Label-Efficient 3D Pseudo-Label Refinement with Segmentation Foundation Model Priors

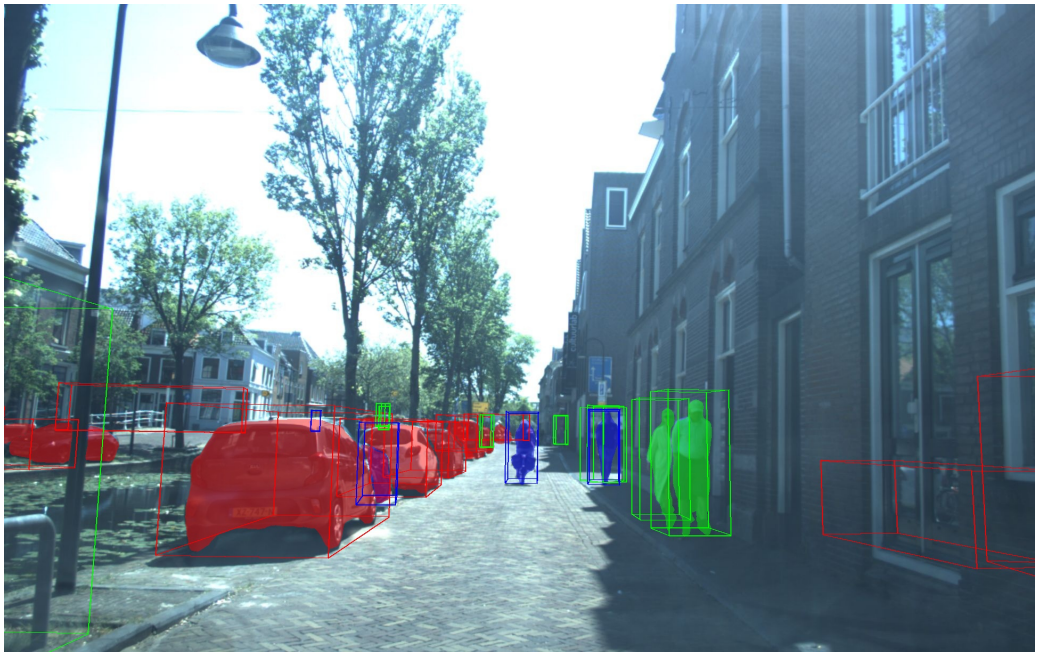
MSc Thesis

Delft University of Technology

Paul Mignot

2 April 2026

Track: MSc Robotics



Supervisors:

Dr. Holger Caesar, TU Delft
Shiming Wang, TU Delft

TU Delft

Label-Efficient 3D Pseudo-Label Refinement with Segmentation Foundation Model Priors

Paul Mignot
TU Delft

Shiming Wang
TU Delft

Holger Caesar
TU Delft

Abstract

High-quality 3D annotations for LiDAR point clouds are expensive to obtain, while domain shift between source and target domains limits the direct use of pretrained 3D detectors. Auto-labeling offers a scalable alternative, but the quality of current pipelines remains largely bounded by LiDAR-based pseudo-label generation. A key observation is that LiDAR auto-labeling pipelines can achieve high recall but produce many false positives, whereas image segmentation foundation models offer high-precision 2D detections with strong zero-shot generalisation, suggesting that the two modalities have complementary error profiles. Building on this complementarity, we propose three modules that inject image segmentation priors into a LiDAR auto-labeling pipeline: rule-based confidence refinement, a weakly supervised learned refinement module, and mask-based 3D proposal generation. On the View of Delft dataset, the learned module achieves the best pseudo-label quality among all configurations. Notably, geometric and confidence cues alone account for the majority of this improvement, with image features providing a consistent but modest additional gain, which indicates that the learned decision boundary, not the raw image signal, is the key enabler.

1. Introduction

Autonomous vehicles must reliably detect other road users to make safe decisions in diverse and unpredictable traffic environments. This requires accurate 3D perception from LiDAR point clouds, yet high-quality 3D annotations remain expensive and slow to obtain at scale. Labeling LiDAR point clouds with tight 3D bounding boxes requires expert annotators, careful handling of sparsity and occlusions, and labeling across long sequences [1, 4, 12, 21]. A further obstacle is *domain shift*: a 3D detector trained on one dataset (source domain) can suffer a dramatic performance drop when deployed in a new environment (target domain), because differences in sensor type, sensor placement and road infrastructure alter the point-cloud distribution [28]. Collecting new annotations for every target domain is extremely expensive, motivating auto-labeling pipelines that generate *pseudo-labels* from raw sensor data [20, 31]. However, the utility of such pipelines is limited by pseudo-label quality, since false positives and missed detections directly hurt downstream training.

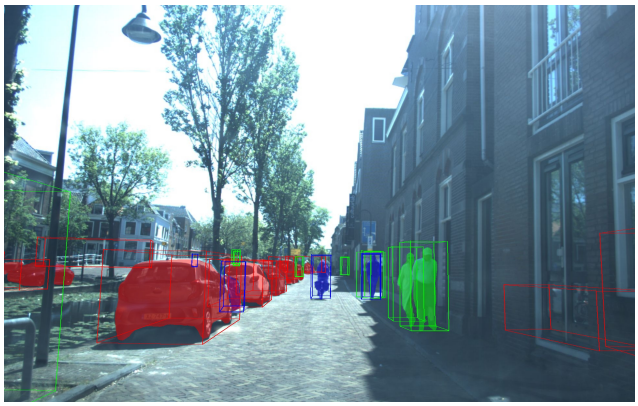


Figure 1. A View of Delft frame overlaid with high-recall LiDAR 3D pseudo-labels and high-precision image segmentation masks, coloured per class: Vehicle, Pedestrian, Cyclist. Cross-modal agreement between the two sources provides evidence for validating, rejecting, or supplementing pseudo-labels.

An effective paradigm for LiDAR auto-labeling is to ensemble multiple source-domain detectors and apply temporal refinement, achieving high recall by fusing diverse predictions and tracking objects over time [25]. However, these pipelines remain purely LiDAR-driven and produce a substantial number of false positives, particularly in sparse or occluded regions. Meanwhile, modern vision foundation models for segmentation offer a complementary strength: high-precision 2D instance masks with strong zero-shot generalisation [2, 8]. As Figure 1 shows, LiDAR auto-labeling pipelines tend to achieve high recall but moderate precision, while image segmentation models exhibit the opposite profile: higher precision but lower recall. Further-

more, these segmentation models are limited to 2D, while LiDAR-based models give us 3D data. This complementarity motivates our central hypothesis: *combining a high-recall LiDAR detector with a high-precision image segmentor should yield pseudo-labels that are more accurate than either modality alone.*

Realising this hypothesis raises three practical challenges: **Robust fusion under asymmetric errors**, since the LiDAR detector over-detects (many false positives) while the image segmentor under-detects (missed objects), and rigid rule-based thresholds make it difficult to distinguish true detections that lack image support from genuine false positives. **Operating with no or few labels**, since fully unsupervised merging limits what can be learned, while full 3D annotation is prohibitively expensive; ideally only cheap binary keep/discard decisions on existing pseudo-labels should suffice. **Bridging the 2D–3D gap**, because lifting 2D masks into 3D bounding boxes requires geometric reasoning and is inherently imprecise when LiDAR only observes the near surface of objects.

We address these challenges by extending a strong LiDAR auto-labeling pipeline with segmentation-model priors and evaluating three ways of incorporating this evidence. Our central finding is that the most effective strategy is not direct multi-modal proposal generation, but lightweight learned pseudo-label refinement: a small MLP trained on cheap binary keep/discard labels substantially improves pseudo-label quality, while segmentation derived features provide a smaller but consistent additional gain. This shows that segmentation priors are most useful as auxiliary evidence inside a learned refinement stage, rather than as a stand-alone source of 3D proposals. Our contributions are the following:

- **Label-efficient learned pseudo-label refinement.** We propose a lightweight MLP that reweights LiDAR pseudo-label confidence using cheap binary keep/discard annotations. This module delivers the largest improvement in pseudo-label quality and substantially outperforms training a 3D detector from scratch under the same annotation budget.
- **Segmentation priors as auxiliary cross-modal evidence.** We study how SAM3-derived agreement and mask features can support pseudo-label refinement. We show that these features provide a consistent but modest gain on top of strong geometric and detector-confidence cues.
- **Analysis of integration strategies for segmentation priors.** We evaluate rule-based refinement, learned refinement, and mask-based 3D proposal generation inside MS3D++, and show that lightweight learned fusion is effective while direct 2D-to-3D lifting remains limited by box-fitting accuracy.

