0.7}$	$73.4_{\pm 0.7}$
Whole Dataset (Hard)		80.4		

mmdetection [40] with RandomResize to 1333×800 for COCO and 1000×600 for VOC. We evaluate under both hard-label and soft-label settings, where the soft labels are produced via PKD [41] from a Faster R-CNN-R101 teacher to a Faster R-CNN-R50 student [4]. Note that OD^3 only reports results under the soft-label setting; we additionally include hard-label evaluation for a more comprehensive comparison. We train detectors for 280 epochs under the hard-label setting and 190 epochs under the soft-label setting with SGD, using 2 NVIDIA V100 GPUs. All results are averaged over 3 runs and reported with standard deviations. Since only OD^3 provides publicly available code, we reproduce their results under the same evaluation protocol for a fair comparison.

4.2 Comparison with State-of-the-art Methods

Comparison on MS COCO. Table 1 reports the results on MS COCO [13]. Under the hard-label setting, LCD^3 with MS-Diffusion outperforms all baselines across compression ratios. The advantage is most notable at the lowest compression ratio, where our best variant outperforms OD^3 [10] by

Table 3: **Cross-architecture evaluation on MS COCO** at 1.0% IPD. The distilled dataset is generated once and used to train different detector architectures. All models use ResNet-50 backbone[42].

Target Detector	OD^3			LCD^3 (MS-Diff-C)		
	mAP%	mAP$_{50}$%	mAP$_{75}$%	mAP%	mAP$_{50}$%	mAP$_{75}$%
Faster R-CNN	$11.2_{\pm 0.3}$	$23.4_{\pm 0.3}$	$9.6_{\pm 0.3}$	$\mathbf{12.5}_{\pm 0.1}$	$\mathbf{24.8}_{\pm 0.3}$	$\mathbf{11.5}_{\pm 0.2}$
RetinaNet	$4.3_{\pm 1.8}$	$9.5_{\pm 3.7}$	$3.4_{\pm 1.5}$	$\mathbf{10.0}_{\pm 0.8}$	$\mathbf{18.8}_{\pm 1.5}$	$\mathbf{9.5}_{\pm 0.8}$
FCOS	$8.5_{\pm 0.8}$	$16.5_{\pm 1.3}$	$7.7_{\pm 0.9}$	$\mathbf{13.3}_{\pm 0.1}$	$\mathbf{24.1}_{\pm 0.1}$	$\mathbf{13.0}_{\pm 0.1}$
DINO	$6.3_{\pm 0.2}$	$10.7_{\pm 0.3}$	$6.5_{\pm 0.2}$	$\mathbf{14.4}_{\pm 0.2}$	$\mathbf{23.9}_{\pm 0.2}$	$\mathbf{14.5}_{\pm 0.1}$

3.2% mAP and 3.9% mAP$_{50}$. As the compression ratio increases to 1.0%, the gap narrows to around 1.4% mAP, but our method remains the best. The 1.0% IPD setting provides around 1,200 training samples, making the differences between methods smaller, as all approaches have sufficient data to cover more of the visual distribution. Under the soft-label setting, where knowledge distillation provides richer supervision, our method maintains a clear lead, surpassing OD^3 by 4.4% mAP and 5.7% mAP$_{50}$ at 0.25% IPD. It is noteworthy that the gap under the soft-label setting is larger than under hard labels at the same compression ratio. One possible explanation is that soft labels are generated by a teacher model extracting features on the synthesized images, and since the images generated by LCD^3 are visually closer to real samples, the teacher produces higher-quality feature responses compared to OD^3's copy-paste images, which contain unnatural artifacts that may degrade the teacher's supervision signal.

Comparison on PASCAL VOC. Table 2 presents results on PASCAL VOC [16, 17], where our method shows even larger improvements over the baselines compared to COCO. Under the hard-label setting at 0.5% IPD, LCD^3 outperforms the state-of-the-art baseline OD^3 [10] by 12.5% mAP$_{50}$. We attribute this to the difference in dataset scale: VOC contains approximately 16.5k training images, so 0.5% IPD corresponds to roughly 80 synthetic images, whereas the same ratio on COCO yields around 590 images. With such a small data budget, the preserved layout and enhanced visual diversity of our 4-quadrant composition become even more critical for covering the dataset's structural information. Under the soft-label setting, LCD^3 leads OD^3 by 7.5% mAP$_{50}$ at 0.5% IPD, with the gains remaining consistent as the IPD increases.

Cross-architecture generalization. A practical requirement for distilled datasets is that they should generalize across different detector architectures, rather than being tied to the specific model used during synthesis. Table 3 evaluates this by training four different detectors on the same distilled dataset generated at 1.0% IPD on COCO. The detectors span two-stage (Faster R-CNN) [4], one-stage (RetinaNet, FCOS) [43, 44], and transformer-detector (DINO) [2] architectures.

The result shows that LCD^3 consistently outperforms OD^3 [10] across all four architectures, with especially large gains on RetinaNet, FCOS, and DINO. It is worth noting that OD^3 performs reasonably on Faster R-CNN but degrades significantly on the other three architectures, whereas LCD^3 maintains stable performance across all detectors. This difference can be attributed to a structural distinction in pipeline design: OD^3 relies on a single Faster R-CNN detector for both object selection during synthesis and annotation cleaning, and its data may be shaped by architecture biases at multiple stages. In contrast, our layout construction and image generation are detector-agnostic, with a detector involved only in the final cleaning step, resulting in a distilled dataset with stronger cross-architecture transferability. We present detailed statistics of the distilled data in §B.

