

Appendix

In the appendix, we supply further details on the proposed stability evaluation framework, the benchmark setup, experimental analyses, and visualizations that are omitted from the main paper for brevity. The appendix is structured as follows:

- Sec. **A** provides additional statements on the use of Large Language Models (LLMs) in this work.
- Sec. **B** presents implementation details of the stability evaluation algorithm pipeline.
- Sec. **C** offers supplementary experimental setups and related ablation studies.
- Sec. **D** presents additional analyses from 10 different models.
- Sec. **E** displays supplementary visualizations, encompassing additional mAP vs. mAS comparisons, and illustrations of how unstable predictions affect downstream tasks.
- Sec. **F** discusses the limitations of our work and provides an outlook on future work.

A STATEMENT ON THE USE OF LARGE LANGUAGE MODELS (LLMs)

In the preparation of this paper, large language models (LLMs) were used solely as an assistive tool for writing refinement and polishing. The core research ideas, theoretical framework, experimental design, data analysis, and result interpretation were entirely conceived and conducted by the human authors. The LLM was employed after the intellectual substance of the work was fully established, specifically to assist with improving grammatical correctness, sentence fluency, and overall clarity of expression in certain parts of the manuscript. It did not contribute to the scientific ideation, methodological development, or conclusions of the research. All final content was thoroughly reviewed, verified, and approved by the authors.

B STABILITY EVALUATION ALGORITHM WITH ADDITIONAL DETAILS

This section provides comprehensive algorithmic details for the multi-dimensional map stability evaluation framework introduced in Section 3. The complete pipeline consists of four main stages: temporal sampling, cross-frame instance matching, geometric alignment and resampling, and stability metric computation. Each stage is implemented through carefully designed algorithms that ensure robust and reproducible evaluation of temporal stability in online HD mapping.

B.1 TEMPORAL SAMPLING

Algorithm 1 implements the temporal sampling stage that constructs frame pairs for stability analysis. The algorithm randomly selects subsequent frames within a predefined maximum temporal interval M for each anchor frame, ensuring comprehensive coverage of temporal variations while maintaining computational efficiency. This approach generates a sampling set S of size $|S| = L - M$, providing the foundational inputs for subsequent stability analysis.

Algorithm 1 Temporal Sampling Algorithm

Require: Model output frame sequence $\{D_1, D_2, \dots, D_L\}$, maximum temporal interval M

Ensure: Frame pair sampling set S

```

1: // Stage 1: Temporal Sampling
2:  $S \leftarrow \emptyset$  {Initialize sampling set}
3: for  $t = 1$  to  $L - M$  do
4:    $k \leftarrow \text{RandomSample}(1, M)$  {Random sampling within  $[1, M]$  range}
5:    $S \leftarrow S \cup \{(D_t, D_{t+k})\}$  {Add frame pair to sampling set}
6: end for
7: return  $S$ 

```

B.2 CROSS-FRAME INSTANCE MATCHING

Algorithm 2 implements the cross-frame instance matching stage that establishes correspondence between map elements across temporal frames.

Algorithm 2 Cross-Frame Instance Matching Algorithm

Require: Frame pair sampling set S
Ensure: Matched polyline pairs $\mathcal{M}$

```

1: // Stage 2: Cross-Frame Instance Matching
2:  $\mathcal{M} \leftarrow \emptyset$  {Initialize matching result set}
3: for each  $(D_t, D_{t+k}) \in S$  do
4:   // Step 2.1: Frame-to-GT Matching
5:    $\text{matches}_t \leftarrow \text{HungarianMatching}(D_t, \text{GT}_t)$ 
6:    $\text{matches}_{t+k} \leftarrow \text{HungarianMatching}(D_{t+k}, \text{GT}_{t+k})$ 
7:   // Step 2.2: GT-based Association
8:    $E \leftarrow \text{FindCommonGTInstances}(\text{matches}_t, \text{matches}_{t+k})$ 
9:   for each  $e \in E$  do
10:     $\text{poly}_t(e) \leftarrow \text{GetPolyline}(\text{matches}_t, e)$ 
11:     $\text{poly}_{t+k}(e) \leftarrow \text{GetPolyline}(\text{matches}_{t+k}, e)$ 
12:     $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\text{poly}_{t+k}(e), \text{poly}_t(e))\}$ 
13:   end for
14: end for
15: return  $\mathcal{M}$ 

```

The algorithm employs a two-step strategy: first matching predictions to ground truth within each frame using the algorithm 3, then establishing temporal correspondence through ground truth-based association.

Algorithm 3 Hungarian Matching Sub-algorithm

Require: Prediction frame D , ground truth frame GT , cost function C
Ensure: Matching result matches

```

1:  $P \leftarrow \text{GetPolygons}(D)$  {Get prediction polygons}
2:  $G \leftarrow \text{GetPolygons}(\text{GT})$  {Get ground truth polygons}
3:  $n \leftarrow |P|, m \leftarrow |G|$ 
4: // Build cost matrix
5:  $C_{\text{matrix}} \leftarrow \text{zeros}(n \times m)$ 
6: for  $i = 1$  to  $n$  do
7:   for  $j = 1$  to  $m$  do
8:      $C_{\text{matrix}}[i, j] \leftarrow C(P[i], G[j])$  {Geometric and semantic similarity cost}
9:   end for
10: end for
11: // Execute Hungarian algorithm
12:  $\text{matches} \leftarrow \text{HungarianAlgorithm}(C_{\text{matrix}})$ 
13: return matches

```

This indirect matching approach leverages the consistency of ground truth annotations to overcome the inherent inconsistencies in model predictions, yielding a set of matched instance pairs for each frame pair.

B.3 GEOMETRIC ALIGNMENT AND RESAMPLING

Algorithm 4 implements the geometric alignment and resampling stage that ensures spatially consistent comparison between matched polylines.

The algorithm transforms historical polylines into the current frame’s coordinate system, applies algorithm 5 to ensure evaluation consistency, and performs uniform resampling along the x -axis. This process guarantees spatially aligned and comparable point sets for subsequent stability analy-

Algorithm 4 Geometric Alignment and Resampling Algorithm

Require: Matched polyline pairs $\mathcal{M}$, perception range $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$, resampling points N

Ensure: Matched and aligned polyline pairs $\mathcal{M}^{\text{sample}}$

- 1: **// Stage 3: Geometric Alignment and Resampling**
- 2: $\mathcal{M}^{\text{sample}} \leftarrow \emptyset$ {Initialize sampled polyline pairs set}
- 3: **for** each $(\text{poly}_{t+k}(e), \text{poly}_t(e)) \in \mathcal{M}$ **do**
- 4: **// Step 3.1: Coordinate Transformation**
- 5: $\text{poly}_{t \rightarrow t+k}(e) \leftarrow T_{\text{world} \rightarrow t+k} \cdot T_{t \rightarrow \text{world}} \cdot \text{poly}_t(e)$
- 6: **// Step 3.2: Perception Range Filtering**
- 7: $\text{poly}_{t \rightarrow t+k}(e) \leftarrow \text{ClipToPerceptionRange}(\text{poly}_{t \rightarrow t+k}(e), [x_{\min}, x_{\max}, y_{\min}, y_{\max}])$
- 8: **// Step 3.3: Uniform Resampling**
- 9: $[x_{\min}^p, x_{\max}^p] \leftarrow \text{GetCommonXRange}(\text{poly}_{t+k}(e), \text{poly}_{t \rightarrow t+k}(e))$
- 10: **for** $i = 1$ to N **do**
- 11: $x_i \leftarrow x_{\min}^p + (i - 1) \cdot \frac{x_{\max}^p - x_{\min}^p}{N - 1}$
- 12: $y_{t+k}(x_i) \leftarrow \text{InterpolateY}(\text{poly}_{t+k}(e), x_i)$
- 13: $y_t(x_i) \leftarrow \text{InterpolateY}(\text{poly}_{t \rightarrow t+k}(e), x_i)$
- 14: **end for**
- 15: $\text{poly}_{t+k}^{\text{sample}}(e) \leftarrow \{(x_i, y_{t+k}(x_i)) \mid i = 1, 2, \dots, N\}$
- 16: $\text{poly}_t^{\text{sample}}(e) \leftarrow \{(x_i, y_t(x_i)) \mid i = 1, 2, \dots, N\}$
- 17: $\mathcal{M}^{\text{sample}} \leftarrow \mathcal{M}^{\text{sample}} \cup \{(\text{poly}_{t+k}^{\text{sample}}(e), \text{poly}_t^{\text{sample}}(e), e)\}$
- 18: **end for**
- 19: **return** $\mathcal{M}^{\text{sample}}$

sis, returning a comprehensive set of matched and aligned polyline pairs, each annotated with the corresponding map element identifier.