2. Related Work

Unsupervised domain adaptation. Unsupervised domain adaptation (UDA) aims to transfer 3D detectors from labelled source domains to unlabelled target domains, often relying on pseudo-labels as intermediate supervision [28, 31]. Self-training and consistency-based approaches mitigate domain shift by iteratively filtering or re-weighting pseudo-labels to stabilise training under noisy supervision [6, 9, 22, 36]. Representative 3D pipelines include ST3D [31], SE-SSD [35], SSDA3D [29], and CL3D [18]. MS3D and MS3D++ further improve pseudo-label quality by ensembling multiple pretrained detectors and fusing their outputs before temporal refinement [24, 25]. These works underscore that pseudo-label selection and refinement are critical, as errors propagate when pseudo-labels serve as training supervision [27]. However, most existing refinement strategies rely on hand-crafted score thresholds or self-training heuristics; our MLP-based approach instead learns a confidence re-weighting function from a small set of binary annotations, combining geometric and cross-modal features.

Multi-modal cues and foundation-model priors. Camera data provides dense semantic cues that complement LiDAR geometry, motivating multi-modal 3D detection and annotation pipelines. Frustum-based methods project 2D detections into 3D and aggregate the enclosed LiDAR points to produce 3D box proposals [15, 19]; our SAM3-driven proposal module follows a similar frustum strategy but replaces a trained 2D detector with foundation-model masks. Sensor-fusion approaches such as PointPainting [26] and BEVFusion [10] jointly process image and LiDAR features for 3D detection, while recent offline multi-modal auto-labeling systems combine both modalities to produce more reliable pseudo-labels [11, 33]. In parallel, vision foundation models have emerged as strong zero-shot priors for diverse visual tasks. Segment Anything (SAM) [8] introduced class-agnostic mask proposals with strong cross-domain generalisation, and its successor SAM3 extends this to open-vocabulary, concept-level segmentation that outputs per-mask class labels and confidence scores [2]. Because these models generalise across domains without fine-tuning, they are particularly attractive for cross-domain auto-labeling where target-domain training data is scarce. Foundation-model priors have also been explored in broader open-world labeling settings for autonomous driving [23]. In this work, we leverage SAM3's class-aware masks as cross-modal evidence to validate, suppress, or extend LiDAR-based pseudo-labels.

3D auto-labeling via temporal aggregation. A common strategy to improve pseudo-label quality in autonomous driving is to exploit the temporal redundancy in sequential point-cloud sweeps. Offline and 4D auto-labeling pipelines associate per-frame detections across time using

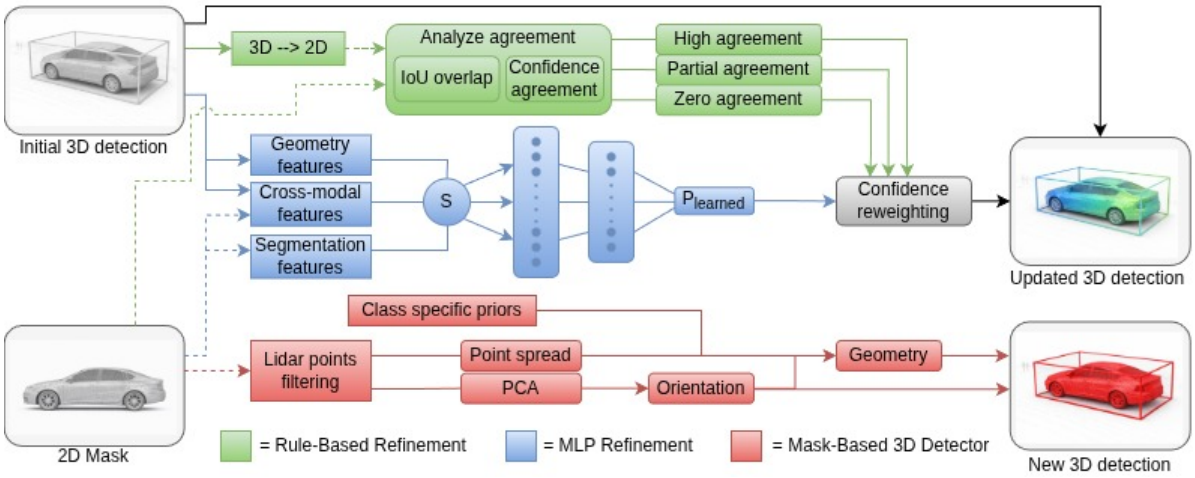


Figure 2. Detailed flowcharts of the three proposed modules. Modules are colour-coded as follows: green for rule-based refinement, blue for learned MLP refinement, and red for the mask-based 3D detector. Dashed arrows denote 2D data and solid arrows denote 3D data. S denotes a standard scaler applied to normalise features to zero mean and unit variance before MLP inference.

multi-object tracking, then aggregate the resulting tracklets to reduce jitter, recover missed detections, and improve box localisation under LiDAR sparsity and occlusions [3, 13, 20, 30]. Tracking-based temporal refinement is particularly effective because a detection that is weak or missing in a single frame can be recovered from neighbouring frames where the same object produces a stronger signal [14, 17]. Recent work further studies robustness and scalability of such 4D auto-labeling across datasets and domains [32]. Our baseline, MS3D++, relies heavily on tracking and temporal refinement to produce high-quality pseudo-labels [25]; the extensions we propose in this work are complementary, operating either on the temporally refined outputs or injecting proposals before tracking, rather than modifying the temporal pipeline itself.

While prior work has combined temporal refinement with cross-domain LiDAR pseudo-labeling, and separate lines of work have shown the value of multi-modal auto-labeling and foundation-model priors, it remains unclear how camera-based segmentation-model evidence should be integrated into a strong domain-adaptation pipeline for 3D pseudo-label generation.

3. Method

As established in the introduction, a high-recall LiDAR auto-labeling pipeline and a high-precision image segmentation model exhibit complementary error profiles: the former over-detects, producing many false positives, while the latter provides reliable per-instance evidence but operates only in 2D and can miss objects. Our goal is to improve 3D pseudo-label quality in a strong LiDAR auto-labeling pipeline under limited target-domain annotation. We investigate whether segmentation-model priors can help achieve this, either by directly refining pseudo-labels or by generating additional proposals. We instantiate this idea by extending MS3D++ [25] with image priors from SAM3 [2].

3.1. Problem Setup

For each frame t, we assume synchronized LiDAR and camera observations. Let $\mathbf{X}_t$ denote LiDAR pointclouds and $\mathbf{I}_t$ the synchronized camera images. A LiDAR auto-labeling pipeline produces 3D pseudo-labels

$$\mathcal{B}_t = \{(b_i, c_i, s_i)\}_{i=1}^{N_t}, \tag{1}$$

where b_i is a 3D bounding box, c_i a semantic class label, and $s_i \in [0, 1]$ a confidence score.

In parallel, a 2D segmentation model generates mask proposals in the image domain:

$$\mathcal{M}_t = \{(m_j, \hat{c}_j, q_j)\}_{j=1}^{M_t}, \tag{2}$$

where m_j is a 2D instance mask, $\hat{c}_j$ is the predicted class label, and $q_j \in [0, 1]$ is its confidence. In our instantiation, pseudo-labels come from MS3D++ [25] and masks from SAM3 [2]. We use $\mathcal{M}_t$ to introduce image evidence that can support, suppress, or add 3D pseudo-labels.

3.2. Cross-Modal Pseudo-Label Refinement

We propose three complementary modules to integrate image segmentation evidence into a LiDAR auto-labeling pipeline, each addressing a different aspect of the problem.

