VESPA: Open-World Auto-Labeling for 3D Object Detection in Autonomous Driving

Supplementary Material

7. Detector details

We use OpenPCDet [23] to train CenterPoint detectors [27] on pseudo labels from the training set, adopting default settings of OpenPCDet: a voxel size of 0.075×0.075 m, batch size of 4, and 20 training epochs. Class-balanced grouping and sampling (CBGS) [34] is enabled for the 3-class and 8-class experiments to mitigate class imbalance; the 1-class setup is trained without class balancing.

8. Execution times

VESPA is composed of several steps, from which the processing speeds for a NuScenes scene are here detailed:

- Removing ground points: 128.0s
- Generating segmentation masks: 333.4s
- Mask to LiDAR reprojection: 1.14s
- Cluster denoising: 1.6s
- Appearance matching correction: 59.8s
- Merging multicamera objects: 2.7s
- Box inflation: 0.0005s
- Total: 526.6s

The processing time for the whole NuScenes dataset is around 24 hours. More importantly, the greatest improvement of the module, caused by the correction steps, takes only around 170 minutes for the whole dataset. Both auto-labeling and training where executed in a Nvidia A40.

9. Modular examples

In the following subsections we present qualitative examples of the composing submodules of VESPA.

9.1. Ground Removal

In Fig. 3 we present an example of RANSAC-based ground removal with the following parameters:

```
acc_M: 5
ground_filter_xyradius_threshold: 40
ground_filter_zmin_threshold: -1
ground_filter_zmax_threshold: +1
global_ransac_min_sample_points: 250
global_ransac_residual_threshold: 0.1
global_ransac_max_trials: 20
cone_ransac_dmax_threshold: 0.30
cone_ransac_num_cones: 8
cone_ransac_min_cone_points: 500
cone_ransac_min_sample_points: 250
cone_ransac_residual_threshold: 0.05
cone_ransac_max_trials: 20
cone_ransac_max_height: 0.15
rangefilter_sky_threshold: 4
rangefilter_range_threshold: 55
```

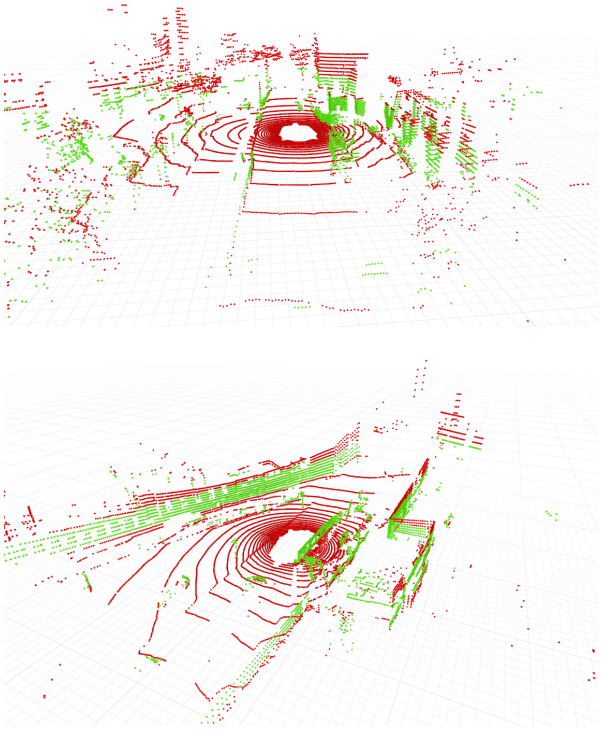

Figure 3. Examples of ground removal

9.2. Zero-shot Object Segmentation

Results for the object segmentation module are shown in Fig. 4, which were obtained with the following parameters:

```
class_names: ["car", "truck", "bus",
"trailer", "excavator", "pedestrian",
"motorcycle", "bicycle"]
detector: "IDEA-Research/
grounding-dino-base"
det_threshold: 0.3
segmentator: "facebook/sam-vit-base"
nms_threshold: 0.5
refine_masks: true
```

Figure 4. Examples of Zero-shot Object segmentation

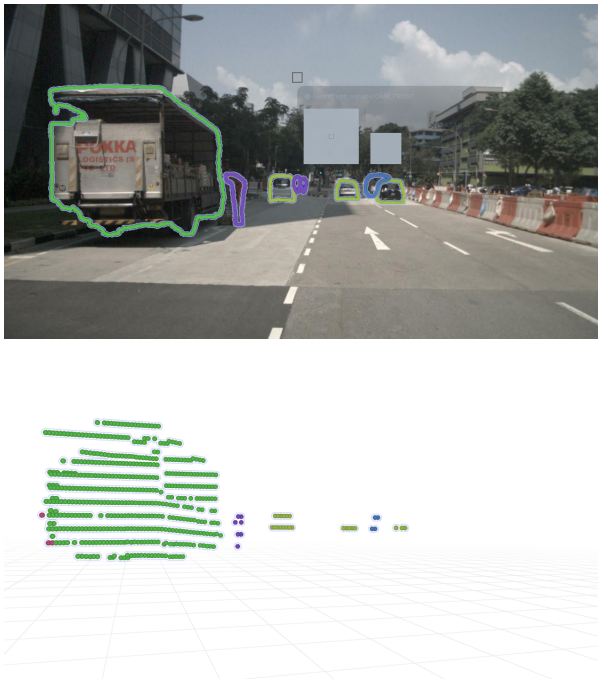

Figure 5. Examples of Object reprojection

9.3. LiDAR reprojection

Examples of object reprojection are shown in Fig. 5 together with the used parameters:

```
erosion_kernel_size: 5
erosion_iterations: 1
filter_bg_lidar: true
filter_bg_segment_ioa_thresh: 0.5
```

9.4. Cluster denoising

Examples with and without cluster denoising are shown in Fig. 6. This module is critical for the performance, because the smallest calibration or synchronisation error could lead to a deviation of several meters of the projected boxes, with the corresponding decrease in AP.

