Commun Transp Res

Research Article

KEPT: Knowledge-enhanced prediction of trajectories from consecutive driving frames with vision-language models

Yujin Wang[1], Tianyi Wang[2], Quanfeng Liu[1], Wenxian Fan[1], Junfeng Jiao[3], Christian Claudel[2], Yunbing Yan[4], Bingzhao Gao[1,✉], Jianqiang Wang[5], Hong Chen[6]

Cite this article: Wang Y J, Wang T Y, Liu Q F, et al. *Commun Transp Res* 2026, **6**(1): 9640012. https://doi.org/10.26599/COMMTR.2026.9640012

ABSTRACT: Accurate short-horizon trajectory prediction is crucial for safe and reliable autonomous driving. However, existing vision language models (VLMs) often fail to accurately understand driving scenes and generate trustworthy trajectories. To address this challenge, this study introduces KEPT, a knowledge-enhanced VLM framework that predicts ego trajectories directly from consecutive front-view driving frames. KEPT integrates a temporal frequency–spatial fusion (TFSF) video encoder, which is trained via self-supervised learning with hard-negative mining, with a k-means & HNSW retrieval-augmented generation (RAG) pipeline. Retrieved prior knowledge is added into chain-of-thought (CoT) prompts with explicit planning constraints, while a triple-stage fine-tuning paradigm aligns the VLM backbone to enhance spatial perception and trajectory prediction capabilities. Evaluated on nuScenes dataset, KEPT achieves the best open-loop performance compared with baseline methods. Ablation studies on fine-tuning stages, Top-K value of RAG, different retrieval strategies, vision encoders, and VLM backbones are conducted to demonstrate the effectiveness of KEPT. These results indicate that KEPT offers a promising, data-efficient way toward trustworthy trajectory prediction in autonomous driving.

KEYWORDS: autonomous driving; trajectory prediction; vision-language model; retrieval-augmented generation; chain-of-thought prompt

1 Introduction

Accurate prediction of vehicle trajectories is a fundamental capability of autonomous driving systems and is crucial for safe and efficient navigation in dynamic traffic environments (Chai et al., 2019; Ngiam et al., 2022; Wang and Wang 2024). Recently, end-to-end autonomous driving methods have emerged as a promising paradigm by directly taking sensor data as input for perception and output planning results with one holistic model (Chen et al., 2024b; Chib and Singh 2024; Liu et al., 2021a). Through extensive data training, end-to-end approaches have demonstrated impressive planning capabilities, providing a streamlined and competitive alternative to traditional modular pipelines (Casas et al., 2021; Chen and Krähenbühl 2022; Yang et al., 2025a).

While end-to-end approaches have been widely adopted and constantly making breakthroughs on challenging benchmarks, they rely solely on fixed-format inputs, which restricts the agent's ability to comprehend multimodal information and interact with the environment and human users (Shao et al., 2023a; Wang et al., 2025b; Yu et al., 2025). Moreover, certain methods skip explicit scene understanding and directly predict driving commands from sensor data (Hu et al., 2022b; Huang et al., 2024b; Weng et al., 2024), which sacrifices interpretability and introduces challenges in optimization and safety validation.

Along this line of work, UniAD (Hu et al., 2023) introduced a query-based design that integrates perception and prediction components, enabling an end-to-end planning scheme. The vectorized scene representation for efficient autonomous driving (VAD) framework (Jiang et al., 2023a) improved computational efficiency and interpretability by adopting a vectorized scene representation, replacing the dense rasterized representations from UniAD. Despite the remarkable progress of end-to-end models in autonomous driving, such approaches inherently struggle in long-tail scenarios, where prediction errors compound, severely degrading downstream planning performance (Chitta et al., 2021; Wu et al., 2022).

Concurrently, large language models (LLMs) have demonstrated impressive capabilities in language comprehension and reasoning, showing the potential to solve this problem (Hegde et al., 2025; Wei et al., 2022). Going beyond text-based prompting, multimodal large language models (MLLMs) integrate image and video inputs to the LLM, enabling tasks such as visual question answer (VQA) and dense captioning (Li et al., 2023a; Liu et al., 2023).

[1] College of Automotive and Energy Engineering, Tongji University, Shanghai 201804, China. [2] Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin Texas 78712, USA. [3] School of Architecture, The University of Texas at Austin, Austin Texas 78712, USA. [4] School of Automotive and Traffic Engineering, Wuhan University of Science and Technology, Wuhan 430081, China. [5] School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China. [6] College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China.

✉ Corresponding author. E-mail: gaobz@tongji.edu.cn

Received: October 17, 2025; Revised: November 25, 2025; Accepted: January 12, 2026

Nomenclature

Symbol	Meaning (unit)
Sets, indices, and dimensions	
H, W, P	Image height, width, patch size (pixels)
T	Number of frames per clip (here $T = 7$)
N_p	Number of patches per frame ($N_p = \frac{H}{P} \times \frac{W}{P}$)
$R,\ C$	Real and complex domains
Images, patches, and transforms	
$I_{\mathrm{RGB}} \in R^{3 \times H \times W}$	Input RGB image (normalized intensity)
$I_g \in R^{1 \times H \times W}$	Grayscale image
$X_{\mathrm{patch}}^{(i)} \in R^{P \times P}$	i-th grayscale patch
$\mathcal{F}\{\cdot\}$	2D FFT operator
$F^{(i)} = \mathcal{F}\{X_{\mathrm{patch}}^{(i)}\} \in C^{P \times P}$	Patch spectrum
$A^{(i)} = \lvert F^{(i)} \rvert \in R^{P \times P}$	Amplitude spectrum
$\mathrm{GAP}\ (\cdot)$	Global average pooling
$a^{(i)} = \mathrm{GAP}\ (A^{(i)}) \in R$	Amplitude statistic of patch i
$w_{\mathrm{freq}} \in [0,1]^{N_p}$	Frequency-domain attention weights
$X_{\mathrm{f}}^{(i)} = w_{\mathrm{freq}}^{(i)} X_{\mathrm{patch}}^{(i)}$	Reweighted patch
$X_{\mathrm{f}} \in R^{1 \times H \times W}$	Frequency-attended image
$X_{\mathrm{spa}}^{(\mathrm{l})}$	Swin-Tiny stage-l spatial feature map
X_{s}	Upsampled/Concatenated spatial representation
$X_{\mathrm{fs}} = [X_{\mathrm{f}}; X_{\mathrm{s}}]$	Channel concatenation of X_{f} and X_{s}
$X_{\mathrm{feat}} \in R^{C' \times H \times W}$	Fused representation after 1 × 1 conv, BN, ReLU
Temporal encoding	
$h_t = \mathrm{GAP}\ (X_{\mathrm{feat},t}) \in R^{C'}$	Frame-t feature
$H_{\mathrm{seq}} = [h_1; \ldots; h_T] \in R^{T \times C'}$	Temporal feature matrix
$H_{\mathrm{in}},\ H_{\mathrm{out}}$	Transformer input/output sequences
$h_{\mathrm{TFSF}} \in R^{C'}$	CLS-token feature used as final vector
Self-supervised TFSF training (InfoNCE)	
$\mathcal{E}_\theta$	TFSF backbone with parameters θ
$\mathcal{G}_\phi$	Projection head with parameters ϕ
$z = \frac{\mathcal{G}_\phi\ (h)}{\lvert\lvert \mathcal{G}_\phi\ (h)\ \rvert\rvert_2}$	l_2-normalized projection on unit hypersphere
$\mathrm{sim}\ (u, v) = u^{\mathrm{T}}\ v$	Inner product (cosine similarity numerator)
τ_{temp}	Temperature for InfoNCE
$\mathcal{N}_i^{\mathcal{B}},\ \mathcal{Q}$	In-batch negatives; memory queue (capacity M)
$\mathrm{TopK}\ (\cdot, K)$	Hard-negative mining (keep top-K)
$\mathcal{L}_i$	InfoNCE loss of sample I; $L = (1/B) \sum_i \mathcal{L}_i$
$B,\ E,\ M,\ K$	Batch size; #epochs; queue capacity; #hard negatives
Embedding & retrieval	
$e_i \in R^{128}$	Normalized embedding of clip i
$n_c,\ c_j$	#Clusters; k-means centroid of cluster j
$d_{\cos}\ (\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\boldsymbol{x}^{\mathrm{T}}\ \boldsymbol{y}}{\lvert\lvert \boldsymbol{x} \rvert\rvert_2 \lvert\lvert \boldsymbol{y} \rvert\rvert_2}$	Cosine distance
$\mathrm{HNSW}\ (M,\ \mathrm{ef}_{\mathrm{construction}})$	HNSW index (connectivity M, build ef)
k	#Nearest neighbors for retrieval
Triple-stage fine-tuning (Section 3.4)	
Stage A: Spatial perception (VQA multitask)	
$\mathcal{L}_{\mathrm{A}}$	Stage-A multitask loss (classification + size + distance)
C	Number of semantic categories for classification
$\eta_{k,n},\ \pi_{k,n}$	One-hot GT label and predicted probability for class k in sample n
$s_{d,n},\ \widehat{s}_{d,n}$	predicted/Reference object size along dimension $d \in \{1,\ 2,\ 3\}$
$d_n,\ \widehat{d}_n$	Predicted/reference distance to ego vehicle
$\alpha_n,\ \beta_n,\ \gamma_n$	Binary indicators activating cls./size/dist. terms for sample n
Stage B: Surround-view trajectory regression	
$\mathcal{L}_{\mathrm{B}}$	Stage-B trajectory regression objective (coords + velocity; with smoothness/curvature regularization)
Q	Number of prediction horizons (here $Q = 3$, at 1 s or 2 s or 3 s)
$q,\ r$	Horizon index $q \in \{1, \ldots, Q\}$; sample index $r \in \{1, \ldots, B\}$
$x_{q,r},\ y_{q,r},\ v_{q,r}$	predicted ego-centric displacements and velocity at horizon q for sample r
$\widetilde{x}_{q,r},\ \widetilde{y}_{q,r},\ \widetilde{v}_{q,r}$	Ground-truth counterparts of $x_{q,r},\ y_{q,r},\ v_{q,r}$
Stage C: Front-view end-to-end trajectory prediction	
$\mathcal{L}_{\mathrm{C}}$	Stage-C loss
Kinematics and evaluation	
(x, y)	Ego-centric coordinates (m)
v	Velocity (km/h)
$\widehat{\tau} = \{(\widehat{x}_i, \widehat{y}_i)\}_{i=1}^{6},\ \tau = \{(x_i, y_i)\}_{i=1}^{6}$	Predicted/ground-truth trajectory at 2 Hz over 3 s (m)
$l_i = \sqrt{(x_i - \widehat{x}_i)^2 + (y_i - \widehat{y}_i)^2}$	Per-step euclidean error (m)

Nevertheless, existing MLLM datasets for autonomous driving remain limited in scale, typically containing fewer than one million VQAs (Wu et al., 2024). Consequently, without careful model and training scheme designs, these MLLMs exhibit poor performance in reasoning and planning tasks due to a lack of scene understanding and grounding capability (Ma et al., 2024b).

In this work, we propose knowledge-enhanced prediction of trajectories (KEPT), a knowledge-enhanced trajectory prediction framework, introducing a novel approach to assist an MLLM in predicting future trajectories according to consecutive driving frames, as depicted in Fig. 1.

The main research contributions of this work are outlined as follows:

- Temporal frequency–spatial fusion (TFSF) encoder for consecutive driving frames: We build a video encoder that mixes frequency cues from a fast Fourier transform (FFT)-based attention module with the multiscale spatial features of a Swin-Tiny backbone. A short temporal transformer then processes seven frames sampled at 2 Hz. Together, these components produce clip-level embeddings that capture the motion patterns most relevant for planning.
- Self-supervised representation learning with hard-negative mining and memory queue: We train the TFSF module with a

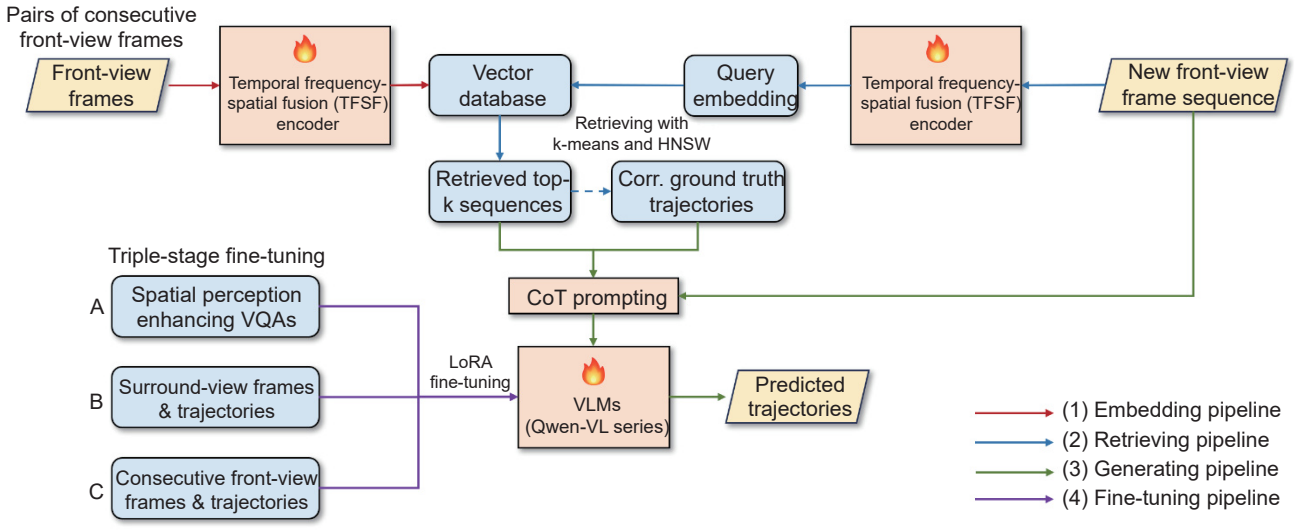

Fig. 1 Overview of our KEPT method. The method consists of four pipelines: (1) The embedding pipeline encodes front-view frame sequences into a vector database via a self-designed and trained temporal frequency–spatial fusion (TFSF) encoder. (2) The retrieving pipeline retrieves the most similar scene from the database via k-means clustering and hierarchical navigable small world graphs (HNSW) searching. (3) The generating pipeline guides vision language models (VLMs) in predicting trajectories in the future 3 s according to the new frame sequence and retrieved information via a chain-of-thought (CoT) prompting strategy. (4) The fine-tuning pipeline involves a triple-stage fine-tuning paradigm, which aims to equip VLMs with spatial grounding, motion feasibility, and temporally conditioned planning capabilities, enabling trajectory prediction from raw perceptual inputs. The fire icon denotes components that are trained or fine-tuned in this work; others are frozen or nonparametric. In our pipeline, the TFSF encoder is trained with self-supervision, the retrieval stack is nonparametric, and the VLM's language head is fine-tuned with LoRA across three stages.

contrastive InfoNCE loss. The batch provides most of the negatives, and we keep an additional fixed-size queue to supply more varied ones. We also use a simple Top-K strategy to choose the harder negatives. The encoder outputs unit-norm embeddings, which are then used directly for retrieval.

