

249 A CM3D Pseudo-Label Refinement

250 Many components in our CM3D pipeline rely on data-driven priors and can only provide rough 3D
251 estimates. We describe several strategies for improving our 3D psuedo-labels below.

252 **Prompt Engineering.** Although VLMs show impressive zero-shot performance, they struggle when
253 the prompted class is different from concepts encountered in their training data [19]. Following
254 prior work [11], we prompt Detic with the standard nuScenes class names and their synonyms (e.g.
255 {human, adult, person, pedestrian} for class pedestrian, and {car, sedan, SUV} for
256 class car). Specifically, we use the [nuScenes annotator guide](#) to understand how nuScenes defines
257 each class and generate synonyms accordingly. As shown in Fig. 2, Detic predicts class names
258 and 2D bounding boxes for each image, along with confidence scores for each detection. We then
259 perform non-maximum suppression (NMS) to remove redundant predictions across synonyms. In-
260 terestingly, Detic is unable to accurately detect classes like barrier even with carefully designed
261 prompts, suggesting that prompting with synonyms is insufficient for certain ambiguously defined
262 classes [19].

263 **Mask Erosion.** Although instance segmentations from SAM [57] are often accurate, we find that
264 background LiDAR points near object boundaries can significantly impact medoid estimation [67].
265 We employ mask erosion to remove noisy LiDAR points near mask boundaries. These points are
266 often unreliable because of depth discontinuities and minor errors in sensor calibration.

267 **LiDAR Accumulation.** LiDAR sweeps are notoriously sparse at range, making it difficult to distin-
268 guish foreground-vs-background. Therefore, the community has adopted the practice of accumulat-
269 ing multiple ego-motion compensated LiDAR sweeps when training 3D detectors [1]. We adopt the
270 same practice in our pseudo-label generation pipeline for two reasons. First, accumulating multiple
271 sweeps makes our medoid estimate more robust to outliers. Second, it biases the medoid towards
272 the surface of the object, making medoid compensation (discussed next) more reliable.

273 **Medoid Compensation.** We find that predicted medoids are radially biased toward the ego vehicle
274 because LiDAR points are denser on visible surfaces of objects as perceived from the ego vehicle. To
275 compensate for this bias, we “push” all predictions radially away from the ego vehicle by a distance
276 proportional to the object’s size as follows:

277 Let \vec{C} be the medoid of the object in the global coordinate frame, \vec{E} be the center of the ego ve-
278 hicle with respect to the global coordinate frame, and θ be the heading of the object in the global
279 coordinate frame. We define a vector $\vec{CE} = \vec{E} - \vec{C}$, and α as the global slope angle of this
280 vector, i.e., $\alpha = \arctan\left(\frac{\vec{CE}_y}{\vec{CE}_x}\right)$. As shown in Figure 5, we “push” the medoid back by distance
281 $d = \min\left(\left|\frac{w}{2\sin(\alpha-\theta)}\right|, \left|\frac{l}{2\cos(\alpha-\theta)}\right|\right)$. Therefore, our new medoid is $\vec{C}'_x = \vec{C}_x - d \cdot \cos(\alpha)$ and
282 $\vec{C}'_y = \vec{C}_y - d \cdot \sin(\alpha)$. We find that this simple geometric trick works surprisingly well in practice.

283 **Non-Maximum Suppression.** nuScenes uses six RGB cameras to capture a 360° view of the envi-
284 ronment, where neighboring cameras capture overlapping regions. Naively generating pseudo-labels
285 across cameras can produce repeated detections for the same instance. Therefore, we perform non-
286 maximum suppression (NMS) in the overlapping regions [68] after medoid compensation to remove
287 duplicate detections.

288 **Late Fusion.** Recall, we define the *center* of each predicted cuboid to be the medoid of the LiDAR
289 points within an instance mask, the *dimensions* (length, width, height) as reported by ChatGPT
290 when prompted with the class name, and the *orientation* to be aligned with lane geometry provided
291 by an HD map. Therefore, the quality of our pseudo-label generation pipeline is entirely dependent
292 on the accuracy of our shape and orientation priors. In contrast, SAM3D [61] does not use priors
293 for shape and orientation estimation, but rather directly estimates a rotated cuboid from a BEV
294 perspective point cloud. Although SAM3D does not predict semantics, we find that its rotation and
295 shape estimates are often more accurate than our priors. Therefore, we propose a simple late-fusion
296 strategy to combine the best attributes of both zero-shot predictions.