The first two modules address the primary weakness of a high-recall LiDAR detector: its tendency to produce false positives. We begin with a simple *rule-based* approach that uses fixed thresholds on cross-modal agreement to penalise

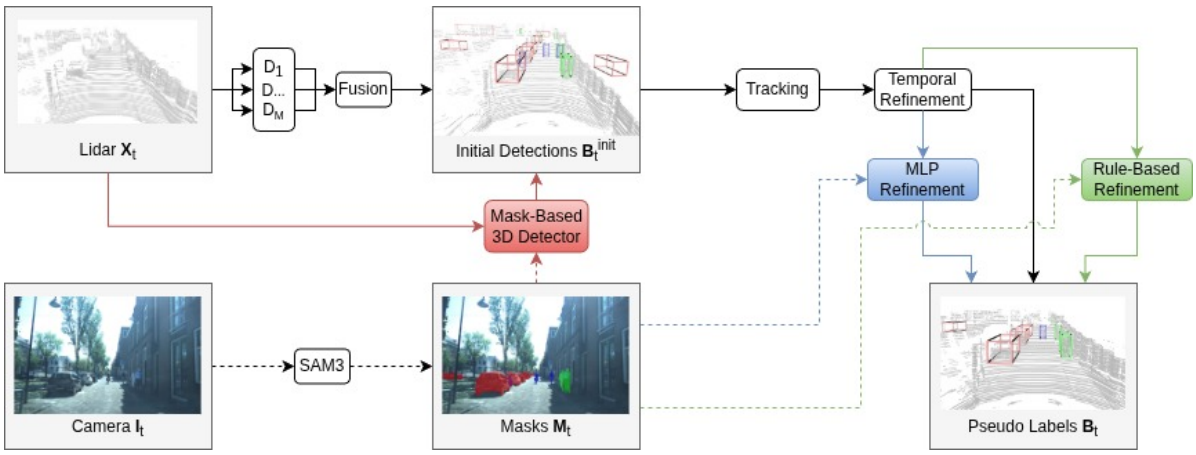


Figure 3. Overview of the MS3D++ baseline and our SAM3-based extensions. The three modules are colour-coded as before: green for rule-based refinement, blue for learned MLP refinement, and red for the mask-based 3D detector. The **black** line shows the MS3D++ baseline. Dashed arrows are 2D images/masks and full arrows are 3D pointclouds/pseudo boxes.

unsupported pseudo-labels. Since rigid rules make it difficult to adapt to the diversity of failure cases, we next propose a *learned refinement* stage that trains a lightweight MLP on a small set of binary keep/discard labels to distinguish these cases. Finally, having addressed precision, we observe that some objects are missed entirely by the LiDAR pipeline. Our third module, a *mask-based 3D detector*, lifts 2D mask proposals into 3D to recover these missed detections and improve recall. A detailed overview of how these methods work is shown in Figure 2. Figure 3 shows how these methods are integrated in the MS3D++ pipeline [25]

3.2.1 Rule-based Refinement

The simplest way to exploit 2D segmentation evidence is to check whether each 3D pseudo-label is supported by a corresponding 2D mask: pseudo-labels without support are likely false positives. We implement this as a post-processing step on the LiDAR pseudo-labels.

For a pseudo box b, we project it into the image and obtain a 2D rectangle $r(b)$. For each segmentation mask m, we compute its 2D rectangle $r(m)$ and the geometric overlap

$$\mathrm{IoU}_{2D}(b, m) = \frac{|r(b) \cap r(m)|}{|r(b) \cup r(m)|}. \tag{3}$$

We combine this geometric overlap with the confidence scores from both sources into a weighted agreement score:

$$a(b, m) = \mathrm{IoU}_{2D}(b, m) \cdot q^{\alpha} \cdot s^{\beta}, \tag{4}$$

where α and β control how strongly the segmentation confidence q and the pseudo-label confidence s influence the agreement decision.

Each pseudo box is matched to the best-overlapping mask with the same predicted class. Based on the weighted agreement score a, we classify each pseudo box into one of three levels: *high agreement* ($a \geq \tau_{\text{high}}$ with matching class), *partial agreement* (overlap exists but $a < \tau_{\text{high}}$ or class mismatch), and *no overlap* (no mask exceeds a minimum score τ_{min}). The pseudo-label confidence is then adjusted adaptively: high agreement increases s_i, partial agreement leaves it unchanged, and no overlap applies a large penalty. Boxes whose updated confidence falls below a minimum threshold are discarded. This module requires no target-domain labels and provides a quick precision improvement.

3.2.2 Learned MLP Refinement

The rule-based approach demonstrates that cross-modal agreement contains useful signal for identifying false positives, but fixed thresholds are too coarse: the causes of missing segmentation overlap are diverse (segmentor failures on small or distant objects, occlusion, class confusion) and cannot be captured by hand-crafted rules. This motivates a learned approach that can weigh these factors jointly.

We train a lightweight MLP to predict the probability that a pseudo-label is a false positive, combining geometric features of the 3D box with cross-modal agreement features and segmentation features from the segmentation model. Critically, the training labels needed are cheap: because the high-recall LiDAR pipeline already produces candidate boxes for most objects in the scene, a human annotator only needs to verify each existing pseudo box as a true or false positive. This is a binary keep/discard decision on boxes that already exist, which is significantly cheaper than creating full 3D bounding-box annotations from scratch. Therefore the MLP learned refinement is a *weakly supervised* setting.

For each pseudo box, we compute a 19-dimensional feature vector $\phi(b)$ comprising three groups (see Appendix A for full definitions):

- **Geometry features (13):** box centre (x, y, z), dimensions (dx, dy, dz), orientation $(\sin\theta, \cos\theta)$, distance to ego, volume, aspect ratio dx/dy, the MS3D++ detector confidence score, and predicted class label.
- **Cross-modal features (3):** 2D IoU between the projected 3D box and the best-matching SAM3 mask, a binary flag indicating whether the SAM3 and MS3D++ class labels agree, and the number of SAM3 masks overlapping the projected box.
- **Segmentation features (3):** the SAM3 mask confidence score, SAM3-predicted class label, and SAM3 mask bounding-box area.

The MLP has two hidden layers (64–32 units, ReLU activations) and is trained with Adam using L_2 regularisation (α=10^{-3}) and early stopping. Features are standardised to zero mean and unit variance. At inference, the MLP outputs $P(\text{FP} \mid \phi(b))$, the estimated probability that a pseudo-label is a false positive. We apply it in a conservative soft mode by downweighting the original confidence: $s_i' = s_i \cdot (1 - P(\text{FP} \mid \phi(b)))$. This soft re-weighting preserves all pseudo-labels (maintaining recall) while pushing likely false positives to low confidence, letting the downstream training pipeline filter them via its own score threshold.

3.2.3 Mask-Based 3D Detector

The previous two modules refine existing pseudo-labels but cannot recover objects that the LiDAR pipeline missed entirely. To improve recall, we generate additional 3D proposals by lifting 2D segmentation masks into 3D using the available LiDAR points.

For each segmentation mask, we project LiDAR points into the image and retain those falling inside the mask. We isolate the foreground object via depth-gap clustering, then fit an oriented 3D bounding box to the remaining points. Because LiDAR only observes the near surface, the point spread alone underestimates dimensions. We address this by blending the observed point spread with class-specific size priors. The orientation is computed using PCA on the filtered points, determining which side of the object is visible. The box centre is offset away from the sensor using the class size priors and the estimated orientation. Each proposal inherits the segmentation mask class $\hat{c}_j$ and derives its confidence from q_j. To avoid redundancy, we reject proposals that overlap existing LiDAR pseudo-labels in BEV, and inject the remaining proposals into $\mathcal{B}_t^{\text{init}}$ before tracking and temporal refinement.

