2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Unified Map Prior Encoder for Mapping and Planning

Zongzheng Zhang[1,2*], Sizhe Zou[1*], Guantian Zheng[1*], Zhenxin Zhu[1], Yu Gao[2], Guoxuan Chi[1], Shuo Wang[2], Yuwen Heng[2], Zhigang Sun[2], Yiru Wang[2], Hao Sun[2], Chao Ma[3], Zhen Li[4], Anqing Jiang[2†], Hao Zhao[1†]

***Abstract*— Online mapping and end-to-end (E2E) planning in autonomous driving are still largely sensor-centric, leaving rich map priors (HD/SD vector maps, rasterized SD maps, and satellite imagery) underused due to heterogeneity, pose drift, and inconsistent availability at test time. We present *UMPE*, a Unified Map Prior Encoder that can ingest any subset of four priors and fuse them with BEV features for both mapping and planning. *UMPE* has two branches. The vector encoder pre-aligns HD/SD polylines with a frame-wise SE(2) correction, encodes points via multi-frequency sinusoidal features, and produces polyline tokens with confidence scores. BEV queries then apply cross-attention with confidence bias, followed by normalized channel-wise gating to avoid length imbalance and to softly down-weight uncertain sources. The raster encoder shares a ResNet-18 backbone conditioned by FiLM (scaling/shift at every stage), performs SE(2) micro-alignment, and injects priors through zero-initialized residual fusion so the network starts from a do-no-harm baseline and learns to add only useful prior evidence. A vector-then-raster fusion order reflects the inductive bias of "geometry first, appearance second." On nuScenes mapping, *UMPE* lifts MapTRv2 from $61.5 \rightarrow 67.4$ mAP ($+5.9$) and MapQR from $66.4 \rightarrow 71.7$ mAP ($+5.3$). On Argoverse2, *UMPE* adds $+4.1$ mAP over strong baselines. *UMPE* is compositional: when trained with all priors, it outperforms single-prior models even when only one prior is available at test time, demonstrating powerset robustness. For E2E planning (VAD backbone, nuScenes), *UMPE* reduces trajectory error from $0.72 \rightarrow 0.42$ m L2 (avg. -0.30 m) and collision rate from $0.22\% \rightarrow 0.12\%$ (-0.10%), surpassing recent prior-injection methods. These results show that a unified, alignment-aware treatment of heterogeneous map priors yields better mapping and better planning. Code and dataset are released at `https://github.com/Ethan-Zheng136/UMPE`**

I. Introduction

Most prior works inject one kind of map prior [1], [2], [3], [4] or a fixed pair [5], [6], [7], [8] into sensor-centric autonomous driving pipelines, which leaves heterogeneous sources hard to combine when availability changes at test time (Tab. I). In contrast, we introduce a unified setting (Fig. 1) where a single encoder can ingest any subset of four complementary map priors—HD/SD vector maps and raster priors (rasterized SD maps, satellite imagery)—and fuse them with BEV features for both online mapping and end-to-end planning. This "powerset" formulation is, to our knowledge, the first to treat map priors as interchangeable signals that can be turned on/off without retraining. Fig. 1 visualizes this design: heterogeneous inputs enter two branches (vector and raster), are aligned and confidence-weighted, and are merged into a single BEV representation shared by mapping and planning heads.

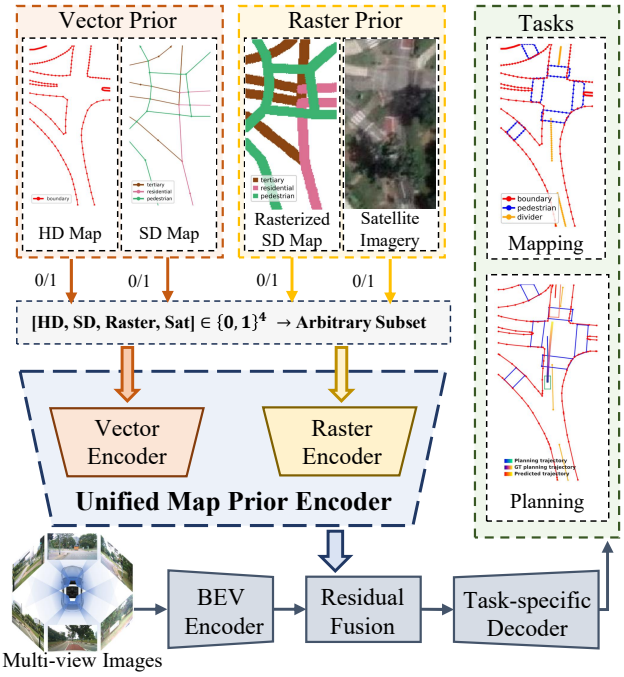

Fig. 1. **Unified Map Prior Encoder (*UMPE*).** *UMPE* ingests an arbitrary subset of four map priors—vector (HD/SD vectorized maps) and raster (rasterized SD map, satellite imagery), and processes them via a vector encoder and a raster encoder. The resulting priors are fused with BEV features, supporting both online HD mapping and end-to-end planning tasks.

Real-world deployments rarely enjoy a single perfect prior. HD vectors may exist in downtown but not in suburbs; SD map coverage is broad but coarse; satellite context is global yet misaligned; and rasterized SD provides topology hints when vectors are missing. Fig. 1 makes this concrete: each prior is togglable (0/1), so our encoder can gracefully degrade—e.g., plan with only SD+rasterized SD when HD is absent, or tighten lane geometry with HD while satellite improves crosswalk texture. The performance also improves even when all priors are present: co-training across sources teaches the model to reconcile geometry ("vector first") with appearance ("raster second"), yielding better BEV features for both mapping and planning.

Our vector encoder pre-aligns HD/SD polylines to the BEV frame via a small frame-wise SE(2) correction, encodes points with multi-frequency sinusoidal features, and produces polyline tokens with confidences. BEV queries then perform dual cross-attention to HD and SD separately to avoid softmax length imbalance, with an additive log-confidence bias

[1]Institute for AI Industry Research (AIR), Tsinghua University. [2]Bosch Corporate Research, China. [3]Shanghai Jiao Tong University. [4]Chinese University of Hong Kong, Shenzhen. *Equal contribution. †Equal advising.

TABLE I

PRIORS USED AND DOWNSTREAM TASKS ACROSS METHODS. TASK: M–MAPPING; T–TOPOLOGY; P–PLANNING

Method	Venues	HD map (vec)	SD map (vec)	Sat. Ima.	SD map (ras)	Task
SMERF [1]	ICRA 2024	✗	✓	✗	✗	M&T
SDmap-GNN [2]	IROS 2024	✗	✗	✗	✓	M&T
SDTagNet [3]	Arxiv 2025	✗	✓	✗	✗	M
SatforHDMap [4]	ICRA 2024	✗	✗	✓	✗	M
SMART [5]	ICRA 2025	✗	✓	✓	✗	M&T
SEPT [6]	RAL 2025	✗	✓	✗	✓	M&T
SATP [7]	CVPR 2025	✓	✓	✗	✗	P
PriorDrive [8]	Arxiv 2024	✓	✓	✗	✗	M
UMPE (Ours)	—	✓	✓	✓	✓	M&P

that down-weights uncertain vectors. A presence-normalized, channel-wise gate mixes sources so that, when one prior is missing (the dashed 0/1 switches in Fig. 1), its channels do not suppress others. This path provides metrically precise lane geometry to downstream heads.