Cross-domain generalization. We also evaluate the effectiveness of LCD^3 when the diffusion prior is not aligned with the target domain. The evaluation is conducted on HRRSD [45], a remote sensing detection benchmark with 21,761 aerial images and 13 categories. We distill from the official training set and evaluate on the test split at 0.5% IPD. The generator is the unmodified MS-Diffusion, and the observer is LAE-DINO [46], an open-vocabulary detector for remote sensing. As shown in Table 4, LCD3 outperforms the selection baselines by a large margin. The framework transfers because the domain appearance does not come from the text prior alone. Both the layouts and the verified reference crops are drawn from the real target-domain data, so the generator performs conditional rendering anchored on real in-domain pixels. For domains that lie far outside natural-image priors, the generator can be directly replaced by one pre-trained or fine-tuned on the target domain without changing the pipeline.

Table 4: Cross-domain evaluation on HRRSD at 0.5% IPD. The generator is the unmodified MS-Diffusion, and the observer is LAE-DINO

Method	mAP%	mAP_{50}%	mAP_{75}%
Random	$1.1_{\pm 0.3}$	$4.0_{\pm 1.2}$	$0.3_{\pm 0.0}$
Uniform	$1.5_{\pm 0.4}$	$4.0_{\pm 0.6}$	$0.7_{\pm 0.6}$
LCD^3 (MS-Diff-D)	$\mathbf{7.2}_{\pm 0.1}$	$\mathbf{16.4}_{\pm 0.5}$	$\mathbf{5.3}_{\pm 0.1}$

Table 5: **(a) Component ablation**. SC: Scene construction; MQ: Multi-Quadrants; IF: Instance Filter; Diff: Diffusion generation. Results are reported on COCO with LCD^3 (MS-Diff-D) at 0.25% IPD. **(b) Ablation on scene embedding for clustering**. Results are reported in mAP_{50}(%) on VOC using MS-Diffusion.

(a)

SC	MQ	IF	Diff	mAP%	mAP_{50}%	mAP_{75}%
–	–	–	✓	$5.1_{\pm 0.1}$	$11.2_{\pm 0.7}$	$3.8_{\pm 0.1}$
✓	–	–	✓	$5.4_{\pm 0.0}$	$11.9_{\pm 0.1}$	$4.3_{\pm 0.1}$
–	✓	–	✓	$5.8_{\pm 0.1}$	$13.7_{\pm 0.1}$	$3.9_{\pm 0.2}$
✓	✓	–	✓	$7.3_{\pm 0.2}$	$15.2_{\pm 0.6}$	$6.1_{\pm 0.1}$
✓	✓	–	–	$5.1_{\pm 0.1}$	$10.8_{\pm 0.4}$	$3.9_{\pm 0.2}$
✓	✓	✓	✓	$\mathbf{8.7}_{\pm 0.3}$	$\mathbf{16.8}_{\pm 0.6}$	$\mathbf{6.8}_{\pm 0.3}$

(b)

Scene Embedding	IPD = 0.5%	IPD = 1%	Time/image
Hand-crafted	$33.0_{\pm 2.4}$	$44.7_{\pm 1.4}$	**0.3** ms
CLIP	$34.2_{\pm 0.6}$	$45.6_{\pm 1.8}$	6.9 ms
DINOv2	$33.5_{\pm 2.4}$	$46.1_{\pm 1.7}$	13.8 ms
SAM 3	$\mathbf{38.2}_{\pm 1.1}$	$\mathbf{48.8}_{\pm 0.4}$	421.6 ms

4.3 Ablation Study and Analysis

Contributions of LCD^3 components. We incrementally add each component of LCD^3 at 0.25% IPD on COCO [13] to evaluate their individual contributions, as shown in Table 5a. The baseline uses MS-Diffusion [33] with single-image layouts and no filtering. Adding scene-based layout construction alone provides a modest improvement, as it diversifies the scene composition of the layouts while the number of objects per image remains unchanged. Multi-quadrant composition alone provides a 2.5% mAP_{50} gain, and combining the two components improves the baseline by 4.0% mAP_{50}, which indicates that the two components are complementary. To isolate the contribution of the visual regeneration, we further replace the generated images with the original pixels of the selected real images under the identical SC and MQ configuration, which drops the result from 15.2% to 10.8% mAP_{50}. Finally, the instance filtering contributes a further 1.6% mAP_{50}, validating its role in ensuring reference quality for the generation process.

Scene embedding comparison. We compare four embedding choices for the scene clustering step in Table 5b: a hand-crafted vector that concatenates per-class object counts, total object count, and mean object area for each image, CLIP, DINOv2, and SAM 3 [36, 39, 47]. Among pretrained model embeddings, all three outperform the hand-crafted baseline, confirming that learned representations capture richer scene-level information for clustering. SAM 3 achieves the best performance by a notable margin, reaching 38.2% mAP_{50} at 0.5% IPD compared to 34.2% for CLIP and 33.5% for DINOv2. However, as a segmentation model, SAM 3's encoder expects an input resolution of 1008×1008 to produce pixel-level features, whereas CLIP and DINOv2 operate at 224×224 for image-level encoding. This results in SAM 3 taking approximately $60\times$ longer to embed the same dataset. This presents a trade-off between clustering quality and processing efficiency. We adopt CLIP and DINOv2 in our main experiments, while SAM 3 serves as an upper bound that demonstrates the potential of higher-quality scene representations for layout construction.