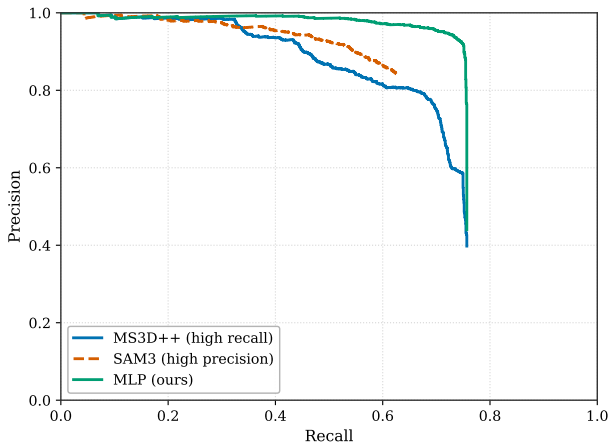


Figure 4. Precision–recall curves for the vehicle class. MS3D++ achieves high recall but moderate precision, while SAM3 achieves higher precision. Combining both modalities through learned refinement yields the best performance.

4. Experiments

We evaluate our three cross-modal extensions on the View of Delft dataset [16] and compare them against the MS3D++ baseline. We first describe the experimental setup (Section 4.1), then present the main results (Section 4.2) and ablation studies on training set size and feature importance (Section 4.3).

4.1. Experimental Setup

Dataset and split. We evaluate on View of Delft (VoD) [16], recorded in Delft, the Netherlands, with a 64-beam LiDAR and a front-facing camera. VoD provides 3D bounding-box annotations for 13 road-user classes within the front-camera field of view (FOV), including *Car*, *Truck*, *Pedestrian*, and *Cyclist* among others [16]. In this work, we map these annotations to three evaluation classes: *Vehicle* (grouping *Car* and *Truck*), *Pedestrian*, and *Cyclist*. We use 8 sequences totaling 2,400 LiDAR frames, split into 1,700 frames for development (MLP training and ablation) and 700 held-out test frames for evaluation. All metrics are computed on the same 700 test frames.

MS3D++ baseline instantiation. We use MS3D++ [25] as our baseline in this research. Concretely, we employ the MS3D++ ensemble with 20 detectors from four architectures (VoxelRCNN and PV-RCNN++, each with an anchor head and a center head) pretrained on nuScenes [1] and Lyft [7]. This detection set is the baseline and kept the same for each method. The MS3D++ temporal and tracking refinement stages are applied only to Vehicle and Pedestrian; the Cyclist class is excluded from these refinement steps due to inconsistent class definitions across the source-domain detectors.

Table 1. Main results on View of Delft (700 test frames). AP is computed using BEV IoU with class-specific thresholds (Veh. = 0.50, Pedestrian/Cyclist = 0.25). P@R.6 denotes precision at recall = 0.6, measuring how many predictions are correct when 60% of ground-truth objects are recovered. Best is in **bold**. M2 and M2 + M3 use 1 700 labelled training frames; MS3D++, M1, and M3 use no target-domain labels.

Method	mAP↑	AP↑			P@R.6↑			Labels
		Veh.	Ped.	Cyc.	Veh.	Ped.	Cyc.	
nuSc. VoxelRCNN	30.5	46.3	36.5	8.7	–	–	–	0
MS3D++ baseline	59.4	66.4	62.8	49.1	0.816	0.729	0.396	0
M1: Rule-based	61.6	68.6	66.2	49.9	0.888	0.825	0.425	0
M2: MLP refinement	**66.3**	**71.7**	**69.1**	**58.2**	**0.972**	**0.909**	**0.707**	1700
M3: Mask-based 3D det.	59.3	66.0	62.9	49.0	0.804	0.729	0.379	0
M2 + M3	**66.4**	**71.4**	**69.0**	**58.8**	**0.968**	**0.904**	**0.724**	1700

SAM3 masks. We compute 2D instance masks with SAM3 [2] using text prompts for a fixed set of categories (`car`, `truck`, `bus`, `bike and cyclist`, `pedestrian`). We retain masks above a fixed confidence threshold and reuse the same masks for all variants.

Calibration and projection. All cross-modal operations use calibrated LiDAR–camera geometry to project 3D boxes and LiDAR points into the image plane [5]. This is used both for matching MS3D++ boxes to SAM3 masks and for selecting mask-filtered LiDAR points for proposal generation.

Front-FOV protocol. Since VoD annotations are limited to the labeled front-camera FOV, we apply the same restriction in evaluation: predictions and ground truth are filtered to the front-FOV before matching, so detections outside the labeled region do not contribute false positives or false negatives.

Evaluation metrics. We report per-class Average Precision (AP) using 11-point interpolation and BEV IoU matching. We use class-specific IoU thresholds: 0.50 for *Vehicle* and 0.25 for *Pedestrian* and *Cyclist*. To quantify precision improvements at a comparable recall operating point, we additionally report Precision at Recall = 0.6 (P@R.6): the precision when 60% of ground-truth objects are recovered.

4.2. Main Results

Table 1 summarises pseudo-label quality on the test set. The main result of Table 1 is that learned pseudo-label refinement (M2) accounts for nearly all of the overall improvement. We compare: (i) a single source-domain detector, (ii) the full MS3D++ baseline, and (iii) our three cross-modal extensions (M1–M3) plus the combined pipeline (M2 + M3).

Recall across methods. Before discussing individual methods, we note that maximum recall is similar across all variants: approximately 0.75 for Vehicle, 0.77–0.79 for Pedestrian, and 0.69 for Cyclist. The LiDAR ensemble already detects the large majority of objects; the main bottleneck is *precision*, which is separating true positives from the many false positives produced by the high-recall pipeline. We therefore focus the discussion on precision improvements, quantified by P@R.6.

Source detector & MS3D++. A single nuScenes-pretrained VoxelRCNN with centerhead achieves only 30.5% mAP when applied to VoD without any adaptation, illustrating the magnitude of the domain shift explained in Section 1. This detector does not reach 0.6 recall for any class, so P@R.6 is undefined. MS3D++ raises mAP to 59.4%, nearly doubling it. This large leap establishes MS3D++ as a strong baseline and sets the ceiling that pure LiDAR-based autolabeling can already reach significant performance without camera evidence. Note that MS3D++'s tracking and temporal refinement stages are applied only to Vehicle and Pedestrian; Cyclist pseudo-labels rely solely on the initial ensemble detections. This is reflected in the Cyclist mAP (49.1%)

M1: Rule-based refinement. Rule-based filtering yields +2.2 mAP over the baseline (61.6%). Boxes without sufficient SAM3 overlap receive a score penalty and are removed if the resulting score falls below a minimum threshold. This directly improves precision: Vehicle P@R.6 rises from 0.816 to 0.888 and Pedestrian P@R.6 from 0.729 to 0.825. However, hard thresholding also discards some true positives whose SAM3 masks were absent or below the confidence cutoff, which slightly reduces recall. The M1 thresholds were selected via an ablation study over 176 configurations on the development set (Appendix B).

M2: MLP refinement. The learned MLP achieves 66.3% mAP (+6.9 over baseline), trained on all 1 700 development frames. The precision improvement is substantial: Vehicle P@R.6 reaches 0.972 (+0.156 over baseline), Pedestrian 0.909 (+0.180), and Cyclist 0.707 (+0.311). Because M2 applies soft score re-weighting rather than hard box removal, it preserves recall while pushing false positives to

low confidence. The PR curves (Figure 4) confirm that M2 maintains precision above 0.9 across most of the recall range, effectively separating true and false positives in score space. The largest per-class AP gain is for Cyclist (+9.1), consistent with the large Cyclist P@R.6 improvement. This indicates that despite the low mAP for the cyclist class at the baseline (49.1%), MLP refinement is able to dramaticaly improve mAP with the use of learned decisions (even for classes that were not refined during the temporal and tracking refinement stages of MS3D++)

M3: Mask-based 3D detector. The mask-based 3D detector adds new detections from SAM3 masks but does not improve recall, and slightly decreases mAP relative to the baseline (-0.1 pp, 59.3%). Although M3 is designed to recover objects missed by LiDAR, the 3D bounding boxes fitted from mask-filtered LiDAR points are not precise enough to match ground truth at the required IoU thresholds, so the added detections do not translate into additional true positives. Moreover, precision also degrades slightly, as the imprecise proposals introduce additional false positives. Fitting tight 3D boxes from partial LiDAR observations without a learned 3D detector is inherently difficult, and an oracle study replacing the 3D fitting with GT box lookup confirms that 3D lifting is the bottleneck (Appendix C).