9.5. Multi-camera object merging

For the nuscenes dataset, there are 6 cameras, meaning there are 4 corners where object could appear in 2 different cameras at the same time, therefore, the merging module help improve the metrics consistently. The match is done if the candidates share more than 1% of the points. We present an example on Fig. 7.

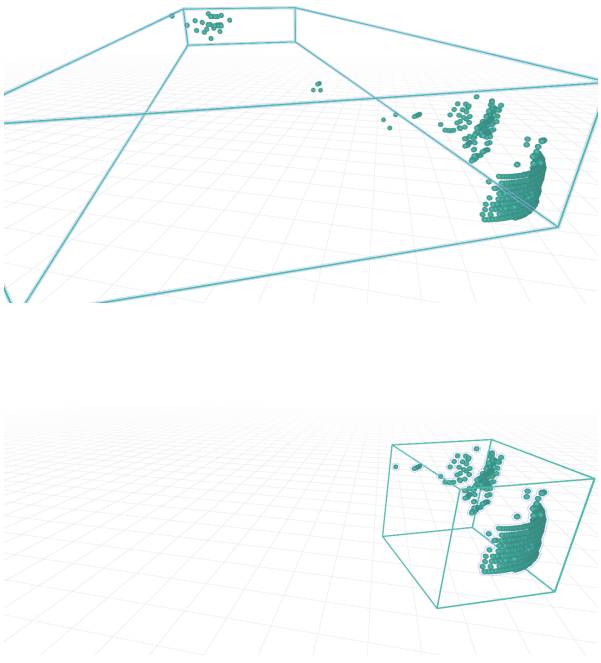

Figure 6. Examples of projection denoising

```
camera_pair_side_threshold: 0.15
lidar_common_threshold: 0.01
compatible_class_groups:
  - ["car", "truck", "bus"]
  - ["truck", "trailer"]
  - ["motorcycle", "bicycle"]
enable_camera_mask_based: true
enable_lidar_point_based: true
```

9.6. Object matching

The tracking module is principally used to correct the heading of the detected boxes, by assuming the heading is equal to the velocity direction. The results are seen in Fig. 8.

```
compatible_class_groups:
    - ["car", "truck", "bus"]
    - ["truck", "trailer"]
    - ["motorcycle", "bicycle"]
  icp_search_size: 20.00 # meters
  icp_search_step: 0.5 # meters
  icp_max_iterations: 50
  max_dist_inlier_threshold: 0.30 # meters
  bbox_iou_threshold: 0.1
  use_icp: true
  icp_2d: true
  filter_candidates_by_speed: true
  velocity_diff_threshold: null
```

9.7. Object inflation

Finally, the goal is to deal with occlusion and ambiguity from the LiDAR point cloud, by assuming prior for the dimensions of the objects, to loose dependency on the seen points. Results are seen in Fig. 9.

10. Implementation Details

To ensure full reproducibility, we provide the key hyperparameters and the exact prompts used to query the Large Language Model (LLM).

10.1. Hyperparameters

The following hyperparameters were used for all experiments on the nuScenes dataset.

- **Ground Removal (Sec. 3.1):**
  - Number of temporal sweeps: 5
  - Ground plane distance threshold: 0.3 m
  - Maximum point range filter: 40 m
  - Maximum point height filter: 4 m
- **Zero-Shot Segmentation (Sec. 3.2):**
  - Grounding DINO confidence threshold: 0.3
  - Mask-based NMS (IoA) threshold: 0.5
- **Multi-camera Object Merging (Sec. 3.5):**
  - Image edge region for adjacency cue: 15% of image width
- **Appearance-Based Tracking (Sec. 3.7):**
  - Motion threshold (moving vs. static): 0.5 m/s

10.2. LLM Prompts

We used the GPT-4 API for all LLM queries. The following prompts were used to obtain dataset-agnostic priors.

For Cluster Denoising (Sec. 3.4): The prompt was structured to get the average width for each object class to use as the DBSCAN distance threshold.

```
What is the typical width in meters
for the following objects in an urban
driving environment? Provide
the answer as a simple list.
- Car
- Pedestrian
- Bicycle
- ... [etc. for all classes]
```

For Bounding Box Inflation (Sec. 3.8): The prompt was designed to retrieve mean dimensions for robustifying box sizes.

```
What are the average dimensions
(length, width, height) in meters for a
typical [CLASS NAME]? For example, for a car.
```

---

11. VESPA Pipeline Algorithm

Algorithm 1 provides a high-level overview of the complete VESPA pipeline, from raw sensor inputs to the final set of 3D pseudo-labels.

12. Complete Ablation

In Table 6 we present the complete ablation studies including TP-Metrics and per class AP results.

13. TruckScenes

In Table 7 we present the results on the original TruckScenes evaluation configuration, where the GT labels are present up to 150m. The results are clearly degradated in comparison to the 50m range, and can be explained through two reasons. First, the VLMs struggle with small objects, or objects which seem small. Second, the farther away the objects are, the more critical small calibration and synchronization issues become. It is possible that for highways scenarios not even our correction modules can clean all the noise introduced by wrong calibrations or synchronisations.