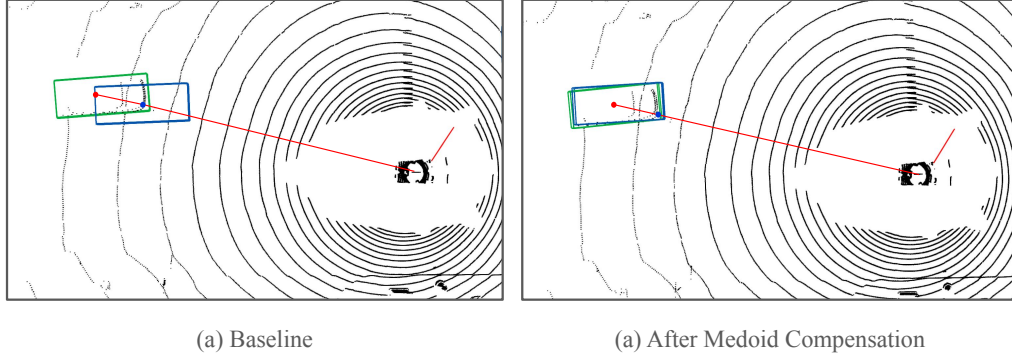


Figure 5: Medoid Compensation. We find that all predicted medoids (shown in blue) tend to be radially biased toward the ego vehicle. This is because the LiDAR pointcloud only captures the visible surface of the car and not its full shape. To compensate for this bias, we “push” all predictions radially away, i.e., along the line connecting the center of the ego vehicle and the object medoid by a distance proportional to the object’s size. The corrected medoid is shown in yellow. Empirically, we show that this geometric trick improves mAP by 1.6% and NDS by 2.1%, respectively.

297 For a given timestep, we greedily match our zero-shot predictions with SAM3D’s predictions using
 298 2D BEV IoU. Spatially matching CM3D and SAM3D predictions yields three categories of
 299 detections: matched detections, unmatched CM3D detections (without corresponding SAM3D de-
 300 tectations), and unmatched SAM3D detections (without corresponding CM3D detections). We dis-
 301 card unmatched SAM3D detections since these are likely false positives because distinguishing
 302 foreground-vs-background with LiDAR-only cues is difficult [69].

303 Fusion of matched predictions from two independent detectors requires their scores to be compa-
 304 rable. Therefore, we use a class-agnostic implementation of score calibration as defined in [70].
 305 Specifically, we scale the logits for SAM3D using a scaling factor τ (obtained by grid search on a
 306 val-set), i.e., confidence value $c = \sigma(\text{logit}/\tau)$. We construct a new set of fused detections by select-
 307 ing the size and orientation from the more confident detection (SAM3D vs. CM3D) after calibration
 308 and use the semantic class predicted by CM3D (since CM3D can more accurately predict semantics
 309 with RGB images). Finally, we add all unmatched CM3D predictions to the set of fused predictions,
 310 unchanged.

311 **Implementation Details.** When generating pseudo-labels with CM3D, we use all Detic predictions
 312 with a confidence greater than 10% and use an IoU threshold of 0.75 for 2D NMS. We use a 3×3
 313 kernel for mask erosion and accumulate the past 3 LiDAR frames for densification. Note that this
 314 is different from the usual 10-frame accumulation in nuScenes since 10-frame LiDAR pointcloud
 315 accumulation creates long “tails” for moving objects, and leads to inaccurate medoid predictions.
 316 For 3D NMS, we use class-specific distance-based thresholds defined in [62].

317 When training all detectors with pseudo-labels, we employ standard augmentation techniques (ex-
 318 cept for copy-paste augmentation, see Appendix C). Following established practices, we aggregate
 319 the past 10 sweeps for LiDAR densification using the provided ego-vehicle poses. We train Center-
 320 Point and BEVFusion using the same hyperparameters prescribed by their respective papers. For
 321 CenterPoint, we first train the detector for 20 epochs with CM3D pseudo-labels and fine-tune the
 322 detector for 20 epochs using the limited set of annotations. For BEVFusion, we first pre-train the
 323 LiDAR-only branch using CM3D pseudo-labels for 20 epochs and fine-tune the LiDAR-only branch
 324 for 20 epochs using the limited set of annotations. We train the fusion model (RGB + LiDAR) for 6
 325 epochs using the limited amount of labeled training data. Lastly, we fine-tune all models using self-
 326 training. Specifically, we use the fine-tuned model to generate new pseudo-labels on the unlabeled
 327 portion of the train-set and retrain the detector on the entire train-set (including the limited set of

Table 2: **Ablation on Pseudo-Label Generation.** We analyze the impact of each component over the baseline (cf. Fig 2). Importantly, we find that prompt engineering, medoid compensation, and non-maximum suppression have the greatest impact.

Method	mAP \uparrow	NDS \uparrow
Baseline	18.6	17.8
+ Prompt Engineering	19.7 (+1.1)	18.4 (+0.6)
+ LiDAR Accumulation	20.0 (+0.3)	18.6 (+0.2)
+ Mask Erosion	20.2 (+0.2)	19.1 (+0.5)
+ Medoid Compensation	21.8 (+1.6)	21.3 (+2.1)
+ Non-Maximal Suppression	22.8 (+1.0)	21.9 (+0.6)
+ Late Fusion	23.0 (+0.2)	22.1 (+0.2)

Table 3: **Self-Training vs. Longer Training Schedule.** We evaluate the impact of training BEV-Fusion + CM3D with a longer-training schedule (2x Schedule, 3x Schedule, and 4x Schedule) and with multiple rounds of self-training (R1, R2, and R3). We note that one round of self training (R1) improves over BEVFusion + CM3D w/ 4x Schedule, suggesting that self-training provides greater benefit than simply training for longer. Further, additional rounds of self-training (R2 and R3) provides modest, but diminishing improvements.

