X-MV3D: BENCHMARKING CROSS-DATASET GENERALIZATION IN MULTI-VIEW 3D OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite substantial progress in multi-view 3D object detection, existing detectors are still developed and evaluated largely within individual datasets, leaving their ability to transfer to newly collected datasets unclear. We introduce X-MV3D, a unified benchmark for cross-dataset scene-level 3D detection from multi-view captures. X-MV3D converts ARKitScenes, ScanNet, HyperSim, and 3RScan into a common gravity-aligned coordinate frame, with standardized scene records, category mappings, and 3D box annotations, enabling the four datasets to be trained and evaluated under the same protocol. We define three evaluation settings: single-source transfer, leave-one-dataset-out (LODO) transfer, and target-included mixed training, with LODO serving as the primary target-unseen setting. Across representative detector families, we find that current multi-view 3D detectors are strongly dataset-conditioned. Multi-source training improves over single-source transfer, but LODO remains far below target-included training, showing that multi-source training alone is insufficient for reliable new-dataset transfer. Failure decomposition further shows the transfer gap is dominated by degraded proposal coverage and box localization, suggesting that acquisition shifts may affect the camera-dependent feature lifting and multi-view aggregation. Motivated by this analysis, we propose CamGeoGate, a lightweight detector-agnostic baseline that injects explicit camera geometry into image features and trains detectors with stochastic view gating. CamGeoGate consistently improves LODO performance across three detector families, showing that camera-aware and view-robust training is a promising direction for reusable multi-view 3D detectors.

1 INTRODUCTION

Multi-view 3D object detection localizes objects as 3D bounding boxes in a shared scene coordinate system from multiple calibrated images. It enables object-level scene understanding for robotics and embodied AI Gu et al. (2024); Wang et al. (2024), AR/VR Ahmadyan et al. (2021); Pan et al. (2023), and digital twins Pan et al. (2023); Sundby et al. (2021). In these applications, predicted object categories and 3D boxes provide structured annotations for downstream reasoning and interaction Gu et al. (2024); Tahara et al. (2020).

Existing multi-view 3D detectors are primarily developed and evaluated in in-domain settings Rukhovich et al. (2022); Tu et al. (2023); Xu et al. (2024); Zhang et al. (2025b). However, in real-world deployments, camera intrinsics, capture trajectories, and view coverage vary across acquisition pipelines. How well existing multi-view 3D detectors perform under such variations remains unclear, particularly given their reliance on camera geometry. Consequently, obtaining reliable 3D detections for downstream tasks may require collecting and annotating target-domain data and retraining the detector, introducing substantial costs. A more generable cross-dataset multi-view 3D object detection model is therefore important, yet this capability remains underexplored.

To fill this blank, we introduce X-MV3D, a unified benchmark for cross-dataset generalization in multi-view 3D object detection. We integrate the most widely used 4 public datasets ARKitScenes Baruch et al. (2021), ScanNet Dai et al. (2017), HyperSim Roberts et al. (2021), and 3RScan Wald et al. (2019), covering mobile RGB-D captures, reconstructed real-world scans, and synthetic scenes. We harmonize their data structures and evaluation protocols to support unified comparisons across datasets and detector families. We define three evaluation settings: single-source

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

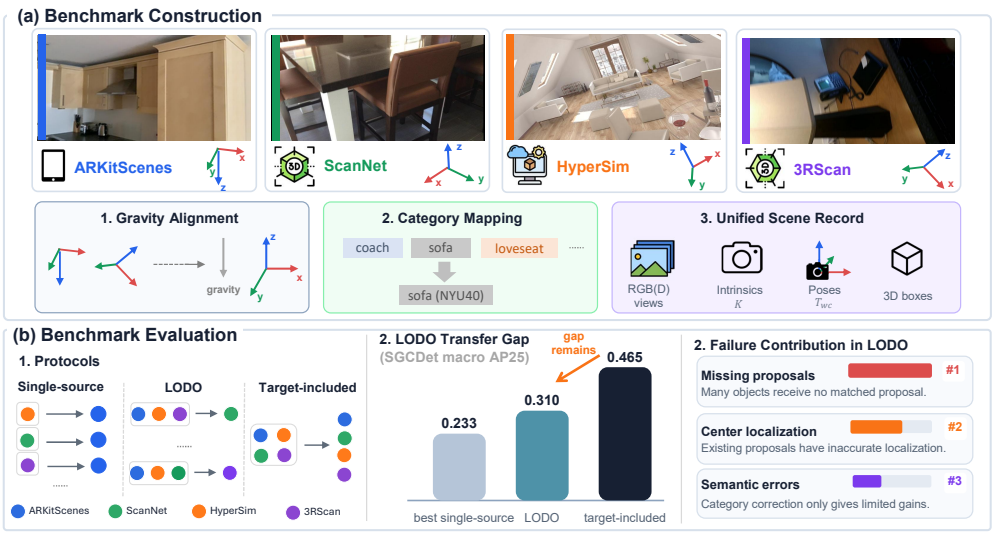


Figure 1: Overview of the X-MV3D benchmark. (a) Benchmark construction: we convert four heterogeneous scene datasets to a shared gravity-aligned coordinate system, map their annotations to a common category vocabulary, and represent them through a unified scene-level record. (b) Benchmark evaluation: we define single-source transfer, leave-one-dataset-out (LODO) transfer, and target-included mixed training. Results show a substantial target-unseen transfer gap, while failure decomposition indicates that the gap mainly arises from reduced proposal coverage and inaccurate box localization.

transfer, leave-one-dataset-out (LODO) transfer, and target-included mixed training (Fig. 1(b)). Our primary setting is LODO transfer: detectors are jointly trained on three datasets and evaluated on the remaining dataset, with each dataset serving as the held-out target in turn. Together, these settings study two questions: whether strong in-domain performance transfers to unseen datasets, and whether training on a larger, more diverse pool of source data can close the gap to target-included performance.

Our benchmark reveals that current multi-view 3D detectors are strongly dataset-conditioned. Across representative detection families, in-domain and target-included performance are much higher than target-unseen transfer. For example, SGCDet obtains 0.233/0.082 macro mAP@0.25/mAP@0.50 under the best single-source setting, improves to 0.310/0.103 under LODO, but reaches 0.465/0.263 when target training data is included. Thus, multi-source training improves transfer, but a substantial gap to target-included performance remains. We further decompose transfer failure and find that the gap mainly comes from degraded proposal coverage and box localization (Fig. 1(b)). We regard acquisition-pipeline shifts as one contributing factor to these geometric failures. Multi-view detectors rely on camera-conditioned lifting of image features into 3D scene representations and subsequent view aggregation. Variations in camera parameters, capture trajectories, and view density across datasets may therefore disrupt the learned correspondence between image observations and 3D structures.

Motivated by this analysis, we introduce CamGeoGate as a lightweight reference baseline for improving target-unseen transfer on X-MV3D. CamGeoGate targets two acquisition-related factors: camera geometry and view availability. It injects explicit camera geometry into image features, including camera intrinsics, camera-to-world rotation, and the camera's up direction in the shared scene coordinate system. It further applies stochastic view gating during training, perturbing view availability while preserving enough active views for scene-level aggregation. This encourages detectors to aggregate evidence from variable camera subsets instead of overfitting to dataset-specific capture trajectories and view coverage patterns. As a detector-agnostic plug-in, CamGeoGate improves LODO macro performance across three detector families: SGCDet from 0.310 to 0.338, ImGeoNet from 0.233 to 0.254, and ImVoxelNet from 0.214 to 0.248. These consistent gains suggest that explicitly modeling camera geometry and view availability is a promising direction for reusable multi-view 3D detectors.

Contributions.

- We introduce X-MV3D, a unified benchmark for cross-dataset scene-level 3D detection from multi-view captures, converting ARKitScenes, ScanNet, HyperSim, and 3RScan into a gravity-aligned coordinate system with standardized records for unified training and evaluation.
- We provide a systematic cross-dataset evaluation and failure analysis across representative detector families, showing that multi-source training improves target-unseen transfer but remains insufficient, with the remaining transfer gap dominated by degraded proposal coverage and box localization, consistent with acquisition-pipeline shifts.
- We propose CamGeoGate, a lightweight detector-agnostic baseline that injects explicit camera geometry into image features and improves view-availability robustness through stochastic view gating, consistently improving LODO performance across SGCDet, ImGeoNet, and ImVoxelNet.

