REFINE CONNECTIONS, CLOSE THE GAP: A RELIABLE ENHANCEMENT FRAMEWORK FOR DRIVING SCENE TOPOLOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

In autonomous driving, understanding scene topology—the connectivity between lanes and traffic elements—is critical for safe path planning and motion control. While current methods excel at detecting individual map elements, their connectivity reasoning often falls short of its theoretical potential, leaving a significant performance gap relative to the theoretical upper-bound (Oracle) achievable given the underlying detections. Furthermore, the decision-ready topology graphs passed to downstream tasks often remain unreliable. Current approaches typically derive connectivity by thresholding continuous topology scores; however, these scores often fail to reflect the true logical likelihood of connectivity, resulting in false positives or missing connections. Existing benchmarks further overlook this issue by primarily evaluating continuous metrics (e.g., TOP score), rather than assessing the discrete connectivity required for decision-making. To bridge these gaps, we propose TopoEnhance, a novel topology enhancement framework designed to unlock the latent potential of existing methods and improve the reliability of decision-ready topology. We formulate topology enhancement as a denoising-based reconstruction process, where the model learns to recover structural consistency from stochastically corrupted ground-truth graphs. This formulation enables the model to resolve logical inconsistencies and rectify unreliable connections, producing robust discrete topology graphs that closely approach theoretical maximum performance. Extensive experiments across different baselines show that TopoEnhance consistently improves both continuous topology metrics (TOP_{ll} and TOP_{lt}), and discrete connectivity measured by our adapted Topology Jaccard Similarity (TJS) metric. As a flexible, source-agnostic framework, TopoEnhance delivers substantial gains across diverse state-of-the-art baselines without requiring retraining.

1 INTRODUCTION

In autonomous driving, understanding driving-scene topology is essential, as it encodes the structure of drivable regions and their interactions with traffic elements. For instance, when executing a left turn at an intersection, accurate topology understanding enables the ego vehicle to determine which lane to follow and which traffic signals govern its movement. Achieving this requires detecting both lanes and traffic elements (map element detection), inferring connectivity between lanes to form valid driving paths (lane–lane topology), and associating traffic elements with lanes to identify relevant control signals (lane–traffic topology). The resulting topology graph, in which nodes represent lanes or traffic elements and edges encode their relations, serves as the foundation for downstream tasks such as path planning (Chai et al., 2020) and trajectory prediction (Gu et al., 2024).

While accurate detection of map elements lays the foundation, the ultimate success of topology reasoning depends on making correct connectivity predictions. Even with perfectly detected map elements, erroneous connections can still lead to critical failures. For example, a planning module may choose an unsafe route if lane connections are predicted incorrectly, despite flawless detection. Beyond these safety concerns, we observe a substantial performance gap between existing methods and the theoretical upper-bound (Oracle) achievable in connectivity reasoning given their detection results. This gap suggests that existing models do not fully exploit the structural information already present in their detections, leaving significant room for improvement on topology connections. A

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

central goal of our work is to bridge this gap by developing a topology refinement framework that fully exploits the latent potential of existing methods.

Alongside this performance gap, another critical issue is the common oversight in how existing methods handle the final decision-ready topology. Specifically, they predict a topology score for every pair of detections and derive the final discrete topology graph through thresholding (Li et al., 2026; Wu et al., 2024; Can et al., 2022; Ye et al., 2025; Lv et al., 2025). However, the numerical values of these scores do not always faithfully reflect the true likelihood that a topology connection should exist. In multi-task learning settings ("detect" and "connect"), models can become prone to overconfidence or underconfidence due to competing objectives and gradient interference (Huang et al., 2025; Rotman & Reichart, 2022). Consequently, the resulting discrete topology graph can be unreliable with missing topology connection or false positive connections. Moreover, the quality of discrete connectivity is rarely evaluated explicitly. Current benchmarks only report mAP on continuous topology confidence scores for every possible detection pair (i.e., TOP score in OpenLane-V2 (Wang et al., 2023)), rather than directly assessing the correctness of the final discrete graph. As a result, existing methods largely overlook the reliability of discrete connectivity. However, downstream modules such as path planning depend on high-quality discrete driving topology graphs, making reliable connectivity essential. This discrepancy exposes a critical gap between the focus of existing methods and the practical needs of autonomous driving.

To address both of these challenges, we propose a novel driving topology enhancement framework, TopoEnhance, which formulates topology refinement as a diffusion-like denoising process to unlock the maximum topological reasoning potential inherent in existing detections. Given a set of detected map elements and topology confidence scores for each pair of detections, TopoEnhance acts as a independent judge that recovers a reliable discrete topologies by leveraging structural context in the driving scene. Since paired topology graphs before and after enhancement are unavailable, we develop a novel forward process (topology noise simulation) that stochastically corrupts ground-truth topology graphs to simulate realistic geometric and structural prediction errors. Complementing this, we introduce a specialized reverse process using a lightweight heterogeneous Graph Neural Network (GNN) to recover structurally consistent connectivity. This module is architected to jointly model the distinct relational dependencies of lane and traffic elements, ensuring both are optimized without one dominating the other. By exploiting this heterogeneous neighborhood information, TopoEnhance can potentially learn to identify true positive connections via general traffic rules—for example, a lane associated with right-turn successor lane should have near-zero likelihood of connecting to left-turn sign. By producing the well-calibrated topology scores, TopoEnhance leads to more reliable topology connections, bridging the gap toward the theoretical maximum performance and enhancing the robustness of downstream decision-making.

As a flexible source-agnostic framework, TopoEnhance enhance topology predictions from different sources without requiring retraining. In our experiments, we evaluate both the calibrated continuous topology scores and the resulting discrete topology after thresholding using the standard TOP score in OpenLane-V2 and a Topology Jaccard Similarity (TJS) metric we designed for this task, respectively. Across predictions generated from methods with diverse input modalities, TopoEnhance consistently improves continuous topology metrics (TOP_{ll}, TOP_{lt}) and delivers substantial gains on the discrete topology metric (TJS). For instance, although TopoLogic already achieves significant improvements in TOP scores over previous state-of-the-art methods, TopoEnhance further increases its TOP_{lt} from 25.4 to 27.2. In addition, while SMART attains the current state-of-the-art performance by leveraging additional auxiliary data (e.g., SD maps and satellite imagery), TopoEnhance still boosts its TOP_{ll} from 37.0 to 40.1 without retraining. In summary, our key contributions are threefold:

- We identify an often-overlooked problem of decision-ready topology quality and highlight the significant performance gap between current methods and their theoretical Oracle upper-bounds, presenting a major opportunity for improvement through topology refinement.
- We propose the first driving topology enhancement framework, TopoEnhance, which frames topology refinement as a denoising process to denoise and enhance the topology connections into more reliable decision-ready topology graphs.
- We conduct comprehensive experiments showing that TopoEnhance consistently improves both continuous topology metrics (TOP_{ll} and TOP_{lt}) and discrete topology metrics (TJS) across predictions from diverse sources, effectively bridging the gap toward theoretical upper-bound performance without retraining.

2 Related Work

Driving Topology Reasoning. Driving topology reasoning extracts lane–lane connectivity and lane–traffic associations to form structured road graphs, as formalized in OpenLane-V2 (Wang et al., 2023). Early DETR-style pipelines like STSU (Can et al., 2021) jointly predict lanes and relations, while subsequent works enhance reasoning through diverse architectures. TopoNet (Li et al., 2026) utilizes GNNs for joint modeling, whereas TopoMLP (Wu et al., 2024) employs a detect-then-reason pipeline with MLP heads. Further improvements integrate geometric–semantic cues (TopoLogic (Fu et al., 2024)), explicit point–lane interactions (TopoPoint (Fu et al., 2026)), and external priors such as SD maps (SMERF (Luo et al., 2024), TopoSD (Yang et al., 2024)), satellite imagery (SMART (Ye et al., 2025)), or 2D lane priors (Topo2D (Li et al., 2024)). Across these approaches, existing methods typically formulate topology as pairwise confidence scores and derive discrete graphs via thresholding. However, these scores often fail to reflect true logical likelihoods due to multi-task overconfidence or gradient interference (Huang et al., 2025; Rotman & Reichart, 2022), leaving a significant performance gap relative to the theoretical upper-bound. Furthermore, the reliability of the final discrete connectivity, though essential for downstream planning, is rarely explicitly optimized or analyzed. To bridge this gap, we introduce TopoEnhance, a dedicated enhancement framework that refines topology scores using structural context to produce more reliable decision-ready graphs.

Graph Neural Networks. Graph Neural Networks (GNNs) excel at modeling structured data by learning relational dependencies through message-passing (Wu et al., 2020). In driving scene topology, TopoNet (Li et al., 2026) pioneered using GNNs to capture map element interactions, followed by SMART (Ye et al., 2025) and TopoLogic (Fu et al., 2024), which integrated GNN-based heads for relational reasoning. More recently, TopoPoint (Fu et al., 2026) introduced a Point–Lane GNN for fine-grained feature aggregation. In this work, we propose a heterogeneous GNN-based refinement model featuring a transformer-inspired architecture. Compared with the homogeneous designs that treat all different relationships uniformly, our heterogeneous modeling employs distinct message-passing kernels for lane–lane and lane–traffic relations, acknowledging their unique semantic roles (Zhang et al., 2019). By employing Graph Attention (GAT) (Veličković et al., 2018) layers, our model adaptively weights neighbors to selectively emphasize structurally consistent connections while suppressing noise. This heterogeneous attention mechanism allows TopoEnhance to implicitly learn traffic-rule-aware patterns, ensuring logically consistent topology enhancement.

3 TopoEnhance

In this paper, our high-level objective is to bridge the performance gap in driving topology reasoning by ensuring that "decision-ready" topologies are structurally consistent and reliable for down-stream autonomous driving tasks. To this end, we propose a topology enhancement framework designed to unlock the latent potential of existing perception backbones. By training on simulated topologies containing diverse predictive errors, our framework learns to recover structural consistency, thereby enabling effective generalization to the inaccuracies found in real-world topology predictions. We frame this as a diffusion-like denoising problem on topology graphs: the forward process (Section 3.2) stochastically corrupts a ground-truth topology $\mathcal{G}^*$ into a noisy graph $\mathcal{G}_{\text{noise}}$ by injecting geometric and structural prediction errors; the reverse process (Section 3.3) learns to recover reliable topology from $\mathcal{G}_{\text{noise}}$ through a lightweight heterogeneous GNN-based refinement model. A high-level overview of TopoEnhance is illustrated in Figure 1.

3.1 Problem Formulation

Representation of Driving Scene Topology. We represent the driving scene topology as a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The node set $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_t$ consists of two distinct types: lane instances $\mathcal{V}_l = \{l_i\}_{i=1}^{n_l}$, encoded as 3D polylines, and traffic elements $\mathcal{V}_t = \{t_j\}_{j=1}^{n_t}$, represented by 2D bounding boxes. The connectivity $\mathcal{E}$ is categorized into two subsets: (1) lane–lane relations $\mathcal{E}_{ll} \subseteq \mathcal{V}_l \times \mathcal{V}_l$, capturing structures such as successors, merges, and splits; and (2) lane–traffic relations $\mathcal{E}_{lt} \subseteq \mathcal{V}_l \times \mathcal{V}_t$, capturing lane-to-sign assignments.

Enhancing Decision-Ready Topology. In existing pipelines, connectivity $\mathcal{E}$ is typically derived by thresholding raw topology scores $\hat{\mathcal{S}} = \{s_{uv} \in [0,1] \mid u, v \in \hat{\mathcal{V}}\}$ to obtain a discrete graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ where $\hat{\mathcal{E}} = \{(u,v) \mid s_{uv} \geq \tau\}$. However, because these scores can be biased and fail to reflect the

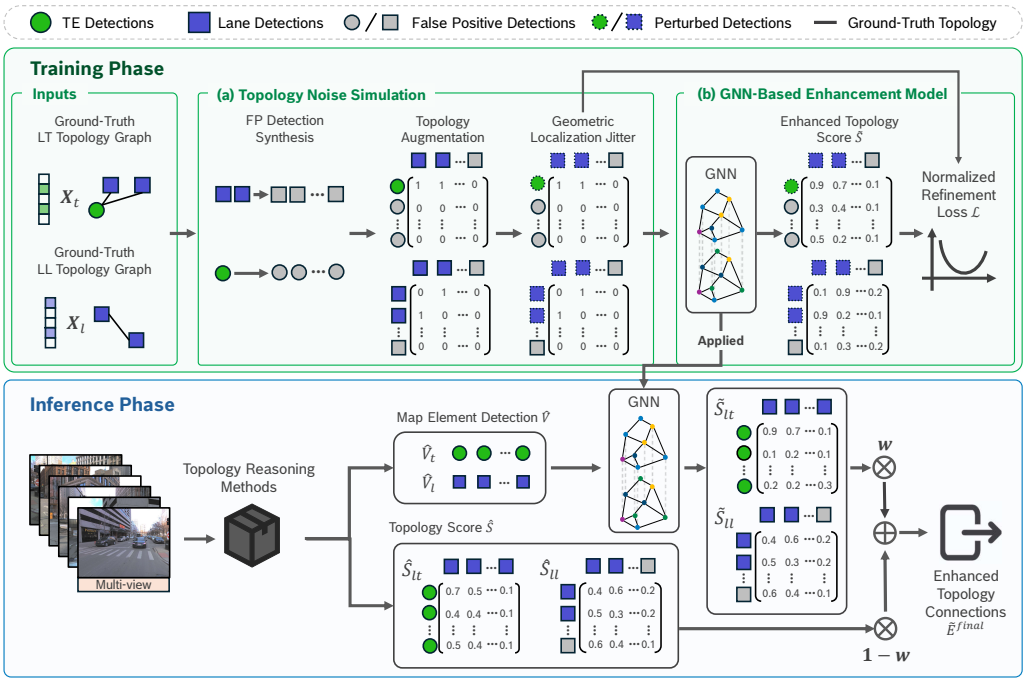


Figure 1: Overall framework of TopoEnhance. Training phase: (a) Forward process (topology noise simulation): a novel noise simulation strategy that stochastically corrupts ground-truth topology graphs into noisy graphs $\mathcal{G}_{\text{aug}}$ by injecting realistic geometric and structural prediction errors. (b) Reverse process (topology denoising): a lightweight heterogeneous GNN learns to recover reliable topology connections from the simulated noisy graphs. Inference phase: once trained, the enhancement model is directly applied to calibrate topology scores from any topology reasoning method, without retraining.

true likelihood of edge existence, the predicted topology $\hat{\mathcal{G}}$ can be viewed as a corrupted observation of the ground-truth $\mathcal{G}^*$, suffering from false-negative or false-positive connections. We therefore formulate topology enhancement as a reverse process: learning a mapping f_θ that recovers calibrated topology scores $\tilde{\mathcal{S}} = f_\theta(\hat{\mathcal{V}}, \hat{\mathcal{S}})$ from the corrupted $\hat{\mathcal{G}}$, such that the resulting discrete topology $\tilde{\mathcal{E}}$ achieves higher structural fidelity to $\mathcal{G}^*$ by resolving logical inconsistencies through heterogeneous neighborhood context.