Training Data	Method	mAP \uparrow	NDS \uparrow
5%	SimIPU [8]	39.1	45.8
	PRC [9] + BEVDistill [66]	41.0	47.5
	CALICO [9]	41.7	47.9
	BEVFusion [63] + CM3D w/ 1x Schedule	48.6	47.8
	BEVFusion [63] + CM3D w/ 2x Schedule	49.3	49.5
	BEVFusion [63] + CM3D w/ 3x Schedule	49.9	50.4
	BEVFusion [63] + CM3D w/ 4x Schedule	50.3	51.1
	BEVFusion [63] + CM3D w/ 1x Schedule + R1 Self-Training	50.8	52.2
	BEVFusion [63] + CM3D w/ 1x Schedule + R2 Self-Training	51.1	52.5
	BEVFusion [63] + CM3D w/ 1x Schedule + R3 Self-Training	51.3	52.6

ground truth labels and pseudo-labels) from scratch. We ablate self-training further in Appendix B. We conduct all experiments on 8 RTX 3090 GPUs.

Ablation Study. We ablate our pseudo-label generation algorithm to determine how each component improves the baseline. As discussed above, we find that medoid compensation has the greatest impact on pseudo-label quality, improving mAP by 1.6% and NDS by 2.1%. Tuning the Detic text prompts by including synonyms also improves results significantly. Finally, the 3D distance-based NMS helps remove duplicates present in the overlapping regions of the ring cameras and increases the mAP by 1%.

B Ablation on Self-Training

To further understand the impact of self-training, we investigate two questions: (1) Does self-training provide any improvement over long-training schedules and (2) how does self-training improve NDS?

Self-Training Algorithm. Given K% annotated training data and (1-K%) unlabeled training data with CM3D pseudo-labels, we first pre-train a randomly initialized detector on (1-K%) pseudo-labels, fine-tune on K% annotated data, and use the resulting fine-tuned model to re-label the (1-K%) unlabeled training data. We iterative refine the (1-K%) pseudo-labels through multiple rounds of self-training. Importantly, we randomly initialize the detector after each round of self-training and pseudo-label generation to simplify training. We find that even one round of self-training significantly improves NDS when fine-tuning detectors with limited annotations. Additional rounds of self-training provide limited improvements.

Table 4: **Analysis NDS Sub-Metric Errors.** We find that all sub-component metrics of NDS are improved with three rounds of self-training. Notably, improved mAP, and reduced orientation, attribute, and velocity estimation errors contribute the most to the NDS results. Surprisingly, self-training does not significantly improve size estimation, suggesting that our ChatGPT shape priors are relatively robust.

Metric (%)	BEVFusion [63] + CM3D w/o Self-Training	w/ R3 Self-Training
mAP ↑	48.6	51.3
mATE ↓	33.8	32.2
mASE ↓	26.8	26.2
mAOE ↓	60.7	41.1
mAVE ↓	136.8	100.3
mAAE ↓	43.0	30.2
NDS ↑	47.8	52.6

Self-Training vs. Longer Training Schedules. We compare the performance of BEVFusion w/ CM3D trained with a 1x schedule (20 epochs LiDAR-only branch pre-training with CM3D pseudo-labels, 20 epochs LiDAR-only branch fine-tuning with K% annotated data, and 6 epochs multi-modal fine-tuning with K% annotated data), 2x schedule, 3x schedule, and 4x schedule in Table 3. Although performance improves when training detectors for longer, self-training (even for one round) significantly improves more than longer training schedules.

Self-Training Improves All Components of NDS. Next, we compare the NDS sub-metrics for BEVFusion + CM3D with and without self-training in Table 4. Notably, we find that self-training improves *all* sub-metrics. We posit that self training improves classification accuracy and contributes to better mAP and lower attribute error. Similarly, we posit that self-training reduces pseudo-label bias for orientation and velocity estimation.

C Ablation on Copy-Paste Augmentation

State-of-the-art 3D detectors are often trained with copy-paste augmentation to improve detection accuracy for rare classes like bicycle or construction vehicle. Specifically, rare instances are pasted onto a LiDAR sweep to artificially increase the number of objects. Since our cross-modal 3D detection distillation pipeline creates an explicit 3D bounding box, we can easily apply copy-paste augmentation during pre-training as well. However, given that our pseudo-labels are noisy, is copy-paste augmentation worth it? We train CenterPoint with and without copy-paste augmentation during pre-training and fine-tuning (on 5% of the training data) to ablate its impact. We find that turning augmentation off during pre-training and turning augmentation on during fine-tuning yields the best mAP. Notably, using copy-paste augmentation for both pre-training and fine-tuning performs the worst. Intuitively, copy-pasting noisy pseudo-labels will decrease the signal-to-noise ratio, leading to worse initialization. In contrast, copy-paste augmentation during fine-tuning improves performance in limited-data regimes because we are effectively increasing the size of the dataset.

Table 5: **Ablation on Copy-Paste Augmentation.** We evaluate different permutations of training CenterPoint with and without copy-paste augmentation during pre-training and fine-tuning. Importantly, we find that turning off copy-paste augmentation during pre-training and turning it on during fine-tuning achieves the best performance.

Pre-train w/ Copy-Paste Augmentation	Fine-tune w/ Copy-Paste Augmentation	Modality	mAP ↑	NDS ↑
x	x	L	45.3	45.6
x	✓	L	46.7	46.1
✓	x	L	46.1	46.4
✓	✓	L	44.7	44.5

Table 6: **nuScenes Class Agnostic Evaluation.** We evaluate class-agnostic performance of CM3D pseudo-labels for fair comparison with prior work. We find that our method significantly outperforms LISO-TF by 14.5% AP.