Fig. 1's raster encoder shares a ResNet-18 backbone across satellite and rasterized SD inputs and conditions it with FiLM at every stage for source awareness. We estimate a lightweight SE(2) micro-alignment to correct residual pose/scale offsets to the BEV lattice. Priors are then injected through a zero-initialized residual pathway into the BEV tokens, with LayerNorm and a learnable scale, implementing a do-no-harm baseline that only adds evidence the task demands. A presence-normalized gate (as in the vector path) selects between satellite and rasterized SD features.

The two branches are composed in a vector-then-raster order that encodes an inductive bias—"geometry first, appearance second." This sequencing preserves clean queries for vector attention and lets raster cues refine dense context afterward. To make the model robust to missing inputs, we introduce SourceDropout that randomly disables sources during training. Together, confidence-biased attention, zero-init residual fusion, and gated mixing yield a single encoder that generalizes across all prior combinations without per-subset retraining.

We validate the unified design in Fig. 1 on two fronts. For online mapping, inserting *UMPE* into strong BEV baselines (e.g., MapTRv2 [9] and MapQR [10]) boosts mAP on nuScenes [11] and Argoverse 2 [12], with per-class gains matching intuition: vector priors enhance boundaries/dividers, while raster priors sharpen pedestrian crossings. For E2E planning, we plug *UMPE* into a VAD-style backbone [13], reducing average trajectory $\mathbf{L}_2$ and collision rate versus prior-injection methods. Ablations validate each design choice, and robustness tests show that a model trained with all priors still outperforms single-prior models even when only one prior is available at test time—the practical payoff of our proposed unified setting.

II. RELATED WORK

A. Online HD Mapping and Motion Forecasting

Online HD mapping has evolved from BEV-based frameworks [14] to end-to-end vectorized prediction with set-structured queries [15], [16], [9], later extended to temporal streaming and instance-consistent tracking [17], [18] and hybrid raster-vector models [19]. In parallel, detection and topology reasoning [20], [21] have been standardized by OpenLane-V2 [22], with graph-based [23], lightweight MLP [24] approaches and lane-segment perception [25] enriching the task definition.

Motion forecasting has similarly progressed from goal/intention-driven models [26], [27], [28], [29] toward tighter coupling with online maps, e.g., direct BEV feature attention [30] and explicit map-uncertainty modeling [31]. Yet most pipelines remain *sensor-centric* (camera/LiDAR), leaving complementary map priors underexploited.

B. Map Prior for Online Mapping

Recent work increasingly augments online mapping with heterogeneous priors—prebuilt HD maps, standard-definition (SD) maps, satellite imagery, and neural radiance fields [32], [33]—while tackling their misalignment with onboard perception and representation gaps across modalities.

HD/SD maps provide vector road skeletons, which, when encoded and fused with onboard features, improve mapping and topology [1], [3], [2]. Satellite imagery contributes global, long-range context with feature-level fusion and BEV-frame alignment [4]. Beyond single sources, mixed priors yield further gains: HD+SD for far-seeing generation [34], SD+satellite priors learned offline then plugged into topology heads [5], SD (vector)+rasterized SD via dual-branch fusion [6], and explicit HD–SD alignment beneficial to mapping and planning [7]. A unified vector prior encoder pushes toward *map-type–agnostic* consumption by embedding SD/HD into a shared space [8].

Accordingly, we further propose a single unified map prior encoder that jointly learns from vectorized HD/SD, satellite, and rasterized SD priors with alignment-aware features.

C. Map Prior for End-to-End Autonomous Driving

E2E driving spans planning-oriented multi-task models [35], [13], [36], [37], [38], [39], generative trajectory policies [40], [41], [42], world-model or sparse-token formulations [43], [44], and specialized designs with temporal memory or language guidance [45], [46], [47].

Within this landscape, explicit priors have shown clear benefits for E2E planning: STAP [7] aligns SD–HD maps and improves closed-loop planning when the aligned priors are fed into VAD-style stacks; GaussianFusion [48] uses a gaussian-based multi-sensor fusion framework, offering a compact alternative to dense BEV features. Extending this direction, we present a unified encoder that integrates well-aligned priors into a single representation for E2E planning.

III. METHOD

In this section, we first describe how four map priors are obtained (Sec. III-A). We then present the detailed architecture of the vector encoder (Sec. III-B) and the raster encoder (Sec. III-C). Finally, we show the unified fusion that integrates the two encoder outputs with BEV tokens (Sec. III-D), enabling *UMPE* to handle any subset of priors.

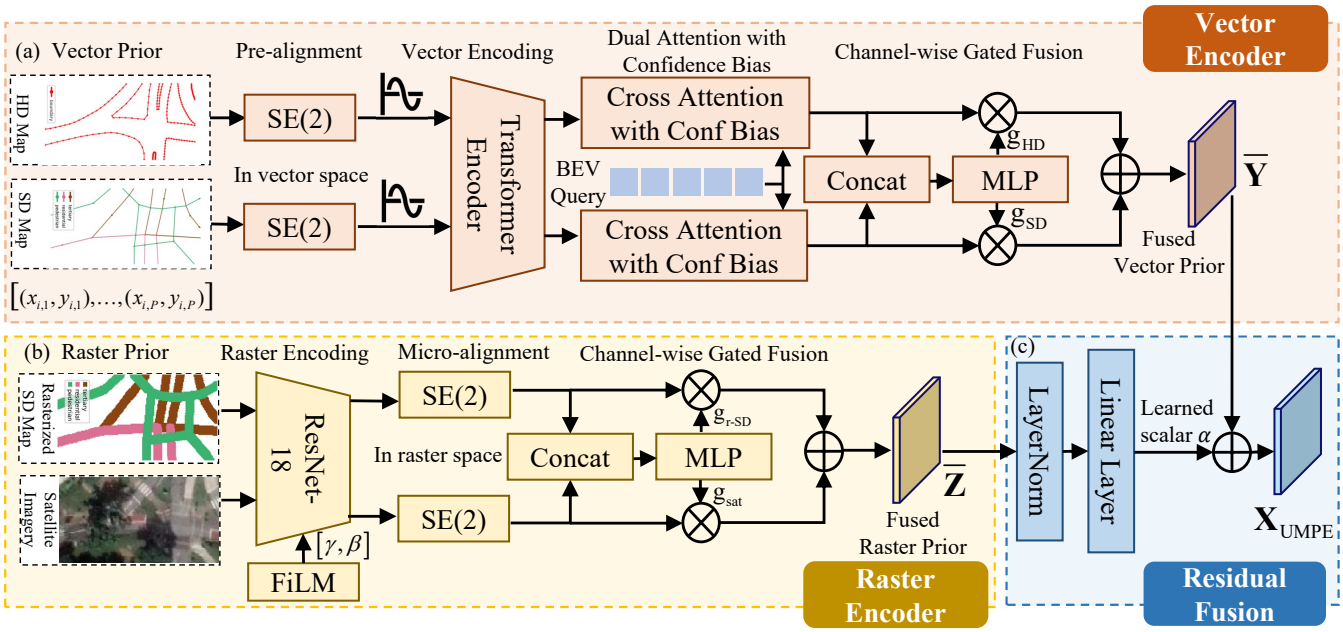

Fig. 2. **Unified Map Prior Encoder (*UMPE*) architecture.** (a) **Vector Encoder**: HD/SD polylines are SE(2) pre-aligned and encoded; BEV queries attend to each source with confidence-biased dual cross-attention. Presence-normalized, channel-wise gating mixes sources to produce fused vector tokens $\bar{\mathbf{Y}}$. (b) **Raster Encoder**: rasterized SD map and satellite imagery pass through a shared FiLM-conditioned ResNet, then undergo SE(2) micro-alignment in raster space; channel-wise gating yields fused raster tokens $\bar{\mathbf{Z}}$. (c) **Residual fusion**: $\bar{\mathbf{Y}}$ and $\bar{\mathbf{Z}}$ are inhected with a learned scalar α, producing $\mathbf{X}_{\text{UMPE}}$.