14. AnonymousScenes

In Table 8, we report the pseudolabeling performance of VESPA evaluated on the *AnonymousScenes* dataset. Overall, the metrics are noticeably lower compared to those

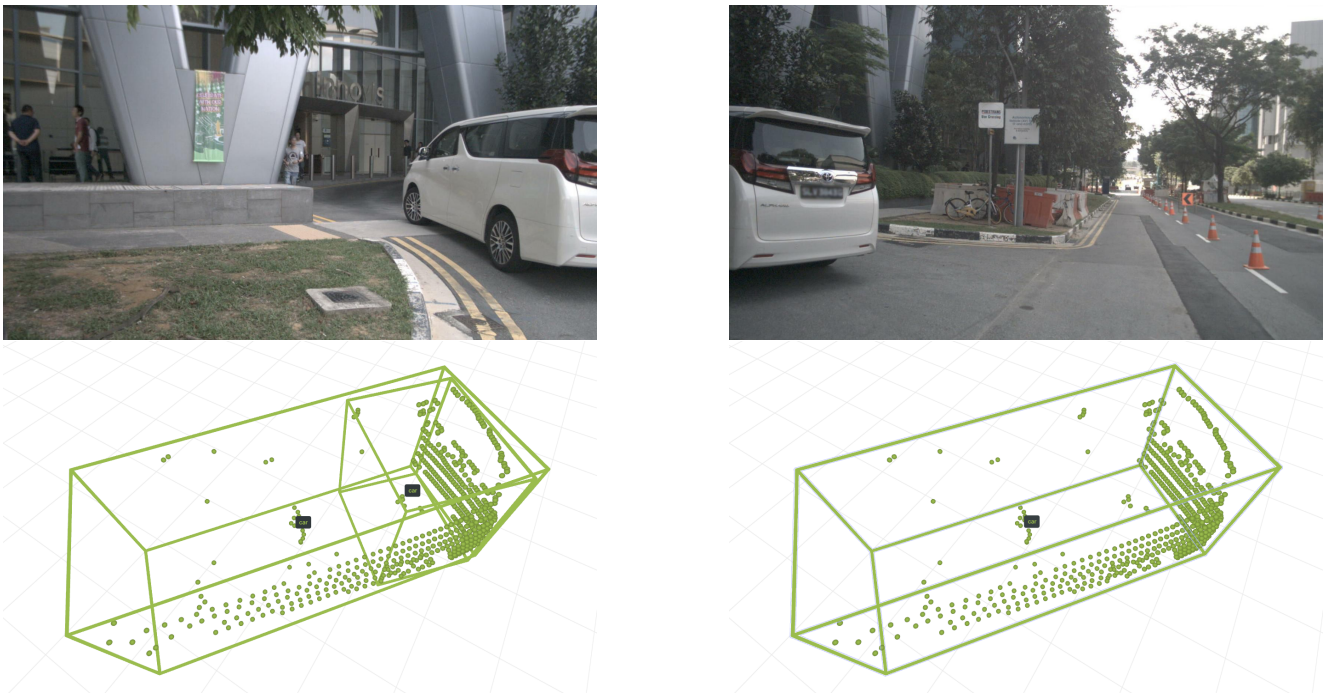

Figure 7. Examples of Multi-camera merging

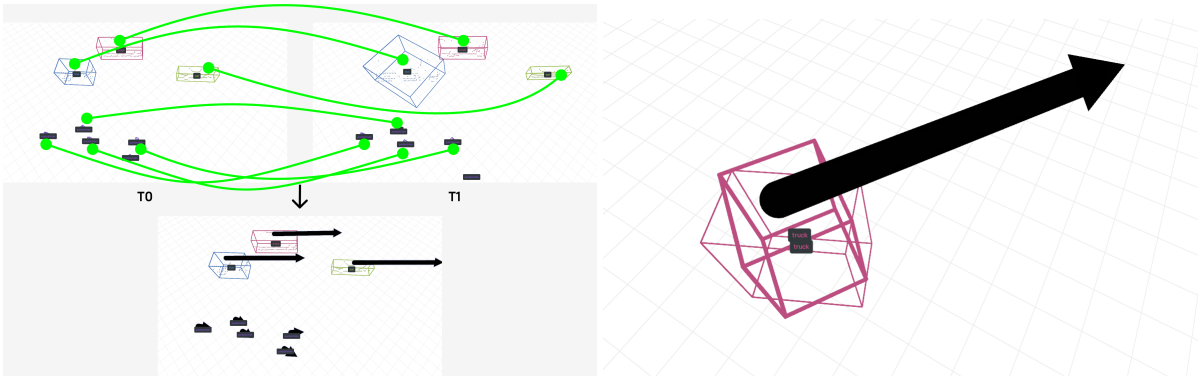

Figure 8. Example of heading correction through object matching

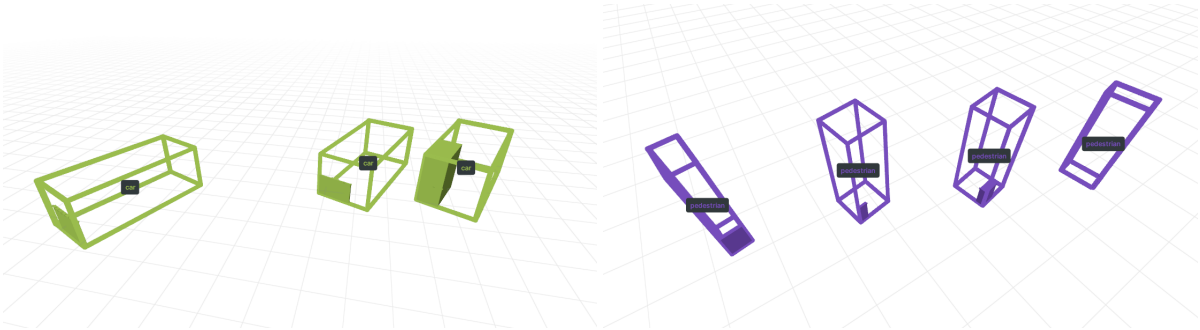

Figure 9. Example of box inflation

Table 6. Ablation of 3-class object detection with VESPA on the nuScenes validation set after training CenterPoint. First 4 columns represent the modules: DN=Cluster denoising; MCM=Multi-camera object merger; AMC=Appearance-matching correction; BI=Bounding box inflation