Method	Modality	AP \uparrow	NDS \uparrow	mATE \downarrow	mAOE \downarrow	mASE \downarrow
DBSCAN [25]	L	0.8	10.9	0.980	3.120	0.962
RSF [71]	L	1.9	18.3	0.774	1.003	0.507
Oyster-CP [72]	L	9.1	21.5	0.784	1.514	0.521
Oyster-TF [72]	L	9.3	23.3	0.708	1.564	0.448
LISO-CP [40]	L	10.9	22.4	0.750	1.062	0.409
LISO-TF [40]	L	13.4	27.0	0.628	0.938	0.408
CM3D (Ours)	L + C	27.9	27.6	0.592	0.872	0.428

D nuScenes Class Agnostic Evaluation

In the main paper, we present nuScenes detection results for CM3D pseudo-labels averaged over 10 classes. However, prior pseudo-label generation methods only evaluate class-agnostic performance [40]. For fair comparison with prior work, we evaluate CM3D pseudo-labels without differentiating between classes in Table 6. We find that our approach outperforms prior work by 14.5% AP and 0.6% NDS, highlighting the benefit of using foundational priors and multi-modal inputs. In addition, prior works use motion cues from scene flow to generate object proposals for moving objects [39]. In contrast, our approach localizes both static and moving objects and classifies them using VLMs.

E Replacing Detic with GroundingDINO

We ablate the impact of different VLMs on pseudo-label quality and downstream detection accuracy. Recall, our pseudo-label generation pipeline first prompts a VLM detector with class names (e.g. car, bus, truck) to generate 2D box proposals. We switch out Detic [15] for GroundingDINO [18] in Table 7. First, we find that CM3D w/ Detic generates better pseudo-labels than CM3D w/ GroundingDINO (23.0 mAP vs. 18.0 mAP). Surprisingly, we find that models pre-trained with Detic-based pseudo-labels achieve higher mAP, while models pre-trained with GroundingDINO-based pseudo-labels achieve higher NDS.

Table 7: **Comparison Between VLMs for Pseudo-Label Generation.** CM3D w/ Detic generates better pseudo-labels than CM3D with GroundingDINO (23.0 mAP vs. 18.0 mAP). Surprisingly, we find that models pre-trained with Detic-based pseudo-labels achieve higher mAP, while models pre-trained with GroundingDINO-based pseudo-labels achieve higher NDS.

Training Data	Method	Modality	mAP \uparrow	NDS \uparrow
0%	SAM3D [61]	L	1.7	2.4
	CM3D w/ Detic (Ours)	L + C	23.0	22.1
	CM3D w/ GroundingDINO (Ours)	L + C	18.0	19.7
	CenterPoint [62] + CM3D w/ Detic (Ours)	L	16.7	21.4
	CenterPoint [62] + CM3D w/ GroundingDINO (Ours)	L	16.0	21.2
5%	CenterPoint [62] + Rand. Init.	L	33.1	37.4
	CenterPoint [62] + CM3D w/ Detic (Ours)	L	46.1	46.4
	CenterPoint [62] + CM3D w/ GroundingDINO (Ours)	L	42.9	52.3
10%	CenterPoint [62] + Rand. Init.	L	41.1	48.0
	CenterPoint [62] + CM3D w/ Detic (Ours)	L	50.8	49.2
	CenterPoint [62] + CM3D w/ GroundingDINO (Ours)	L	49.8	56.7

Table 8: **Chat-GPT Shape Prior.** We compare the average 3D shapes of objects in nuScenes (denoted by {Width, Length, Height}) with predicted 3D shapes from ChatGPT. Impressively, ChatGPT provides reasonable 3D object extents.

Class Name	Real Shape Prior	ChatGPT Shape Prior
Car	{1.91, 4.62, 1.68}	{1.80, 4.50, 1.50}
Truck	{2.38, 6.89, 2.60}	{2.60, 8.00, 3.60}
Bus	{2.59, 11.47, 3.81}	{2.50, 12.00, 4.00}
Trailer	{2.29, 10.20, 3.70}	{2.60, 12.00, 3.60}
Construction Vehicle	{2.47, 5.5, 2.38}	{2.00, 4.50, 2.50}
Pedestrian	{0.60, 0.73, 1.76}	{0.40, 0.70, 1.70}
Motorcycle	{0.76, 1.95, 1.57}	{0.80, 2.10, 1.70}
Bicycle	{0.63, 1.82, 1.39}	{0.60, 1.80, 1.40}
Traffic Cone	{0.42, 0.43, 0.70}	{0.30, 0.30, 0.70}
Barrier	{0.60, 2.32, 1.06}	{0.50, 1.20, 0.90}

F Comparing ChatGPT Priors vs. Real Shape Priors

We determine the shape (e.g. width, length, and height) of a class by prompting a large language model (ChatGPT) with the following:

“What are the average sizes in meters (values can be floats) of the following object classes? Give the answer in the form of a JSON file using the following format: [width (side to side), length (front to back), height]
Object classes: car, truck, bus, trailer, construction_vehicle, pedestrian, motorcycle, bicycle, traffic_cone, traffic_barrier.
Do not answer with anything other than the JSON output.”