M2 + M3: Combined pipeline. Combining learned refinement (M2) with the mask-based 3D detector (M3) yields the highest overall mAP at 66.4% (+7.0 over baseline). M3 first injects additional proposals, and M2 then re-weights the enriched pseudo-label set, suppressing the imprecise M3 proposals while retaining the few that are correct. This is the best method however, the +0.1 mAP gain over M2 alone is marginal, indicating that M2 accounts for nearly all of the improvement. We therefore consider M2 the primary result in the remainder of the paper.

4.3. Ablation Studies

We conduct ablation studies on M2, the core learned component driving the largest gains. Although M1 and M3 have no trainable parameters, they depend on threshold and heuristic choices; we ablate these in Appendix B and Appendix C, respectively. M2 offers two natural axes for ablation: (i) how many labelled training frames are needed, and (ii) which features drive the MLP's decisions.

4.3.1 Effect of Training Set Size

A key practical question is how much labelled data M2 requires to be effective. We train the MLP on subsets of increasing size $N \in \{25, 50, 100, 200, 400, 800, 1600\}$ frames drawn from the 1 700-frame development set, and evaluate each model on the fixed 700 test frames. To reduce variance from random subset selection, we average over 20 random seeds per data point.

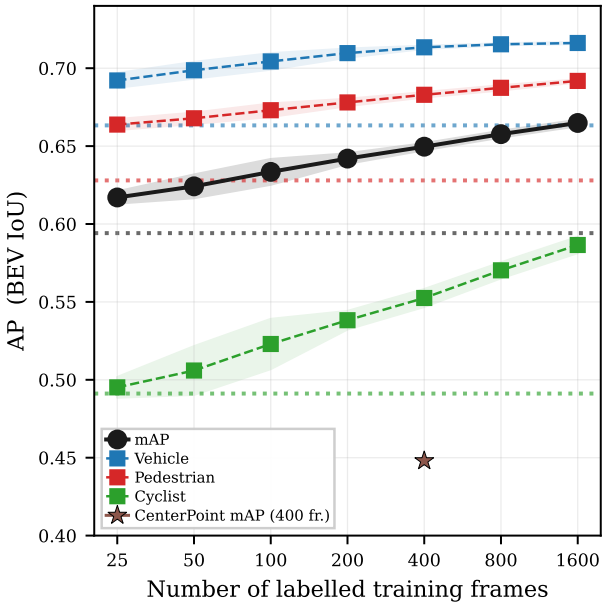


Figure 5. mAP and per-class AP vs. number of labelled training frames for M2 (MLP refinement), averaged over 20 random seeds. Dashed horizontal lines mark the MS3D++ baseline per class. Shaded regions indicate ± 1 standard deviation. The star marker shows CenterPoint trained on 400 fully annotated frames (44.8% mAP). Performance saturates around 400 frames.

Table 2. Comparison of M2 (pseudo-label refinement) vs. CenterPoint (trained from scratch), both using 400 frames. M2 labels require only binary verification; CenterPoint requires full 3D box annotations.

Method	mAP	Veh.	Ped.	Cyc.	Annotation
MS3D++ baseline	59.4	66.4	62.8	49.1	none
CenterPoint (400)	44.8	71.1	43.7	19.7	3D boxes
M2 (400 frames)	**65.0**	**71.3**	**68.3**	**55.3**	binary verify

Figure 5 shows mAP as a function of N. The MLP already surpasses the MS3D++ baseline with as few as 25 labelled frames (61.7% vs. 59.4%), demonstrating that even a small amount of labels is sufficient to improve pseudo-label quality. Performance increases steeply up to approximately 400 frames, where the mAP reaches 65.0%, capturing over 80% of the full-data gain. Beyond 400 frames, returns diminish: the curve plateaus, with the 1 600-frame model (66.5%) only marginally outperforming the 400-frame model (65.0%). This saturation suggests that the MLP quickly learns which feature patterns distinguish true positives from false positives, and that the dominant signal comes from a relatively small number of diverse examples rather than sheer volume.

To contextualise annotation efficiency, we compare M2 trained on 400 frames against a CenterPoint [34] detector

Table 3. Effect of SAM3 features on M2 performance (averaged over 20 seeds). "All features" uses all 19 MLP inputs; "No SAM3" removes the 6 SAM3-derived features and retrains. Δ shows the SAM3 feature contribution.

Feature set	mAP	Veh.	Ped.	Cyc.
MS3D++ baseline (no MLP)	59.4	66.4	62.8	49.1
M2 — no SAM3 features (13 feat.)	65.2	71.2	67.7	56.6
M2 — all features (19 feat.)	**66.3**	**71.7**	**69.1**	**58.2**
Δ (SAM3 contribution)	+1.1	+0.4	+1.5	+1.6

trained from scratch on the same 400 frames with full 3D bounding-box annotations. Table 2 shows that M2 achieves 65.0% mAP while CenterPoint reaches only 44.8%: a gap of +20.2 mAP. The difference is largest for Pedestrian (68.3 vs. 43.7) and Cyclist (55.3 vs. 19.7).

4.3.2 Feature Importance

To understand which information sources drive the MLP's decisions, we measure permutation importance: for each feature, we randomly shuffle its values across the test set and measure the drop in MLP accuracy relative to the unperturbed baseline (92.3% accuracy). Figure 6 shows the ten most important features.

The three most important features are `box_dy` (3D box width, Δacc = 0.170), `aspect_ratio_xy` (0.132), and `ms3d_score` (the baseline pseudo-label confidence, 0.106). This indicates that LiDAR box geometry and the original detector confidence are the dominant cues. Among the SAM3-derived features, `sam3_class_match` (whether the SAM3 and MS3D++ classes agree) ranks 4th (0.082) and `sam3_iou_2d` (geometric overlap with SAM3 masks) ranks 7th (0.040), showing that camera evidence contributes but remains secondary to geometric confidence features.

SAM3 features vs. geometry only. To isolate the contribution of SAM3 features, we compare the full 19-feature MLP against a variant trained on only the 13 MS3D++ features, with all six SAM3 features removed. Results are averaged over 20 random seeds to reduce variance. Table 3 shows the result: removing SAM3 features drops mAP from 66.3% to 65.2% (-1.1 pp). Notably, the geometry-only MLP already achieves 65.2% mAP, which is an improvement of +5.8 mAP over the MS3D++ baseline (59.4%), demonstrating that learned score re-weighting from box dimensions, distance, and detector confidence accounts for the major gain of M2. SAM3 features provide a consistent improvement, but their relative contribution is considerably smaller than the geometric cues. This suggests that much of the information captured by SAM3 agreement may already be implicitly encoded in the geometric features of MS3D++.

