

Learning 3D Perception from Others' Predictions

Anonymous ICCV DriveX Workshop submission

Paper ID 1

Abstract

001 *Accurate 3D object detection in real-world environments*
002 *requires a huge amount of annotated data with high quality.*
003 *Acquiring such data is tedious and expensive, and often*
004 *needs repeated effort when a new sensor is adopted or*
005 *when the detector is deployed in a new environment. We*
006 *investigate a new scenario to construct 3D object detec-*
007 *tors: learning from the predictions of a nearby unit that is*
008 *equipped with an accurate detector. For example, when a*
009 *self-driving car enters a new area, it may learn from other*
010 *traffic participants whose detectors have been optimized for*
011 *that area. This setting is label-efficient, sensor-agnostic, and*
012 *communication-efficient: nearby units only need to share*
013 *the predictions with the ego agent (e.g., car). Naively using*
014 *the received predictions as ground-truths to train the detec-*
015 *tor for the ego car, however, leads to inferior performance.*
016 *We systematically study the problem and identify viewpoint*
017 *mismatches and mislocalization (due to synchronization and*
018 *GPS errors) as the main causes, which unavoidably result*
019 *in false positives, false negatives, and inaccurate pseudo la-*
020 *bels. We propose a distance-based curriculum, first learning*
021 *from closer units with similar viewpoints and subsequently*
022 *improving the quality of other units' predictions via self-*
023 *training. We further demonstrate that an effective pseudo*
024 *label refinement module can be trained with a handful of*
025 *annotated data, largely reducing the data quantity necessary*
026 *to train an object detector. We validate our approach on the*
027 *recently released real-world collaborative driving dataset,*
028 *using reference cars' predictions as pseudo labels for the ego*
029 *car. Extensive experiments including several scenarios (e.g.,*
030 *different sensors, detectors, and domains) demonstrate the*
031 *effectiveness of our approach toward label-efficient learning*
032 *of 3D perception from other units' predictions.*

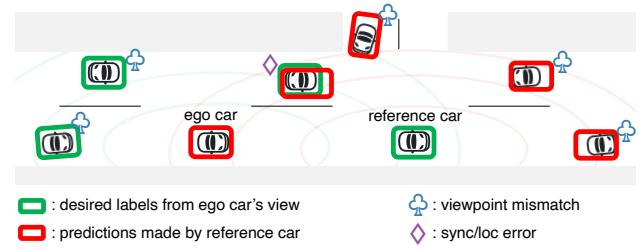


Figure 1. **Research problem of learning from others' predictions.** We study the scenario where an agent (*e.g.*, ego car) leverages the predictions made by another agent (*e.g.*, a high-end reference car) as supervision to train its own 3D object detector. We observe two challenges: (1) viewpoint mismatch between two cars and (2) mislocalization due to synchronization/GPS errors.

039 Acquiring them is laborious and expensive, and is seldom
040 a once-and-for-all effort. Whenever an agent enters a new
041 environment and encounters new objects, its detector needs
042 adaptation to remain accurate. Whenever a new sensor is
043 adopted (*e.g.*, for energy or space efficiency), the different
044 patterns in sensor data (*e.g.*, LiDAR point cloud style and
045 density) necessitate the detector to be retrained. All these
046 updates to the detector imply yet another round of tedious
047 labeled data acquisition.

048 Could we bypass or, at least, reduce the repeated labeling
049 effort? In this paper, we investigate the scenario in which
050 there are other nearby agents equipped with accurate 3D
051 object detectors (but not necessarily with the same sensor
052 configuration). This scenario is realistic and promising. For
053 example, self-driving taxis (*e.g.*, Waymo, Baidu) or local
054 facilities (*e.g.*, surveillance systems, roadside units) are likely
055 to be equipped with optimized detectors for their specific
056 geo-fenced areas. While it may be infeasible for these local
057 "experts" to directly share their raw sensor data or detectors
058 (*e.g.*, due to data size and format; commercial and intellectual
059 properties; implementation incompatibility), the *predictions*
060 (*e.g.*, detected 3D bounding boxes) are more lightweight and
061 standardized. Several recent works also show that sharing
062 predictions would benefit each participating agent's percep-
063 tion accuracy [4, 18, 19, 26, 42, 50, 51, 53, 64], further
064 incentivizing such a collaborative scenario. Last but not least,
065 sharing predictions implies that there is no need for all the

1. Introduction

033 Accurate detection of mobile objects (*e.g.*, vehicles, humans)
034 in 3D is essential for an intelligent agent (*e.g.*, self-driving
035 car, service robot) to operate safely and reliably [20, 37, 38,
036 43]. Constructing such a 3D object detector is never easy
037 — it requires a huge amount of high-quality *labeled* data.
038

066 agents to use the same sensors. An agent adopting a new sensor
067 or entering an unfamiliar environment thus could borrow
068 the predictions made by other agents, potentially equipped
069 with higher-end sensors, as labels to train its detector. (Please
070 see Sec. 3.1 for a feasibility and practicality discussion of
071 our setting.)

072 In this paper, we thus investigate a new scenario to con-
073 struct a 3D object detector: *learning from the predictions*
074 of a nearby agent equipped with an accurate detector. We
075 use the real-world collaborative driving dataset [53] as the
076 testbed. For each 3D road scene, this dataset records two Li-
077 DAR point clouds from two nearby cars distancing between
078 0 ~ 100 meters and offers object labels separately for each
079 point cloud. We use one of the cars as the reference agent,
080 equipped with an accurate 3D detector, to provide predicted
081 labels from which the other (ego) car can learn.

082 At first glance, this research problem may appear trivially
083 as a standard supervised learning problem — using another
084 agent’s predictions as labels to train the detector for the
085 ego car. However, our preliminary attempt showed that this
086 straightforward approach results in poor performance. We
087 identify two major challenges (Fig. 1). First, in real-world
088 applications, inaccuracies such as GPS errors and synchro-
089 nization delays between agents are common. For example,
090 a minor delay of just 0.1 seconds can cause a discrepancy
091 of several meters in localization for a vehicle traveling at 60
092 mph. Second, the viewpoints of the two agents can vary sig-
093 nificantly. An object visible to one agent might be obscured
094 or out of range for the other due to occlusion or distance,
095 leading to false positives and negatives in the predictions.
096 Training with such *mislocalized* and *viewpoint-mismatched*
097 labels inevitably results in suboptimal performance for the
098 new 3D detector of the ego car.

099 To address these challenges, we propose a learning
100 pipeline termed as *Refining & Discovering Boxes for 3D*
101 *Perception from Others’ Predictions (R&B-POP)*. For mis-
102 localization, we train a box refinement module to rank the
103 noisy candidates and correct their locations. Notably, this
104 module requires very few human labels (1% or less), or even
105 no human labels if simulation data are available. We also
106 develop a coarse-to-fine approach to search for high-quality
107 candidates around the predicted object locations efficiently,
108 tackling large localization errors. For viewpoint mismatch
109 that results in false negatives in the ego car’s perspective, we
110 present an effective self-training strategy empowered by a
111 novel distance-based curriculum, enabling the detector to
112 first learn from a subset of high-quality labels and in turn
113 fill in the missing labels for the model to continually learn
114 from. With these approaches, we significantly improve the
115 quality of pseudo labels and, consequently, produce a much
116 more accurate 3D detector for the ego car, with very limited
117 human labeling — the Average Precision (AP) at IoU 0.5
118 increases from 22% to 56.5% using only 40 labeled frames!

In summary, we introduce a novel research problem that
learns 3D perception for a new agent with reference agent’s
predictions. We identify the main challenges about the label
quality and propose corresponding solutions. With extensive
experiments, we demonstrate the applicability of the new
learning scenario as well as the improvements achieved by
our designs.

2. Related Work

3D object detection serves an important role in real-world
applications such as autonomous driving. The detector takes
3D signals (*e.g.*, LiDAR points) as input, and predicts the
existence and the location of objects of interest. Notable
development has been made thanks to the recently curated
large datasets [3, 12, 39, 53, 64]. The existing approaches
can be categorized as voxel-based (or pillar-based) meth-
ods [20, 54, 68], which subdivide irregular 3D point space
into regular space, and point-based methods [59, 67], which
directly extract discriminative point-wise features from the
given point clouds. Regardless of approaches, these methods
require manually annotated data of high-quality to achieve
satisfactory performance. In this study, we aim to bypass
such a labeling cost and demonstrate our new label-efficient
learning method with representative 3D detectors [20, 54].
Label-efficient learning. Self-supervised learning is a
promising way to bypass extensive label annotation [5, 6, 15,
17]. Pre-trained with abundant, easily collectible unlabeled
data, the detector backbone is shown to largely reduce the
labeled data for fine-tuning [31, 49, 60]. Label-free 3D ob-
ject detection from point clouds has gained attention due to
its effective data utilization [1, 8, 27, 29, 36, 55, 61, 62, 66]
and generalization beyond specific class information dur-
ing training [30]. Researchers have also explored semi-
supervised methods [23, 24, 41, 46, 47, 58, 65] or offboard
detectors [28, 34] to reduce the manual labeling efforts. Or-
thogonally, we study a new scenario to learn the detector in a
label-efficient way by considering *beyond a single source of*
information. Specifically, the predictions from well-trained
detectors of reference units near the ego car are leveraged as
(pseudo) labels.

Domain adaptation. Our setting is related to domain adap-
tation (DA), as we aim to improve an object detector in a
new environment (*e.g.*, a new location or data pattern). Ex-
isting studies [7, 44, 55, 56] mostly focus on the generic,
single-agent unsupervised DA setting, while a few leverage
application-specific cues, *e.g.*, repetitions [63], to facilitate
adaptation. Our setting belongs to the second branch, in
which we explore a multi-agent scenario. Our goal is not to
compete with the generic setting. Instead, generic DA tech-
niques, *e.g.*, advanced self-training [16, 32, 40, 45], can be
compatible with our setting to further boost the performance.
Curriculum learning. Many studies have shown that prop-
erly ordering the data to progressively add harder samples

171 during training leads to superior performance. The so-called
 172 “curriculum learning” [2] has also been explored in object de-
 173tection [22, 35]. For LiDAR-based 3D detection, researchers
 174 have applied the concept for better data augmentation during
 175 training [55, 69]. We investigate the task-specific character-
 176 istics from the data and discover a meaningful correlation
 177 between the label quality and the ego-car-reference-car dis-
 178 tance. We then apply this observation to design an effective
 179 training curriculum.