Despite not having access to our specific 3D training data, LLMs have seen many descriptions of object shapes in their training data and can offer plausible 3D sizes for common objects. We compare shape priors from ChatGPT with anchor boxes derived from the training data in Table 8.

G More Qualitative Results

We present additional qualitative results comparing the predictions from BEVFusion trained from scratch on 5% data (top), BEVFusion + CM3D (middle), and BEVFusion + CM3D w/ Self-Training (bottom). Please see Figure 6 for detailed analysis.

Table 9: **Zero-Shot Waymo 3D Detection.** CM3D marginally improves over SAM3D’s LiDAR-only predictions even though it has access to both RGB and LiDAR. However, pre-training a LiDAR-only detector (CenterPoint) on CM3D pseudo-labels extracted from the train-set significantly improves accuracy, illustrating the benefit of cross-modal learning. Importantly, we note that lower performance for pedestrian and cyclist can be attributed to the stricter matching criteria used by WOD. Specifically, a detection is considered a true positive only if it overlaps with the ground truth with 3D IOU greater than 0.7 for vehicle and 0.5 for other classes.

Method	Test Modality	Vehicle		Pedestrian		Cyclist	
		AP \uparrow	APH \uparrow	AP \uparrow	APH \uparrow	AP \uparrow	APH \uparrow
SAM3D	L	19.1	13.0	0.0	0.0	0.0	0.0
CM3D (Ours)	L + C	19.4	13.4	0.2	0.1	0.7	0.5
CenterPoint + CM3D (Ours)	L	23.7	17.5	0.1	0.1	2.6	1.3

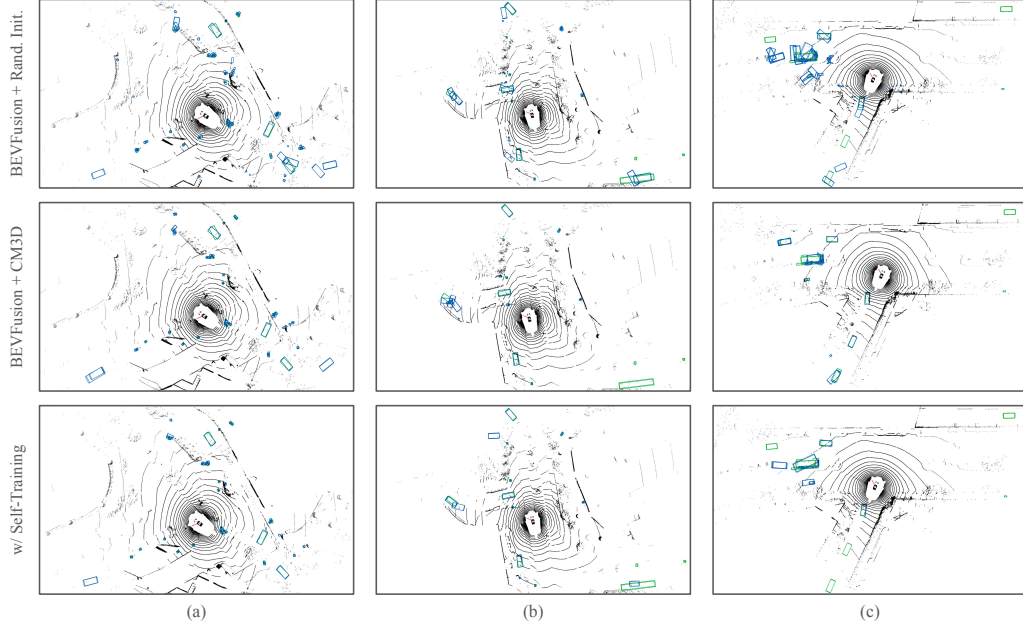


Figure 6: **More Qualitative Results.** We present additional qualitative results comparing the predictions from BEVFusion trained from scratch on 5% data (top), BEVFusion + CM3D (middle), and BEVFusion + CM3D w/ Self-Training (bottom). Ground truth bounding boxes are shown in green, and predictions are shown in blue. Across all three examples, we find that the model trained from scratch produces many high confidence false positives. Pre-training BEVFusion with CM3D pseudo-labels improves performance by reducing the number of false positives. However, many of the predictions have incorrect orientation estimates. Lastly, we find that self-training improves orientation estimation.

Table 10: **Waymo 3D Detection Results.** We report Level-2 mAP and mAPH averaged over 3 classes (Vehicle, Pedestrian, and Cyclist). CM3D consistently improves over random initialization. However, we find that our method performs slightly worse than prior LiDAR-only methods like PRC. Notably, we are unable to train our model for the same number of epochs as prior work due to time constraints, but expect that our method will improve with further training.