2 Related Work

2.1 Multi-View 3D Object Detection

Multi-view 3D object detection aims to localize objects in a scene-level 3D coordinate system from multiple calibrated images. Recent supervised detectors mainly differ in how they lift and aggregate posed image features into 3D space. ImVoxelNet Rukhovich et al. (2022) projects image features into voxel representations, ImGeoNet Tu et al. (2023) builds image-induced geometry-aware voxel features, MVSDet Xu et al. (2024) introduces efficient plane-sweep geometry reasoning, and SGCDet Zhang et al. (2025b) adaptively constructs sparse 3D volumes for indoor detection. Despite these advances, existing methods are primarily trained and evaluated within individual datasets, leaving their cross-dataset generalization largely unexplored. Recent work has also explored more general 3D detection routes beyond closed-set supervised training Hsu et al. (2025); DeTone et al. (2026); Lemeshko et al. (2026); Lan et al. (2025); Mao et al. (2025); Kolodiazhnyi et al. (2025). Boxer DeTone et al. (2026) trains a category-agnostic 2D-to-3D box lifter on a large mixture of indoor datasets, Zoo3D Lemeshko et al. (2026) provides a training-free scene-level pipeline that constructs 3D boxes from class-agnostic mask clustering and assigns open-vocabulary labels with image-language features, and language-oriented systems such as SpatialLM Mao et al. (2025) cast scene structure prediction through multimodal language modeling. These methods broaden the applicability of 3D detection, but do not systematically evaluate target-unseen cross-dataset generalization. In contrast, we systematically evaluate this capability, analyze the underlying transfer failures, and develop a plug-in framework to improve cross-dataset generalization in scene-level multi-view 3D detection.

2.2 Cross-Dataset Generalization in 3D Detection

Cross-dataset generalization has been explored mainly in monocular 3D detection. Several methods Brazil et al. (2023); Li et al. (2024); Zhang et al. (2025a); Ma et al. (2024); Jhang et al. (2025); Yang et al. (2025) train unified monocular detectors on multiple datasets. Omni3D Brazil et al. (2023) demonstrates qualitative predictions on unseen COCO images. DetAny3D Zhang et al. (2025a) explicitly targets zero-shot generalization to unseen categories and camera configurations, but also operates in the monocular setting. In contrast, scene-level multi-view detectors are still largely trained and evaluated within individual datasets. Boxer DeTone et al. (2026) takes a step toward broader generalization by training on heterogeneous multi-device data, but provides only limited evidence of unseen-dataset transfer rather than a systematic cross-dataset evaluation.

Cross-dataset generalization has been studied more extensively for multi-camera 3D detection in autonomous driving, where methods address shifts across driving datasets, sensor configurations, and geographic domains Wang et al. (2023); Zheng et al. (2023); Chang et al. (2024). However, these works focus on vehicle-mounted cameras and outdoor road scenes, which differ substantially from scene-level captures with free-moving cameras and variable view coverage. Target-unseen cross-dataset generalization remains largely unexplored for scene-level multi-view 3D object detection.

Table 1: Dataset statistics in X-MV3D. We report the processed train/validation splits used in the benchmark.

Dataset	Train samp.	Val samp.	Train frames	Val frames	Train boxes	Val boxes	Train cls.	Val cls.
ARKitScenes	4,492	549	224,227	54,900	50,618	5,962	11	11
ScanNet	1,201	312	58,452	30,478	15,733	4,364	16	16
HyperSim	614	77	30,435	3,808	61,412	4,587	18	18
3RScan	1,266	42	62,064	2,110	7,960	361	11	11

To our knowledge, we establish the first systematic benchmark for this setting, analyze its failure modes, and develop a lightweight baseline to improve cross-dataset transfer.

3 X-MV3D Construction

3.1 Datasets and Inputs

X-MV3D is designed to evaluate both the cross-dataset transfer of existing multi-view 3D detectors and whether pooling training data across datasets fills the generalization gap to unseen datasets. We therefore integrate public datasets that provide (i) scene-level multi-view observations, (ii) object-level 3D annotations that can be represented as detection targets, and (iii) diverse acquisition settings and scene distributions.

Following this design, X-MV3D includes four representative datasets: ARKitScenes Baruch et al. (2021), ScanNet Dai et al. (2017), HyperSim Roberts et al. (2021), and 3RScan Wald et al. (2019). ARKitScenes provides mobile-device RGB-D captures with commodity LiDAR. ScanNet and 3RScan are both reconstructed real-world RGB-D scan datasets, but differ in camera viewpoint distribution, image orientation, reconstruction style, and annotation statistics. HyperSim provides photorealistic synthetic indoor scenes with rendered multi-view observations. Together, these datasets cover mobile RGB-D capture, real-world reconstructed scans, and synthetic renderings, making them suitable for evaluating cross-dataset transfer.

A controlled cross-dataset evaluation requires heterogeneous datasets to be converted into a shared representation. This is non-trivial because existing datasets differ in scene coordinate systems, camera-pose conventions, category definitions, annotation formats, and data organization. As illustrated in Fig. 1(a), we first align the coordinate systems across datasets. Specifically, we rotate each dataset's world coordinate system to align its vertical axis with gravity and apply the same transformation to all camera poses and 3D boxes. We then map dataset-specific category names to a shared NYU40 vocabulary for training. Finally, we convert each dataset into a common scene-level interface that stores RGB/depth paths, camera intrinsics and poses, and 3D box annotations. These steps allow the same detector interface and evaluation protocol to be applied across all four datasets. Table 1 summarizes the processed train/validation splits used in X-MV3D. Additional preprocessing details for each dataset, full category mappings, more statistics, and a detailed description of the data interface are provided in Appendix A.

3.2 Evaluation Protocol

As shown in Fig. 1(b), we define three evaluation settings. The primary setting in X-MV3D is leave-one-dataset-out (LODO) transfer. For each target dataset t, models are trained on the remaining three datasets and evaluated on t without using its training annotations. This directly measures whether existing public scene datasets provide reusable supervision for a new acquisition pipeline. We complement LODO with two reference settings. In *single-source transfer*, a model is trained on one non-target dataset and evaluated on the target, which exposes pairwise transfer asymmetry. In *target-included training*, the model is trained on all datasets including the target and evaluated with the same target-side evaluation classes and box metric. These rows are reported as target-seen references rather than target-unseen transfer results. Following prior work Zhang et al. (2025b), we evaluate 3D detection using mean average precision at IoU thresholds 0.25 and 0.50, denoted as mAP@0.25 and mAP@0.50. Results are reported per target dataset and as a macro average across the four targets. This protocol allows us to separate three distinct questions: how asymmetric transfer is across datasets, how much source diversity helps when the target is excluded, and how much is gained once the model is exposed to the target distribution.

4 X-MV3D EVALUATION

4.1 EVALUATED DETECTOR FAMILIES

We evaluate three trainable multi-view detector families in the main paper. ImVoxelNet provides a direct image-to-voxel baseline, ImGeoNet augments voxel features with image-induced geometry, and SGCDet combines geometry-aware multi-view aggregation with adaptive sparse volume construction. All experiments are conducted on 8 NVIDIA A800 GPUs. Each single-source model is trained for 6,000 optimization steps, while each mixed-source model, including the LODO and target-included settings, is trained for 10,000 steps. For mixed-source training, we concatenate the datasets under the unified dataset interface. We use a fixed view budget for all trainable detectors: 40 views are uniformly sampled during training, and 40 uniformly sampled views are used for evaluation. ScanNet boxes are axis-aligned and do not provide yaw supervision, whereas ARKitScenes, HyperSim, and 3RScan use yaw-supervised 7D boxes. The unified detection head masks the yaw loss for datasets without orientation annotations. Additional implementation details are provided in Appendix B.