180 **Collaborative perception.** To mitigate limited detection
 181 range and occlusion, self-driving researchers have recently
 182 focused on integrating nearby detectors’ information [4, 18,
 183 19, 26, 42, 50, 51, 53, 64]. During inference, more than
 184 one detector communicates with each other and shares their
 185 information (*e.g.*, input signal, feature, or predicted boxes)
 186 to detect objects better. While also leveraging other cars’
 187 information, our research focus is different — we investigate
 188 a new label-efficient learning scenario, using other (expert)
 189 cars’ predictions as supervision to build the ego car’s detector
 190 offline.

191 3. Learning 3D Perception from Others’ Predic- 192 tions

193 We study a novel research problem in autonomous driving:
 194 training a 3D detector using bounding boxes supplied by a
 195 nearby agent. This scenario, while unexplored, can reduce
 196 or even eliminate labeling efforts. We identify the key chal-
 197 lenges and propose the learning pipeline to address them.

198 3.1. Problem definition and feasibility

199 **Problem setup.** Without loss of generality, we assume that
 200 around the ego car (*i.e.*, E), there is a reference car (*i.e.*, R)
 201 equipped with an accurate 3D object detector f_R . E and
 202 R are both equipped with 3D sensors (*e.g.*, LiDAR) and
 203 collect their point clouds (*i.e.*, X_E and X_R) in the same road
 204 scene. Notice that X_E and X_R can have different patterns
 205 due to variations in hardware. R , the car that E learns from,
 206 share 3D bounding boxes of foreground objects in the global
 207 coordinate from its detector, *i.e.*, $Y_R = f_R(X_R)$. Our goal
 208 is to train a 3D detector f_E that works with X_E , by using
 209 Y_R . Please see Sec. S1 for more detailed problem setup.

210 **Feasibility and practicality.** Before proceeding, we con-
 211 sider two critical questions, “Why can nearby agents obtain
 212 accurate detectors?” and “Why can they not directly share
 213 their detectors?” Besides the examples mentioned in Sec. 1
 214 (*e.g.*, self-driving taxis), we emphasize that these nearby
 215 agents need not be “omniscient.” Instead, they only need to
 216 be experts in geo-fenced areas where the ego agent passes by
 217 and can even be static, making training their detectors easier
 218 and much more label-efficient, *e.g.*, using the repetition or
 219 background cues [9, 61, 62].

220 Regarding “why these agents cannot just share their de-
 221 tectors,” we note that while open-sourcing is common in

Table 1. **Label quality** in recall and precision at IoU 0.5 with E ’s GT. Our methods improve the label quality significantly.

pseudo label	R ’s GT	R ’s pred
	rec. / prec.	rec. / prec.
① initial boxes	54.8 / 43.2	56.1 / 48.0
② + basic filtering	54.1 / 65.6	55.3 / 71.4
③ + our refinement	66.2 / 85.4	65.2 / 79.0
④ + our self-train	72.5 / 90.0	74.4 / 87.7
⑤ sharing detector	-	78.8 / 86.8

the research community, there are many considerations and
 222 constraints when it comes to practical scenarios. First, the
 223 ego and the reference agents do not need to have the same
 224 sensors. Indeed, they may not even perceive the environ-
 225 ment from the same views, *e.g.*, the reference agent can
 226 be a roadside unit placed six meters high and facing down
 227 [9, 57]. This discrepancy makes the direct deployment of
 228 the reference agent’s model to the ego agent suboptimal.
 229 Second, the two agents may be equipped with different com-
 230 putational platforms, *e.g.*, the reference one is equipped with
 231 GPUs while the ego one with FPGA boards and hardware
 232 acceleration code [13, 14], making direct deployment more
 233 challenging. Last but not least, reference agents’ detectors
 234 may be specifically designed and trained, *e.g.*, using private
 235 data. Sharing them thus raises intelligent property or privacy
 236 concerns. Putting things together, we argue that our setting
 237 is realistic and has significant practical implications.

238 3.2. Challenge

239 **First attempt.** We use the recently released real-world col-
 240 laborative driving dataset [53] as the testbed. For each 3D
 241 road scene (with a time tag), the dataset provides LiDAR
 242 point clouds and ground-truth 3D bounding boxes from each
 243 agent’s perspective. (We keep the data and experimental
 244 details in Sec. 4.) We begin by employing Y_R (*i.e.*, R ’s
 245 predictions) directly as labels for E to train f_E , after trans-
 246 forming Y_R into E ’s coordinate system. To establish an
 247 upper bound, we also train a detector using E ’s ground-truth
 248 labels. The result shows that the detector performance by
 249 naively using Y_R is way much worse than the upper bound
 250 (AP at IoU 0.5: ① 22.0 vs. ♦ 58.4 in Table 2).

251 At first glance, such a gap, with no doubt, must come
 252 from reference car R ’s prediction errors. To eliminate the
 253 effect, we use R ’s ground-truths as labels (*i.e.*, R ’s GT) to
 254 train another detector for the ego car E . To our surprise,
 255 using R ’s GT can hardly improve the detector’s performance,
 256 suggesting the existence of other, more fundamental factors
 257 in the real-world environment.

258 **Key challenges.** To search for the root cause of the poor
 259 detector performance, we visualize the point clouds and
 260 ground-truth bounding boxes of the two cars in Fig. 2. We
 261 identify two major sources of errors: viewpoint mismatch
 262 and mislocalization. Viewpoint mismatch occurs when ob-

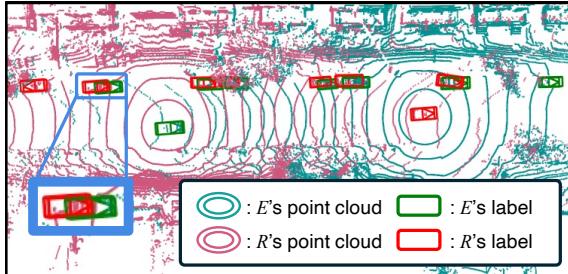


Figure 2. **Point and box discrepancies** between ego and reference cars on the real dataset [53].

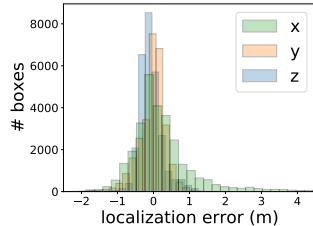


Figure 3. **Mislocalization** between E's and R's GT.

jects are obscured from *one* sensor's view due to occlusion or field of view limitations, while mislocalization results from GPS inaccuracies and synchronization delays. For instance, a communication delay of 0.1 seconds in a car traveling at 60 mph can result in a localization discrepancy of 2.7 meters. These errors can significantly degrade the quality of the learned detector f_E for the ego car E — the training labels are simply *noisy*. To further dive into these challenges, we measure the precision and recall of Y_R vs. the ego car E 's ground-truth labels to assess label quality, as shown in Table 1. Even after applying basic filtering commonly used in autonomous driving (*e.g.*, removing distant boxes or those with few points that are beyond E 's field of view), the label quality remains unsatisfactory (Table 1 ②). In the following sections, we introduce our pipeline R&B-POP to tackle these challenges.

3.3. Label-efficient box refinement

Preliminaries. We conduct a detailed analysis of the localization discrepancies in each coordinate (x forward, y leftward, z upward) between R 's and E 's overlapping ground-truth boxes, as illustrated in Fig. 3. Notice that a mere 0.5-meter discrepancy in the x and y coordinates can drastically reduce the IoU from 100% to 30%. Training with such inaccurate pseudo labels inevitably leads to suboptimal performance in E 's 3D detector. A refinement module for the labels is thus necessary!

Baseline approach with heuristics. To begin with, we adopt the algorithm proposed in Luo et al. [27], which refines boxes using heuristics. Specifically, multiple boxes are sampled around the initial noisy boxes, and the optimal boxes are selected based on the best alignment of edges and sizes between the boxes and point clouds. However, this

method requires certain conditions to achieve satisfactory performance, such as multiple trajectories at the same location, potentially limiting the applicability. As in Table 2, adapting it to our problem brings marginal gains, especially for the high IoU of 0.7 (AP ① 4.2 vs. ③ 10.3).

Label-efficient box ranker. To address this limitation, we propose to train a *box ranker* that evaluates the localization quality of given bounding boxes. Instead of predicting a 3D box from scratch (*i.e.*, a typical detection problem), learning to select and adjust among noisy candidates is a much easier task. *We thus expect learning such a ranker needs much fewer labeled data!* To investigate this idea, we sample a handful of E 's point clouds with ground-truths to train the ranker. We randomly sample multiple boxes around each annotated object box and crop point clouds outlined by those sampled boxes (with expansion). The training objective is dual: to regress the IoU between a sampled box and the annotated box, serving as the indicator of localization quality, and to estimate the offset to the annotated box, further refining its location. During inference, we use Y_R as initial boxes and sample N boxes around each. The top-ranked boxes are selected among all candidates to construct Y'_R as pseudo labels for training the 3D detector f_E for E (see Fig. 4). We adopt a neural network similar to PointNet [33] for the ranker for its simplicity. Please see the supplementary for details.

Coarse-to-fine (C2F) refinement. As previously discussed, minor time delays can result in large discrepancies of several meters. To address this and expand the search region, thereby increasing the number of high-quality bounding box candidates for our ranker, we employ a two-stage approach during *inference*, as illustrated in Fig. 5. In the first stage, we generate $\frac{N}{2}$ candidate boxes for each initial box by sampling translations from a wider range using a uniform distribution, while keeping the scale and pose of the boxes fixed. In the second stage, we select the top- K boxes from the first stage to serve as new initials and sample another $\frac{N}{2}$ total new boxes around them, this time considering all degrees of freedom (*i.e.*, translation, scale, and pose) but from a narrower range using normal distributions. This coarse-to-fine (C2F) strategy effectively bridges the large localization gap and enhances the refinement quality of Y'_R . With the box ranker and C2F, we raise the label quality from a recall of 55.3 to 65.2 and a precision of 71.4 to 79.0 upon basic filtering, as shown in Table 1 ② vs. ③, *using only 40 labeled frames*. Consequently, the performance of f_E also shows a significant gain from 22.0% to 38.0% in AP at IoU 0.5, as reported in Table 2 ① vs. ④.

Ranker-based filtering. The trained ranker not only refines the given boxes but also estimates their IoU with ground-truth boxes. Applying a threshold on predicted IoU effectively removes false positives, thus improving detection performance as shown in Table 4b.