Algorithm 5 Perception Range Filtering Sub-algorithm

Require: Transformed polyline $\text{poly}_{t \rightarrow t+k}(e)$, perception range $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$

Ensure: Filtered polyline $\text{poly}_{\text{filtered}}$

- 1: $\text{poly}_{\text{filtered}} \leftarrow \emptyset$
- 2: **for** each point $(x, y) \in \text{poly}_{t \rightarrow t+k}(e)$ **do**
- 3: **if** $x_{\min} \leq x \leq x_{\max}$ **AND** $y_{\min} \leq y \leq y_{\max}$ **then**
- 4: $\text{poly}_{\text{filtered}} \leftarrow \text{poly}_{\text{filtered}} \cup \{(x, y)\}$
- 5: **end if**
- 6: **end for**
- 7: **return** $\text{poly}_{\text{filtered}}$

B.4 STABILITY METRIC COMPUTATION

Algorithm 6 implements the core stability metric computation that quantifies temporal stability across three dimensions: presence, localization, and shape. The algorithm processes each matched polyline pair to compute individual stability scores, then aggregates these scores at the class and model levels. The presence stability evaluates detection consistency, the localization stability quantifies positional jitter, and the shape stability assesses geometric consistency through curvature comparison.

The specific implementation of the ComputeCurvature function mentioned in Algorithm 6 is detailed in Algorithm 7, which approximates polyline curvature by computing the average angles between consecutive segments, thereby providing a robust geometric measurement method with translation and rotation invariance.

This curvature-based approach enables effective comparison of geometric consistency across temporal frames while capturing subtle shape variations that may indicate instability in the model’s predictions.

Algorithm 6 Multi-dimensional Map Stability Evaluation Framework - Stability Metric Computation

Require: Matched and aligned polyline pairs $\mathcal{M}^{\text{sample}}$, detection threshold τ , weighting parameter ω , scaling parameter β

Ensure: Overall model stability score mAS

```

1: // Stage 4: Stability Metric Computation
2:  $\mathcal{C} \leftarrow \text{GetAllClasses}(\mathcal{M}^{\text{sample}})$  {Get all classes}
3: for each class  $\in \mathcal{C}$  do
4:    $\mathcal{I}_{\text{class}} \leftarrow \text{GetInstancesOfClass}(\mathcal{M}^{\text{sample}}, \text{class})$ 
5:    $\text{stability}_{\text{class}} \leftarrow 0$ 
6:   for each  $(\text{poly}_{t+k}^{\text{sample}}(e), \text{poly}_t^{\text{sample}}(e), e) \in \mathcal{I}_{\text{class}}$  do
7:     // Compute Presence Stability
8:     if  $\text{score}_{t+k}(e) \geq \tau$  AND  $\text{score}_t(e) \geq \tau$  OR  $\text{score}_{t+k}(e) < \tau$  AND  $\text{score}_t(e) < \tau$  then
9:        $\text{Presence}(e) \leftarrow 1$ 
10:    else
11:       $\text{Presence}(e) \leftarrow 0.5$  {Flickering case}
12:    end if
13:    // Compute Localization Stability
14:     $\text{avg\_deviation} \leftarrow \frac{1}{N} \sum_{i=1}^N |y_{t+k}(x_i) - y_t(x_i)|$ 
15:     $\text{Loc}(e) \leftarrow \beta \cdot \text{avg\_deviation}$ 
16:    // Compute Shape Stability
17:     $\kappa_{t+k} \leftarrow \text{ComputeCurvature}(\text{poly}_{t+k}^{\text{sample}}(e))$ 
18:     $\kappa_t \leftarrow \text{ComputeCurvature}(\text{poly}_t^{\text{sample}}(e))$ 
19:     $\text{Shape}(e) \leftarrow 1 - \frac{|\kappa_{t+k} - \kappa_t|}{\pi}$ 
20:    // Compute Comprehensive Stability
21:     $\text{Stability}(e) \leftarrow \text{Presence}(e) \cdot [\omega \cdot \text{Loc}(e) + (1 - \omega) \cdot \text{Shape}(e)]$ 
22:     $\text{stability}_{\text{class}} \leftarrow \text{stability}_{\text{class}} + \text{Stability}(e)$ 
23:  end for
24:   $\text{Stability}_{\text{class}} \leftarrow \frac{\text{stability}_{\text{class}}}{|\mathcal{I}_{\text{class}}|}$ 
25: end for
26: // Compute Overall Model Stability
27:  $\text{mAS} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{\text{class} \in \mathcal{C}} \text{Stability}_{\text{class}}$ 
28: return mAS

```

Algorithm 7 Curvature Computation Sub-algorithm

Require: Resampled polyline $\text{poly}^{\text{sample}} = \{(x_i, y_i) \mid i = 1, 2, \dots, N\}$

Ensure: Curvature κ

```

1:  $\kappa \leftarrow 0$ 
2: for  $j = 1$  to  $N - 1$  do
3:    $\vec{v}_j \leftarrow (x_{j+1} - x_j, y_{j+1} - y_j)$  {Compute vector  $\vec{v}_j$ }
4:    $\vec{v}_{j+1} \leftarrow (x_{j+2} - x_{j+1}, y_{j+2} - y_{j+1})$  {Compute vector  $\vec{v}_{j+1}$ }
5:    $\theta_j \leftarrow \cos^{-1} \left( \frac{\vec{v}_j \cdot \vec{v}_{j+1}}{|\vec{v}_j| \cdot |\vec{v}_{j+1}|} \right)$  {Compute angle}
6:    $\kappa \leftarrow \kappa + \theta_j$ 
7: end for
8:  $\kappa \leftarrow \frac{\kappa}{N-1}$  {Compute average curvature}
9: return  $\kappa$ 

```

Upon completion of all sub-metric evaluations, the final mean Average Stability (mAS) score is computed, providing a holistic measure of the model’s temporal stability across all evaluated classes.

C FURTHER DETAILS ON THE EXPERIMENTAL SETUP

C.1 SUPPLEMENTAL DETAILS ON THE EXPERIMENTAL SETUP

Prior to conducting experiments, it is necessary to configure certain hyperparameters. This section primarily elaborates on the detailed configurations of these hyperparameters adopted in our study, along with the rationale for these choices:

- **Maximum frame interval ($M=2$):** The configuration of different frame intervals essentially represents distinct evaluation scenarios for stability assessment, each carrying unique implications. Therefore, in addition to the experiments with $M=2$ presented in the main text, as shown in chapter C.2, we have conducted supplementary experiments with $M=3$, $M=5$, and $M=10$ to provide as comprehensive a stability evaluation as possible for existing models.
- **Number of resampling points ($N=100$):** The purpose of resampling is to adjust the distribution of map points on two instance polylines to be identical, thereby facilitating the calculation of stability metrics. Thus, the value of N should not be set too small to avoid undue influence from individual outliers on the instances in subsequent computations. However, beyond a sufficient threshold, variations in N do not significantly affect stability evaluation outcomes. Conversely, excessively large values of N may substantially reduce computational efficiency. Balancing resampling granularity and computational cost, we ultimately set $N = 100$. This value can be appropriately adjusted in different experimental settings.
- **Position Stability Scaling Factor ($\beta=15.0$):** The specific implication of this scaling factor is that when the distance between two matched map points (i.e., points sampled at identical x -values on matched map instances) in adjacent frames equals β , their positional stability is considered zero. Consequently, the value of β corresponds to the distance threshold representing extreme instability. Typically, we define such extreme cases using the shorter map radius (half the length of the map’s shorter side). In prevailing map construction paradigms, the standard map range is generally defined as $x \in [-15, 15]$, $y \in [-30, 30]$. Therefore, in our experiments, β is set to 15.0.