A. Map Priors Preparation

Vectorized HD and SD Map. We retrieve SD maps from OpenStreetMap [49]. For each frame, given the ego GPS and heading orientation, we query a local OSM region, project coordinates to a local Cartesian frame, apply a rigid transform to the ego frame (rotation by yaw and translation by ego position), and crop an ego-centric BEV window of 60m×30m, the same as the spatial extent covered by $\mathbf{F}_{\text{BEV}}$. OSM roadways are parsed as polylines and annotated with eight one-hot classes: `highway`, `primary`, `secondary`, etc. The dataset-provided HD vectors undergo the same projection, rigid transform, and crop. Both SD and HD coordinates are expressed in the exact coordinate system of the $\mathbf{F}_{\text{BEV}}$. To standardize density and batching, every polyline is uniformly resampled to the same number of points.

Satellite Imagery. We fetch satellite tiles via the Mapbox Raster Tiles API given the ego GPS. We compute the tile indices at a zoom level chosen to match the pixel–meter resolution of $\mathbf{F}_{\text{BEV}}$. All tiles covering the 60m × 30m area around the ego are downloaded, then rotated by the ego yaw. The result is an RGB image $\mathbf{I}^{\text{sat}} \in \mathbb{R}^{H\times W\times 3}$ for every frame; H and W equal the BEV canvas used by the network.

Rasterized SD Map. Given the SD map, we render each of the eight SD categories with a fixed color. This produces an RGB raster $\mathbf{I}^{\text{sd}} \in \mathbb{R}^{H\times W\times 3}$ used as the rasterized SD prior.

B. Vector Encoder for Vectorized HD/SD Map Priors

Vector map priors provide exact lane geometry but arrive with small pose drift and a variable number of polylines that do not align to a fixed BEV grid. We therefore (i) correct coordinates by a small frame-wise SE(2) **motion**, (ii) encode each polyline as a fixed-width token using sinusoidal point features plus semantics (category and source one-hots), then apply a transformer to obtain tokens and confidence, and (iii) fuse into BEV by **dual cross-attention** with **confidence bias** and **presence normalized gating**, so that BEV queries selectively pull geometrically relevant, reliable vectors while softly suppressing uncertain or absent sources (Fig. 2 (a)).

Coordinate-level SE(2) Pre-alignment. To compensate for small pose drift between the visual BEV frame and the vector priors, we estimate a *frame-wise* rigid correction for each available source $\text{src} \in \{\text{HD}, \text{SD}\}$. For source src, the resampled polyline set is $\mathscr{P}^{\text{src}} = \{\mathbf{p}_i^{\text{src}}\}_{i=1}^{N_{\text{src}}}$, $\mathbf{p}_i^{\text{src}} = \big[(x_{i,1}, y_{i,1}), \ldots, (x_{i,P}, y_{i,P})\big]$, P=11, where N_{src} is the number of polylines in the current frame. We predict a small rigid motion $(\Delta x, \Delta y, \Delta\theta)$ and correct every point:

$$\tilde{\mathbf{p}}_i = \mathbf{R}(\Delta\theta)\mathbf{p}_i + \mathbf{T},\ \mathbf{R}(\Delta\theta) = \begin{bmatrix}\cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta\end{bmatrix},\ \mathbf{T} = [\Delta x, \Delta y]^\top. \tag{1}$$

We regularize its magnitude $\mathscr{L}_{\text{se2}}^{\text{vec}} = \lambda_t \|\mathbf{T}\|_2^2 + \lambda_r(\Delta\theta)^2$ to keep corrections small.

Vector Encoding and Tokenization. Each corrected point $(\tilde{x}, \tilde{y})$ is mapped by a multi-frequency sinusoidal embedding $\phi(\tilde{x}, \tilde{y}) = \big[\sin(\omega_k\tilde{x}), \cos(\omega_k\tilde{x}), \sin(\omega_k\tilde{y}), \cos(\omega_k\tilde{y})\big]$, with geometrically spaced ω_k. For polyline i, we *flatten* its P point encodings and concatenate a one-hot category $\mathbf{e}_i^{\text{cat}} \in \{0,1\}^{K_{\text{cat}}}$ and one-hot source $\mathbf{e}_i^{\text{src}} \in \{0,1\}^2$: $\mathbf{z}_i = \Big[\text{vec}\big(\phi(\tilde{\mathbf{x}}_{i,1}), \ldots, \phi(\tilde{\mathbf{x}}_{i,P})\big)\ ;\ \mathbf{e}_i^{\text{cat}}\ ;\ \mathbf{e}_i^{\text{src}}\Big]$. We then apply a 6-layer transformer encoder: $\mathbf{T}_{\text{vec},i}^{\text{src}} = \text{TrEnc}^{(6)}(\mathbf{z}_i)$, and predict a sigmoid confidence for each token $U_i^{\text{src}} \in (0,1)$.

Dual Cross-Attention with Confidence Bias. Let BEV feature $\mathbf{F}_{\text{BEV}} \equiv \mathbf{X} \in \mathbb{R}^{B\times(HW)\times C}$ be the BEV tokens (queries), and let $\mathbf{T}_{\text{vec}}^{\text{src}} \in \mathbb{R}^{B\times N_{\text{src}}\times C}$ be the contextualized polyline tokens from source $\text{src} \in \{\text{HD}, \text{SD}\}$. We com-

pute multi-head projections $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K}^{\text{src}} = \mathbf{T}^{\text{src}}_{\text{vec}}\mathbf{W}_K$, $\mathbf{V}^{\text{src}} = \mathbf{T}^{\text{src}}_{\text{vec}}\mathbf{W}_V$, where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times(hd)}$. Let $\tilde{\mathbf{U}}^{\text{src}} = \text{clamp}(\mathbf{U}^{\text{src}}, \varepsilon, 1) \in (0,1]^{B\times N_{\text{src}}}$ be the confidence with lower bound ε. For each source, we fuse separately to avoid length imbalance in a single softmax:

$$\mathbf{Y}^{\text{src}} = \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{K}^{\text{src}})^\top}{\sqrt{d}} + \log\tilde{\mathbf{U}}^{\text{src}}\right)\mathbf{V}^{\text{src}} \quad \in \mathbb{R}^{B\times(HW)\times C}, \tag{2}$$

where $\log\tilde{\mathbf{U}}^{\text{src}}$ is broadcast along query positions and heads to form a $(B\times HW\times N_{\text{src}})$ bias matrix. The additive $\log\tilde{\mathbf{U}}^{\text{src}}$ acts as a multiplicative prior inside the softmax, privileging reliable polylines while softly suppressing uncertain ones.