- Scalable embedding-retrieval pipeline for trajectory priors: We build a vector database of scene embeddings and run a two-step approximate nearest-neighbor search. The first step uses k-means clustering to narrow down the candidates, and the second applies HNSW with cosine distance to find the closest matches. From these results, we retrieve the Top-K similar scenes together with their ground-truth trajectories, which serve as useful priors for the planner.
- Triple-stage fine-tuning of VLMs bridging perception to planning: (1) We improve spatial perception by training with VQAs, covering attributes such as object class, size, and distance, and we apply LoRA to keep the adaptation lightweight. (2) Then, we train it to predict trajectories using the six surround-view images and a few basic kinematics. The loss mainly keeps the motion smooth and prevents unsafe bends or sudden shifts. (3) In the last stage, the model learns to predict the whole trajectory from consecutive front-view frames. We keep the same regression loss, but here, the goal is to help the language head pick up a short-term temporal structure.
- Chain-of-Thought (CoT) promotion with exemplar comparison and planning constraints: We provide the VLMs with several retrieved reference scenes before the target scene. The models are then guided to compare these scenes, subject to safety constraints. Finally, a trajectory in a specific format is required for the models to generate.

The rest of this paper is organized as follows: Section 2 reviews related works in this field thoroughly. In Section 3, details of the KEPT are introduced. In Section 4, comparative experiments, ablation studies, and case studies are designed to demonstrate the effectiveness of the KEPT. Section 5 summarizes the work and discusses future research directions.

2 Related work

2.1 Foundation models in autonomous driving

Recent advancements in foundation models have demonstrated their ability to reason over complex contextual information, interpret human intent, and generate logically consistent outputs in autonomous driving (Ma et al., 2024c; Mao et al., 2023; Wei et al., 2022; Zhu et al., 2025). These works investigated the capabilities of LLMs to generalize to uncommon scenarios as well as their ability to reason about driving scenes in text. "Drive as you speak" (Cui et al., 2024) enriched LLMs with comprehensive environmental data from different vehicle modules, supporting safer and more context-aware driving decisions. "Driving with LLMs" (Chen et al., 2024a) introduced LLMs that generated 10,000 driving scenarios for agent training. "Drive like a human" (Fu et al., 2024) demonstrated LLMs' capabilities of interacting with environments in closed-loop systems and effectively navigating long-tail autonomous driving scenarios. Dilu (Wen et al., 2024) first leveraged knowledge-driven capability in decision-making for autonomous vehicles. Cascading cooperative multi-agent in autonomous driving merging (CCMA)(Zhang et al., 2025) integrated reinforcement learning for individual optimization and LLMs for regional and global coordination, significantly improving traffic flow and safety in autonomous driving scenarios.

Meanwhile, recent works have investigated the incorporation of MLLMs into end-to-end frameworks (Brandstätter et al., 2025; Liao et al., 2025; Wang et al., 2024; Xing et al., 2025). Autonomous driving systems integrating multisource information, such as vision, language, and rules, tend to exhibit greater generalization in complex or long-tail scenarios (Chen et al., 2025; Ma et al., 2025; Wang et al., 2025c). VLP (Pan et al., 2024) incorporated linguistic descriptions into the training process and aligned them with visual features, which significantly improved cross-domain generalization. DriveGPT4 (Xu et al., 2024) utilized VLMs to predict control commands and simultaneously generate textual justifications using an iterative VQA format. To enhance

embodied intelligence, LMDrive (Shao et al., 2024) integrated a vision encoder with LLMs, enabling multimodal scene understanding and natural language command execution. Omnidrive (Wang et al., 2025a) introduced a 3-dimensional (3D) VLM architecture to strengthen planning and reasoning capabilities. DriveVLM (Tian et al., 2024) incorporated traditional 3D perception into a multistage reasoning chain, combining scene description, dynamic analysis, and hierarchical planning to bridge cognitive depth with real-time control. DiMA (Hegde et al., 2025) employed a knowledge distillation approach to transfer planning capabilities from MLLMs to lightweight vision-based planners via surrogate tasks.

2.2 End-to-end trajectory prediction

Early end-to-end trajectory prediction models primarily used LiDAR point clouds (Casas et al., 2020; Liang et al., 2020; Wu et al., 2020). However, due to the dependence on accurate bounding box detection, these approaches often fail to generalize in the presence of occluded or unclassified objects. To solve this problem, recent studies have shifted toward vision-centric methods based on bird's-eye view (BEV) representations, which provide a unified spatial-temporal understanding of the driving scene (Akan and Güney 2022; Hu et al., 2021a; Liang et al., 2022). For instance, PowerBEV (Li et al., 2023b) realized semantic and instance-level trajectory prediction by simply forecasting segmentation and centripetal backward flow. TPV (Huang et al., 2023) extended BEV with two additional perpendicular planes and utilized an attention mechanism to aggregate the image features corresponding to each query in each TPV plane. LiDAR camera fusion approaches have also emerged. For example, detecting as labeling (DAL) (Huang et al., 2024a) contained an attention-free prediction pipeline and an easy training process to achieve a trade-off between the speed and accuracy of 3D object detection and prediction through efficient multimodal fusion. BEVFormer (Li et al., 2025) leveraged a unified multimodality spatial attention mechanism for integrating camera and LiDAR data with historical BEV features. Despite these advances, Bird's-eye view (BEV)-basedmethods still face limitations in capturing fine-grained 3D structures and object-level semantics.

To address these shortcomings, vision-centric Four-Dimensional (4D) perception emerges as a promising alternative to effectively extend temporal occupancy prediction with camera images as input but also broaden both semantic and instance prediction beyond the BEV format and predefined categories (Tian et al., 2023; Tong et al., 2023; Wang et al., 2023; Wei et al., 2023). Cam4Docc (Ma et al., 2024a) was the first to achieve a vision-centric 4D occupancy forecasting task, allowing simultaneous prediction of future trajectories for both general movable and static objects. Building upon this, Drive-OccWorld (Yang et al., 2025b) investigated potential applications of the camera-based 4D occupancy forecasting world model by injecting action conditions and integrating this generative capability with end-to-end planners for safe driving.

2.3 Retrieval-augmented generation in vision-language models

In vision-language tasks, retrieval-augmented generation (RAG) mitigates knowledge limitations by leveraging external knowledge bases, enabling models to extract insights from images while supplementing them with retrieved contextual data (Shao et al., 2023b; Wang et al., 2025c). Prior studies (Jiang et al., 2023b; Ram et al., 2023) on VLMs have shown that retrieval can improve contextual integration and multistep reasoning under knowledge deficits, strengthen performance on complex question answering when paired with strong pretrained encoders, and yield better downstream results when large external corpora are incorporated during pretraining and fine-tuning. Follow-up works (Hussien et al., 2025; Zheng et al., 2024) further highlighted RAG's advantages for adaptive, multimodal generation in data-scarce domains and its ability to tighten cross-modal associations for more faithful grounding.

For autonomous driving applications, dynamic retrieval policies that adjust what to fetch at run time have been proposed to match task requirements and latency budgets (Bandyopadhyay et al., 2025). Domain knowledge can also be queried explicitly: a traffic-regulation agent retrieves applicable rules and guideline hints conditioned on ego state and scene context to inform compliance-aware planning (Gao et al., 2025). RAG offers a principled way to compensate for the limited world knowledge and long-tail corner cases that arise in autonomous driving (Atakishiyev et al., 2024). For instance, RAG-Driver (Yuan et al., 2024) introduced a retrieval-augmented in-context learning mechanism through a curated multimodal driving in-context instruction tuning dataset and a vector similarity-based retrieval engine specifically tailored for driving applications. VistaRAG (Dai et al., 2024) positioned the RAG as a safe and trustworthy layer that dynamically retrieves prior driving experience, live road-network state, and other contextual evidence to ground decisions and make reasoning auditable. SenseRAG (Luo et al., 2025) constructed an LLM-readable environmental knowledge base from multimodal sensor/V2X streams and issued proactive queries to retrieve time-critical facts, yielding measurable gains in perception and prediction under latency constraints.

In summary, recent advances show that VLMs and RAG substantially enhance grounding, contextual reasoning, and data efficiency for autonomous driving; however, most pipelines (1) retrieve generic knowledge rather than planning-specific cues, (2) model temporal dynamics weakly for long-horizon forecasts, and (3) offer limited safety calibration and auditability when reasoning under distribution shifts. In addition, the training signal often fails to reflect metric accuracy or basic physical constraints, which makes the model behave unreliably in crowded scenes where rules and interactions matter.

To address these limitations, we introduce KEPT, a retrieval-augmented VLM planner that couples a temporal frequency–spatial fusion encoder with scene-aligned retrieval, CoT prompting and a triple-stage fine-tuning strategy that grounds reasoning in metric trajectories and collision risks. KEPT is designed to turn externally retrieved knowledge into actionable, long-horizon plans, improving both accuracy and safety.

3 Methodology

3.1 Temporal frequency–spatial fusion encoder for consecutive driving frames

In this subsection, we introduce a novel TFSF encoder for consecutive driving frames. The TFSF encoder integrates frequency-domain and spatial-domain representations at multiple granularities and employs a temporal transformer for encoding temporal dependencies. An overview of the architecture is illustrated in Fig. 2.

For an input RGB image $I_{\mathrm{RGB}} \in R^{3 \times H \times W}$, we first convert it into a grayscale representation $I_{\mathrm{g}} \in R^{1 \times H \times W}$ by standard luminance-based weighting:

$$I_{\mathrm{g}} = 0.299 I_{\mathrm{R}} + 0.587 I_{\mathrm{G}} + 0.114 I_{\mathrm{B}} \tag{1}$$

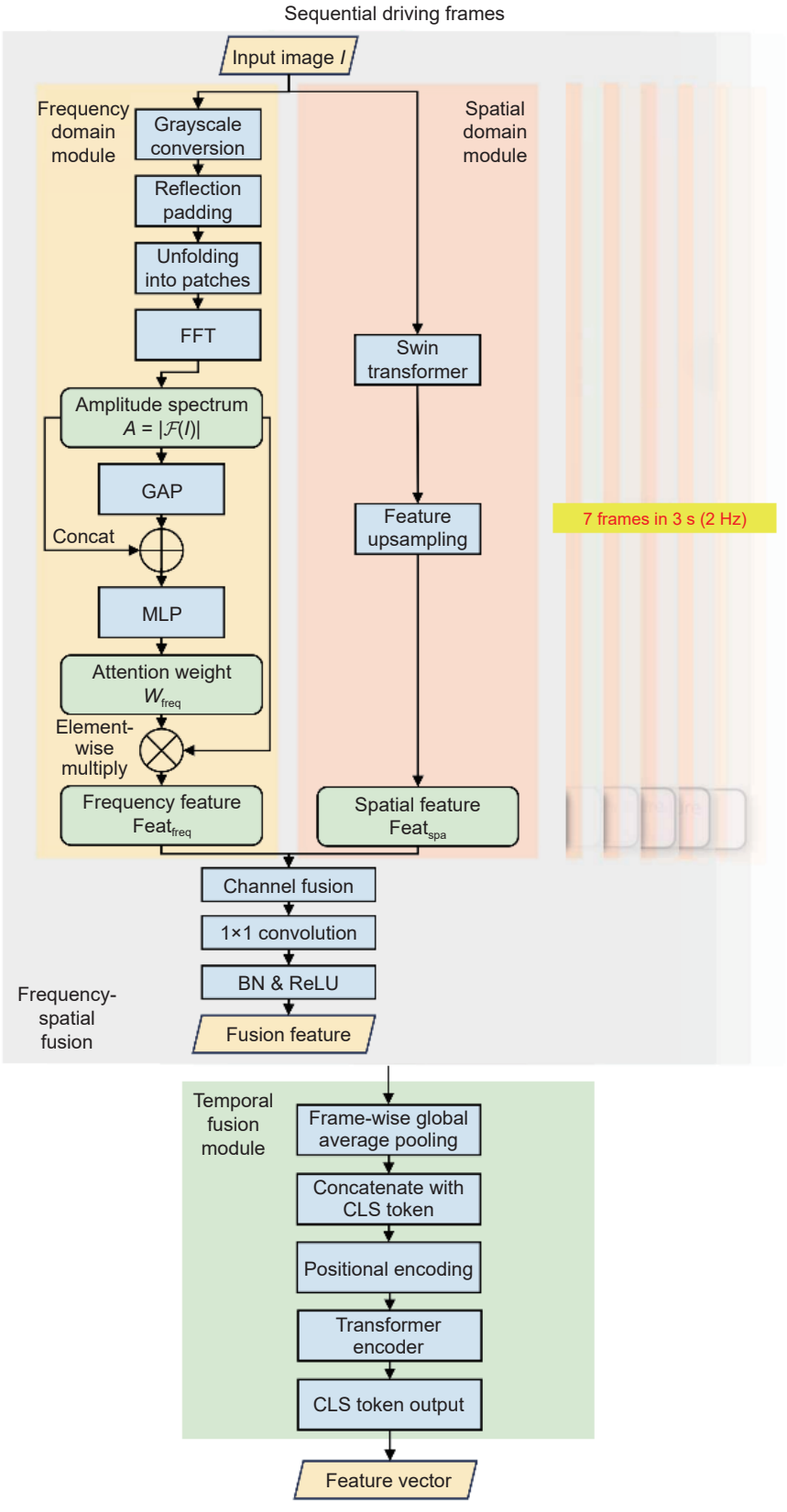

Fig. 2 Architecture of our proposed TFSF encoder for consecutive driving frames.