As complementary inference-only references, we also evaluate Boxer, Zoo3D, and SpatialLM with their released models and without benchmark-specific optimization. These methods represent different directions toward more generalizable 3D perception: Boxer trains on heterogeneous multi-device datasets and lifts open-world 2D detections into 3D boxes, Zoo3D performs training-free scene-level 3D detection through open-vocabulary mask clustering, and SpatialLM predicts structured scene representations through language-based modeling. We include them to evaluate whether such released generalist systems can be directly reused across the datasets in X-MV3D. The released Boxer model includes ScanNet in its training data, while Zoo3D and SpatialLM use target-scene point clouds constructed from ground-truth depth. Their results therefore complement, rather than constitute strictly matched counterparts to, the target-unseen LODO experiments.

4.2 DATASET CONDITIONING AND SOURCE DIVERSITY

We first examine pairwise source-to-target transfer. Figure 2 shows that in-domain performance is a poor proxy for cross-dataset transfer and that transfer is strongly asymmetric. For example, ARKitScenes is a useful source for 3RScan, whereas ScanNet transfers weakly to ARKitScenes and 3RScan despite strong in-domain performance. The same qualitative pattern appears across SGCDet, ImGeoNet, and ImVoxelNet, suggesting that dataset conditioning is a detector-family-level phenomenon rather than a peculiarity of one architecture. Full source-to-target transfer matrices are provided in Appendix C.6.

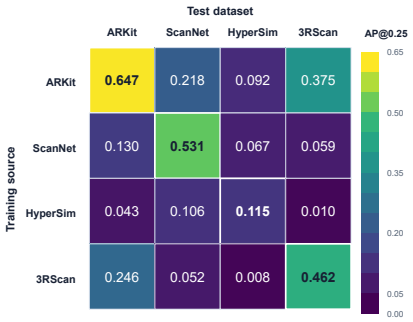


Figure 2: Single-source SGCDet transfer (mAP@0.25): training datasets in rows and test datasets in columns.

Table 2 summarizes the transfer performance under three settings: the best single-source transfer, leave-one-dataset-out (LODO) transfer, and target-included training. These settings respectively measure the effect of source diversity without target annotations and target exposure.

First, source diversity helps: LODO usually improves over the best single source, showing that multi-source training learns useful shared geometry and object priors. For SGCDet, LODO improves over the best single source on ARKitScenes ($0.246 \rightarrow 0.369$), ScanNet ($0.218 \rightarrow 0.318$), and 3RScan ($0.375 \rightarrow 0.415$). Similar gains are observed for ImGeoNet on ARKitScenes ($0.172 \rightarrow 0.264$) and ScanNet ($0.170 \rightarrow 0.229$), and for ImVoxelNet on ARKitScenes ($0.168 \rightarrow 0.293$) and ScanNet ($0.067 \rightarrow 0.132$). For SGCDet, the improvement also holds under a matched 6k-step training budget (Appendix C.5).

Second, multi-source training is not sufficient: LODO remains far below target-included training. For SGCDet, the gap is especially large on ARKitScenes (0.369 vs. 0.651) and ScanNet (0.318 vs. 0.520), and remains clear in the macro average (0.310 vs. 0.465). The same pattern holds for

Table 2: Transfer ladder on X-MV3D, reported as mAP@0.25/mAP@0.50. Best single-source selects the non-target single-source model with the highest mAP@0.25. The Macro row averages these target-wise results. LODO trains on the other three datasets without target annotations, while Target included uses all four datasets and serves as a target-seen reference.

Detector	Target	Best single-source	LODO	Target included
SGCDet	ARKitScenes	0.246 / 0.024	0.369 / 0.059	0.651 / 0.475
	ScanNet	0.218 / 0.068	0.318 / 0.127	0.520 / 0.195
	HyperSim	0.092 / 0.021	0.141 / 0.047	0.143 / 0.058
	3RScan	0.375 / 0.214	0.415 / 0.178	0.547 / 0.325
	Macro	0.233 / 0.082	0.310 / 0.103	0.465 / 0.263
ImGeoNet	ARKitScenes	0.172 / 0.017	0.264 / 0.036	0.620 / 0.403
	ScanNet	0.170 / 0.036	0.229 / 0.049	0.414 / 0.147
	HyperSim	0.059 / 0.006	0.077 / 0.037	0.096 / 0.050
	3RScan	0.351 / 0.172	0.362 / 0.173	0.464 / 0.233
	Macro	0.188 / 0.058	0.233 / 0.074	0.398 / 0.208
ImVoxelNet	ARKitScenes	0.168 / 0.010	0.293 / 0.040	0.595 / 0.370
	ScanNet	0.067 / 0.006	0.132 / 0.030	0.399 / 0.167
	HyperSim	0.054 / 0.008	0.046 / 0.019	0.080 / 0.029
	3RScan	0.363 / 0.162	0.384 / 0.163	0.476 / 0.265
	Macro	0.163 / 0.047	0.214 / 0.063	0.387 / 0.208

Table 3: Results of inference-only generalist systems. Values are mAP@0.25/mAP@0.50. Boxer training includes ScanNet, whereas Zoo3D and SpatialLM use depth at inference.

Method	ARKit	ScanNet	HyperSim	3RScan	Macro
Boxer	0.125/0.102	0.335/0.197	0.212/0.131	0.055/0.055	0.182/0.121
Zoo3D	0.213/0.086	0.328/0.204	0.163/0.105	0.212/0.079	0.229/0.118
SpatialLM	0.116/0.028	0.162/0.069	0.100/0.061	0.152/0.106	0.133/0.066

ImGeoNet (0.233 vs. 0.398 macro) and ImVoxelNet (0.214 vs. 0.387 macro). These gaps show that although diverse source datasets help, current multi-view detectors still fall well short of the performance when target training data is available.

Can released generalists avoid dataset conditioning? The transfer ladder above evaluates detectors retrained on the available source datasets. Table 3 asks a complementary question: how well can released generalist systems transfer directly to X-MV3D without benchmark-specific retraining? We evaluate the released Boxer model, the training-free Zoo3D configuration, and the language-based SpatialLM.

The released generalist systems produce non-trivial detections, but none generalizes consistently across all four datasets. Boxer achieves its highest performance on ScanNet but drops substantially on the other targets, including $0.055/0.055$ on 3RScan. Zoo3D obtains the most balanced macro result among the three systems at $0.229/0.118$, yet its macro mAP@0.25 remains below the SGCDet LODO result of 0.310. SpatialLM reaches a macro score of $0.133/0.066$ and similarly exhibits limited performance across targets. Overall, existing released generalist systems provide useful capabilities for direct reuse, but do not yet achieve reliable cross-dataset transfer across heterogeneous scene datasets.

4.3 Transfer Failure Analysis

Category and box controls. We restrict evaluation to the nine categories shared by all datasets and convert both predictions and ground truth to horizontal axis-aligned bounding boxes (AABBs), without retraining. Table 4 shows that target-included training still exceeds LODO by 0.155, 0.190, and 0.205 macro mAP@0.25 for SGCDet, ImGeoNet, and ImVoxelNet, respectively. Thus, using the same evaluation categories and box representation does not remove the transfer gap.

Semantic errors and proposal coverage. We compare LODO and target-included models to examine how detection failures differ when target-domain training data are unavailable (Table 5).

Table 4: Common-9 with AABBs at evaluation. Scores are four-target macro mAP@0.25/mAP@0.50; the gap is target-included minus LODO mAP@0.25.

Detector	LODO	Target-included	Gap (AP25)
SGCDet	0.414 / 0.155	0.569 / 0.312	0.155
ImGeoNet	0.314 / 0.086	0.504 / 0.251	0.190
ImVoxelNet	0.279 / 0.090	0.484 / 0.244	0.205

Table 5: Failure diagnostics. All entries are four-target macro averages. The label oracle is a relabeling diagnostic. R@500 is class-agnostic proposal recall.

Detector	Training	AP25/50	Label oracle AP25	R@500 25/50
SGCDet	LODO	0.310 / 0.103	0.346	0.518 / 0.224
	Target-included	0.465 / 0.263	0.492	0.625 / 0.383
ImGeoNet	LODO	0.233 / 0.074	0.277	0.448 / 0.159
	Target-included	0.398 / 0.208	0.406	0.565 / 0.310
ImVoxelNet	LODO	0.214 / 0.063	0.237	0.434 / 0.148
	Target-included	0.387 / 0.208	0.403	0.585 / 0.311

Target-included models serve as target-seen references. We first apply a label oracle that assigns each prediction the category of its best-overlap ground truth while preserving predicted boxes and scores. For SGCDet, relabeling raises LODO macro mAP@0.25 from 0.310 to 0.346, but a substantial gap remains when both settings are relabeled (0.346 vs. 0.492). Thus, this label-only intervention does not remove the performance gap. We next measure R@500, which evaluates GT coverage by the top-500 predictions while ignoring category labels during matching. LODO has lower R@500 than target-included training for all three detectors at both IoU thresholds. For example, SGCDet achieves $0.518/0.224$ under LODO, compared with $0.625/0.383$ under target-included training at IoU $0.25/0.50$. These results show that **insufficient proposal coverage is severe under target-unseen transfer** across detector families.