DN	MCM	AMC	BI	mAP↑	NDS↑	ATE↓	ASE↓	AOE↓	AVE↓	AP_{car} ↑	$AP_{pedestrian}$ ↑	$AP_{cyclist}$ ↑
✓	✓	✓	✓	**46.76**	**43.47**	**0.384**	**0.363**	**0.877**	**0.365**	**48.9**	**75.2**	16.1
×	✓	✓	✓	27.86	26.72	0.593	0.442	0.982	0.702	23.8	46.4	13.4
✓	×	✓	✓	45.50	42.44	0.396	0.364	0.887	0.382	45.6	**75.2**	15.7
✓	✓	×	✓	25.48	21.84	0.658	0.431	1.483	1.720	18.7	45.6	12.2
✓	✓	✓	×	40.68	35.71	0.400	0.766	0.902	0.393	30.9	73.8	17.3
✓	✓	×	×	40.35	28.48	0.393	0.776	1.473	1.650	30.4	72.6	**18.1**
✓	×	×	×	38.69	27.20	0.422	0.792	1.459	1.630	25.9	73.1	17.2
×	×	×	×	23.55	16.93	0.655	0.828	1.467	1.716	14.4	43.7	12.5

Algorithm 1 The VESPA Pipeline

Input: LiDAR point clouds P_t, Camera images I_t, for timestamps $t \in \{1, ..., T\}$
Output: 3D pseudo-labels L_t for each frame
for each frame t **do**
  $P'_t \leftarrow$ AggregateTemporalSweeps$(P_t, P_{t-1}, ...)$
  $P_{nonground} \leftarrow$ GroundRemoval(P'_t)
  $D_{2D} \leftarrow \{\}$ ▷ Initialize 2D detections
  for each camera image $I_{t,c}$ **do**
    $B_{2D} \leftarrow$ ZeroShotObjectDetection$(I_{t,c})$
    $M_{2D} \leftarrow$ GenerateSegmentationMasks$(B_{2D}, I_{t,c})$
    $M'_{2D} \leftarrow$ MaskNMS(M_{2D})
    $D_{2D} \leftarrow D_{2D} \cup M'_{2D}$
  end for
  $C_{3D} \leftarrow$ LiDARReprojection$(P_{nonground}, D_{2D})$
  $C'_{3D} \leftarrow$ ClusterDenoising(C_{3D})
  $O_{3D} \leftarrow$ FitOrientedBoundingBoxes(C'_{3D})
  $O'_{3D} \leftarrow$ MultiCameraMerge(O_{3D})
  $E_t \leftarrow$ ExtractAppearanceEmbeddings(O'_{3D})
  if $t > 1$ **then**
    $Matches \leftarrow$ TrackObjects$(O'_t, E_t, O'_{t-1}, E_{t-1})$
    $Velocities \leftarrow$ EstimateMotionICP$(Matches)$
    $O''_{3D} \leftarrow$ RefineBoxOrientation$(O'_{3D}, Velocities)$
  end if
  $L_t \leftarrow$ BoundingBoxInflation(O''_{3D})
end for

obtained on nuScenes. To contextualize this performance drop, it is important to consider the differences between the two datasets.

AnonymousScenes employs a denser LiDAR sensor with 64 channels, in contrast to the 32-channel sensor used in nuScenes. However, all surrounding cameras in AnonymousScenes—except for the front-facing one—have a significantly lower resolution of 380×240 pixels. The results

Table 7. Comparison of VESPA and UNION on the TruckScenes dataset across 3-class settings with detection range 150m.

Metric	**Pseudo Labels**		**Centerpoint**	
	UNION	**VESPA**	**UNION**	**VESPA**
1-class results				
AP	1.03	**9.93**	11.19	**18.12**
ATE	0.8615	**0.8165**	0.6230	**0.5944**
ASE	0.5751	**0.3391**	0.4324	**0.2706**
AOE	1.1589	**0.6565**	0.6213	**0.5256**
AVE	9.7372	**7.9889**	6.3246	**3.8235**
NDS	6.16	**16.85**	16.43	**25.15**
3-class results				
mAP	0.18	**6.81**	3.41	**16.91**
mATE	0.9453	**0.5279**	0.7612	**0.4067**
mASE	0.8534	**0.4549**	0.6643	**0.4191**
mAOE	1.0426	**0.7737**	0.7983	**0.6244**
mAVE	4.3068	**3.3378**	4.4311	**2.3879**
NDS	2.10	**15.84**	9.47	**23.96**
AP				
bicycle	0.00	**0.2**	0.000	**4.60**
pedestrian	0.00	**10.10**	42.87	**27.5**
vehicle	0.5	**10.00**	38.71	**18.7**

suggest that VESPA is relatively robust to variations in LiDAR configuration, but is highly sensitive to image resolution. In particular, low-resolution imagery appears to substantially degrade the quality of the VLM-based detections, likely due to the inability to resolve small or fine-grained objects. This highlights a limitation of current VLMs when applied to low-resolution inputs and emphasizes the importance of high-quality visual data for the success of the VESPA pipeline.