Presence-normalized Channel-wise Gated Fusion. After dual cross-attention, we obtain $\mathbf{Y}^{\text{HD}}$, $\mathbf{Y}^{\text{SD}}$; we concatenate along channels and pass through a lightweight network to produce logits $\mathbf{L} \in \mathbb{R}^{B\times(2C)}$. Splitting $\mathbf{L}$ into $\mathbf{L}_{\text{HD}}, \mathbf{L}_{\text{SD}} \in \mathbb{R}^{B\times C}$, we compute per-channel, presence-normalized gates via a softmax across sources:

$$\begin{aligned}\left[g_{\text{HD}}, g_{\text{SD}}\right] = \text{softmax}\Big(\big[&\mathbf{L}_{\text{HD}} + \log(\pi_{\text{HD}}+\varepsilon),\\ &\mathbf{L}_{\text{SD}} + \log(\pi_{\text{SD}}+\varepsilon)\big]\Big).\end{aligned} \tag{3}$$

where $\pi = [\pi_{\text{HD}}, \pi_{\text{SD}}] \in \{0,1\}^2$ is the source-presence mask for the current frame (π_s=1 if source s is available, 0 otherwise). We then take the gated mix as the fused vector-prior BEV tokens:

$$\bar{\mathbf{Y}} = g_{\text{HD}} \odot \mathbf{Y}^{\text{HD}} + g_{\text{SD}} \odot \mathbf{Y}^{\text{SD}} \in \mathbb{R}^{B\times(HW)\times C}. \tag{4}$$

C. Raster Encoder for Satellite and Rasterized SD Priors

We introduce a source-aware raster encoder that ingests two raster priors. The module has three stages: **source-aware encoding** (shared backbone with FiLM conditioning), **SE(2) micro-alignment** to correct residual pose/scale mismatches, and **gated fusion** to produce fused raster tokens $\bar{\mathbf{Z}}$ (Fig. 2(b)).

Source-aware Backbone with FiLM. Both sources $\mathbf{I}^{sat}, \mathbf{I}^{sd} \in \mathbb{R}^{H\times W\times 3}$ are processed by a shared ResNet-18 backbone [50] equipped with every-stage FiLM conditioning. Let $\mathbf{A} \in \mathbb{R}^{B\times C\times H\times W}$ be an intermediate activation and $c \in \mathbb{R}^D$ a learned source embedding (one-hot identity passed through an MLP). FiLM computes channel-wise affine parameters:

$$[\gamma, \beta] = \mathbf{W}c + \mathbf{b} \in \mathbb{R}^{2C}, \text{FiLM}(\mathbf{A}, c) = (1+\gamma)\odot\mathbf{A} + \beta, \tag{5}$$

broadcast over spatial dimensions. A 1×1 projection followed by resizing produces source-aligned BEV feature:

$$\begin{aligned}\mathbf{F}^{\text{src}}_{\text{ras}} = \text{resize}\Big(&\text{Conv}_{1\times1}\big(\text{FiLM}(\text{Res}(\mathbf{I}^{\text{src}}), c_{\text{src}})\big),\\ &(H,W)\Big) \in \mathbb{R}^{B\times C\times H\times W}, \quad \text{src} \in \{\text{sat}, \text{r-sd}\}.\end{aligned} \tag{6}$$

Then, we flatten feature maps to BEV tokens using FlattenHW$(\cdot)$: $\mathbf{T}^{\text{src}}_{\text{ras}} = \text{FlattenHW}(\mathbf{F}^{\text{src}}_{\text{ras}}) \in \mathbb{R}^{B\times(HW)\times C}$.

SE(2) **Micro-alignment.** We keep the same objective and regularize as in Sec. III-B but regress the pose from raster features plus the BEV reference instead of polylines to predict $(\Delta x, \Delta y, \Delta\theta)$. With meters-per-pixel $(m^{\text{src}}_x, m^{\text{src}}_y)$, we form the normalized affine for `grid_sample`:

$$t_x = \frac{2}{W-1}\frac{\Delta x}{m_x}, t_y = \frac{2}{H-1}\frac{\Delta y}{m_y}, \Theta = \begin{bmatrix}\cos\Delta\theta & -\sin\Delta\theta & t_x\\ \sin\Delta\theta & \cos\Delta\theta & t_y\end{bmatrix}. \tag{7}$$

The aligned prior feature and tokens are

$$\widetilde{\mathbf{F}}^{\text{src}}_{\text{ras}} = \text{grid\_sample}\big(\mathbf{F}^{\text{src}}_{\text{ras}}, G(\Theta)\big), \widetilde{\mathbf{T}}^{\text{src}}_{\text{ras}} = \text{FlattenHW}\Big(\widetilde{\mathbf{F}}^{\text{src}}_{\text{ras}}\Big). \tag{8}$$

Presence-normalized Channel-wise Gated Fusion. We reuse the per-channel, presence-normalized softmax gates of Eq. (3) to weight the two raster streams after SE(2):

$$\bar{\mathbf{Z}} = g_{\text{sat}} \odot \widetilde{\mathbf{T}}^{\text{sat}}_{\text{ras}} + g_{\text{r-sd}} \odot \widetilde{\mathbf{T}}^{\text{r-sd}}_{\text{ras}} \in \mathbb{R}^{B\times(HW)\times C}, \tag{9}$$

yielding fused raster prior $\bar{\mathbf{Z}}$.

D. Four Map Priors Residual Fusion

We fuse the four priors in a vector-first, raster-second sequence that mirrors their roles: vectors provide precise geometry and topology; rasters supply dense appearance.

Vector stage (geometry first): As Sec. III-B mentioned, HD/SD vector priors are encoded into polyline tokens with confidences. BEV queries $\mathbf{X}$ attend to each source separately via dual cross-attention (Eq. (2)). Then we apply a channel-wise gate fusion, yielding $\bar{\mathbf{Y}} \in \mathbb{R}^{B\times(HW)\times C}$ (Fig. 2 (a)).

Raster stage (dense refinement): Unlike the vector path, we **do not** use BEV$\leftrightarrow$ prior cross-attention here: raster priors and the online BEV are both **image-like, pixel-aligned** on the BEV lattice, so a zero-initialized fusion preserves spatial locality and avoids the length-imbalance issues of attention over dense grids (Fig. 2 (c)):

$$\mathbf{X}_{\text{UMPE}} = \text{LN}(\bar{\mathbf{Y}}) + \alpha\,\mathbf{W}\,\text{LN}(\bar{\mathbf{Z}}), \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{C\times C}$ is a linear layer initialized to zero, α is a learned scalar scheduled from 0 to ~ 0.6. LayerNorm aligns token scales to avoid one source dominating the fusion. The zero-initialized residual ensures the network starts from the BEV baseline and learns to add prior information only where it improves the mapping objective. The final fused representation $\mathbf{X}_{\text{UMPE}} \in \mathbb{R}^{B\times(HW)\times C}$ is then fed to the task-specific decoder for online mapping or end-to-end planning.

IV. Experiment

We evaluate *UMPE* under three hypotheses: (H1) **Mapping generality**—*UMPE* yields consistent gains across datasets and baselines (Sec. IV-B); (H2) **Planning benefit**—*UMPE* improves end-to-end planning (Sec. IV-C); (H3) **Module effectiveness & any-subset robustness**—each design choice in *UMPE* contributes measurably, and *UMPE* handles arbitrary prior subsets (Sec. IV-D).

A. Experimental Setup

Datasets and Metrics. We evaluate online HD map construction on nuScenes [11] and Argoverse2 [12] using their official splits. For each frame, we extract four priors in an ego-centric BEV crop of 60m×30m: vectorized HD map, vectorized SD map, satellite imagery, and rasterized SD map.