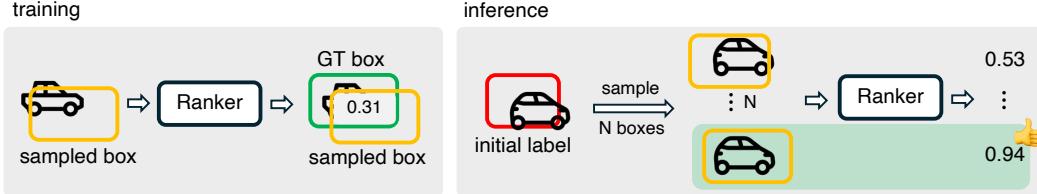


Figure 4. **Box ranker for refining localization error.** With a few annotated frames, we train a ranker that can estimate the quality of a given box. During inference for pseudo labels, we sample multiple candidates near the initial noisy box and choose the one with the best IoU.

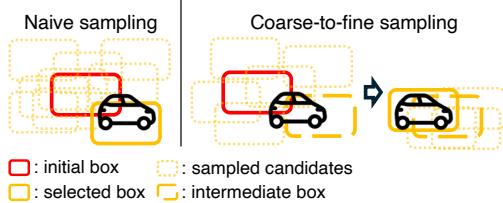


Figure 5. **Sampling methods for box refinement.** Proposed C2F is more effective in large mislocalization.

349 **3.4. Distance-based curriculum**

Viewpoint mismatch introduces false positives (*i.e.*, objects
350 should not be visible from E 's perspective) and negatives
351 (*i.e.*, objects should be visible to E but are not provided by R)
352 in Y_R . While false positives can be removed by filtering (*e.g.*,
353 basic and ranker-based filtering), false negatives are much
354 harder to be recovered. It becomes necessary to discover
355 new boxes from E 's perspective.
356

357 **Box discovery from the ego car E 's view.** Inspired by the
358 previous work [62], self-training [21, 48] is a popular tech-
359 nique to propagate labels to unlabeled data. This method
360 typically employs high-quality labels to train the base
361 detector and subsequently uses its predictions to generate new
362 pseudo labels for further fine-tuning cycles. However, as
363 discussed in previous sections, our initial (pseudo) labels
364 are inherently noisy, which can hinder the efficacy of self-
365 training. This leads to a pivotal question: *How to ensure the*
366 *quality of pseudo labels for effective self-training?*

367 **Key observation about distance.** We find out that there
368 exists a unique property in our learning scenario — the
369 extent of viewpoint mismatch is correlated with the distance
370 between E and R . Specifically, discrepancies are typically
371 reduced when the two are closer and increased when they
372 are distant (Fig. 7). This intuition leads us to use the distance
373 of two cars as an indicator of the quality of pseudo labels
374 provided by R . Building on this observation, we develop two
375 distance-based methods in the following.

376 **Distance-based curriculum for self-training.** We create a
377 high-quality subset of pseudo labels by applying a simple
378 distance threshold T_{E-R} to all frames, meaning that we trust
379 pseudo labels from R when two cars are close enough. In the
380 first round of self-training, the 3D detector f_E is exclusively
381 trained on this high-quality subset. In later rounds, we fine-
382 tune on *all frames* with pseudo labels predicted by f_E . This

approach propagates labels learned from confident frames to
383 unconfident ones.

384 **Distance-based filtering.** Self-training needs a filtering
385 mechanism to select high-quality predictions by the current
386 detector, which are then treated as true labels to supervise
387 the next round of detector training. Normally, this is done
388 by setting a fixed threshold T_c in prediction confidence¹.
389 Here, we employ a distance-based threshold, inspired by our
390 self-training procedure. Specifically, since we trust frames
391 with smaller ego-car-to-reference-car distances and train
392 the detector with them in the first round, the detector will
393 inherently be overly confident in these frames. As such, a
394 higher threshold shall be assigned when selecting pseudo
395 labels from them. We implement this idea by increasing the
396 confidence threshold with a negative linear function of the
397 distance (*i.e.*, $T_c + \lambda/distance(E, R)$).

398 Put together, these two distance-based approaches not
399 only uncover boxes that should be visible to E but also
400 preserve the quality of pseudo labels for self-training. As
401 shown in Table 1 ④, self-training with distance-based cur-
402 riculum further improves label quality from 79.0% to 87.7%
403 in precision and 65.2% to 74.4% in recall, resulting in an
404 enhancement of f_E 's performance from AP of 38.0% to
405 56.5%, as detailed in Table 2 ④ vs. ⑩. As a reference, we
406 show the predicted label quality on X_E using f_R in Table 1
407 ⑤, simulating the ideal case where the object detector can
408 be shared. Our R&B-POP achieves similar label quality,
409 demonstrating the applicability of our setting for learning
410 high-quality detectors from others' predictions.

412 **3.5. Overall pipeline**

413 Putting everything together, our overall *offline* pipeline in-
414 volves the following steps (Fig. 6).

415 Step 0. Ranker training with few annotated labels
416 (**Sec. 3.3**).

417 Step 1. First-round self-training: Preparing pseudo
418 labels after receiving R 's predictions (applying basic filter-
419 ing), further improving labels with ranker + C2F and ranker-
420 thresholding (**Sec. 3.3**), and then training the detector f_E
421 with closer frames (**Sec. 3.4**).

422 Step 2. Second-round self-training: Preparing pseudo

¹We note that the detector's confidence is not the same as the IoU predicted by the ranker in Sec. 3.3.

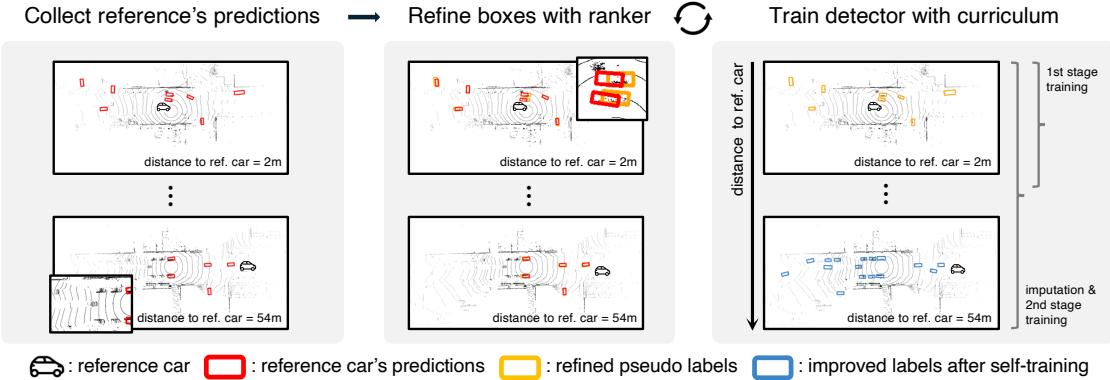


Figure 6. **Overall pipeline of R&B-POP.** The ego car first receives reference’s predictions which contain inherent noises (Sec. 3). It refines their localization with proposed box ranker (Sec. 3.3). Then, it creates high-quality pseudo labels by distance-based curriculum for self-training (Sec. 3.4).

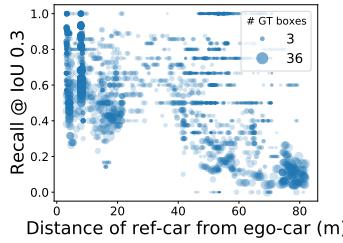


Figure 7. **Quality of pseudo labels from R’s predictions** drops when two cars are farther apart.

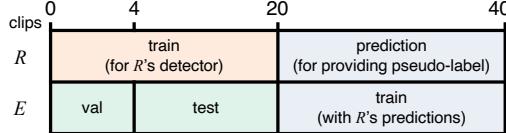


Figure 8. **Dataset split.** We re-split V2V4Real [53] for our setting.

423 labels after receiving f_E ’s prediction (applying distance-
424 filtering in Sec. 3.4), further improving labels with ranker
425 + C2F and ranker-thresholding (Sec. 3.3), and then training
426 the detector f_E with all frames.

427 4. Experiments

428 4.1. Experimental setups

429 **Datasets.** To validate the effectiveness of our method,
430 we conduct experiments primarily on the V2V4Real
431 dataset [53], which consists of 40 clips with a total of 18k
432 frames by driving two cars, Tesla and Honda, together within
433 100m. LiDAR points are acquired with a Velodyne VLP-32
434 LiDAR sensor. The dataset provides annotations for different
435 types of vehicles, such as cars and trucks. (Please see
436 additional results on the OPV2V dataset [52] in the sup-
437 plementary.)

438 To align with our research purpose, we re-split the orig-
439 inal data into three portions: “R pretraining”, “R predic-
440 tion/E training”, and “E validation/test” (Fig. 8). Specifi-

cally, we split them into two subsets containing 20 clips and use the first subset to pre-train R ’s detector f_R . Then we inference on the second subset to provide pseudo labels Y_R for training E ’s detector f_E together with E ’s point clouds. We validate and test the E ’s performance on the first subset by splitting it into 20% and 80%. Our re-split gives 4,488 frames for R pretraining, $4,463 \times 2$ frames for E training, and 870 and 3,618 frames for E validation/test respectively. The performance in the paper is reported on E ’s test set.

Evaluation. We follow Xu et al. [53] to merge different types of vehicles (*e.g.*, cars, trucks) into a single category². We report the average precision (AP) of detectors in the bird’s-eye view with IoU thresholds of 0.5 and 0.7. Specifically, we set the region of interest to [-80, 80]m for the heading direction and [-40, 40]m for the direction perpendicular to the moving direction. We also report the AP on different depth ranges [0-30, 30-50, 50-80, 0-80]m following Luo et al. [27].

Implementation. We conduct experiments with PointPillars [20] as a default **detector**. We train it with 60 epochs and a batch size of 64 on 8 NVIDIA Tesla P100 GPUs. We use Adam optimizer and an initial learning rate of 2e-3 dropped to 2e-5 by cosine annealing decaying strategy [25]. For training the **box ranker**, a PointNet [33] specified in the supplementary, we use 40 annotated frames (< 1% of training data) to generate 11k samples. In the *offline* ranker inference, we sample $N = 512$ boxes around each prediction provided by the reference car. We first sample 256 boxes in the coarse stage, select top-3 boxes, and then sample the remaining 256 boxes near the selected boxes in the fine stage. For curriculum learning, we set T_{E-R} to 40m. Also, we set the ranker

²We note that V2V4Real [53] does not label objects beyond vehicles and the data distributions across different types of vehicles are largely imbalanced. Thus, it is infeasible to study multi-class vehicle detection. That said, extending our approach to a multi-class setup would be straightforward if a suitable dataset is available. The key is to make the ranker category aware. Please refer to our experiments on extending the ranker to a multi-class setting in the supplementary.