Further details of the AnonymousScenes dataset used to

Table 8. Evaluation of the pseudo labels in AnonymousScenes

Metric	AnonymousScenes
mAP	4.72
mATE	0.8013
mASE	0.5487
mAOE	1.4734
mAVE	5.9457
NDS	8.86
Per-Class AP	
bicycle	1.46
bus	5.86
car	15.14
construction vehicle	0.000
motorcycle	2.11
pedestrian	10.81
trailer	0.000
truck	2.40

Class	Total Count
Car	54837
Pedestrian	40894
Bicycle	14879
Barrier	2512
Scooter	2413
Motorcycle	1679
Truck	1378
Trailer	403
Tram	350
Bus	335
Traffic Cone	27

Table 9. Class distribution across AnonymousScenes.

further evaluate VESPA are presented in Tab. 9. On Fig. 10 we present to examples of camera images of AnonymousScenes:

15. Qualitative examples

In this section, we qualitatively compare the pseudo-labels generated by UNION [13], the current state-of-the-art open-source method, with those from our proposed approach, Vision-language Enabled Shelf-supervised Pseudo-label Assignment (VESPA). We examine a set of randomly selected frames and analyze the 3D bounding boxes projected to the bird's eye view, green being ground truth and blue predicted boxes. Our focus is on differences in detection quality, particularly the handling of small and distant objects, as well as bounding box accuracy (including size and orientation) and the rate of false positives.

Across all examined scenes, VESPA consistently achieves higher recall, particularly in the detection of small and distant objects. Its bounding box estimates are often more precise in terms of size and orientation. However, it also introduces more false positives and occasional orientation errors. In contrast, UNION exhibits higher precision but often fails to capture smaller objects and struggles with box alignment. These results highlight VESPA's superior capability in dense or more complex environments with more difficult object classes.

Figure 10. Camera examples of AnonymousScenes

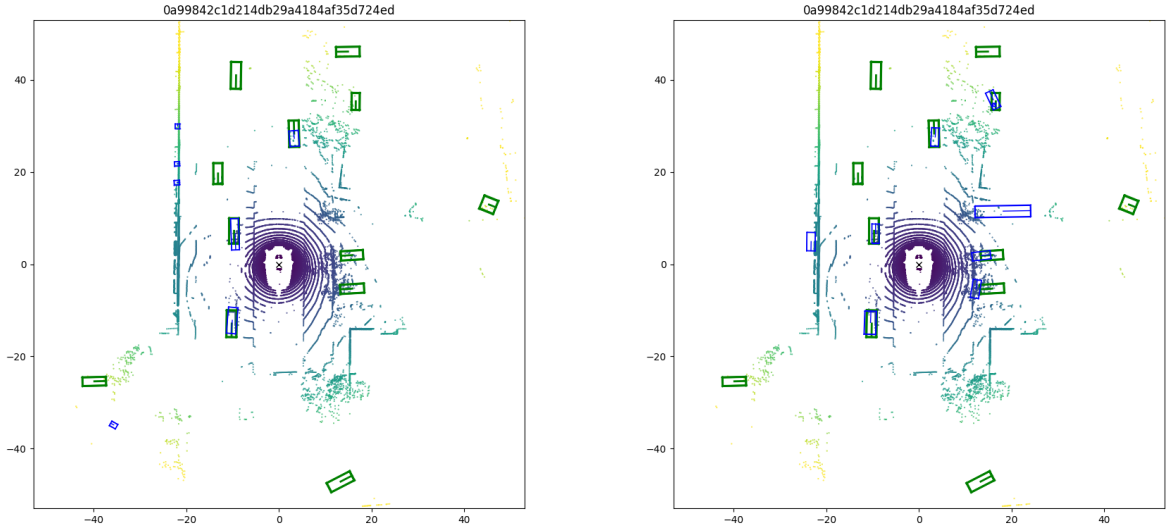

Figure 11. UNION (Left) produces a few false positives corresponding to smaller objects and does not estimate the sizes of the true positives as accurately. VESPA (Right) also introduces two false positives, this time corresponding to larger objects, but provides more complete and correctly sized detections of the dominant objects in the scene.