Layout construction strategy. We evaluate different strategies within the layout construction module in Table 6a, examining whether scene diversity across quadrants and class-balanced layout sampling can improve the downstream training accuracy of the generated images. In the first configuration, all four quadrants come from a single cluster and layouts are randomly selected, meaning each synthetic image contains visually similar scenes. In the second, quadrants are drawn from four different clusters but still with random layout selection. The third configuration is the one applied in our main pipeline, where diverse clusters are combined with class-balanced selection that matches the original dataset's class distribution. The results show that switching from single to diverse clusters improves mAP by

Table 6: **(a) Ablation on layout construction strategy.** Each synthetic image consists of four quadrants. *Cluster sampling*: whether quadrants draw from the same or different scene clusters. *Layout selection*: how the source image is chosen within each cluster. Results on COCO at 0.5% IPD using MS-Diff-D. **(b) Impact of the number of sub-images per synthetic image.** We compare diffusion-generated images with directly resized and combined real images to isolate the contribution of diffusion. Results are reported in mAP(%) on COCO at 0.25% IPD.

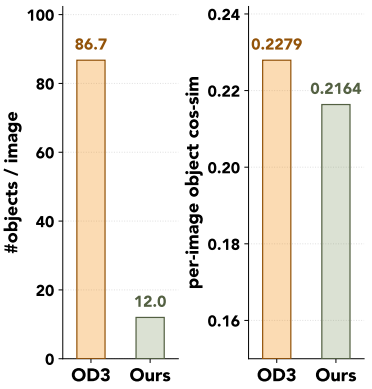
(a)

Cluster Sampling	Layout Selection	mAP%	mAP$_{50}$%	mAP$_{75}$%
Single cluster	Random	$9.1_{\pm 0.9}$	$19.4_{\pm 1.9}$	$7.4_{\pm 0.7}$
Diverse clusters	Random	$9.6_{\pm 0.1}$	$20.3_{\pm 0.2}$	$7.7_{\pm 0.1}$
Diverse clusters	Class-balanced	$\mathbf{10.4}_{\pm 0.3}$	$\mathbf{20.6}_{\pm 0.4}$	$\mathbf{9.6}_{\pm 0.4}$

(b)

# Sub-image	Direct Paste	Diffusion
1	$3.3_{\pm 0.7}$	$\mathbf{6.2}_{\pm 0.2}$
4	$5.1_{\pm 0.1}$	$\mathbf{8.7}_{\pm 0.3}$
9	$5.7_{\pm 0.3}$	$\mathbf{7.5}_{\pm 0.1}$





(a) (b)

Figure 3: (a) **Qualitative comparison** between original data, OD3, and our method. Our distilled images provide significantly better layout preservation and lower redundancy than OD^3 (More in §F). (b) **Density vs. diversity of synthetic images.** Our method uses significantly fewer objects per image yet achieves lower per-image object similarity, reflecting higher object diversity with less redundancy.

0.5% at 0.5% IPD, and further adopting class-balanced selection adds another 0.8% mAP, confirming that both scene diversity and category coverage contribute to the final performance.

Number of sub-images and effect of diffusion. The number of sub-images controls the trade-off between scene diversity per image and per-quadrant resolution. We compare 1×1, 2×2, and 3×3 grids in Table 6b, along with a direct paste baseline that resizes and combines real images without diffusion. The 2×2 setting yields the best overall performance, confirming the benefit of packing diverse scenes into a single image under tight image budgets. Diffusion-generated images show better performance than direct paste across all grid sizes, demonstrating the contribution beyond the layout composition alone. The advantage is greatest at 2×2 and narrows at 3×3, as the reduced per-quadrant resolution limits the diffusion model's ability to generate fine-grained details. Based on these results, the 2×2 design with diffusion is selected as the default configuration.

Qualitative comparison. In addition to quantitative evaluation, we compare the distilled data qualitatively. Figure 3a presents a visual comparison between OD^3 and our methods. Each original training image captures a single scene with a limited number of objects, which means that under strict compression budgets, the distilled dataset can only cover a small fraction of the available visual contexts. OD^3 addresses this by densely packing object crops onto a single canvas, but the resulting images exhibit severe visual artifacts: objects of vastly different scales and contexts overlap with inconsistent lighting and unnatural spatial arrangements, breaking the local spatial context that each object originally inhabited. Moreover, OD^3 directly copies real object crops without deduplication, leading to redundant object appearances across the distilled dataset. In contrast, LCD^3 synthesizes images containing multiple distinct scene types across quadrants, reflecting the effectiveness of our scene-based clustering in covering a broad range of visual contexts. Within each quadrant, the

generated content preserves the original spatial relationships among co-occurring objects, including relative positions, scales, and background context, demonstrating that layout-conditioned diffusion can produce locally coherent scenes while introducing visual diversity across the dataset.

Density vs. diversity analysis. We further conduct a statistical analysis on the object count and diversity between our method and OD^3, with results shown in Figure 3b. OD^3 packs an average of 86.7 objects per image through dense copy-paste, whereas our method contains only 12 objects per image. Despite this significantly lower count, our distilled images achieve a lower per-image object cosine similarity of 0.2164 compared to 0.2279 for OD^3, indicating that our method captures greater object diversity with substantially less redundancy. More empirical results are in §C.

5 Conclusion

We presented LCD^3, a layout-conditioned diffusion framework for dataset distillation in object detection. Our approach is motivated by the observation that detection images exhibit a structural asymmetry between spatial layouts and visual realizations: layouts define the detection task and should be preserved and diversified, while visual realizations can be redundant across the dataset and benefit from regeneration. Building on this separation principle, LCD^3 recombines original layouts from diverse scene groups into composite images to increase layout coverage, and uses layout-conditioned diffusion to render diversified visual realizations while maintaining coherent scene structure. The generation process is further grounded with semantically verified real object crops to retain alignment with the original visual distribution. Experiments on MS COCO and PASCAL VOC demonstrate consistent improvements over existing methods across multiple compression ratios and detector architectures, supporting the effectiveness of layout-preserving visual regeneration as a principled approach to dataset distillation for object detection.

