A APPENDIX

A.1 LLM USAGE

In preparing this manuscript, we used a large language model (LLM) to help polish phrasing and grammar.

A.2 IMPLEMENTATION DETAILS

A.2.1 OVERALL OPTIMIZATION OBJECTIVE

As discussed in the main paper, our training objective comprises detection losses for individual elements and reasoning losses for their relationships. Following prior methods Li et al. (2023a); Wu et al. (2023); Zhu et al. (2020), traffic element detection is supervised using a combination of classification loss ($\lambda_1 = 2$), L1 loss ($\lambda_2 = 5$), and IoU loss ($\lambda_3 = 2$). Lane detection is similarly optimized with classification and regression losses. The regression component includes L1 loss on control points and a chamfer distance loss $L_{\text{BézierCD}}$ computed on sampled on-curve points, introduced in our curve-guided cross-attention. We set the number of sampled points K to 11. The corresponding weights are $\lambda_4 = 1.5$ for classification, $\lambda_5 = 0.05$ for L1 loss, and $\lambda_6 = 0.02$ for the chamfer distance loss. For topology reasoning, we adopt a focal loss for classification, as in prior work, and introduce a contrastive loss $\mathcal{L}_{\text{con}}$ to further enhance relational learning. Each contrastive pair includes one positive and three negative samples. The focal and contrastive losses are weighted by $\lambda_7 = 5$ and $\lambda_8 = 0.1$, respectively. The overall loss function is defined as:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{relation}}, \tag{11}$$

$$\mathcal{L}_{\text{det}} = \mathcal{L}^{\text{te}}_{reg} + \mathcal{L}^{\text{lane}}_{reg} \tag{12}$$

$$= (\lambda_1 \mathcal{L}^{\text{te}}_{\text{class}} + \lambda_2 \mathcal{L}^{\text{te}}_{\text{L1}} + \lambda_3 \mathcal{L}^{\text{te}}_{\text{IoU}}) + (\lambda_4 \mathcal{L}^{\text{lane}}_{\text{class}} + \lambda_5 \mathcal{L}^{\text{lane}}_{\text{L1}} + \lambda_6 \mathcal{L}_{\text{BézierCD}}), \tag{13}$$

$$\mathcal{L}_{\text{relation}} = \lambda_7 \mathcal{L}^{\text{topo}}_{\text{class}} + \lambda_8 \mathcal{L}_{\text{con}}. \tag{14}$$

A.2.2 ILLUSTRATION OF TOPOLOGY REASONING

To better illustrate the computation flow of topology prediction, we provide an algorithm for the L2L prediction pipeline (see Algorithm 1). The L2T head follows a similar structure, differing mainly in the way of building the pairwise embeddings $\mathbf{G}_{(.)}$, which can refer to the main paper Sec. 3.3.2.

Algorithm 1 Geometry-Enhanced L2L Reasoning

Require: Lane query features $\mathbf{Q}_{\text{lane}} \in \mathbb{R}^{N \times C}$, lane endpoints $\mathbf{P}_{\text{lane}} \in \mathbb{R}^{N \times 2 \times 2}$
Ensure: Connectivity logits $\mathbf{T}_{\text{L2L}} \in \mathbb{R}^{N \times N \times 1}$

$\mathbf{E}_{\text{pre}} \leftarrow \text{MLP}_1(\mathbf{Q}_{\text{lane}})$ ▷ $\in \mathbb{R}^{N \times \frac{C}{2}}$
$\mathbf{E}_{\text{suc}} \leftarrow \text{MLP}_2(\mathbf{Q}_{\text{lane}})$ ▷ $\in \mathbb{R}^{N \times \frac{C}{2}}$
$(\mathbf{PE}_{\text{pre}}, \mathbf{PE}_{\text{suc}}) \leftarrow \text{PosEnc}(P_{\text{lane}})$ ▷ sinusoidal positional encoding
$\tilde{\mathbf{E}}_{\text{pre}} \leftarrow \mathbf{E}_{\text{pre}} + \mathbf{PE}_{\text{pre}}$
$\tilde{\mathbf{E}}_{\text{suc}} \leftarrow \mathbf{E}_{\text{suc}} + \mathbf{PE}_{\text{suc}}$
$\mathbf{G}_{\text{L2L}} \leftarrow \text{BroadcastConcat}(\tilde{\mathbf{E}}_{\text{pre}}, \tilde{\mathbf{E}}_{\text{suc}})$ ▷ pairwise feature, $\in \mathbb{R}^{N \times N \times C}$
$\text{Dist}_{ij} \leftarrow \texttt{distance}(P^e_i - P^s_j)$ ▷ distance: lane i end point $\rightarrow$ lane j start point, $\in \mathbb{R}^{N \times N \times 1}$
$\text{DistEmbed} \leftarrow \text{MLP}(\text{Dist}_{ij})$ ▷ $\in \mathbb{R}^{N \times N \times C}$
$\mathbf{T}_{\text{L2L}} \leftarrow \text{MLP}(\mathbf{G}_{\text{L2L}} + \text{DistEmbed})$ ▷ $\in \mathbb{R}^{N \times N \times 1}$
return $\mathbf{T}_{\text{L2L}}$