Localization within existing proposals. To further separate localization errors among existing proposals, we apply box-parameter oracles to predictions with pre-edit IoU ≥ 0.10. Replacing predicted centers with ground-truth centers raises macro mAP@0.50 from 0.103 to 0.187, whereas replacing box size alone yields only 0.143 mAP@0.50 and no consistent mAP@0.25 improvement. Thus, **center localization is a more prominent bottleneck than box size** among existing proposals.

Attribution to acquisition shifts. Multi-view 3D detectors rely on camera geometry to lift image features into 3D and aggregate observations across views. We hypothesize that changes in camera parameters and view availability contribute to transfer failures. We probe this hypothesis through two controlled in-domain experiments: jointly cropping and resampling images with corresponding intrinsic updates, and changing the view sampling distribution at a fixed view budget. These test-time interventions mimic aspects of the camera and view changes encountered across datasets. For SGCDet trained and evaluated on ScanNet, the camera and view sampling perturbations reduce mAP@0.25 by 8.91 and 4.23 points, respectively. These results support acquisition-side changes as a contributor to detection failures.

Implications for detector design. These findings motivate addressing acquisition shifts at the shared image-to-3D lifting and aggregation. We target two complementary aspects: the geometric context of individual observations and the availability of observations across views. We condition image features on camera geometry, making the acquisition context explicit before lifting. We further apply stochastic view gating during training, randomly retaining view subsets subject to a minimum-view constraint. This regularizer exposes the aggregation module to varying observation subsets, encouraging it to rely less on dataset-specific acquisition patterns.

5 CamGeoGate

Building on previous analysis, we propose CamGeoGate, a lightweight reference baseline designed to reduce detectors' dependence on dataset-specific camera and view configurations. CamGeoGate

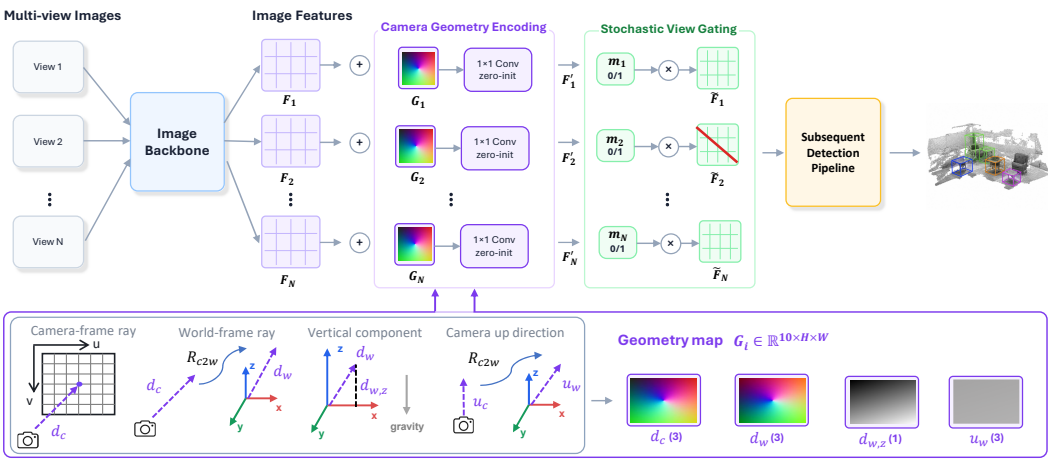


Figure 3: Overview of CamGeoGate. The framework encodes camera geometry into per-view image features through a zero-initialized residual adapter and applies stochastic view gating during training. The geometry map contains camera-frame rays, world-frame rays, their gravity-aligned components, and camera up directions.

targets two acquisition-related factors, camera geometry and view availability, by explicitly injecting camera geometry into image features and exposing the detector to varying view subsets during training. The former makes the geometric conditions of each observation directly available to the image features, while the latter discourages reliance on a fixed view configuration. CamGeoGate requires no target annotations or additional 3D supervision.

Camera geometry encoding. CamGeoGate explicitly encodes camera geometry into image features, including camera intrinsic information, camera-to-world rotation, its gravity alignment, and camera up direction in the shared scene coordinate system. For each input view i and location (u, v) on the feature grid, we first compute the corresponding camera-frame ray:

$$\mathbf{d}_c^i(u,v) = \mathrm{norm}\left(\mathbf{K}_i^{-1}[u,v,1]^\top\right), \tag{1}$$

where $\mathbf{K}_i$ is the camera intrinsic matrix after image transformations. This ray encodes how the camera observes each image location according to its intrinsic parameters.

We then transform the camera-frame ray into the shared gravity-aligned scene coordinate system:

$$\mathbf{d}_w^i(u,v) = \mathbf{R}_{c2w}^i \mathbf{d}_c^i(u,v), \tag{2}$$

where $\mathbf{R}_{c2w}^i$ denotes the camera-to-world rotation. This provides the viewing direction of each image location under the global scene orientation. We further extract its gravity-aligned vertical component:

$$d_{w,z}^i(u,v) = \mathbf{d}_w^i(u,v)[2]. \tag{3}$$

In addition, we encode the camera up direction in the shared scene coordinate system:

$$\mathbf{u}_w^i = \mathbf{R}_{c2w}^i[0,-1,0]^\top. \tag{4}$$

The camera up direction is constant within each view and is broadcast to all feature locations.

The four geometric cues are concatenated into a 10-dimensional geometry vector:

$$\mathbf{g}_i(u,v) = [\mathbf{d}_c^i(u,v); \mathbf{d}_w^i(u,v); d_{w,z}^i(u,v); \mathbf{u}_w^i] \in \mathbb{R}^{10}. \tag{5}$$

The resulting geometry map provides explicit camera intrinsics, camera-to-world rotation, its gravity alignment, and camera up direction for each input view.

Let $\mathbf{F}_i \in \mathbb{R}^{C\times H\times W}$ denote the image feature map and $\mathbf{G}_i \in \mathbb{R}^{10\times H\times W}$ the geometry map. We fuse them using a residual adapter:

$$\mathbf{F}_i' = \mathbf{F}_i + \mathcal{A}(\mathbf{G}_i), \qquad \mathcal{A} = \mathrm{Conv}_{1\times 1}(10, C). \tag{6}$$

The weights and bias of $\mathcal{A}$ are initialized to zero. Therefore, CamGeoGate starts from the original detector and gradually learns the geometry residual during optimization.

Table 6: Per-target LODO results across detector families. Each cell reports mAP@0.25/mAP@0.50.

Detector	Setting	ARKitScenes	ScanNet	HyperSim	3RScan	Macro
SGCDet	Baseline	0.369 / 0.059	0.318 / 0.127	0.141 / 0.047	0.415 / 0.178	0.310 / 0.103
	$+CamGeoGate$	**0.389 / 0.063**	**0.340 / 0.148**	**0.167 / 0.064**	**0.455** / 0.178	**0.338 / 0.114**
	Δ	+0.020 / +0.004	+0.022 / +0.021	+0.026 / +0.017	+0.040 / +0.000	**+0.028 / +0.011**
ImGeoNet	Baseline	0.264 / 0.036	0.229 / 0.049	0.077 / 0.037	0.362 / 0.173	0.233 / 0.074
	$+CamGeoGate$	**0.318 / 0.042**	0.229 / **0.062**	**0.080 / 0.040**	**0.389 / 0.233**	**0.254 / 0.095**
	Δ	+0.054 / +0.006	+0.000 / +0.013	+0.003 / +0.003	+0.027 / +0.060	**+0.021 / +0.021**
ImVoxelNet	Baseline	0.293 / **0.040**	0.132 / 0.030	0.046 / 0.019	0.384 / 0.163	0.214 / 0.063
	$+CamGeoGate$	**0.316** / 0.039	**0.174 / 0.032**	**0.068 / 0.036**	**0.433 / 0.211**	**0.248 / 0.079**
	Δ	+0.023 / -0.001	+0.042 / +0.002	+0.022 / +0.017	+0.049 / +0.048	**+0.034 / +0.016**

Stochastic view gating. During training, we sample a binary gate $m_i \in \{0, 1\}$ for each of the N input views, subject to a minimum-view constraint:

$$\sum_{i=1}^{N} m_i \geq N_{\min}. \tag{7}$$

In all experiments, we set $N_{\min} = 16$. The gated feature for view i is

$$\widetilde{\mathbf{F}}_i = m_i \mathbf{F}'_i. \tag{8}$$

The same gate is applied to all inputs associated with that view before lifting and multi-view aggregation during training. By exposing the detector to different subsets of cameras, stochastic view gating reduces its reliance on the fixed view density and capture trajectory of the source datasets. Implementation and sampling details are provided in Appendix B.3.