We partition the grayscale image into nonoverlapping patches of size $P \times P$, generating a tensor $X_{\text{patch}} \in R^{N_p \times P \times P}$, where $N_p = \frac{H}{P} \times \frac{W}{P}$. Each patch $X_{\text{patch}}^{(i)}$ is independently transformed into the frequency-domain via a 2-dimensional (2D) FFT:

$$F^{(i)} = \mathcal{F}\{X_{\text{patch}}^{(i)}\}, F^{(i)} \in C^{P \times P} \tag{2}$$

We subsequently compute the amplitude spectrum $A^{(i)}$:

$$A^{(i)} = \left|F^{(i)}\right|, A^{(i)} \in R^{P \times P} \tag{3}$$

Then, global average pooling (GAP) is applied to $A^{(i)}$:

$$a^{(i)} = \mathrm{GAP}\left(A^{(i)}\right) = \frac{1}{P^2}\sum_{u=1}^{P}\sum_{v=1}^{P} A_{u,v}^{(i)}, a^{(i)} \in R \tag{4}$$

These amplitude statistics across all patches form a vector $a \in R^{N_p}$. A multilayer perceptron (MLP), parameterized by weights W_f and biases b_f, predicts frequency-domain attention weights $w_\mathrm{f} \in R^{N_p}$:

$$w_f = \sigma\left(\mathrm{MLP}\left(a; W_\mathrm{f}, b_\mathrm{f}\right)\right), w_\mathrm{f} \in [0,1]^{N_p} \tag{5}$$

where σ denotes the sigmoid activation function. The patches are then reweighted in the spatial domain by multiplying each original grayscale patch $X_{\mathrm{patch}}^{(i)}$ with the corresponding scalar attention weight $w_\mathrm{f}^{(i)}$:

$$X_\mathrm{f}^{(i)} = w_\mathrm{f}^{(i)} \cdot X_{\mathrm{patch}}^{(i)}, X_\mathrm{f}^{(i)} \in R^{P \times P} \tag{6}$$

Finally, we reconstruct the frequency-attended image $X_\mathrm{f} \in R^{1 \times H \times W}$ by folding patches back to their original spatial positions.

To effectively capture hierarchical spatial representations, we employ a pretrained hierarchical Swin Transformer (Liu et al., 2021b) (Swin-Tiny) backbone. Given the original RGB image I_{RGB}, we feed it into the Swin Transformer:

$$X_{\mathrm{spa}}^{(\mathrm{l})} = \mathrm{SwinTransformer}_l\left(I_{\mathrm{RGB}}\right), l = 1,2,3,4 \tag{7}$$

where each stage l yields feature maps at progressively coarser scales, thus providing multiscale spatial representations. To align these multiscale features with the original spatial resolution, we apply bilinear interpolation-based upsampling, resulting in a refined spatial representation $S \in R^{C \times H \times W}$:

$$X_\mathrm{s} = \mathrm{Upsample}\left(\mathrm{Concat}\left(X_{\mathrm{spa}}^{(1)}, X_{\mathrm{spa}}^{(2)}, X_{\mathrm{spa}}^{(3)}, X_{\mathrm{spa}}^{(4)}\right)\right) \tag{8}$$

To integrate complementary information from the frequency and spatial domains, we concatenate the frequency-attended image X_f with the spatial features X_s along the channel dimension, yielding the fused tensor $X_\mathrm{fs} \in R^{(C+1) \times H \times W}$:

$$X_\mathrm{fs} = [X_\mathrm{f}; X_\mathrm{s}] \tag{9}$$

Subsequently, a 1×1 convolution followed by batch normalization (BN) and ReLU activation generates the unified spatial-frequency representation X_feat:

$$X_\mathrm{feat} = \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}_{1\times1}\left(X_\mathrm{fs}\right)\right)\right), X_\mathrm{feat} \in R^{C' \times H \times W} \tag{10}$$

Given a sequence of T consecutive image frames, we encode temporal dependencies among frame-level representations. First, we perform GAP on each spatial-frequency feature map $X_{\mathrm{feat},t}$ (frame t), obtaining temporal feature vectors $h_t \in R^{C'}$:

$$h_t = \mathrm{GAP}\left(X_{\mathrm{feat},t}\right), t = 1,2,\ldots,T \tag{11}$$

The resulting feature vectors across all frames form a temporal feature matrix $H_\mathrm{seq} \in R^{T \times C'}$:

$$H_\mathrm{seq} = [h_1; h_2; \ldots; h_T] \tag{12}$$

We prepend a learnable classification token $H_\mathrm{cls} \in R^{1 \times C'}$ and add positional embeddings $P_\mathrm{pos} \in R^{(T+1) \times C'}$:

$$H_\mathrm{in} = [h_\mathrm{cls}; h_1; h_2; \ldots; h_T] + P_\mathrm{pos} \tag{13}$$

This sequence is then processed through an L-layer transformer encoder (multihead self-attention (MHSA) and MLP blocks):

$$H_\mathrm{out} = \mathrm{TransformerEncoder}^{(L)}\left(H_\mathrm{in}\right), H_\mathrm{out} \in R^{(T+1) \times C'} \tag{14}$$

We take the output corresponding to the classification token as the final feature representation f_TFSF:

$$h_\mathrm{TFSF} = H_\mathrm{out}^{(0)}, h_\mathrm{TFSF} \in R^{C'} \tag{15}$$

For both training and evaluation in this work, we form each input clip by a sliding 3-second window sampled at 2 Hz (T = 7 frames). This makes the sequential input strictly causal and temporally aligned with the reported horizons. Short-horizon planning depends on high-frequency changes that signal relative motion and multiscale spatial structure that stabilizes geometry over a brief, causal clip. The TFSF encoder explicitly combines FFT-based frequency attention with multiscale Swin features and a temporal transformer over the 2 Hz window, so we expect these improvements to grow with horizon length (2–3 s) where single-frame cues are relatively weaker.

3.2 Self-supervised training of the TFSF encoder

The TFSF encoder is trained through a label-free contrastive learning paradigm that combines in-batch negatives, a fixed-capacity memory queue and a targeted hard-negative mining strategy, as shown in Algorithm 1.

Let $\mathcal{V} = \{V_i\}_{i=1}^{N}$ denote an unlabeled video corpus where each clip $V_i = \{I_{i,t}\}_{t=1}^{T}$ comprises T RGB frames $I_{i,t} \in R^{3 \times H \times W}$. Two independently drawn stochastic augmentations $\mathcal{A}_1$ and $\mathcal{A}_2$ (random-resized crop, horizontal flip, color jitter, Gaussian blur) yield a positive pair:

$$\widetilde{\mathcal{V}}_i^1 = \mathcal{A}_1\left(V_i\right), \widetilde{\mathcal{V}}_i^2 = \mathcal{A}_2\left(V_i\right) \tag{16}$$

The TFSF encoder $f_\theta: R^{T \times 3 \times H \times W} \to R^d$ maps a clip to a d-dimensional representation $h = f_\theta\left(\widetilde{\mathcal{V}}\right)$. A three-layer projection head $\mathcal{G}_\phi: R^d \to R^m$ projects h onto the unit hypersphere:

$$z = \frac{\mathcal{G}_\phi\left(h\right)}{||\mathcal{G}_\phi\left(h\right)||_2}, z \in R^m, d = 64,\ m = 128 \tag{17}$$

For mini-batch index i, we denote the anchor as $z_i^{(1)}$ and its augmented counterpart as $\boldsymbol{z}_i^{(2)}$. Their similarity (positive logits) is calculated as

$$s_i^+ = \frac{\boldsymbol{z}_i^{(1)\mathrm{T}}\boldsymbol{z}_i^{(2)}}{\tau_\mathrm{temp}}, \tau_\mathrm{temp} > 0 \tag{18}$$

where two negative pools are employed, namely, the in-batch negatives $\mathcal{N}_i^{\mathcal{B}} = \left\{\boldsymbol{z}_j^{(k)} | j \neq i \text{ or } k = 2\right\}$ and the memory-queue negatives $\mathcal{Q} = \{q_l\}_{l=1}^{M}$ with fixed size $M = 1024$, storing projections from recent mini-batches.

Since most negatives are easily separable, we retain only the K most similar hard negatives:

$$\mathcal{N}_i^* = \mathrm{TopK}\left(\mathcal{N}_i^{\mathrm{B}} \cup \mathcal{Q}, \boldsymbol{z}_i^{(1)\mathrm{T}}\boldsymbol{z}\right), K = 10 \tag{19}$$

The InfoNCE loss can be computed using Eq. (20):

Algorithm 1 Self-supervised training of the TFSF encoder

Inputs:

$\mathcal{V}$ – unlabeled video corpus

$\mathcal{E}_\theta, \mathcal{G}_\phi$ – TFSF backbone and projection head

E, B – epochs, mini-batch size

τ_{temp} – temperature

K_h – hard negatives

M_{Q} – queue capacity

Outputs:

θ^*, ϕ^* – trained parameters

procedure $\{TRAIN\}\{\mathcal{V}, \mathcal{E}_\theta, \mathcal{G}_\phi, E, B, \tau_{\text{temp}}, K_h, M_{\text{Q}}\}$
  initialize θ, ϕ
  $Q \leftarrow \varnothing$ ▷FIFO memory queue
  for epoch $\leftarrow 1$ **to** E **do**
    for mini-batch$B \leftarrow \text{SAMPLE}(\mathcal{V}, B)$ **do**
      for each clip$V_i \in B$ **do**
        $V_i^1 \leftarrow \mathcal{A}_1(V_i)$; $V_i^2 \leftarrow \mathcal{A}_2(V_i)$
        $z_i^1 \leftarrow \text{l}_2\text{\_\_norm}\left(\mathcal{G}_\phi\left(\mathcal{E}_\theta\left(V_i^1\right)\right)\right)$
        $z_i^2 \leftarrow \text{l}_2\text{\_\_norm}\left(\mathcal{G}_\phi\left(\mathcal{E}_\theta\left(V_i^2\right)\right)\right)$
      end for
      $N \leftarrow \{z_j^k \mid j \neq i \text{ or } k = 2\}_{\forall i} \cup Q$
      for $i \leftarrow 1$ **to** B **do**
        $H_i \leftarrow \text{TOP-K}_{\text{roll}}\left(\text{sim}\left(z_i^1, \cdot\right), N, K_h\right)$ ▷K most similar
        $s_i^+ \leftarrow \text{sim}\left(z_i^1, z_i^2\right)/\tau_{\text{temp}}$
        $\Delta_i \leftarrow \sum_{z \in H_i} \exp\left(\text{sim}\left(z_i^1, z\right)/\tau_{\text{temp}}\right)$
        $\mathcal{L}_i \leftarrow -\log\left(\dfrac{\exp\left(s_i^+\right)}{\exp\left(s_i^+\right) + \Delta_i}\right)$
      end for
      $\mathcal{L} \leftarrow (1/B)\sum_i \mathcal{L}_i$
      $(\theta, \phi) \leftarrow \text{ADAMW}\left((\theta, \phi), \nabla\mathcal{L}\right)$
      $Q \leftarrow \text{ENQUEUE}\left(Q, \{z_i^2\}\right)$; $\text{TRIM}(Q, M_Q)$
    end for
  end for
  return θ^*, ϕ^*
end procedure

Notation: $\text{sim}(u, v) = u^{\text{T}} v$; $\text{l}_2\text{\_\_norm}(x) = x/|x|_2$; $\text{TOP-K}_{\text{roll}}$ returns the K largest inner-products.

$$\mathcal{L}_i = -\log \frac{\exp\left(s_i^+\right)}{\exp\left(s_i^+\right) + \sum_{\boldsymbol{z} \in \mathcal{N}_i^*} \exp\left(\dfrac{\boldsymbol{z}_i^{(1)\text{T}} \boldsymbol{z}}{\tau_{\text{temp}}}\right)} \tag{20}$$

and the batch loss can be defined as $\mathcal{L} = B^{-1} \sum_{i=1}^{B} \mathcal{L}_i$.

After each iteration, the vectors $\left\{\boldsymbol{z}_i^{(2)}\right\}_{i=1}^{B}$ are en-queued, and the oldest B items are dequeued, keeping $|\mathcal{Q}| = M$. Both the backbone parameters θ and the projection parameters ϕ receive direct gradient updates:

$$\theta \leftarrow \theta - \eta_e \nabla_\theta \mathcal{L}, \phi \leftarrow \phi - \eta_p \nabla_\phi \mathcal{L} \tag{21}$$

with learning rates $\eta_e = 1 \times 10^{-5}$, $\eta_p = 1 \times 10^{-4}$ and weight decay 1×10^{-4}. Training is conducted for $E = 50$ epochs on 7-frame clips, batch size $B = 8$ and temperature $\tau_{\text{temp}} = 0.07$ on a single 3090 GPU. The model with minimum validation loss is preserved for downstream embedding and retrieval tasks.