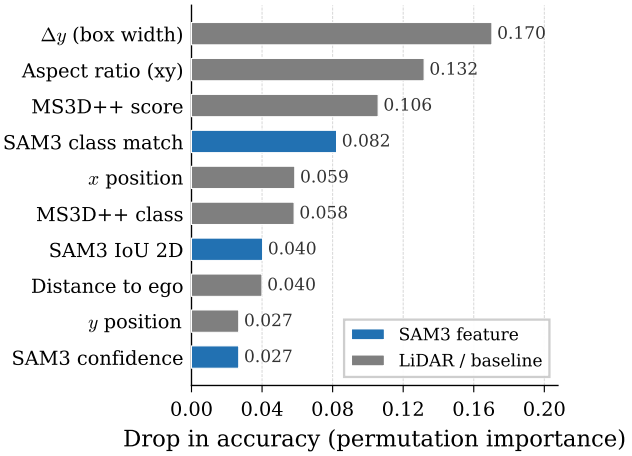


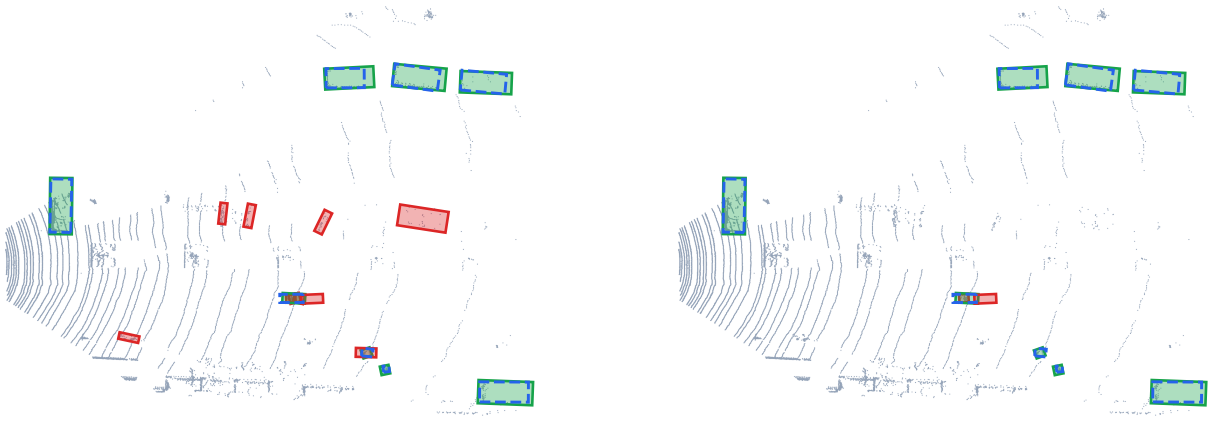
Figure 6. Permutation feature importance (top 10 of 19 features) for the M2 MLP. Bars show the drop in MLP accuracy when each feature's values is randomly permuted across test samples. Blue bars indicate SAM3-derived features; grey bars are LiDAR geometry and baseline features. Three of the top ten features are SAM3-derived.

4.3.3 Qualitative result of MLP Refinement

To illustrate how M2 improves pseudo-label quality, Figure 7 shows BEV visualisations of a test frame. This figure shows that the MLP effectively removes most of the false positives and keeps the true positives. Furthermore, we show in Appendix D that the VOD-trained MLP can be applied without re-training to an independent real-road dataset collected with a different sensor platform, where it produces sensible refinements for close-to-mid-range objects despite the domain shift.

5. Discussion

The main finding of this paper is not simply that combining LiDAR and image evidence improves pseudo-label quality, but that the effectiveness of this combination depends strongly on how the two modalities are integrated. Learned MLP refinement yields the largest improvement over the LiDAR-only baseline across all three classes (Table 1), showing that lightweight learned refinement is the most effective use of cross-modal information in our setting. Rule-based merging improves precision by penalising detections without SAM3 support, but its fixed thresholds cannot adapt to the diverse causes of missing overlap and therefore suppress true positives together with false ones. In contrast, the MLP can jointly weigh box geometry, detector confidence, and cross-modal agreement, which allows it to reject false positives while preserving recall. At the same time, the feature analysis shows that most of the gain comes from geometry and confidence cues, while SAM3-derived features provide a smaller additional benefit. This suggests



(a) MS3D++ baseline

(b) After MLP refinement

Figure 7. BEV visualisation of MLP refinement (M2) on a test frame. Dashed blue outlines = ground truth; green filled = true positives; red filled = false positives. The baseline (a) contains many false positives; after refinement (b) most false positives are removed while all true positives are preserved.

that segmentation priors are most useful as auxiliary evidence within a learned refinement stage rather than as the primary source of improvement. On the completeness side, generating new 3D proposals from 2D masks can recover missed objects, but the benefit is limited by inaccurate 3D box fitting. Finally, M2 is also attractive in practice: with as few as 25 binary-labelled frames it already surpasses the unsupervised baseline, and with a modest labelling budget it clearly outperforms training a detector from scratch on the same number of fully annotated frames (Table 2).

Limitations. Several limitations should be noted. First, our evaluation is conducted on a single dataset (View of Delft) with a single front-facing camera. VoD's relatively high LiDAR density (64-beam) and limited field of view may amplify the benefit of camera evidence compared to settings with wider FOV or sparser LiDAR. Generalisation to other sensor configurations and domains therefore remains to be validated. Second, all methods rely on calibrated LiDAR–camera projection. Calibration errors would affect the 2D–3D matching on which M1, M2, and M3 all depend. While VoD provides reliable calibration, real-world deployments may experience drift, which would degrade cross-modal agreement features. Third, the usefulness of SAM3-derived features appears to be class dependent, with smaller gains for cyclists than for vehicles and pedestrians. M1 and M3 depend directly on mask quality, while M2 is more robust because the geometry-only variant already captures most of the improvement. Finally, M2 requires a small labelled set, breaking the fully unsupervised setting of MS3D++. While the annotation cost is low, because it only requires binary labels on existing pseudo-labels, it remains an additional requirement that M1 and M3 avoid.

6. Conclusion

We studied how segmentation-model priors can be integrated into a strong LiDAR autolabeling pipeline to improve 3D pseudo-label quality. Building on MS3D++, we introduced three complementary extensions: rule-based refinement, learned MLP refinement, and mask-based 3D detection. On the View of Delft dataset, learned MLP refinement is the main contributor, improving mAP by +6.9 points over the baseline while requiring only cheap binary annotations on a small set of existing pseudo-labels, making it substantially more cost-effective than training a 3D detector from scratch. Our analysis shows that geometric and confidence cues account for most of this gain, while segmentation features provide a consistent but comparatively small additional benefit. Future work could evaluate the proposed methods across different sensor configurations, LiDAR densities, and geographic domains to assess generalisability. It would also be valuable to investigate whether the approach transfers to other segmentation models and high-recall LiDAR detectors. Finally, improving the 3D fitting quality of the mask-based detector remains an open challenge; a more accurate lifting stage could increase recall and, when combined with MLP refinement, further improve overall pseudo-label quality. Overall, our results show that lightweight learned refinement can substantially reduce reliance on expensive 3D annotations in new domains, supporting scalable autolabeling for autonomous driving.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. 1, 5

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. 1, 2, 3, 6

[3] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection, 2023. 3

[4] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 6

[6] Deepti Hegde, Vishwanath Sindagi, Velat Kilic, A. Brinton Cooper, Mark Foster, and Vishal Patel. Uncertainty-aware mean teacher for source-free unsupervised domain adaptive 3d object detection, 2021. 2

[7] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset, 2020. 5

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1, 2

[9] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2017. 2

[10] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework, 2022. 2

[11] Chang Liu, Xiaoyan Qian, Binxiao Huang, Xiaojuan Qi, Edmund Lam, Siew-Chong Tan, and Ngai Wong. Multimodal transformer for automatic 3d annotation and object detection, 2022. 2

[12] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C. Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook, 2024. 1

[13] Tao Ma, Xuemeng Yang, Hongbin Zhou, Xin Li, Botian Shi, Junjie Liu, Yuchen Yang, Zhizheng Liu, Liang He, Yu Qiao, Yikang Li, and Hongsheng Li. Detzero: Rethinking offboard 3d object detection with long-term sequential point clouds, 2023. 3

[14] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R. Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving, 2022. 3

[15] Anshul Paigwar, David Sierra Gonzalez, Özgur Erkent, and Christian Laugier. Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar. pages 2926–2933, 10 2021. 2

[16] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Dariu M. Gavrila. Multi-class road user detection with 3+1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022. 5, 13

[17] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking, 2021. 3

[18] Xidong Peng, Xinge Zhu, and Yuexin Ma. Cl3d: Unsupervised domain adaptation for cross-lidar 3d detection, 2022. 2

[19] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data, 2018. 2

[20] Charles R. Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences, 2021. 1, 3

[21] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020. 1

[22] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2018. 2

[23] Levente Tempfli, Esteban Rivera, and Markus Lienkamp. Vespa: Towards un(human)supervised open-world pointcloud labeling for autonomous driving, 2025. 2

[24] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Ms3d: Leveraging multiple detectors for unsupervised domain adaptation in 3d object detection, 2023. 2

[25] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Ms3d++: Ensemble of experts for multi-source unsupervised domain adaptation in 3d object detection. *IEEE Transactions on Intelligent Vehicles*, page 1–16, 2024. 1, 2, 3, 4, 5

[26] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection, 2020. 2

[27] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J. Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection, 2021. 2

[28] Yan Wang, Xiangyu Chen, Yurong You, Li Erran, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize, 2020. 1, 2

