

Incorporating Contextual Priors in Autonomous Vehicles

by

Prajwal Chougule
MSc Robotics

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday August 26, 2026.

Thesis committee: Dr. H. Caesar, TU Delft, supervisor
Dr. J.F.P. Kooij, TU Delft

Incorporating Contextual Priors in Autonomous Vehicles

Prajwal Chougule
Delft University of Technology

Abstract

Manual annotation of 3D bounding boxes in LiDAR point clouds is a major cost in constructing autonomous driving datasets, motivating offline auto-labelling pipelines. We investigate whether a graph neural network operating on object detector proposals can improve label quality. We construct a radius graph over detections in each nuScenes keyframe and apply a shared GATv2 backbone with classification and regression heads, including a direction classifier. Our joint model improves mAP from 50.25 to 52.48 ± 0.13 , NDS from 60.16 to 61.29 ± 0.08 , and mAOE from 0.3956 to 0.3783 ± 0.0022 ($n = 3$) on CenterPoint proposals evaluated on the nuScenes validation split. This improvement arises from rescoring and discrete 180° heading correction.

1. Introduction

Autonomous driving systems aim to reduce traffic accidents and improve transportation efficiency by enabling vehicles to perceive and interact with their surrounding environment. A reliable perception system provides the category, location, orientation and size of surrounding agents, and directly supports tracking, prediction and planning. In modern systems this is commonly performed using LiDAR point clouds, camera images, radar, or their combination [23, 30]. Perception in real traffic scenes nevertheless remains difficult: LiDAR point clouds are unstructured and non-uniform in density, producing sparse observations of distant or occluded objects, while camera-based methods suffer from depth ambiguity and sensitivity to lighting.

Recent work has turned to structured priors to incorporate contextual and semantic knowledge into detection pipelines. Map priors use high-definition (HD) maps to supply static semantic and geometric information such as lane topology and drivable area. This constrains where objects can plausibly be and how they should be oriented. It remains available beyond the immediate sensing range and under adverse observation conditions. Group priors model dependencies among the objects in a scene rather than treating detections as independent. They exploit spatial, geometric, semantic and temporal relationships be-

tween agents [15]. Patented systems combining both [2] underscore their practical relevance.

Annotating 3D bounding boxes in LiDAR point clouds is a major cost in constructing perception datasets for autonomous driving, motivating automated and semi-automated labelling pipelines. In this setting, proposal-level refinement offers a practical alternative to modifying the detector itself: contextual priors can be applied directly to its outputs without retraining or access to its internal representations. The resulting detection set is also naturally represented as a graph, allowing relational information between objects to be exploited. However, proposal-level post-processing is inherently limited to errors represented in the detector’s outputs and cannot recover missed objects or access the original point cloud. Moreover, improvements from graph-based refinement must be distinguished from those obtainable from per-detection evidence alone. This motivates a systematic evaluation of both the recoverable error at the proposal level and the additional value of relational reasoning.

We treat detection refinement as post-processing over a frozen CenterPoint [39]. A radius graph is built over the proposals in each nuScenes [3] keyframe, with nodes carrying box, map and LiDAR point features, and a shared GATv2 trunk feeds a rescoring head and a geometric refinement head using the SECOND residual encoding [36]. Modified detections are evaluated with the official nuScenes protocol. An oracle establishes the ceiling any regressor could reach, together with rule-based and graph-free learned baselines. Replicate training runs provide an estimate of run-to-run variance, and we report differences against it.

Our best configuration raises mAP from 50.25 to 52.48 and NDS from 60.16 to 61.29, and reduces mean orientation error from 0.3956 to 0.3783 (means over three runs). The improvement is driven by within-class confidence re-ranking and a discrete 180° heading correction, which resolves the rotational ambiguity of oriented boxes that cannot be determined from box parameters alone. A graph-free logistic regression on box and map features recovers a substantial fraction of the rescoring gain. This highlights the contribution of per box features, as opposed to relational

reasoning.

This work makes the following contributions:

- **A systematic evaluation of post detection refinement.** We quantify the available headroom with an oracle and compare rule-based, graph-free, and graph-based refinement under the official nuScenes protocol, with replicate runs used to quantify variability. The results show that the useful improvements are confidence-related and a discrete 180° heading correction, while continuous geometric refinement provides no measurable benefit despite substantial oracle headroom.
- **An evaluation of contextual priors for proposal refinement.** We compare detector box features, semantic map context, and per-object LiDAR statistics. Map features improve re-scoring, whereas LiDAR statistics provide no measurable benefit over box features and do not enable geometric refinement.
- **An analysis of how contextual priors should be integrated.** Map information improves refinement when provided as learned input features but degrades performance when imposed as hard geometric constraints. Moreover, a graph-free model using comparable per-detection features recovers a substantial fraction of the re-scoring gain. In this setting, contextual priors are therefore more effective as per-detection evidence than as relational constraints.

2. Related Work

2.1. Map priors

Maps are used in autonomous vehicles for various tasks ranging from object detection [9, 37] to motion forecasting [16]. They encode the static structure of the driving environment. The representation determines the ease of map integration into autonomous driving modules.

Rasterized maps encode map information in bird’s eye view (BEV) grids, where each cell stores semantic or geometric attributes such as drivable area. The grid structure of these maps allows priors to be combined at a fixed resolution naturally with CNN based encoders. This is beneficial at long ranges, where LiDAR point density is sparse [37]. Attention based fusion mechanisms mitigate scale and BEV misalignments [14]. However rasterized maps are incompatible with downstream systems such as path planning. They do not include instance information to differentiate map elements of the same class. Lastly, the centimetre level accuracy of HD maps is costly to construct. The cost of annotations can be mitigated by using global neural representations [35] or by using SD maps and satellite images as weakly aligned priors [12, 17].

Vectorized map representations offer a compact alternative to dense raster grids. They capture long-range dependencies and road topology more effectively than raster-

ized grids, as demonstrated by polyline and graph based approaches such as VectorNet [11] and LaneGCN [20]. Raster-to-vector methods such as HDMapNet [19] rely on non-differentiable post-processing, causing geometric and topological errors [7]. End-to-end approaches such as VectorMapNet [22] avoid these issues, but sequential prediction suffers from low inference efficiency and error accumulation, motivating parallel methods [16].

2.2. Group priors

Object detection has traditionally recognised objects independently, using only local cues. Such approaches can struggle in scenes where objects are partially occluded or embedded in cluttered environments. Several works in detection and segmentation frameworks show that context can improve object recognition performance [6, 8, 13, 40]. Statistical co-occurrence and feature aggregation are insufficient to capture the spatial structure of real world environments. In particular, geometric relationships between objects impose constraints on plausible detections [25].