3.3 Embedding and retrieval pipeline

The overview of the embedding and retrieval pipeline is shown in Algorithm 2. Given a collection of unlabeled video sequences, a database embedding set could be constructed first using the trained TFSF encoder coupled with a projection head. Specifically, for each video clip $V_i = \{I_{i,t}\}_{t=1}^{T}$, containing $T = 7$ consecutive frames, we perform deterministic image preprocessing, including resizing to 224 × 224, normalization by ImageNet mean and standard deviation, and tensor stacking. These preprocessed frames form an input tensor $X_i \in R^{1 \times 7 \times 3 \times 224 \times 224}$.

Algorithm 2 Embedding and retrieval pipeline

Inputs:

$V_{\text{db}}, V_{\text{val}}$ – Database and validation video clips

$\mathcal{E}_\theta, \mathcal{G}_\phi$ – Trained TFSF encoder and projection head

nc, M – Number of clusters, HNSW connectivity

θ^*, ϕ^* – trained parameters

procedure*EmbeddingandRetrieval*
  1. Embedding Generation
  for each clip $V_i \in V_{\text{db}}$ **do**
    $X_i \leftarrow \text{Preprocess}(V_i)$
    $h_i \leftarrow \mathcal{E}_\theta(X_i)$
    $e_i \leftarrow \text{Normalize}\left(\mathcal{G}_\phi(h_i)\right)$
  end for
  Save embeddings$\{e_i\} \rightarrow$ database
  2. Clustered HNSW Indexing
  $\{c_j\}$, labels $\leftarrow$ k-means$(\{e_i\}$, n_clusters $=$ nc$)$
  for each cluster j **do**
    Build HNSW$(\{e_i | \text{labels}[i] = j\}, \text{connectivity} = M)$
  end for
  3. Retrieval
  for each evaluation clip $V_k \in V_{\text{val}}$ **do**
    $X_k' \leftarrow \text{Preprocess}(V_k)$
    $h_k' \leftarrow \mathcal{E}_\Theta\left(X_k'\right)$, $e_k' \leftarrow \text{Normalize}\left(\mathcal{G}_\phi\left(h_k'\right)\right)$
    $c_j \leftarrow \text{k-means.predict}\left(e_k'\right)$
    retrieved_indices $\leftarrow$ HNSW_query$\left(e_k'\right.$, cluster $= c_j$, topk $= k)$
    retrieved_trajectories $\leftarrow$ Map indices $\rightarrow$ trajectories
    Output retrieved_trajectories
  end for
end procedure

Each video sequence embedding is computed as

$$h_i = \mathcal{E}_\theta(X_i), e_i = \frac{\mathcal{G}_\phi(h_i)}{||\mathcal{G}_\phi(h_i)||_2} \in R^{128} \tag{22}$$

where $\mathcal{E}_\theta$ denotes the trained TFSF encoder and $\mathcal{G}_\phi$ represents a multilayer projection head consisting of linear layers, batch normalization, and ReLU activations. This produces an embedding $\mathbf{e}_i$ that resides on the 128-dimensional unit hypersphere. Embeddings for all sequences are subsequently stored in a database along with unique identifiers for retrieval tasks.

To facilitate efficient retrieval, we adopt a two-stage indexing strategy combining unsupervised clustering with hierarchical navigable small-world graphs (HNSW) (Malkov and Yashunin 2020).

Given the database embeddings $\mathcal{E}_{\text{db}} = \{e_i\}_{i=1}^{N_{\text{db}}}$, we first partition them into n_c clusters via k-means clustering:

$$\arg\min_{\{\boldsymbol{c}_j\}} \sum_{i=1}^{N_{\text{db}}} \min_j ||\boldsymbol{e}_i - \boldsymbol{c}_j||_2^2 \tag{23}$$

where $\{\boldsymbol{c}_j\}_{j=1}^{n_c}$ denote cluster centroids. Each embedding is thus associated with a cluster label.

Within each cluster, we construct a separate HNSW index designed for efficient approximate nearest-neighbor search under cosine distance:

$$d_{\cos}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}}{||\boldsymbol{x}||_2 ||\boldsymbol{y}||_2} \tag{24}$$

Specifically, for cluster j, an HNSW index is initialized and built using all embeddings within that cluster. Parameters including $M = 16$ and ef_construction $= 200$ are set to balance retrieval speed and accuracy.

Given validation embeddings $\mathcal{E}_{\text{val}} = \left\{e_k'\right\}_{k=1}^{N_{\text{val}}}$, retrieval is conducted by predicting cluster assignments via the prefitted k-means model and querying the corresponding cluster-specific HNSW index to identify the k nearest database embeddings:

$$\{e_l\}_{l=1}^{k} = \text{HNSW\_knn\_query}(e_l') \tag{25}$$

Each retrieved embedding index corresponds to an entry in the original database JSON file, from which associated ground truth trajectories are extracted. Formally, given the validation embedding index k and retrieved indices $\{l_1, \ldots, l_k\}$, the output structure is

$$R_k = \{\mathcal{T}(l_j)\}_{j=1}^{k} \tag{26}$$

where $\mathcal{T}(l_j)$ denotes the trajectory annotation corresponding to database embedding index l_j. This is a pipeline designed for large-scale validation datasets. In practical applications, a new scene sequence can be independently embedded and used for one-to-one retrieval, retaining a single trajectory.

Representative retrieval examples are illustrated in Fig. 3. The top row displays a sequence of query driving frames sampled at uniform intervals over the past 3 s, capturing a scenario involving a truck ahead. The subsequent three rows illustrate the top-3 retrieval results obtained by the proposed embedding and retrieval method. Each retrieval result consists of a visually similar sequence of frames retrieved from the training database using the learned embeddings. The ground truth trajectories corresponding to each retrieved sequence are provided, including position coordinates (x, y) and velocities (in km/h) for future timestamps at intervals of 0.5 s, spanning the next 3 s. These retrieved cases demonstrate the ability of the embedding pipeline to effectively capture visual similarity and retrieve semantically relevant driving scenarios, thereby providing useful trajectory priors for downstream prediction tasks.

3.4 Triple-stage fine-tuning strategy

Reliable trajectory prediction in autonomous driving relies on VLMs' ability to accurately perceive spatial relationships in complex traffic scenes. Although VLMs have already demonstrated strong reasoning capabilities, their depth and scale estimation from monocular inputs often remains suboptimal, particularly for distant targets. Such deficiencies can cascade through the prediction pipeline, leading to unsafe trajectories. Therefore, we adopt a triple-stage fine-tuning strategy for VLMs, in which the first stage is designed to enhance spatial perception

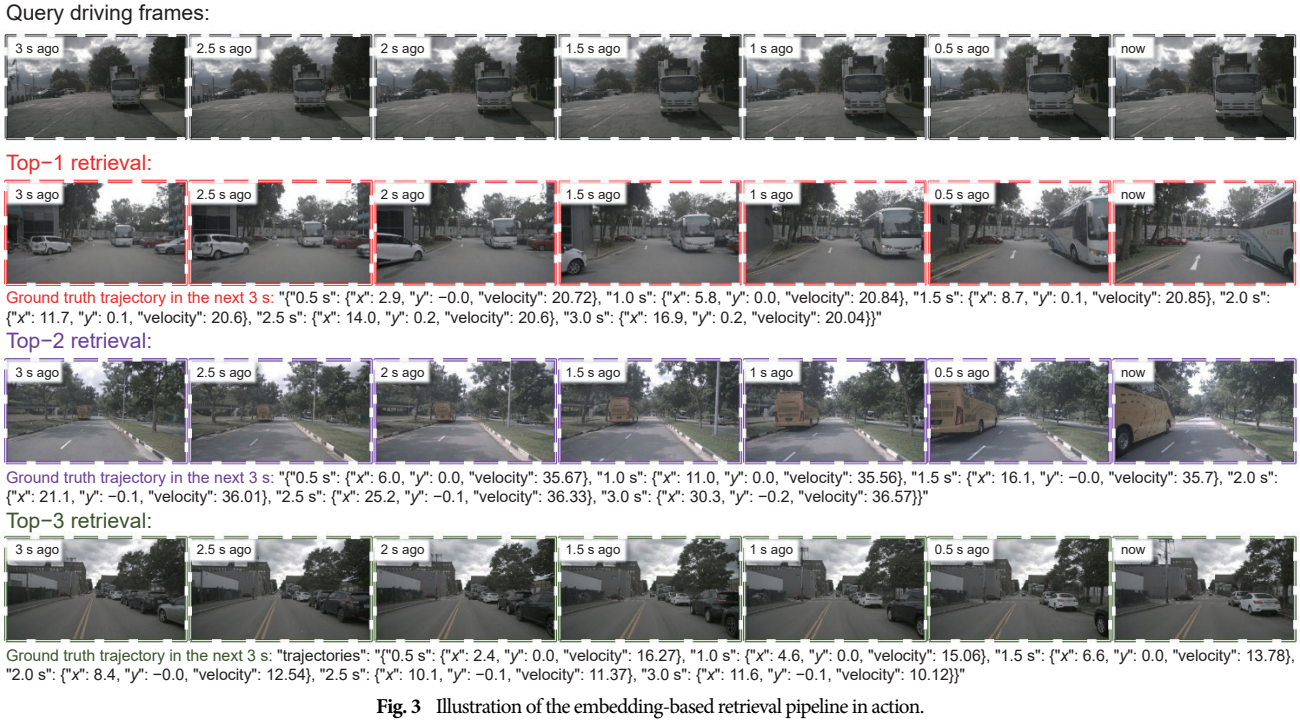

Fig. 3 Illustration of the embedding-based retrieval pipeline in action.

capabilities from front-view frames, while the second and third stages optimize trajectory prediction using surround-view images and consecutive front-view frames, respectively, along with ground-truth trajectories.

3.4.1 Stage A: Toward enhancement of the spatial perception capabilities of VLMs

In this stage, the vision encoder is frozen, and only the LLM backbone of the VLM is adapted to enhance spatial perception. We construct a spatial perception dataset from nuScenes in a VQA format, containing over 100,000 verified question-answer pairs that cover three perception tasks essential for planning: object classification, physical size estimation, and distance inference (Wang et al., 2025c). Examples of distance inference tasks in the training dataset are illustrated in Fig. 4.

The multitask loss for this stage is formulated as

$$\mathcal{L}_A = \frac{1}{B}\sum_{n=1}^{B}\left[\alpha_n \cdot \left(-\sum_{k=1}^{C}\eta_{k,n}\log\pi_{k,n}\right) + \beta_n \cdot \frac{1}{3}\sum_{d=1}^{3}\left(s_{d,n}-\hat{s}_{d,n}\right)^2 + \gamma_n \cdot \left(d_n-\hat{d}_n\right)^2\right] \tag{27}$$

where B denotes the batch size; C is the number of semantic categories; $\eta_{k,n}$ and $\pi_{k,n}$ represent the one-hot ground-truth label and predicted probability for category k of sample n, respectively; $s_{d,n}$ and $\hat{s}_{d,n}$ indicate the predicted and reference object size along dimension d, respectively; d_n and $\hat{d}_n$ refer to the predicted and actual distance to the ego vehicle, respectively; and α_n, β_n, and γ_n are binary indicators specifying whether the classification, size regression, and distance regression terms are active for that sample.

We adopt Qwen-series VLMs as the backbone and use LoRA (Hu et al., 2022a) as the fine-tuning framework. This stage aims to enhance the spatial perception ability of VLMs, thereby forming a robust perceptual foundation for subsequent fine-tuning.

3.4.2 Stage B: Toward scene-conditioned trajectory prediction from surround-view images

Stage B is designed to train VLMs to output trajectories from six input surround-view images. Each training instance includes the surround-view images, along with the ego vehicle's current status, the historical trajectory over the past three seconds, and a guiding way-point that indicates the intended direction within the prediction horizon.

Visual tokens are extracted by a frozen vision encoder and fused with the current velocity, the past three-second trajectory, and the way-point before entering the LLM backbone. The training objective fits coordinates and velocities through a standard regression loss, with simple regularizers that encourage smooth motion and reasonable curvature over time:

$$\mathcal{L}_B = \frac{1}{B}\sum_{r=1}^{B}\frac{1}{Q}\sum_{q=1}^{Q}\left[\left(x_{q,r}-\tilde{x}_{q,r}\right)^2+\left(y_{q,r}-\tilde{y}_{q,r}\right)^2 + \left(v_{q,r}-\tilde{v}_{q,r}\right)^2\right] \tag{28}$$

where $Q=3$ denotes the prediction horizons (1 s, 2 s, 3 s), $(x_{q,r}, y_{q,r}, v_{q,r})$ are the predicted egocentric x/y coordinates and velocity for sample r at horizon q, and tildes indicate their ground-truth counterparts. The LoRA fine-tuning is still applied in this stage.

This stage trains VLMs to generate physically realizable, short-term trajectories according to surround-view images. The resulting motion-aligned representation forms the bridge from Stage A's perception enhancement to Stage C's end-to-end trajectory prediction.

3.4.3 Stage C: toward end-to-end trajectory prediction from consecutive front-view frames

Stage C finalizes the triple-stage pipeline by training VLMs to produce short-horizon ego trajectories directly from a consecutive front-view frame sequence. Each training instance supplies seven front-view frames timestamped at −3.0, −2.5, −2.0, −1.5, −1.0, −0.5, 0.0 s relative to the present, a short history of ego positions and velocities expressed in the ego frame, and an optional soft way-point that provides a directional prior without imposing arrival within the prediction horizon.

Architecturally, the seven frames are tokenized by the frozen vision encoder and augmented with relative-time embeddings to preserve temporal order. These visual tokens are concatenated with structured kinematic features (history trajectory waypoints and velocities). Consistent with earlier stages, we retain a Qwen-series VLM and adapt only the LLM parameters via LoRA, which keeps adaptation parameter-efficient while focusing the optimization on temporal reasoning from consecutive frames.