5.1 Method Results

CamGeoGate improves macro LODO metrics across all three detector families (Table 6): SGCDet from $0.310/0.103$ to $0.338/0.114$, ImGeoNet from $0.233/0.074$ to $0.254/0.095$, and ImVoxelNet from $0.214/0.063$ to $0.248/0.079$. mAP@0.25 improves in 11 of the 12 detector–target pairs, indicating benefits across architectures and targets. Appendix C.3 reports per-class gains under Common-9.

Ablation study. On SGCDet, geometry encoding alone improves macro LODO performance from $0.310/0.103$ to $0.322/0.115$, while view gating alone reaches $0.328/0.110$. Their combination achieves $0.338/0.114$. Both components improve mAP@0.25, with the combination performing best.

Limitations. Although CamGeoGate consistently improves target-unseen transfer, its performance remains substantially below target-included training, especially at the stricter IoU threshold. This indicates that camera geometry encoding and stochastic view gating address only part of the cross-dataset shift. Developing a 3D detector that can be trained on existing detection datasets and reliably transferred zero-shot to a newly collected dataset therefore remains an open problem.

6 Conclusion

We presented X-MV3D, a cross-dataset benchmark for scene-level 3D detection across ARKitScenes, ScanNet, HyperSim, and 3RScan. Current detectors show strong in-domain performance but substantially weaker target-unseen transfer. Multi-source training improves over single-source transfer, yet remains far from target-included performance. Our diagnostics show that this gap is dominated by insufficient proposal coverage and inaccurate center localization. We regard acquisition shifts as important factors affecting multi-view feature lifting and aggregation. Motivated by this analysis, we propose CamGeoGate, a lightweight baseline that combines explicit camera geometry encoding with stochastic view gating and consistently improves LODO macro performance across SGCDet, ImGeoNet, and ImVoxelNet. The remaining gap indicates that developing reliable and reusable 3D detectors for unseen datasets remains an open research problem.

AI USE STATEMENT

We used generative AI tools to assist with coding and idea implementation, and manuscript preparation and polishing. The core research ideas were developed by the authors, and the authors are responsible for the final content.

ETHICS STATEMENT

This work uses existing ARKitScenes, ScanNet, HyperSim, and 3RScan datasets without new participant recruitment or data collection. Users should obtain the source data from their official distributions and follow the respective access and use conditions.

REPRODUCIBILITY STATEMENT

Appendix A documents the dataset preprocessing, category mappings, data interface, and evaluation protocols. Appendix B describes the detector configurations, software environments, random seeds, and CamGeoGate implementation. Appendix C provides additional experiments and diagnostics.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7822–7831, 2021.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS Datasets and Benchmarks*, 2021.

Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3d object detection in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Gyusam Chang, Jiwon Lee, Donghyun Kim, Jinkyu Kim, Dongwook Lee, Daehyun Ji, Sujin Jang, and Sangpil Kim. Unified domain generalization and adaptation for multi-view 3d object detection. In *Advances in Neural Information Processing Systems*, 2024.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel. Boxer: Robust lifting of open-world 2d bounding boxes to 3d, 2026.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE International Conference on Robotics and Automation*, pp. 5021–5028, 2024.

Peng-Hao Hsu, Ke Zhang, Fu-En Wang, Tao Tu, Ming-Feng Li, Yu-Lun Liu, Albert Y. C. Chen, Min Sun, and Cheng-Hao Kuo. OpenM3D: Open vocabulary multi-view indoor 3d object detection without human annotations. In *IEEE/CVF International Conference on Computer Vision*, 2025.

Jin-Cheng Jhang, Tao Tu, Fu-En Wang, Ke Zhang, Min Sun, and Cheng-Hao Kuo. V-MIND: Building versatile monocular indoor 3d detector with diverse 2d annotations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 9559–9568, 2025. doi: 10.1109/WACV61041.2025.00927.

Maksim Kolodiazhnyi, Anna Vorontsova, Matvey Skripkin, Danila Rukhovich, and Anton Konushin. UniDet3D: Multi-dataset indoor 3d object detection. In *AAAI Conference on Artificial Intelligence*, 2025.

Yuqing Lan, Chenyang Zhu, Zhirui Gao, Jiazhao Zhang, Yihan Cao, Renjiao Yi, Yijie Wang, and Kai Xu. BoxFusion: Reconstruction-free open-vocabulary 3d object detection via real-time multi-view box fusion. *Computer Graphics Forum*, 2025. doi: 10.1111/cgf.70254.

Andrey Lemeshko, Bulat Gabdullin, Nikita Drozdov, Anton Konushin, Danila Rukhovich, and Maksim Kolodiazhnyi. Zoo3D: Zero-shot 3d object detection at scene level. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zhuoling Li, Xiaogang Xu, SerNam Lim, and Hengshuang Zhao. UniMODE: Unified monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16561–16570, 2024.

Fulong Ma, Xiaoyang Yan, Guoyang Zhao, Xiaojie Xu, Yuxuan Liu, Jun Ma, and Ming Liu. Every dataset counts: Scaling up monocular 3d object detection with joint datasets training. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 11574–11580, 2024. doi: 10.1109/IROS58592.2024.10802623.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *IEEE/CVF International Conference on Computer Vision*, pp. 20133–20143, 2023.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2397–2406, 2022.

Tiril Sundby, Julia Maria Graham, Adil Rasheed, Mandar Tabib, and Omer San. Geometric change detection in digital twins. *Digital*, 1(2):111–129, 2021. doi: 10.3390/digital1020009.

Tomu Tahara, Takashi Seno, Gaku Narita, and Tomoya Ishikawa. Retargetable AR: Context-aware augmented reality in indoor scenes based on 3d scene graph. In *IEEE International Symposium on Mixed and Augmented Reality Adjunct*, pp. 249–255, 2020.

Tao Tu, Shun-Po Chuang, Yu-Lun Liu, Cheng Sun, Ke Zhang, Donna Roy, Cheng-Hao Kuo, and Min Sun. Imgeonet: Image-induced geometry-aware voxel representation for multi-view 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Niessner. Rio: 3d object instance re-localization in changing indoor environments. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Shuo Wang, Xinhai Zhao, Hai-Ming Xu, Zehui Chen, Dameng Yu, Jiahao Chang, Zhen Yang, and Feng Zhao. Towards domain generalization for multi-view 3d object detection in bird-eye-view. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13333–13342, 2023.

Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. EmbodiedScan: A holistic multi-modal 3d perception suite towards embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19757–19767, 2024.

Yating Xu, Chen Li, and Gim Hee Lee. Mvsdet: Multi-view indoor 3d object detection via efficient plane sweeps. In *Advances in Neural Information Processing Systems*, 2024.

Yung-Hsu Yang, Luigi Piccinelli, Mattia Segu, Siyuan Li, Rui Huang, Yuqian Fu, Marc Pollefeys, Hermann Blum, and Zuria Bauer. 3D-MOOD: Lifting 2d to 3d for monocular open-set object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7429–7439, 2025.

Hanxue Zhang, Haoran Jiang, Qingsong Yao, Yanan Sun, Renrui Zhang, Hao Zhao, Hongyang Li, Hongzi Zhu, and Zetong Yang. Detect anything 3d in the wild. In *IEEE/CVF International Conference on Computer Vision*, 2025a.