3.2 Forward Process: Topology Noise Simulation

The forward process defines the corruption distribution $q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*)$, which stochastically transforms a ground-truth topology $\mathcal{G}^*$ into a noisy topology $\mathcal{G}_{\text{noise}}$. Unlike image-based diffusion models where Gaussian noise can be added directly to continuous pixel values, designing a forward process for driving topology graphs is non-trivial. A driving topology graph is a heterogeneous discrete structure consisting of 3D polylines, 2D bounding boxes, and categorical connectivity relations. Corruption must be applied separately and meaningfully to each component, such that the resulting corruption distribution $q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*)$ closely matches the distribution of real-world topology predictions $p(\hat{\mathcal{G}})$. Without this alignment, the reverse process $p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})$ will fail to generalize on real-world topology predictions. To address this, we design the forward process through three components that jointly simulate the realistic error distribution of real-world topology predictions.

False-Positive Detection Synthesis. A primary failure in driving topology reasoning is the detection of false positives, i.e., lane instances or traffic elements that are either geometrically disparate from the

ground-truth instances or entirely non-existent in the physical scene. We simulate these by expanding the detection set $\mathcal{V}_{\text{noise}} = \mathcal{V}^* \cup \mathcal{V}_{fp}$. To generate the false-positive detection set $\mathcal{V}_{fp}$, we sample from a high-variance distribution, accepting a candidate detection v_{fp} only if its geometric distance from the source ground-truth instance v exceeds a category-specific distance threshold δ_{type}:

$$v_{fp} = v + \epsilon, \quad \text{s.t.} \quad \text{dist}(v, v_{fp}) > \delta_{type}, \quad \epsilon \sim \mathcal{N}(0, \sigma_{fp}^2 \mathbf{I}), \tag{1}$$

where $v \in \mathcal{V}^*$ is a ground-truth instance and $type \in \{l, t\}$ refers to the lane or traffic element category.

Topology Augmentation. These false-positive detections are then used to synthesize unreliable connectivities that f_θ must learn to prune. We define the candidate edge set $\mathcal{E}_{\text{noise}}$ by generating all potential pairs $\{(u, v) \mid u, v \in \mathcal{V}_{\text{noise}}\}$. To supervise the reverse process, we assign ground-truth edge labels y_{uv} as:

$$y_{uv} = \mathbb{I}[(u, v) \in \mathcal{E}^*]. \tag{2}$$

This labeling scheme naturally ensures that any connection involving at least one false-positive detection from $\mathcal{V}_{fp}$ is assigned a negative label ($y_{uv} = 0$), as these nodes by definition do not exist in $\mathcal{V}^*$. By populating $\mathcal{V}_{fp}$ such that the augmented node set matches the expected cardinality of detections during inference ($|\mathcal{V}_{\text{noise}}| = \mathbb{E}[|\hat{\mathcal{V}}|]$), the candidate edge space $\mathcal{E}_{\text{noise}}$ matches the size of the raw topology scores $\hat{\mathcal{S}}$ to be calibrated.

Geometric Localization Jitter. In practice, the map element predictions $\hat{\mathcal{V}}$ can never perfectly reconstruct ground-truth geometries; even true-positive detections contain localization jitter in their 3D polylines or 2D bounding boxes. If f_θ were trained only on perfect true-positive detections, it would fail to generalize to the noisy coordinates encountered during inference. To bridge this gap, we apply a constrained perturbation to all ground-truth detections $v \in \mathcal{V}^*$, transforming the raw geometry into a jittered representation v':

$$v' = v + \epsilon, \quad \text{s.t.} \quad \text{dist}(v, v') \leq \delta_{type}, \quad \epsilon \sim \mathcal{N}(0, \sigma_{tp}^2 \mathbf{I}) \tag{3}$$

By suppressing the perturbation within δ_{type}, these detections remain semantically valid yet distinguishable from the false positives in $\mathcal{V}_{fp}$, forcing f_θ to resolve connectivity from topological context rather than exact coordinate matching.

Adaptive Noise Curriculum. To expose the forward process $q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*)$ to a broad spectrum of corruption severities, we design the noise variances σ_{tp}^2 and σ_{fp}^2 as dynamic functions of training progress. Inspired by curriculum learning (Wang et al., 2021), we design a curriculum-like schedule which enables a gradual transition from coarse error correction to fine-grained discrimination. Specifically, σ_{tp} is initialized near zero and progressively increased toward the distance threshold δ_{type}, while σ_{fp} starts at a relatively large value and is gradually reduced toward δ_{type}. As both noise profiles converge to δ_{type}, the geometric distinction between true and false detections becomes increasingly subtle. This design can theoretically improve training efficiency and generalization of our framework (Arora et al., 2026).

The complete forward process is summarized in Algorithm 1. By design, the simulated corruption $q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*)$ closely approximates the error distribution of real-world predictions $\hat{\mathcal{G}}$. This alignment ensures that error-correction patterns learned on $\mathcal{G}_{\text{noise}}$ naturally transfer to $\hat{\mathcal{G}}$ during inference, without requiring retraining on any specific prediction source.

3.3 Reverse Process: Topology Enhancement Model

The reverse process learns the conditional distribution $p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})$, which recovers reliable topology from the corrupted topology $\mathcal{G}_{\text{noise}}$. It is achieved with a heterogeneous GNN-based enhancement model f_θ that performs type-aware message passing over lane and traffic-element nodes, enabling distinct processing of lane–lane and lane–traffic relations. Through attention-based aggregation, f_θ leverages structural context to suppress inconsistent edges while reinforcing connectivity patterns aligned with lane flow and traffic regulations.

Heterogeneous GNN Architecture. Given the noisy graph $\mathcal{G}_{\text{noise}} = (\mathcal{V}_{\text{noise}}, \mathcal{E}_{\text{noise}})$, we employ a heterogeneous GNN to learn relational representations. Node features (X_l, X_t) are initialized by encoding lane instances using their 3D polyline coordinates, while traffic elements are represented by visual features extracted from their 2D bounding boxes. The node embeddings are iteratively updated

through heterogeneous message passing. Specifically, for each node $v \in \mathcal{V}_{\text{noise}}$, its embedding $\mathbf{h}_v^{(k)}$ at layer k is updated by aggregating information from its multi-type neighborhood:

$$\mathbf{h}_v^{(k)} = \sigma\left(\sum_{r\in\mathcal{R}} \text{AGG}_{(u,v)\in\mathcal{E}_{\text{noise}}}\left(\psi_r\left(\mathbf{h}_v^{(k-1)}, \mathbf{h}_u^{(k-1)}\right)\right)\right), \tag{4}$$

where $\mathcal{R} = \{ll, lt\}$ denotes lane–lane and lane–traffic relations, and ψ_r is a relation-specific transformation function. We utilize a lightweight two-layer Heterogeneous Graph Attention Network (GAT) as aggregation function, whose attention mechanism learns relation-aware coefficients to emphasize structurally relevant neighbors while suppressing noisy or false-positive detections. After obtaining final node embeddings, we compute the refined score $\tilde{s}_{uv} \in \tilde{\mathcal{S}}$ via sigmoid-activated cosine similarity:

$$\tilde{s}_{uv} = \sigma\big(\mathbf{h}_u^\top \mathbf{h}_v / (\|\mathbf{h}_u\|\|\mathbf{h}_v\|)\big), \quad \forall (u,v) \in \mathcal{E}_{\text{noise}} \tag{5}$$

where $\sigma(\cdot)$ denotes the sigmoid function, ensuring $\tilde{s}_{uv} \in (0,1)$.

Normalized Refinement Loss. We derive the training objective directly from the reverse process $p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})$. Assuming conditional independence of candidate edges given $\mathcal{G}_{\text{noise}}$, the reverse distribution factorizes as:

$$p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}}) = \prod_{(u,v)\in\mathcal{E}_{\text{noise}}} \text{Bernoulli}(y_{uv};\, \tilde{s}_{uv}) = \prod_{(u,v)\in\mathcal{E}_{\text{noise}}} \tilde{s}_{uv}^{y_{uv}}(1-\tilde{s}_{uv})^{1-y_{uv}}, \tag{6}$$

where y_{uv} is the ground-truth edge label from Equation 2 and $\tilde{s}_{uv} \in (0,1)$ is the sigmoid-activated cosine similarity from Equation 5. Taking the negative log-likelihood of Equation 6 yields the standard binary cross-entropy (BCE) over all candidate edges, as shown in Lemma 1. However, applying BCE jointly over all edges is problematic: the much larger lane–lane candidate space dominates the gradient, starving the lane–traffic relation of useful learning signal. We therefore adopt a relation-aware, volume-normalized BCE. For each relation $r \in \{ll, lt\}$, the per-relation loss is:

$$\mathcal{L}^{(r)} = -\frac{1}{|\mathcal{E}_r|}\left(\sum_{(u,v)\in\mathcal{E}_r^{\text{pos}}} \log \tilde{s}_{uv} + \sum_{(u,v)\in\mathcal{E}_r^{\text{neg}}} \log(1-\tilde{s}_{uv})\right), \tag{7}$$

where $|\mathcal{E}_r|$ is the total number of candidate edges for relation r, $\mathcal{E}_r^{\text{pos}} = \{(u,v) \in \mathcal{E}_r \mid y_{uv} = 1\}$, and $\mathcal{E}_r^{\text{neg}} = \{(u,v) \in \mathcal{E}_r \mid y_{uv} = 0\}$. The overall objective balances the two relation types as:

$$\mathcal{L} = \mathcal{L}^{(ll)} + \mathcal{L}^{(lt)}. \tag{8}$$

As formalized in Lemma 1 and Proposition 1 (Appendix A), $\mathcal{L}$ can be interpreted as a relation-balanced formulation of the reconstruction term in a single-step $T{=}1$ diffusion evidence lower bound (ELBO) (Sohl-Dickstein et al., 2015; Ho et al., 2020).

Training and Inference. During training, f_θ is optimized on simulated noisy graphs $\mathcal{G}_{\text{noise}}$ from the forward process. The parameters θ are optimized against ground-truth labels $\mathcal{Y}$:

$$\theta^* = \arg\min_\theta \sum_{(u,v)\in\mathcal{E}_{\text{noise}}} \mathcal{L}(\tilde{s}_{uv}, y_{uv}), \quad \text{where } y_{uv} \in \mathcal{Y}. \tag{9}$$

During inference, given detection predictions $\hat{\mathcal{V}}$ and raw topology scores $\hat{\mathcal{S}} = \{s_{uv} \mid u, v \in \hat{\mathcal{V}}\}$, we construct the inference graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}}^{\text{cand}})$, where $\hat{\mathcal{E}}^{\text{cand}} = \hat{\mathcal{V}} \times \hat{\mathcal{V}}$. Passing $\hat{\mathcal{G}}$ through f_θ yields refined scores $\tilde{\mathcal{S}} = f_\theta(\hat{\mathcal{G}})$, which are combined with the initial uncalibrated scores via a relation-specific fusion weight w_r:

$$s_{uv}^{\text{final}} = w_r\, \tilde{s}_{uv} + (1-w_r)\, \hat{s}_{uv}, \quad \forall (u,v) \in \hat{\mathcal{E}}^{\text{cand}}. \tag{10}$$

Finally, the decision-ready discrete topology is obtained by thresholding:

$$\tilde{\mathcal{E}}^{\text{final}} = \{(u,v) \in \hat{\mathcal{E}}^{\text{cand}} \mid s_{uv}^{\text{final}} \geq \tau\}. \tag{11}$$

4 Experiments

Dataset. Our experimental studies are performed on OpenLane-V2 (Wang et al., 2023), a comprehensive benchmark tailored for driving scene topology understanding and developed on top of the Argoverse2 (Wilson et al., 2021) and nuScenes (Caesar et al., 2020) datasets. The benchmark

supplies rich annotations spanning lane centerline reconstruction, traffic element recognition, and topological relationship modeling. It is organized into two subsets, Argoverse2 (subset_A) and nuScenes (subset_B), each containing 1,000 annotated driving sequences captured at 2 Hz with synchronized multi-view camera inputs. Both splits provide supervision for lane geometry, traffic elements, and the connectivity relations between lanes as well as between lanes and traffic elements. *For all experiments, we train one model per dataset and apply it to enhance predictions from different methods at inference time, without any method-specific retraining.* For more implementation details, please refer to Appendix C.