To avoid mathematical duplication, Stage C reuses the trajectory regression objective introduced in Stage B ($\mathcal{L}_C = \mathcal{L}_B$), which jointly penalizes coordinate and velocity errors. In practice, Stage C minimizes $\mathcal{L}_C$ without introducing new loss terms, and only task-specific tuning of the regularization weights is performed to accommodate the reduced field-of-view and the stronger reliance on temporal cues.

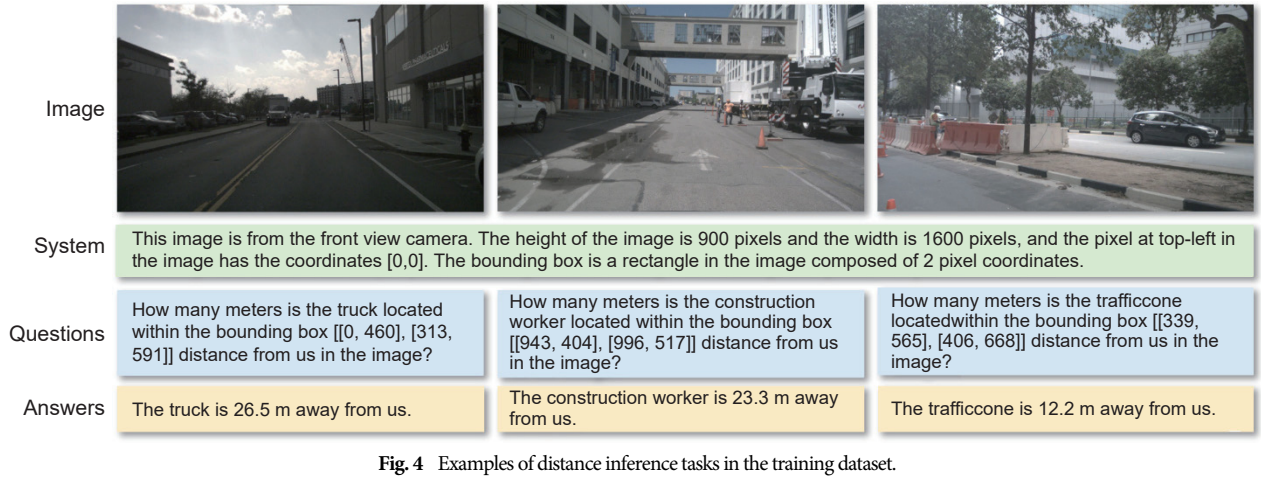

Fig. 4 Examples of distance inference tasks in the training dataset.

Stage C therefore completes this triple-stage fine-tuning strategy by converting the perception capabilities from Stage A and the surround-view priors from Stage B into a front-view-centric temporal planner. Stage C seamlessly integrates into the unified optimization framework while contributing front-view-specific temporal reasoning essential for practical, deployable single-camera planning.

3.4.4 Practical choice between LoRA and GRPO

While reinforcement learning (RL)-based preference optimization methods, such as group relative policy optimization (GRPO), have recently shown promise for aligning VLMs, our design objective in this work is to optimize short-horizon trajectory coordinates with physics-aware regularization and safety proxies under a RAG framework. In this setting, LoRA-based fine-tuning of the LLM backbone of a VLM offers three practical advantages:

Objective alignment and stability: Our loss directly matches the evaluation metrics (L2 displacement and collision rate). Supervised, gradient-based training with LoRA offers a straightforward and stable optimization process. In contrast, using GRPO-style reinforcement learning would require designing surrogate rewards for trajectories, handling temporal credit assignment over long horizons, and carefully tuning safety-related factors, which are all known to introduce instability and high variance in practice.

Data efficiency and modularity: LoRA updates only a small set of parameters and fits neatly into our triple-stage fine-tuning pipeline, without needing to unfreeze the vision encoders or retrain any reward models. This makes the training process parameter-efficient, easy to reproduce, and well aligned with the deployment-latency requirements of our retrieval pipeline.

For these reasons, we use LoRA in this work because it aligns well with our metric-supervised objectives and deployment needs, and it provides a stable and low-variance way to optimize the model. GRPO-style alignment is still valuable, but mainly for future efforts that focus on preference- or rule-based refinements such as comfort, driving style, or normative compliance, which are outside the scope of our current open-loop evaluation.

3.5 Chain-of-thought prompting paradigm

To strengthen the reasoning capabilities of VLMs in trajectory prediction for autonomous driving, we employ a CoT prompting strategy that encourages the VLM to reason step by step using retrieved reference scenes. Instead of relying on standard prompts that ask for a prediction based solely on the current scene, our method incorporates structured multimodal context and comparative reasoning so that the VLM can follow a more human-like decision process.

As illustrated in Fig. 5, our CoT prompting is organized around a multiscene structure. The model is first presented with several reference scenes that are retrieved through our embedding and retrieval pipeline, where the number of retrieved samples is controlled by the Top-K setting. After processing these reference scenes, the model receives the target scene for inference. Each reference scene contains the following components:

- A sequence of seven front-view images captured at time steps {3.0 s, 2.5 s, 2.0 s, 1.5 s, 1.0 s, 0.5 s, 0.0 s} before the present,
- Historical trajectory data, including position and velocity in the ego vehicle coordinate system,
- An optional navigation way-point representing a desired future direction,
- The ground truth future trajectory of the ego vehicle.

This exemplar-guided context helps the model form a clear understanding of how scene dynamics relate to vehicle motion and the types of future trajectories that are feasible. By examining multiple instances, the model learns to recognize recurring patterns, interpret scene-specific constraints, and transfer this understanding to new but related driving scenarios. To ensure coherent and safe trajectory generation, the system prompt

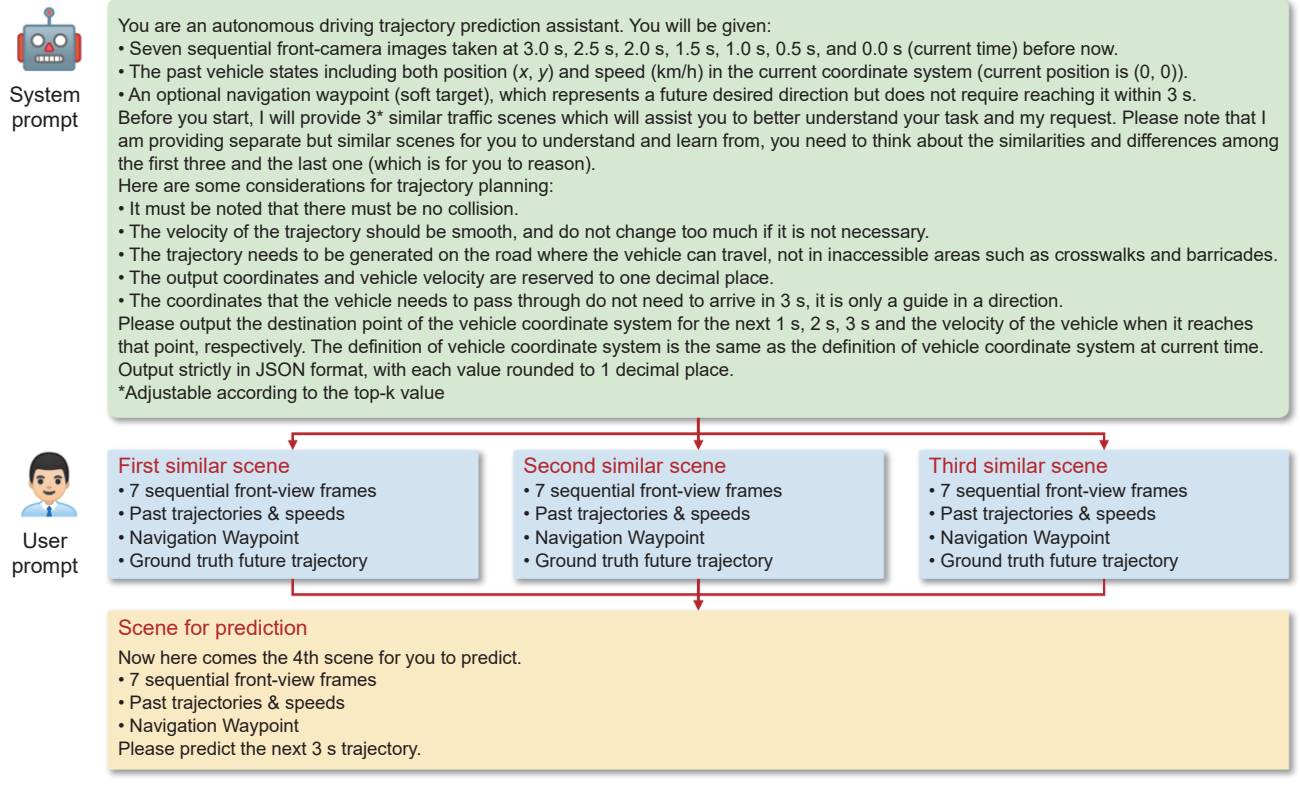

Fig. 5 Example of the CoT prompting paradigm.

specifies several concrete planning considerations, such as collision avoidance, velocity smoothness, and adherence to drivable areas. These considerations improve the feasibility of outputs.

A key feature of our CoT paradigm is its focus on relational reasoning. The model is guided to compare the reference scenes with the target scene and to identify both their common structure and distinguishing characteristics. This strengthens the transfer of knowledge across scenes and supports effective contextual adaptation. In addition, all spatial information is represented in a unified vehicle-centric coordinate system, and temporal cues are explicitly annotated, which provides clear temporal and spatial grounding for the trajectory prediction task.

Similarly, the prompt structure is designed so that the ego vehicle's dynamic status, such as velocity, acceleration, and yaw angle, can be incorporated directly during inference. These values could be supplied to the system without extra preprocessing, which allows the model to make more accurate and contextwise decisions.

4 Experiment

4.1 Experimental setup

4.1.1 Data preparation

We divide the 34,000 scenes from the nuScenes dataset (Caesar et al., 2020) into four subsets:

- The first subset consists of 10,000 scenes. After removing redundant frames that lack either the first 3 s of historical context or the last 3 s of future trajectory information, a total of 9062 overlapping scene sequences remain. We use this subset to train the TFSF vision encoder.
- The second subset also contains 10,000 scenes. In Stage A, we use the surround-view images from this subset to build a fine-tuning dataset to enhance the spatial perception abilities of the VLMs. In Stage B, we further fine-tune VLMs using both the surround-view images and corresponding 3-s future trajectory ground truths, enhancing its trajectory prediction performance. Following the same filtering strategy as in the first subset, the front-view frames of these 10,000 scenes are divided into 1117 nonoverlapping scene sequences. These sequences, paired with their 3-s future trajectory ground truths, are utilized in Stage C, serving as the final stage to jointly optimize spatiotemporal reasoning and trajectory prediction.
- The third subset also includes 10,000 scenes. As with the first subset, we extract 9062 overlapping scene sequences. Each sequence is paired with its corresponding 3-s future trajectory and encoded into a joint vector representation with the TFSF encoder. These representations are stored in a vector-based knowledge base, which is used in the RAG pipeline.
- The remaining 4000 scenes are used to construct the validation set, from which 3871 valid overlapping scene sequences are obtained. This subset is used to evaluate the overall performance of VLMs.

All training or fine-tuning subsets (for the TFSF encoder training, Stage-A/B/C fine-tuning, and building the database) are constructed exclusively from the nuScenes training data and are disjointed from the held-out official test set. All comparative results reported in Section 4.2 are obtained on the nuScenes official test split.

4.1.2 Evaluation metrics

The prediction of trajectories is evaluated using the L2 displacement error and the collision rate. More specifically, we adopt the L2 and collision rate based on UniAD (Hu et al., 2023) and ST-P3 (Hu et al., 2022b). UniAD's NoAvg protocol computes the error at each horizon using only the prediction made for that specific horizon, without any temporal averaging. In contrast, ST-P3's TemAvg protocol computes the score at a horizon as the cumulative arithmetic mean of the intermediate scores of 0.5 s up to that horizon, which smooths the results over time. In the comparative experiments, we report KEPT's performance under both protocols so that it is directly comparable with different baselines. In the ablation study, since external comparisons are not needed, we evaluate performance using the NoAvg protocol.

4.1.3 Vision-language model backbone

In our experiments, we adopt the Qwen2-VL-2B model as the default VLM backbone. As previously noted, this component is modular and can be substituted with other models in the Qwen family, including Qwen2-VL-7B, Qwen2.5-VL-3B, Qwen2.5-VL-7B, or larger-scale variants. We choose the 2B model because it requires far less computation during both training and inference. This makes it suitable for real-time use and for deployment on resource-limited platforms. A further ablation study on various VLM backbones is conducted in Section 4.3.5. All evaluations are conducted on four NVIDIA A800 GPUs.

4.2 Comparative experiments

All comparisons are performed on the official nuScenes test split using a shared open-loop setting. Baselines that operate on single-frame vision inputs are evaluated directly on the same test scenes without retraining. In contrast, KEPT processes a rolling three-second clip sampled at 2 Hz, which provides seven frames, and we report both the NoAvg (per-horizon) and TemAvg (temporal average up to the horizon) results to match common reporting conventions. Methods that make use of ego status information such as velocity, acceleration, or yaw angle are marked with an asterisk in Table 1. We provide the results of KEPT under both configurations to maintain fair comparisons. This evaluation setup preserves the intended usage of each baseline and highlights the specific benefit introduced by KEPT's short-horizon temporal context.