Graph representations: Early graph based methods constructed graphs directly from raw point clouds, preserving fine grained geometric information at the cost of computational and memory overhead. Point-GNN [29] models LiDAR points as graph nodes connected by spatial proximity, while Graph R-CNN [38] constructs local proposal-level graphs augmented with semantic features from camera images. Subsequent methods introduced multi-scale graph representations. They aggregate point-level features into higher level representations [4] and facilitate information exchange across feature levels [5], which improves robustness in sparse and noisy point clouds.

Beyond point-level graphs, recent methods construct graphs at the feature [21] or proposal [33] level which reduces the graph size and computational overhead. These methods extend two-stage detectors by representing first-stage object proposals as graph nodes, with edges defined using K-nearest neighbour (KNN) or spatial proximity. This enables contextual reasoning by allowing each proposal to leverage information from its local neighbourhood rather than being refined independently. However, their effectiveness remains dependent on the quality of the initial detections, as missed objects cannot be recovered during subsequent graph reasoning.

In addition to graph based relational modelling, explicitly modelling geometric consistency/priors can complement contextual reasoning. BADet [27] uses correlations among neighbouring proposals to recover boundary information lost when proposals are offset from the ground truth, via iterative message passing to aggregate complementary features across an object. GACE [28] incorporates instance-specific and contextual geometric properties, including point distribution and sensor viewing angle, into

confidence estimation. This is particularly beneficial for sparsely observed vulnerable road users, where limited geometric evidence often leads to low-confidence detections.

Attention and Transformer architectures provide an alternative mechanism for modelling long-range contextual dependencies, often operating on structured representations such as graphs or BEV features. CenterFormer [42] leverages center features as initial query embedding for a transformer decoder. Additionally, it aggregates information across multiple BEV resolutions to capture object level context. VoTr [24] applies dilated attention over sparse voxels to efficiently search for distant voxels. Attention can also be integrated with graph representations: SAT-GCN [31] combines graph convolution with self-attention to adaptively reweigh neighbouring features. These approaches extend the receptive field beyond conventional convolutional operations and enable more flexible spatial interaction, although their increased modeling capacity can make them more susceptible to overfitting when training data is limited or lacks diversity.

2.3. Temporal context

In dynamic driving environments, temporal context reduces inconsistencies between frames by resolving occlusions. This connects object detection with tracking and motion reasoning, as demonstrated by PolarMOT [18], which exploits localized polar geometric relationships for multi-object tracking. STEMMD [41] models detections as nodes in a spatio-temporal graph where temporal cross attention maintains consistency in trajectories across frames. Ret3D [33] connects objects within tracked sequences via multi head self attention, demonstrating that transformers are efficient for modelling temporal object interactions.

Table 1 summarizes the methods discussed in this section according to the use of map and group priors. Group priors indicate explicit modelling of interactions between detections. As shown, existing approaches typically focus on either map based context or relational reasoning, while relatively few methods combine both within a unified framework.

3. Methodology

In this section, the overall pipeline of the proposed method is discussed, as shown in Fig 1. We treat detection refinement as a post-processing problem over the output of a frozen detector. For a single LiDAR keyframe, let

$$\mathcal{B} = \{b_1, \dots, b_N\}, \quad b_i = (\mathbf{c}_i, \mathbf{s}_i, \theta_i, p_i, k_i), \quad (1)$$

be the set of proposals emitted by a detector, where $\mathbf{c}_i$ is the box location, $\mathbf{s}_i$ its size, θ_i its heading, p_i the detector's confidence and k_i its predicted class. Our goal is to assign each proposal a refined confidence $\hat{p}_i$ and to predict geometric residual δ_i that modifies the box.

Table 1. Comparison of representative methods according to the use of map and group priors. R and V denote rasterized and vectorized map representations, respectively. Task abbreviations are Detection (Det), Map Construction (Map), Motion Forecasting (For), and Tracking (Tra).

Paper	Map	Rel.	Task
HDNet [37]	R	✗	Det
MapFusion [9]	R	✗	Det
LaneFusion [10]	R	✗	Det
MENet [14]	R	✗	Det
MIM [34]	R	✗	Det
BEVSegFormer [26]	R	✗	Map
Neural Map Prior [35]	R	✗	Map
HDMAPNet [19]	V	✗	Map
VectorNet [11]	V	✗	For
LaneGCN [20]	V	✓	For
CenterFormer [42]	✗	✓	Det
PolarMOT [18]	✗	✓	Tra
GraphRelate3D [21]	✗	✓	Det
GACE [28]	✗	✓	Det
Ret3D [33]	✗	✓	Det

Family	Dim.	Content
Box	19	position, size, yaw (sine-cosine), detector score, class one-hot
Map	11	multi-hot layer occupancy; distance, relative heading and validity of the nearest lane
LiDAR Point	16	log point count, range, viewing angle, per axis spread of enclosed points

Table 2. Node feature families.

3.1. Graph construction and message passing

Each LiDAR frame/sample is a graph $G = (V, E)$ whose nodes are the detector's proposals.

Node features are propagated by a two-layer GATv2 [1] trunk. For a query node i with neighbourhood $\mathcal{N}(i)$, GATv2 updates each node as follows:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{i,j} \Theta_t \mathbf{x}_j \quad (2)$$

The edge attribute $\mathbf{e}_{ij}$ enters GATv2 only through the scalar coefficient α_{ij} . The neighbour transform Θ_t is shared across all edges, so relative geometry can reweigh a message but cannot change its content.

3.2. Dataset construction

3.2.1. Node features

Node features are grouped into three families as summarised in Table 2. **Box features** are constructed from the