4.1 Evaluation Metrics

For all topology metrics, we first establish a correspondence between predicted detections $\hat{V}$ and ground-truth detections V^*. Following OpenLane-V2 (Wang et al., 2023), detections are matched using the discrete Fréchet distance for lanes and IoU for traffic elements. We define the set of true-positive detections, $\hat{V}_{\text{TP}} \subseteq \hat{V}$, as those predictions that successfully match a ground-truth counterpart.

Continuous Topology Metrics (OpenLane-V2 TOP). To evaluate the impact of our enhancement on the continuous topology scores $\mathcal{S}$, we adopt the TOP metric from the OpenLane-V2 benchmark (Wang et al., 2023) (version 2.1.0). TOP is an mAP-based metric that measures how well potential connections are ranked according to their topology scores $\mathcal{S}$, ensuring that true connections are ordered above false ones. TOP score computes average mAP between (V^*, E^*) and $(\hat{V}, \hat{E})$ over all vertices:

$$\text{TOP} = \frac{1}{|V^*|} \sum_{v \in V^*} \frac{\sum_{\hat{n} \in \hat{N}(v)} P(\hat{n}) \mathbf{1}(\hat{n} \in N(v))}{|N(v)|}, \tag{12}$$

where $P(\hat{n})$ is the precision of neighbor $\hat{n}$ in the ordered list ranked by s_{uv}. By evaluating the full ranking of candidate neighbors, TOP directly captures improvements in score calibration: promoting true positive connections on top of false positives. We report TOP both before and after enhancement to quantify improvements in ranking quality. Also, TOP_{ll} and TOP_{lt} are computed for lane–lane and lane–traffic relations, respectively.

Discrete Topology Metrics (TJS). While TOP score evaluates the relative ranking of continuous confidence scores, downstream autonomous driving modules operate on a discrete topology graph obtained by thresholding. A model can achieve a high TOP score by ranking edges correctly in a relative sense, yet still produce a final graph with significant missing or spurious connections if the scores are not well-calibrated. To directly assess the quality of this final "decision-ready" output, we introduce the Topology Jaccard Score (TJS), which adapts the Jaccard Similarity (Jaccard, 1912) to evaluate topological connectivity. TJS measures the overlap between the predicted and ground-truth edge sets, providing a direct assessment of discrete graph quality. Let $\mathcal{E}^*$ be the set of ground-truth edges and $\hat{\mathcal{E}}^{cand}$ be the set of all candidate edges. TJS is defined as:

$$\text{TJS} = |\mathcal{E}^* \cap \hat{\mathcal{E}}| / |\mathcal{E}^* \cup \hat{\mathcal{E}}|, \text{ where } \hat{\mathcal{E}} = \{(u, v) \in \hat{\mathcal{E}}^{cand} : s_{uv} > \tau\}. \tag{13}$$

By explicitly penalizing both false positives and false negatives in the final discrete topology graph, *TJS serves as a discrete complement to the continuous TOP score, and together they provide a more comprehensive evaluation of topology connection quality.*

Theoretical Upper-Bound (Upper-Bound). To isolate the performance of topology reasoning from the quality of the upstream detector, we define a theoretical upper bound for both TOP and TJS. This bound represents the maximum achievable performance given the existing set of detection predictions $\hat{\mathcal{V}}$. Specifically, we assume an oracle that assigns the final confidence scores s^*_{uv},

$$s^*_{uv} = \begin{cases} 1.0 & \text{if } (u, v) \in \mathcal{E}^* \text{ and } u, v \in \hat{\mathcal{V}}_{\text{TP}} \\ 0.0 & \text{otherwise} \end{cases}. \tag{14}$$

By evaluating the metrics using s^*_{uv}, we establish a performance ceiling where all possible true connectivities are perfectly recovered and ranked, while errors stemming from missing detections or false-positive detections remain. Comparing our results against this bound quantifies the remaining "headroom" for topological refinement and distinguishes between errors caused by connectivity reasoning versus those inherent to the detection backbone.

Table 1: Enhanced results on OpenLane-V2 measured by the TOP metric (version 2.1.0), which evaluates the quality of continuous topology scores. Bolded blue values indicate improved score over the corresponding baselines before applying TopoEnhance.

Dataset	Method	Before TopoEnhance		After TopoEnhance		Upper-Bound	
		TOP_{ll} ↑	TOP_{lt} ↑	TOP_{ll} ↑	TOP_{lt} ↑	TOP_{ll} ↑	TOP_{lt} ↑
Argoverse2 (Subset_A)	TopoNet (Li et al., 2026)	10.9	23.8	**21.8**	**25.8**	24.4	28.8
	TopoMLP (Wu et al., 2024)	21.6	26.9	**24.4**	**28.7**	25.0	31.4
	Topo2D (Li et al., 2024)	22.3	26.2	**24.3**	**27.3**	25.0	29.8
	TopoLogic (Fu et al., 2024)	23.9	25.4	**24.5**	**27.2**	26.0	30.6
	SMART (Ye et al., 2025)	37.0	33.0	**40.1**	**35.4**	40.9	38.9
nuScenes (Subset_B)	TopoNet (Li et al., 2026)	6.7	16.7	**19.5**	**17.8**	21.2	20.3
	TopoMLP (Wu et al., 2024)	21.0	19.8	**22.8**	**21.5**	23.5	24.1

4.2 Enhanced Results

In this section, we present enhanced results obtained by applying TopoEnhance on existing topology reasoning methods with publicly available checkpoints to refine their topology connections. To the best of our knowledge, the methods with officially released checkpoints include TopoNet (Li et al., 2026), TopoMLP (Wu et al., 2024), Topo2D (Li et al., 2024), TopoLogic (Fu et al., 2024), and SMART (Ye et al., 2025). Among these methods, SMART achieves state-of-the-art performance in topology reasoning by leveraging additional data modalities, such as SD maps and satellite imagery. In contrast, the other methods rely solely on perspective camera images as input. Beyond methods with publicly released checkpoints, we also attempted to train several additional approaches from scratch. However, we were unable to reliably reproduce their reported results, likely due to differences in training environments or hardware configurations. To ensure fairness and reliability in our comparisons, we therefore restrict our evaluation to methods with official checkpoints. Additionally, please refer to Appendix D for additional evaluation results and Appendix E for evaluation on our practical usability toward real-world applications.

Quantitative Comparison on Continuous Topology Scores. Table 1 summarizes the performance enhancement on continuous topology scores (OpenLane-V2 TOP) across multiple existing topology reasoning methods. Our enhancer demonstrates consistent and significant improvements in the TOP metric for all evaluated baselines, regardless of their underlying architecture or input modalities. Notably, our model nearly triples the performance of TopoNet ($6.7 \rightarrow 19.5$ TOP_{ll}) on nuScenes and provides a significant boost to TopoMLP ($21.6 \rightarrow 24.4$ TOP_{ll}) on Argoverse2. Even for the state-of-the-art methods SMART, which utilizes auxiliary data such as satellite imagery and SD maps, our model can still further boosts performance ($37.0 \rightarrow 40.1$ TOP_{ll}) and ($33.0 \rightarrow 35.4$ TOP_{lt}) during inference, *without any retraining and adaptation to the additional input modalities*. This indicates that by modeling the structural dependencies between predicted entities, our model successfully pushes true positive edges above spurious ones in the topology score ranking. A key observation from Table 1 is that our refined topology scores closely approach the Theoretical Upper-Bound. Since the Upper-Bound represents the maximum achievable performance given the fixed detection predictions $\hat{\mathcal{V}}$, the narrow gap remaining indicates that our enhancer has nearly exhausted the potential for topological refinement within the limits of the provided detections. Specifically, on Argoverse2, our enhancer reduces the margin to the maximum reachable performance in TOP_{ll} to just 0.6 for TopoMLP and 0.7 for Topo2D. For the SMART baseline, the enhanced TOP_{ll} of 40.1 sits within a 0.8 margin of the theoretical ceiling. These results validate that our approach effectively serves as a universal enhancement layer, capable of leveraging heterogeneous neighborhood context to rectify local connectivity errors inherent in existing prediction heads. *Notably, while our main motivation is to improve discrete connectivities via topology calibration, our calibration also boost commonly agreed TOP score by correcting the relative ordering between true positives and false positives.*

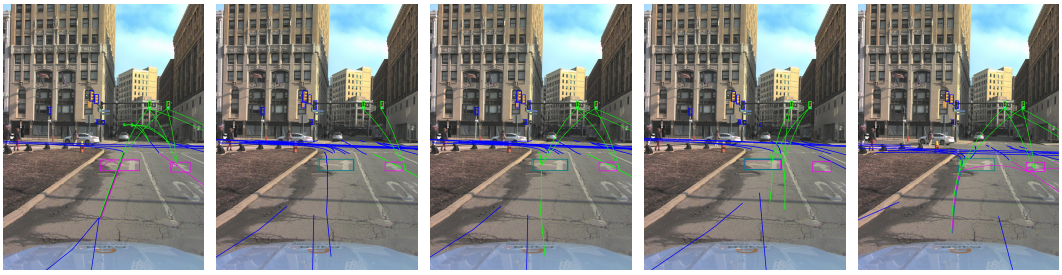
Quantitative Comparison on Discrete Topology Connections. While the TOP metric measures ranking quality of continuous topology scores, Topology Jaccard Score (TJS) serves as a discrete complement to evaluate the accuracy of the final discrete graph used for downstream tasks. As shown in Table 2, our enhancer consistently and reliably delivers substantial gains in final decision-ready topology graphs. For TopoMLP on Argoverse2, the TJS_{ll} nearly triples from a 18.5 to 59.7 and the TJS_{lt} doubles from a 31.5 to 59.8. For TopoNet on nuScnenes, the TJS_{ll} rises from 9.0 to 42.3 and the TJS_{lt} improves from a 35.6 to 56.8. These significant boost indicates that while baseline methods

Table 2: Enhanced results on OpenLane-V2 evaluated using the TJS metric, which measures the quality of discrete topology connections.

Dataset	Method	Before TopoEnhance		After TopoEnhance		Upper-bound	
		TJS_{ll} ↑	TJS_{lt} ↑	TJS_{ll} ↑	TJS_{lt} ↑	TJS_{ll} ↑	TJS_{lt} ↑
Argoverse2 (Subset_A)	TopoNet (Li et al., 2026)	16.3	31.0	**32.9**	**55.8**	44.3	63.7
	TopoMLP (Wu et al., 2024)	18.5	31.5	**59.7**	**59.8**	69.7	72.8
	Topo2D (Li et al., 2024)	18.9	34.5	**55.0**	**51.5**	68.7	70.8
	TopoLogic (Fu et al., 2024)	21.0	32.5	**43.9**	**53.3**	50.9	71.4
	SMART (Ye et al., 2025)	38.1	36.6	**87.7**	**82.8**	92.9	90.1
nuScenes (Subset_B)	TopoNet (Li et al., 2026)	9.0	35.6	**42.3**	**56.8**	48.5	66.7
	TopoMLP (Wu et al., 2024)	15.9	29.3	**51.3**	**62.6**	68.9	82.2

may have a reasonable relative rankings on topology scores, their numerical values of topology scores are often poorly calibrated—exhibiting overconfidence for false positives or underconfidence for true connections. This lack of calibration leads to numerous missing or false-positive connections when scores are thresholded to produce the final graph. Even for the state-of-the-art SMART model, which already achieves the strongest baseline performance, our enhancer boosts the performance even further, enhancing TJS_{ll} from 38.1 to 87.7 and TJS_{lt} from 36.6 to 82.8. These results suggest that TopoEnhance, trained on a curriculum of diverse simulated noise, produces more reliable decision-ready topology graphs. By reducing both false positives and false negatives, it yields cleaner topology structures for downstream tasks such as path planning.

Qualitative Comparison. Figure 2 shows that TopoEnhance effectively corrects missing and fragmented connections in TopoNet and TopoMLP, such as unlinked traffic lights in TopoNet or inconsistent lane merges in TopoMLP. More qualitative results are provided in Appendix G.



(a) Ground-truth Topology | (b) TopoNet Prediction | (c) TopoNet + TopoEnhance | (d) TopoMLP Prediction | (e) TopoMLP + TopoEnhance

Figure 2: Qualitative comparison of topology connections before and after enhancement.

5 Conclusion

Reliable topology connectivity is fundamental to autonomous driving, as downstream modules such as trajectory prediction and path planning rely on high-fidelity discrete topologies. In this work, we identify an often-overlooked gap in existing topology reasoning systems: while prior methods focus on predicting continuous topology scores, the reliability of the final decision-ready topology graph after thresholding is rarely considered or explicitly evaluated. To bridge this gap, we propose TopoEnhance, a source-agnostic topology enhancement framework that refines topology connections using heterogeneous structural context. By reconstructing stochastically corrupted graphs, TopoEnhance learns to resolve logical inconsistencies and recover structural consistency. Experiments on OpenLane-V2 show that TopoEnhance consistently improves both topology score ranking (TOP) and discrete topology accuracy (TJS) across diverse sources without retraining. These results highlight structural refinement as a key step toward closing the performance gap in decision-ready topology. More broadly, this work emphasizes the necessity of moving beyond continuous topology score prediction to prioritize the structural reliability of the discrete connections used by autonomous systems. We hope our findings encourage future research to focus on producing robust, structurally consistent driving topology connections for safe and reliable autonomous driving.