The methods summarized in Table 1 fall into four representative families, each incorporating geometry, temporal cues and semantics into the planning process in a different way. End-to-end BEV-centric planners such as UniAD (Hu et al., 2023), ST-P3 (Hu et al., 2022b), and BEV-Planner (Li et al., 2024) transform multiview images into BEV features and decode waypoints within a unified planning network. These methods are closely tied to the planning objective, yet they usually rely on single-frame inputs during inference, which could make long-horizon behavior sensitive to momentary visual noise. Vectorized approaches such as VAD-Base (Jiang et al., 2023a) model lanes and agents as structured vectors, which serve as geometric constraints, making the planner more disciplined and more efficient. Occupancy and world-model pipelines, such as OccNet (Ahmed et al., 2023) and Drive-OccWorld (Yang et al., 2025b), take different routes. They reconstruct or predict the multidimensional occupancy of the environment before generating a plan. This design provides clear geometric grounding, although it increases computation and extends the planning chain. The last group of methods uses VLMs or LLMs, including GPT-Driver (Mao et al., 2023), RDA-Driver (Huang et al., 2024b), DME-Driver (Han et al., 2025), DriveVLM (Tian et al., 2024), and Omni-Drive (Wang et al., 2025a). These models bring language-based reasoning into the planning process, which improves interpretability and supports higher-level decision

Table 1 Performance comparison with representative published baselines and our KEPT

NoAvg	TemAvg	Method	L2 (m)↓				Collision (%)↓			
			1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
×	×	FF (Hu et al., 2021b)	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
×	×	EO (Khurana et al., 2022)	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
√	×	ST-P3 (Hu et al., 2022b)	1.72	3.26	4.86	3.28	0.44	1.08	3.01	1.51
√	×	OccNet (Ahmed et al., 2023)	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
√	×	UniAD (Hu et al., 2023)	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
√	×	VAD-Base (Jiang et al., 2023a)	0.54	1.15	1.98	1.22	0.10	0.24	0.96	0.43
√	×	Drive-OccWorld (Yang et al., 2025b)	0.32	0.75	1.49	0.85	0.05	0.17	0.64	0.29
√	×	KEPT (ours, on Qwen2-VL-2B)	0.23	0.63	1.23	0.70	0.05	0.14	0.44	0.21
×	√	ST-P3 (Hu et al., 2022b)	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
×	√	UniAD (Hu et al., 2023)	0.44	0.67	0.96	0.69	0.04	0.08	0.23	0.12
×	√	VAD-Base (Jiang et al., 2023a)	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
×	√	Drive-OccWorld (Yang et al., 2025b)	0.25	0.44	0.72	0.47	0.03	0.08	0.22	0.11
×	√	KEPT (ours, on Qwen2-VL-2B)	0.17	0.35	0.59	0.37	0.03	0.10	0.34	0.16
√	×	GPT-Driver* (Mao et al., 2023)	0.27	0.74	1.52	0.84	0.07	0.15	1.10	0.34
√	×	RDA-Driver* (Huang et al., 2024b)	0.23	0.73	1.54	0.80	0.00	0.13	0.83	0.32
√	×	DME-Driver* (Han et al., 2025)	0.43	0.91	1.58	0.97	0.04	0.14	0.64	0.27
√	×	VLP* (Pan et al., 2024)	0.36	0.68	1.19	0.74	0.03	0.12	0.32	0.16
√	×	KEPT* (ours, on Qwen2-VL-2B)	0.19	0.52	1.12	0.61	0.02	0.11	0.29	0.14
×	√	UniAD* (Hu et al., 2023)	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37
×	√	VAD-Base* (Jiang et al., 2023a)	0.17	0.34	0.60	0.37	0.04	0.27	0.67	0.33
×	√	GPT-Driver* (Mao et al., 2023)	0.20	0.40	0.70	0.44	0.04	0.12	0.36	0.17
×	√	RDA-Driver* (Huang et al., 2024b)	0.17	0.37	0.69	0.40	0.01	0.05	0.26	0.10
×	√	BEV-Planner* (Li et al., 2024)	0.16	0.32	0.57	0.35	0.00	0.29	0.73	0.34
×	√	VLP* (Pan et al., 2024)	0.30	0.53	0.84	0.55	0.01	0.07	0.38	0.15
×	√	DriveVLM* (Tian et al., 2024)	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27
×	√	OmniDrive* (Wang et al., 2025a)	0.14	0.29	0.55	0.33	0.00	0.13	0.78	0.30
×	√	Drive-OccWorld* (Yang et al., 2025b)	0.17	0.31	0.49	0.32	0.02	0.24	0.62	0.29
×	√	KEPT* (ours, on Qwen2-VL-2B)	0.14	0.28	0.51	0.31	0.01	0.06	0.13	0.07

Note: The results from preprints are excluded. ↓ indicates "lower is better". * indicates the use of ego status as inputs.

making. They often require additional components to connect their text-based reasoning to accurate geometric constraints.

From this perspective, KEPT follows a different design strategy. It introduces short-horizon temporal fusion at the input stage through the TFSF encoder, incorporates compact retrieval priors by RAG, and performs decoding with CoT prompts. With these design choices, the results in Table 1 form a clear trend. KEPT performs better under both the NoAvg and TemAvg protocols. This suggests that its improvements reflect real gains in trajectory quality. Temporal averaging mainly smooths out frame-to-frame noise. Since KEPT already takes in temporally coherent inputs, averaging simply makes this stability more visible, especially compared with single-frame baselines.

At the same time, the largest performance gaps in Table 1 appear at the far-horizon range of 2–3 s. This is exactly where single-frame BEV planners tend to fail, especially under low visibility or partial occlusion. In such conditions, some water reflections or shadows could be mistaken for real structures and cause lateral drift. Additionally, the lack of historical motion cues could delay longitudinal responses. KEPT addresses these issues in two complementary ways. The TFSF encoder extracts features from a short rolling clip, which strengthens signals that remain stable across frames while reducing the influence of transient artifacts. The RAG pipeline then provides additional prior knowledge to help stabilize the decoder when the visual evidence becomes sparse. By the 2–3 s horizon, this combination produces trajectories that stay closer to the ground truth and maintain a smoother velocity pattern.

Moreover, comparisons within each method family help clarify practical trade-offs. Compared with occupancy and world-model pipelines, KEPT retains temporal consistency and avoids the heavy cost of multidimensional forecasting and the long generative chain. This makes KEPT more suitable for deployment under realistic computational limits. Compared with other VLM-based planners, KEPTs give up detailed natural-language explanations in exchange for better spatial grounding. This trade-off leads to higher-quality open-loop waypoints.

In conclusion, KEPT's performance should be understood not as a small leaderboard gain but as evidence that combining short-horizon temporal fusion with RAG and CoT leads to more stable far-horizon trajectories across a range of methods on a shared benchmark.

4.3 Ablation studies

4.3.1 Ablation study on fine-tuning stages

We assess the contribution of the three fine-tuning stages by selectively enabling or disabling Stage A and Stage B while keeping Stage C active. The numerical results are provided in Table 2. As discussed in Section 3.4, Stage A enhances spatial perception, Stage B adds motion reasoning with surround-view images, and Stage C directly optimizes trajectory prediction on consecutive front-view frames.

1) Without ego status. Enabling all three stages produces the best accuracy and safety, with an average L2 error of 0.71 m and

an average collision rate of 0.24%. When Stage A is removed, the average L2 increases to 0.76 m, and the collision rate rises to 0.48%. Removing Stage B leads to an average L2 of 0.78 m and a collision rate of 0.49%. The largest differences appear at longer horizons. The 3-s L2 decreases from 1.40 to 1.41 m (without Stage A or without Stage B) to 1.25 m when all stages are active, while the 1-s L2 remains close at 0.23 m. This shows that the combination of Stage A and Stage B mainly strengthens long-horizon stability.

2) With ego status. With all stages active, the KEPT reaches an average L2 error of 0.63 m and a collision rate of 0.16%. Removing Stage A or Stage B increases the average L2 to 0.75 m, with collision rates of 0.45% and 0.51%, respectively. Differences at shorter horizons remain small, while most of the improvements come from longer horizons. The 3-s L2 decreases from 1.39 or 1.36 to 1.13 m, and the 2-s L2 decreases from 0.64 or 0.66 to 0.56 m when combining all three stages together.

Across both usage modes, having Stage A and Stage B together in addition to Stage C is crucial. Removing either stage weakens the long-horizon accuracy and increases the collision risk, while the full triple-stage fine-tuning strategy consistently ensures the most accurate and safest trajectories. This result is consistent with the design goals outlined in Section 3.4.

4.3.2 Ablation study on the Top-K value of RAG

Table 3 evaluates how many retrieved examples (K) should be provided. All variants use the same triple-stage fine-tuning strategy so that the effect of retrieval depth can be examined in isolation. In line with the RAG pipeline in Section 3.3 and the CoT integration in Section 3.5, the results follow a concave pattern. A small Top-K offers useful priors that complement the target scene, while a large Top-K weakens the planner's reasoning.

1) Without ego status. Increasing the retrieval depth from Top-0 (without any retrieval scenes) to Top-2 improves both accuracy and safety. The average L2 decreases from 0.73 to 0.70 m, and the average collision rate decreases from 0.26% to 0.21%. The largest improvements appear at the 3-second horizon, where L2 drops from 1.30 to 1.23 m and the collision rate decreases from 0.55% to 0.44%. When K is increased beyond 2, the trend reverses, which suggests that many retrieved scenes introduce noise rather than helpful context.

2) With ego status. The Top-2* configuration reaches an average L2 of 0.61 m and an average collision rate of 0.14%. This represents an improvement over Top-0*. Long-horizon metrics are also improved, with the 3-s L2 decreasing from 1.15 to 1.12 m and the collision rate decreasing from 0.38% to 0.29%. As before, using a large Top-K value leads to worse performance.

A small but effective retrieval set works best in our pipeline. Using $K = 2$ consistently provides the strongest balance between accuracy and safety. This matches the motivation in Sections 3.3 and 3.5. For this reason, we use Top-2 as the default setting for the main results in Section 4.2.

4.3.3 Ablation study on different retrieval strategies

To evaluate the effectiveness of our retrieval strategy, we conduct an ablation study comparing the proposed k-means & HNSW pipeline with a simple similarity search, a k-means only variant, and a standard HNSW search. Since clustering and HNSW indexing have little influence on retrieval accuracy but significantly reduce retrieval time, we focus on their efficiency benefits. For this ablation, we report the average time required to retrieve results for a single test scene, which serves as the primary measure of retrieval performance.

For each retrieval strategy, we evaluate the performance using

Table 2 Ablation study on fine-tuning stages

Fine-tuning stage			Ego status	L2 (m)↓				Collision (%)↓			
Stage A	Stage B	Stage C	Current (v, a, yaw)	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
×	√	√	×	0.23	0.65	1.40	0.76	0.15	0.39	0.89	0.48
√	×	√	×	0.23	0.69	1.41	0.78	0.13	0.42	0.93	0.49
√	√	√	×	0.23	0.64	1.25	0.71	0.06	0.16	0.51	0.24
×	√	√	√	0.23	0.64	1.39	0.75	0.14	0.37	0.83	0.45
√	×	√	√	0.22	0.66	1.36	0.75	0.11	0.44	0.97	0.51
√	√	√	√	0.21	0.56	1.13	0.63	0.03	0.13	0.31	0.16

Note: Only the NoAvg protocol is adopted. ↓ indicates lower is better. The Top-K value of RAG is set to 1 accordingly. The VLM backbone is Qwen2-VL-2B.

Table 3 Ablation study on the Top-K value of RAG

Top-K Retrieval	L2 (m)↓				Collision (%)↓			
	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Top-0 (w/o RAG)	0.24	0.66	1.30	0.73	0.06	0.18	0.55	0.26
Top-1	0.23	0.64	1.25	0.71	0.06	0.16	0.51	0.24
Top-2	0.23	0.63	1.23	0.70	0.05	0.14	0.44	0.21
Top-3	0.25	0.68	1.27	0.73	0.07	0.19	0.59	0.28
Top-4	0.31	0.78	1.41	0.83	0.09	0.23	0.74	0.35
Top-0* (w/o RAG)	0.21	0.58	1.15	0.65	0.05	0.15	0.38	0.19
Top-1*	0.21	0.56	1.13	0.63	0.03	0.13	0.31	0.16
Top-2*	0.19	0.52	1.12	0.61	0.02	0.11	0.29	0.14
Top-3*	0.22	0.58	1.17	0.66	0.07	0.19	0.44	0.23
Top-4*	0.24	0.62	1.26	0.71	0.08	0.21	0.53	0.27

Note: ↓indicates lower is better. * indicates the use of ego status as inputs. Only the NoAvg protocol is adopted. The VLM backbone is Qwen2-VL-2B.

Top-K values from 1 to 5. We then compute the final metric as the arithmetic mean across these Top-K settings, giving a balanced view of retrieval speed. As shown in Table 4, a simple cosine-similarity search already achieves millisecond-level performance on the database of nearly ten thousand scene embeddings. Adding k-means clustering lowers the average query time to approximately 0.1 ms. HNSW offers an additional improvement, reducing the retrieval latency to approximately 0.03 ms. Our k-means & HNSW method achieves the best efficiency, with an average retrieval time of 0.014 ms per query. All retrieval tests are run on a single Intel i7-13700 CPU without GPU acceleration.

4.3.4 Ablation study on various vision encoders

To validate the design of the TFSF encoder, we compare three vision encoders in this ablation, namely, a temporal spatial-only variant that aggregates per-frame features over time in the image spatial domain, a temporal frequency-only variant that operates purely in the temporal frequency domain, and the full TFSF encoder that combines both branches. The results are summarized in Table 5.

The numbers show a consistent pattern. Removing either branch leads to a degradation in both L2 and collision rate, while the full TFSF encoder achieves the best performance across all horizons. The gap is particularly clear at 2-3 s, where single-branch encoders are more prone to lateral drift and delayed reactions.