A.3 Extra Experiments

A.3.1 More Analysis on Geometry Topology.

TopoLogic Fu et al. (2024) proposes a geometric distance topology (GDT), which shares certain similarities with our Geometry-Biased Self-Attention and Geometry-Enhanced L2L Topology, but *differs fundamentally in design and effectiveness*.

Table 4: **Experiments on Geometric Distance Topology (GDT).** "Topologic – GDT_{L2L}": Inference without GDT_{L2L}. TopoLogic†: Inference using the hyperparameters learned during their lane representation training. "Ours + GDT_{lane}": Replaces our geometry encoding with TopoLogic's GDT for lane representation learning. "Ours + GDT_{L2L}": Ensembles our L2L predictions with GDT, same as TopoLogic.

Model	DET_l	DET_t	TOP_{ll}	TOP_{lt}	OLS
Topologic	29.9	47.2	23.9	25.4	44.1
Topologic – GDT_{L2L}	29.9	47.2	11.6	25.4	40.4
Topologic†	29.9	47.2	23.2	25.4	43.9
Ours	**33.8**	**50.9**	**29.2**	**32.2**	**48.9**
Ours + GDT_{lane}	32.8	50.0	28.4	30.8	47.9
Ours + GDT_{L2L}	33.8	50.9	28.7	32.2	48.7

Lane Representation Learning. GDT Fu et al. (2024) focuses exclusively on connectivity estimation by computing the end-to-start point distance between lanes and updating representations using an additional GCN block. *In contrast*, our method captures richer spatial cues beyond connectivity, such as inter-lane distances and angular relations. These geometric priors—*e.g.*, parallelism, perpendicularity, and merging—are common in real-world road layouts and intuitively leveraged by humans. We explicitly encode the shortest endpoint distance and angular difference between lanes into high-dimensional relational features, which are further projected as biases in self-attention to enhance both perception and downstream topology reasoning.

L2L Topology Reasoning. Our method enhances geometric reasoning by leveraging the fact that annotated connected lanes share overlapping endpoints. We encode these end-to-start distances as high-dimensional embeddings and integrate them directly into the L2L relation features during training, enabling the model to learn from geometric cues in an end-to-end manner. In contrast, TopoLogic Fu et al. (2024) computes a scalar topology score for each lane pair (ranging from 0 to 1) based on their GDT, and applies this score to adjust L2L predictions (GDT_{L2L}) **only** at inference time. This post-hoc refinement is not incorporated in learning and relies on manually assigned hyperparameters, which may not generalize well across different scenarios. As shown in Tab. 4, this design limits model performance and degrades robustness, further highlighting the advantage of our approach.

To better understand the above discussed differences, we conduct three comparative experiments summarized in Tab. 4, and analyze the results below.

Exp. 1: GDT *vs.* Geometry-Biased Self-Attention. To compare GDT with our proposed geometry-biased self-attention, we replace our geometry encoding with TopoLogic's end-to-start point distance to construct self-attention bias for lane representation learning. As shown in Tab. 4, this modification leads to noticeable performance degradation in lane detection: DET_l drops by 1.0 (Ours *vs.* Ours + GDT_{lane}), TOP_{ll} decreases by 0.8, and TOP_{lt} falls by 1.4. These results suggest that TopoLogic's GDT is insufficient for capturing the rich inter-lane spatial relationships essential for effective topology reasoning, reaffirming the advantage of our proposed relation embedding.

Exp. 2: Evaluating GDT in L2L Topology Reasoning. Unlike our (geometry-enhanced L2L topology) method, which integrates geometric cues directly into the L2L relation embedding during training, TopoLogic treats GDT as an auxiliary component applied at inference time to adjust L2L predictions. This approach relies on several hyperparameters to balance the influence between GDT and model-based predictions. Although these weights are optimized during training as part of lane representation learning, they are not used during L2L inference—manual values are assigned instead.