Table 2. Main results: validation of the proposed learning scenario and methods. The results indicate a new research problem of *learning with others' predictions* has inherent challenges. With proposed R&B-POP, we significantly close the gap to the upper bound that directly uses ego car's ground-truth labels (⑩ 56.5 vs. ♠ 58.4). The performance is reported on PointPillars [20] with 32-beam LiDAR. : uses GT labels. : our proposed methods.

pseudo label	box refinement	self-training	AP @ IoU 0.5				AP @ IoU 0.7			
			0-30m	30-50m	50-80m	0-80m	0-30m	30-50m	50-80m	0-80m
① R 's pred	-	-	34.7	13.5	8.6	22.0	7.6	2.2	1.5	4.2
② R 's GT	-	-	29.7	14.1	7.3	19.6	5.9	2.2	1.8	3.7
③ R 's pred	heuristic [27]	-	53.2	22.0	16.9	37.8	16.5	4.5	3.9	10.3
④ R 's pred	ranker	-	50.3	24.7	18.2	38.0	33.6	12.9	8.9	22.9
⑤ R 's pred	-	naive [62]	45.9	18.7	16.5	32.4	13.8	4.6	5.9	9.2
⑥ R 's pred	heuristic [27]	naive [62]	50.4	19.6	15.4	35.4	13.8	4.2	3.6	8.9
⑦ R 's pred	ranker	naive [62]	60.6	29.7	19.2	45.0	40.8	16.7	9.5	28.0
⑧ R 's pred	-	distance-based curriculum	57.3	29.6	21.0	42.5	21.0	5.9	3.9	12.7
⑨ R 's pred	heuristic [27]	distance-based curriculum	60.5	25.5	17.0	43.2	18.3	4.4	3.0	11.0
⑩ R 's pred	ranker	distance-based curriculum	73.3	43.3	23.3	56.5	47.1	21.1	10.0	32.6
♠ E 's GT	-	-	75.2	45.9	28.8	58.4	51.7	25.4	14.8	36.3



Figure 9. Qualitative results. The quality of pseudo labels is gradually improved with the proposed R&B-POP. Our ranker successfully fixes mislocalization errors, and distance-based curriculum further discovers new objects from E 's view.

threshold to 0.5, and λ for the distance-based threshold to 1 with a fixed confidence threshold T_c of 0.2. Please refer to the supplementary material for more details.

4.2. Experimental results

We first demonstrate our scenario, *learning with others' predictions*, with a basic setup: R and E both have 32-beam sensors with PointPillars [20], but only R 's detector was pre-trained. Table 2 compares different methods for pseudo labels, including baselines such as standard self-training [62] and heuristic-based label refinement [27]. For fair comparisons, we note that we only utilize annotated boxes from 40 frames to train our ranker, not to train the detector. For the heuristic-based refinement [27], as our dataset has no repeated traversal to estimate movable objects, we use RANSAC [11] instead.

Firstly, our ranker brings a notable gain over heuristics-based refinement (③ 10.3 vs. ④ 22.9, ⑥ 8.9 vs. ⑦ 28.0, and

⑨ 11.0 vs. ⑩ 32.6 on AP at IoU 0.7), demonstrating its effectiveness to address mislocalization. Secondly, our distance-based curriculum consistently improves the performance over the standard self-training (⑤-⑦ 32.4/35.4/45.0 vs. ⑧-⑩ 42.5/43.2/56.5), demonstrating the necessity of using higher-quality samples for self-training. Finally, by comparing with detectors trained with E 's ground-truth, our method achieves on par with the upper bound (⑩ 56.5/32.6 vs. ♠ 58.4/36.3).

Fig. 9 visualizes the improvement of pseudo labels with our method. The ranker successfully refines mislocalized pseudo labels provided by R . Moreover, our distance-based curriculum discovers new bounding boxes that E couldn't receive from R 's perception, without introducing many false positives.

4.2.1. Ablation study

Analysis on ranker training. We first check the performance of the ranker trained with different number of anno-

489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505

Table 3. Ablations on box ranker. (a) The training of the ranker requires very few human labels. Using simulation data further eliminates the need yet performs on par. (b) The proposed inference strategies effectively generate high-quality pseudo labels, resulting in better performance.

training data	# annot. frames	AP @ IoU	
		0.5	0.7
E 's GT	20	54.8	30.0
	40	56.5	32.6
	80	55.5	32.8
Simulation [52]	21k	52.2	28.4

(a) Analysis on training labels.

Table 4. Ablations on curriculum self-training. The results show that each individual component in our proposed method contributes to the optimal performance. (a) Our curriculum selectively leverages useful frames during each stage of training. (b) Our label thresholding effectively discards noisy labels for training, resulting in improved performance.

	pseudo label		AP @ IoU	
	stage 1	stage 2	0.5	0.7
①	0-90m	0-90m	45.0	28.0
②	0-40m	0-40m	50.3	24.7
③	0-40m	40-90m	51.1	26.0
④	40-90m	40-90m	33.5	20.0
⑤	0-40m	0-90m	56.5	32.6

(a) Analysis on different curriculums.

offset	sampling		AP @ IoU	
	naive	C2F	0.5	0.7
①	✓		50.2	28.5
②	✓	✓	56.5	30.7
③	✓		56.5	32.6
④		✓	54.4	31.0

(b) Analysis on inference strategies.

ranker threshold	distance-based threshold	AP @ IoU	
		0.5	0.7
		50.2	26.2
	✓	54.9	30.3
✓		52.2	28.3
✓	✓	56.5	32.6

(b) Analysis on box filtering.

506 tated frames (*i.e.*, 20, 40, and 80) in **Table 3a**. Notably, we
507 observe a performance boost when the ranker is trained with
508 40 frames compared to 20 frames, and the performance gain
509 diminishes with more than 40 frames. This demonstrates that
510 the ranker already perform well with very few labeled data
511 (*i.e.*, 1% of total). Moreover, we train our ranker with 21k
512 samples generated by CARLA simulator [52] and achieves
513 on par performance, exploring to remove the need of human
514 labels. As shown, our ranker trained with off-the-shelf simu-
515 lated data can achieve 28.4% AP at IoU 0.7, higher than the
516 11.0% AP at IoU 0.7 achieved by [27] (**Table 2 ⑨**).

517 **Analysis on ranker inference.** We investigate the impact
518 of inference strategies for the refinement on the final de-
519 tection performance in **Table 3b**. The results indicate that
520 sampling boxes (① 28.5 vs. ② 30.7) and coarse-to-fine refine-
521 ment (② 30.7 vs. ③ 32.6) contribute to superior performance,
522 especially for the fine-grained quality of IoU 0.7. We also
523 observe the benefit of using predicted offsets to further refine
524 box locations (④ 31.0 vs. ③ 32.6). This demonstrates the
525 effectiveness of our sophisticated box refinement strategies.

526 **Analysis on curriculum for self-training.** In our method,
527 we split the training data into two subsets with the threshold
528 T_{E-R} of 40m. We train the detector on different combina-
529 tions of subsets and check its performance (**Table 4a**). We
530 observe that using initial low-quality frames to train the
531 model gives significantly lower performance (④ 33.5 vs. ⑤
532 56.5). Also, we see that the performance escalates as we
533 utilize more high-quality frames during the next self-training
534 stage (② 50.3, ③ 51.1 vs. ⑤ 56.5), verifying the effectiveness

of our curriculum choice.

Analysis on box filtering for self-training. To prevent the
535 model from introducing false positive boxes during the self-
536 training, we design a distance-based confidence threshold
537 and ranker-based filtering. As shown in **Table 4b**, the de-
538 tector improves with our strategies. This indicates a good bal-
539 ance between recall and precision provided by our method,
540 resulting in better detection performance.

4.2.2. Additional empirical studies

We leave additional results in the supplementary, including
544 applications to different scenarios, the ideal scenario where
545 the object detector can be shared, analysis on ranker and
546 self-training, and extension to another dataset.

5. Conclusion and Discussion

In this work, we have introduced *learning with others' pre-
549 dictions*, a new way to train a 3D detector with the predic-
550 tions of reference units. We have systematically identified
551 the inevitable task-specific problems: false positive, false
552 negative, and noisy boxes due to either viewpoint mismatch
553 or synchronization/GPS errors. Next, we have proposed to
554 improve the quality of pseudo labels by two solutions: A box
555 ranker and distance-based curriculum self-training. We have
556 demonstrated a wide applicability of our learning scenario
557 with different detectors, sensors, and domains.