Table 7: Ablation Study on the Temporal Sampling Interval ($M=3$)

Method	Venue	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$
MapTR	ICLR’23	$\times$	C	GKT	24	44.1	89.3	59.9	90.6	67.6
MapTR	ICLR’23	$\times$	C & L	GKT	24	62.8	91.0	69.3	91.8	73.7
BeMapNet	CVPR’23	$\times$	C	IPM-PE	30	61.4	100.0	58.8	97.7	78.2
PivotNet	ICCV’23	$\times$	C	PersFormer	30	57.1	100.0	45.2	98.3	71.7
MapTRv2	IJCV’24	$\times$	C	BEVPool	24	61.4	90.6	59.5	91.8	69.0
GeMap	ECCV’24	$\times$	C	BEVFormer-1	24	51.3	90.9	66.6	92.9	73.3
MGMMap	CVPR’24	$\times$	C	BEVFormer-1	24	57.9	91.8	68.8	92.3	74.4
MapQR	ECCV’24	$\times$	C	BEVFormer-3	24	66.4	89.4	66.9	91.1	73.4
MapTR	ICLR’23	$\checkmark$	C	GKT	24	51.3	86.8	55.2	89.1	62.9
StreamMapNet	WACV’24	$\checkmark$	C	BEVFormer-1	30	63.3	96.9	96.6	95.8	93.2
MapTracker	ECCV’24	$\checkmark$	C	BEVFormer-2	72	75.95	93.7	96.2	93.4	88.7
HRMapNet	ECCV’24	$\checkmark$	C	BEVFormer-1	24	67.2	91.2	70.4	92.2	75.4

C.2 ABLATION STUDY ON THE TEMPORAL SAMPLING INTERVAL

Temporal sampling serves as the initial step in the stability evaluation benchmark. In the default configuration, the time interval M is set to 2 to assess the granularity of map changes, as delineated in Table 1. To provide a more comprehensive illustration of our evaluation framework’s performance across different temporal sampling intervals, we conducted further experiments with M values of 3, 5, and 10. The results are presented in Tables 7, 8, and 9 respectively. All other experimental settings remain consistent with Table 1.

Table 8: Ablation Study on the Temporal Sampling Interval ($M=5$)

Method	Venue	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$
MapTR	ICLR'23	$\times$	C	GKT	24	44.1	89.0	64.7	90.0	68.8
MapTR	ICLR'23	$\times$	C & L	GKT	24	62.8	89.4	69.2	91.0	72.1
BeMapNet	CVPR'23	$\times$	C	IPM-PE	30	61.4	100.0	50.0	97.5	73.7
PivotNet	ICCV'23	$\times$	C	PersFormer	30	57.1	100.0	41.5	98.3	70.0
MapTRv2	IJCV'24	$\times$	C	BEVPool	24	61.4	89.0	52.8	90.8	64.8
GeMap	ECCV'24	$\times$	C	BEVFormer-1	24	51.3	88.8	61.7	91.6	69.3
MGMap	CVPR'24	$\times$	C	BEVFormer-1	24	57.9	90.0	67.1	91.5	71.9
MapQR	ECCV'24	$\times$	C	BEVFormer-3	24	66.4	88.1	59.9	89.6	67.4
MapTR	ICLR'23	$\checkmark$	C	GKT	24	51.3	85.0	55.8	90.3	61.8
HRMapNet	ECCV'24	$\checkmark$	C	BEVformer-1	24	67.2	89.6	70.0	92.1	73.8

Table 9: Ablation Study on the Temporal Sampling Interval ($M=10$)

Method	Venue	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$
MapTR	ICLR'23	$\times$	C	GKT	24	44.1	88.0	47.4	90.1	61.9
MapTR	ICLR'23	$\times$	C & L	GKT	24	62.8	89.0	58.2	90.9	66.5
BeMapNet	CVPR'23	$\times$	C	IPM-PE	30	61.4	100.0	41.6	97.5	69.5
PivotNet	ICCV'23	$\times$	C	PersFormer	30	57.1	100.0	28.3	98.9	63.6
MapTRv2	IJCV'24	$\times$	C	BEVPool	24	61.4	82.5	49.9	90.3	58.8
GeMap	ECCV'24	$\times$	C	BEVFormer-1	24	51.3	85.4	61.2	92.3	65.6
MGMap	CVPR'24	$\times$	C	BEVFormer-1	24	57.9	89.7	57.5	92.4	68.1
MapQR	ECCV'24	$\times$	C	BEVFormer-3	24	66.4	84.9	43.8	95.4	59.0
MapTR	ICLR'23	$\checkmark$	C	GKT	24	51.3	92.8	41.0	92.8	62.3
HRMapNet	ECCV'24	$\checkmark$	C	BEVFormer-1	24	67.2	83.8	55.8	92.8	62.4

The ablation studies reveal several important patterns regarding temporal stability assessment. First, as the temporal interval M increases from 2, 3, 5, 10, most models exhibit a progressive decline in stability scores across all metrics, particularly in Localization Stability. This pattern is consistent across different architectural paradigms and demonstrates the challenge of maintaining consistency over longer time horizons. For instance, MapTR (Liao et al., 2022) with camera-only input shows a reduction in mAS from 71.6 ($M=2$) to 61.9 ($M=10$), primarily driven by decreasing Localization Stability. Similar trends are observed for other non-temporal models, with BeMapNet (Qiao et al., 2023) maintaining superior Presence Stability but experiencing significant Localization Stability degradation from 65.8 ($M=2$) to 41.6 ($M=10$).

The comparative analysis reveals distinctive robustness characteristics across representation paradigms. Models with inherent temporal modeling capabilities, such as StreamMapNet (Yuan et al., 2024) and MapTracker (Chen et al., 2024), demonstrate remarkable resilience to increasing temporal intervals. StreamMapNet maintains exceptional stability with mAS of 93.2 at $M=3$, significantly outperforming non-temporal counterparts. This performance advantage is particularly pronounced in Localization Stability, where temporal models consistently exceed 96.0 even at larger intervals, compared to the substantial degradation observed in static models.

The studies also reveal paradigm-specific sensitivity patterns. Geometric-prior-based models like BeMapNet (Qiao et al., 2023) and PivotNet (Ding et al., 2023) maintain perfect Presence Stability across all intervals but exhibit considerable vulnerability in Localization Stability. In contrast, learning-based BEV representation models show more balanced degradation across stability dimensions. The performance variations across intervals provide additional evidence that accuracy (mAP) and stability (mAS) represent independent evaluation dimensions, as models with comparable mAP scores exhibit dramatically different stability characteristics under extended temporal intervals.

These findings underscore the importance of evaluating temporal stability across multiple time scales, as different representation paradigms exhibit distinct degradation patterns. The comprehensive interval analysis reinforces our central thesis that temporal stability constitutes a fundamental performance dimension that requires explicit consideration in online HD mapping system design and evaluation.