The temporal spatial branch maintains a high-fidelity, multiscale description of the scene layout as frames evolve, which is crucial for understanding static obstacles and road geometry. In contrast, the temporal frequency branch emphasizes temporal variation in the feature space, which helps the encoder remain stable when objects move in and out of view or when lighting conditions change suddenly. When only the spatial branch is used, the encoder lacks robustness to blocks and inadequate lighting, and its long-horizon predictions become more imprecise. When only the frequency branch is used, the vision encoding loses spatial precision. The full TFSF encoder brings these two perspectives together and therefore achieves better performance in downstream tasks.

4.3.5 Ablation study on various VLM backbones

To examine how KEPT performs with different VLM backbones, we replace Qwen2-VL-2B in our pipeline with three other Qwen-series VLMs, namely, Qwen2-VL-7B, Qwen2.5-VL-3B, and Qwen2.5-VL-7B. Table 6 summarizes the L2 displacement error, collision rate, and average inference time, which is crucial for the potential application of the framework.

The results indicate that increasing the parameter scale of VLM backbones can indeed bring measurable benefits to both the L2 and collision rate metrics. Both 7B VLMs improve upon the 2B baseline, and Qwen2.5-VL-7B achieves the lowest errors, with consistent gains at 1, 2, and 3 s. Since the parameter scale of Qwen2.5-VL-3B is comparable to that of Qwen2-VL-2B, their overall performance is also quite similar.

At the same time, this ablation study also makes it clear that these gains come with a substantial increase in inference time. Both 7B VLMs, although they achieve better performance, still incur noticeable slow-downs. From the perspective of an autonomous driving system, a longer inference time directly increases computational cost and could significantly harm real-time performance. In addition, the 7B VLMs are clearly harder to deploy on a vehicle than the 2B VLM, especially once RAG is enabled and the prompt becomes longer, which further raises the memory requirements.

Table 4 Ablation study on the retrieval time of different retrieval strategies (Unit: ms)

Retrieval strategy	Top-1 Avg.	Top-2 Avg.	Top-3 Avg.	Top-4 Avg.	Top-5 Avg.	Overall Avg.
Simple search	0.766	0.783	0.748	0.796	0.756	0.770
k-means only	0.162	0.159	0.149	0.150	0.150	0.154
HNSW only	0.032	0.033	0.031	0.032	0.034	0.032
k-means & HNSW	0.014	0.014	0.015	0.013	0.014	0.014

Table 5 Ablation study on vision encoders

Vision encoder	L2 (m)↓				Collision (%)↓			
	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Temporal spatial-only*	0.22	0.61	1.23	0.69	0.04	0.19	0.41	0.21
Temporal frequency-only*	0.23	0.62	1.25	0.70	0.04	0.21	0.48	0.24
TFSF*	0.21	0.56	1.13	0.63	0.03	0.13	0.31	0.16

Note: Only the NoAvg protocol is adopted. ↓ indicates lower is better. * indicates the use of ego status as inputs. The Top-K value of RAG is set to 1 accordingly. The VLM backbone is Qwen2-VL-2B.

Table 6 Ablation study on VLM backbones

VLM backbone	L2 (m)↓				Collision (%)↓				Avg. Inf. Time (s)↓
	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.	
Qwen2-VL-2B*	0.21	0.56	1.13	0.63	0.03	0.13	0.31	0.16	7.36
Qwen2-VL-7B*	0.20	0.53	1.09	0.61	0.03	0.11	0.28	0.14	11.27
Qwen2.5-VL-3B*	0.21	0.54	1.11	0.62	0.03	0.13	0.33	0.16	8.11
Qwen2.5-VL-7B*	0.19	0.52	1.07	0.59	0.03	0.10	0.26	0.13	14.43

Note: Only the NoAvg protocol is adopted. ↓ indicates lower is better. * indicates the use of ego status as inputs. The Top-K value of RAG is set to 1 accordingly.

In summary, this ablation shows that KEPT could benefit from larger VLMs. However, when jointly considering accuracy and computational cost, Qwen2-VL-2B offers the most balanced trade-off. We therefore adopt Qwen2-VL-2B as the default VLM backbone for all main experiments and regard the 7B VLMs as upper-bound variants that illustrate the potential of the framework when latency constraints are relaxed.

4.4 Case studies

In this subsection, we examine six representative scenes from the nuScenes official test set: three long-tail success and three failure cases to evaluate model behavior in challenging, long-tail conditions. For the failure cases, we also demonstrate how prompt refinement could benefit the predicted trajectories.

4.4.1 Long-tail success case analysis

In Fig. 6, we demonstrate how the KEPT provides feasible trajectories in three long-tail cases. Scene #1 shows that the ego vehicle is entering a nonstandard intersection with relatively low visibility. Temporary barriers, the bus, and weak lane markings make it difficult to determine the correct path. Scene #2 shows that the ego vehicle is driving along an urban straight road in heavy rain. The wet surface produces strong reflections, and the right side is occupied by a construction area. Scene #3 shows that the ego vehicle is approaching a nonstandard intersection with ambiguous lane boundaries. In all demonstrated cases, KEPT could predict trajectories with good alignment to ground truth trajectories.

These successful cases provide a concrete view of how KEPT performs in such long-tail situations. The TFSF encoder helps to capture both frequency and spatial patterns when parts of the scene are occluded or when illumination varies from frame to frame. The retrieved reference scenes provide additional prior knowledge, which is further combined with CoT prompts to ensure feasible trajectories. These components together guarantee reasonable motion planning even when the road layout and visual condition are not ideal.

4.4.2 Long-tail failure case analysis

In Fig. 7, we also demonstrate that the KEPT might fail in scene understanding and trajectory prediction in several cases. Scene #4 shows that the ego vehicle is approaching a T-junction with a car trying to merge from the left side road. Scene #5 shows that the ego vehicle is driving along an urban road with the current lane occupied by a slow vehicle ahead. Scene #6 shows that the ego vehicle is trying to turn left at an irregular intersection with a bus stopped ahead.

In Scene #4, the ego vehicle is expected to decelerate to avoid collision with the merging car, as the ground truth trajectory shows. However, the trajectory predicted by KEPT does not tend to decelerate, which might lead to potential collisions. In Scene #5, the ego vehicle might change lanes to the right early to ensure safety, while the predicted trajectory seems to hold the current lane. In Scene #6, the ground truth trajectory indicates that it is safe for the ego vehicle to turn left, but KEPT chooses to stop at the intersection, as all predicted way-points are (0.0, 0.0).

Since the overall retrieval and fine-tuning paradigm is fixed, possible improvement of the predicted trajectories is expected with slight modifications on prompts. We therefore refine the CoT prompts according to different failure cases. Specifically, we change the listed considerations in the CoT prompting paradigm in Fig. 5 according to different failure cases, and the refined parts are listed in Fig. 8.

In Fig. 9, we demonstrate the improved trajectories by refining the prompts of the failure cases. In Scene #4, with the help of the modified prompts, KEPT captures the merging tendency of the car in the intersection and therefore generates a trajectory of slowing down. In Scene #5, KEPT notices the slow car ahead and recognizes the potential blockage of the current lane and therefore decides to change lanes to the right. In Scene #6, according to the refined prompt, the KEPT regards the current situation as safe and therefore generates a left-turn trajectory. All improved trajectories maintain good alignment with the ground truth, suggesting that the quality of generated trajectories could benefit from modifications on the CoT prompts in case of failure.

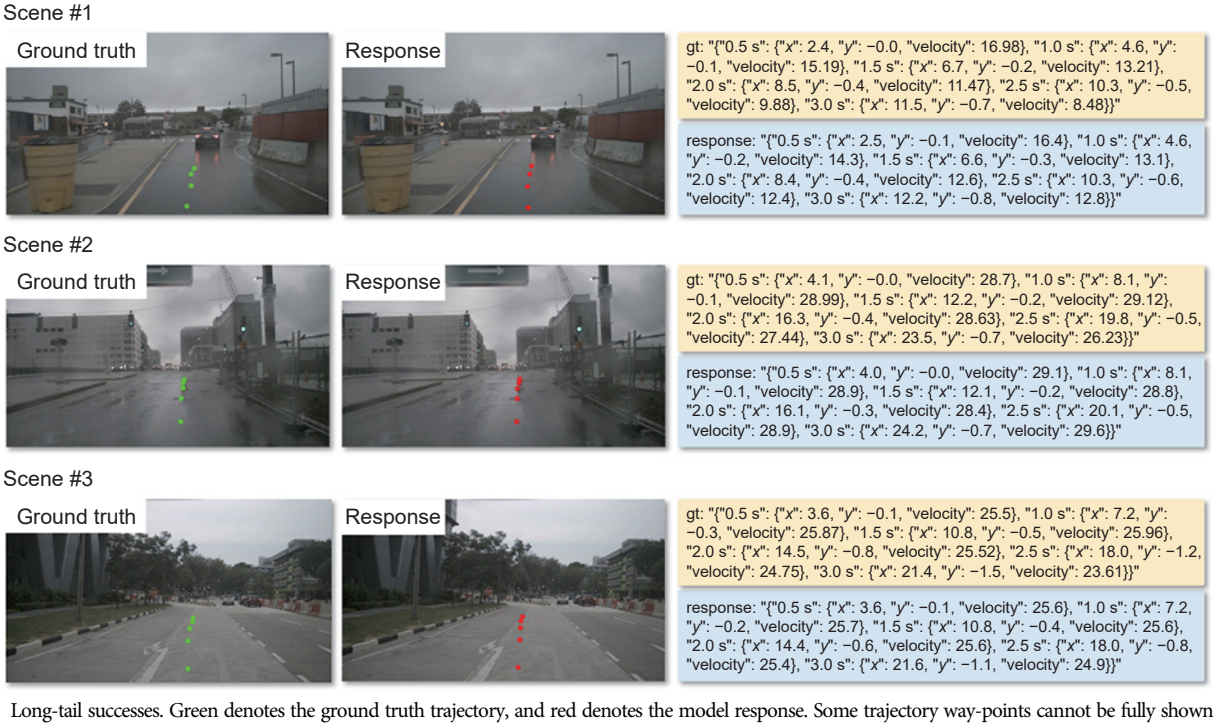

Fig. 6 Long-tail successes. Green denotes the ground truth trajectory, and red denotes the model response. Some trajectory way-points cannot be fully shown in the front-view image due to field-of-view limits. The Top-K value of RAG is set to 2 accordingly. The VLM backbone is Qwen2-VL-2B.

Fig. 7 Failure case analysis. Green denotes the ground truth trajectory, and red denotes the model response. Some trajectory way-points cannot be fully shown in the front-view image due to field-of-view limits. The Top-K value of RAG is set to 2 accordingly. The VLM backbone is Qwen2-VL-2B.

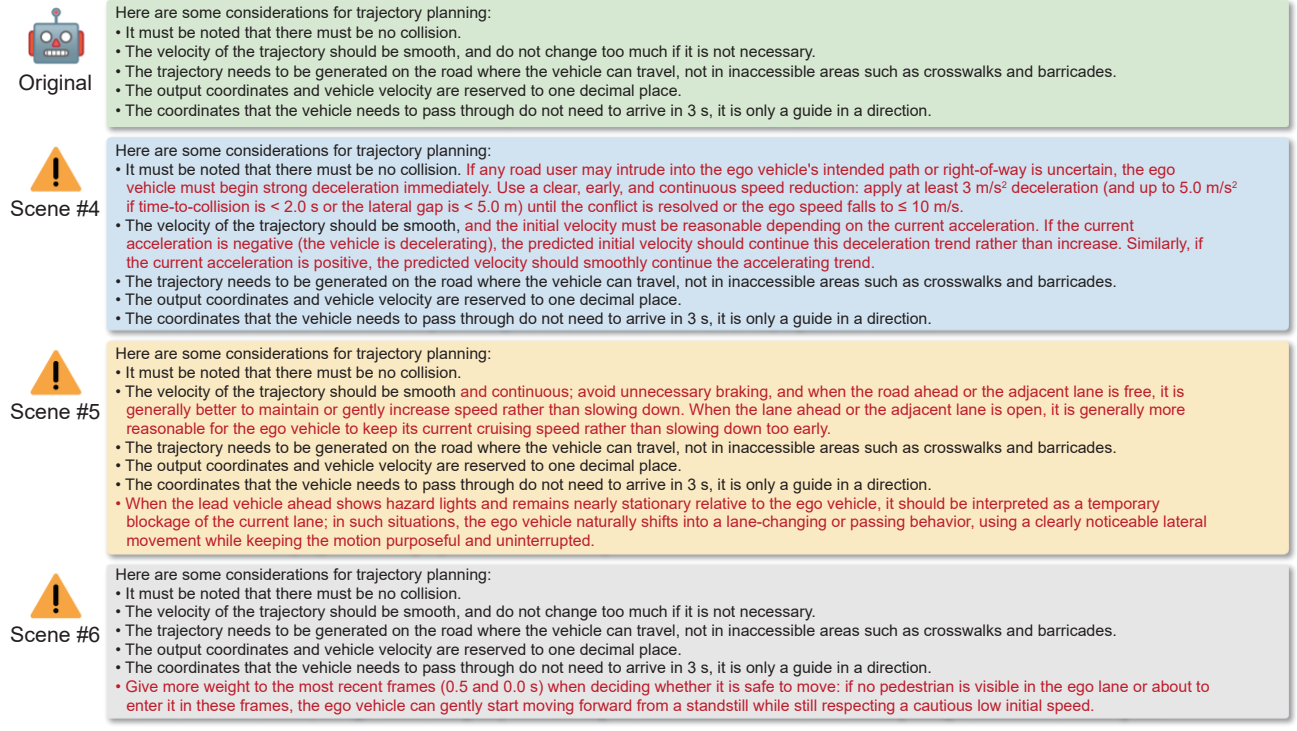

Fig. 8 Refined prompts according to different failure cases. Only the considerations part of system prompts is modified. The supplementary considerations aim to guide the VLM to understand the current scene better, and to execute more appropriate actions.