Training Data	Method	Modality		mAP \uparrow	mAPH \uparrow
		Train	Test		
0%	SAM3D	L	L	6.4	4.3
	CM3D (Ours)	L + C	L + C	6.8	4.7
	CenterPoint [62] + CM3D (Ours)	L + C	L	8.8	6.2
20%	SECOND [73] + Rand. Init.	L	L	53.1	49.1
	FixMatch [74] (SECOND Backbone)	L	L	55.8	51.5
	ProficientTeachers [75] (SECOND Backbone)	L	L	58.6	54.2
	CenterPoint [62] + Rand. Init.	L	L	63.2	61.0
	PointContrast [6]	L	L	65.2	62.6
	ProposalContrast [7]	L	L	66.3	63.7
	PRC [9]	L	L	68.6	65.5
	CenterPoint [62] + CM3D (Ours)	L + C	L	67.6	64.4

399 H Evaluating CM3D on Waymo

400 We evaluate our method on the Waymo Open Dataset [2] in Table 9. Following prior work, we
 401 report mAP [65] and mAPH [2] (a custom metric proposed by Waymo that incorporates heading).
 402 For fair comparison with SAM3D [61], we only evaluate predictions between 0 and 30 meters
 403 from the ego-vehicle. Notably, SAM3D achieves significantly better performance on Waymo than
 404 nuScenes, likely due to Waymo’s greater point density. Importantly, SAM3D only produces class-

Table 11: **nuScenes BEV Map Segmentation Results.** Although BEVFusion + CM3D is pre-trained on noisy 3D bounding boxes, it performs better on BEV map segmentation than random initialization! However, our method performs worse than other state-of-the-art methods. This suggests that aligning pre-training and fine-tuning tasks improves semi-supervised performance for the target task (cf. Table 1) at the cost of generalizing to other downstream tasks.

Training Data	Method	mIOU \uparrow
5%	BEVFusion [63] + Rand. Init.	36.3
	SimIPU [8]	38.5
	PRC [9] + BEVDistill [66]	40.9
	CALICO [9]	42.0
	BEVFusion [63] + CM3D (Ours)	38.5
10%	BEVFusion [63] + Rand. Init.	43.8
	SimIPU [8]	45.1
	PRC [9] + BEVDistill [66]	46.4
	CALICO [9]	47.3
	BEVFusion [63] + CM3D (Ours)	44.4

Table 12: **Distilling Multi-Modal 3D Psuedo-Labels to RGB-based 3D Detectors.** We train the camera-only branch of BEVFusion with CM3D pseudo-labels. Although distilling CM3D detections into BEVFusion-C performs considerably worse than BEVFusion-L + CM3D and BEVFusion + CM3D, we demonstrate that our shelf-supervised framework allows us to distill multi-modal information from expensive RGB + LiDAR sensors into models trained with cheaper RGB-only sensors.

Training Data	Method	Modality		mAP \uparrow	NDS \uparrow
		Train	Test		
0% (Unsupervised)	CM3D	L + C	L + C	23.0	22.1
	BEVFusion-L [62] + CM3D	L + C	L	16.7	21.4
	BEVFusion-C [63] + CM3D	L + C	C	11.7	16.1
	BEVFusion [63] + CM3D	L + C	L + C	20.6	23.3

agnostic boxes and assigns the class `Vehicle` to all predictions. Therefore, it achieves 0 AP for both `Pedestrian` and `Cyclist`. In contrast, CM3D explicitly predicts semantic classes using RGB images. However, we find that our approach only achieves marginal improvement over SAM3D. We attribute this to Waymo’s evaluation metric and sensor setup.

Unlike nuScenes, which uses a distance-based threshold to match predictions with ground truth boxes when computing mAP, Waymo uses a stricter matching criteria based on 3D IoU with high thresholds of 0.7 for `Vehicle` and 0.5 for other classes. Recall that our size estimates are from ChatGPT and our orientation estimates are derived from an HD map. These imprecise estimates significantly harm detection accuracy since high IoU between predictions and ground truth boxes requires precise size and orientation. Furthermore, the Waymo Open Dataset does not include rear-facing cameras, making it impossible for CM3D to predict bounding boxes behind the ego-vehicle. As a result, we find that our method is heavily dependent on late-fusion with SAM3D, which is able to correct the size and orientation of CM3D pseudo-labels, and generate predictions for parts of the scene without RGB information.

We pre-train CenterPoint with our pseudo-labels for 30 epochs and fine-tune the model for 30 epochs. We find that pre-training with CM3D pseudo-labels consistently improves over random initialization. However, we find that our method performs slightly worse than prior LiDAR-only methods like PRC. We posit that the lack of RGB camera coverage for more than 50% of each LiDAR sweep, and our reliance on SAM3D’s class-agnostic pseudo-labels significantly diminishes the benefit of our proposed approach.

I Limitations and Future Work

We propose a simple cross-modal detection distillation framework that leverages paired multi-modal data and vision-language models to generate zero-shot 3D bounding boxes. We demonstrate that pre-training 3D detectors with our zero-shot 3D bounding boxes yields strong semi-supervised detection accuracy. We discuss several limitations of our approach below.

Limitation: Aligning Pre-Training and Fine-Tuning Task Limits Generalizability. Contrastive learning has been widely adopted for self-supervised learning because it encodes task-agnostic representations that can be used for diverse downstream applications. In contrast, our approach uses prior knowledge about the downstream task to design a suitable pretext task. While this works well when the pre-training and fine-tuning tasks are well aligned, it does not provide a significant improvement when this is not the case. We evaluate the generalization of BEVFusion pre-trained on CM3D pseudo-labels for BEV map segmentation. Surprisingly, our pre-training strategy performs better at BEV map segmentation than random initialization! However, our method performs slightly worse than other state-of-the-art self-supervised methods. This suggests that aligning our pre-training and fine-tuning task can provide significant improvements (cf. Table 1 in the main paper) at the cost of generalizability to diverse tasks. Future work should explore different self-supervised pretext tasks to improve semi-supervised accuracy for diverse tasks in low data settings.