Runmin Zhang, Zhu Yu, Si-Yuan Cao, Lingyu Zhu, Guangyi Zhang, Xiaokai Bai, and Hui-Liang Shen. Boosting multi-view indoor 3d object detection via adaptive 3d volume construction. In *IEEE/CVF International Conference on Computer Vision*, 2025b.

Liangtao Zheng, Yicheng Liu, Yue Wang, and Hang Zhao. Cross-dataset sensor alignment: Making visual 3d object detector generalizable. In *Proceedings of The 7th Conference on Robot Learning*, pp. 1903–1929, 2023.

A Dataset Construction and Protocol

A.1 Released Data Representation

We release one train and one validation metadata file for each dataset. Images and depth maps remain as ordinary files, while the metadata stores relative paths and all information required to construct a multi-view detection example. The release is organized as follows:

```
X-MV3D/
  {arkit, scannet, hypersim, 3rscan}/
    DATASET_infos_train.pkl
    DATASET_infos_val.pkl
    RGB/depth assets and scene folders
```

Each PKL file is a Python list whose entries follow the same logical record format:

```
records = [record, ...]
record = {
  record_id:  str,
  rgb_paths:  str[V],
  depth_paths:  str[V],
  intrinsics:  float[V,3,3],
  camera_poses:  float[V,4,4],
  annotations:  {
    boxes_3d:  float[M,6/7],
    labels:  int[M]
  }
}
```

Here V and M are the numbers of retained views and annotated boxes. A record corresponds to an ARKitScenes video, a ScanNet scan, a HyperSim scene–camera trajectory, or a 3RScan scan. Camera poses and boxes use the same metric, gravity-aligned world frame with vertical $+Z$. Oriented boxes are stored as $[c_x, c_y, c_z, d_x, d_y, d_z, \theta_z]$, where θ_z is yaw about $+Z$; ScanNet retains its native axis-aligned 6D boxes. The loader maps labels to NYU40 and samples views without regenerating the metadata.

A.2 Dataset-Specific Preprocessing

ARKitScenes. For ARKitScenes (Baruch et al., 2021), we use the official 3D object detection train/validation videos and their low-resolution wide RGB, LiDAR depth, trajectories, calibration, and `obbAligned` annotations. RGB, depth, and poses are associated by timestamp; invalid poses are removed, and we retain at most 50/100 frames per train/validation video. Depth is converted from millimeters to meters. ARKitScenes is already gravity aligned, so its camera poses and oriented 7D boxes require no additional world rotation.

ScanNet. For ScanNet (Dai et al., 2017), we use the official 1,201/312 train/validation scans, posed RGB-D frames, calibration, axis-alignment matrices, and standard object-detection instance annotations. Frames are ordered temporally, invalid poses are removed, and at most 50/100 frames are retained per train/validation scan. The provided axis-alignment transform is applied consistently to the camera poses and boxes. We preserve ScanNet's released axis-aligned 6D boxes and do not infer yaw from images or meshes.

HyperSim. For HyperSim (Roberts et al., 2021), we follow the public-release split and treat each scene–camera trajectory as one record, retaining at most 50 frames. We use tone-mapped RGB, metric depth, rendered camera parameters, and object-aligned boxes. Camera positions and box extents are converted with the provided meters-per-asset-unit scale, and the camera axes are transformed to the common convention. HyperSim depth is retained as metric distance along the camera ray; adapters requiring pinhole z-depth explicitly perform the ray-to-z conversion. Semantic and instance IDs are joined to obtain NYU40 object labels, and invalid or unlabeled instances are discarded.

3RScan. For 3RScan (Wald et al., 2019), we use the released scan split and retain 1,266 training and 42 validation scans after validity checks. RGB, depth, and poses are matched by frame identifier, and at most 50/60 frames are retained per train/validation scan. JPEG orientation is applied identically to RGB and depth, with the corresponding update to the intrinsic matrix. The scan poses and OBB annotations already share a z-up frame; we apply the same horizontal heading normalization to both and convert the provided OBBs to the common upright 7D representation while preserving their oriented cuboid geometry.

A.3 Category Harmonization

We normalize the original ScanNet, ARKitScenes, HyperSim, and 3RScan category names to the NYU40 naming convention. Tables 9 and 10 list every retained mapping for the datasets.

A.4 Dataset Statistics

Table 11 summarizes the released train and validation metadata after conversion. Counts refer to record-level annotation occurrences before category-specific metric aggregation. Empty records are retained in this audit but are not sampled by the training loaders.

The target validation sets are also class imbalanced within the fixed dataset categories listed below. HyperSim has only two toilets and four bathtubs, while 3RScan has one bathtub; per-class results and the identical Common-9 control are therefore reported in Appendix C.

A.5 Benchmark Protocol

Splits, views, and boxes. We follow the official ARKitScenes and ScanNet 3D-detection splits and annotations, the public HyperSim split and NYU40 semantic labels, and the released 3RScan split and OBB annotations. No converted record appears in both training and validation. The standard input contains 40 views: training uses approximately uniform temporal samples with small index jitter, and evaluation uses deterministic linear sampling. If a record contains fewer than 40 retained frames, frames are repeated to preserve the input shape. Captured uint16 depth is converted from millimeters to meters, while HyperSim retains its native metric ray distance.

We preserve the annotation geometry supplied by each benchmark: ARKitScenes, HyperSim, and 3RScan use yaw-supervised upright 7D boxes, whereas ScanNet uses its official axis-aligned boxes and does not supervise yaw. Following standard indoor 3D-detection practice in ScanNet and ARKitScenes, we report class-aware 3D AP at IoU thresholds 0.25 and 0.50.

Dataset categories. After direct NYU40 normalization, the benchmark contains 11 ARKitScenes, 16 ScanNet, 18 HyperSim, and 11 3RScan object categories, listed in Table 7. The lists retain unambiguous object categories and omit room-envelope and non-specific `other*` labels. HyperSim requires no semantic remapping; categories without validation GT or without training examples in the three non-target datasets are omitted at this stage. No category is selected using its AP or a post-hoc frequency threshold.

The common nine-category control in Appendix C applies the identical target taxonomy to all four datasets and therefore removes the remaining target-specific category-set difference.

B Implementation and Evaluation Details

B.1 Training Setup

We follow the released configurations of SGCDet (Zhang et al., 2025b), ImGeoNet (Tu et al., 2023), and ImVoxelNet (Rukhovich et al., 2022), including their architectures, optimization, losses, and inference settings; no hyperparameter is tuned for an individual target dataset. ImGeoNet retains its auxiliary occupancy supervision, whereas ImVoxelNet does not use occupancy or depth-map supervision.

Within each detector family, all variants use the same fixed seed (1234 for SGCDet and 0 for ImGeoNet and ImVoxelNet). Training uses eight NVIDIA A800 80 GB GPUs with one scene per

Table 7: Fixed object categories for each dataset after direct NYU40 normalization.

Dataset (classes)	Object categories
ARKitScenes (11)	bathtub, bed, cabinet, chair, refrigerator, shelves, sink, sofa, table, television, toilet
ScanNet (16)	bathtub, bed, bookshelf, cabinet, chair, counter, curtain, desk, door, picture, refrigerator, sink, sofa, table, toilet, window
HyperSim (18)	bathtub, bed, bookshelf, cabinet, chair, counter, curtain, desk, door, picture, refrigerator, shelves, sink, sofa, table, television, toilet, window
3RScan (11)	bathtub, bed, cabinet, chair, refrigerator, shelves, sink, sofa, table, television, toilet

Table 8: Validation annotation statistics on the fixed dataset categories. Min. and median are computed over per-class GT counts.

Target	GT boxes	Classes	Min.	Median	Dominant class (count; share)
ARKitScenes	5,331	11	82	281	cabinet (2,255; 42.3%)
ScanNet	3,806	16	31	97.5	chair (1,368; 35.9%)
HyperSim	1,893	18	2	31.5	chair (763; 40.3%)
3RScan	354	11	1	13	chair (143; 40.4%)

GPU, and evaluation uses one GPU. SGCDet uses PyTorch 1.11 with CUDA 11.3; ImGeoNet and ImVoxelNet use PyTorch 1.7.1 with CUDA 11.0, following their released codebases.