D COMPREHENSIVE EVALUATION RESULTS WITH FURTHER DETAILS

We present a detailed analysis of the performance of various online HD mapping models in terms of both accuracy (mAP) and temporal stability (mAS), based on the comprehensive results summarized in Tables 10 - Table 19. Our analysis highlights how different architectural choices, including backbone networks, temporal modeling, sensor modalities, and BEV encoders, affect these two critical performance dimensions.

Table 10: Evaluation Results of MapTR

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$	Parameters $\downarrow$
R18	$\times$	C	GKT	24	32.4	87.8	75.0	88.5	72.8	15.4M
R18	$\times$	C	GKT	110	45.5	86.0	71.7	94.8	71.8	15.4M
R50	$\times$	C	GKT	24	44.1	91.2	65.4	90.6	71.6	36.2M
R50	$\times$	C	GKT	110	50.5	89.8	63.2	91.0	68.2	36.2M
R50	$\times$	C	BEVFormer-1	24	41.6	89.6	69.7	90.6	71.3	36.3M
R50	$\times$	C	BEVPool	24	50.1	89.3	69.8	88.5	71.9	32.3M
R50	$\checkmark$	C	GKT	24	51.3	88.6	59.7	89.3	66.6	36.2M
R50	$\checkmark$	C	BEVFormer-1	24	53.3	90.4	69.5	91.2	73.0	36.3M
R50 & Sec.	$\times$	C & L	GKT	24	62.8	90.1	75.2	90.8	74.9	40.1M

D.1 COMPREHENSIVE ANALYSIS TOWARDS MAPTR

As presented in Table 10, the extensive variants of MapTR (Liao et al., 2022) provide a controlled setting to dissect how distinct representation paradigms influence model behavior. By altering key components while holding the core architecture constant, we can isolate their effects on both accuracy (mAP) and temporal stability (mAS).

Temporal Fusion. The effect of incorporating temporal fusion is not uniform but is mediated by the underlying BEV representation. When applied to the GKT-based (Chen et al., 2022) representation, temporal fusion disrupts its core strength. Presence Stability drops from 91.2 to 88.6, and Localization Stability plummets from 65.4 to 59.7, leading to a significant decrease in mAS (71.6 to 66.6). This indicates that the representation formed by GKT is not easily aligned or integrated across time; the temporal module may introduce noise rather than useful context. In contrast, when applied to the BEVFormer-based (Li et al., 2024) representation, which is already designed for spatiotemporal modeling, temporal fusion acts as a complementary enhancement. It improves mAP substantially (41.6 to 53.3) while slightly improving or maintaining stability scores, resulting in a higher mAS (71.3 to 73.0). This demonstrates that temporal fusion is most effective when the base representation is inherently compatible with processing sequential data. The effectiveness of temporal fusion is not a standalone property but is contingent on the representational capacity of the BEV encoder. It amplifies the capabilities of a temporally-aware representation (BEVFormer) but can degrade the performance of a primarily spatially-focused one (GKT). Therefore, temporal fusion serves as a force multiplier for representations already predisposed to temporal modeling, but can be detrimental to those that are not.

2D Backbone. Changing the 2D backbone from ResNet-18 to ResNet-50 shifts the model’s representational focus towards more complex visual patterns. This shift has a clear effect: it consistently improves mAP (32.4 to 44.1) by leveraging higher-capacity feature extraction. However, this comes with a redistribution of stability properties: Presence Stability improves (87.8 to 91.2), but Localization Stability worsens significantly (75.0 to 65.4). The deeper network appears to learn a representation that is more sensitive to semantic content but potentially more susceptible to per-frame variations in texture or lighting, which can harm geometric consistency. The backbone network influences the type of features that form the basis of the map representation. More powerful backbones enhance semantic discrimination but can introduce high-frequency noise that undermines geometric stability, suggesting that representations favoring stability may require features that are invariant to superficial appearance changes.

Table 11: Evaluation Results of BeMapNet

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$	Parameters $\downarrow$
Effb0	$\times$	C	IPM-PE	30	60.7	100.0	67.9	97.9	82.9	55.4M
R50	$\times$	C	IPM-PE	30	61.4	100.0	65.8	97.9	81.9	73.8M
SwinT	$\times$	C	IPM-PE	30	64.1	100.0	62.8	98.0	80.4	79.6M
R50	$\times$	C	IPM-PE	110	66.2	100.0	62.1	98.2	80.2	73.8M
SwinT	$\times$	C	IPM-PE	110	68.3	100.0	64.0	98.2	81.1	79.6M

Table 12: Evaluation Results of PivotNet

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$	Parameters $\downarrow$
Effb0	$\times$	C	PersFormer	30	57.8	100.0	71.8	97.2	84.5	17.1M
R50	$\times$	C	PersFormer	30	57.1	100.0	71.4	97.2	84.3	41.2M
Swin-T	$\times$	C	PersFormer	30	61.6	100.0	71.6	97.2	84.4	44.8M
Swin-T	$\times$	C	PersFormer	110	66.4	100.0	72.1	97.4	84.8	44.8M

D.2 IN-DEPTH ANALYSIS OF BEMAPNET AND PIVOTNET

As shown in Table 11 and Table 12, both BeMapNet(Qiao et al., 2023) and PivotNet(Ding et al., 2023) demonstrate stable mAP and mAS performance, with BeMapNet achieving mAP scores of 60.7 to 68.3 and mAS values of 80.2 to 82.9, while PivotNet demonstrates mAP values of 57.1 to 66.4 and mAS scores of 84.3 to 84.8. This indicates that these two models are insensitive to backbone network selection and training epoch configurations. The difference lies in the fact that PivotNet achieves its highest mAS when using Swin-T(Liu et al., 2021) as the backbone, while BeMapNet attains its peak mAS value with an EfficientNet-B0(Tan & Le, 2019) backbone.

It should be specifically noted that both BeMapNet and PivotNet adopt a “dynamic vectorized sequence” representation for map encoding, which explains their consistently perfect presence metrics (100%). However, this representation format severely limits the localization stability of map instances, resulting in significantly lower performance compared to models like GeMap(Zhang et al., 2024b), MGMap(Liu et al., 2024a), and MapQR(Liu et al., 2024b).

Table 13: Evaluation Results of MapTRv2

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP $\uparrow$	Presence $\uparrow$	Loc $\uparrow$	Shape $\uparrow$	mAS $\uparrow$	Parameters $\downarrow$
R18	$\times$	C	BEVPool	24	57.2	91.0	73.2	91.2	75.6	27.9M
R50	$\times$	C	BEVPool	24	61.4	91.5	68.6	91.0	74.0	40.6M

D.3 EVALUATION AND ANALYSIS OF MAPTRV2

MapTRv2 (Liao et al., 2025b) improves upon MapTR (Liao et al., 2022) with higher baseline mAP (57.2–61.4) and mAS (74.0–75.6). Interestingly, the R18 backbone achieves higher mAS (75.6) than R50 (74.0), despite a lower mAP (57.2 vs. 61.4), as illustrated in Table 10 and Table 13, reinforcing the independence of accuracy and stability.

D.4 DISCUSSION ON GEMAP

As illustrated in Table 14, GeMap (Zhang et al., 2024b) presents a particularly instructive case for examining the complex relationship between accuracy and stability. The model demonstrates a strong capacity for high accuracy, with its mAP score scaling significantly from 51.3 to a top score of 76.0 when employing a powerful V2-99 (Lee et al., 2019) backbone and extended training. However, this pursuit of accuracy often introduces instability, as evidenced by its mAS scores, which range from a moderate 71.8 to a more competitive 78.1.