Limitations and future work. Our work has two limitations. First, we only evaluate on the object detection task, although the layout-conditioned generation framework is not inherently task-specific. Second, the number of sub-images per composite canvas is fixed for all images, ignoring the varying complexity of individual scene layouts. In future work, we plan to extend the framework to other dense prediction tasks such as instance segmentation, evaluate it on domains far outside natural-image priors, such as medical imaging, and explore adaptive grid strategies that adjust the partition based on scene content.

Acknowledgments and Disclosure of Funding

This work was supported by the Mitacs Accelerate Program (Grant No. IT42711), by TerraSense Analytics, and in part by grants from the US National Science Foundation (OAC-2118240, HDR Institute: Imageomics).

References

[1] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020. 1

[2] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1, 5, 7, 18

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 6, 7

[5] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions. *arXiv preprint arXiv:2305.01975*, 2023. 1

[6] Ping Liu and Jiawei Du. The evolution of dataset distillation: Toward scalable and generalizable solutions. *arXiv preprint arXiv:2502.05673*, 2025.

[7] Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25570–25580, 2025. 3

[8] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 3

[9] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15793–15803, 2024. 1, 3

[10] Salwa K Al Khatib, Ahmed ElHagry, Shitong Shao, and Zhiqiang Shen. Od3: Optimization-free dataset distillation for object detection. *arXiv preprint arXiv:2506.01942*, 2025. 1, 2, 3, 5, 6, 7, 16, 17, 18

[11] Ding Qi, Jian Li, Jinlong Peng, Bo Zhao, Shuguang Dou, Jialin Li, Jiangning Zhang, Yabiao Wang, Chengjie Wang, and Cairong Zhao. Fetch and forge: Efficient dataset condensation for object detection. *Advances in Neural Information Processing Systems*, 37:119283–119300, 2024. 3, 5

[12] Ding Qi, Jian Li, Junyao Gao, Shuguang Dou, Ying Tai, Jianlong Hu, Bo Zhao, Yabiao Wang, Chengjie Wang, and Cairong Zhao. Towards universal dataset distillation via task-driven diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10557–10566, 2025. 1, 2, 3, 5

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5, 6, 8

[14] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22511–22521, 2023. 2, 3, 5

[15] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9390–9399, 2024. 2, 3

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2, 5, 7

[17] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 2, 5, 7

[18] Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2023. 3

[19] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE transactions on pattern analysis and machine intelligence*, 46(1):150–170, 2023. 3

[20] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl. 3

[21] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022. 3

[22] Hansong Zhang, Shikun Li, Fanzhao Lin, Weiping Wang, Zhenxing Qian, and Shiming Ge. DANCE: Dual-view distribution alignment for dataset condensation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2024. 3

[23] Haoyang Liu, Yijiang Li, Tiancheng Xing, Vibhu Dalal, Luwei Li, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric. *arXiv preprint arXiv:2311.18531*, 2023. 3

[24] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023. 3

[25] Jeffrey A. Chan Santiago, Praveen Tirupattur, Gaurav Kumar Nayak, Gaowen Liu, and Mubarak Shah. MGD^3: Mode-guided dataset distillation using diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. 3

[26] Duo Su, Huyu Wu, Huanran Chen, Yiming Shi, Yuzhu Wang, Xi Ye, and Jun Zhu. Diffusion models as dataset distillation priors. *arXiv preprint arXiv:2510.17421*, 2025. 3

[27] Haoxuan Wang, Zhenghao Zhao, Junyi Wu, Yuzhang Shang, Gaowen Liu, and Yan Yan. Cao2: Rectifying inconsistencies in diffusion-based dataset distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4722–4731, 2025. 3

[28] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8584–8593, 2019. 3

[29] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10531–10540, 2019.

[30] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 3

[31] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 3

[32] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6232–6242, 2024. 3

[33] Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. Ms-diffusion: Multi-subject zero-shot image personalization with layout guidance. *arXiv preprint arXiv:2406.07209*, 2024. 3, 5, 8

[34] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007. 4

[35] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009. 4

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 5, 8

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[38] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 5

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5, 8

[40] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[41] Weihan Cao, Yifan Zhang, Jianfei Gao, Anda Cheng, Ke Cheng, and Jian Cheng. Pkd: General distillation framework for object detectors via pearson correlation coefficient. *Advances in Neural Information Processing Systems*, 35:15394–15406, 2022. 6

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[43] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7

[44] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 7

[45] Yuanlin Zhang, Yuan Yuan, Yachuang Feng, and Xiaoqiang Lu. Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5535–5548, 2019. 7

[46] Jiancheng Pan, Yanxing Liu, Yuqian Fu, Muyuan Ma, Jiahao Li, Danda Pani Paudel, Luc Van Gool, and Xiaomeng Huang. Locate anything on earth: Advancing open-vocabulary object detection for remote sensing community. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6281–6289, 2025. 7

[47] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 8

Appendix

The Appendix is organized as follows:

- In §A, we present detailed instructions for the layout construction step to help reproduce the experimental results.
- In §B, we illustrate the statistics of the distilled detection samples.
- In §C, we show more empirical results, including the ablation on random layout construction, parameter analysis, and inference efficiency analysis.
- In §D, we provide the analysis on crop semantic verification.
- In §E, we analyze the choice of observer model.
- In §F, we show more distilled data samples produced by our method.
- In §G, we discuss potential societal impacts of the proposed method.