AI USE STATEMENT

In this work, generative AI tools were used only to improve the clarity of the writing. They were not used to generate research ideas, develop methods, or conduct experiments. All research contributions were developed and verified by the authors. The authors reviewed all AI-assisted text and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of our work, we have included details of the model implementation, training configurations, and evaluation protocol in Section 3, Section 4, Appendix B, Appendix C, and Appendix D. We will publicly release our implementation of noise simulation, model training and inference, and evaluation metrics.

REFERENCES

Raman Arora, Yunjuan Wang, and Kaibo Zhang. When does curriculum learning help? a theoretical perspective. In *Proceedings of Advances in Neural Information Processing Systems*, volume 38, pp. 7595–7630, 2026.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *Proceedings of International Conference on Learning Representations*, 2022.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11618–11628, 2020.

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pp. 15641–15650, 2021.

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Topology preserving local road network estimation from single onboard camera image. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17242–17251, 2022.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of Conference on Robot Learning*, pp. 86–99, 2020.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *Proceedings of International Conference on Learning Representations Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Yanping Fu, Wenbin Liao, Xinyuan Liu, Hang Xu, Yike Ma, Yucheng Zhang, and Feng Dai. Topologic: An interpretable pipeline for lane topology reasoning on driving scenes. In *Proceedings of Advances in Neural Information Processing Systems*, volume 37, pp. 61658–61676, 2024.

Yanping Fu, Xinyuan Liu, Tianyu Li, Yike Ma, Yucheng Zhang, and Feng Dai. Topopoint: Enhance topology reasoning via endpoint detection in autonomous driving. In *Proceedings of Advances in Neural Information Processing Systems*, volume 38, pp. 118229–118250, 2026.

Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14521–14530, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Wenke Huang, Jian Liang, Guancheng Wan, Didi Zhu, He Li, Jiawei Shao, Mang Ye, Bo Du, and Dacheng Tao. Be confident: Uncovering overfitting in MLLM multi-task tuning. In *Proceedings of International Conference on Machine Learning*, 2025.

Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

Han Li, Zehao Huang, Zitian Wang, Wenge Rong, Naiyan Wang, and Si Liu. Enhancing 3d lane detection and topology reasoning with 2d lane priors. *arXiv preprint arXiv:2406.03105*, 2024.

Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Hang Xu, Chunjing Xu, Junchi Yan, Ping Luo, and Hongyang Li. Graph-based topology reasoning for driving scenes. *Science China Information Sciences*, 69(5):152103, 2026.

Adam Lilja, Junsheng Fu, Erik Stenborg, and Lars Hammarstrand. Localization is all you evaluate: Data leakage in online mapping datasets and how to fix it. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22150–22159, 2024.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of International Conference on Learning Representations*, 2017a.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017b.

Katie Z Luo, Xinshuo Weng, Yan Wang, Shuang Wu, Jie Li, Kilian Q Weinberger, Yue Wang, and Marco Pavone. Augmenting lane perception and topology understanding with standard definition navigation maps. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4029–4035, 2024.

Changsheng Lv, Mengshi Qi, Liang Liu, and Huadong Ma. T^2SG: Traffic topology scene graph for topology reasoning in autonomous driving. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17197–17206, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Guy Rotman and Roi Reichart. Multi-task active learning for pre-trained transformer-based models. *Transactions of the Association for Computational Linguistics*, 10:1209–1228, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of International Conference on Machine Learning*, pp. 2256–2265, 2015.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of International Conference on Learning Representations*, 2018.

Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, Feng Wen, Hang Xu, Ping Luo, Junchi Yan, Wei Zhang, and Hongyang Li. OpenLane-v2: A topology reasoning benchmark for unified 3D HD mapping. In *Proceedings of Advances in Neural Information Processing Systems*, volume 36, pp. 18873–18884, 2023.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2021.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of Annual Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Dongming Wu, Jiahao Chang, Fan Jia, Yingfei Liu, Tiancai Wang, and Jianbing Shen. TopoMLP: A simple yet strong pipeline for driving topology reasoning. In *Proceedings of International Conference on Learning Representations*, pp. 45604–45615, 2024.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

Sen Yang, Minyue Jiang, Ziwei Fan, Xiaolu Xie, Xiao Tan, Yingying Li, Errui Ding, Liang Wang, and Jingdong Wang. TopoSD: Topology-enhanced lane segment perception with SDMap prior. *arXiv preprint arXiv:2411.14751*, 2024.

Junjie Ye, David Paz, Hengyuan Zhang, Yuliang Guo, Xinyu Huang, Henrik I Christensen, Yue Wang, and Liu Ren. SMART: Advancing scalable map priors for driving topology reasoning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3298–3304, 2025.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 793–803, 2019.

Appendix

Table of Contents

A Theoretical Diffusion-Based Interpretation of TopoEnhance **13**

B Algorithmic Summary of Forward Process: Topology Noise Simulation **15**

C Implementation Details **15**

D Additional Evaluation Results **16**
- D.1 TopoEnhance on Geographically Non-Overlapping Splits 16
- D.2 Enhancing TopoPoint Predictions 16
- D.3 Complete OpenLane-V2 Metrics 17
- D.4 Ablation Studies 18

E Practical Usability Toward Real-World Applications **19**
- E.1 Robustness to Varying Levels of Topology Noise 19
- E.2 Temporal Consistency of Refined Topology Predictions 20
- E.3 Safety-Relevant Topology Error Analysis 21
- E.4 Scene-Level Ego-Route Recoverability for Navigation and Planning 23
- E.5 Efficiency Analysis For Real-Time Inference 23

F Limitation and Future Works **24**

G Additional Qualitative Results **25**

A Theoretical Diffusion-Based Interpretation of TopoEnhance

In this section, we provide a theoretical diffusion-based interpretation of TopoEnhance (Sections 3.2 and 3.3). We first show that the normalized relation-aware BCE loss (Equation 7–Equation 8) is a normalized Bernoulli negative log-likelihood (Lemma 1). We then show that this denoising likelihood corresponds to the learnable reconstruction term of a $T{=}1$ diffusion ELBO (Proposition 1). Together, these results justify interpreting TopoEnhance as a single-step diffusion-based denoising model on topology graphs.

Setup. Let $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ denote a clean ground-truth topology graph drawn from $p_{\mathrm{data}}(\mathcal{G})$. The forward process (Section 3.2) defines a single-step corruption distribution $q(\mathcal{G}_{\mathrm{noise}} \mid \mathcal{G}^*)$ that stochastically transforms $\mathcal{G}^*$ into a noisy graph $\mathcal{G}_{\mathrm{noise}} = (\mathcal{V}_{\mathrm{noise}}, \mathcal{E}_{\mathrm{noise}})$ by applying constrained Gaussian jitter to true-positive detections and synthesizing false-positive detections. The reverse process f_θ learns the conditional denoising distribution $p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\mathrm{noise}})$, supervised via the normalized BCE objective (Equation 7–Equation 8). We further assume that (i) the per-edge model output $\tilde{s}_{uv} \in (0, 1)$ defines a valid Bernoulli probability, and (ii) candidate edges are conditionally independent given $\mathcal{G}_{\mathrm{noise}}$.

Lemma 1. *The training objective $\mathcal{L}$ of TopoEnhance (Equation 7–Equation 8) is a relation-balanced, volume-normalized Bernoulli negative log-likelihood over the candidate edge set $\mathcal{E}_{\mathrm{noise}}$.*

Proof. We define a correspondence map $\pi : \mathcal{V}_{\mathrm{noise}} \to \mathcal{V}^* \cup \{\varnothing\}$, where $\pi(v) \in \mathcal{V}^*$ if $v \in \mathcal{V}_{\mathrm{tp}}$ is a jittered true-positive detection, and $\pi(v) = \varnothing$ if $v \in \mathcal{V}_{\mathrm{fp}}$ is a synthesized false-positive detection. The ground-truth edge label is then

$$y_{uv} = \mathbb{I}[\pi(u) \neq \varnothing,\ \pi(v) \neq \varnothing,\ (\pi(u), \pi(v)) \in \mathcal{E}^*]\,, \tag{15}$$

consistent with Equation 2.

Under the conditional independence assumption stated in the Setup, the reverse distribution factorizes as in Equation 6:

$$p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}}) = \prod_{(u,v)\in\mathcal{E}_{\text{noise}}} \text{Bernoulli}(y_{uv}; \tilde{s}_{uv}). \tag{16}$$

Using the Bernoulli probability mass function,

$$\text{Bernoulli}(y_{uv}; \tilde{s}_{uv}) = \tilde{s}_{uv}^{y_{uv}}(1-\tilde{s}_{uv})^{1-y_{uv}}, \tag{17}$$

we obtain

$$p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}}) = \prod_{(u,v)\in\mathcal{E}_{\text{noise}}} \tilde{s}_{uv}^{y_{uv}}(1-\tilde{s}_{uv})^{1-y_{uv}}. \tag{18}$$

Taking the negative logarithm yields the Bernoulli negative log-likelihood, i.e., binary cross-entropy over the candidate edge set:

$$-\log p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}}) = -\sum_{(u,v)\in\mathcal{E}_{\text{noise}}} \left[y_{uv}\log\tilde{s}_{uv} + (1-y_{uv})\log(1-\tilde{s}_{uv})\right]. \tag{19}$$

Partitioning Equation 19 by relation type $r \in \{ll, lt\}$ and normalizing each partition by its cardinality $|\mathcal{E}_r|$ yields exactly $\mathcal{L}^{(r)}$ from Equation 7. Therefore, $\mathcal{L}$ is obtained from the Bernoulli negative log-likelihood in Equation 19 by a normalization of its per-relation terms, and can be interpreted as a relation-balanced training objective for the conditional denoising likelihood $p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})$. □

Proposition 1. *Let $\mathcal{G}^* \sim p_{\text{data}}(\mathcal{G})$ be a clean topology graph and let $q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*)$ be the single-step corruption process defined above, with a fixed prior $p(\mathcal{G}_{\text{noise}})$ independent of θ. Then minimizing the denoising objective*

$$\mathcal{L}_{\text{denoise}} = -\,\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}},\, \mathcal{G}_{\text{noise}}\sim q(\mathcal{G}_{\text{noise}}|\mathcal{G}^*)}[\log p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})] \tag{20}$$

over θ is equivalent, up to an additive constant independent of θ, to minimizing the learnable reconstruction term of the T=1 diffusion ELBO.

Proof. We assume $p(\mathcal{G}_{\text{noise}})$ is fixed and independent of θ, mirroring the standard diffusion setting in which the terminal prior is pre-specified.

A T-step diffusion model defines a forward Markov chain

$$q(\mathcal{G}_{1:T} \mid \mathcal{G}^*) = \prod_{t=1}^{T} q(\mathcal{G}_t \mid \mathcal{G}_{t-1}), \tag{21}$$

gradually corrupting $\mathcal{G}^* \equiv \mathcal{G}_0$ into $\mathcal{G}_1, \ldots, \mathcal{G}_T$. The learned reverse process is

$$p_\theta(\mathcal{G}^*, \mathcal{G}_{1:T}) = p(\mathcal{G}_T)\prod_{t=1}^{T} p_\theta(\mathcal{G}_{t-1} \mid \mathcal{G}_t). \tag{22}$$

Applying Jensen's inequality to the marginal log-likelihood yields the ELBO:

$$\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}}}[-\log p_\theta(\mathcal{G}^*)] \le \mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}},\, q(\mathcal{G}_{1:T}|\mathcal{G}^*)}\left[-\log\frac{p_\theta(\mathcal{G}^*, \mathcal{G}_{1:T})}{q(\mathcal{G}_{1:T} \mid \mathcal{G}^*)}\right]. \tag{23}$$

Expanding via the Markov factorizations and applying Bayes' rule to rewrite each forward transition $q(\mathcal{G}_t \mid \mathcal{G}_{t-1})$ in terms of the posterior $q(\mathcal{G}_{t-1} \mid \mathcal{G}_t, \mathcal{G}^*)$ gives

$$\begin{aligned}\text{ELBO} = \mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}}}\Bigg[&\underbrace{\mathbb{E}_{q(\mathcal{G}_1|\mathcal{G}^*)}[-\log p_\theta(\mathcal{G}^* \mid \mathcal{G}_1)]}_{\mathcal{L}_{\text{recon}}} + D_{\text{KL}}(q(\mathcal{G}_T \mid \mathcal{G}^*)\,\|\,p(\mathcal{G}_T)) \\ &+ \sum_{t=2}^{T}\mathbb{E}_{q(\mathcal{G}_t|\mathcal{G}^*)}[D_{\text{KL}}(q(\mathcal{G}_{t-1} \mid \mathcal{G}_t, \mathcal{G}^*)\,\|\,p_\theta(\mathcal{G}_{t-1} \mid \mathcal{G}_t))]\Bigg].\end{aligned} \tag{24}$$

Setting $T{=}1$ and identifying $\mathcal{G}_1 \equiv \mathcal{G}_{\text{noise}}$, the intermediate sum vanishes and the ELBO reduces to

$$\begin{aligned}\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}}}[-\log p_\theta(\mathcal{G}^*)] \le & \underbrace{\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}},\, \mathcal{G}_{\text{noise}}\sim q(\mathcal{G}_{\text{noise}}|\mathcal{G}^*)}[-\log p_\theta(\mathcal{G}^* \mid \mathcal{G}_{\text{noise}})]}_{\mathcal{L}_{\text{recon}}} \\ & + \underbrace{\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}}}[D_{\text{KL}}(q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*) \,\|\, p(\mathcal{G}_{\text{noise}}))]}_{\text{constant w.r.t. } \theta}.\end{aligned} \tag{25}$$

Since the KL term in Equation 25 depends only on the fixed forward process and the fixed prior,

$$\frac{\partial}{\partial\theta}\mathbb{E}_{\mathcal{G}^*\sim p_{\text{data}}}[D_{\text{KL}}(q(\mathcal{G}_{\text{noise}} \mid \mathcal{G}^*) \,\|\, p(\mathcal{G}_{\text{noise}}))] = 0. \tag{26}$$

Minimizing the $T{=}1$ ELBO over θ therefore reduces to minimizing $\mathcal{L}_{\text{recon}}$. By the tower property, the nested expectation $\mathbb{E}_{\mathcal{G}^*}\big[\mathbb{E}_{q(\mathcal{G}_{\text{noise}}|\mathcal{G}^*)}[\,\cdot\,]\big]$ equals the joint expectation in Equation 20, so $\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{denoise}}$. □

Combining Lemma 1 with Proposition 1, the normalized relation-aware BCE objective used by TopoEnhance is a relation-balanced implementation of the learnable reconstruction term in a $T{=}1$ diffusion ELBO. Thus, TopoEnhance can be interpreted as optimizing a single-step diffusion-based denoising objective on topology graphs.