A critical observation is the divergent effect of LiDAR fusion. While integrating LiDAR data with a ResNet-50 (He et al., 2016) backbone yields a predictable improvement in mAP (+6.1%, from 62.7 to 66.5), it conversely leads to a decrease in mAS (-3.9%, from 74.7 to 71.8). This result challenges the conventional wisdom that more sensor data invariably leads to more robust perception.

Table 14: Evaluation Results of GeMap

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R50	✗	C	BEVFormer-1	24	51.3	92.3	69.7	92.6	75.5	44.1M
R50	✗	C	BEVFormer-1	110	62.7	91.1	67.5	94.5	74.7	44.1M
Swin-T	✗	C	BEVFormer-1	110	72.0	92.2	74.9	93.2	78.1	50.5M
V2-99	✗	C	BEVFormer-1	110	72.0	89.2	71.5	92.6	74.2	92.6M
V2-99(DD3D)	✗	C	BEVFormer-1	110	76.0	93.4	66.9	93.7	75.1	92.6M
R50 & second	✗	C & L	BEVFormer-1	110	66.5	89.1	66.3	92.7	71.8	48.0M

It suggests that GeMap’s architecture, while effectively leveraging LiDAR for geometric precision in a single frame, may lack the necessary mechanisms to harmonize the potentially noisy or asynchronous multi-modal signals across time, leading to increased jitter or flickering.

Furthermore, the model exhibits high sensitivity to backbone design. The Swin-T (Liu et al., 2021) backbone strikes the most favorable balance, achieving the highest mAS (78.1) alongside a high mAP (72.0). In contrast, the larger V2-99 (Lee et al., 2019) backbone, despite achieving the peak mAP (76.0), produces a lower mAS (75.1). The degradation in Localization Stability (from 74.9 with Swin-T (Liu et al., 2021) to 66.9 with V2-99 (Lee et al., 2019)) is especially notable, implying that the increased representational power of the larger backbone may overfit to single frame features at the expense of temporal coherence. This pattern underscores that for stability, simply scaling up model capacity is not a sufficient strategy and may even be counterproductive without explicit temporal regularization.

Table 15: Evaluation Results of MGMap

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R50	✗	C	BEVFormer-1	24	58.0	92.2	75.0	92.3	78.0	55.9M

D.5 ANALYSIS FOR MGMAP

MGMap (Liu et al., 2024a) achieves a balanced profile (mAP: 58.0, mAS: 78.0) with a ResNet-50 backbone (He et al., 2016) and BEVFormer (Li et al., 2024) encoder, as presented in Table 15. Its strong Localization and Shape Stability scores (75.0 and 92.3, respectively) suggest robustness against geometric jitter.

Table 16: Evaluation Results of MapQR

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R18	✗	C	BEVFormer-3	24	62.3	88.2	73.1	92.5	74.1	112.6M
R50	✗	C	BEVFormer-3	24	66.4	91.8	75.6	91.6	77.8	125.4M
R50	✗	C	BEVFormer-3	110	72.6	92.4	75.9	96.4	80.3	125.4M

D.6 STUDY OF MAPQR

As shown in Table 16, MapQR (Liu et al., 2024b) shows a clear positive scaling trend: larger backbones and longer training improve both mAP (62.3 to 72.6) and mAS (74.1 to 80.3). This indicates that the model’s architecture supports stable learning under increased capacity.

D.7 TOWARDS A COMPREHENSIVE ANALYSIS OF STREAMMAPNET

StreamMapNet (Yuan et al., 2024) stands out as the paradigm for temporally stable online mapping, achieving the highest mAS scores in our benchmark, ranging from 83.8 to an exceptional 94.4. This performance is primarily driven by its native temporal architecture, which is explicitly designed to model consistency across frames.

The most striking feature of StreamMapNet is its near-perfect Localization Stability (97.7–98.5), the highest among all models, as shown in Table 1 and Table 17. This indicates an exceptional

Table 17: Evaluation Results of StreamMapNet

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R50	✗	C	BEVFormer-1	30	51.7	87.0	97.8	95.1	83.8	56.0M
R18	✓	C	BEVFormer-1	30	27.8	87.1	98.4	94.6	85.0	42.5M
R50	✓	C	BEVFormer-1	30	63.4	96.6	97.7	92.3	91.9	56.3M
R50	✓	C	BEVFormer-1	24	51.2	97.0	98.5	96.1	94.4	56.3M

ability to suppress the positional jitter of map elements over time, a critical factor for downstream planning tasks. The analysis clearly shows that temporal fusion is not merely an optional add-on but the core determinant of its performance. Enabling temporal modeling (comparing the R50, ✓ vs. R50, ✗ configurations) results in a dramatic improvement in both mAP (+22.6%, from 51.7 to 63.4) and mAS (+9.7%, from 83.8 to 91.9). This dual improvement confirms that effectively leveraging historical context can simultaneously enhance per-frame accuracy and inter-frame consistency.

Table 18: Evaluation Results of MapTracker

Backbone	Temp	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R18	✗	C	BEVFormer-2	72	62.8	95.3	97.3	85.9	87.4	60.7M
R50	✗	C	BEVFormer-2	72	68.3	94.5	97.9	93.8	90.8	74.0M
R18	✓	C	BEVFormer-2	48	69.3	94.8	98.2	94.8	91.5	60.7M
R18	✓	C	BEVFormer-2	72	71.9	92.9	98.5	94.8	89.9	60.7M
R50	✓	C	BEVFormer-2	48	73.0	91.7	98.5	96.0	91.7	74.0M
R50	✓	C	BEVFormer-2	72	76.0	93.3	98.1	95.8	90.4	74.0M

D.8 A COMPREHENSIVE ANALYSIS OF MAPTRACKER

MapTracker (Chen et al., 2024) represents another strong temporal model that successfully balances state-of-the-art accuracy with high stability, which is shown in Table 1 and Table 18. It achieves the highest overall mAP (76.0) in our benchmark while maintaining mAS scores above 87.4, peaking at 91.7.

Similar to StreamMapNet (Yuan et al., 2024), MapTracker’s integration of temporal fusion (“Temp = ✓”) consistently boosts mAP (e.g., from 62.8 to 69.3 for R18) while preserving high mAS. This reinforces the conclusion that architectures designed with temporal reasoning in mind from the ground up are essential for high-performance online mapping. The model also exhibits very strong Localization and Shape Stability, often exceeding 98.0 and 94.0, respectively, which is characteristic of models that effectively aggregate information over time.

However, MapTracker reveals a nuanced trade-off related to training duration. For both the R18 and R50 backbones, extending training from 48 to 72 epochs leads to a further increase in mAP but a slight decrease in mAS (e.g., R50: mAP 73.0 to 76.0, mAS 91.7 to 90.4). This pattern, which we term optimization sensitivity, suggests that as the model continues to minimize a primarily accuracy-oriented loss function, it may gradually overfit to single-frame details, thereby sacrificing some temporal smoothness. This highlights a key area for future research: the development of loss functions or regularization techniques that explicitly penalize temporal instability during training to prevent this erosion.

Table 19: Evaluation Results of HRMapNet

Backbone	Temp	Initial Map	Modal	BEV Encoder	Epoch	mAP↑	Presence↑	Loc↑	Shape↑	mAS↑	Parameters↓
R50	✓	✗	C	BEVFormer-1	24	67.2	92.3	70.5	91.5	75.9	47.3M
R50	✓	Testing Map	C	BEVFormer-1	24	73.0	94.9	71.4	93.0	78.4	47.3M
R50	✓	Training Map	C	BEVFormer-1	24	83.6	89.9	75.9	93.2	76.7	47.3M
R50	✓	✗	C	BEVFormer-1	110	73.5	90.5	74.1	92.7	75.9	47.3M

D.9 DETAILED INVESTIGATION OF HRMAPNET

As presented in Table 19, HRMapNet (Zhang et al., 2024a) incorporates a distinct representation paradigm by integrating static map priors into a temporal mapping framework. The performance variations across its configurations provide critical insights into the interaction between dynamic sensory input and static prior knowledge.