To examine the effect of GDT on topology learning, we evaluate TopoLogic's official model under two configurations: 1). TopoLogic†: which uses the hyperparameters learned during training; 2) TopoLogic – GDT_{L2L}: which removes GDT from the L2L topology inference process. As shown in Tab. 4, both configurations lead to significant performance degradation, with a severe drop in L2L topology accuracy when GDT is removed during L2L inference (TopoLogic *vs.* TopoLogic – GDT_{L2L}). This suggests that TopoLogic's lane features alone do not sufficiently encode relational information for effective topology reasoning. The reliance on post-hoc adjustments and manual tuning limits the robustness and generalizability of the method.

Exp 3: Applying GDT to Our L2L Inference. We further investigate the generalizability of GDT by incorporating it into our model's L2L inference, following TopoLogic's ensembling strategy. Specifically, we fuse our L2L predictions with GDT outputs using their predefined hyperparameters, without modifying our trained model. As shown in Tab. 4, this ensembling leads to a decrease in TOP_{ll} by 0.5 (Ours + GDT_{L2L} *vs.* Ours). This outcome highlights the limited transferability of TopoLogic's formulation. In contrast, our model achieves strong L2L performance without requiring additional tuning or post-processing, demonstrating that our relation embedding offers a more effective and learnable approach to modeling geometric topology.

Table 5: Comparsion with different bézier representation.

Lane Rep.	DET_l	DET_t	TOP_{ll}	TOP_{lt}	OLS
BeMapNet	29.9	50.0	25.9	28.9	46.2
BézierFormer	31.0	49.7	26.1	29.6	46.5
Ours	**33.8**	**50.9**	**29.2**	**32.2**	**48.9**

A.3.2 Comparsion of Bézier representation.

We compare our Bézier-based lane representation with other alternative formulations in Tab. 5. BézierFormer uses `grid_sample` to sample features and separately predicts offsets and attention weights for each point within deformable attention, which increases computational overhead and compromises instance-level consistency in lane prediction. In contrast, our method predicts offsets and attention weights jointly for the entire lane instance using a single lane query, resulting in improved efficiency and coherence. BeMapNet employs a piecewise Bézier representation by dynamically predicting the number of segments, which increases model complexity and uncertainty. In the task of lane topology reasoning, such dynamic segmentation does not yield performance gains and lacks the structural regularity that our fixed Bézier formulation provides.

A.3.3 Limitations and Future Work

Through qualitative analysis on test scenes, we identify recurring patterns where topology reasoning is more error-prone. These failure cases highlight opportunities for future enhancements:

- **Small traffic elements.** Tiny signs or lights at long range often appear with low pixel resolution in front-view images, making their detection and subsequent association to lanes difficult.
- **Distant lanes.** Some lanes far away project to only a few pixels in front-view, undermining their detectability and relational cue strength.
- **Occluded or partially visible lanes.** In dense urban scenes, lanes may be partially blocked or overlapped by other objects, confusing relational inference.

These challenging cases point toward future directions: incorporating **temporal continuity** (*e.g.*, across frames), leveraging motion cues or sequential context, and fusing richer spatiotemporal relational information may help improve robustness in such difficult scenes.

A.3.4 Qualitative Results

We present additional qualitative results in Fig. 5 and Fig. 6. Leveraging our proposed components, RelTopo demonstrates superior perception of lane centerlines compared to previous methods, as shown in the second row of the figures. Moreover, RelTopo achieves more accurate lane-to-lane (L2L)