A Additional Details of the Layout Construction

We provide additional details on the layout construction process described in §3.2, covering the cluster-to-quadrant assignment, the per-slot image selection, the text prompt construction, and the coordinate transformation.

Cluster-to-quadrant assignment. Since there are M synthetic images each requiring four quadrants, the total number of slots is $4M$. With M clusters, each cluster is assigned to exactly four slots. We construct a deck of $4M$ cluster indices where each cluster ID appears four times and deal it into M groups of four. To ensure no canvas contains duplicate cluster IDs, we perform a single forward pass of pairwise swaps: whenever a slot duplicates a cluster ID already present in its group, we exchange it with a compatible slot from a later group.

Per-slot image selection. Image selection fills the $4M$ slots in three phases. In Phase 1, we place the medoid of each cluster into its first assigned quadrant slot and update the running class accumulators $n_c^{\mathcal{S}}$ and $N^{\mathcal{S}}$, so that all subsequent scoring reflects the instances already committed to the dataset. In Phase 2, the remaining $3M$ slots are sorted by each cluster's aggregate class deficit in descending order, and for each slot, we score candidate images within the assigned cluster via Equation 4 and place the top-scoring candidate. When multiple candidates yield the same score, ties are broken by the image order stored in the cluster file. Phase 3 activates once the running class distribution is sufficiently close to the target distribution for all classes; the remaining slots are then filled by uniform random sampling within each assigned cluster.

Text prompt construction. The text prompt for each composite layout uses a quadrant-explicit template that specifies both position and content: "`top-left: [classes], top-right: [classes], bottom-left: [classes], bottom-right: [classes]`", where each quadrant's entry lists the object count and class name for all instances in that quadrant (e.g., "`2 person 1 dog`"), sorted alphabetically. The same class appearing in different quadrants is listed in each, since quadrants represent distinct sub-scenes.

Quadrant coordinate transformation. For a source image $\boldsymbol{x}_i$ assigned to quadrant Q of the synthetic image $\hat{\boldsymbol{x}}_j$, we transform its bounding box annotations to fit the quadrant. Let $\boldsymbol{b} = (x_c, y_c, w, h)$ be a normalized bounding box in $\boldsymbol{x}_i$. The transformed box in $\hat{\boldsymbol{x}}_j$ is:

$$\hat{\boldsymbol{b}} = \left(\frac{x_c}{2} + \delta_x,\ \frac{y_c}{2} + \delta_y,\ \frac{w}{2},\ \frac{h}{2} \right), \tag{8}$$

where $(\delta_x, \delta_y) \in \{(0,0), (0.5,0), (0,0.5), (0.5,0.5)\}$ are the quadrant offsets for $\{\text{TL}, \text{TR}, \text{BL}, \text{BR}\}$, respectively. Each bounding box is rescaled to half resolution in both dimensions and shifted to the corresponding quadrant position within the synthetic image.

B Statistics of the Distilled Data

We summarize the distribution of the distilled dataset on PASCAL VOC at 0.5% IPD in Figure 4. For each class, we match the original detection layout to the final annotations in the distilled images.

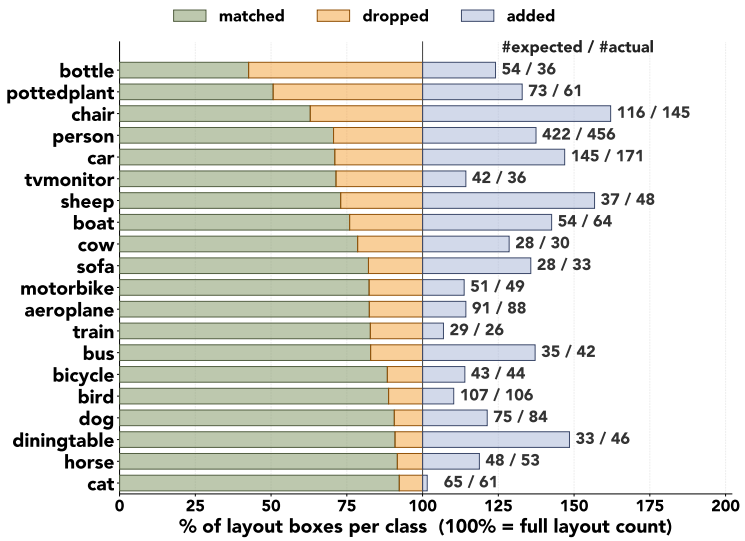


Figure 4: **The class-wise sample distribution** on PASCAL VOC at 0.5% IPD. "Matched": final annotation with ≥ 0.3 IoU with an original conditioning bounding box. "Dropped": original box without matched final annotation. "Added": final annotation without matched original box. Overall, the final annotations broadly preserve the intended class distribution while recovering additional plausible objects in categories that commonly appear in groups.

A box is marked as "matched" if it matches a final annotation with IoU ≥ 0.3, as generated objects may not exactly align with the conditioning boxes after diffusion rendering. Original boxes with no matched final annotations are marked as "dropped," and final annotations without matched original boxed are marked as "added." Across most classes, most of the layout boxes are successfully matched, confirming that the diffusion model reliably generates objects at the specified locations. Classes with low matched ratios are generally small objects, such as *bottle* and *pottedplant*. It is challenging for diffusion models to generate these objects at the precise location. Many classes yield more annotated objects than the originally specified conditioning. These classes, including *chair*, *sheep*, and *boat*, are objects that typically appear in groups in natural scenes. The diffusion model leverages its learned scene priors to generate additional instances of these objects beyond what the layout explicitly specifies, and the observer successfully recovers them as extra training signals. Overall, the resulting class distribution matches that of the original layout conditioning.