TABLE II

MAPPING RESULTS ON NUSCENES VALIDATION DATASET. PRIORS: VP–VECTORIZED HD/SD MAP PRIORS; RP–RASTERIZED SD MAP/SATELLITE IMAGERY PRIORS. BACKBONE: R–RESNET; T–TRANSFORMER. FPS IS MEASURED ON A SINGLE RTX 3090. SINCE NUSCENES LACKS BULIT-IN HD MAPS, WE FOLLOW [51] TO CREATE AN HD MAP BY RETAINING ONLY ROAD BOUNDARIES AND REMOVING PEDESTRIAN CROSSINGS AND DIVIDERS.

Method	VP	RP	Backbone	Epoch	$\mathbf{AP}_{\text{ped}}$	$\mathbf{AP}_{\text{div}}$	$\mathbf{AP}_{\text{bou}}$	**mAP**	#Param.(M)	FPS ↑
VectorMapNet [15]			R50	110	42.5	51.4	44.1	46.0	36.7	16.17
MapTR [16]			R50	24	46.3	51.5	53.1	50.3	36.3	15.10
StreamMapNet [17]			R50	24	64.1	58.2	59.4	60.6	57.7	10.16
MGMap [52]			R50	30	61.8	65.0	67.8	64.8	55.9	11.60
MapTRv2 [9]			R50	24	59.8	62.4	62.4	61.5	40.3	11.57
+ SMERF [1]	✓		R50&T	24	60.4	63.4	63.2	62.3 (+0.8)	48.6	10.32
+ PriorDrive [8]	✓		R50&T	24	61.5	65.6	68.3	65.1 (+3.6)	43.4	–
+ SATP [7]	✓		R50&T	24	–	–	–	61.9 (+0.4)	40.6	–
+ Vector Encoder (Ours)	✓		R50&T	24	63.0	68.6	68.8	66.8 (+5.3)	43.5	11.37
+ SatforHDmap [4]		✓	R50&R18	24	63.6	63.1	64.4	63.7 (+2.2)	58.0	10.32
+ Raster Encoder (Ours)		✓	R50&R18	24	65.7	66.1	68.2	66.7 (+5.2)	51.8	11.36
+ UMPE (Ours)	✓	✓	R50&T&R18	24	66.6	67.2	68.2	67.4 (+5.9)	54.8	10.98
MapQR [10]			R50	24	63.4	68.0	67.7	66.4	125.0	7.72
+ Vector Encoder (Ours)	✓		R50	24	66.8	73.6	71.4	70.6 (+4.2)	128.3	7.26
+ Raster Encoder (Ours)		✓	R50	24	68.7	73.4	72.2	71.4 (+5.0)	136.6	7.35
+ UMPE (Ours)	✓	✓	R50	24	**69.0**	**73.8**	**72.3**	**71.7** (+5.3)	139.5	7.07

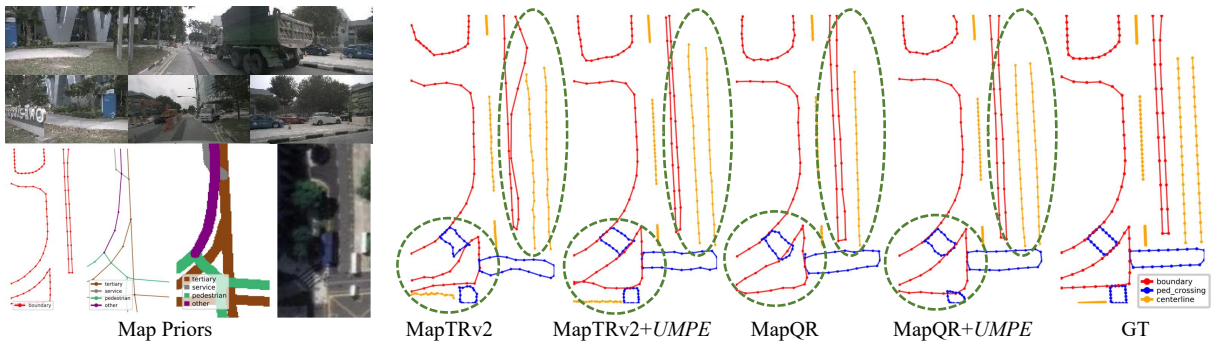

Fig. 3. **Online mapping visualization on nuScenes**. Adding *UMPE* to both MapTRv2 [9] and MapQR [10] produces more accurate maps, especially in the green-highlighted regions: baselines show broken pedestrian crossings, kinked boundaries and missing dividers; *UMPE* straightens, restores them.

Following standard protocols for vectorized mapping, we report **mAP** computed from average precision over Chamfer distance thresholds $\tau \in \{0.5, 1.0, 1.5\}$m between predicted vectors and ground-truth map elements.

For end-to-end autonomous driving, we evaluate on nuScenes [11]. The same four priors are extracted per frame and injected into the policy via our unified map prior encoder. We report two standard metrics: **L2 error**—the mean Euclidean distance between the planned and ground-truth ego trajectories and **collision rate**—the frequency of rollouts where the ego trajectory collides with other agents.

Implementation Details. We train *UMPE* from scratch with a two-stage curriculum and SourceDropout. We randomly drop one source with a 0.3 probability in each encoder. Stage 1: we optimize the full model with separate parameter groups—higher LR for the prior branches and a lower LR for the BEV encoder and map decoder. The residual scales α are linearly ramped from 0 to 0.2. Stage 2: we reduce LRs and relax α to 0.6, keeping fusion projections zero-initialized so the residual paths remain "do-no-harm" early. For mapping, we train 24 epochs on nuScenes and 6 epochs on Argoverse 2 using 4×RTX 3090; for planning, we train 60 epochs on nuScenes using 8×A800.

B. Online Mapping Results

Quantitative Results. On the NueScens [11] dataset, we adopt MapTRv2 [9] as the baseline and compare against representative methods that inject either vector priors or raster priors (Tab. II). Our *UMPE* consistently surpasses others when using a single prior encoder and achieves the highest mAP when both prior encoders are enabled. Despite the accuracy gains, runtime remains comparable to prior work. To assess scalability, we also plug *UMPE* into the stronger MapQR [10] baseline and observe consistent improvements, indicating that our modules are model-agnostic and transferable across backbones. On Argoverse 2 [12], *UMPE* again improves over the baseline and clearly outperforms other prior-fusion methods, confirming its robustness across datasets and scalable fusion (Tab. III).

Per-class trends are consistent with prior semantics. The vector encoder chiefly improves divider and boundary AP since its accurate lane geometry. The raster encoder contributes most on pedestrian crossing AP, as satellite input provides top-down texture for crosswalk patterns.

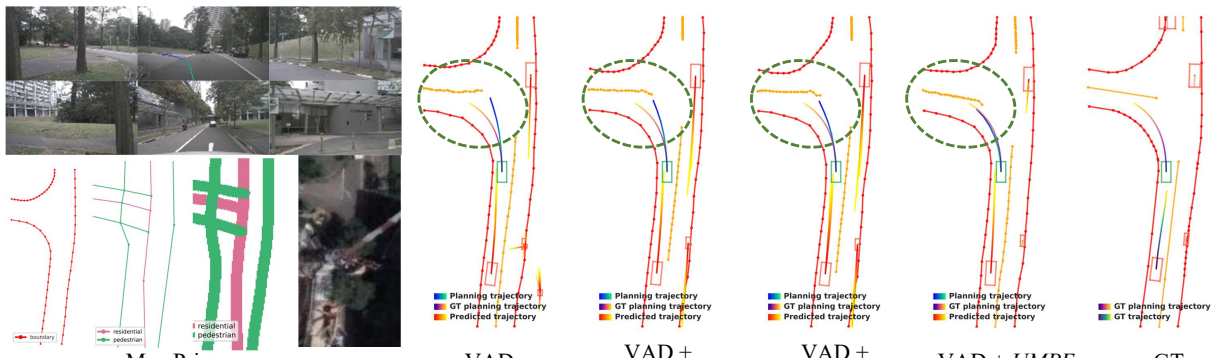

Fig. 4. **End-to-end Planning visualization on nuScenes.** The ego vehicle is turning left. VAD without priors drifts toward the oncoming lane; adding the vector encoder or raster encoder improves lane adherence but leaves lateral error, while VAD+*UMPE* produces a trajectory tightly overleaps the GT.