Source-only models train 6,000 steps. LODO and target-included models train 10,000 steps. Appendix C reports matched-budget and convergence controls.

B.2 Unified Training and Evaluation Interface

All detectors use 40 views per scene, sampled with temporal jitter during training and linearly at test time. ARKitScenes, HyperSim, and 3RScan supervise 7D boxes with yaw. ScanNet supervises axis-aligned 6D boxes; its yaw loss is masked while the remaining box and class losses are retained.

Evaluation follows SGCDet and reports mAP@0.25 and mAP@0.50 on the dataset categories in Table 7. The macro score is the unweighted mean over the four targets.

B.3 CamGeoGate Architecture and Training

At each location of the first FPN level, we concatenate the normalized camera-frame ray (3 channels), world-frame ray (3), its vertical component (1), and the camera up direction in the scene frame (3). A zero-initialized 1×1 convolution projects this descriptor to the image-feature dimension, and the residual is added before image-to-volume lifting.

For view gating, we sample $p \sim \mathcal{U}(0, 0.5)$ and retain each of the 40 training views with probability $1 - p$, with at least 16 views retained. The same sampled subset is used throughout lifting and aggregation. Testing always uses the deterministic 40-view sample.

C Additional Experiments and Diagnostics

C.1 Taxonomy-Controlled Evaluation

The common intersection contains *bathtub, bed, cabinet, chair, refrigerator, sink, sofa, table, toilet*. Table 12 fixes these nine classes for all targets. LODO still exceeds the target-wise best single source, while target-included training remains substantially stronger.

Joint taxonomy–box control. Converting both predictions and GT to horizontal AABBs on Common-9 changes the macro from $0.419/0.149$ to $0.414/0.155$ (Table 13); taxonomy and yaw convention therefore do not explain the transfer gap.

Table 9: Category mappings for ARKitScenes, ScanNet, and HyperSim.

Dataset	Original category → final category
ARKitScenes	cabinet→cabinet; refrigerator→refrigerator; shelf→shelves; bed→bed; sink→sink; toilet→toilet; bathtub→bathtub; chair→chair; table→table; tv_monitor→television; sofa→sofa.
ScanNet	cabinet→cabinet; bed→bed; chair→chair; sofa→sofa; table→table; door→door; window→window; bookshelf→bookshelf; picture→picture; counter→counter; desk→desk; curtain→curtain; refrigerator→refrigerator; toilet→toilet; sink→sink; bathtub→bathtub.
HyperSim	cabinet→cabinet; bed→bed; chair→chair; sofa→sofa; table→table; door→door; window→window; bookshelf→bookshelf; picture→picture; counter→counter; desk→desk; shelves→shelves; curtain→curtain; fridge→refrigerator; tv→television; toilet→toilet; sink→sink; bathtub→bathtub.

Table 10: Category mappings for 3RScan.

Original category or categories	Final category
chair	chair
table	table
sofa	sofa
bed	bed
cabinet, night stand, dresser, commode	cabinet
sink	sink
toilet	toilet
bathtub	bathtub
refrigerator	refrigerator
shelf, shelves	shelves
tv, television, monitor	television

C.2 Common-9 Failure Decomposition

Table 14 fixes Common-9 across targets. Following the main paper, the label oracle changes only predicted categories, while proposal R@500 ignores labels and measures GT coverage by the top 500 predictions.

Common-9 raises absolute AP and recall but leaves the diagnosis unchanged: the label oracle adds only 0.014 macro AP25, while R@500 is $0.604/0.282$. The label oracle is diagnostic rather than an upper bound because it can also relabel low-IoU false positives.

C.3 Category-Wise CamGeoGate Gains on Common-9

Table 15 compares LODO detectors with and without CamGeoGate on the nine shared categories. Each class entry is the equal-weight mean of its AP across the four target datasets; the Macro row then averages the nine classes. Gain denotes CamGeoGate minus baseline AP. We reuse the existing checkpoints and predictions, restricting evaluation to Common-9 without changing box parameterization. CamGeoGate includes geometry conditioning and training-time view drop.

C.4 Yaw-Free Box-Convention Control

Converting predictions and GT to horizontal AABBs changes SGCDet LODO macro AP only from $0.310/0.103$ to $0.308/0.108$ (Table 16). Yaw convention affects individual targets but does not explain the transfer gap.

C.5 Training-Budget and Convergence Controls

We compare 6k-step source-only models with the 6k-step LODO checkpoints. LODO still improves macro AP from $0.233/0.082$ to $0.307/0.106$; its 10k final result is $0.310/0.103$ (Table 17). The source-diversity gain therefore does not result from the larger LODO budget.

A 4k-step continuation at learning rate 2×10^{-5} from the 10k weights produces $0.305/0.103$, $0.313/0.102$, $0.310/0.108$, and finally $0.306/0.110$, versus $0.310/0.103$ before continuation. The absence of a sustained gain supports treating 10k as converged.

Table 11: Statistics of the released benchmark metadata. F/rec. and B/rec. denote the average numbers of retained frames and annotated boxes per record.

Dataset	Split	Records	Frames	F/rec.	Boxes	B/rec.	Empty
ARKitScenes	train	4,492	224,227	49.9	50,618	11.3	0
	val	549	54,900	100.0	5,962	10.9	0
ScanNet	train	1,201	58,452	48.7	15,733	13.1	2
	val	312	30,478	97.7	4,364	14.0	0
HyperSim	train	614	30,435	49.6	61,412	100.0	52
	val	77	3,808	49.5	4,587	59.6	2
3RScan	train	1,266	62,064	49.0	7,960	6.3	0
	val	42	2,110	50.2	361	8.6	0
Total	train	7,573	375,178	–	135,723	–	54
	val	980	91,296	–	15,274	–	2

Table 12: Common-9 transfer ladder (four-target macro mAP@0.25 / mAP@0.50). Best single source is selected per target by mAP@0.25.

Detector	Best single	LODO	Target included
SGCDet	0.328/0.114	0.419/0.149	0.574/0.340
ImGeoNet	0.262/0.077	0.316/0.101	0.504/0.272
ImVoxelNet	0.214/0.059	0.281/0.085	0.488/0.271

C.6 Complete Single-Source Transfer Matrices

Table 18 reports single-source transfer matrices; diagonal cells are in-domain.

C.7 Boxer Adapter and Fusion Audit

HyperSim stores ray distance, which we convert to pinhole z-depth before lifting. On 3RScan, only toilet has non-zero AP; its three matches all exceed IoU 0.5, explaining $0.055/0.055$. Coordinate checks rule out an axis or pose error, and we retain the released fusion threshold because lowering it reduces the four-target macro.

C.8 Box-Parameter Oracle

The box oracle in Table 19 attributes most recoverable AP50 error to center localization; neither edit recovers missing proposals.

C.9 Qualitative Analysis

Figure 4 shows one selected scene per target. Gray dashes denote GT, orange the canonical detector, blue CamGeoGate, and yellow a GT recovered by CamGeoGate at IoU 0.25. Display-only filtering uses score ≥ 0.05 and the top 100 predictions.

Table 13: SGCDet LODO on Common-9 with standard boxes or yaw-free AABBs.

Target	Standard	Common-9 AABB
ARKitScenes	0.444/0.072	0.460/0.114
ScanNet	0.512/0.215	0.515/0.216
HyperSim	0.230/0.083	0.230/0.116
3RScan	0.491/0.226	0.450/0.174
Macro	0.419/0.149	0.414/0.155

Table 14: Common-9 failure decomposition using the oracle and proposal metrics from the main paper.

Target	AP25/50	Oracle-label AP25	R@500 25/50	GT boxes
ARKitScenes	0.444/0.072	0.451	0.632/0.225	4,844
ScanNet	0.512/0.215	0.531	0.797/0.474	2,512
HyperSim	0.230/0.083	0.280	0.266/0.125	1,368
3RScan	0.491/0.226	0.468	0.722/0.303	277
Macro	0.419/0.149	0.433	0.604/0.282	–

Table 15: Common-9 per-class LODO AP25/AP50, averaged equally across four targets. Gains are absolute AP differences; negative changes are retained.