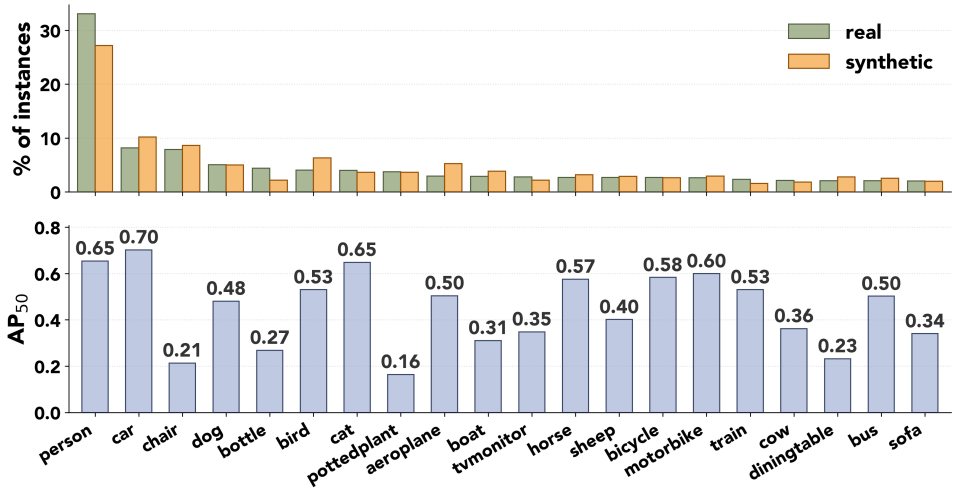


Figure 5: **Class distribution and per-class detection performance** of the distilled dataset. The upper panel compares the distribution of real and distilled data. The lower panel reports per-class AP_{50} of the detector trained on the distilled dataset.

We further compare the class distribution of the distilled dataset and the full original dataset. As shown in Figure 5, the distilled set faithfully recovers the original data distribution, especially for the long-tailed classes. It suggests that the layout construction and annotation cleaning do not severely distort the original category structure. The largest difference appears for dominant categories such as person, where the distilled set has a lower proportion than the real set.

The per-class AP_{50} results in the lower panel of Figure 5 show that class distribution does not have a strong correlation with detection performance. Some categories with moderate or small instance shares achieve strong AP_{50}, indicating that the generated samples provide effective visual and spatial supervision for these classes.

C More Empirical Results

C.1 Ablation on Random Layout Construction

In this paper, we propose to preserve the layouts of the original images and synthesize specific visual realizations with generative models. Table 6b demonstrates that synthesizing more diverse content based on the original layout yields improved performance. We further investigate the necessity of using real layouts from original images as generation conditioning. In Table 7, we compare constructing random layouts as the generation conditioning with our strategy. The results show that using real layouts yields a consistent advantage over random layouts. This observation is consistent with our advantage over OD^3, further proving the importance of maintaining the informative spatial structure of the original data.

Table 7: **Comparison between random layouts and real layouts** as generation conditioning. Results are based on MS-Diffusion-D, evaluated with Faster R-CNN-50 on COCO at 0.25% IPD.

Layout Construction	mAP%	mAP_{50}%	mAP_{75}%
Random layout	$6.1_{\pm 0.6}$	$14.0_{\pm 1.6}$	$4.2_{\pm 0.4}$
Real layout	$\mathbf{8.7}_{\pm 0.3}$	$\mathbf{16.8}_{\pm 0.6}$	$\mathbf{6.8}_{\pm 0.3}$

C.2 Parameter Analysis on Observer Confidence Threshold

We study the sensitivity of the observer confidence threshold τ_c introduced in §3.3. As reported in Table 8, a low threshold with $\tau_c = 0.1$ admits noisy detections that introduce incorrect annotations, while a high threshold with $\tau_c = 0.5$ aggressively filters out valid objects, reducing annotation density. The optimal threshold of $\tau_c = 0.2$ balances these two effects, and the results are consistent across both 0.25% and 0.5% IPD. The results are also aligned with the observation in [10].

Table 8: **Impact of different observer cleaning thresholds**. Results are based on MS-Diffusion-D, evaluated with Faster R-CNN-50 on COCO.

Threshold	IPD=0.25%			IPD=0.5%		
	mAP%	mAP_{50}%	mAP_{75}%	mAP%	mAP_{50}%	mAP_{75}%
0.1	$7.6_{\pm 0.3}$	$16.3_{\pm 0.5}$	$6.2_{\pm 0.4}$	$9.7_{\pm 0.3}$	$19.7_{\pm 0.4}$	$8.5_{\pm 0.5}$
0.2	$\mathbf{8.7}_{\pm 0.3}$	$\mathbf{16.8}_{\pm 0.6}$	$\mathbf{6.8}_{\pm 0.3}$	$\mathbf{10.4}_{\pm 0.3}$	$\mathbf{20.6}_{\pm 0.4}$	$\mathbf{9.6}_{\pm 0.4}$
0.3	$7.8_{\pm 0.2}$	$16.1_{\pm 0.4}$	$6.7_{\pm 0.1}$	$9.9_{\pm 0.3}$	$19.4_{\pm 0.4}$	$9.1_{\pm 0.4}$
0.5	$6.4_{\pm 0.4}$	$12.8_{\pm 0.8}$	$5.6_{\pm 0.5}$	$7.9_{\pm 0.2}$	$15.4_{\pm 0.8}$	$7.2_{\pm 0.5}$