The most pronounced effect is observed on single frame accuracy. Utilizing a map prior during training yields a substantial improvement in mAP, elevating the score from 67.2 to 83.6. This result indicates that the model’s representation effectively internalizes the structural constraints provided by the high quality offline map, leading to superior geometric precision in individual frames.

In contrast, the impact of map priors on temporal stability is more complex and less direct. The configuration employing a prior only during testing achieves the highest mAS of 78.4 and the highest Presence Stability of 94.9. This suggests that an externally provided prior can serve as a stabilizing reference during inference, enhancing detection consistency without being fully baked into the model parameters.

However, when the model is trained with the map prior, a different pattern emerges. While this configuration achieves the highest mAP, its mAS of 76.7 is lower than the testing prior variant. Notably, its Presence Stability decreases to 89.9. This indicates that deep integration of the static prior during training may lead to a representation that is overly reliant on persistent features, potentially at the expense of robustness to real world variations encountered in a temporal sequence. The model may become less adept at handling cases where the prior is imperfect or where dynamic scenes deviate from the stored map.

Furthermore, extending training to 110 epochs without any initial map prior improves mAP to 73.5 but leaves mAS unchanged at 75.9. This stability saturation effect underscores that prolonged training on a single frame accuracy objective has diminishing returns for temporal consistency. The gain in stability achieved through the intelligent use of a testing time prior surpasses that achieved by simply training the baseline model longer.

In summary, HRMapNet demonstrates that static map priors constitute a powerful representation for enhancing perceptual accuracy. Their utility for improving temporal stability, however, is contingent on the method of integration. A prior used as a dynamic guidance signal at inference can bolster consistency, whereas deeply embedding the prior into the model through training may introduce a trade off, favoring accuracy over stability. This highlights that the effective fusion of dynamic and static representations remains a key challenge for robust online mapping.

D.10 SUMMARY OF COMPREHENSIVE EVALUATION

The comprehensive evaluation of ten representative online HD mapping models demonstrates that different representation paradigms induce distinct performance characteristics along the accuracy-stability spectrum. Our analysis reveals that these two performance dimensions are independently influenced by specific architectural choices and their underlying representational biases.

Models incorporating strong geometric priors, such as BeMapNet (Qiao et al., 2023) and PivotNet (Ding et al., 2023), achieve exceptional Presence Stability due to their structure-aware representations. In contrast, architectures based on learned view transformations like BEVFormer (Li et al., 2024) exhibit superior Localization Stability, benefiting from their spatially coherent bird’s eye view representations.

Temporal modeling effectiveness shows fundamental dependence on representational compatibility. Architectures with native temporal designs demonstrate that explicit sequence modeling produces the highest stability scores. However, the integration of temporal modules requires careful alignment with the base representation, as evidenced by the varied outcomes when adding temporal components to different BEV encoders.

Multi-modal integration exhibits model-dependent effects on stability. While sensor fusion generally enhances accuracy, its impact on temporal consistency varies across architectures, indicating that effective multi-modal representation requires specialized design beyond simple feature combination.

The relationship between model capacity and performance reveals consistent patterns. Larger backbones produce substantial accuracy gains but yield inconsistent effects on stability, suggesting that representational capacity alone cannot address temporal consistency requirements.

The comparison between static priors and dynamic modeling highlights their complementary roles. While static priors significantly boost accuracy, dynamic temporal modeling proves essential for achieving temporal stability, indicating that these two approaches address distinct aspects of the mapping problem.

These findings collectively suggest that accuracy and stability are governed by different aspects of representation design. This understanding points to the need for future architectures that can simultaneously support high-fidelity spatial representation and robust temporal consistency through integrated design principles.

E SUPPLEMENTAL VISUALIZATION AND ANALYSIS

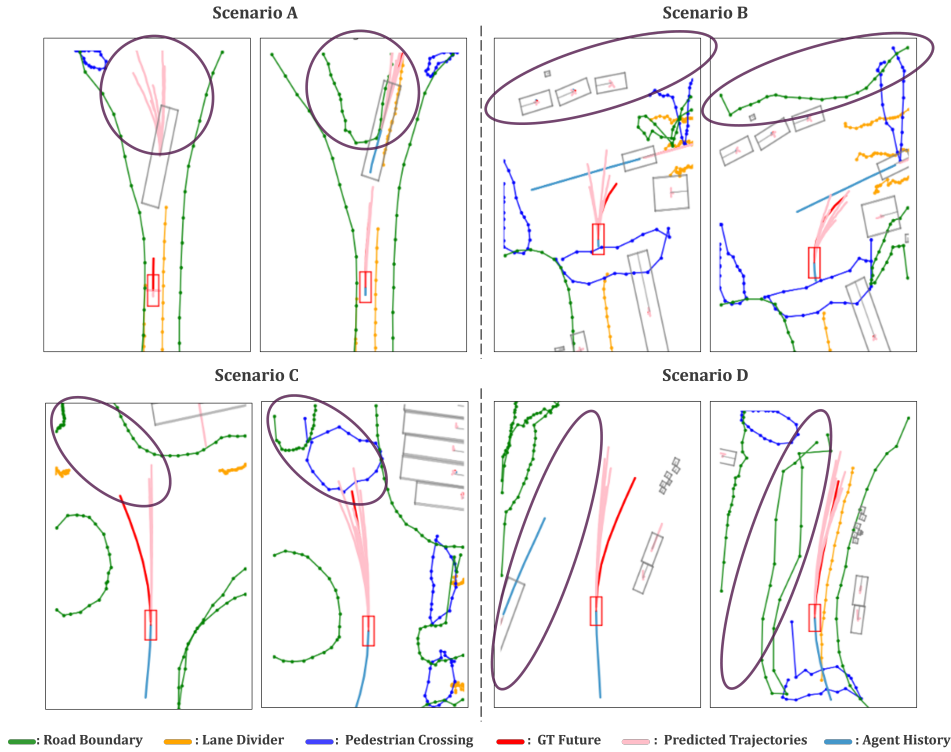


Figure 6: **Impact of Temporal Inconsistency in Map Element Presence on Downstream Tasks**

E.1 IMPACT OF UNSTABLE MAP PREDICTIONS ON DOWNSTREAM TASKS

In this part, we visualize how temporal fluctuations in online mapping predictions impact downstream tasks, as shown in Figures 6 and 7.

Figure 6 illustrates the effect of map changes between consecutive frames on these tasks.

In Scenario A at time t , the ego vehicle fails to detect an intersection ahead due to occlusion by a leading vehicle, leading it to predict the vehicle will turn left. At time $t+1$, the ego vehicle successfully identifies the intersection, resulting in a corrected prediction of the leading vehicle’s trajectory.

In Scenario B, occlusion by three vehicles directly ahead prevents the ego vehicle from detecting the road boundary behind them, causing it to plan a straight path that would collide with the curb. At time $t+1$, after moving forward, the ego vehicle observes the previously hidden map element and correctly plans a right turn.