559 **References**

- 560 [1] Stefan Baur, Frank Moosmann, and Andreas Geiger. Liso:
561 Lidar-only self-supervised 3d object detection. In *ECCV*,
562 2024.
- 563 [2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and
564 Jason Weston. Curriculum learning. In *ICML*, 2009.
- 565 [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora,
566 Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Gi-
567 ancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal
568 dataset for autonomous driving. In *CVPR*, 2020.
- 569 [4] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and
570 Song Fu. F-cooper: Feature based cooperative perception for
571 autonomous vehicle edge computing system using 3d point
572 clouds. In *ACM/IEEE Symposium on Edge Computing*, 2019.
- 573 [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geof-
574 frey Hinton. A simple framework for contrastive learning of
575 visual representations. In *ICML*, 2020.
- 576 [6] Xinlei Chen and Kaiming He. Exploring simple siamese
577 representation learning. In *CVPR*, 2021.
- 578 [7] Xiangyu Chen, Zhenzhen Liu, Katie Z Luo, Siddhartha Datta,
579 Adhitya Polavaram, Yan Wang, Yurong You, Boyi Li, Marco
580 Pavone, Wei-Lun Chao, et al. Diffubox: Refining 3d object
581 detection with point diffusion. In *NeurIPS*, 2024.
- 582 [8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d
583 spatio-temporal convnets: Minkowski convolutional neural
584 networks. In *CVPR*, 2019.
- 585 [9] Minh-Quan Dao, Holger Caesar, Julie Stephany Berrio, Mao
586 Shan, Stewart Worrall, Vincent Frémont, and Ezio Malis.
587 Label-efficient 3d object detection for road-side units. In *IV*,
588 2024.
- 589 [10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yany-
590 ong Zhang, and Houqiang Li. Voxel r-cnn: Towards high
591 performance voxel-based 3d object detection. In *AAAI*, 2021.
- 592 [11] Martin A Fischler and Robert C Bolles. Random sample
593 consensus: a paradigm for model fitting with applications to
594 image analysis and automated cartography. *Communications
595 of the ACM*, 1981.
- 596 [12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we
597 ready for autonomous driving? the kitti vision benchmark
598 suite. In *CVPR*, 2012.
- 599 [13] Cong Hao and Deming Chen. Software/hardware co-design
600 for multi-modal multi-task learning in autonomous systems.
601 In *AICAS*, 2021.
- 602 [14] Cong Hao, Atif Sarwari, Zhijie Jin, Husam Abu-Hamed,
603 Daryl Sew, Yuhong Li, Xinheng Liu, Bryan Wu, Dongdong
604 Fu, Junli Gu, et al. A hybrid gpu+ fpga system design for
605 autonomous driving cars. In *SiPS*, 2019.
- 606 [15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross
607 Girshick. Momentum contrast for unsupervised visual repre-
608 sentation learning. In *CVPR*, 2020.
- 609 [16] Deepthi Hegde, Velat Kilic, Vishwanath Sindagi, A Brinton
610 Cooper, Mark Foster, and Vishal M Patel. Source-free unsu-
611 pervised domain adaptation for 3d object detection in adverse
612 weather. In *ICRA*, 2023.
- 613 [17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon,
614 Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua
Bengio. Learning deep representations by mutual in-
formation estimation and maximization. *arXiv preprint
arXiv:1808.06670*, 2018.
- 615 [18] Shixin Hong, Yu Liu, Zhi Li, Shaohui Li, and You He. Multi-
616 agent collaborative perception via motion-aware robust com-
617 munication network. In *CVPR*, 2024.
- 618 [19] Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Si-
619 heng Chen. Communication-efficient collaborative perception
620 via information filling with codebook. In *CVPR*, 2024.
- 621 [20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou,
622 Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders
623 for object detection from point clouds. In *CVPR*, 2019.
- 624 [21] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient
625 semi-supervised learning method for deep neural networks.
626 In *ICML Workshop*, 2013.
- 627 [22] Siyang Li, Xiangxin Zhu, Qin Huang, Hao Xu, and C-C Jay
628 Kuo. Multiple instance curriculum learning for weakly su-
629 pervised object detection. *arXiv preprint arXiv:1711.09191*,
630 2017.
- 631 [23] Chuandong Liu, Chenqiang Gao, Fangcen Liu, Jiang Liu,
632 Deyu Meng, and Xinbo Gao. Ss3d: Sparsely-supervised 3d
633 object detection from point cloud. In *CVPR*, 2022.
- 634 [24] Chang Liu, Xiaoyan Qian, Bin Xiao Huang, Xiaojuan Qi, Ed-
635 mund Lam, Siew-Chong Tan, and Ngai Wong. Multimodal
636 transformer for automatic 3d annotation and object detection.
637 In *ECCV*, 2022.
- 638 [25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay
639 regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- 640 [26] Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen,
641 and Yanfeng Wang. An extensible framework for open het-
642 erogeneous collaborative perception. In *ICLR*, 2024.
- 643 [27] Katie Z Luo, Zhenzhen Liu, Xiangyu Chen, Yurong You,
644 Sagie Benaim, Cheng Perng Phoo, Mark Campbell, Wen
645 Sun, Bharath Hariharan, and Kilian Q. Weinberger. Reward
646 finetuning for faster and more accurate unsupervised object
647 discovery. In *NeurIPS*, 2023.
- 648 [28] Tao Ma, Xuemeng Yang, Hongbin Zhou, Xin Li, Botian Shi,
649 Junjie Liu, Yuchen Yang, Zhizheng Liu, Liang He, Yu Qiao,
650 et al. Detzero: Rethinking offboard 3d object detection with
651 long-term sequential point clouds. In *ICCV*, 2023.
- 652 [29] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen
653 Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired
654 unsupervised perception and prediction in autonomous driv-
655 ing. In *ECCV*, 2022.
- 656 [30] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen
657 Yan, Scott Ettinger, and Dragomir Anguelov. Unsupervised
658 3d perception with 2d vision-language distillation for au-
659 tonomous driving. In *ICCV*, 2023.
- 660 [31] Tai-Yu Pan, Chenyang Ma, Tianle Chen, Cheng Perng Phoo,
661 Katie Z Luo, Yurong You, Mark Campbell, Kilian Q Wein-
662 berger, Bharath Hariharan, and Wei-Lun Chao. Pre-training
663 lidar-based 3d object detectors through colorization. In *ICLR*,
664 2024.
- 665 [32] Xidong Peng, Xinge Zhu, and Yuexin Ma. Cl3d: Unsuper-
666 vised domain adaptation for cross-lidar 3d detection. In *AAAI*,
667 2023.
- 668 [33] Xidong Peng, Xinge Zhu, and Yuexin Ma. Cl3d: Unsuper-
669 vised domain adaptation for cross-lidar 3d detection. In *AAAI*,
670 2023.

- 671 [33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas.
672 Pointnet: Deep learning on point sets for 3d classification and
673 segmentation. In *CVPR*, 2017.
- 674 [34] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo,
675 Boyang Deng, and Dragomir Anguelov. Offboard 3d object
676 detection from point cloud sequences. In *CVPR*, 2021.
- 677 [35] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu
678 Sebe. Self paced deep learning for weakly supervised object
679 detection. *TPAMI*, 2018.
- 680 [36] Jenny Seidenschwarz, Aljoša Ošep, Francesco Ferroni, Simon
681 Lucey, and Laura Leal-Taixé. Semoli: What moves together
682 belongs together. In *CVPR*, 2024.
- 683 [37] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointr-
684 cnn: 3d object proposal generation and detection from point
685 cloud. In *CVPR*, 2019.
- 686 [38] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping
687 Shi, Xiaogang Wang, and Hongsheng Li. Pv-renn: Point-
688 voxel feature set abstraction for 3d object detection. In *CVPR*,
689 2020.
- 690 [39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien
691 Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou,
692 Yuning Chai, Benjamin Caine, et al. Scalability in perception
693 for autonomous driving: Waymo open dataset. In *CVPR*,
694 2020.
- 695 [40] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo
696 Nebot, and Stewart Worrall. Ms3d: Leveraging multiple
697 detectors for unsupervised domain adaptation in 3d object
698 detection. In *ITSC*, 2023.
- 699 [41] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J
700 Guibas. 3dioumatch: Leveraging iou prediction for semi-
701 supervised 3d object detection. In *CVPR*, 2021.
- 702 [42] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin
703 Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-
704 to-vehicle communication for joint perception and prediction.
705 In *ECCV*, 2020.
- 706 [43] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan,
707 Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar
708 from visual depth estimation: Bridging the gap in 3d object
709 detection for autonomous driving. In *CVPR*, 2019.
- 710 [44] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath
711 Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-
712 Lun Chao. Train in germany, test in the usa: Making 3d object
713 detectors generalize. In *CVPR*, 2020.
- 714 [45] Yan Wang, Junbo Yin, Wei Li, Pascal Frossard, Ruigang
715 Yang, and Jianbing Shen. Ssda3d: Semi-supervised domain
716 adaptation for 3d object detection from point cloud. In *AAAI*,
717 2023.
- 718 [46] Qiming Xia, Jinhao Deng, Chenglu Wen, Hai Wu, Shaoshuai
719 Shi, Xin Li, and Cheng Wang. Coin: Contrastive instance feature
720 mining for outdoor 3d object detection with very limited
721 annotations. In *ICCV*, 2023.
- 722 [47] Qiming Xia, Wei Ye, Hai Wu, Shijia Zhao, Leyuan Xing, Xun
723 Huang, Jinhao Deng, Xin Li, Chenglu Wen, and Cheng Wang.
724 Hinted: Hard instance enhanced detector with mixed-density
725 feature fusion for sparsely-supervised 3d object detection. In
726 *CVPR*, 2024.
- 727 [48] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V
728 Le. Self-training with noisy student improves imagenet clas-
729 sification. In *CVPR*, 2020.
- 730 [49] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas
731 Guibas, and Or Litany. Pointcontrast: Unsupervised pre-
732 training for 3d point cloud understanding. In *ECCV*, 2020.
- 733 [50] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei
734 Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view
735 semantic segmentation with sparse transformers. In *CoRL*,
736 2022.
- 737 [51] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-
738 Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything
739 cooperative perception with vision transformer. In *ECCV*,
740 2022.
- 741 [52] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and
742 Jiaqi Ma. Opv2v: An open benchmark dataset and fusion
743 pipeline for perception with vehicle-to-vehicle communica-
744 tion. In *ICRA*, 2022.
- 745 [53] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang,
746 Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong,
747 Rui Song, et al. V2v4real: A real-world large-scale dataset
748 for vehicle-to-vehicle cooperative perception. In *CVPR*, 2023.
- 749 [54] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embed-
750 ded convolutional detection. *Sensors*, 2018.
- 751 [55] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and
752 Xiaojuan Qi. St3d: Self-training for unsupervised domain
753 adaptation on 3d object detection. In *CVPR*, 2021.
- 754 [56] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and
755 Xiaojuan Qi. St3d++: Denoised self-training for unsupervised
756 domain adaptation on 3d object detection. *TPAMI*, 2022.
- 757 [57] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li
758 Wang, Xinyu Zhang, and Peng Chen. Bevheight: A robust
759 framework for vision-based roadside 3d object detection. In
760 *CVPR*, 2023.
- 761 [58] Yuxue Yang, Lue Fan, and Zhaoxiang Zhang. Mixsup: Mixed-
762 grained supervision for label-efficient lidar-based 3d object
763 detection. *arXiv preprint arXiv:2401.16305*, 2024.
- 764 [59] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya
765 Jia. Ipod: Intensive point-based object detector for point
766 cloud. *arXiv preprint arXiv:1812.05276*, 2018.
- 767 [60] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-
768 Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposal-
769 contrast: Unsupervised pre-training for lidar-based 3d object
770 detection. In *ECCV*, 2022.
- 771 [61] Yurong You, Katie Z Luo, Xiangyu Chen, Junan Chen, Wei-
772 Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and
773 Kilian Q. Weinberger. Hindsight is 20/20: Leveraging past
774 traversals to aid 3d perception. In *ICLR*, 2022.
- 775 [62] Yurong You, Katie Z Luo, Cheng Perng Phoo, Wei-Lun Chao,
776 Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q.
777 Weinberger. Learning to detect mobile objects from lidar
778 scans without labels. In *CVPR*, 2022.
- 779 [63] Yurong You, Cheng Perng Phoo, Katie Z Luo, Travis Zhang,
780 Wei-Lun Chao, Bharath Hariharan, Mark Campbell, and Kil-
781 lian Q. Weinberger. Unsupervised adaptation from repeated
782 traversals for autonomous driving. In *NeurIPS*, 2022.