Limitation: Orientation Estimation Requires HD Maps. Our proposed approach uses lane direction from HD maps to estimate vehicle orientation. This heuristic does not work well when vehicles are turning into an intersection, for non-vehicles, and when HD maps are unavailable. Instead, future work should explore using multi-object trackers to generate heading estimates from consecutive detections [40]. We posit that this can improve NDS, and can potentially eliminate the need for self-training.

Limitation: Data Sampling Strategy. Although our semi-supervised experiments follow the suggested protocol in [7, 50, 9] and sample $K\%$ of the training data uniformly from the entire training set, this may be unrealistic in practice. Sampling training data uniformly artificially inflates the diversity of training samples and is more time consuming to annotate than sampling training data from consecutive frames. Future work should explore the semi-supervised setting with data sampled from consecutive frames.

Future Work: Distilling Multi-Modal 3D Pseudo-Labels to RGB-based 3D Detectors. We repurposed our self-supervised framework to learn RGB-only 3D detectors, allowing us to distill multi-modal information from expensive RGB + LiDAR sensors into models trained with cheaper RGB-only sensors. Future work should explore ways of improving cross modal detection distillation of RGB + LiDAR into RGB-only models.

Future Work: Combining Pretext Tasks to Improve Performance. Since our pre-training approach is entirely disjoint contrastive pre-training methods, we hypothesize that our pre-training setup can be used in tandem with such methods to make more accurate predictions and improve results. This may provide a middle-ground between task-agnostic contrastive learning and task-specific pseudo-label pre-training.

References

- [1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nusenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [2] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [3] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.
- [4] K. Vedder, N. Peri, N. Chodosh, I. Khatri, E. Eaton, D. Jayaraman, Y. Liu, D. Ramanan, and J. Hays. Zeroflow: Fast zero label scene flow via distillation. *arXiv preprint arXiv:2305.10424*, 2023.
- [5] W. Chen, A. Edgley, R. Hota, J. Liu, E. Schwartz, A. Yizar, N. Peri, and J. Purtle. Rebound: An open-source 3d bounding box annotation tool for active learning. *arXiv preprint arXiv:2303.06250*, 2023.
- [6] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.
- [7] J. Yin, D. Zhou, L. Zhang, J. Fang, C.-Z. Xu, J. Shen, and W. Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022.
- [8] Z. Li, Z. Chen, A. Li, L. Fang, Q. Jiang, X. Liu, J. Jiang, B. Zhou, and H. Zhao. Simipu: Simple 2d image and 3d point cloud unsupervised pre-training for spatial-aware visual representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1500–1508, 2022.
- [9] J. Sun, H. Zheng, Q. Zhang, A. Prakash, Z. M. Mao, and C. Xiao. Calico: Self-supervised camera-lidar contrastive pre-training for bev perception. *arXiv preprint arXiv:2306.00349*, 2023.
- [10] B. Wilson, Z. Kira, and J. Hays. 3d for free: Crossmodal transfer learning using hd maps. *arXiv preprint arXiv:2008.10592*, 2020.
- [11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [12] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284, 2020.
- [13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

- [14] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021.
- [15] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.
- [16] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.
- [17] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.
- [18] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [19] A. Madan, N. Peri, S. Kong, and D. Ramanan. Revisiting few-shot object detection with vision-language models. *arXiv preprint arXiv:2312.14494*, 2023.
- [20] A. Ošep, T. Meinhardt, F. Ferroni, N. Peri, D. Ramanan, and L. Leal-Taixé. Better call sal: Towards learning to segment anything in lidar. *arXiv preprint arXiv:2403.13129*, 2024.
- [21] M. Najibi, J. Ji, Y. Zhou, C. R. Qi, X. Yan, S. Ettinger, and D. Anguelov. Unsupervised 3d perception with 2d vision-language distillation for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8602–8612, 2023.
- [22] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu. Segment any point cloud sequences by distilling vision foundation models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [23] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3d lidar scans. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 4508–4513. IEEE, 2016.
- [24] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying unknown instances for autonomous driving. In *Conference on Robot Learning*, pages 384–393. PMLR, 2020.
- [25] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017.
- [26] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu. Open-set 3d object detection. In *2021 International Conference on 3D Vision (3DV)*, pages 869–878. IEEE, 2021.
- [27] H. Tian, Y. Chen, J. Dai, Z. Zhang, and X. Zhu. Unsupervised object detection with lidar clues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5962–5972, 2021.
- [28] A. Collet, S. S. Srinivasay, and M. Hebert. Structure discovery in multi-modal data: a region-based approach. In *2011 IEEE International Conference on Robotics and Automation*, pages 5695–5702. IEEE, 2011.
- [29] G. M. García, E. Potapova, T. Werner, M. Zillich, M. Vincze, and S. Frintrop. Saliency-based object discovery on rgb-d data with a late-fusion approach. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1866–1873. IEEE, 2015.
- [30] J. Shin, R. Triebel, and R. Siegwart. Unsupervised discovery of repetitive objects. In *2010 IEEE International Conference on Robotics and Automation*, pages 5041–5046. IEEE, 2010.