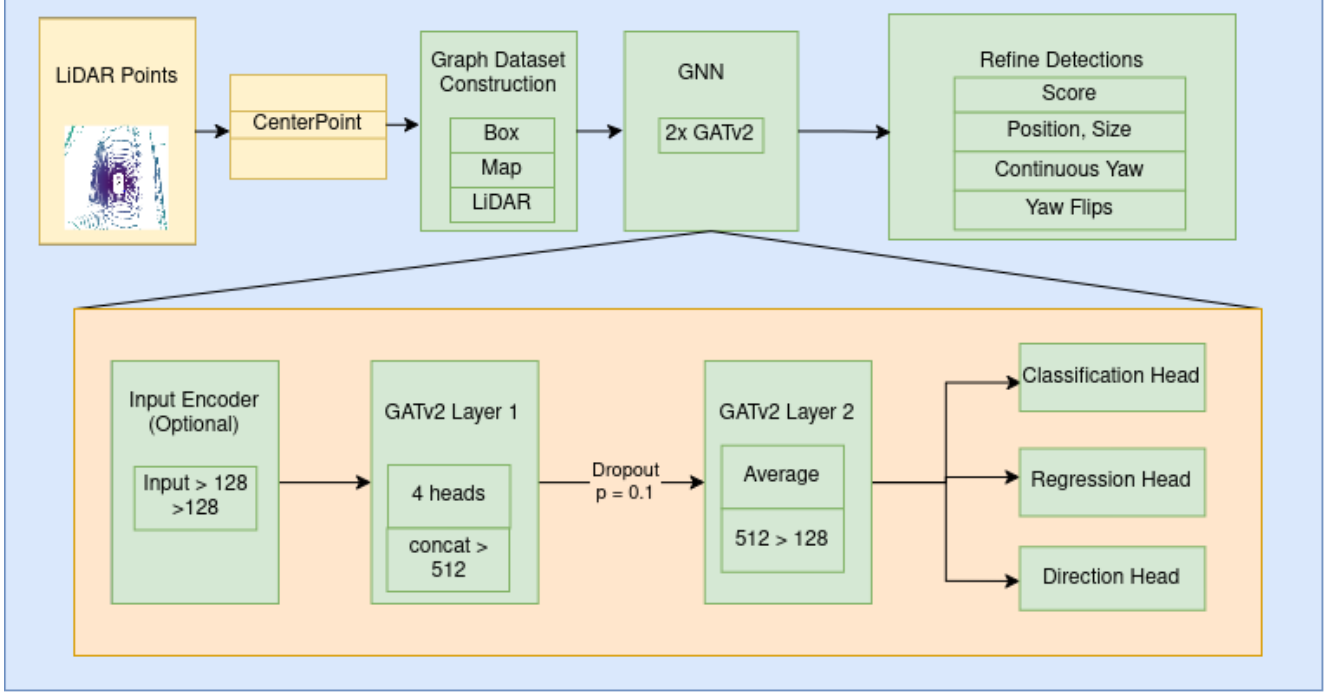


Figure 1. Overall Framework

proposals output by the object detector. These comprise of the box position, geometry, class and confidence score. Yaw is encoded as a sine–cosine pair rather than a raw angle to avoid discontinuity.

Map features add eleven dimensions derived from the nuScenes [3] semantic map. Seven of these indicate the map layers the box falls on, namely drivable area, road segment, lane, pedestrian crossing, walkway, stop line, and car park. They are multi-hot encoded because the layers overlap: a point on a lane also lies on the drivable area. The remaining four capture information about the nearest lane obtained by sampling lane centrelines at half metre resolution. A flag indicates whether any centreline was found within the search radius. The distance to the nearest lane is normalised and clipped at a threshold. Lastly the difference in yaw of the proposal box and the lane tangent is stored as a sine cosine pair.

The **LiDAR point** group adds sixteen features of per-object statistics inspired by [28]: First the log point count inside the box, the distance to the sensor, the sine and cosine of the viewing angle. The rest describe the spread of the LiDAR points in the box via the per-axis mean, standard deviation and minimum and maximum of the enclosed points. The points are normalised by translating to the box center, aligned to its heading, and scaled to unit extent. Ten LiDAR sweeps are aggregated per keyframe with ego-motion compensation into the keyframe’s sensor frame, matching the sweep count the detector operates on. A single sweep

yields a median of roughly two points per box, too few to estimate the per-axis statistics above.

3.2.2. Edges

Two proposals are joined by a bidirectional edge when their BEV centres lie within a radius threshold. The radius is a hyperparameter rather than a fixed property of the method, and we ablate it in Sec. 4. Each edge (i, j) carries a four-dimensional geometric attribute

$$\mathbf{e}_{ij} = [\Delta x_{ij}, \Delta y_{ij}, \|\Delta_{ij}\|, \Delta\theta_{ij}] \quad (3)$$

namely the relative BEV displacement, Euclidean distance, and the heading difference of the two proposals wrapped to $(-\pi, \pi]$.

Graph construction costs a dense pairwise distance evaluation over a few hundred proposals per frame, which is negligible beside the detector itself.

3.3. Prediction heads

The shared node embedding feeds two lightweight heads, and a third direction head under the SECOND encoding.

Classification head. A binary head predicts whether a proposal is a true or false positive under the nuScenes matching rule (center distance < 2 m to a ground-truth of the same class). It emits two logits trained with cross-entropy; at inference the softmax probability of the “true positive” class replaces the detector’s confidence score.

Regression head. A residual head predicts a per-proposal correction to the box, supervised only on nodes matched to ground truth. Its output layout depends on the target encoding : 8 dimensions $\Delta x, \Delta y, \Delta z, \Delta w, \Delta l, \Delta h, \sin \Delta \theta, \cos \Delta \theta$ for the plain encoding, or 7 dimensions for the SECOND encoding similar to [36], in which case a third *direction* head emits two logits resolving the π -flip ambiguity that the sine-error loss cannot see.

3.4. Loss functions

The model is trained with a weighted sum of the two head losses,

$$\mathcal{L} = \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \quad (4)$$

where $\mathcal{L}_{\text{cls}}$ is cross-entropy over all nodes and $\mathcal{L}_{\text{reg}}$ is a per-dimension Smooth-L1 applied *only to proposals matched to ground truth*; a false positive has no meaningful box residual, so regressing it would inject label noise.

Plain encoding. The residual loss is a weighted sum over the eight target dimensions,

$$\mathcal{L}_{\text{reg}}^{\text{plain}} = \sum_{k=1}^8 w_k \text{SmoothL1}(\hat{r}_k - r_k). \quad (5)$$

SECOND encoding. Translation and size use the same Smooth-L1 over the six scale-normalized dimensions, the yaw uses the sine-error loss, and a direction classifier resolves the π -flip:

$$\begin{aligned} \mathcal{L}_{\text{reg}}^{\text{SECOND}} = & \sum_{k \in \{x, y, z, w, l, h\}} w_k \text{SmoothL1}(\hat{r}_k - r_k) \\ & + w_{\theta} \text{SmoothL1}(\sin(\hat{\theta} - \theta)) \\ & + \beta \mathcal{L}_{\text{dir}}. \end{aligned} \quad (6)$$

with $\mathcal{L}_{\text{dir}}$ the cross-entropy of the flip classifier.

3.5. Training modes

To attribute effects to the two tasks separately, we train three variants: a joint model with both heads, a classification only model (rescoring), and a regression only model (refinement). The regression-only model keeps the detector’s original scores at inference, isolating box refinement from any change in confidence. As an upper bound on refinement, we also report an oracle that applies the ground truth residual to every matched proposal.