[29] Yan Wang, Junbo Yin, Wei Li, Pascal Frossard, Ruigang Yang, and Jianbing Shen. Ssda3d: Semi-supervised domain adaptation for 3d object detection from point cloud, 2022. 2

[30] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. Auto4d: Learning to label 4d objects from sequential point clouds, 2021. 3

[31] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection, 2021. 1, 2

[32] Zhiyuan Yang, Xuekuan Wang, Wei Zhang, Xiao Tan, Jinchen Lu, Jingdong Wang, Errui Ding, Zhihui Lai, and Cairong Zhao. Uni4dal: A unified baseline for multi-dataset 4d auto-labeling. In Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa, Cheng-Lin Liu, Saumik Bhattacharya, and Umapada Pal, editors, *Pattern Recognition*, pages 167–182, Cham, 2025. Springer Nature Switzerland. 3

[33] Zhiyuan Yang, Xuekuan Wang, Wei Zhang, Xiao Tan, Jincheng Lu, Jingdong Wang, Errui Ding, and Cairong Zhao. Fusion4dal: Offline multi-modal 3d object detection for 4d auto-labeling. *International Journal of Computer Vision*, 133:3951–3969, 02 2025. 2

[34] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking, 2021. 7

[35] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud, 2021. 2

[36] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training, 2020. 2

Supplementary Material

This supplement provides additional details that support the main paper. Appendix A lists all 19 MLP input features. Appendix B and C present ablation studies for the rule-based refinement and mask-based 3D detector, respectively. Appendix D shows a qualitative cross-domain evaluation of the MLP refinement module on an independent real-road dataset.

A. MLP Feature Descriptions

Table 4 provides a detailed description of the 19 features used by the MLP refinement module (Section 3.2.2). Features are grouped into three categories: *geometry features* describing 3D bounding-box properties and MS3D++ detector outputs, *cross-modal features* that relate the 3D pseudo-label to the 2D segmentation masks, and *segmentation features* derived purely from the matched SAM3 mask.

B. Ablation Study: Rule-Based Refinement

Rule-Based Refinement depends on six tuneable thresholds grouped into two stages: *agreement* (how pseudo-labels are categorised based on SAM3 overlap) and *refinement* (how confidence scores are adjusted per category). We ablate both stages independently on the 1,700-frame development set and evaluate the final result once on the 700-frame test set.

Agreement parameters (Phase A). We sweep the agreement threshold $\tau_{\text{high}} \in \{0.15, 0.25, 0.35, 0.45\}$, the minimum overlap threshold $\tau_{\text{min}} \in \{0.05, 0.10, 0.15\}$, and the confidence exponents $\alpha \in \{0, 0.25, 0.5, 1.0\}$ and $\beta \in \{0, 0.25, 0.5, 1.0\}$ over 176 valid configurations, keeping refinement parameters at their defaults. Table 5 reports the average mAP for each parameter value, marginalised over all others.

The minimum overlap threshold τ_{min} is the most sensitive parameter (3.4 pp spread), followed by α (2.6 pp). Lower thresholds consistently outperform higher ones, indicating that permissive matching preserves more true positives. The best configuration is then used once on the test set, giving 61.6 mAP (Table 1).

Refinement parameters (Phase B). Using the best Phase A configuration, we independently sweep each of the six refinement parameters (`delta_high`, `delta_partial`, `delta_no_overlap`, `drop_no_overlap_below`, `drop_partial_below`, `min_score_after_adjust`). Performance is remarkably stable, which confirms that the agreement-level thresholds (Phase A) dominate M1 performance, while the refinement score adjustments have negligible impact.

C. Oracle Method: Mask-Based 3D Detector

The main bottleneck of M3 is 3D bounding-box fitting: given only mask-filtered LiDAR points, estimating a tight 3D box is inherently noisy. To quantify this, we construct an *oracle* variant that replaces the geometric 3D fitting with a ground-truth lookup: for each SAM3 2D mask, we project all GT box centres onto the image and select the closest class-matching GT box whose centre falls inside the mask. This isolates the quality of the 2D proposals from the 3D lifting step.

We test two SAM3 operating points: the *default* masks (same as the main experiments) and *high-recall* masks obtained by lowering the SAM3 confidence threshold, which increases the number of detected objects at the cost of more false positives. Table 6 reports the results.

With the default SAM3 masks, geometric lifting yields -0.2 mAP relative to the baseline, while the oracle yields $+1.9$ mAP. The gain is concentrated in Vehicle ($+5.2$ AP), where the oracle provides perfectly sized boxes, confirming that 3D fitting quality is the primary bottleneck. Pedestrian and Cyclist AP are largely unchanged.

With high-recall masks and geometric lifting, mAP remains at 59.3%, which is virtually identical to the default-threshold lifted variant (59.3%) and marginally below the baseline (59.5%). This is a striking result: lowering the SAM3 confidence threshold adds many more 2D proposals and would be expected to add many more false positives; however, the AP remains the same. The explanation for this is that the temporal and tracking refinement of MS3D++ already refines those false positives. Additionally, the low-confidence masks tend to cover objects with fewer LiDAR points or less distinct geometry, making the resulting 3D proposals even noisier than those from the default masks. These imprecise boxes become false positives that exactly offset any recall gain, leaving AP unchanged.

The oracle variant confirms this interpretation. With the same high-recall masks, replacing geometric lifting with GT box lookup raises mAP to 71.4%, with large gains across all classes. This represents an upper bound: every GT object that SAM3 detects in 2D is recovered with a perfect 3D box. The 12.1 pp gap between the high-recall lifted and oracle variants quantifies the headroom lost to imprecise 3D fitting and underscores that improving the 3D lifting step is the key to unlocking M3's potential.

D. Cross-Domain Qualitative Evaluation

To assess whether the MLP refinement module generalises beyond its training domain, we apply the VOD-trained model *without any re-training or fine-tuning* to an independent real-road dataset collected with a different sensor platform.

Table 4. The 19 MLP input features grouped by category. Geometry features describe the 3D pseudo-label box itself and the MS3D++ detector output; cross-modal features relate the 3D box to the 2D segmentation masks; segmentation features are derived purely from the matched SAM3 mask.

Group	#	Feature	Description
Geometry (13)	1	box_x	x-coordinate of the box centre in the LiDAR frame (metres).
	2	box_y	y-coordinate of the box centre in the LiDAR frame (metres).
	3	box_z	z-coordinate of the box centre in the LiDAR frame (metres).
	4	box_dx	Bounding-box length along the heading axis (metres).
	5	box_dy	Bounding-box width perpendicular to the heading axis (metres).
	6	box_dz	Bounding-box height (metres).
	7	box_sin_yaw	Sine of the heading (yaw) angle.
	8	box_cos_yaw	Cosine of the heading (yaw) angle; with $\sin\theta$ avoids wrap-around discontinuities.
	9	ms3d_score	MS3D++ pseudo-label confidence $s \in [0, 1]$ from the multi-teacher ensemble.
	10	ms3d_class	MS3D++ predicted class label (1 = Car, 2 = Pedestrian, 3 = Cyclist).
	11	distance_to_ego	Euclidean distance $\sqrt{x^2 + y^2 + z^2}$ from the ego vehicle (metres).
	12	box_volume	Box volume $dx \cdot dy \cdot dz$ (m^3); correlates with object class.
	13	aspect_ratio_xy	Length-to-width ratio dx/dy; distinguishes elongated vs. compact boxes.
Cross-modal (3)	14	sam3_iou_2d	2D IoU between the projected 3D box and the best-matching SAM3 mask bounding box; 0 if no overlap.
	15	sam3_class_match	Binary flag: 1 if the SAM3 and MS3D++ class labels agree, 0 otherwise.
	16	sam3_num_overlaps	Number of SAM3 masks that overlap the projected 3D box above a minimum 2D IoU threshold.
Segm. (3)	17	sam3_confidence	SAM3 confidence score of the best-matching mask $q \in [0, 1]$.
	18	sam3_class	SAM3-predicted class label of the matched mask (1/2/3 or 0 if no match).
	19	sam3_bbox_area	Pixel area of the matched SAM3 mask's bounding box (px^2); reflects apparent object size.