TABLE III

MAPPING RESULTS ON ARGOVERSE 2 [12] VALIDATION DATASET.

Method	AP_{ped}	AP_{div}	AP_{bou}	mAP
VectorMapNet [15]	35.6	34.9	37.8	36.1
MapTR [16]	48.1	50.4	55.0	51.1
StreamMapNet [17]	56.9	55.9	61.4	58.1
MGMap [52]	52.8	67.5	68.1	62.8
MapQR [10]	60.1	71.2	66.2	65.9
MapTRv2 [9]	60.7	68.9	64.5	64.7
+ SMERF [1]	60.5	69.1	67.8	65.8 (+1.1)
+ SDTagNet [3]	62.1	66.1	70.7	66.3 (+1.6)
+ Vector Encoder (Ours)	62.4	67.7	**70.9**	67.0 (+2.3)
+ Raster Encoder (Ours)	63.7	**72.6**	67.4	67.9 (+3.2)
+ UMPE (Ours)	**63.8**	**72.6**	70.0	**68.8** (+4.1)

Qualitative Results. As shown in Fig. 3, *UMPE* regularizes the baseline geometry: fragmented pedestrian crossings become closed and well-shaped, lane boundaries and dividers straighten and align with the road layout. These corrections appear consistently across backbones, bringing the predictions noticeably closer to the ground truth.

C. End-to End Autonomous Driving Results

Quantitative Results. We have shown that incorporating map priors into a unified encoder significantly improves mapping performance. A natural next step is to ask whether these improvements in perception and map understanding can transfer to the planning domain. We therefore plug the *UMPE* into VAD [13] as an auxiliary prior-fusion branch that augments the BEV feature before the planning decoder. We compare *UMPE* against three representative prior-injection methods [7], [34], [8] with the same VAD backbone. Tab. IV shows that our vector encoder already yields a large $\mathbf{L}_2$ drop, while the raster encoder also helps. Combining both in *UMPE* gives the strongest improvement ($\mathbf{L}_2$ avg -0.30m vs. VAD and Collision Rate -0.10% vs. VAD), outperforming all prior-injection methods. The effect is mainly because vector priors provide metrically accurate geometry selected via confidence-biased cross-attention, while raster priors add dense drivable context; the two are complementary.

Qualitative Results. Fig. 4 corroborates the quantitative results. We attribute these gains to stronger BEV perception and mapping. Once these map priors are fused, the planner "sees" a more structured map, leading to lower trajectory error and collisions. That is better mapping, better planning.

TABLE IV

PLANNING RESULTS ON THE NUSCENES [11] VALIDATION DATASET.

Method	L2 (m) ↓				Col. Rate (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
UniAD [35]	0.45	0.70	1.04	0.73	0.62	0.58	0.63	0.61
GenAD [42]	0.28	0.49	0.78	0.52	0.08	0.14	0.34	0.19
SparseDrive [43]	0.29	0.58	0.96	0.61	**0.01**	**0.05**	0.18	**0.08**
BEVPlanner [36]	0.27	0.54	0.90	0.57	–	–	–	–
Epona [44]	0.61	1.17	1.98	1.25	**0.01**	0.22	0.85	0.36
MomAD [45]	0.31	0.57	0.91	0.60	**0.01**	**0.05**	0.22	0.09
BridgeAD [46]	0.29	0.57	0.92	0.59	**0.01**	**0.05**	0.22	0.09
DiffusionDrive [40]	0.27	0.54	0.90	0.57	0.03	**0.05**	**0.16**	**0.08**
VAD [13]	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
+ SATP [7]	0.39	0.62	0.94	0.65 (-0.07)	0.14	0.21	0.42	0.26 (+0.04)
+ PmapNet [34]	0.35	0.65	1.05	0.68 (-0.04)	0.37	0.54	1.06	0.66 (+0.44)
+ PriorDrive [8]	0.35	0.68	1.16	0.73 (+0.01)	0.26	0.46	1.12	0.61 (+0.39)
+ Vector Enc (Ours)	0.25	0.45	0.74	0.48 (-0.24)	0.09	0.11	0.31	0.17 (-0.05)
+ Raster Enc (Ours)	0.28	0.53	0.83	0.55 (-0.17)	0.09	0.13	0.35	0.19 (-0.03)
+ UMPE (Ours)	**0.21**	**0.39**	**0.65**	**0.42** (-0.30)	0.03	0.08	0.25	0.12 (-0.10)

D. Ablation Study

Vector Encoder Modules. To verify the effectiveness of each component in our vector encoder, we conduct a step-by-step ablation on the MapTRv2 trained for 24 epochs (Tab. V). First, for tokenization, **Sine-PE** outperforms raw (x,y)→MLP (2 vs. 3), indicating that multi-frequency point encoding better preserves fine lane geometry. With tokenization fixed, switching the fusion strategy from concatenation + single cross-attention to **dual cross-attention** yields a clear gain (2 vs. 4), showing that separating HD/SD avoids length imbalance in the softmax. Adding the **confidence bias** inside attention brings further improvement (4 vs. 5) by down-weighting uncertain polylines. Incorporating presence-normalized **gating** adds another gain (5 vs. 6). Finally, appending the **SE(2) pre-alignment** on top performs best (6 vs. 7) by removing small pose drift before tokenization.

Raster Encoder Modules. On MapTRv2, we ablate the raster encoder under the same backbone and schedule (Tab. VI). First, applying FiLM at **every** ResNet stage outperforms late-stage FiLM (2 vs. 3), indicating that distributed modulation better handles domain shift across layers. With SH-FiLM fixed, **feature-based gating** (Feagate) is clearly preferable to conditioning-vector gating (Congate)(4 vs. 5), showing that gates should depend on encoded raster evidence rather than metadata alone. The zero-initialized **residual** injection is the largest contributor to accuracy (2 vs. 6). Finally, adding **SE(2) micro-alignment** on the full setting polishes residual pose/scale mismatches (7 vs. 8). Overall,

TABLE V

ABLATIONS ON **VECTOR-PRIOR FUSION**. **MLP**: REPLACE SINE-PE WITH RAW (x, y) TO A POLYLINE MLP. **SINGLE-ATTN**: CONCAT HD/SD THEN A SINGLE CROSS-ATTENTION. (+Δ) DENOTES THE ABSOLUTE GAIN OVER THE BASELINE [9]. ALL OTHERS MODULES ARE DEFINED IN SEC. III-B.