3.6. Inference and decoding

At test time the trained module is applied to the frozen detector’s outputs by rebuilding the identical graph used in training- the same radius, edge features, and node features- and running a single forward pass. The classification head

yields a refined score (the softmax probability of the true-positive class), and the regression head yields a residual that is decoded by inverting the target encoding. Normalized translation and log-size residuals are mapped back to metres and multiplicative scales, and, under the SECOND encoding, the direction logit applies the π flip. The decoded residual is added to the proposal box. The classification-only model leaves geometry untouched, and the regression-only model retains the detector’s original score, so each variant modifies exactly one aspect of the detection. Refined detections are written in the nuScenes submission format and scored by the official evaluator.

4. Experiments

4.1. Dataset and metrics

We evaluate on the nuScenes validation split using the official evaluator. Matching is by 2D center distance averaged over thresholds $\{0.5, 1, 2, 4\}$ m and the ten classes to give mAP. The true-positive metrics (mATE, mASE, mAOE, mAVE, mAAE) are computed at the 2m threshold over matched detections, and NDS combines them as $\text{NDS} = (5 \cdot \text{mAP} + \sum (1 - \min(1, \text{err}))) / 10$.

Because matching uses center distance, a corrected heading cannot promote a detection into the matched set, and an improved mAOE can therefore never raise mAP by construction. Rescoring and orientation refinement are consequently visible in disjoint columns of Tab. 3, and must be read separately.

4.2. Implementation details

All models use a two-layer GATv2 trunk of width 128 with $H=4$ heads, trained on NVIDIA A40 GPU for 100 epochs with Adam ($\text{lr} = 10^{-3}$, weight decay 10^{-4}) and a ReduceLROnPlateau schedule. Node features are 19-dimensional, extended to 30 with map features or 35 with LiDAR point statistics. Unless stated otherwise the graph radius is $\tau = 20$ m.

4.3. Main results and discussion

Tab. 3 reports the results. The configuration and features column provides information about the training. This includes whether the model was trained on the classification and regression tasks separately or together. Secondly it describes the node features used in the graph, as discussed in Sec 3.2.1. Lastly the decode column specifies what properties were modified in the bounding boxes. The letter S stands for confidence rescoring, P for location and size regression, Y for continuous yaw regression and F for the flip classifier. The velocity and attribute error columns have been greyed out as they are not trained for directly but are reported for completeness.

#	Configuration,Features	Decode	mAP $\uparrow$ (%)	NDS $\uparrow$ (%)	mAOE $\downarrow$ (rad)	mATE $\downarrow$ (m)	mASE $\downarrow$	mAVE $\downarrow$ (m/s)	mAAE $\downarrow$
1	CenterPoint	—	50.25	60.16	0.3956	0.3129	0.2593	0.3302	0.1989
2	<i>Oracle</i>	PY	55.83	72.58	0.0008	0.0027	0.0005	0.3307	0.1991
3	<i>Flip Oracle</i>	F	50.25	62.72	0.1390	0.3129	0.2593	0.3302	0.1989
<i>Re-scoring only</i>									
4	Box (Logreg)	S	50.21	60.08	0.3956	0.3163	0.2594	0.3323	0.1986
5	Box + Map (Logreg)	S	50.76	60.35	0.3948	0.3173	0.2590	0.3342	0.1976
6	Box	S	50.96	60.47	0.3917	0.3135	0.2579	0.3326	0.2055
7	Box+LiDAR	S	51.12	60.50	0.3903	0.3163	0.2582	0.3343	0.2074
<i>Regression only</i>									
8	Box+LiDAR	PYF	50.22	60.10	0.3967	0.3144	0.2610	0.3301	0.1990
9	Box + Map (EdgeConv)	PYF	50.23	60.20	0.3870	0.3151	0.2602	0.3301	0.1989
10	Box + Map	PYF	50.20	60.17	0.3875	0.3154	0.2607	0.3300	0.1988
11	Box + Map	YF	50.25	60.24	0.3876	0.3129	0.2593	0.3302	0.1989
12	Box + Map	F	50.25	60.24	0.3872	0.3129	0.2593	0.3302	0.1989
<i>Joint training</i>									
13	Box+Map (n=4)	SF	51.86	60.95	0.3811	0.3166	0.2576	0.3380	0.2041
14	Box+Map (r = 10m, n=3),	SF	52.27	61.16	0.3819	0.3169	0.2577	0.3386	0.2025
15	Box+Map (r = 10m , n=3, variant)	SF	52.48	61.29	0.3783	0.3173	0.2577	0.3388	0.2029
<i>Plain encoding</i>									
16	Box (w/o class)	SPY	50.65	60.21	0.3945	0.3157	0.2588	0.3364	0.2067
17	Box	SPY	51.17	60.57	0.3940	0.3163	0.2588	0.3325	0.1995

Table 3. Main results on the nuScenes validation split. Rows 13-15 report means over n independently trained runs

The first row reports the metrics for the unmodified Centerpoint detections. The oracle run replaces the regression targets from the ground truth in the dataset and keeps the original confidence scores of the boxes. This determines the ceiling for the regression targets. The mAP is not 100% due to undetected ground truth boxes and false positives. The flip oracle shows that 65% of the orientation error comes from a yaw angle difference greater than 90 degrees.

Rescoring: Rows 4 to 7 are re-scoring only models. #4 and 5 are graph free baselines. It rescored each proposal using its own features, hence no relational reasoning is involved. While just Box features have no effect, including map features improves its performance. Rule based baselines degraded performance for re-scoring as well as orientation and hence have not been included in this table. Details are mentioned in Sec. 8.

Rescoring aims to reduce the score of false positives and increase the score of true positives as observed in Fig 2. This allows the correct proposals to be matched to the ground truth. Even though this changes no other properties of the proposal than the score, it changes the other metrics. This is due to the fact that the boxes being selected as true positives change and have different location/size etc. The class-wise breakdown of average precision (Fig 3) shows that the most improvement is in classes with low AP such as motorcycle and trailer, while car has only a slight improvement.

The addition of LiDAR features gives a further +0.16 mAP. These configurations were not replicated, so we do not claim the difference as an improvement. The main reason to introduce it was to provide the model data about the location and size of the boxes. As it did not help in the regression task (#8), it was not used further.

Regression: Rows 10-12 demonstrate that the reduction in orientation error in the regression tasks is solely from the flip classifier. All the three runs use the same trained checkpoint and differ only in the decode method, which isolates their effect. The orientation error is almost constant even after zeroing the location, size and continuous yaw. The feature sets used here with the architecture do not successfully refine those variables. Additionally, the noise introduced by refining them increases translation and scale errors and hence they are not decoded in the joint runs. To test whether the aggregation operator limits the refinement, GATv2 was replaced by EdgeConv [32]. In GATv2 relative geometry influences only the attention coefficient, whereas EdgeConv incorporates relative geometry directly into the message function, allowing the message content itself to vary with the edge. Neither operator produced any reduction in translation or scale error.