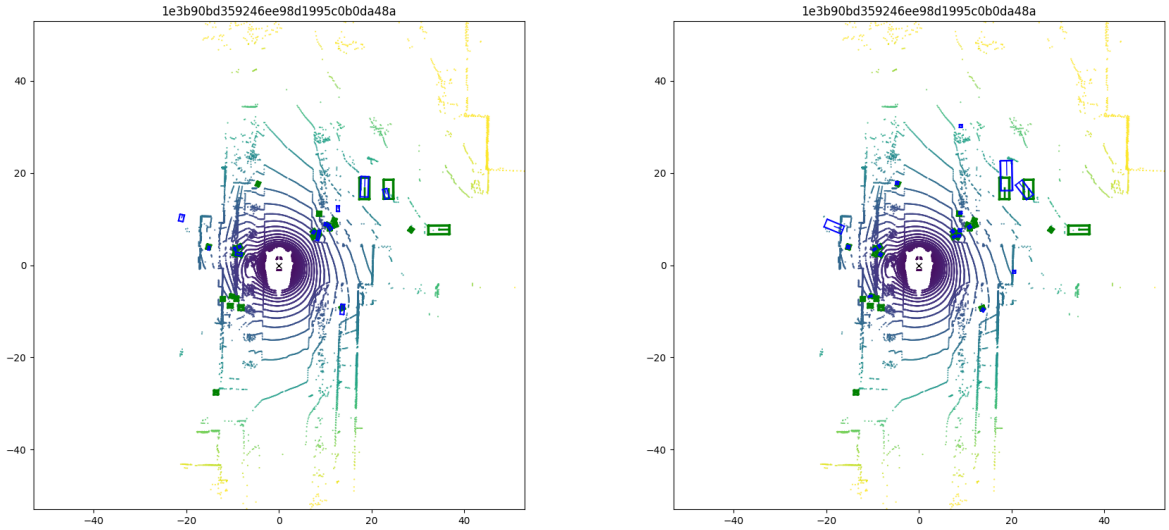

Figure 12. VESPA (Right) shows a clear advantage in separating and detecting smaller objects. UNION (Left) detects one large object more accurately, but misestimates another. VESPA's prediction for that object is slightly tilted, though overall object coverage is higher.

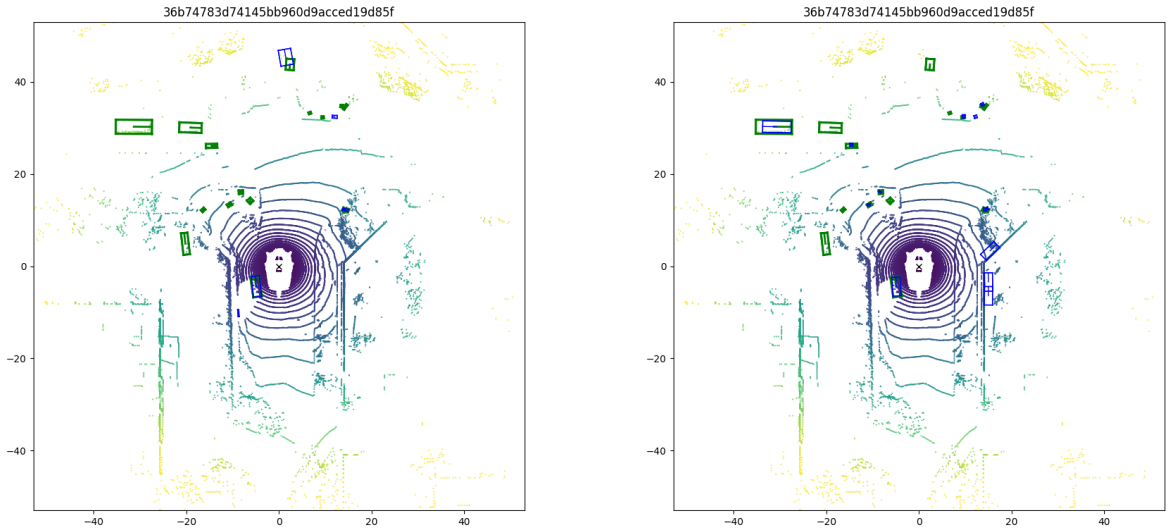

Figure 13. VESPA (Right) demonstrates strength in small object recall. It detects several small vehicles missed by UNION, though it also introduces more false positives. UNION (Left) shows higher precision, but its predictions are sparse and less complete.

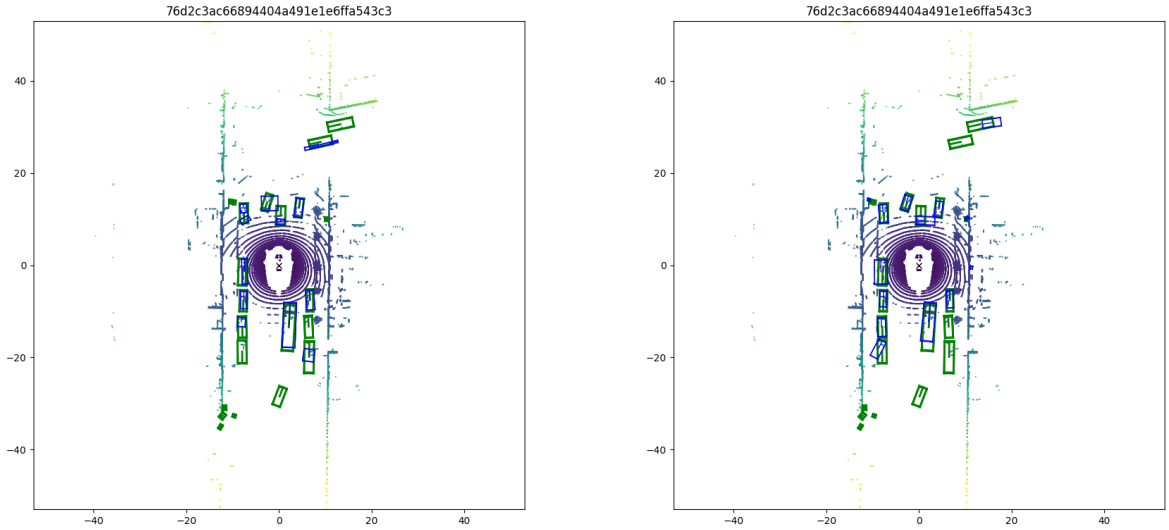

Figure 14. Both methods perform similarly on car-like objects. VESPA (Right) detects two additional small vehicles that UNION (Left) misses and offers slightly more accurate bounding boxes in some cases.