- 783 [64] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang,
784 Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui
785 Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-
786 infrastructure cooperative 3d object detection. In *CVPR*,
787 2022.
- 788 [65] Dingyuan Zhang, Dingkang Liang, Zhikang Zou, Jingyu Li,
789 Xiaoqing Ye, Zhe Liu, Xiao Tan, and Xiang Bai. A sim-
790 ple vision transformer for weakly semi-supervised 3d object
791 detection. In *ICCV*, 2023.
- 792 [66] Lunjun Zhang, Anqi Joyce Yang, Yuwen Xiong, Sergio Casas,
793 Bin Yang, Mengye Ren, and Raquel Urtasun. Towards unsu-
794 pervised object detection from lidar point clouds. In *CVPR*,
795 2023.
- 796 [67] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and
797 Vladlen Koltun. Point transformer. In *ICCV*, 2021.
- 798 [68] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning
799 for point cloud based 3d object detection. In *CVPR*, 2018.
- 800 [69] Ziyue Zhu, Qiang Meng, Xiao Wang, Ke Wang, Liujiang Yan,
801 and Jian Yang. Curricular object manipulation in lidar-based
802 object detection. In *CVPR*, 2023.

Supplementary Material for Learning 3D Perception from Others' Predictions

803 In this supplementary material, we provide implementation details and experimental results in addition to the main paper:
804 • Sec. S1: provides additional discussion of the scenario.
805 • Sec. S2: provides further implementation details.
806 • Sec. S3: provides additional empirical studies.

807 S1. Additional Discussion

808 **More details on problem setup.** In our study, the overall goal is to develop a 3D perception model that can be deployed in an
809 online setting. A conventional development process can typically be decomposed into four stages:

- 810 • Stage 1: data collection (online)
811 • Stage 2: data annotation, often by humans (offline)
812 • Stage 3: model training and validation (offline)
813 • Stage 4: model deployment and evaluation (online)

814 We exactly follows the four stages, except that in Stage 1, we assume that some nearby agents (*e.g.*, robotaxi, roadside unit)
815 share their predictions as pseudo labels (*e.g.*, bounding boxes). We study how to leverage these pseudo labels to reduce human
816 annotation in Stage 2 while maintaining the trained model quality in Stage 3.

817 Here, the stage where the ego car collects the nearby agents' predictions is the first stage, which is online. We note that
818 there is no training or inference regarding the ego car's detector during this stage. After we collect the pseudo labels from the
819 nearby agents, we refine the noisy pseudo labels with our proposed method and train the ego car's detector, which is offline
820 (Stages 2 and 3). For our final model evaluation (Stage 4), which is online, the detector's computational cost is exactly the
821 same as the standard detector.

822 **Discussion on offboard methods.** On the surface, both the offboard methods [28, 34] and our learning scenario assume
823 the existence of a pre-trained model. However, the accessibility to the model is different. More specifically, in the offboard
824 methods, the pre-trained offboard model is directly accessible. One can use it to label the unlabeled data collected by the ego
825 car. The resulting pseudo-labeled data can then be used to train the final onboard model. However, in our scenario, we do not
826 have direct access to the pre-trained model, as it is deployed on the nearby agent, not the ego car. As such, we cannot use it to
827 label the unlabeled data collected by the ego car. What we can access are the nearby agent's predictions on the data it collects,
828 and we attempt to use them as pseudo labels of to train the ego car's onboard model.

829 **Limitations and future work.** The ego car's detector can benefit from other reference units, such as roadside units in
830 smart cities. In our future work, we plan to investigate more diverse scenarios. Moreover, the ego car and the reference car
831 using different modalities would be a direction to explore further. We believe that our findings and approach have set the
832 foundation for it. At a high level, our approach is sensor and modality-agnostic. Regardless of the type of sensors and detectors
833 (camera-based or LiDAR-based) used, if we aim at 3D perception, they will produce 3D bounding boxes as pseudo labels. Our
834 method does not necessitate a specific model for providing the pseudo labels and can be easily adapted to various sensor types.
835 We leave this extension to future work.

836 S2. Additional Implementation Details

837 The entire training pipeline for the experiments takes 2.5 hours with eight NVIDIA P100 GPUs. During training, we apply
838 conventional data augmentation techniques such as rotation, scaling, and flipping following Xu et al. [53]. For training
839 SECOND [54] in our ablation study, we set the number of epochs to 40, with the remaining training settings the same as
840 PointPillars [20]. For the domain adaptation experiments, we decrease the initial learning rate to 2e-4 and decay it to 2e-6, and
841 fine-tune the model using the pre-trained parameters.

842 **Ranker architecture.** In the main paper, we train a ranker to select the best candidate from the sampled boxes. Specifically,
843 we build our ranker upon PointNet [33], taking a normalized box and its corresponding points as inputs (see Fig. S1). We use
844 two linear layers with ReLU non-linearity for both score head and offset head. Our box ranker's goal is similar to IoU scoring
845 methods adopted in the existing detectors [10, 38]. However, we are motivated to build a ranker assuming we already have
846 pseudo labels for objects but with a certain amount of localization errors. We demonstrate that this objective can be easily
847 achieved with only a few frames (or even without any frame if we are able to use simulation data; Table 3 (a)) and without

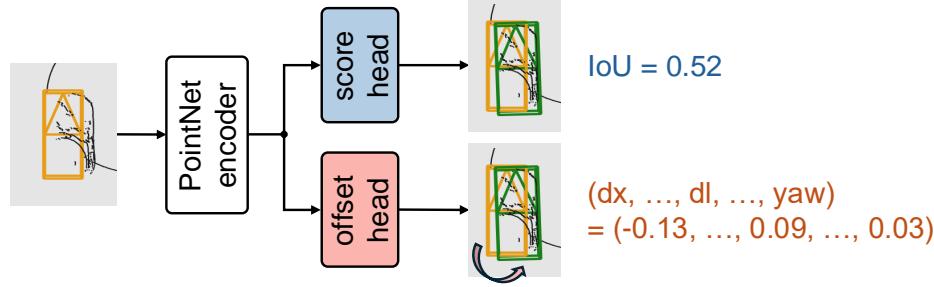


Figure S1. **Ranker architecture.** We give the initial noisy bounding box together with its nearby point clouds to the ranker as inputs and predict the IoU and offset to the ground-truth box.

sophisticated architectural design. Therefore, we keep it to a very simple regressor that can already be satisfactorily used to demonstrate the effectiveness in refining the noisy pseudo labels.

Ranker training. We crop the point cloud to the range of $\times 3$ of the box size, and predict the IoU and the offset from the object. To prepare the training data, we take the first two frames for each of the 20 clips. We then sample approximately 100 boxes around each ground-truth label, crop out the point cloud around the sampled boxes to serve as training input, and compute the IoU and offset information for labels of the training set. During the ranker training, we also simulate random occlusion and point dropping. For the design of the loss function, we use a weighted combination: $\mathcal{L}_{\text{total}} = 5 * \mathcal{L}_{\text{IoU}} + \mathcal{L}_{\text{offset}}$. Here, the IoU loss \mathcal{L}_{IoU} is defined by the mean squared error between \hat{y} and y , where \hat{y} denotes estimated IoU and y denotes the actual IoU. For the offset prediction loss, since we only use the offset from the top-k boxes, the offset will most likely be applied to high IoU candidate boxes with smaller offset values, as shown in Fig. S2. As such, we are able to prevent training samples with lower IoU and larger offsets from dominating the training process. Based on this observation, we test several different loss functions and see that the ranker performed best using Smooth L1 loss, without adjusting for the sampled box’s offset loss if the IoU is less than 0.3.

Ranker inference. For the ranker refinement module, we sample $N = 512$ boxes in total for both sampling strategies. In the naive sampling strategy, 512 boxes are sampled using Gaussian distributions with translational noise (on xyz) having a standard deviation of 1, and scaling and rotational noise (on lwh, yaw) having a standard deviation of 0.1, all with a mean of zero. For the C2F sampling, we employ both coarse and fine sampling. During coarse sampling, noise is uniformly sampled from $[-1.0, 1.0]$ for xy and from a Gaussian distribution with a standard deviation of 0.5 for z to help us sample $\frac{N}{2} = 256$ boxes. We then select the top $k = 3$ boxes based on the IoU reported by the ranker, apply the predicted offsets to these boxes, and proceed to sample a total of 256 boxes around each candidate for the fine sampling stage. In this stage, translational noise is sampled from a Gaussian distribution with a standard deviation of 0.25, noise for height and width with a standard deviation of 0.2, length with a standard deviation of 0.4, and rotational noise with a standard deviation of 0.1. We ultimately select the box with the highest IoU among all 256 sampled boxes and apply the predicted offset to obtain the refined label.

S3. Additional Empirical Studies

S3.1. Comparison to semi-supervised method

Table S1. **Experiments on semi-supervised approach.** Our method and the semi-supervised method, 3DIoUMatch [41], can complement each other. The performance is reported on AP at IoU 0.7. * and † indicate supervision and approach, respectively.

	labeled 40 frames*	R's pred*	3DIoUMatch†	Ranker (Sec. 3.3)†	Curriculum (Sec. 3.4)†	0-30m	30-50m	50-80m	0-80m
①	✓		✓			60.7	24.4	6.0	37.1
②	✓		✓	✓	✓	65.1	34.1	12.6	41.7
③	✓		✓	✓		60.6	30.7	8.6	39.5
④	✓		✓	✓	✓	68.1	36.8	12.9	44.4

We investigate a direct semi-supervised approach, using the 40 annotated frames and other unannotated frames of the ego car’s data to train the detector. We apply 3DIoUMatch [41], a widely used and representative semi-supervised learning approach in this setting. We note that the official code of 3DIoUMatch used the PV-RCNN detector [38], not the PointPillar detector [20] in our main paper. As such, we rerun our approach using the PV-RCNN detector for a fair comparison.

848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870

873
874
875
876

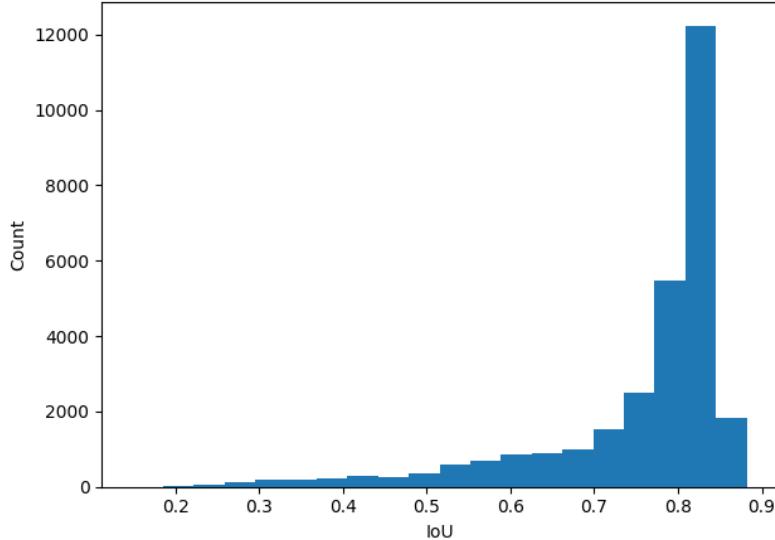


Figure S2. **Top IoU predicted from sampled boxes by ranker.** We computed the statistics for the IoU of the sampled boxes selected by the ranker during refinement. Then, the offset predicted by the ranker was applied to these selected boxes. The result indicates that the offset is most frequently used when the IoU of the box is sufficiently high.