B Algorithmic Summary of Forward Process: Topology Noise Simulation

The algorithm of the topology noise simulation is presented at Algorithm 1.

Algorithm 1 Forward Process: Topology Noise Simulation

Require: Ground-truth $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$, thresholds δ_{type}, training step t
Ensure: Noisy graph $\mathcal{G}_{\text{noise}}$, labels $\mathcal{Y}$
1: Update $\sigma_{tp}(t)$, $\sigma_{fp}(t)$ via adaptive noise curriculum
2: $\mathcal{V}_{tp} \leftarrow \{v + \epsilon \mid v \in \mathcal{V}^*,\, \epsilon \sim \mathcal{N}(0, \sigma_{tp}^2\mathbf{I}),\, \text{dist}(v, v') \le \delta_{type}\}$
3: $\mathcal{V}_{fp} \leftarrow \{v + \epsilon \mid v \in \mathcal{V}^*,\, \epsilon \sim \mathcal{N}(0, \sigma_{fp}^2\mathbf{I}),\, \text{dist}(v, v_{fp}) > \delta_{type}\}$
4: $\mathcal{V}_{\text{noise}} \leftarrow \mathcal{V}_{tp} \cup \mathcal{V}_{fp}$ {Scaled to match $\mathbb{E}[|\hat{\mathcal{V}}|]$}
5: $\mathcal{E}_{\text{noise}} \leftarrow \mathcal{V}_{\text{noise}} \times \mathcal{V}_{\text{noise}}$ {Candidate edge space}
6: $y_{uv} \leftarrow \mathbb{I}[(u, v) \in \mathcal{E}^*]$ **for all** $(u, v) \in \mathcal{E}_{\text{noise}}$
7: **return** $\mathcal{G}_{\text{noise}}$, $\mathcal{Y} = \{y_{uv}\}$

C Implementation Details

For topology noise simulation, the dynamic noise scheduler is initialized with $\sigma_{tp} = 1\text{e}{-5}$ and $\sigma_{fp} = 10$. Following the OpenLane-V2 protocol, we set $\delta_l = 3.0$ and $\delta_t = 0.75$ as the distance thresholds for determining true-positive lane and traffic-element detections, respectively. To encode both semantic and geometric context, traffic-element node features $\mathbf{X}_t$ are initialized using 1024-dimensional DinoV2-ViT-L embeddings (Oquab et al., 2024) extracted from front-view images. Lane and traffic features $(\mathbf{X}_l, \mathbf{X}_t)$ are then projected into a shared latent space via separate MLP heads, followed by a two-layer GATv2 encoder (Brody et al., 2022) implemented in `PyG` (Fey & Lenssen, 2019) with a hidden dimension of 64, 8 attention heads, ReLU activation, and a dropout rate of 0.1. The model is trained with a batch size of 64 using the AdamW optimizer (Loshchilov & Hutter, 2017b), with an initial learning rate of 0.001, weight decay of 0.01, and a CosineAnnealingLR scheduler (Loshchilov & Hutter, 2017a) that decays the learning rate to 10^{-4}. During inference, the calibration fusion weight is fixed at 0.6 for lane–traffic relations and varies between 0.6 and 0.9 for lane–lane relations. To derive discrete topology connections from continuous topology scores, we follow the common practice in binary classification and use a threshold of $\tau = 0.5$, indicating that a connection is predicted when we are more than 50% confidence to its existence. All experiments are conducted on a single NVIDIA H200 GPU. TopoEnhance is a lightweight refinement module, requiring approximately 16 hours in total for training and just 0.1 ms per sample at inference, enabling efficient topology enhancement.

Table 3: Evaluation on the geographically non-overlapping Argoverse2 GeoSplits. Both TopoNet and TopoEnhance are trained on Argoverse2 GeoSplits.

Method	**TOP**$_{ll}$	**TOP**$_{lt}$	**TJS**$_{ll}$	**TJS**$_{lt}$
TopoNet	3.34	14.40	11.84	20.49
+ TopoEnhance	**10.23**	**16.37**	**12.87**	**35.49**
Upper-Bound	11.47	18.42	16.90	37.37

D Additional Evaluation Results

In this section, we provide additional experiments to further evaluate TopoEnhance from more comprehensive perspectives. We first evaluate TopoEnhance on geographically non-overlapping splits to examine whether its improvements rely on dataset-specific geographic biases (see Appendix D.1). We then extend our evaluation to TopoPoint (Fu et al., 2026), a recent strong topology reasoning method in Appendix D.2. Additionally, in Appendix D.3, although TopoEnhance does not modify or refine the upstream detection results, we provide the complete OpenLane-V2 evaluation metrics, including detection metrics (DET) and the overall OpenLane-V2 score (OLS), as a complement to the topology results reported in the main paper. Finally, the ablation studies on different node feature extractor, loss function, and the geometric localization jitter are presented in Appendix D.4.

D.1 TopoEnhance on Geographically Non-Overlapping Splits

Geographic overlap between training and validation data can lead to data leakage and inflate the performance of online map construction models. To evaluate whether the improvements of TopoEnhance depend on such geographic biases, we conduct an additional experiment using the geographically non-overlapping Argoverse2 GeoSplits introduced by Lilja et al. (2024). Since none of the baseline methods evaluated in the main paper provide checkpoints trained on these splits, we train TopoNet from scratch on the Argoverse2 GeoSplits. To further rule out the possibility that TopoEnhance itself memorizes geographic patterns, we also retrain TopoEnhance on the same geographically non-overlapping split.

As shown in Table 3, TopoEnhance continues to move the topology predictions substantially closer to the Upper-Bound under geographically non-overlapping evaluation. In particular, TOP$_{ll}$ improves from 3.34 to 10.23, while TJS$_{lt}$ increases from 20.49 to 35.49. These results indicate that the improvements of TopoEnhance do not rely on memorizing dataset-specific geographic patterns. Instead, the enhancement model can learn transferable structural patterns for topology refinement even when geographic overlap between training and validation data is removed.

D.2 Enhancing TopoPoint Predictions

The experiments presented in the main paper evaluate methods with officially released checkpoints, since applying TopoEnhance requires sample-level lane detections, traffic-element detections, and pairwise topology confidence matrices rather than only the aggregate metrics reported in the corresponding papers. TopoPoint (Fu et al., 2026) does not provide pretrained checkpoints for both OpenLane-V2 subsets. Nevertheless, because TopoPoint is a recent strong topology reasoning method, we additionally train it on both Argoverse2 (Subset_A) and nuScenes (Subset_B) using its official implementation[1] and evaluate TopoEnhance on the resulting predictions.

On Argoverse2, our reproduction closely matches the topology performance reported by TopoPoint, obtaining 28.10 versus 28.70 on TOP$_{ll}$ and 30.36 versus 30.00 on TOP$_{lt}$. Applying TopoEnhance improves all four topology metrics. In particular, TOP$_{lt}$ increases from 30.36 to 32.28, TJS$_{ll}$ from 22.31 to 25.03, and TJS$_{lt}$ from 35.94 to 55.13. This demonstrates that TopoEnhance remains complementary to TopoPoint's endpoint-aware topology reasoning architecture and able to further outperform the current state-of-the-art method.

[1] https://github.com/Franpin/TopoPoint

Table 4: Additional evaluation of TopoEnhance on TopoPoint. "Reported" denotes the results reported in the TopoPoint paper, while "our reproduction" uses checkpoints trained by us with the official implementation. Since TJS is not reported by TopoPoint, the corresponding entries are omitted.

Dataset	Method	**DET$_l$**	**DET$_t$**	**TOP$_{ll}$**	**TOP$_{lt}$**	**TJS$_{ll}$**	**TJS$_{lt}$**
Argoverse2 (Subset_A)	TopoPoint, reported	31.40	55.30	28.70	30.00	–	–
	TopoPoint, our reproduction	28.82	54.20	28.10	30.36	22.31	35.94
	+ TopoEnhance	28.82	54.20	**28.21**	**32.28**	**25.03**	**55.13**
	Upper-Bound	28.82	54.20	29.65	36.58	31.66	56.67
nuScenes (Subset_B)	TopoPoint, reported	31.20	60.20	28.30	27.10	–	–
	TopoPoint, our reproduction	23.67	60.66	24.73	22.42	14.09	35.13
	+ TopoEnhance	23.67	60.66	**25.00**	**23.78**	**15.03**	**50.52**
	Upper-Bound	23.67	60.66	26.73	27.08	22.52	54.25

Table 5: Complete OpenLane-V2 evaluation corresponding to Table 1. DET$_l$ and DET$_t$ measure lane and traffic-element detection, respectively. Since TopoEnhance does not modify detections, these metrics remain unchanged. OLS is computed from the displayed detection and topology metrics following the standard OpenLane-V2 evaluation protocol.

Dataset	Method	**Detection**		**Before TopoEnhance**			**After TopoEnhance**			**Upper-Bound**		
		DET$_l$ ↑	DET$_t$ ↑	TOP$_{ll}$ ↑	TOP$_{lt}$ ↑	OLS ↑	TOP$_{ll}$ ↑	TOP$_{lt}$ ↑	OLS ↑	TOP$_{ll}$ ↑	TOP$_{lt}$ ↑	OLS ↑
Argoverse2 (Subset_A)	TopoNet (Li et al., 2026)	28.6	48.6	10.9	23.8	39.8	**21.8**	**25.8**	**43.7**	24.4	28.8	45.1
	TopoMLP (Wu et al., 2024)	28.5	49.5	21.6	26.9	44.1	**24.4**	**28.7**	**45.2**	25.0	31.4	46.0
	Topo2D (Li et al., 2024)	29.1	50.6	22.3	26.2	44.5	**24.3**	**27.3**	**45.3**	25.0	29.8	46.1
	TopoLogic (Fu et al., 2024)	29.9	47.2	23.9	25.4	44.1	**24.5**	**27.2**	**44.7**	26.0	30.6	45.9
	SMART (Ye et al., 2025)	46.6	47.7	37.0	33.0	53.1	**40.1**	**35.4**	**54.3**	40.9	38.9	55.2
nuScenes (Subset_B)	TopoNet (Li et al., 2026)	24.3	55.0	6.7	16.7	36.5	**19.5**	**17.8**	**41.4**	21.2	20.3	42.6
	TopoMLP (Wu et al., 2024)	26.6	58.3	21.0	19.8	43.8	**22.8**	**21.5**	**44.8**	23.5	24.1	45.6

We also attempted to reproduce TopoPoint on nuScenes (Subset_B). Unfortunately, the released repository does not provide a Subset_B configuration, and the corresponding code path contains dataset-specific inconsistencies in lane dimensionality and image processing. After implementing the necessary fixes, our reproduced traffic-element detection performance closely matches the reported result (DET$_t$ = 60.66 versus 60.20), while lane detection remains lower (DET$_l$ = 23.67 versus 31.20). We therefore report these results explicitly as our reproduction rather than an exact reproduction of the published Subset_B result. Despite the weaker reproduced lane detections, TopoEnhance still improves TOP$_{ll}$, TOP$_{lt}$, TJS$_{ll}$, and TJS$_{lt}$, with TJS$_{lt}$ increasing from 35.13 to 50.52.

D.3 Complete OpenLane-V2 Metrics

Table 5 complements Table 1 in the main paper by reporting the complete OpenLane-V2 metric suite, including DET$_l$, DET$_t$, and OLS in addition to TOP$_{ll}$ and TOP$_{lt}$. Since TopoEnhance only refines topology connections and does not modify the upstream map-element detections, DET$_l$ and DET$_t$ remain identical before and after enhancement. We therefore report each detection metric once for every baseline.

OLS jointly evaluates detection and topology reasoning under the standard OpenLane-V2 protocol. We recompute OLS using the same detection results and the topology scores reported in Table 1, ensuring that the overall scores are consistent with our OpenLane-V2 v2.1.0 evaluation. Because the detection terms remain fixed, any improvement in OLS after applying TopoEnhance is entirely attributable to improved topology reasoning.

As shown in Table 5, improving topology reasoning also consistently improves the overall OpenLane-V2 score despite leaving the detection outputs unchanged. For example, TopoEnhance increases OLS from 36.5 to 41.4 for TopoNet on nuScenes and from 53.1 to 54.3 for SMART on Argoverse2. Moreover, the enhanced OLS values consistently move toward their corresponding detection-conditioned Upper-Bounds. These results further demonstrate that existing detection outputs contain substantial unrealized potential and that improving connectivity alone can translate into meaningful gains in overall driving-scene topology performance.

D.4 Ablation Studies

To evaluate the effectiveness of our core component, we conduct ablation studies to examine the impact of different visual feature extractors, loss functions and the geometric localization jitter component from topology noise simulation strategy.