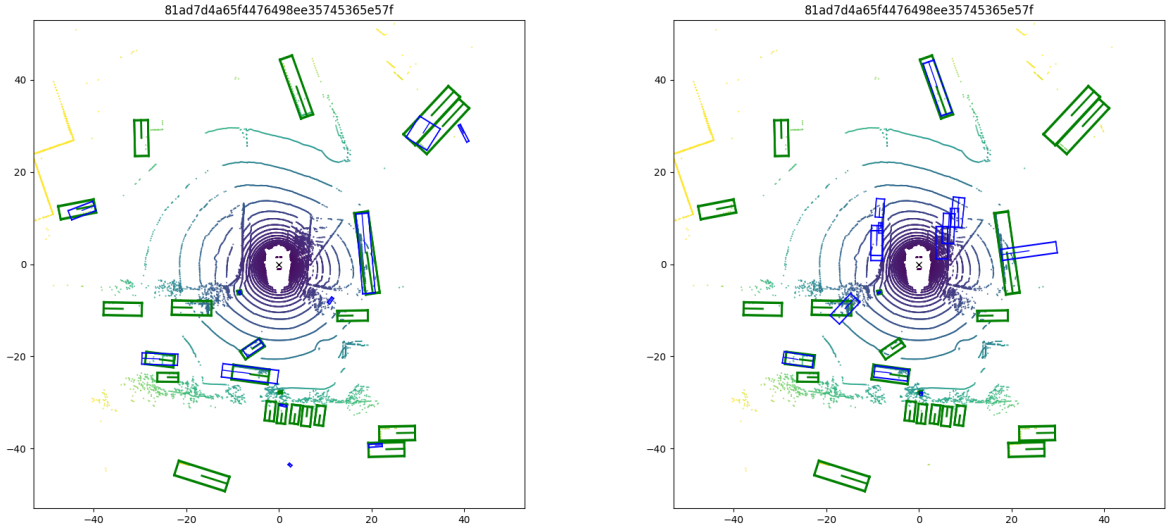

Figure 15. VESPA (Right) detects three true positives with high accuracy. One orientation estimate is off by 90 degrees, and several false positives appear near a vehicle. UNION (Left) produces much fewer false positives in this case.

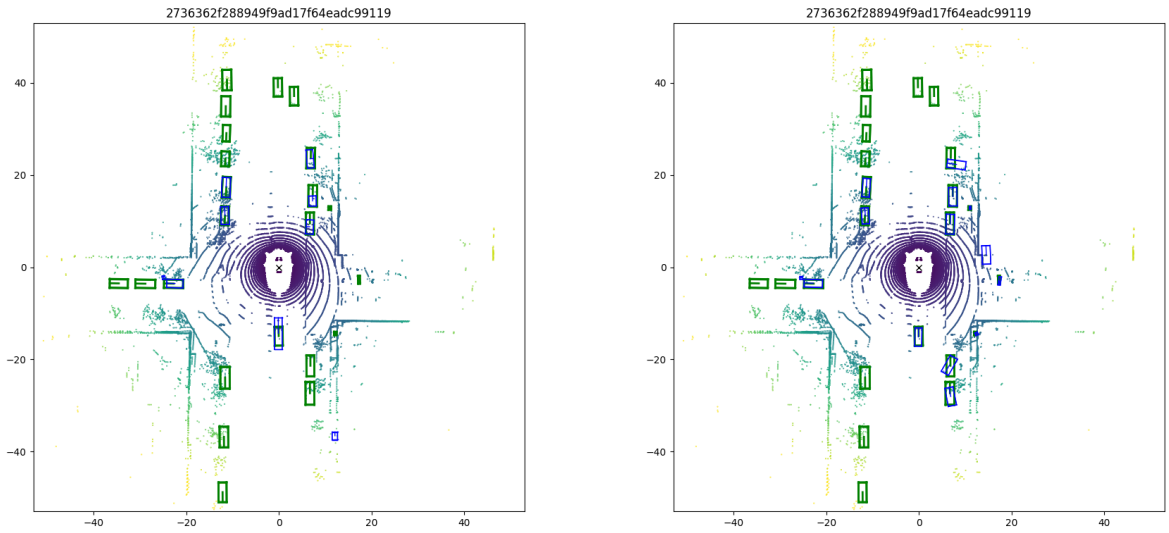

Figure 16. VESPA (Right) achieves significantly better recall, detecting all four small objects and two additional large ones. Most of its boxes, especially those closer to the ego vehicle, are well estimated. One box shows a 90-degree orientation error. UNION (Left) detects only one of the small objects and provides less accurate estimates overall.