877 **Table S1** summarizes the results. ① is the result of 3DIoUMatch, and ② is the result of our approach (*cf.*, **Table 2** ⑩, but
 878 using PV-RCNN as the detector). We see that our approach outperforms 3DIoUMatch, demonstrating the value of using
 879 reference cars’ predictions as auxiliary supervisions.

880 More importantly, we explore complementary nature of the two approaches. Specifically, we use our ranker to refine
 881 reference cars’ predictions and add those high-quality ones (*i.e.*, <40 meters, defined in [Sec. 3.4](#)) as extra labels to 3DIoUMatch.
 882 ③ shows the results: we see a 2.4% boost in 0-80 meters against ①, justifying the compatibility of ours and 3DIoUMatch. On
 883 top of ③, we further apply our distance-based curriculum for self-training ([Sec. 3.4](#)), using 3DIoUMatch’s predictions on all
 884 the data as pseudo labels to re-train the detector. ④ shows the results: we see another 4.9% boost against ③ and 2.7% boost
 885 against ②. In sum, these results demonstrate 1) the effectiveness of our approach in leveraging reference cars’ predictions as
 886 supervision (③ and ② vs. ①) and 2) the compatibility of our approach with existing direct semi-supervised learning approaches
 887 to further boost the accuracy (④ vs. ③ and ②). We view such compatibility as a notable strength: it demonstrates our approach
 888 as a valuable add-on when nearby agents’ predictions are available.

889 S3.2. Effect of the number of training data

Table S2. Number of training data and performance.

# clips	AP @ IoU	
	0.5	0.7
5	17.1	6.7
10	37.3	16.1
15	41.3	18.2
20	47.1	21.1

890 We conduct experiment to investigate how much the number of training data collected by following nearby agents could
 891 benefit the detector’s final performance. In doing so, we train detectors with four different numbers of training clips, including
 892 5, 10, 15, and 20, and report the overall AP at IoU of 0.5 and 0.7. The result in **Table S2** shows that the performance consistently
 893 improves as the ego car collects more data (pseudo labels) from nearby agents. This again highlights the effectiveness of our
 894 newly explored scenario of learning from nearby agents’ predictions.

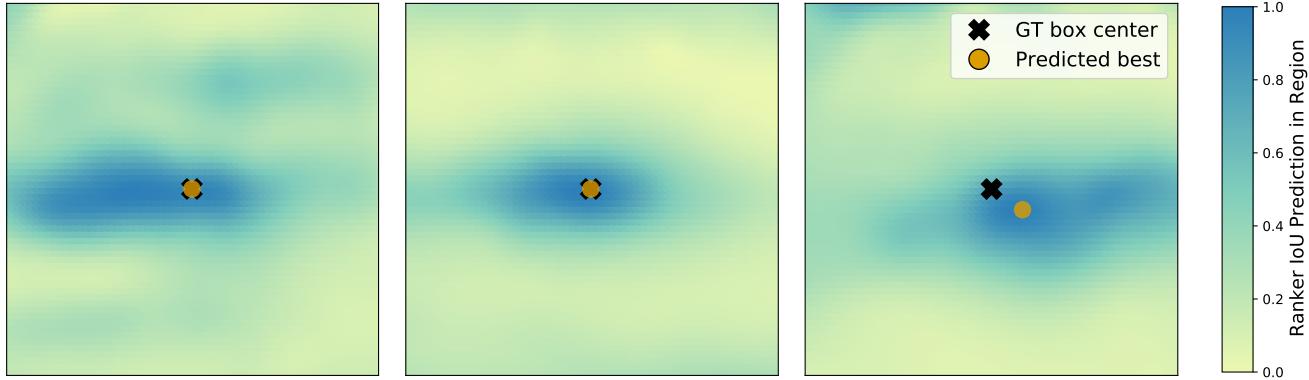


Figure S3. **Ranker IoU prediction behavior.** We visualize the IoU predicted by the ranker on sampled boxes versus the actual ground-truth location. We see that our ranker trained with a handful of annotated frames successfully refines initially mislocalized boxes by giving high scores to samples with more accurate box centers.

Table S3. **Detector performance on the ideal case.** R&B-POP also brings meaningful performance gain in the ideal scenario where the object detector can directly be shared.

method	AP @ IoU 0.7			
	0-30m	30-50m	50-80m	0-80m
sharing detector	56.3	29.0	14.9	40.1
+ R&B-POP	60.9	32.3	17.8	44.4

S3.3. Analysis on sharing detector

Our study introduces a novel learning scenario to build the detector by sharing object box predictions from the reference cars, which is realistic and practical. As shown in Table 1 in the main paper, the pseudo label quality achieved by R&B-POP is as competitive as directly sharing detector, which the scenario faces various constraints (*cf.*, Sec. 1, Sec. 3.1). In this section, to investigate if our method can still improve such cases, we apply our algorithm on top of the object detector shared from the reference car. As shown in Table S3, we observe further performance gain benefit from additional high-quality pseudo labels provided by the reference car in combination with our algorithm. We note that the overall performance is higher than Table 2 in the main paper, as the train and test sets share the same distribution (*i.e.*, clips 1-20 in Fig. 8 in the main paper). This highlights the effectiveness of our study of learning from reference agents’ predictions beyond a single agent.

S3.4. Additional Analysis on ranker

S3.4.1. Visualization of predicted scores

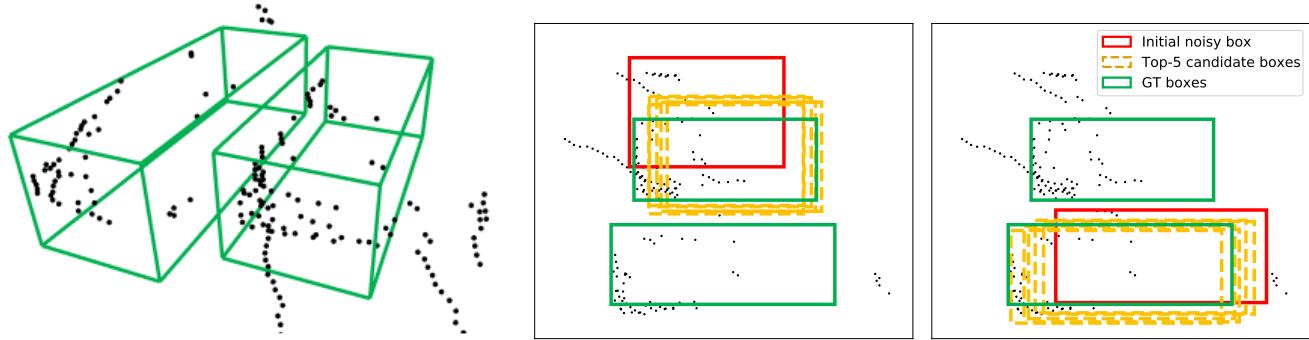
In the ranker training ablation study in our main manuscript, we mention our ranker design performs well with as little as 40 annotated frames. To illustrate its effectiveness further, we visualize the IoU prediction on sampled boxes *vs.* the actual location of the ground-truth as shown in Fig. S3. We observe that IoUs predicted by our ranker are consistent, and boxes with the highest predicted IoUs are close to the ground-truth. Moreover, the result implies that the ranker can effectively remove the false positives, as the region far away from the actual object tends to be given a lower score.

S3.4.2. Ranker on high density of objects

To analyze the behavior of our ranker when objects are close, we carefully select examples where cars are close to each other (*i.e.*, within 2.5 meters of the box centers). We note that while the coarse sampling strategy considers translations of one to two meters, compared to the closest centers of two cars (*i.e.*, 2.5 meters), such translations do not necessarily misassign a box to a nearby car. We see that if the reference car predicts a box for each of the two nearby cars, our method can successfully recall both of them. Fig. S4 demonstrates the ranker’s performance in correctly identifying and selecting the appropriate vehicles in the coarse sampling stage.

Indeed, the purpose of the ranker is to refine the noisy pseudo labels from the reference to the correct location, size, and pose with respect to the ego agent’s view (*cf.*, Sec. 3.2). Therefore, even if the reference only predicts a single box or two

895
896
897
898
899
900
901
902
903904
905
906
907
908
909
910911
912
913
914
915
916
917918
919



(a) Real example of a scenario where two objects with the **same category** are close.
(b) Visualization of the ranker refinement. Given an initial box assigned to a certain object, our ranker correctly identifies and selects the appropriate objects.

Figure S4. Additional experiments on a high density of objects.

predicted boxes are refined to the same car, leaving one false negative, our subsequent curriculum self-training is capable of discovering the remaining object.

S3.4.3. Extension to multi-class

Table S4. Extension of ranker to multi-class refinement.

pseudo label	car	truck
	rec. / prec.	rec. / prec.
initial boxes	56.1 / 71.0	57.2 / 61.0
+ our refinement	60.6 / 76.7	64.2 / 68.4

In the main paper, we focus on a single category that includes various vehicles (*e.g.*, cars, trucks). Still, the pipeline proposed in the main paper is model-agnostic, meaning it can handle both single-class and multi-class detection. The key is to train separate rankers to capture class-specific information (*e.g.*, sizes, shapes) to provide high-quality pseudo labels for the subsequent distance-based curriculum self-training.

Therefore, we explore the multi-class setup by further separating regular cars and trucks in V2V4Real [53] and employing car-specific and truck-specific rankers, respectively. As shown in Table S4, we observe significant improvements in label quality for both classes. Additionally, we conduct a study with selecting cases where nearby objects belong to different categories (*i.e.*, car vs. truck). This is considered challenging because cars and trucks have similar shapes but differ mainly in size. As shown in Fig. S5, we see that the two rankers capture class-specific information and score boxes of different sizes differently, for example, the car ranker gives smaller-size boxes a higher score. We believe such a property would reduce the chance of mistakenly assigning a box of one class to a nearby object of a different class. Moreover, by experimenting with tens of such cases with nearby objects of different classes, we find that the class-specific rankers can correctly maintain class distinctions (*i.e.*, not flipping the classes) with 72.5%, indicating that the rankers effectively capture class-specific information to provide high-quality pseudo labels.