Detector	Category	LODO	+CamGeoGate	Gain
SGCDet	bathtub	0.737/0.401	0.730/0.413	-0.007/+0.011
	bed	0.707/0.069	0.706/0.110	-0.001/+0.041
	cabinet	0.161/0.042	0.155/0.029	-0.006/-0.013
	chair	0.436/0.106	0.468/0.110	+0.032/+0.005
	refrigerator	0.293/0.051	0.396/0.069	+0.103/+0.018
	sink	0.116/0.002	0.175/0.003	+0.059/+0.001
	sofa	0.329/0.161	0.376/0.182	+0.047/+0.021
	table	0.299/0.103	0.337/0.103	+0.039/0.000
	toilet	0.692/0.406	0.855/0.474	+0.162/+0.067
	Macro	0.419/0.149	0.466/0.166	+0.048/+0.017
ImGeoNet	bathtub	0.417/0.264	0.439/0.255	+0.022/-0.009
	bed	0.556/0.010	0.556/0.022	0.000/+0.012
	cabinet	0.106/0.037	0.099/0.016	-0.007/-0.022
	chair	0.338/0.064	0.363/0.077	+0.025/+0.013
	refrigerator	0.244/0.040	0.275/0.205	+0.031/+0.166
	sink	0.095/0.003	0.115/0.001	+0.021/-0.002
	sofa	0.261/0.078	0.328/0.126	+0.067/+0.047
	table	0.209/0.043	0.228/0.054	+0.019/+0.011
	toilet	0.621/0.374	0.628/0.399	+0.007/+0.025
	Macro	0.316/0.101	0.337/0.128	+0.020/+0.027
ImVoxelNet	bathtub	0.429/0.270	0.436/0.268	+0.007/-0.002
	bed	0.412/0.010	0.509/0.019	+0.097/+0.008
	cabinet	0.086/0.025	0.092/0.008	+0.007/-0.016
	chair	0.299/0.057	0.347/0.070	+0.049/+0.013
	refrigerator	0.260/0.021	0.311/0.128	+0.051/+0.107
	sink	0.053/0.000	0.107/0.003	+0.054/+0.002
	sofa	0.218/0.029	0.287/0.083	+0.069/+0.054
	table	0.185/0.024	0.233/0.041	+0.048/+0.017
	toilet	0.586/0.325	0.638/0.352	+0.051/+0.028
	Macro	0.281/0.085	0.329/0.108	+0.048/+0.023

Table 16: SGCDet LODO with standard or yaw-free boxes (mAP@0.25 / mAP@0.50).

Target	Standard	Axis aligned
ARKit	0.369/0.059	0.393/0.095
ScanNet	0.318/0.127	0.323/0.128
HyperSim	0.141/0.047	0.141/0.068
3RScan	0.415/0.178	0.378/0.142
Macro	0.310/0.103	0.308/0.108

Table 17: SGCDet budget control (mAP@0.25/mAP@0.50).

Target	Best single source (6k)	LODO (6k)	LODO (10k)
ARKitScenes	0.246/0.024	0.373/0.071	0.369/0.059
ScanNet	0.218/0.068	0.306/0.128	0.318/0.127
HyperSim	0.092/0.021	0.133/0.065	0.141/0.047
3RScan	0.375/0.214	0.416/0.161	0.415/0.178
Macro	0.233/0.082	0.307/0.106	0.310/0.103

Table 18: Source-target transfer matrices measured by mAP@0.25. Diagonal entries are in-domain rows.

	SGCDet					ImGeoNet					ImVoxelNet				
Train/Test	ARKit	ScanNet	HyperSim	3RScan	Macro	ARKit	ScanNet	HyperSim	3RScan	Macro	ARKit	ScanNet	HyperSim	3RScan	Macro
ARKit	**0.647**	0.218	0.092	0.375	0.333	**0.629**	0.170	0.059	0.351	0.302	**0.590**	0.067	0.054	0.363	0.269
ScanNet	0.130	**0.531**	0.067	0.059	0.197	0.021	**0.433**	0.011	0.003	0.117	0.023	**0.435**	0.002	0.002	0.115
HyperSim	0.043	0.106	**0.115**	0.010	0.068	0.063	0.081	**0.065**	0.027	0.059	0.056	0.057	**0.069**	0.016	0.049
3RScan	0.246	0.052	0.008	**0.462**	0.192	0.172	0.002	0.016	**0.482**	0.168	0.168	0.005	0.015	**0.418**	0.152

Table 19: SGCDet LODO box-parameter oracle (mAP@0.25 / mAP@0.50).

Oracle edit	ARKit	ScanNet	HyperSim	3RScan	Macro
None	0.369/0.059	0.318/0.127	0.141/0.047	0.415/0.178	0.310/0.103
GT center (xyz)	0.407/0.189	0.343/0.209	0.151/0.099	0.439/0.251	0.335/0.187
GT size	0.308/0.070	0.298/0.168	0.092/0.045	0.412/0.287	0.278/0.143

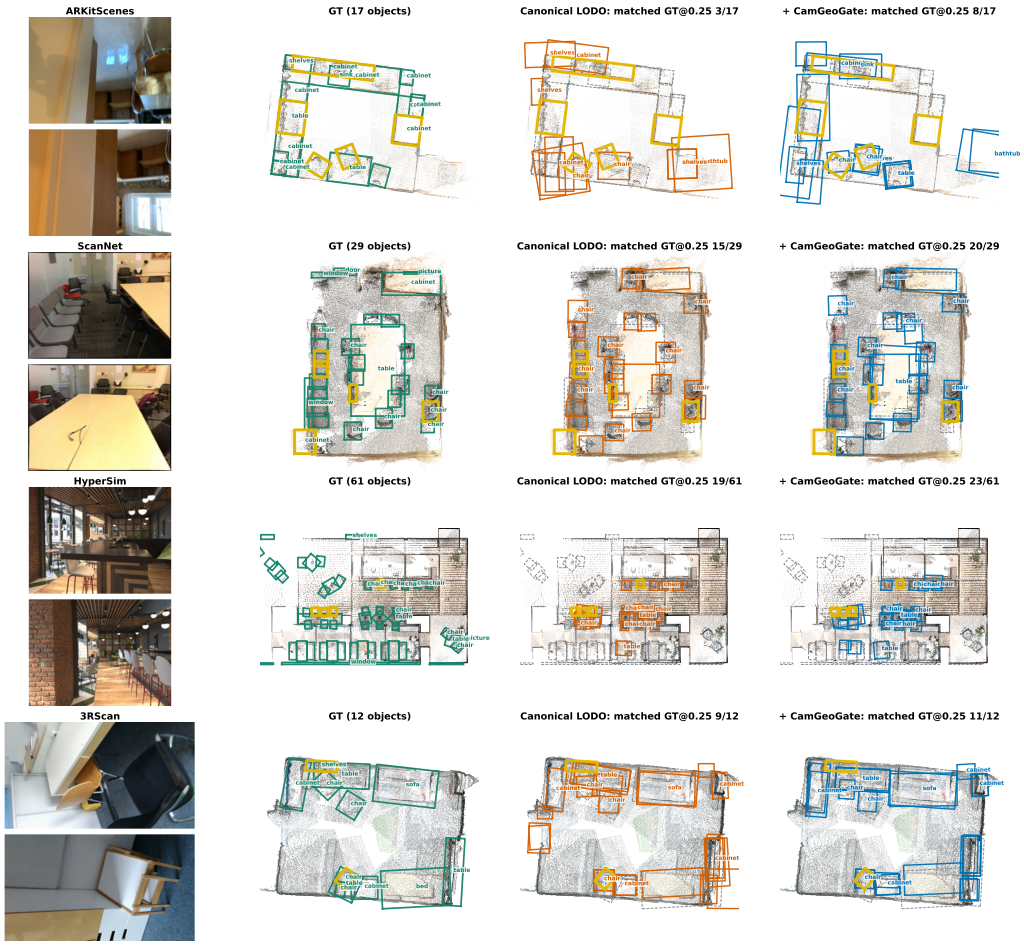


Figure 4: Selected qualitative cases. Canonical→CamGeoGate matched-GT counts at IoU 0.25 are 3→8/17 (ARKitScenes), 15→20/29 (ScanNet), 19→23/61 (HyperSim), and 9→11/12 (3RScan).