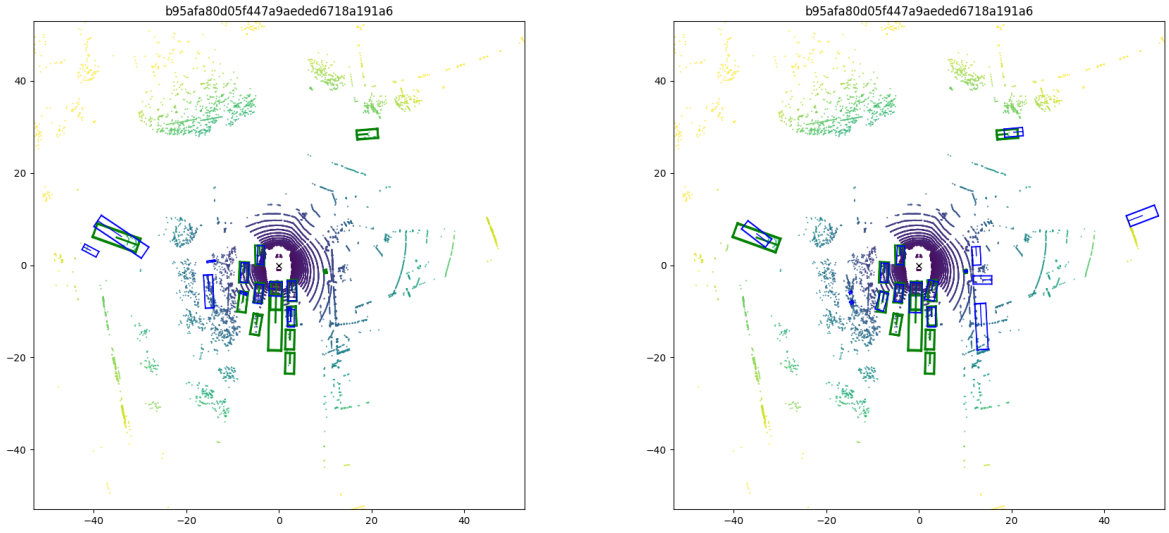

Figure 17. VESPA (Right) successfully identifying a distant object missed by UNION (Left), while also capturing a small nearby one. It produces several false positives, though its total true positive count slightly exceeds UNION's.

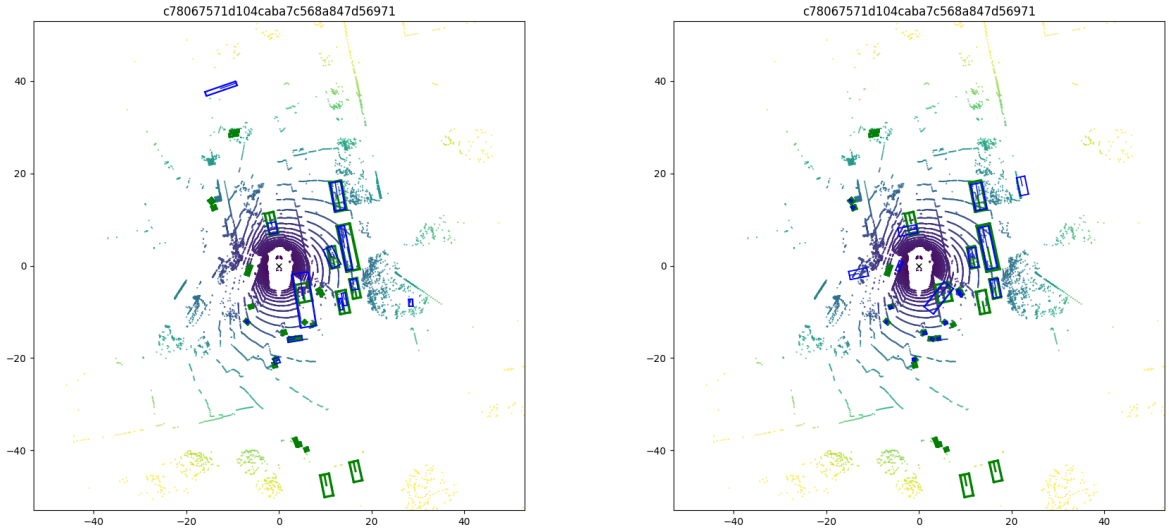

Figure 18. VESPA (Right) again outperforms UNION (Left) in small object detection. It identifies nearly all of them while UNION captures only a few. Most large object estimates by VESPA are close to ground truth, though two are misoriented. UNION's estimates for large objects are less accurate regarding box sizes.

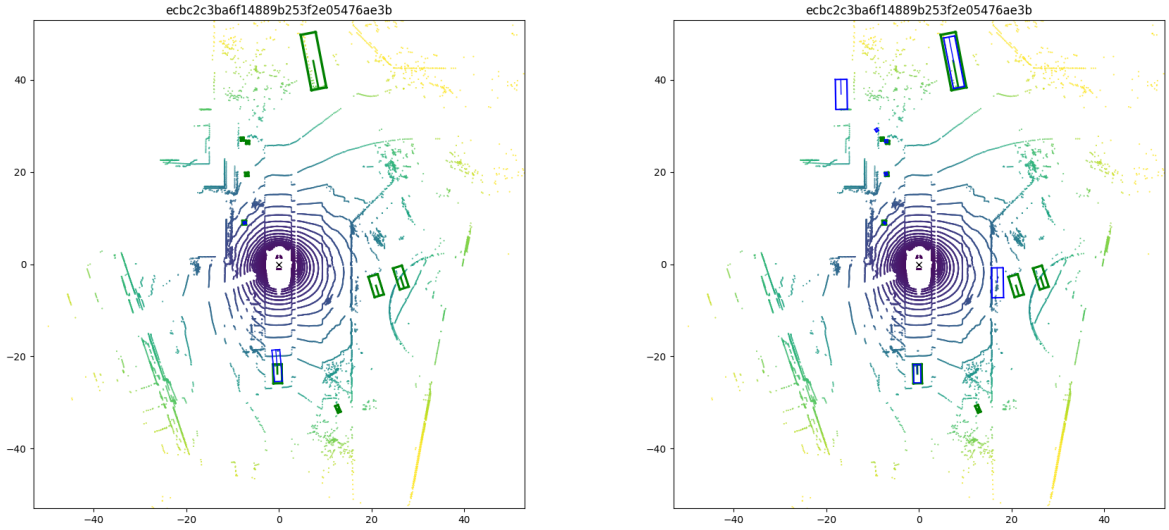

Figure 19. VESPA (Right) detects all four small objects, though one is slightly misplaced. It also captures a large distant vehicle with good box alignment, but also introduces 2 false positives. UNION (Left) misses three of the small objects and the distant vehicle. It detects a nearby car but overestimates its size.