ID	Vector Encoder Modules							AP ↑				#Param.(M)
	Sine-PE	MLP	Single-Attn	Dual-Attn	ConfBias	Gating	SE(2)	ped.	div.	bou.	mean	
1								59.8	62.4	62.4	61.5	40.3
2	✓		✓					61.2	65.4	64.1	63.6 (+2.1)	43.2
3		✓	✓					61.1	63.5	62.5	62.3 (+0.8)	45.9
4	✓			✓				61.7	68.0	64.9	64.9 (+3.4)	43.3
5	✓			✓	✓			62.1	68.3	65.8	65.4 (+3.9)	43.3
6	✓			✓	✓	✓		62.5	68.5	66.0	65.7 (+4.2)	43.3
7	✓			✓	✓	✓	✓	**63.0**	**68.6**	**68.8**	**66.8** (+5.3)	43.5

TABLE VI

ABLATIONS ON **RASTER-PRIOR FUSION**. **FILM PLACEMENT**: APPLY FILM AT EVERY STAGE OF THE RESNET BACKBONE (**SH-FILM**) OR ONLY AT THE LAST STAGE (**LY-FILM**). **GATING**: PREDICTED EITHER FROM THE CONDITIONING VECTOR (**CONGATE**) OR FROM RASTER FEATURES (**FEAGATE**). OTHER MODULES ARE DEFINED IN SEC. III-C.

ID	Raster Encoder Modules						AP ↑			
	SH-FiLM	LY-FiLM	Congate	Feagate	Residual	SE(2)	ped.	div.	bou.	mean
1							59.8	62.4	62.4	61.5
2	✓						62.5	64.5	64.0	63.7 (+2.2)
3		✓					62.1	64.0	63.1	63.1 (+1.5)
4	✓		✓				59.9	62.8	62.9	61.8 (+0.3)
5	✓			✓			63.5	66.1	65.0	64.9 (+3.4)
6	✓				✓		65.4	67.1	66.1	66.2 (+4.7)
7	✓			✓	✓		65.5	67.1	66.5	66.4 (+4.9)
8	✓			✓	✓	✓	**65.6**	**67.4**	**66.9**	**66.7** (+5.2)

TABLE VII

ABLATIONS ON **FUSION ORDER** OF VECTOR AND RASTER PRIORS.

Fusion Order	$\mathbf{AP}_{\text{ped}}$	$\mathbf{AP}_{\text{div}}$	$\mathbf{AP}_{\text{bou}}$	**mAP**
Raster→Vector	63.8	65.1	66.7	65.2
Vector→Raster	66.6	67.2	68.2	67.4

the complete raster path delivers +5.2 mAP over the baseline.

Fusion Order. We test whether the order of fusing vector and raster priors affects the fusion performance. In **V→R**, BEV tokens first absorb vector priors via dual cross-attention and then fuse raster priors through residual projection. In **R→V**, BEV tokens are first updated by raster residual fusion before serving as queries for vector cross-attention. Tab. VII shows vector-first raster-second fusion order is better. Vector-first preserves clean BEV queries for cross-attention, whereas raster-first injects dense, potentially misaligned signals that degrade attention selectivity. The order also matches the inductive bias: vectors set the global geometric structure, and raster then provides local appearance refinements, yielding better convergence and higher AP.

Arbitrary Combinations of Map Priors. We assess source robustness and compositionality of *UMPE* by evaluating subsets of the four priors on MapTRv2. Concretely, we compare (i) single-prior baselines trained with that source only to (ii) our final unified encoder trained with all priors but toggled at test time (no retraining) (Tab. VIII). This tests that our encoder can consume **any** available subset without retraining, in contrast to prior methods that assume a fixed prior modality or availability. Surprisingly, when we toggle to a single prior at test time, *UMPE* outperforms models trained only on that single prior. We attribute this to two factors: (i) **Shared-target, multi-source co-training**. All priors are optimized together. The model learns complementary cues and avoids overfitting to any one modality's biases, so each branch is stronger even in isolation. (ii) **Robustness by design**. Zero-initialized residual fusion makes each branch modular, while SourceDropout exposes the network during training to missing prior scenarios.

TABLE VIII

EFFECT OF COMBINING FOUR MAP PRIORS ON MAPPING. GREEN ROWS ARE *single-prior* BASELINES (TRAINED WITH THAT PRIOR ONLY). ALL OTHER ROWS USE OUR **FINAL UNIFIED ENCODER** TRAINED WITH *all* PRIORS, WITH SUBSETS TOGGLED *at test time only* (NO RETRAINING).

Map Priors				AP ↑			
HD (vec)	SD (vec)	Sat.Ima.	SD (ras)	ped.	div.	bou.	mean
				59.8	62.4	62.4	61.5
✓				59.7	62.5	66.5	62.9 (+1.4)
✓				60.1	62.7	67.7	63.5 (+2.0)
	✓			61.1	63.5	62.5	62.3 (+0.8)
	✓			62.3	64.3	63.6	63.4 (+1.9)
✓	✓			63.0	68.6	68.8	66.8 (+5.3)
		✓		63.7	65.4	64.5	64.5 (+3.0)
		✓		63.4	65.4	65.9	64.9 (+3.4)
			✓	62.1	65.1	64.3	63.8 (+2.3)
			✓	63.9	65.9	65.9	65.3 (+3.8)
		✓	✓	65.6	67.4	66.9	66.7 (+5.2)
	✓	✓		66.1	67.1	67.2	66.8 (+5.3)
	✓		✓	65.4	66.8	66.4	66.2 (+4.7)
	✓	✓	✓	66.3	66.8	67.6	66.9 (+5.4)
✓	✓	✓	✓	**66.6**	**67.2**	**68.2**	**67.4** (+5.9)

V. CONCLUSION

We presented *UMPE*, a unified, alignment-aware encoder that accepts any subset of four complementary map priors and fuses them with BEV features for online mapping and end-to-end planning. Across nuScenes and Argoverse 2, *UMPE* delivers consistent mAP gains on strong backbones and, when plugged into VAD, substantially reduces trajectory $\mathbf{L}_2$ and collision rate, surpassing recent prior-injection baselines. The encoder is compositional and robust to missing sources, enabling test-time toggling without retraining. Future work will prioritize investigating the role of map priors in closed-loop, end-to-end autonomous driving.

REFERENCES

[1] K. Z. Luo, X. Weng, Y. Wang, S. Wu, J. Li, K. Q. Weinberger, Y. Wang, and M. Pavone, "Augmenting lane perception and topology understanding with standard definition navigation maps," *ICRA*, pp. 4029–4035, 2023.

[2] H. Zhang, D. Paz, Y. Guo, A. Das, X. Huang, K. Haug, H. I. Christensen, and L. Ren, "Enhancing online road network perception and reasoning with standard definition maps," in *IROS*, 2024.

[3] F. Immel, J.-H. Pauls, R. Fehler, F. Bieder, J. Merkert, and C. Stiller, "Sdtagnet: Leveraging text-annotated navigation maps for online hd map construction," *arXiv preprint arXiv:2506.08997*, 2025.

[4] W. Gao, J. Fu, Y. Shen, H. Jing, S. Chen, and N. Zheng, "Complementing onboard sensors with satellite maps: a new perspective for hd map construction," in *ICRA*, pp. 11103–11109, IEEE, 2024.

[5] J. Ye, D. Paz, H. Zhang, Y. Guo, X. Huang, H. I. Christensen, Y. Wang, and L. Ren, "Smart: Advancing scalable map priors for driving topology reasoning," *arXiv preprint arXiv:2502.04329*, 2025.

[6] M. Pei, J. Shan, P. Li, J. Shi, J. Huo, Y. Gao, and S. Shen, "Sept: Standard-definition map enhanced scene perception and topology reasoning for autonomous driving," *RAL*, 2025.

[7] Z. Dong, R. Ding, W. Li, P. Zhang, G. Tang, and J. Guo, "Leveraging sd map to augment hd map-based trajectory prediction," in *CVPR*, pp. 17219–17228, 2025.