C.3 Efficiency Analysis

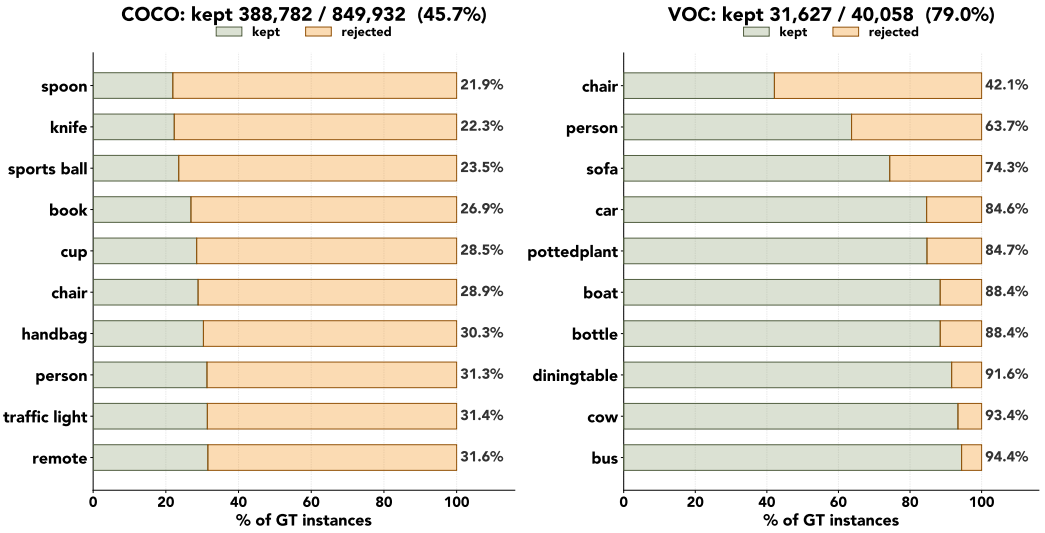
We analyze the synthesis efficiency of LCD^3 (MS-Diff-D) on PASCAL VOC at 0.5% IPD, with all experiments conducted on a single NVIDIA V100 GPU. Our framework requires no training and completes the entire synthesis pipeline in approximately $1{,}844$ seconds. The dominant cost is the diffusion generation step at roughly 13 seconds per image, while the remaining steps, including CLIP-based instance verification, scene embedding extraction, and observer-based post-processing,

are comparatively lightweight. Under the same setting, OD^3 [10] takes approximately 2,022 seconds in total. Its primary bottleneck is the object selection and placement stage, which costs roughly 23 seconds per image due to the large number of random placement attempts per object candidate. As datasets scale to larger benchmarks such as COCO, the time cost of this repeated trial-and-error placement becomes increasingly expensive, whereas our layout construction scales more favorably to larger datasets.

D Crop Semantic Verification Analysis

We apply a crop semantic verification to select only reliable instances as generation references, as illustrated in §3.3. We analyze the retention rate of our CLIP-based crop semantic verification across all classes on both VOC and COCO. Figure 6a and Figure 6b show the percentage of ground-truth instances that pass the top-1 class matching for the 10 lowest-retention classes on each dataset. On VOC, the verification retains 79.0% of all instances overall, with most classes exceeding 80.0%. Classes with lower retention, such as *chair* at 42.1% and *person* at 63.7%, tend to involve small, occluded, or truncated instances whose crops lack sufficient visual information for reliable classification. On COCO, the overall retention rate drops to 45.7%, reflecting the larger number of small objects and fine-grained categories in this dataset. Classes such as *spoon* at 21.9% and *knife* at 22.3% are frequently rejected due to their small size and visual ambiguity when cropped in isolation.

Figure 7 visualizes representative rejected crops from both datasets. On VOC, rejected instances often reflect annotation ambiguity or heavy occlusion. For example, some crops labeled as *chair* actually depict sofas, which the CLIP verification correctly identifies as a category mismatch. Other *chair* instances are largely occluded by dining tables, leaving only fragments that are insufficient for reliable classification. On COCO, the dataset contains a much larger set of categories, some of which are inherently small in size, such as *spoon* and *sports ball*. Under certain viewpoints or scene configurations, these objects may occupy only a few pixels in the original image, and their crops contain too little visual detail to be reliably classified or to serve as meaningful reference images for layout-conditioned generation. In both cases, the filtering effectively rejects instances that would introduce noise when serving as references rather than adding useful supervision to the distilled dataset. In such cases, only instance labels and bounding boxes are used as generation conditioning.



(a) MS COCO (b) PASCAL VOC

Figure 6: **Per-class retention rate** for the 10 lowest-retention classes on (a) MS COCO and (b) PASCAL VOC, respectively.

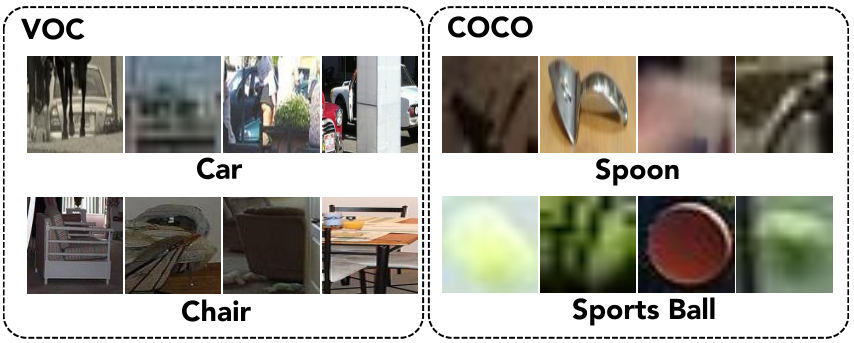


Figure 7: **Examples of rejected crops** from VOC and COCO.