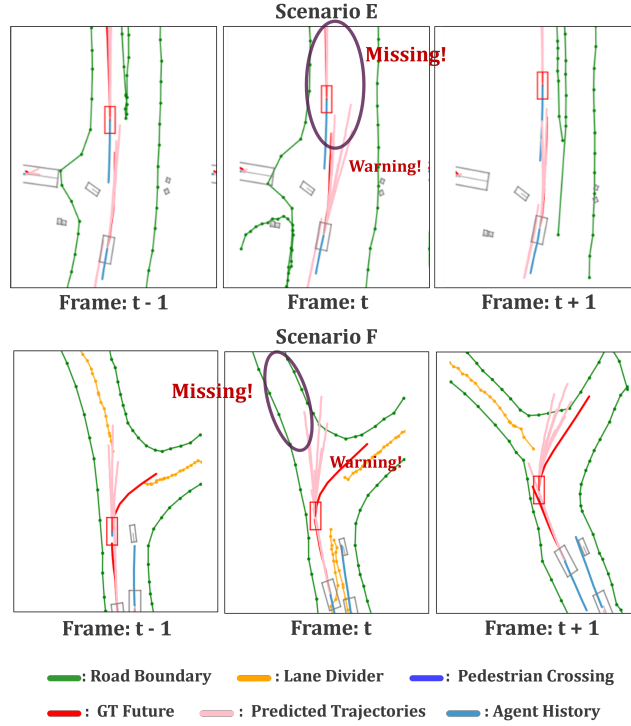


Figure 7: **Impact of Flickering in Predicted Map Elements on Downstream Tasks.**

In Scenario C at time t , the vehicle does not observe the crosswalk ahead and plans to continue straight. After advancing and detecting the map element, the ego vehicle adjusts its plan accordingly.

In Scenario D’s initial frame, occlusion by other vehicles prevents the ego vehicle from predicting the road boundary to its left, leading to a straight path plan that risks a curb collision. Upon detecting the map element on the left in the next frame, it adjusts its trajectory for safe navigation.

Based on the analysis of Figure 6, a key conclusion can be drawn: map elements are critical for autonomous systems to perform downstream tasks such as trajectory prediction and planning. Temporal instability in the perception of these elements can lead to significantly different and potentially unsafe predictions and plans.

As shown in Figure 7, we provide further visualization of how flickering map elements impact downstream tasks over time.

In Scenario E at frame $t-1$, the ego vehicle is proceeding normally. However, at frame t , a flicker occurs in the predicted road boundary to the right of the lead vehicle, caused by instability in the online mapping model. This leads the ego vehicle to perceive an opportunity to overtake on the right, resulting in a planning decision to steer right and attempt a pass. By frame $t+1$, the model correctly perceives the road boundary again, causing the ego vehicle to abort the maneuver and resume a straight path.

In Scenario F at frame $t-1$, the ego vehicle observes the lane divider ahead and plans a normal trajectory. At frame t , however, the predicted lane divider suddenly disappears, causing the planning module to become uncertain and unable to confidently decide between a lane change or continuing straight.

E.2 MAP vs. MAS

In this section, we present additional cases where mAP proves to be a misleading indicator for evaluating temporal stability, whereas our proposed mAS correctly assesses temporal stability, as demonstrated in Figure 8 and Figure 9. These examples clearly show that mAP should not be used

as a criterion for temporal stability evaluation, whereas mAS provides a more accurate assessment of temporal stability.

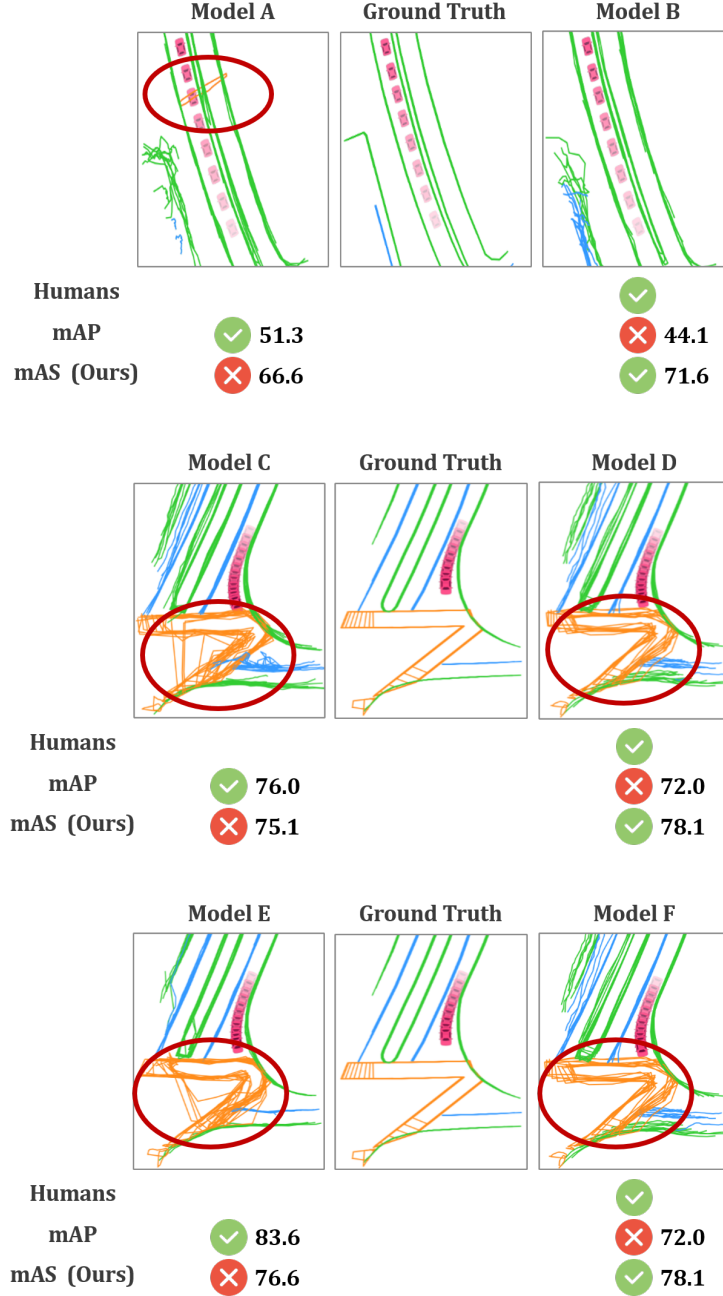


Figure 8: Evaluating trustworthiness of online mapping models using human judgment, traditional mAP, and our mAS metric.

As shown in Figure 8, model A represents the MapTR (Liao et al., 2022) model integrated with temporal features, employing a ResNet-50 backbone (He et al., 2016) and GKT encoder (Chen et al., 2022), trained for 24 epochs. Model B represent the MapTR Liao et al. (2022) model without temporal features, utilizing the same ResNet-50 backbone (He et al., 2016) and GKT encoder (Chen et al., 2022), also trained for 24 epochs. Although model A achieves a relatively high mAP of 51.3, its stability is inferior to that of model B. Specifically, in the visualization results of model A, a crosswalk flickers into view on the road, and additionally, the leftmost lane divider visualized by

model A flickers frequently. In contrast, although model B has a lower mAP compared to model A, it does not produce sudden flickering of other map elements in the middle of the road and is able to consistently predict the lane divider on the roadside in nearly every frame.

As illustrated in Figure 8, both model C and model D represent GeMap (Zhang et al., 2024b) models. Model C was trained for 110 epochs using the Swin-T (Liu et al., 2021) backbone network and the BEVFormer encoder. Model D was trained for 110 epochs using the V2-99 (DD3D) (Lee et al., 2019) backbone network and the BEVFormer encoder (Li et al., 2024). Although the performance indicators of model C are superior to those of model D, our mAS evaluation indicates that the stability of model C is relatively poor. By visually comparing the outputs of the two, we find that model C occasionally detects non-existent pedestrian crossings in individual frames, which is a manifestation of poor field stability. This observation result confirms that the stability of model C is indeed weaker than that of model D, which is consistent with the mAS evaluation result.

As depicted in Figure 8, model E represents the HRMapNet (Zhang et al., 2024a) model with a training map as initial map, employing a ResNet-50 backbone (He et al., 2016) and BevFormer encoder (Li et al., 2024), trained for 24 epochs. Model F represents the GeMap model (Zhang et al., 2024b), employing a Swin-T backbone (Liu et al., 2021) and BevFormer encoder (Li et al., 2024), trained for 110 epochs. Model E achieves a higher mAP value than model F, yet according to our mAS metric evaluation, model E exhibits inferior stability compared to model F. A visual comparison of their inference results reveals that the crosswalks predicted by model E show more pronounced geometric jitter, while other instances remain similar between the two models. This observation confirms that model E’s stability is indeed poorer than Model F’s, consistent with the assessment provided by the mAS metric.