[8] S. Zeng, X. Chang, X. Liu, Z. Pan, and X. Wei, "Driving with prior maps: Unified vector prior encoding for autonomous vehicle mapping," *arXiv preprint arXiv:2409.05352*, 2024.

[9] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Maptrv2: An end-to-end framework for online vectorized hd map construction," *IJCV*, vol. 133, no. 3, pp. 1352–1374, 2025.

[10] Z. Liu, X. Zhang, G. Liu, J. Zhao, and N. Xu, "Leveraging enhanced queries of point sets for vectorized map construction," in *European Conference on Computer Vision*, pp. 461–477, Springer, 2024.

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[12] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[13] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *CVPR*, pp. 8340–8350, 2023.

[14] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022.

[15] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Vectormapnet: End-to-end vectorized hd map learning," in *International Conference on Machine Learning*, pp. 22352–22369, PMLR, 2023.

[16] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized hd map construction," *arXiv preprint arXiv:2208.14437*, 2022.

[17] T. Yuan, Y. Liu, Y. Wang, Y. Wang, and H. Zhao, "Streammapnet: Streaming mapping network for vectorized online hd map construction," in *CVPR*, pp. 7356–7365, 2024.

[18] J. Chen, Y. Wu, J. Tan, H. Ma, and Y. Furukawa, "Maptracker: Tracking with strided memory fusion for consistent vector hd mapping," in *ECCV*, pp. 90–107, Springer, 2024.

[19] Y. Zhou, H. Zhang, J. Yu, Y. Yang, S. Jung, S.-I. Park, and B. Yoo, "Himap: Hybrid representation learning for end-to-end vectorized hd map construction," in *CVPR*, pp. 15396–15406, 2024.

[20] Z. Zhang, X. Li, S. Zou, G. Chi, S. Li, X. Qiu, G. Wang, G. Zheng, L. Wang, H. Zhao, *et al.*, "Chameleon: Fast-slow neuro-symbolic lane topology extraction," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3752–3758, IEEE, 2025.

[21] Y. Li, Z. Zhang, X. Qiu, X. Li, Z. Liu, L. Wang, R. Li, Z. Zhu, H.-a. Gao, X. Lin, *et al.*, "Reusing attention for one-stage lane topology understanding," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2025.

[22] H. Wang, T. Li, Y. Li, L. Chen, C. Sima, Z. Liu, B. Wang, P. Jia, Y. Wang, S. Jiang, *et al.*, "Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping," *NeurIPS*, pp. 18873–18884, 2023.

[23] T. Li, L. Chen, H. Wang, Y. Li, J. Yang, X. Geng, S. Jiang, Y. Wang, H. Xu, C. Xu, *et al.*, "Graph-based topology reasoning for driving scenes," *arXiv preprint arXiv:2304.05277*, 2023.

[24] D. Wu, J. Chang, F. Jia, Y. Liu, T. Wang, and J. Shen, "Topomlp: A simple yet strong pipeline for driving topology reasoning," *arXiv preprint arXiv:2310.06753*, 2023.

[25] T. Li, P. Jia, B. Wang, L. Chen, K. Jiang, J. Yan, and H. Li, "Lanesegnet: Map learning with lane segment perception for autonomous driving," *arXiv preprint arXiv:2312.16108*, 2023.

[26] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying," *TPAMI*, vol. 46, no. 5, pp. 3955–3971, 2024.

[27] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *CVPR*, pp. 15303–15312, 2021.

[28] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *CVPR*, 2022.

[29] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *CVPR*, pp. 17863–17873, 2023.

[30] X. Gu, G. Song, I. Gilitschenski, M. Pavone, and B. Ivanovic, "Accelerating online mapping and behavior prediction via direct bev feature attention," in *ECCV*, pp. 412–428, Springer, 2024.

[31] X. Gu, G. Song, I. Gilitschenski, M. Pavone, and B. Ivanovic, "Producing and leveraging online map uncertainty in trajectory prediction," in *CVPR*, pp. 14521–14530, 2024.

[32] T. Yuan, Y. Mao, J. Yang, Y. Liu, Y. Wang, and H. Zhao, "Presight: Enhancing autonomous vehicle perception with city-scale nerf priors," in *ECCV*, pp. 323–339, Springer, 2024.

[33] X. Xiong, Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Neural map prior for autonomous driving," in *CVPR*, pp. 17535–17544, 2023.

[34] Z. Jiang, Z. Zhu, P. Li, H.-a. Gao, T. Yuan, Y. Shi, H. Zhao, and H. Zhao, "P-mapnet: Far-seeing map generator enhanced by both sdmap and hdmap priors," *RAL*, 2024.

[35] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *CVPR*, pp. 17853–17862, 2023.

[36] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?," in *CVPR*, pp. 14864–14873, 2024.

[37] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, *et al.*, "Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation," *arXiv:2406.06978*, 2024.

[38] Z. Zhang, X. Qiu, B. Zhang, G. Zheng, X. Gu, G. Chi, H.-a. Gao, L. Wang, Z. Liu, X. Li, *et al.*, "Delving into mapping uncertainty for mapless trajectory prediction," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2025.

[39] Y. Gao, J. Wang, Z. Zhang, A. Jiang, Y. Wang, Y. Heng, S. Wang, H. Sun, Z. Hu, and H. Zhao, "Uniuncer: Unified dynamic static uncertainty for end to end driving," *arXiv preprint arXiv:2603.07686*, 2026.

[40] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, *et al.*, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," in *CVPR*, 2025.

[41] Z. Xing, X. Zhang, Y. Hu, B. Jiang, T. He, Q. Zhang, X. Long, and W. Yin, "Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving," in *CVPR*, pp. 1602–1611, 2025.

[42] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen, "Genad: Generative end-to-end autonomous driving," in *ECCV*, Springer, 2024.

[43] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng, "Sparsedrive: End-to-end autonomous driving via sparse scene representation," *arXiv preprint arXiv:2405.19620*, 2024.

[44] K. Zhang, Z. Tang, X. Hu, X. Pan, X. Guo, Y. Liu, J. Huang, L. Yuan, Q. Zhang, X.-X. Long, *et al.*, "Epona: Autoregressive diffusion world model for autonomous driving," *arXiv:2506.24113*, 2025.

[45] Z. Song, C. Jia, L. Liu, H. Pan, Y. Zhang, J. Wang, X. Zhang, S. Xu, L. Yang, and Y. Luo, "Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving," in *CVPR*, 2025.

[46] B. Zhang, N. Song, X. Jin, and L. Zhang, "Bridging past and future: End-to-end autonomous driving with historical prediction and planning," in *CVPR*, pp. 6854–6863, 2025.

[47] H. Fu, D. Zhang, Z. Zhao, J. Cui, D. Liang, C. Zhang, D. Zhang, H. Xie, B. Wang, and X. Bai, "Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation," *arXiv preprint arXiv:2503.19755*, 2025.

[48] S. Liu, Q. Liang, Z. Li, B. Li, and K. Huang, "Gaussianfusion: Gaussian-based multi-sensor fusion for end-to-end autonomous driving," *arXiv preprint arXiv:2506.00034*, 2025.

[49] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.

[51] R. Sun, L. Yang, D. Lingrand, and F. Precioso, "Mind the map! accounting for existing map information when estimating online hdmaps from sensor," *arXiv preprint arXiv:2311.10517*, 2023.

[52] X. Liu, S. Wang, W. Li, R. Yang, J. Chen, and J. Zhu, "Mgmap: Mask-guided learning for online vectorized hd map construction," in *CVPR*, pp. 14812–14821, 2024.