Joint training: Rows 13 to 15 report the joint models, in which the classification and direction heads are trained on a shared trunk and decoded with confidence rescoring and the flip classifier (SF). Each configuration was trained

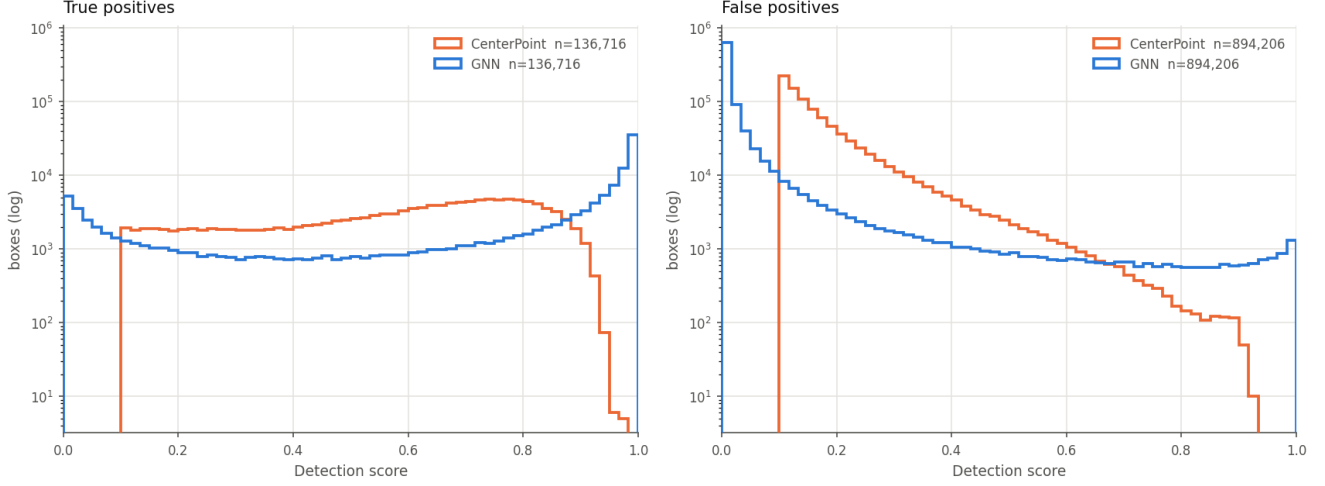


Figure 2. Detection score distribution

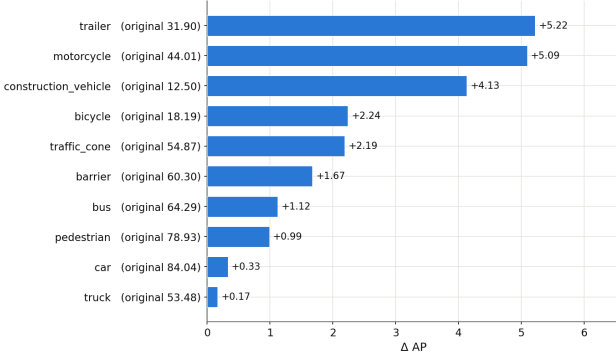


Figure 3. Per class Δ AP

independently four, three and three times respectively. Per run metrics and standard deviation is reported in Sec. 9

Row 15 differs from row 14 in one respect: traffic cones and barriers are excluded from the direction-classifier loss. The reason for their removal is that they contribute to 56.9% of the samples whose heading is inverted, while they cannot change mAOE and their direction targets carry no information. Traffic cone has no AOE and barrier is scored modulo π . Excluding them reduces the percentage of matched boxes needing flips from 20.2% to 11.6%.

All three configurations improve over CenterPoint. Reducing the graph radius to 10m (row 14) and additionally restricting the direction loss to orientation scored classes (row 15) raises the mean on mAP, NDS and mAOE. Bicycle is the only class the 20 m model degrades, and it recovers at 10 m. This is consistent with a rare class being diluted by an over-connected neighbourhood.

While the flip classifier reduces orientation error with precision around 0.82, it doesn't reduce the orientation error

completely due to its low recall (around 0.15). Rows 16 and 17 use the plain yaw parameterisation, in which the residual is regressed as a $(\sin \Delta\theta, \cos \Delta\theta)$ pair.

They differ only in whether class one-hot features are present, and the addition of class features raises mAP from 50.65 to 51.17. A qualitative example is shown in Fig. 4. Detections are correctly oriented at a distance of 40 metres, where LiDAR is sparser. Flips on false positives in grey don't affect the metric.

Attention analysis shows broad but spatially structured aggregation rather than strong neighbour selection (Sec 10).

5. Limitations

The experiments use a single detector which is CenterPoint. The transferability of the method across detector architectures has not been checked. Since the method operates on detector outputs, this is a limitation of our evaluation rather than of the method itself.

Moreover, refinement is limited by the detector's proposal set. 27.1% of the ground truth detections have no proposal within 2 m and are therefore unreachable by proposal level refinement, including the oracle method.

The detector is frozen and the GNN does not receive the dense BEV features used by the detector. Similarly, the single frame formulation excludes temporal cues, which are particularly informative for heading estimation.

Finally, labels in the pre-processed dataset are generated using greedy class-aware 2 m matching against unfiltered ground truth, rather than the range and point-count filtered ground truth used by the nuScenes evaluator.