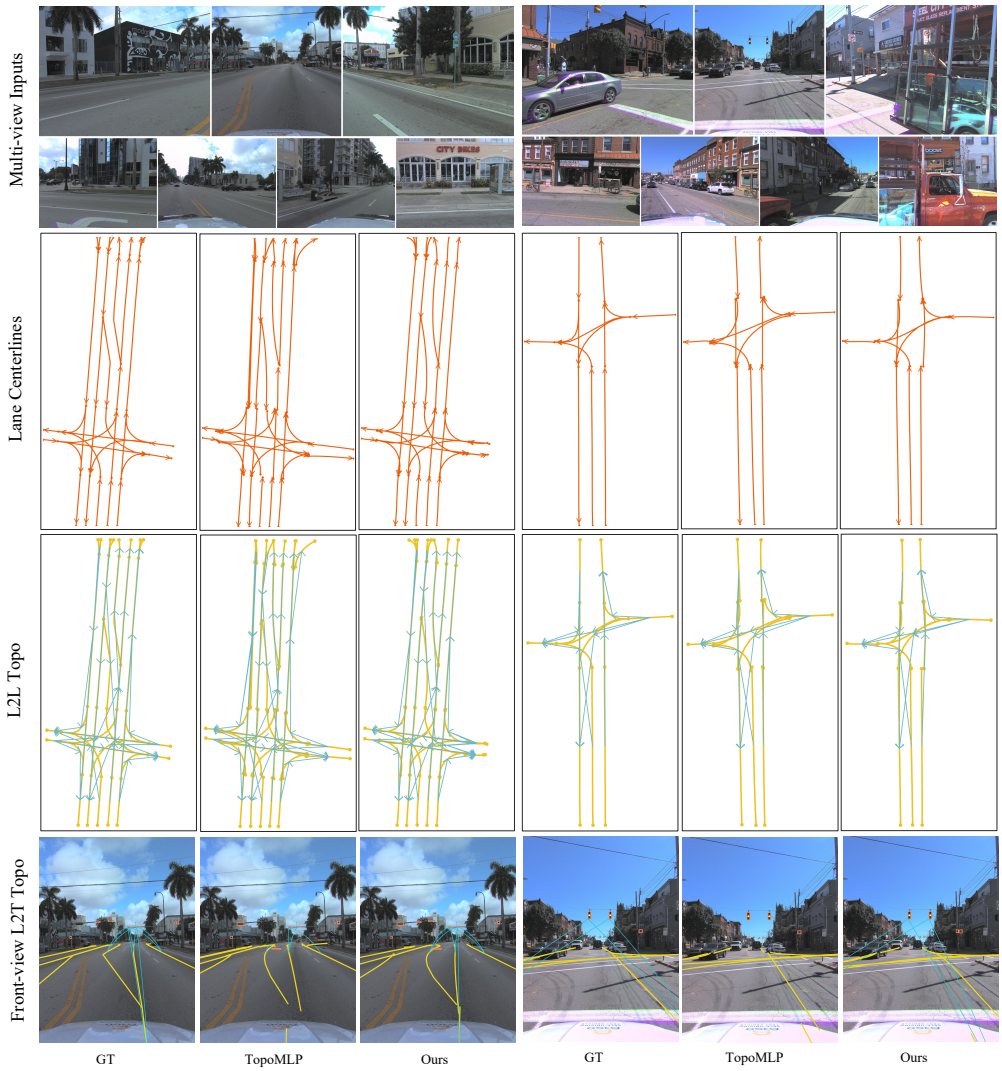

Figure 5: Qualitative results comparison. The 1st row presents the multi-view input images, the 2nd row shows the predicted lane centerline results, and the 3rd row illustrates the predicted L2L topology results, with light blue arrowlines indicating directed connectivity between lanes. The final row depicts the front-view L2T topology predictions, where orange boxes highlight detected traffic elements (*e.g.*, traffic lights and signs), mbevlane lines denote projected lanes, and blue lines in FV irepresent the pairing relationships between traffic elements and lanes. Each result row consists of three columns: the left column shows ground truth, the center column shows results from TopoMLP, and the right column presents our results. Two data samples are illustrated in this figure.

and lane-to-traffic-element (L2T) topology reasoning, as illustrated in the last two rows. Notably, RelTopo exhibits significantly improved L2T topology reasoning performance in complex scenarios involving intricate L2T relationships.

A.3.5 On DET_t Variation

In our ablation studies, interestingly, we observed variations in DET_t When modifying modules while keeping the traffic element decoder structure fixed. This phenomenon, also present in prior work but seldom discussed, deserves further analysis. We interpret this behavior as *inter-task interference*, a well-documented phenomenon in multi-task learning (MTL). In our experiments, introducing stronger geometric and relational biases can shift shared feature representations, which may transiently affect traffic element detection.