As can be seen from Figure 9, both model G and model H represent GeMap (Zhang et al., 2024b) models. Model G was trained for 110 epochs using the Swin-T backbone network (Liu et al., 2021) and the BEVFormer encoder (Li et al., 2024), while model H was trained using the V2-99 (DD3D) backbone network (Lee et al., 2019) and the BEVFormer encoder (Li et al., 2024). It was also trained for 110 epochs. Although the performance indicators of model G are superior to those of model H, our mAS evaluation indicates that the stability of model G is relatively poor. Through visual comparison of the outputs of the two, we find that model G has significant spatial offset and morphological fluctuation in the prediction of road boundary lines in consecutive frames, which is a manifestation of poor field stability. This observation result confirms that the stability of Model G is indeed weaker than that of model H, which is consistent with the mAS evaluation result.

As illustrated in Figure 9, both model I and model J represent GeMap models (Zhang et al., 2024b). Model I was trained for 110 epochs using the Swin-T backbone network (Liu et al., 2021) and the BEVFormer encoder (Li et al., 2024). Model J was trained for 110 epochs using the V2-99 (DD3D) backbone network (Lee et al., 2019) and the BEVFormer encoder (Li et al., 2024). Model I achieves a higher mAP value than model J, yet according to our mAS metric evaluation, model I exhibits inferior stability compared to model J. Model J demonstrates superior delineation in the demarcated regions, with map instances exhibiting more precise spatial localization, while maintaining comparable performance to other models in remaining areas. This observed enhancement in output quality confirms model J’s higher stability, which aligns consistently with the mAS evaluation results.

As shown in Figure 9, model K represents the MapTR (Liao et al., 2022) model integrated with temporal features, employing a ResNet-50 backbone (He et al., 2016) and GKT encoder (Chen et al., 2022), trained for 24 epochs. Model L represent the MapTR Liao et al. (2022) model without temporal features, utilizing the same ResNet-50 backbone (He et al., 2016) and GKT encoder (Chen et al., 2022), also trained for 24 epochs. Model K achieves a higher mAP value than model L, yet according to our mAS metric evaluation, model K exhibits inferior stability compared to model L. Model L produces clearer map results in the demarcated areas with more accurate spatial positioning of map instances, while maintaining similar performance to other models in remaining regions. This demonstrates Model L’s superior stability, which is consistent with the mAS evaluation outcomes.

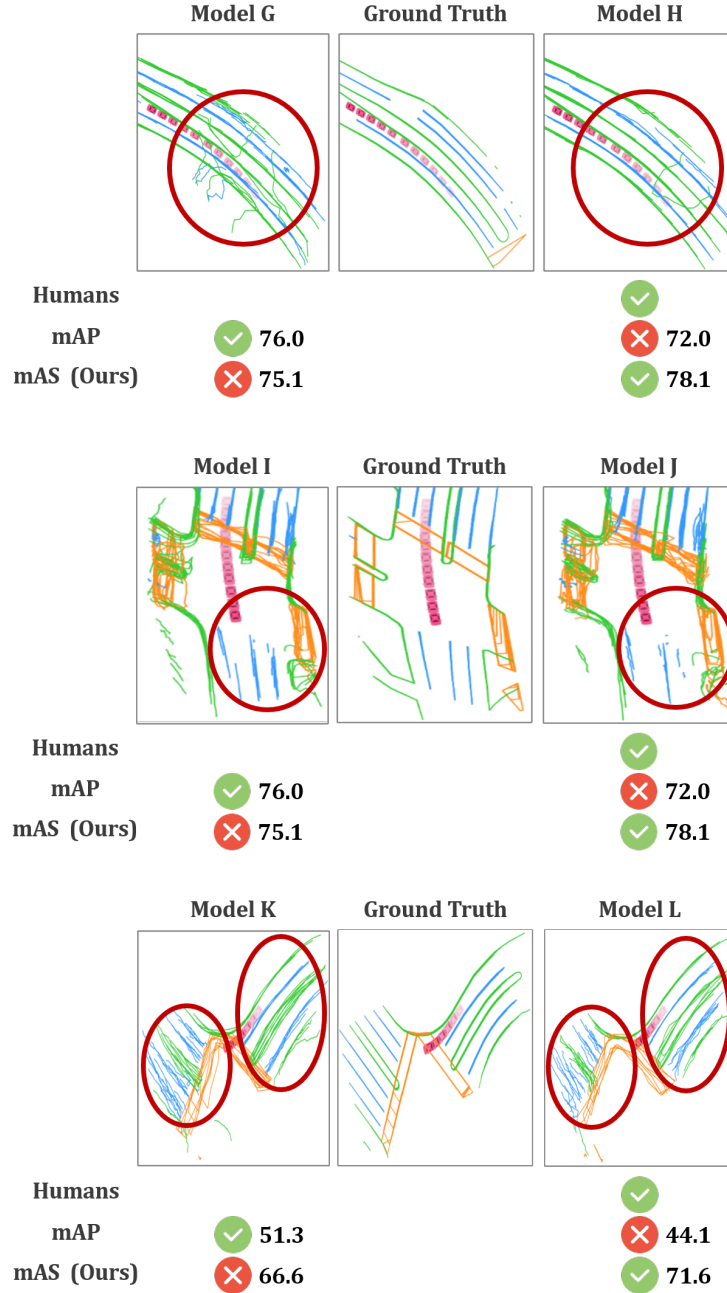


Figure 9: Evaluating trustworthiness of online mapping models using human judgment, traditional mAP, and our mAS metric.

F LIMITATIONS AND FUTURE WORK

This study presents the first dedicated benchmark for temporal stability evaluation in online HD mapping, yet several limitations indicate directions for future research. The current benchmark is constrained by the scope of existing datasets, particularly in representing complex real-world scenarios. Our evaluation primarily relies on standard driving sequences from the nuScenes dataset (Caesar et al., 2020), which lacks systematic coverage of challenging conditions such as extreme weather, adverse illumination, and intentional adversarial scenarios. Consequently, the current assessment may not fully reflect model stability under critical edge cases that are essential for safe autonomous driving.

Another limitation stems from the rapid evolution of this research field. While our benchmark encompasses 42 model variants representing major architectural paradigms, new methodologies continue to emerge at a rapid pace. The current static snapshot of model comparisons requires continuous updates to maintain relevance and comprehensiveness.

To address these limitations, we outline two primary directions for future work. First, we will establish a continuously maintained benchmark platform that systematically incorporates new research developments. This living benchmark will implement standardized evaluation protocols for emerging methodologies, ensuring fair comparisons and tracking progress over time. The platform will feature regular updates to model implementations, evaluation metrics, and dataset expansions, fostering community-wide collaboration and providing a reliable foundation for assessing advancements in temporal stability.

Second, we will expand the benchmark to include diverse challenging scenarios that better reflect real-world complexity. This expansion will incorporate data from multiple geographic regions with varying road infrastructures and traffic patterns. Specifically, we will integrate specialized datasets containing extreme weather conditions (heavy rain, snow, fog), low-light and night-time driving scenarios, and challenging urban environments with complex intersections and dense traffic. Furthermore, we will develop evaluation protocols for synthetic adversarial scenarios designed to stress-test model stability, such as sensor degradation simulations and challenging weather transitions. These enhancements will provide a more comprehensive assessment of model robustness under critical conditions.

We believe these efforts will significantly advance the development of more reliable and robust online HD mapping systems.