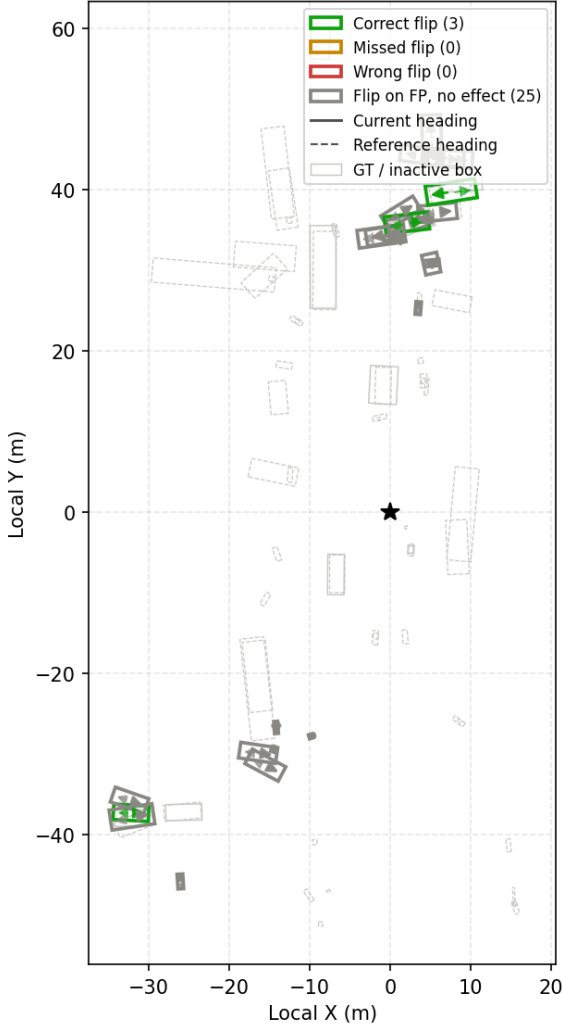


Figure 4. Qualitative example showing flips

6. Future Work

Future work can utilise the dense BEV features produced by the detector. The refinement can take place before the detection head by applying the graph over BEV features sampled at proposal locations. Additionally, extending the graph to include temporal reasoning can resolve π ambiguity as well as address sparsity and occlusion. Another direction is to formalize the theoretical limits of post-processing by characterizing the maximum performance recoverable from a fixed detector output and identifying error modes that are irrecoverable from those outputs. Such a framework could explain when black-box refinement is beneficial. Lastly, the graph can be used to hypothesise objects in occluded or sparsely-sampled regions from context. This will recover boxes not found by the detector.

7. Conclusion

This thesis investigated whether a graph neural network operating on the output of a frozen 3D object detector can improve detection quality by exploiting relationships between proposals and their map context. On nuScenes validation, the joint model improves mAP from 50.25 to 52.48 and NDS from 60.16 to 61.29 over CenterPoint, with orientation error reduced from 0.3956 to 0.3783.

Confidence rescaling accounts for a large fraction, and it concentrates on classes that don't perform well on the detector: trailer +5.22 AP, motorcycle +5.09, construction vehicle +4.13, against +0.33 for car. Geometric refinement of translation and continuous yaw does not improve over the detector under any feature set or operator tested, despite +5.6 mAP of headroom in the targets. The orientation gain comes entirely from the direction classifier.

References

- [1] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2022. 3
- [2] Holger Caesar and Edouard Francois Marc Capellier. Object orientation determination from map and group parameters. *World Intellectual Property Organization (WIPO), Patent No. WO2024030384A1*, 2024. <https://patents.google.com/patent/WO2024030384A1/en>. 1
- [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4
- [4] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020. 2
- [5] Jiaxin Chen, Xiang Li, Jin Xie, Jun Li, Jianjun Qian, and Jian Yang. Cbi-gnn: Cross-scale bilateral graph neural network for 3d object detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23124–23135, 2022. 2
- [6] Xinlei Chen and Abhinav Gupta. Spatial Memory for Context Reasoning in Object Detection. 2
- [7] Yixiao Chen, Ruining Yang, Xin Chen, Jia He, Dongliang Xu, and Yue Yao. From Static to Dynamic: a Survey of Topology-Aware Perception in Autonomous Driving, 2025. arXiv:2509.23641 [cs]. 2
- [8] Wenqing Chu and Deng Cai. Deep Feature Based Contextual Model for Object Detection, 2016. arXiv:1604.04048 [cs]. 2
- [9] Jin Fang, Dingfu Zhou, Xibin Song, and Liangjun Zhang. MapFusion: A General Framework for 3D Object Detection with HDMs, 2021. arXiv:2103.05929 [cs]. 2, 3
- [10] Taisei Fujimoto, Satoshi Tanaka, and Shinpei Kato. LaneFusion: 3D Object Detection with Rasterized Lane Map. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 396–403, 2022. 3

- [11] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation, 2020. arXiv:2005.04259 [cs]. 2, 3
- [12] Wenjie Gao, Jiawei Fu, Yanqing Shen, Haodong Jing, Shitao Chen, and Nanning Zheng. Complementing Onboard Sensors with Satellite Map: A New Perspective for HD Map Construction, 2024. arXiv:2308.15427 [cs]. 2
- [13] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. 2
- [14] Yuanxian Huang, Jian Zhou, Xicheng Li, Zhen Dong, Jinsheng Xiao, Shurui Wang, and Hongjuan Zhang. MENet: Map-enhanced 3D object detection in bird’s-eye view for LiDAR point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 120:103337, 2023. 2, 3
- [15] Mahtab Jamali, Paul Davidsson, Reza Khoshkangini, Martin Georg Ljungqvist, and Radu-Casian Mihailescu. Context in object detection: a systematic literature review. *Artificial Intelligence Review*, 58(6), 2025. Publisher: Springer Science and Business Media LLC. 1
- [16] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023. IEEE. 2
- [17] Zhou Jiang, Zhenxin Zhu, Pengfei Li, Huan-ang Gao, Tianyuan Yuan, Yongliang Shi, Hang Zhao, and Hao Zhao. P-MapNet: Far-seeing Map Generator Enhanced by both SDMap and HDMap Priors, 2024. arXiv:2403.10521 [cs]. 2
- [18] Aleksandr Kim, Guillem Brasó, Aljoša Ošep, and Laura Leal-Taixé. PolarMOT: How Far Can Geometric Relations Take Us in 3D Multi-Object Tracking?, 2022. arXiv:2208.01957 [cs]. 3
- [19] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An Online HD Map Construction and Evaluation Framework, 2022. arXiv:2107.06307 [cs]. 2, 3
- [20] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning Lane Graph Representations for Motion Forecasting, 2020. arXiv:2007.13732 [cs]. 2, 3
- [21] Mingyu Liu, Ekim Yurtsever, Marc Brede, Jun Meng, Walter Zimmer, Xingcheng Zhou, Bare Luka Zagar, Yuning Cui, and Alois Knoll. GraphRelate3D: Context-Dependent 3D Object Detection with Inter-Object Relationship Graphs, 2024. arXiv:2405.06782 [cs]. 2, 3
- [22] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. VectorMapNet: End-to-end Vectorized HD Map Learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 22352–22369. PMLR, 2023. 2
- [23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1
- [24] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiaoshi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3164–3173, 2021. 3
- [25] Jiyang Pan and Takeo Kanade. Coherent Object Detection with 3D Geometric Context from a Single Image. In *2013 IEEE International Conference on Computer Vision*, pages 2576–2583, Sydney, Australia, 2013. IEEE. 2
- [26] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. BEVSegFormer: Bird’s Eye View Semantic Segmentation From Arbitrary Camera Rigs, 2022. arXiv:2203.04050 [cs]. 3
- [27] Rui Qian, Xin Lai, and Xirong Li. Badet: Boundary-aware 3d object detection from point clouds. *Pattern Recognition*, 125:108524, 2022. 2
- [28] David Schinagl, Georg Krispel, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. GACE: Geometry Aware Confidence Enhancement for Black-box 3D Object Detectors on LiDAR-Data. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6543–6553, Paris, France, 2023. IEEE. 2, 3, 4
- [29] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020. 2
- [30] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 1
- [31] Li Wang, Ziyang Song, Xinyu Zhang, Chenfei Wang, Guoxin Zhang, Lei Zhu, Jun Li, and Huaping Liu. SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving. *Knowledge-Based Systems*, 259:110080, 2023. Publisher: Elsevier BV. 3
- [32] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 6
- [33] Yu-Huan Wu, Da Zhang, Le Zhang, Xin Zhan, Dengxin Dai, Yun Liu, and Ming-Ming Cheng. Ret3d: Rethinking object relations for efficient 3d object detection in driving scenes. *arXiv preprint arXiv:2208.08621*, 2022. 2, 3
- [34] Jinsheng Xiao, Shurui Wang, Jian Zhou, Ziyue Tian, Hongping Zhang, and Yuan-Fang Wang. MIM: High-Definition Maps Incorporated Multi-View 3D Object Detection. *IEEE Transactions on Intelligent Transportation Systems*, 26(3): 3989–4001, 2025. 3
- [35] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Neural Map Prior for Autonomous Driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023. IEEE. 2, 3
- [36] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 5