Table 6: Ablation studies on different visual backbones for traffic-element node feature extraction. All experiments are conducted using TopoNet as the baseline on Argoverse2 (Subset_A).

Backbone Model	Dim	**TOP**$_{ll}$	**TOP**$_{lt}$
DINOv2-ViT-S	384	20.3	24.64
DINOv2-ViT-B	768	21.8	25.06
DINOv2-ViT-L	1024	**21.84**	**25.8**
DINOv2-ViT-G	1536	**21.84**	25.75
DINOv3-ViT-S	384	21.56	24.48
DINOv3-ViT-B	768	21.83	24.35
DINOv3-ViT-L	1024	**21.83**	24.49
ResNet-50	256	**21.8**	25.5

Traffic-Element Node Feature Extractor. We investigate the impact of various visual backbones for extracting traffic-element (TE) node features, as the semantic richness of these embeddings is critical for establishing lane–traffic associations. We compare several variants of DINOv2 and DINOv3 across different parameter scales (Small, Base, Large, and Giant), as well as a standard ResNet-50 baseline. As shown in Table 6, the performance on lane–lane topology (TOP$_{ll}$) remains relatively stable across most high-capacity backbones, suggesting that lane-to-lane connectivity is primarily driven by geometric and structural priors rather than deep visual semantics of traffic signs. However, lane–traffic topology (TOP$_{lt}$) exhibits higher sensitivity to the choice of feature extractor. While DINOv3 variants provide competitive results, the DINOv2-ViT-L backbone achieves the highest overall performance (TOP$_{lt}$ of 25.8). Interestingly, increasing the model capacity further to DINOv2-ViT-G does not yield additional gains, indicating a point of diminishing returns where the feature dimensionality of 1536 may begin to overfit the relatively sparse traffic-element distribution. At the same time, the performance gap between ResNet-50 and DINOv2 ViT Large is only 0.2. This demonstrates that our method is largely backbone agnostic and remains effective even with lightweight feature extractors. Based on these results, we adopt DINOv2-ViT-L as our default TE feature extractor to balance semantic depth with computational efficiency. It is also worth noting that TopoNet uses ResNet 50 as its visual backbone. Even under the same backbone setting, our method significantly outperforms TopoNet by 11.04 on TOP$_{ll}$ and 1.7 on TOP$_{lt}$, as shown in Table 6 and Table 1. Overall, these results suggest that while stronger backbones provide marginal gains, the main performance improvement comes from our noise driven training strategy and refinement design, making the method robust and broadly applicable across different feature sources.

Table 7: Ablation studies on loss functions. This experiment is conducted to enhance TopoNet's predictions on Argoverse2 (Subset_A) under a consistent training configuration.

Loss	**TOP**$_{ll}$	**TOP**$_{lt}$
BCE	12.1	22.0
BCE + Focal	**22.2**	22.0
Normalized BCE (Ours)	21.8	**25.8**

Loss Function. We compare our normalized BCE loss against standard Binary Cross-Entropy (BCE) and a hybrid BCE–Focal variant. As shown in Table 7, standard BCE performs poorly due to the extreme sparsity and structural imbalance of the topology graph. In standard formulations, the total loss is dominated by the more populous lane–lane candidate space, causing the model to neglect the sparser but equally critical lane–traffic associations. While the hybrid BCE–Focal loss successfully stabilizes lane–lane topology (TOP$_{ll}$), it lacks a mechanism to balance the relative importance between heterogeneous relationship types, leading to suboptimal performance in traffic-element associations. In contrast, our normalized BCE achieves the most balanced overall performance

by introducing relation-aware scaling. By normalizing the loss of each category by its specific candidate edge volume, we ensure that gradient updates are not overwhelmed by the density of the lane-lane relationships. This balancing mechanism ensures that the enhancer effectively captures the subtle structural dependencies of both relation types, preventing the optimization process from being biased toward the high-frequency connectivity patterns. By automatically adapting to varying connectivity densities across different scene geometries, this approach minimizes the reliance on sensitive, manually tuned weighting hyperparameters, making it a more robust choice for a universal topology enhancer.

Table 8: Ablation studies on geometric localization jitter, while keeping other components from topology noise simulation unchanged. This experiment is conducted to enhance TopoNet's predictions on Argoverse2 (Subset_A) under a consistent training configuration.

Strategy	**TOP**$_{ll}$	**TOP**$_{lt}$
TopoEnhance	**21.8**	**25.8**
TopoEnhance without Jitter	21.7	25.4

Geometric Localization Jitter. Although our primary objective is to rectify structural connectivity, the geometric precision of map elements significantly influences the learning process. In our framework, geometric localization jitter serves as a crucial regularizer. By introducing constrained, semantics-preserving perturbations to the node coordinates of ground-truth detections ($v \in \mathcal{V}^*$), we prevent the model from overfitting to idealized geometries. This jittering forces the heterogeneous GNN to prioritize invariant structural patterns and relational context over exact coordinate matching, which is essential for generalizing to the noisy predictions encountered during real-world inference. We emphasize that Table 8 isolates the effect of the geometric jitter component, rather than ablating the entire noise simulation strategy. Since the full framework already incorporates adaptive noise curriculum and false positive detection synthesis to simulate realistic perception errors, the additional gains from geometric jitter are expected and reflect its complementary role. Nevertheless, we observe consistent improvements across both metrics, including a notable +0.4 gain in TOP$_{lt}$. This indicates that geometric jitter effectively complements the modeling of structural inconsistencies by introducing realistic localization noise. As a result, the TopoEnhance becomes more robust to the coordinate inaccuracies commonly present in diverse perception systems, without compromising the learned topological priors.

E Practical Usability Toward Real-World Applications

In this section, we provide additional experiments to further evaluate the practical usability of TopoEnhance from multiple perspectives. We first analyze the robustness of TopoEnhance under varying levels of topology corruption in Appendix E.1 to examine whether it remains effective when the quality of the input topology varies substantially. We then evaluate temporal consistency across consecutive frames in Appendix E.2 to study whether the refined topology predictions remain consistent over time. Additionally, in Appendix E.3, we analyze safety-relevant topology errors to examine whether TopoEnhance corrects connections that are directly related to driving behaviors. We further evaluate scene-level ego-route recoverability in Appendix E.4 to assess whether the refined topology graphs better preserve complete driving routes for downstream navigation and planning. Finally, the computational efficiency of TopoEnhance, including inference latency and memory overhead, is presented in Appendix E.5.

E.1 Robustness to Varying Levels of Topology Noise

In practical autonomous driving systems, the quality of topology predictions can vary substantially across upstream models and driving scenes. A topology enhancement method should therefore remain effective not only for high-quality predictions, but also when the input connectivity becomes increasingly unreliable. Evaluating robustness under different levels of topology noise is particularly important for TopoEnhance because it is designed as a source-agnostic enhancement module that can be applied to predictions from different upstream methods. A controlled corruption experiment provides a systematic way to stress-test this capability by gradually degrading the input topology and measuring how much connectivity can still be recovered.

Table 9: Robustness analysis under varying levels of topology noise on Argoverse2 (Subset_A).

Model	Noise	TOP_{ll}		TOP_{lt}	
		Before TopoEnhance	**After TopoEnhance**	**Before TopoEnhance**	**After TopoEnhance**
TopoNet	$p = 0.0$	10.90	**21.80**	23.80	**25.80**
	$p = 0.1$	4.62	**9.99**	15.81	**17.11**
	$p = 0.2$	2.78	**6.14**	11.31	**12.35**
	$p = 0.3$	1.99	**4.39**	8.48	**9.39**
	$p = 0.5$	1.30	**2.77**	4.94	**5.64**
	$p = 0.7$	0.99	**2.02**	2.93	**3.52**
	$p = 0.9$	0.82	**1.59**	1.85	**2.32**
SMART	$p = 0.0$	37.00	**40.10**	33.00	**35.40**
	$p = 0.1$	12.65	**15.03**	21.84	**23.58**
	$p = 0.2$	6.59	**8.95**	15.61	**17.00**
	$p = 0.3$	4.01	**6.26**	11.46	**12.68**
	$p = 0.5$	1.81	**3.89**	6.49	**7.41**
	$p = 0.7$	0.99	**2.78**	3.89	**4.61**
	$p = 0.9$	0.33	**2.20**	2.20	**2.81**

To directly evaluate the denoising capability of TopoEnhance under different levels of input topology quality, we conduct a controlled corruption experiment on the Argoverse2 (Subset_A) validation set. For each predicted topology score $\hat{s}_{uv}$, we independently replace it with $1 - \hat{s}_{uv}$ with probability p. We evaluate $p \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$ on both TopoNet and SMART, which have substantially different initial topology performance. The same trained TopoEnhance is applied at every corruption level without retraining or adaptation. Here, $p = 0$ corresponds to the uncorrupted predictions reported in the main paper.

As shown in Table 9, although $p = 0.1$ corrupts only a small fraction of topology scores, it already causes a substantial decrease in TOP. This nonlinear degradation is expected because TOP is based on mean Average Precision (mAP). Replacing a score $\hat{s}_{uv}$ with $1 - \hat{s}_{uv}$ can move confident false-positive connections toward the top of the ranking while pushing true-positive connections downward. Since Average Precision is particularly sensitive to errors near the top of the ranked list, even a small amount of corruption can cause a large performance drop. As the ranking becomes increasingly disrupted, additional corruption has a smaller marginal effect. For $p \geq 0.1$, increasing p progressively degrades the input topology quality for both baselines and both relation types.

Nevertheless, TopoEnhance consistently improves TOP_{ll} and TOP_{lt} across all evaluated settings. For TopoNet, the improvement in TOP_{ll} ranges from 0.77 to 10.90, while the improvement in TOP_{lt} ranges from 0.47 to 2.00. For SMART, TopoEnhance improves TOP_{ll} by 1.79 to 3.10 and TOP_{lt} by 0.61 to 2.40 across the full range of input quality. Even at the most severe corruption level, $p = 0.9$, TopoEnhance improves SMART from 0.33 to 2.20 on TOP_{ll} and from 2.20 to 2.81 on TOP_{lt}.

These results show that TopoEnhance remains effective as the quality of the input topology varies substantially. Its consistent recovery of performance under controlled score corruption provides direct empirical evidence of its topology-denoising capability. Specifically, TopoEnhance can recover plausible connectivity from degraded topology predictions by leveraging geometric, semantic, and heterogeneous structural context, demonstrating that its denoising behavior remains effective across a broad range of input topology quality.

E.2 Temporal Consistency of Refined Topology Predictions

In autonomous driving, topology predictions should remain stable across consecutive frames when the underlying scene structure does not change. Frequent changes in predicted connectivity or large fluctuations in topology scores can lead to inconsistent scene understanding and potentially affect downstream decision-making. We therefore evaluate whether the topology refinement produced by TopoEnhance improves the temporal consistency of predicted connections across consecutive frames.

We evaluate TopoNet, TopoMLP, TopoLogic, and SMART on the Argoverse2 (Subset_A) validation set. To establish correspondence across frames, we use the persistent lane segment IDs inherited from the underlying Argoverse2 HD map. For every ground-truth lane pair that remains co-visible for at least $K = 5$ consecutive frames, we compute an *edge flip rate*, defined as the fraction of adjacent

Table 10: Temporal consistency measured by edge prediction flip rate on Argoverse2 (Subset_A). Lane correspondences across frames are established using persistent lane segment IDs, and only ground-truth lane pairs that remain co-visible for at least $K = 5$ consecutive frames are evaluated. The flip rate is the fraction of adjacent frame pairs in which the predicted discrete connectivity changes between positive and negative. Lower values indicates that the edge flip less frequently.

Model	**Before TopoEnhance**	**After TopoEnhance**
TopoNet	24.48%	**24.21%**
TopoMLP	12.62%	**7.77%**
TopoLogic	10.68%	**8.27%**
SMART	10.52%	**6.38%**

Table 11: Temporal consistency measured by the standard deviation of topology scores on Argoverse2 (Subset_A). Lane correspondences across frames are established using persistent lane segment IDs, and only ground-truth lane pairs that remain co-visible for at least $K = 5$ consecutive frames are evaluated. For each lane pair, the temporal standard deviation is computed over its topology scores across consecutive frames. Lower values indicate more stable topology confidence over time.

Model	**Before TopoEnhance**	**After TopoEnhance**
TopoNet	0.166	**0.118**
TopoMLP	0.111	**0.080**
TopoLogic	0.285	**0.228**
SMART	0.104	**0.075**

frame pairs for which the predicted edge changes between positive and negative. We additionally measure the temporal standard deviation of topology scores to quantify the variation in connection confidence across consecutive frames.

As shown in Tables 10 and 11, TopoEnhance reduces both the edge flip rate and the temporal variation of topology scores across all four baselines. The lower edge flip rate indicates that the refined discrete connections are more consistent across consecutive frames, while the lower temporal standard deviation indicates more stable topology confidence scores. Such temporal stability is practically important because downstream modules, such as trajectory prediction and path planning, rely on a consistent understanding of the scene topology over time. Reducing frame-to-frame fluctuations in predicted connectivity can help provide a more stable topology graph to these downstream modules and reduce inconsistent decisions caused by transient topology changes. Therefore, in addition to improving topology accuracy, TopoEnhance produces temporally more consistent predictions that are better suited for use in an autonomous driving pipeline.