S3.5. Additional Analysis on self-training

S3.5.1. Combining different sets of pseudo labels

Table S5. Anaylsis on pseudo label combination during self-training. [†]distance-based curation: using only R's predictions within its 40m.

using R's pred	distance-based curation [†]	AP @ IoU 0.5			
		0-30m	30-50m	50-80m	0-80m
x (main paper)	-	73.3	43.3	23.3	56.5
o	x	71.8	41.5	26.5	55.1
o	o	74.5	42.0	25.1	57.0

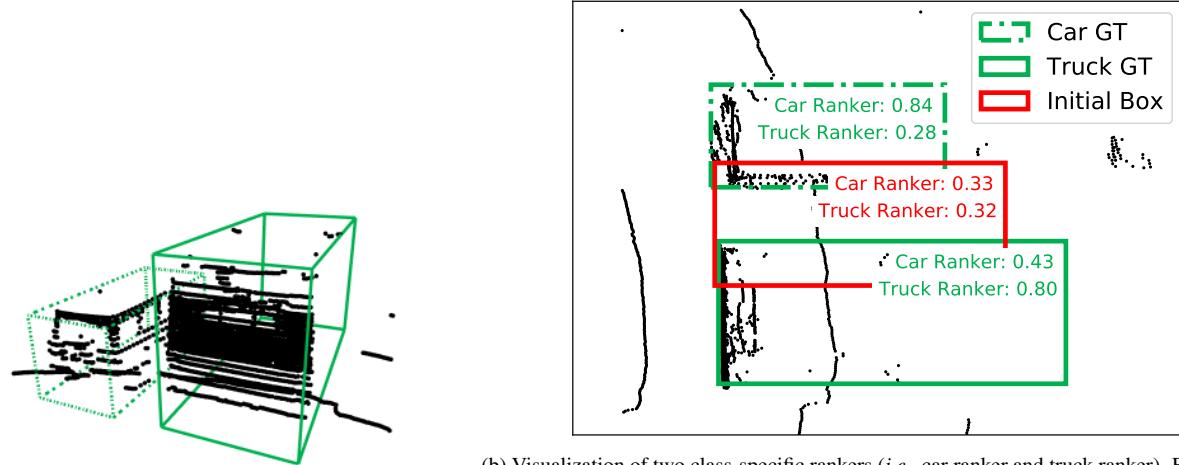


Figure S5. Additional experiments on multi-class for ranker.

939
940
941
942
943
944
945
946
947
948
949

950

In the main paper, we propose to utilize the predicted output of the trained detector, rather than the initially provided predictions from the reference car. In this section, we investigate different combining strategies. First, we naively combined the ego's and reference's predictions in Stage 2. As shown in Table S5, the overall performance (0-80m) dropped from 56.5 (row 1) to 55.1 (row 2). However, we also observe that the performance in the 50-80m range increased from 23.3 to 26.5. We hypothesize that predictions closer to the ego agent are actually farther from the reference, introducing noisier pseudo labels for the ego agent with the naive solution. Conversely, the reference provides more confident predictions for objects closer to it, which are farther from the ego-agent. Based on this assumption, we further explore a simple distance-based curation strategy, combining only predictions within 40m of the reference. As shown in the table, this approach improves the overall performance (0-80m) from 56.5 (row 1) to 57.0 (row 3) and maintained the performance in the 0-30m range (73.3 vs 74.5). These simple experiments demonstrate the potential for many interesting ideas that can be built upon our proposed learning scenario, and we leave it for the future study.

S3.6. Additional Results on Other Dataset

950

Table S6. Additional experimental results on OPV2V dataset [52]. The performance is reported on PointPillars [20] with 64-beam LiDAR. The evaluation metric is AP at IoU 0.5. : uses GT labels.

pseudo label	box refinement	self-training	time delay = 1				time delay = 2			
			0-30m	30-50m	50-80m	0-80m	0-30m	30-50m	50-80m	0-80m
① R 's pred	-	-	84.4	62.7	31.1	71.7	80.9	58.6	22.0	67.3
② R 's GT	-	-	86.7	65.7	33.6	74.2	84.7	65.1	27.8	72.0
③ R 's pred	ranker	-	92.3	68.4	26.1	77.0	91.1	62.2	18.6	73.5
④ R 's pred	-	distance-based curriculum	94.4	74.7	28.2	80.8	94.1	72.8	28.0	79.9
⑤ R 's pred	ranker	distance-based curriculum	96.1	77.4	34.6	83.2	95.3	74.3	31.9	81.4
♣ E 's GT	-	-	97.6	89.4	68.4	90.8	97.6	89.4	68.4	90.8

In the main paper, we conduct experiments on the real-world dataset, V2V4Real [53]. To see the generalizability of R&B-POP, we also evaluate our method on OPV2V [52], a simulation dataset containing 2~7 connected cars within the scene. To suit our study, we re-split the entire 69 clips into 33 and 36 similar to Fig. 8 in the main paper. Also, we only use frames where the distance between the ego car and the reference car is within 90m. To simulate real-world noise, we sample random Gaussian noise with a zero mean and 0.2 standard deviation for localization error and consider a time delay of one and two frames. We set the training epoch to 15, and other hyperparameters remain the same. We use a total of 72 frames for the ranker training, which is two frames per ego car training clip. As shown in Table S6, we see that R&B-POP consistently improves performance on different data and settings, witnessing the general applicability of our method.

951
952
953
954
955
956
957
958

Table S7. Experiments on the number of reference cars. Detection performance improves further as the number of reference cars increases. The evaluation metric is AP at IoU 0.5.

# reference car	time delay = 1				time delay = 2			
	0-30m	30-50m	50-80m	0-80m	0-30m	30-50m	50-80m	0-80m
1	96.1	77.4	34.6	83.2	95.3	74.3	31.9	81.4
2	94.3	78.2	43.5	83.4	96.1	79.1	38.6	84.0

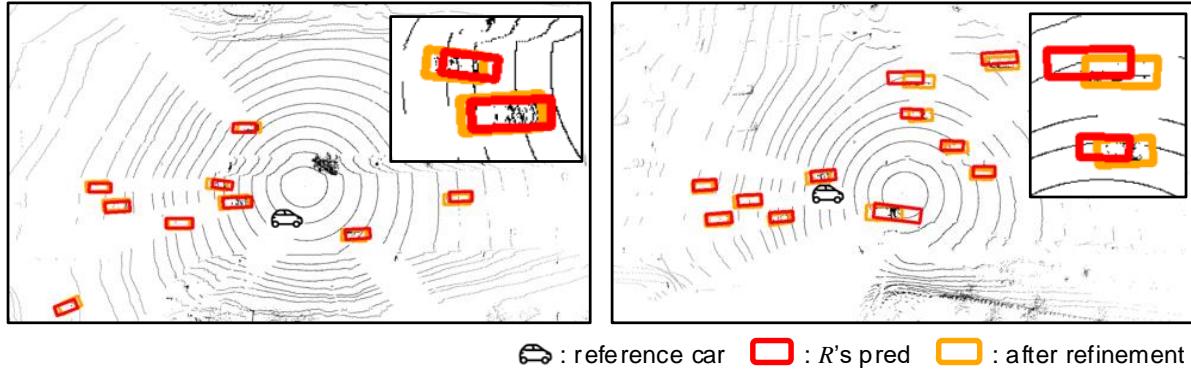


Figure S6. Additional qualitative results of our ranker.

959

S3.6.1. Multiple reference cars

960

Since OPV2V [52] has scenes with more than one reference car, we investigate the relationship between the number of reference cars and detection performance. We use non-maximum suppression with a ranker score to combine two sets of pseudo labels. As shown in Table S7, we see that leveraging the predicted boxes from more reference cars improves final detection performance as different sets of pseudo labels from different views can supplement each other.

961

962

963

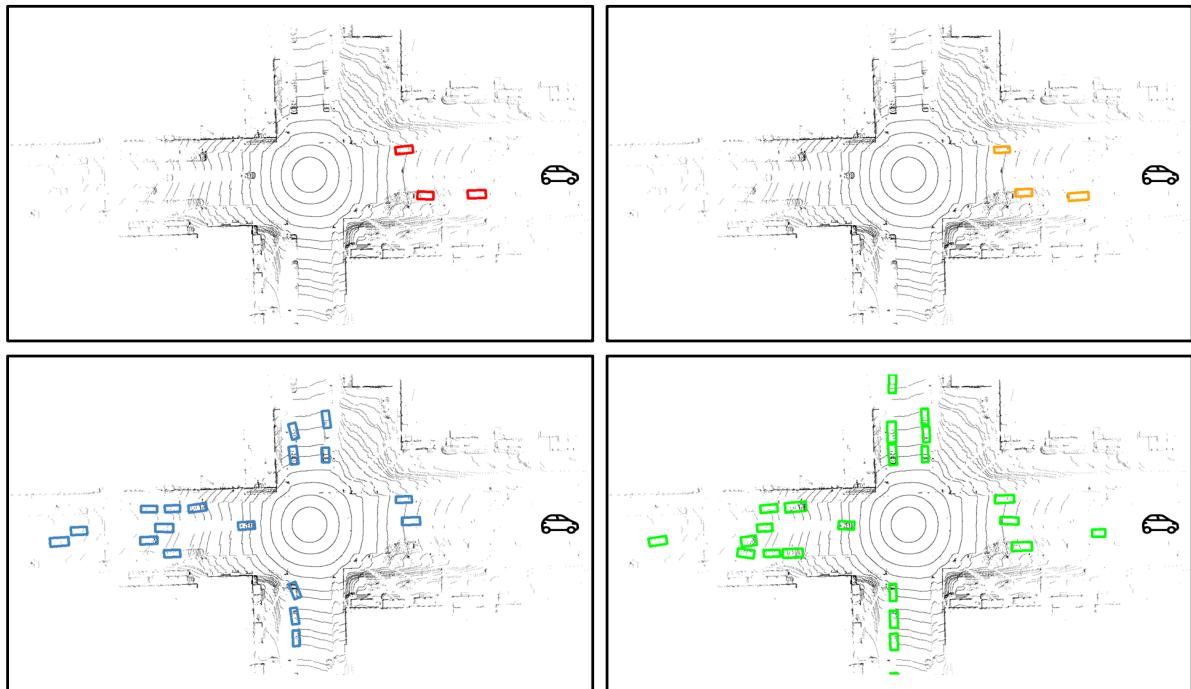
964

S3.7. Additional Qualitative Results

965

966

We provide additional visual results on V2V4Real [53] in Fig. S6 and Fig. S7. Notably, our pipeline improves pseudo label quality by adjusting mislocalization and by discovering and filtering out boxes appropriately.



🚗 : reference car 🟥 : R 's pred 🟨 : after refinement 🟦 : after self-training 🟩 : E 's GT

Figure S7. Additional qualitative results of our overall pipeline.