- [37] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018. [2](#), [3](#)
- [38] Honghui Yang, Zili Liu, Xiaopei Wu, Wenxiao Wang, Wei Qian, Xiaofei He, and Deng Cai. Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph. In *European conference on computer vision*, pages 662–679. Springer, 2022. [2](#)
- [39] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking, 2021. [1](#)
- [40] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-Occurrent Features in Semantic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 548–557, Long Beach, CA, USA, 2019. IEEE. [2](#)
- [41] Yifan Zhang, Zhiyu Zhu, Junhui Hou, and Dapeng Wu. Spatial-temporal graph enhanced detr towards multi-frame 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. [3](#)
- [42] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. CenterFormer: Center-based Transformer for 3D Object Detection, 2022. arXiv:2209.05588 [cs]. [3](#)

Incorporating Contextual Priors in Autonomous Vehicles

Supplementary Material

8. Rule-Based Baselines

To establish that the learned refinement is not reproducible by simple post-processing, we implemented handcrafted rules operating on the detected boxes and the map features.

8.1. Orientation Rules

The orientation rules modify only the yaw of each proposal. Thus every row in Tab. 4 reproduces the detector’s mAP, mATE and mASE. All rules decide only whether to add π . The “snap” variants additionally apply a continuous rotation. The rules work as follows:

Lane alignment. A box lying within $d_{\max}$ of a lane centreline is flipped when its heading opposes the lane tangent. Boxes with no lane within 10 m are excluded via the map’s validity flag.

Neighbourhood consensus. Motivated by parking lots, same-class neighbours within radius r that are parallel cast a score-weighted vote on the shared direction. Box i is flipped when the opposing evidence outweighs the agreeing evidence plus its own confidence. Three map and geometry based gates restrict who may vote. Opposite lane gate removes voting from detections in the lane going in the opposite direction. The lateral gate prioritises detections which lie on the left and right side. Lastly, the car park only gate applies flips only on detection in car parks layer of the map.

Consensus snapping. Rather than flipping, each box is rotated onto the score-weighted mean direction of its neighbours. This is the only rule that alters yaw by an amount other than π .

8.2. Rescoring Rules

The rescoring rules modify only detection score and leave all geometry intact.

Duplicate suppression. Within each class, detections are traversed in descending score order. A detection within 1 m of an already-retained detection is multiplied by 0.1.

Map consistency. Vehicle-class detections whose centres do not lie on drivable area, road segment or car park are multiplied by 0.5.

8.3. Results

Handcrafted rules increase mAOE compared to not refining the detections. (Tab. 4). One possible cause is the asymmetric cost of flips. An incorrect flip adds nearly π of error, whereas a correct flip repairs a box whose error only marginally exceeds $\pi/2$.

At $d_{\max} = 3\text{m}$ the nearest lane centreline includes opposing lanes, which flips correctly oriented vehicles. Reducing

Method	Flips	Prec.	mAOE↓
<i>Reference</i>			
CenterPoint (unmodified)	0	–	0.3956
GNN, flip decode	–	–	0.3872
<i>Orientation rules</i>			
Lane, $d_{\max} = 3\text{m}$	35,553	0.514	0.4416
Lane, $d_{\max} = 2\text{m}$	20,264	0.507	0.4303
Lane, $d_{\max} = 1\text{m}$	10,527	0.554	0.4002
Neighbour, $r = 4\text{m}$	17,679	0.362	0.4073
Neighbour, $r = 10\text{m}$	30,673	0.297	0.4549
+ opposite-lane gate	27,902	0.344	0.4465
+ lateral gate	14,322	0.276	0.4252
+ both gates	12,710	0.311	0.4208
+ car-park only	2,852	0.308	0.4105
Consensus snap only	0	–	0.3958
<i>Upper bounds</i>			
Flip oracle	27,597	1.000	0.1390

Table 4. Rule-based orientation baselines.

Method	mAP↑	NDS↑
CenterPoint score	50.25	60.16
Rule: duplicate only	49.30	59.63
Rule: map + duplicate	48.03	58.61
Logistic reg. (box)	50.21	60.08
Logistic reg. (box + map)	50.76	60.35

Table 5. Rule-based rescoring

the distance reduces mAOE. In the neighbourhood rule, the map gate prohibits voting between boxes whose lane tangents oppose. This removed 2771 incorrect flips, supporting the earlier hypothesis.

The duplicate rule reduces pedestrian AP from 0.789 to 0.702 presumably penalising detections in crowds. The same map information, supplied as input features to a logistic regression rather than as a hard rule, instead improves mAP from 50.21 to 50.76 (Tab. 5).

However, the handcrafted rules work in certain situations. For example, Tab. 6 shows that the neighbourhood consensus rule behave as intended for trailers, and that each gate improves it. It works better at $r = 10$ metres than at 4 metres, presumably due to the elongated size of trailers. The gates work as designed to increase the precision. Restricting it to car parks gives the highest precision but it acts on fewer detection resulting in a lower AOE reduction.

Since mAOE weights all classes equally, these gains are

Consensus variant	Flips	Prec.	AOE↓	mAOE↓
CenterPoint (no rule)	0	–	0.600	0.3956
$r = 4m$ (default)	263	0.471	0.605	0.4073
$r = 10m$	254	0.689	0.545	0.4549
+ opposite-lane gate	246	0.699	0.545	0.4465
+ lateral gate	118	0.817	0.531	0.4252
+ both gates	118	0.817	0.531	0.4208
+ car-park only	63	0.838	0.542	0.4105

Table 6. Effect of the neighbourhood consensus rule on trailers.

outweighed by degradation on other classes and every rule remains harmful in aggregate. Additionally the rules do not capture all the flips and hence the recall is low. Hence the assumption for the rules apply to a small minority of the objects in the dataset.