E.3 Safety-Relevant Topology Error Analysis

Aggregated topology metrics such as TOP and TJS measure overall connectivity quality, but they do not indicate whether the corrected connections are particularly relevant to driving behavior. In practical autonomous driving, different topology errors can have very different consequences. For example, a missing successor along the ego route can interrupt a feasible driving path, while a missing lane–traffic association can prevent the topology graph from correctly identifying the traffic control governing a lane. We therefore examine whether the improvements produced by TopoEnhance extend to topology relations that are directly relevant to navigation and traffic-rule understanding. For lane–lane topology, we consider *missed ego successors*, where a connection along the ego vehicle's driven path is missing, and *missed intersection turns*, where a valid turning connection at an intersection or connector is absent. For lane–traffic topology, we evaluate missed associations with red, green, and yellow traffic lights, as well as turn signals (i.e., `no_left_turn`, `no_right_turn`, `turn_left` and `turn_right`). We evaluate only topology connections whose endpoints are successfully detected and matched, thereby isolating topology refinement from upstream detection quality. In Table 12, "Corrected" denotes a previously missing ground-truth connection recovered by

Table 12: Safety-relevant topology errors corrected and introduced by TopoEnhance on Argoverse2 (Subset_A), grouped into lane–lane and lane–traffic relations. Lane–lane errors include missed ego successors and intersection turns, while lane–traffic errors include missed associations with traffic signals. Only relations whose endpoint detections are successfully matched to ground truth are evaluated. "Corrected" denotes a previously missing ground-truth connection recovered after refinement, while "Broken" denotes an originally correct connection that becomes incorrect.

Model	Relation	Error Type	**Total**	**Corrected**	**Broken**
TopoNet	Lane–Lane	Missed ego successor	243	60	0
		Missed intersection turn	5,378	1,502	0
	Lane–Traffic	Missed red light	153	103	0
		Missed green light	398	311	0
		Missed yellow light	54	23	0
		Missed turn signal	49	36	0
TopoMLP	Lane–Lane	Missed ego successor	243	13	0
		Missed intersection turn	5,353	555	2
	Lane–Traffic	Missed red light	150	112	0
		Missed green light	385	279	0
		Missed yellow light	54	30	0
		Missed turn signal	39	25	0
TopoLogic	Lane–Lane	Missed ego successor	244	7	0
		Missed intersection turn	5,437	102	0
	Lane–Traffic	Missed red light	158	102	0
		Missed green light	391	311	0
		Missed yellow light	58	22	0
		Missed turn signal	45	30	0
SMART	Lane–Lane	Missed ego successor	250	11	0
		Missed intersection turn	5,902	489	16
	Lane–Traffic	Missed red light	150	107	0
		Missed green light	398	316	0
		Missed yellow light	63	43	0
		Missed turn signal	47	42	0

TopoEnhance, while "Broken" denotes an originally correct connection that becomes incorrect after enhancement.

As shown in Table 12, TopoEnhance recovers many driving-relevant connections while introducing very few new errors. Across the four baselines, it corrects 7–60 missed ego successors and 102–1,502 missed intersection-turn connections, while breaking at most 16 intersection-turn connections for SMART and none for the other three methods. For lane–traffic topology, TopoEnhance recovers 102–112 missed red-light associations, 279–316 missed green-light associations, 22–43 missed yellow-light associations, and 25–42 missed turn-signals associations, without introducing new errors in any of these traffic-element categories.

These results provide a more practically meaningful view of the improvements than aggregated topology metrics alone. Recovering ego-route successors and valid intersection turns helps preserve the connectivity of feasible driving paths, while recovering lane–traffic associations provides more complete information about the traffic controls and maneuver constraints governing those paths. At the same time, the very small number of newly introduced errors indicates that these gains are achieved without substantially disrupting connections that were already correct. Overall, this analysis shows that the improvements from TopoEnhance extend to topology relations that are directly relevant to how downstream autonomous driving modules interpret and use the driving-scene topology.

Table 13: Scene-level ego-route recoverability on Argoverse2 (Subset_A). Frame-level topology predictions are aggregated into a scene-level lane graph using persistent Argoverse2 lane IDs. We evaluate only routes for which the required lane detections are successfully detected and matched to ground truth. For each such scene, the ego vehicle's actual driving route is considered recoverable only if every lane transition along the complete route is correctly recovered in the predicted scene-level topology graph. Higher ego-route recoverability indicates that a larger fraction of scenes have their complete ego trajectories successfully recovered.

Model	**Before TopoEnhance**	**After TopoEnhance**
TopoNet	39.19%	**68.92%**
TopoMLP	94.81%	**100%**
TopoLogic	92.41%	**93.67%**
SMART	91.36%	**98.77%**

E.4 Scene-Level Ego-Route Recoverability for Navigation and Planning

While correcting individual topology connections improves topology quality at local frames, downstream navigation and planning require a continuous sequence of valid lane transitions along a driving route. Even if most individual connections are correct, a single erroneous transition can prevent the complete route from being recovered. We therefore evaluate scene-level ego-route recoverability to determine whether the local connectivity improvements produced by TopoEnhance translate into more complete driving routes for downstream navigation and planning.

Since OpenLane-V2 provides frame-level topology annotations, we first reconstruct a scene-level lane graph by aggregating topology predictions across frames using the persistent lane IDs inherited from the underlying Argoverse2 HD map. To isolate topology quality from upstream detection errors, we evaluate ego-route recoverability only when the lane detections required along the driven route are successfully detected and matched. We then examine whether the ego vehicle's true driving route can be recovered from the predicted scene-level road topology graph. Specifically, a route is considered recoverable only if every lane transition along the ego vehicle's complete driven trajectory, typically spanning 70–110 meters, is correctly preserved in the predicted graph. This provides a strict evaluation of route-level topology quality, since a single missing or incorrect transition can prevent the complete route from being recovered.

As shown in Table 13, TopoEnhance consistently improves full-route recoverability across all four baselines, from 39.19% to 68.92% for TopoNet, 94.81% to 100% for TopoMLP, 92.41% to 93.67% for TopoLogic, and 91.36% to 98.77% for SMART. These improvements show that the benefits of topology refinement extend beyond correcting isolated edges and lead to more complete scene-level driving paths.

This result has direct practical relevance because downstream navigation and planning modules depend on consecutive lane connectivity to identify feasible routes through a driving scene. A topology graph can achieve strong aggregate performance while still being difficult to use if a small number of missing connections interrupt an otherwise valid route. By increasing the fraction of scenes in which the complete ego route can be recovered, TopoEnhance produces topology graphs that preserve the structural information needed to support route-level reasoning more reliably. Although this experiment does not directly evaluate a downstream planner, it provides complementary evidence that the improvements from TopoEnhance extend beyond aggregate topology metrics to a structural property that is directly relevant to the practical use of topology graphs for autonomous navigation and planning.

E.5 Efficiency Analysis For Real-Time Inference

To evaluate the computational feasibility of TopoEnhance in real-time autonomous driving stacks, we provide a detailed breakdown of its inference time and memory overhead. We benchmark the inference pipeline on a representative nuScenes (Subset_B) scene using TopoNet as the base topology model. All measurements are conducted on a single NVIDIA H200 GPU. To ensure a conservative "stress-test" estimate, we evaluate a dense graph configuration containing 200 lane nodes and 100

traffic-element (TE) nodes. This results in 39,800 candidate lane–lane edges and 20,000 candidate lane–TE edges, representing the upper bound of complexity in typical urban driving scenarios.

Table 14: Latency breakdown of TopoEnhance on a representative OpenLane-V2 scene (200 lanes, 100 TEs).

Component	**Latency (ms)**	**Percentage**
Feature Extraction	0.21	3.2%
GNN Message Passing	5.21	80.9%
Score Prediction	0.64	9.9%
Total	**6.43**	**100%**

Inference Latency. Table 14 details the latency for each stage of the enhancement process. The total inference overhead of TopoEnhance is 6.43 ms, which is well within the restricted computational budgets of real-time autonomous systems. This efficiency is primarily attributed to our streamlined feature extraction stage (0.21 ms), which allows for instantaneous node embedding projection, and our optimized edge-decoder for score prediction (1.02 ms). While the GNN message-passing stage constitutes the majority of the runtime (80.9%), the 5.20 ms overhead remains remarkably low given that the model must evaluate approximately 60,000 candidate edges. This performance is sustained by the parallelized nature of the Graph Attention Network (GAT) layers, where sparse-to-dense matrix multiplications are highly optimized for GPU acceleration. By decoupling the structural enhancement from the computationally expensive backbone perception model, TopoEnhance provides a scalable solution that maintains predictable sub-10 ms latency even under dense connectivity stress tests.

Table 15: Memory overhead of each inference component.

Component	**Memory Overhead**
Feature Extraction	+0.1 MB
GNN Message Passing	+0.0 MB
Score Prediction	+0.2 MB
Total	**+0.3 MB**

Memory Overhead. Table 15 summarizes the memory overhead incurred by the enhancement module. TopoEnhance is exceptionally lightweight, requiring only +0.3 MB of additional VRAM, making it highly suitable for memory-constrained edge computing platforms. The negligible memory footprint during GNN message passing (+0.0 MB) is achieved through the use of optimized graph kernels and in-place operations that avoid the allocation of large intermediate adjacency matrices. The minor overhead in the feature extraction and score prediction stages is primarily due to the storage of node embeddings and the parameters of the heterogeneous GNN layers. Because the module operates on high-level graph abstractions rather than high-resolution feature maps, it maintains a small and constant memory profile. This ensures that TopoEnhance can be integrated as a post-hoc component without competing for limited hardware resources with the primary perception backbone.

F Limitation and Future Works

We have to admit that, relying on camera images to refine lane–traffic connectivity introduces a limitation, as it restricts the applicability of TopoEnhance to settings where camera images are available. This is a deliberate design choice because reasoning about lane–traffic connectivity requires semantic understanding of traffic elements. For example, the enhancement model must distinguish whether a traffic element corresponds to a "turn left" sign or a "go straight" sign in order to correctly refine lane–traffic connections. An alternative to camera images would be to use the semantic labels of traffic elements directly. However, such labels are typically unavailable at inference time and require an additional perception or annotation pipeline to obtain. In contrast, camera images are already available in most autonomous driving systems and naturally provide the semantic cues needed for refinement without relying on ground-truth semantic annotations. Moreover, this modality

requirement only affects lane–traffic connectivity refinement. When camera images are unavailable, TopoEnhance can still be applied to lane–lane connectivity, which does not depend on image features. Future work could explore alternative semantic representations or additional modalities to further broaden the applicability of lane–traffic refinement.

G Additional Qualitative Results

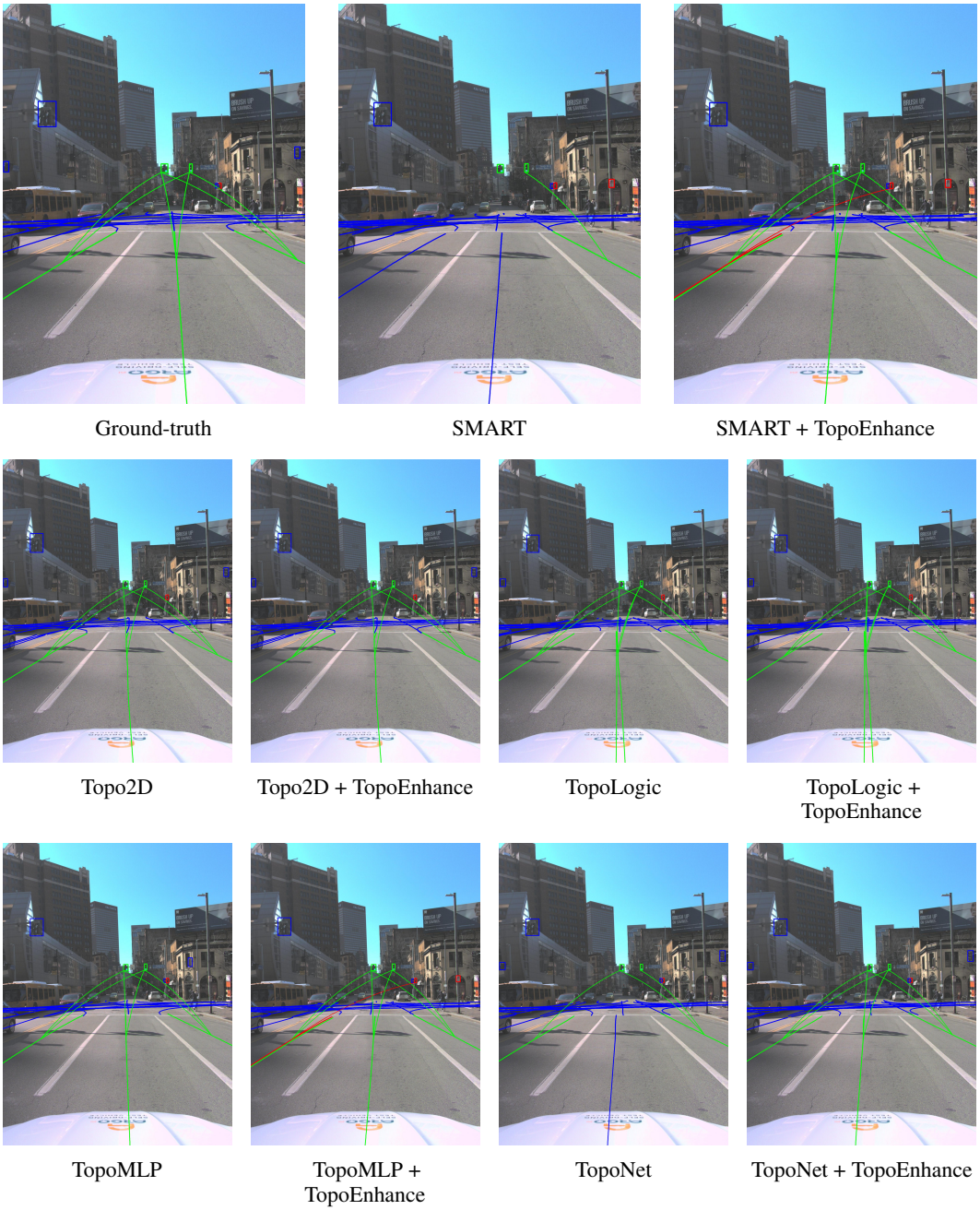
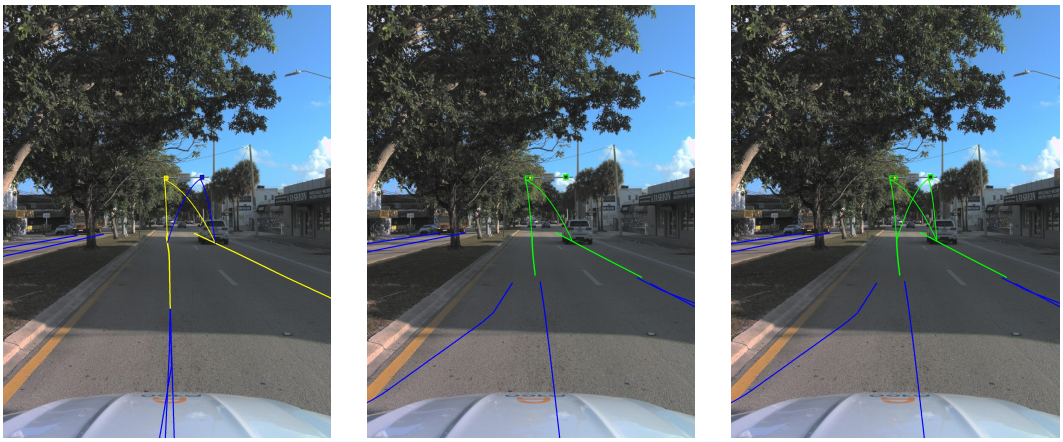
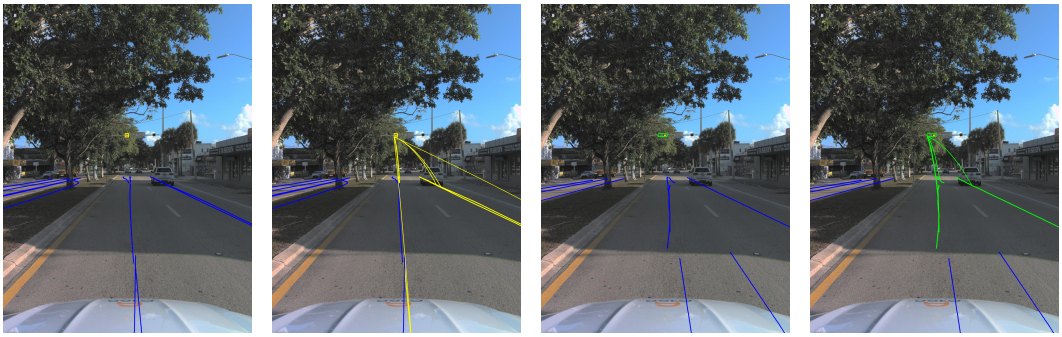