D.1. Domain Shift

The MLP is trained on View of Delft (VOD) [16], which uses a 64-beam LiDAR, a 1936×1216 camera, and KITTI-format calibration. The target dataset is collected with a Toyota Prius equipped with a 128-beam LiDAR and a 1440×928 front-facing camera at a different location in Delft. The camera intrinsics, extrinsics, and mounting position all differ from VOD, requiring nuScenes-format quaternion calibration instead of the KITTI projection matrices used during training. Despite these differences, the 19-dimensional MLP feature vector (Table 4) is constructed identically: geometry, cross-modal, and segmentation features are sensor-agnostic by design, making the model applicable across platforms without architectural changes.

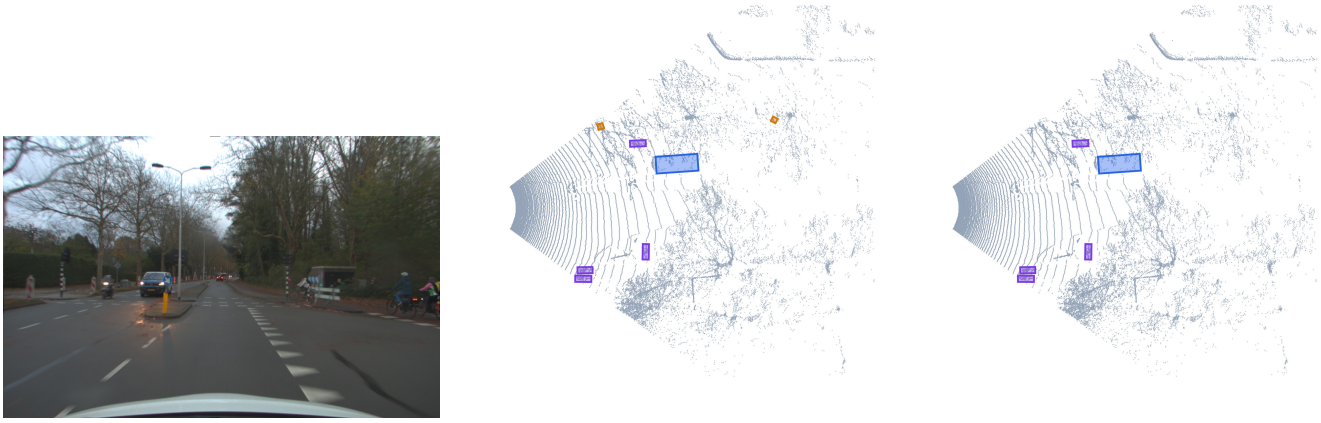
D.2. Qualitative Results

We run the full MS3D++ pipeline on 82 unlabeled real-road frames to generate baseline pseudo-labels, then apply the VOD-trained MLP in `fov_soft` mode (Section 3.2.2) to produce refined labels. Since no ground truth is available, we evaluate by visual inspection of the camera image alongside BEV point-cloud plots. Two representative frames are shown in Figures 8 and 9.

Frame 1 (Figure 8) contains one car, three cyclists, and one motorcycle in the camera field of view. The MLP correctly retains all five objects while suppressing two spurious pedestrian detections, demonstrating that the learned Keep/Discard patterns transfer well for nearby, well-resolved objects.

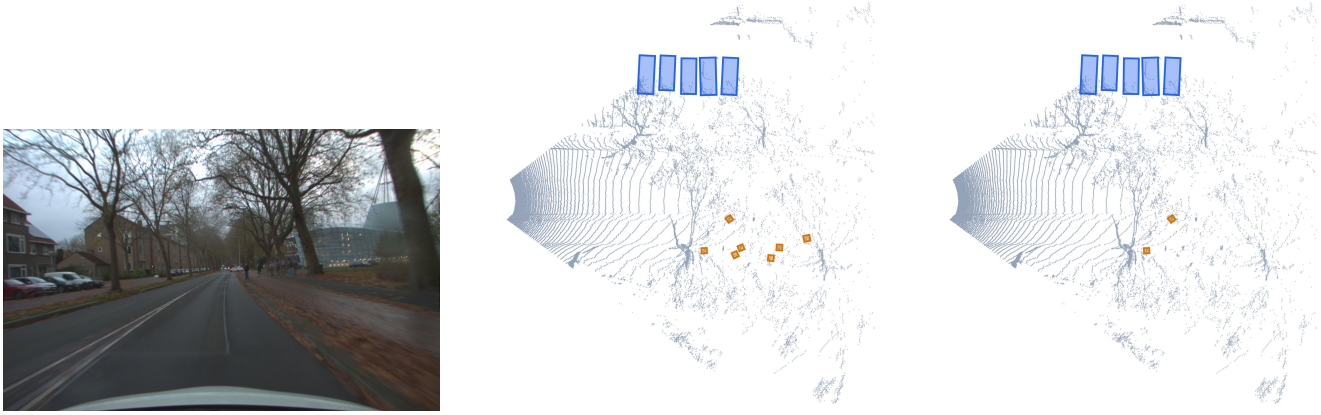
Frame 2 (Figure 9) is a scene with five parked cars and multiple pedestrians. All five parked cars are correctly preserved. However, some pedestrian detections are discarded: of the visible pedestrians, only two are retained. The discarded pedestrians are distant or not clearly visible in the image due to the low quality of these images, where the 128-beam LiDAR still provides sparse point evidence and the SAM3 masks offer weaker cross-modal support.

Discussion. Overall, the MLP produces sensible refinements despite the domain shift: nearby, well-supported objects are retained and isolated false positives are removed. Performance degrades at longer range, where fewer LiDAR points per object and smaller projected image regions reduce the discriminative power of both the geometry and cross-modal features. This is consistent with the feature importance analysis (Section 4.3.2), which shows that distance-correlated features such as `sam3_bbox_area` and `distance_to_ego` are among the most influential. The results suggest that the MLP's learned decision boundary is largely sensor-agnostic for close-to-mid-range objects, but dedicated fine-tuning on in-domain data would likely improve performance at longer ranges.



(a) Camera view (b) MS3D++ baseline (c) After MLP refinement

Figure 8. Cross-domain MLP refinement on the real-road dataset (Frame 1). The MLP correctly retains one car, three cyclists, and one motorcycle while suppressing two false pedestrian detections. Boxes are coloured by class: blue = Vehicle, orange = Pedestrian, violet = Cyclist.



(a) Camera view (b) MS3D++ baseline (c) After MLP refinement

Figure 9. Cross-domain MLP refinement on the real-road dataset (Frame 2). All five cars are correctly retained. Some distant or partially occluded pedestrians are incorrectly discarded, illustrating the range-dependent degradation under domain shift.

Table 5. M1 agreement-parameter sensitivity: average mAP (%) for each value, marginalised over all other parameters. Best value per parameter in **bold**.

Param.	**Values**				**Spread**
τ_{high}	**0.15: 60.5**	0.25: 59.5	0.35: 59.2	0.45: 59.0	1.5
τ_{min}	**0.05: 60.9**	0.10: 59.5	0.15: 57.5	—	3.4
α	**0.0: 60.5**	0.25: 59.9	0.5: 59.5	1.0: 58.0	2.6
β	**0.0: 60.0**	0.25: 59.8	0.5: 59.6	1.0: 58.5	1.5

Table 6. M3 oracle ablation. "Lifted" uses geometric 3D fitting; "Oracle" replaces it with GT box lookup. "High-recall" uses a lower SAM3 confidence threshold.

SAM3	**3D src.**	**mAP**	**Veh.**	**Ped.**	**Cyc.**	**MaxR**		
						V.	**P.**	**C.**
MS3D++ baseline		59.5	66.6	62.8	49.1	.757	.774	.688
Default	Lifted	59.3	66.0	62.9	49.0	.755	.787	.696
Default	Oracle	61.4	71.8	63.4	49.1	.807	.795	.697
High-rec.	Lifted	59.3	65.9	62.9	49.0	.759	.787	.696
High-rec.	Oracle	**71.4**	**77.1**	**78.4**	**58.8**	.814	.810	.724