E Details of the Observer Model

We use a DINO observer model [2] to perform annotation cleaning. The model is trained on COCO, which is consistent with the setting in OD^3 [10], ensuring a fair comparison under the established evaluation protocol. The observer is used only for post-generation annotation filtering, not for layout construction or image synthesis. We choose a transformer-based model as the backbone because its architecture differs from the ResNet-50-based Faster R-CNN detectors used for the main evaluation, thereby reducing direct architectural coupling. We note that even if the observer model is trained on COCO, LCD^3 yields even larger performance improvement over previous methods on VOC, as shown in Table 2. We additionally study the sensitivity of LCD^3 to the observer detector used for annotation cleaning in Table 9. As expected, stronger observers produce cleaner annotations and lead to better downstream performance. Meanwhile, the influence of observer models is relatively mild compared with that of the proposed components in Table 5a, especially on the mAP$_{50}$ metric.

Table 9: **Impact of observer model choice**. All experiments use MS-Diffusion-D on COCO at 0.25% IPD. The evaluation model is Faster R-CNN-50.

Observer Model	mAP%	mAP$_{50}$%	mAP$_{75}$%
Faster R-CNN	$7.4_{\pm 0.3}$	$16.8_{\pm 0.3}$	$5.3_{\pm 0.4}$
DINO	$8.7_{\pm 0.3}$	$16.8_{\pm 0.6}$	$6.8_{\pm 0.3}$

Examples of annotation cleaning. Figure 8 shows examples of annotation cleaning with the observer model. Green boxes denote final annotations produced by the observer, and red boxes denote objects specified in the original layout but not retained after cleaning. The examples show that layout-conditioned generation may fail in accurately rendering some intended objects. Using the original layout as ground truth would introduce label noise. The observer-based cleaning removes these unreliable annotations and keeps only objects that are actually present and recognizable in the generated images.

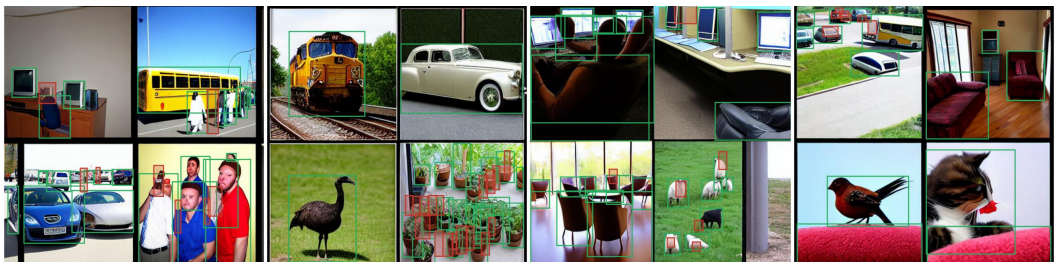


Figure 8: **Visualization of observer-based annotation cleaning**. Green boxes indicate the final annotation produced by the observer model. Red boxes indicate objects that were specified in the original layout but either failed to be generated or could not be recognized by the observer.



Figure 9: Additional qualitative results of our synthesized images on MS COCO. For each dataset, we show samples generated by MS-Diffusion (top) and GLIGEN (bottom). Each synthetic image contains four quadrants drawn from different scene clusters.



Figure 10: Additional qualitative results of our synthesized images on PASCAL VOC. For each dataset, we show samples generated by MS-Diffusion (top) and GLIGEN (bottom). Each synthetic image contains four quadrants drawn from different scene clusters.

F Additional Qualitative Results

We present additional samples of our synthesized images in Figure 9 and Figure 10 for COCO and VOC, respectively. Across both datasets and both diffusion backends, each synthetic image covers diverse scene types across its four quadrants while maintaining locally coherent object placement within each quadrant.

G Broader Impact

This work may have positive societal impacts by reducing the computational overhead of training object detectors. It will lower the barrier for training new detection models in domains such as environmental monitoring, agriculture, and scientific imaging. Thereby, dataset distillation can help detection models to be more accessible to researchers and practitioners with limited resources. At the same time, the method relies on pretrained generative models and visual encoders, which may inherit biases from their training data and propagate them into the distilled datasets. The use of synthetic

images may also amplify spurious visual patterns or reduce coverage of rare and small objects if the generation or filtering process fails. Therefore, deployment of this technique should include careful auditing of generated data and subgroup performance analysis. The used pretrained models and downstream application context should also be considered upon deployment.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have conducted extensive empirical experiments to support the effectiveness of our designed paradigm.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have included discussion on limitations in §5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have provided detailed instructions regarding how to generate quadrants in §A.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [No]

Justification: The code is developed as part of an industry research collaboration and is subject to proprietary restrictions and potential intellectual property considerations, which prevent public release at this time. We are working with our organization to explore the possibility of open-sourcing the code in the future. All datasets used in our experiments are publicly available, and we include comprehensive implementation details in the paper to support reproducibility.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We have provided the hyperparameter setting in §4.1 and the parameter analysis in §C.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We have reported the standard deviation for the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We have included detailed experimental settings in §4.1. We have also reported the inference time in §C.3, since the method itself does not involve a new training process.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have included the discussion on broader impacts in §G.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The produced data is based on existing object detection datasets and pretrained models, hence does not introduce new risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have cited the original publication of the used pretrained models and datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets. The code and data are proprietary and are not publicly released.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.