- [31] L. Ma and G. Sibley. Unsupervised dense object discovery, detection, tracking and reconstruction. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part II 13*, pages 80–95. Springer, 2014.
- [32] L. Ma, M. Ghafarianzadeh, D. Coleman, N. Correll, and G. Sibley. Simultaneous localization, mapping, and manipulation for unsupervised object discovery. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1344–1351. IEEE, 2015.
- [33] D. Kochanov, A. Ošep, J. Stückler, and B. Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1785–1792. IEEE, 2016.
- [34] S. Choudhary, A. J. Trevor, H. I. Christensen, and F. Dellaert. Slam with object discovery, modeling and mapping. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1018–1025. IEEE, 2014.
- [35] Y. You, K. Luo, C. P. Phoo, W.-L. Chao, W. Sun, B. Hariharan, M. Campbell, and K. Q. Weinberger. Learning to detect mobile objects from lidar scans without labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1130–1140, 2022.
- [36] E. Herbst, X. Ren, and D. Fox. Rgb-d object discovery via multi-scene analysis. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4850–4856. IEEE, 2011.
- [37] E. Herbst, P. Henry, X. Ren, and D. Fox. Toward object discovery and modeling via 3-d scene comparison. In *2011 IEEE international conference on robotics and automation*, pages 2623–2629. IEEE, 2011.
- [38] M. Najibi, J. Ji, Y. Zhou, C. R. Qi, X. Yan, S. Ettinger, and D. Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *European Conference on Computer Vision*, pages 424–443. Springer, 2022.
- [39] J. Seidenschwarz, A. Ošep, F. Ferroni, S. Lucey, and L. Leal-Taixé. Semoli: What moves together belongs together. *arXiv preprint arXiv:2402.19463*, 2024.
- [40] S. Baur, F. Moosmann, and A. Geiger. Liso: Lidar-only self-supervised 3d object detection. *arXiv preprint arXiv:2403.07071*, 2024.
- [41] A. Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 626–642. Springer, 2020.
- [42] J. Sauder and B. Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019.
- [43] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028. IEEE, 2020.
- [44] J. Hou, B. Graham, M. Nießner, and S. Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021.
- [45] Z. Zhang, Y. Dong, Y. Liu, and L. Yi. Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17661–17670, 2023.

- [46] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.
- [47] J. Hou, S. Xie, B. Graham, A. Dai, and M. Nießner. Pri3d: Can 3d priors help 2d representation learning? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5693–5702, 2021.
- [48] S. Lal, M. Prabhudesai, I. Mediratta, A. W. Harley, and K. Fragkiadaki. Coconets: Continuous contrastive 3d scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12487–12496, 2021.
- [49] L. Li and M. Heizmann. A closer look at invariances in self-supervised pre-training for 3d vision. In *European Conference on Computer Vision*, pages 656–673. Springer, 2022.
- [50] X. Tian, H. Ran, Y. Wang, and H. Zhao. Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13570–13580, 2023.
- [51] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [52] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [53] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021.
- [54] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022.
- [55] A. Mahmoud, J. S. Hu, T. Kuai, A. Harakeh, L. Paull, and S. L. Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023.
- [56] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu. Segment any point cloud sequences by distilling vision foundation models. *arXiv preprint arXiv:2306.09347*, 2023.
- [57] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [58] J. Cen, Z. Zhou, J. Fang, W. Shen, L. Xie, D. Jiang, X. Zhang, Q. Tian, et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36, 2024.
- [59] Q. Shen, X. Yang, and X. Wang. Anything-3d: Towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*, 2023.
- [60] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023.
- [61] D. Zhang, D. Liang, H. Yang, Z. Zou, X. Ye, Z. Liu, and X. Bai. Sam3d: Zero-shot 3d object detection via segment anything model. *arXiv preprint arXiv:2306.02245*, 2023.
- [62] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

- [63] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [65] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [66] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao. Bevdistill: Cross-modal bev distillation for multi-view 3d object detection. *arXiv preprint arXiv:2211.09386*, 2022.
- [67] T. Yin, X. Zhou, and P. Krähenbühl. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34:16494–16507, 2021.
- [68] T. Wang, X. Zhu, J. Pang, and D. Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.
- [69] N. Peri, A. Dave, D. Ramanan, and S. Kong. Towards long-tailed 3d detection. In *Conference on Robot Learning*, pages 1904–1915. PMLR, 2023.
- [70] Y. Ma, N. Peri, S. Wei, W. Hua, D. Ramanan, Y. Li, and S. Kong. Long-tailed 3d detection via 2d late fusion. *arXiv preprint arXiv:2312.10986*, 2023.
- [71] D. Deng and A. Zakhor. Rsf: Optimizing rigid scene flow from 3d point clouds without labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1277–1286, January 2023.
- [72] L. Zhang, A. J. Yang, Y. Xiong, S. Casas, B. Yang, M. Ren, and R. Urtasun. Towards unsupervised object detection from lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9317–9328, 2023.
- [73] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. ISSN 1424-8220. doi:10.3390/s18103337. URL <https://www.mdpi.com/1424-8220/18/10/3337>.
- [74] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.
- [75] J. Yin, J. Fang, D. Zhou, L. Zhang, C.-Z. Xu, J. Shen, and W. Wang. Semi-supervised 3d object detection with proficient teachers. In *European Conference on Computer Vision*, pages 727–743. Springer, 2022.