9. Run analysis

For a metric measured on n such runs, $x_1, \dots, x_n$, we report the sample mean, the Bessel-corrected sample standard deviation, and the standard error of the mean:

Sample mean

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Sample standard deviation (Bessel-corrected; $n-1$ because $\bar{m}$ is estimated from the same data, which costs one degree of freedom)

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Standard error of the mean

$$SE = \frac{s}{\sqrt{n}}$$

s describes the spread of one run; SE the uncertainty in the mean of n runs.

The CenterPoint baseline is a single deterministic evaluation of a fixed checkpoint and is therefore a constant μ_0 rather than a random variable. Each configuration is compared against it with a one-sample t -statistic,

$$t = \frac{\bar{x} - \mu_0}{SE}, \quad \nu = n - 1$$

and $|t|$ compared against tabulated two-tailed critical values: $t_{0.975} = 4.303$ and $t_{0.995} = 9.925$ for $\nu = 2$; $t_{0.975} = 3.182$ and $t_{0.995} = 5.841$ for $\nu = 3$. Table 7 reports the per run metrics, while table 8 reports the standard deviation and t -values.

Run	mAP ↑	NDS ↑	mAOE ↓	mATE ↓	mASE ↓	mAVE ↓	mAAE ↓
CenterPoint	50.25	60.16	0.3956	0.3129	0.2593	0.3302	0.1989
<i>Row 13: Box+Map, $r=20m$ ($n=4$)</i>							
1	52.28	61.18	0.3786	0.3171	0.2574	0.3364	0.2065
2	51.63	60.82	0.3783	0.3187	0.2582	0.3398	0.2043
3	52.16	61.15	0.3809	0.3163	0.2576	0.3360	0.2016
4	51.35	60.66	0.3866	0.3143	0.2571	0.3398	0.2040
<i>Row 14: Box+Map, $r=10m$ ($n=3$)</i>							
1	52.49	61.32	0.3791	0.3164	0.2575	0.3367	0.2027
2	51.99	60.97	0.3876	0.3169	0.2578	0.3379	0.2026
3	52.33	61.19	0.3789	0.3175	0.2578	0.3412	0.2020
<i>Row 15: Box+Map, $r=10m$, restricted dir. loss ($n=3$)</i>							
1	52.54	61.36	0.3759	0.3174	0.2575	0.3382	0.2020
2	52.34	61.20	0.3802	0.3184	0.2579	0.3379	0.2026
3	52.56	61.31	0.3789	0.3162	0.2577	0.3403	0.2041

Table 7. Per-run results for the replicated joint configurations (rows 13–15 of Tab. 3).

Metric	$\bar{x}$	s	SE	Δ	$ t $
<i>Row 13 ($n=4, \nu=3$)</i>					
mAP	51.86	0.44	0.22	+1.61	7.33 [†]
NDS	60.95	0.26	0.13	+0.80	6.25 [†]
mAOE	0.3811	0.0039	0.0019	−0.0145	7.50 [†]
<i>Row 14 ($n=3, \nu=2$)</i>					
mAP	52.27	0.26	0.15	+2.02	13.68 [†]
NDS	61.16	0.18	0.10	+1.00	9.72 [*]
mAOE	0.3819	0.0050	0.0029	−0.0137	4.78 [*]
<i>Row 15 ($n=3, \nu=2$)</i>					
mAP	52.48	0.13	0.07	+2.23	30.70 [†]
NDS	61.29	0.08	0.05	+1.13	23.39 [†]
mAOE	0.3783	0.0022	0.0013	−0.0173	13.58 [†]

Table 8. Sample mean $\bar{x}$, standard deviation s , standard error $SE = s/\sqrt{n}$, difference $\Delta = \bar{x} - \mu_0$ from unmodified CenterPoint, and the one-sample statistic $|t| = |\Delta|/SE$ with $\nu = n - 1$. *: $p < 0.05$; [†]: $p < 0.01$

10. Attention analysis

We characterise the structure of the learned attention distributions. Both statistics are computed from the head-averaged GATv2 attention weights a_{ij} . Attention is softmax-normalised over the incoming edges of each target node, including the self-loop added by GATv2. Statistics are computed per frame over the validation set and then averaged.

Normalised attention entropy. For each node i , with $n_i > 1$ incoming attention terms, we compute

$$H_i = \frac{-\sum_j a_{ij} \log a_{ij}}{\log n_i}, \quad H \in [0, 1]. \quad (7)$$

Here n_i denotes the number of incoming terms after self-loop insertion. $H_i = 1$ corresponds to uniform attention,

Table 9. Attention statistics for the trained models. H is the normalised attention entropy (Eq. 7); r is the attention–distance correlation (Eq. 8). Each row is a single trained checkpoint.

Model	Entropy H		Distance corr. r	
	conv1	conv2	conv1	conv2
Joint, 20 m	0.846	0.914	−0.422	−0.287
Joint, 10 m	0.851	0.893	−0.554	−0.440
Cls-only, 20 m	0.903	0.913	−0.114	−0.067

while values near zero indicate concentration on a small number of edges. We also use the corresponding effective support size $n_{\text{eff}} = n^H$ as an intuitive measure of distributional spread.

Attention–distance correlation. For non-self-loop edges, we compute the Pearson correlation between attention weight and Euclidean centre distance,

$$r = \text{corr}(a_{ij}, d_{ij}), \quad r \in [-1, 1]. \quad (8)$$

Negative values indicate greater attention to nearer detections. The correlation is computed within each frame and then averaged, characterising the spatial ordering of neighbours rather than between-frame variation.

The attention distributions are diffuse, with $H = 0.85$ – 0.91 across configurations. The high entropy indicates that attention is broadly distributed across the neighbourhood rather than concentrated on a small subset of neighbours such as those of the same class. This suggests that the graph aggregation provides a relatively broad pooled context, which may be reproducible by simpler models.

In the joint models attention is negatively correlated with distance in both layers. Thus they preferentially weigh closer detections while the classification-only model shows substantially weaker correlations.

Limitations These statistics provide descriptive characterisations of the learned attention rather than evidence of causal neighbour importance. Attention weights need not directly reflect a neighbour’s contribution to the prediction.

Each configuration is represented by a single checkpoint, so the results do not capture run-to-run variability. Additionally the distance correlation may reflect spatially correlated properties of the detections rather than distance alone.