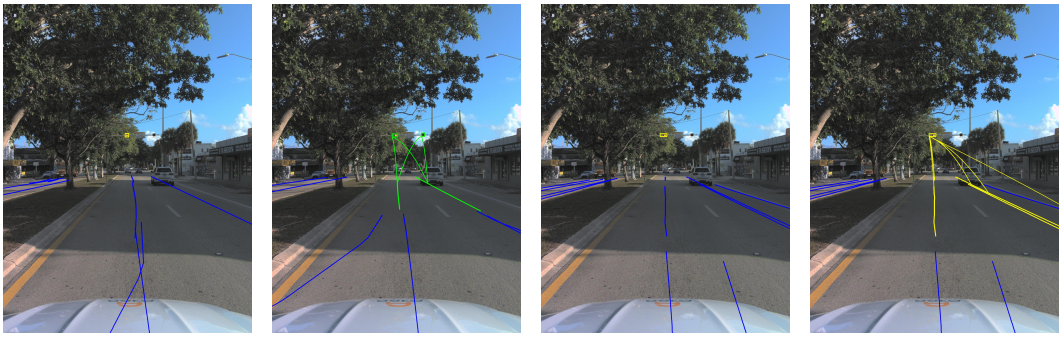
Figure 3: Qualitative results for scene 10023: predictions before and after refinement with TopoEnhance across baselines.



Ground-truth | SMART | SMART + TopoEnhance

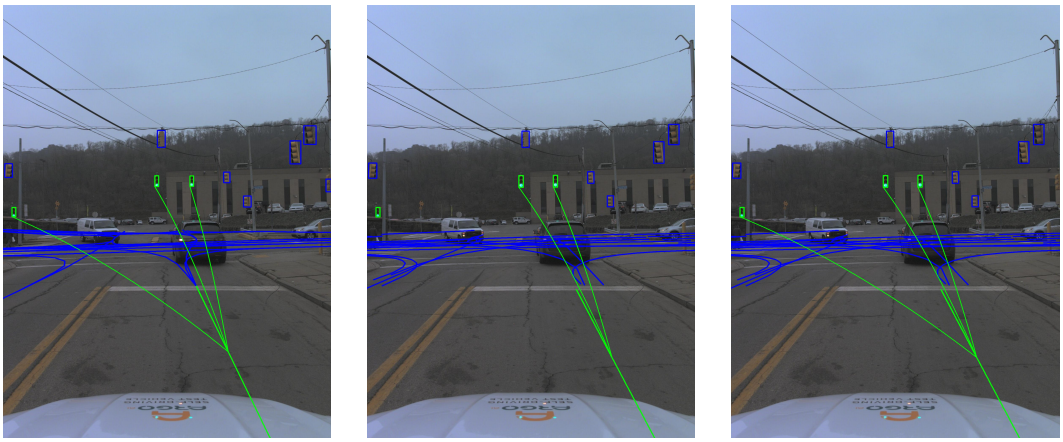


Topo2D | Topo2D + TopoEnhance | TopoLogic | TopoLogic + TopoEnhance

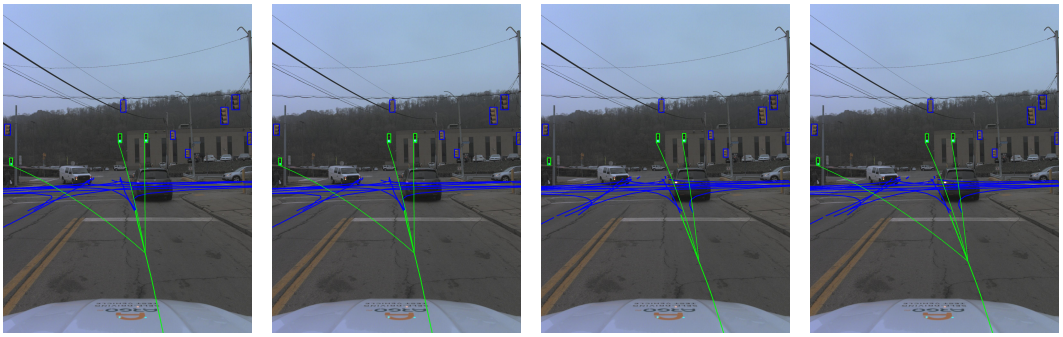


TopoMLP | TopoMLP + TopoEnhance | TopoNet | TopoNet + TopoEnhance

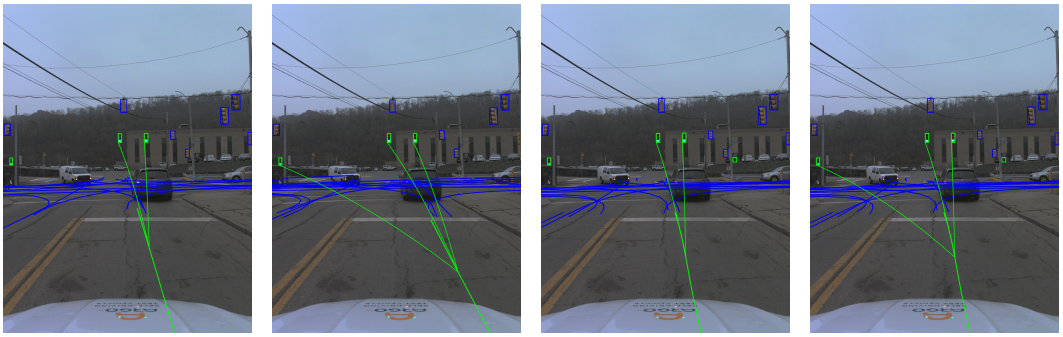
Figure 4: Qualitative results for scene 10013: predictions before and after refinement with TopoEnhance across baselines (SMART, Topo2D, TopoLogic, TopoMLP, TopoNet). All panels correspond to the same ground-truth view (left).



Ground-truth | SMART | SMART + TopoEnhance

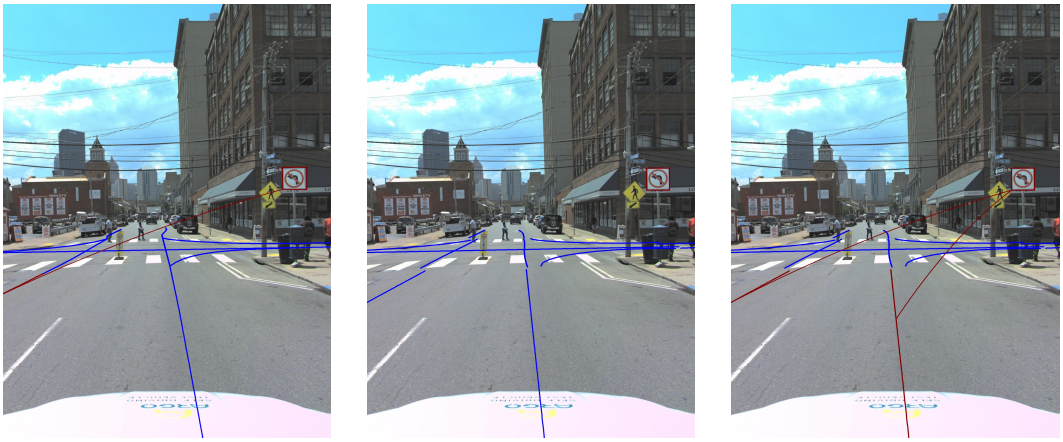


Topo2D | Topo2D + TopoEnhance | TopoLogic | TopoLogic + TopoEnhance

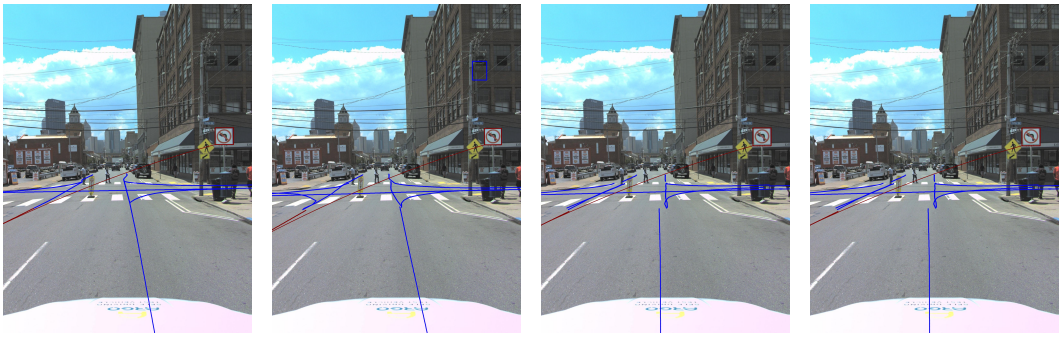


TopoMLP | TopoMLP + TopoEnhance | TopoNet | TopoNet + TopoEnhance

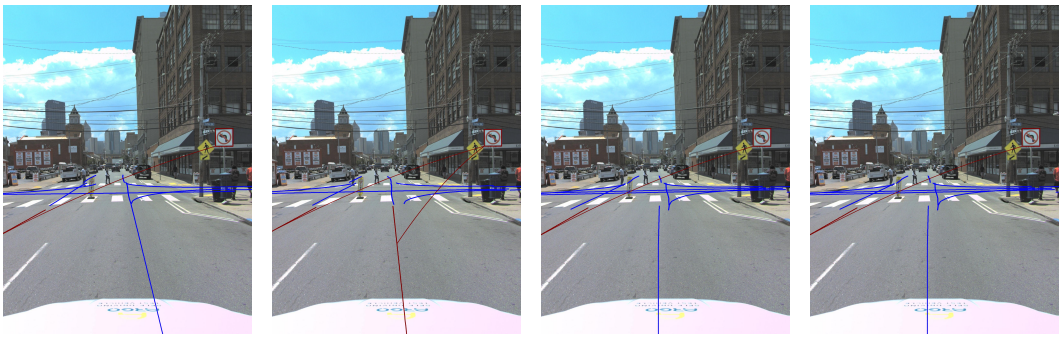
Figure 5: Qualitative results for scene 10021: predictions before and after refinement with TopoEnhance across baselines (SMART, Topo2D, TopoLogic, TopoMLP, TopoNet). All panels correspond to the same ground-truth view (left).



Ground-truth | SMART | SMART + TopoEnhance



Topo2D | Topo2D + TopoEnhance | TopoLogic | TopoLogic + TopoEnhance



TopoMLP | TopoMLP + TopoEnhance | TopoNet | TopoNet + TopoEnhance

Figure 6: Qualitative results for scene 10030: predictions before and after refinement with TopoEnhance across baselines.