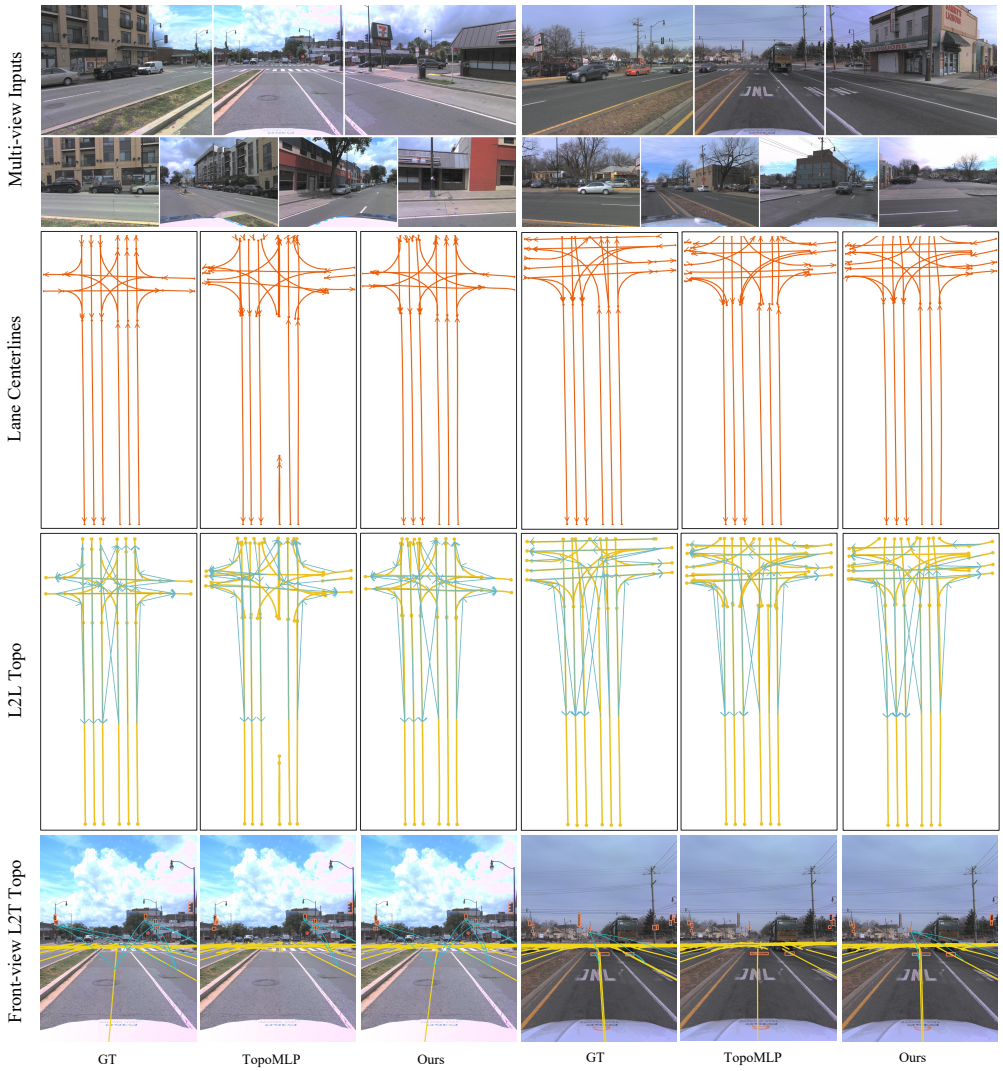

Figure 6: More qualitative result comparison. With our proposed designs, RelTopo achieves more accurate predictions for lane centerlines, as well as L2L and L2T topology reasoning.

Consistent with this interpretation, prior works report similar variation ranges: TopoMLP Wu et al. (2023) shows DET_t values spanning 48.5–50.9 (Tab.5 in their paper), while TopoLogic Fu et al. (2024) reports 44.1–47.2 across different settings (Tab.3–4 in their paper). Despite these variations in ablations, our final model consistently improves across all metrics, including DET_t (*e.g.*, comparing #R1 baseline vs. #R6 + relational perception and reasoning, and #R1 vs. #R7 + all designs in Tab. 2). This indicates that our multi-level relational modeling ultimately yields a net benefit across tasks.

While addressing cross-task interference is not the primary focus of this work, we recognize it as an important avenue for future research. Potential strategies such as gradient balancing may further stabilize performance.

A.3.6 Generalization to Stronger Encoders

We designed RelTopo to be encoder-agnostic and complementary to visual backbones: the relational modules sit on top of feature extraction and do not depend on a particular encoder. To assess the generalizability and scalability of our design, we conduct additional experiments using a stronger backbone, including ResNet-101 and Swin-Base Liu et al. (2021b).

As shown in Tab. 6, our relational modules consistently provide significant improvements regardless of backbone strength. For the default ResNet-50, adding our modules increases OLS from 44.5 to 48.9. For ResNet-101, OLS improves from 45.2 to 49.4. Even with Swin-Base, we see a jump from 47.5 to 51.1. These results clearly demonstrate that RelTopo is not tied to a specific encoder; instead, it can enhance performance across architectures.

Table 6: Results with different backbones (without vs with our designs).

Encoder	+ Our designs	OLS
ResNet-50	×	44.5
ResNet-50	✓	48.9
ResNet-101	×	45.2
ResNet-101	✓	49.4
Swin-Base	×	47.5
Swin-Base	✓	51.1