5 Conclusions

In this study, we introduce KEPT, a retrieval-augmented VLM framework for short-horizon trajectory prediction from consecutive driving frames. KEPT integrates a TFSF encoder that produces motion-aware scene embeddings, a scalable retrieval module based on k-means clustering and HNSW indexing, a triple-stage fine-tuning strategy for the VLM backbone, and a CoT prompting paradigm designed around planning constraints.

Experiments on the nuScenes dataset show that KEPT improves a range of baseline methods across both open-loop protocols. Under NoAvg, it achieves lower L2 errors and collision rates than recent modular and VLM-based planners. Under TemAvg, it stays competitive without ego-status inputs and delivers the best accuracy-safety balance when ego information is provided. Ablation studies highlight two key factors. Stages A, B, and C should work together, for removing either spatial perception or trajectory-supervised reasoning fine-tuning harms

Fig. 9 Improved trajectories by refining prompts of the failure cases. Green denotes the ground truth trajectory, and red denotes the model response. Some trajectory way-points cannot be fully shown in the front-view image due to field-of-view limits. The Top-K value of RAG is set to 2 accordingly. The VLM backbone is Qwen2-VL-2B.

long-horizon accuracy and safety. Additionally, Top-2 retrieval consistently offers the clearest gains by providing useful priors without adding noise. Finally, the proposed k-means & HNSW retrieval achieves submillisecond latency, supporting real-time use.

Despite these gains, KEPT has several limitations. Our retrieval priors are built from nuScenes-like distributions, coverage and clustering granularity bound performance, and long-tail or out-of-domain scenes (e.g., extreme weather, unusual road rules) may yield suboptimal priors. The TFSF encoder currently uses seven 2 Hz front-view frames and predicts short horizons; richer sensors and longer temporal context could improve stability. CoT prompting adds variability and modest latency, and our constraint set is partly hand-crafted. Finally, we report open-loop metrics; closed-loop, real-time evaluations on a vehicle are required to verify safety and robustness.

Future work will therefore (1) extend KEPT to multisensor inputs and cross-city generalization with domain adaptation and (2) develop active and safety-aware retrieval that updates the knowledge base in real time. In addition, RL-based alignment (e.g., GRPO) is orthogonal and complementary to our present approach. A principled integration couples our supervised LoRA stages with (1) a reward model encoding multiobjective driving preferences (e.g., comfort, legality, social compliance) and (2) simulation- or dataset-in-the-loop rollouts for closed-loop credit alignment. We anticipate two promising paths: the first path is LoRA followed by GRPO, where LoRA provides stable metric grounding and GRPO performs light posttraining for preference trade-offs; the second path is constrained RL, where retrieval-supplied priors and CoT constraints act as safety shields during policy updates. We leave these preference-centric extensions to future work, as they require additional infrastructure (reward definition, rollouts, and safety monitors) beyond the open-loop, metric-supervised scope of this paper. We believe these directions, together with the lightweight nature of the Qwen-series backbones used here, make KEPT a promising foundation for trustworthy, interpretable, and scalable VLM planning in autonomous driving.

Author contributions

Yujin Wang: Writing – original draft, Validation, Methodology, Data curation, Experiments, Conceptualization. **Tianyi Wang**: Writing – original draft, Validation. **Quanfeng Liu**: Data curation, Experiments. **Wenxian Fan**: Data curation, Experiments. **Junfeng Jiao**: Writing – review & editing. **Christian Claudel**: Writing – review & editing. **Yunbing Yan**: Writing – review & editing. **Bingzhao Gao**: Supervision, Methodology, Formal analysis, Funding acquisition. **Jianqiang Wang**: Formal analysis, Funding acquisition. **Hong Chen**: Supervision, Formal analysis.

Replication and data sharing

The source codes and replication package are available on ETS data at https://doi.org/10.26599/ETSD.2025.9190073.

Acknowledgements

This research was supported by the National Key R&D Program of China (No. 2023YFB2504400), the National Nature Science Foundation of China (Nos. 62088101, 62373289, and 62273256), Shanghai Municipal Science and Shanghai Automotive Industry Science and Technology Development Foundation (No. 2407), and the Fundamental Research Funds for the Central Universities.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Jianqiang Wang is the Associate Editor member of this journal.

References

Ahmed, A., Zeng, X., Tunio, M. H., Butt, M. H., Shah, S. A., Yan, C., et al., 2023. OCCNET: Improving imbalanced multi-centred ovarian cancer subtype classification in whole slide images. In: 2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 1–8.

Akan, A. K., Güney, F., 2022. StretchBEV: Stretching future instance pre-

diction spatially and temporally. In: Computer Vision – ECCV, 444–460.

Atakishiyev, S., Salameh, M., Yao, H., Goebel, R., 2024. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. *IEEE Access*, **12**, 101603–101625.

Bandyopadhyay, S., Cole, J., Goldstein, T., Jacobs, D., 2025. Multimodal agentic model predictive control. In: Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, 2844–2848.

Brandstätter, F., Schütz, E., Winter, K., Flohr, F. B., 2025. BEV-LLM: Leveraging multimodal BEV maps for scene captioning in autonomous driving. In: 2025 IEEE Intelligent Vehicles Symposium (IV), 345–350.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., et al., 2020. nuScenes: A multimodal dataset for autonomous driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11618–11628.

Casas, S., Gulino, C., Liao, R., Urtasun, R., 2020. SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), 9491–9497.

Casas, S., Sadat, A., Urtasun, R., 2021. MP3: A unified model to map, perceive, predict and plan. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14398–14407.

Chai, Y., Sapp, B., Bansal, M., Anguelov, D., 2019. Multipath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. https://arxiv.org/abs/1910.05449

Chen, D., Krähenbühl, P., 2022. Learning from all vehicles. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17201–17210.

Chen, L., Sinavski, O., Hünermann, J., Karnsund, A., Willmott, A. J., Birch, D., et al., 2024a. Driving with LLMs: Fusing object-level vector modality for explainable autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), 14093–14100.

Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H., 2024b. End-to-end autonomous driving: Challenges and frontiers. *IEEE Trans Pattern Anal Mach Intell*, **46**, 10164–10183.

Chen, X., Huang, L., Ma, T., Fang, R., Shi, S., Li, H., 2025. SOLVE: Synergy of language-vision and end-to-end networks for autonomous driving. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12068–12077.

Chib, P. S., Singh, P., 2024. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Trans Intell Veh*, **9**, 103–118.

Chitta, K., Prakash, A., Geiger, A., 2021. NEAT: Neural attention fields for end-to-end autonomous driving. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 15773–15783.

Cui, C., Ma, Y., Cao, X., Ye, W., Wang, Z., 2024. Receive, reason, and react: Drive as you say, with large language models in autonomous vehicles. *IEEE Intell Transp Syst Mag*, **16**, 81–94.

Dai, X., Guo, C., Tang, Y., Li, H., Wang, Y., Huang, J., et al., 2024. VistaRAG: Toward safe and trustworthy autonomous driving through retrieval-augmented generation. *IEEE Trans Intell Veh*, **9**, 4579–4582.

Fu, D., Li, X., Wen, L., Dou, M., Cai, P., Shi, B., et al., 2024. Drive like a human: Rethinking autonomous driving with large language models. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 910–919.

Gao, X., Wu, Y., Wang, R., Liu, C., Zhou, Y., Tu, Z., 2025. LangCoop: Collaborative driving with language. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 4226–4237.

Han, W., Guo, D., Xu, C. Z., Shen, J., 2025. DME-driver: Integrating human decision logic and 3D scene perception in autonomous driving. *Proc AAAI Conf Artif Intell*, **39**, 3347–3355.

Hegde, D., Yasarla, R., Cai, H., Han, S., Bhattacharyya, A., Mahajan, S., et al., 2025. Distilling multimodal large language models for autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference, 27575–27585.

Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., et al., 2021a. FIERY: Future instance prediction in bird's-eye view from surround monocular cameras. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 15253–15262.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al., 2022a. LoRA: Low-rank adaptation of large language models. https://doi.org/10.48550/arXiv.2106.09685

Hu, P., Huang, A., Dolan, J., Held, D., Ramanan, D., 2021b. Safe local motion planning with self-supervised freespace forecasting. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12727–12736.

Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D., 2022b. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: Computer Vision – ECCV, 533–549.

Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., et al., 2023. Planning-oriented autonomous driving. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17853–17862.

Huang, J., Ye, Y., Liang, Z., Shan, Y., Du, D., 2024a. Detecting as labeling: Rethinking LiDAR-camera fusion in 3D object detection. In: Computer Vision – ECCV, 439–455.

Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J., 2023. Tri-perspective view for vision-based 3D semantic occupancy prediction. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9223–9232.

Huang, Z., Tang, T., Chen, S., Lin, S., Jie, Z., Ma, L., et al., 2024b. Making large language models better planners with reasoning-decision alignment. In: Computer Vision – ECCV, 73–90.

Hussien, M. M., Melo, A. N., Ballardini, A. L., Maldonado, C. S., Izquierdo, R., Sotelo, M. Á., 2025. RAG-based explainable prediction of road users behaviors for automated driving using knowledge graphs and large language models. *Expert Syst Appl*, **265**, 125914.

Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., et al., 2023a. VAD: Vectorized scene representation for efficient autonomous driving. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 8306–8316.

Jiang, Z., Xu, F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., et al., 2023b. Active retrieval augmented generation. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 7969–7992.

Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D., Ramanan, D., 2022. Differentiable raycasting for self-supervised occupancy forecasting. In: Computer Vision – ECCV, 353–369.

Li, J., Li, D., Savarese, S., Hoi, S., 2023a. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models, In: Proceedings of the 40th International Conference on Machine Learning , 19730–19742.

Li, P., Ding, S., Chen, X., Hanselmann, N., Cordts, M., Gall, J., 2023b. PowerBEV: A powerful yet lightweight framework for instance prediction in bird's-eye view. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 1080–1088.

Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., et al., 2025. BEVFormer: Learning bird's-eye-view representation from LiDAR-camera *via* spatiotemporal transformers. *IEEE Trans Pattern Anal Mach Intell*, **47**, 2020–2036.

Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., et al., 2024. Is ego status all you need for open-loop end-to-end autonomous driving? In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14864–14873.

Liang, M., Yang, B., Zeng, W., Chen, Y., Hu, R., Casas, S., et al., 2020. PnPNet: End-to-end perception and prediction with tracking in the loop. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11550–11559.

Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., et al., 2022. BEVFusion: A Simple and Robust LiDAR-camera Fusion Framework.

https://arxiv.org/abs/2205.13790

Liao, H., Kong, H., Wang, B., Wang, C., Ye, W., He, Z., et al., 2025. CoT-drive: Efficient Motion Forecasting for Autonomous Driving with LLMs and Chain-of-thought Prompting. https://ieeexplore.ieee.org/document/10980428

Liu, H., Li, C., Wu, Q., Lee, Y.J., 2023. Visual instruction tuning. *Adv Neural Inf Process Syst*, **36**, 34892–34916.

Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B., 2021a. Multimodal motion prediction with stacked transformers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7573–7582.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al., 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 9992–10002.

Luo, X., Ding, F., Yang, F., Zhou, Y., Loo, J., Tew, H. H., et al., 2025. SenseRAG: Constructing environmental knowledge bases with proactive querying for LLM-based autonomous driving. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 899–906.

Ma, J., Chen, X., Huang, J., Xu, J., Luo, Z., Xu, J., et al., 2024a. Cam4DOcc: Benchmark for camera-only 4D occupancy forecasting in autonomous driving applications. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 21486–21495.

Ma, Y., Abdelraouf, A., Gupta, R., Moradipari, A., Wang, Z., Han, K., 2025. Video token sparsification for efficient multimodal LLMs in driving visual question answering. In: 2025 IEEE Intelligent Vehicles Symposium (IV), 2235–2242.

Ma, Y., Cao, Y., Sun, J., Pavone, M., Xiao, C., 2024b. Dolphins: Multimodal language model for driving. In: Computer Vision – ECCV, 403–420.

Ma, Y., Cui, C., Cao, X., Ye, W., Liu, P., Lu, J., et al., 2024c. LaMPilot: An open benchmark dataset for autonomous driving with language model programs. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15141–15151.

Malkov, Y. A., Yashunin, D. A., 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans Pattern Anal Mach Intell*, **42**, 824–836.

Mao, J., Qian, Y., Ye, J., Zhao, H., Wang, Y., 2023. Gpt-driver: Learning to Drive with Gpt. https://arxiv.org/abs/2310.01415

Ngiam, J., Vasudevan, V., Caine, B., Zhang, Z., Chiang, H.T.L., Ling, J., et al., 2022. Scene Transformer: A Unified Architecture for Predicting Future Trajectories of Multiple Agents. https://arxiv.org/abs/2106.08417

Pan, C., Yaman, B., Nesti, T., Mallik, A., Allievi, A. G., Velipasalar, S., et al., 2024. VLP: Vision language planning for autonomous driving. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14760–14769.

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., et al., 2023. In-context retrieval-augmented language models. In: Transactions of the Association for Computational Linguistics, 1316–1331.

Shao, H., Hu, Y., Wang, L., Song, G., Waslander, S. L., Liu, Y., et al., 2024. LMDrive: Closed-loop end-to-end driving with large language models. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15120–15130.

Shao, H., Wang, L., Chen, R., Waslander, S. L., Li, H., Liu, Y., 2023a. ReasonNet: End-to-end driving with temporal and global reasoning. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13723–13733.

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., Chen, W., 2023b. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In: Findings of the Association for Computational Linguistics: EMNLP, 9248–9274.

Tian, X., Gu, J., Li, B., Liu, Y., Wang, Y., Zhao, Z., et al., 2024. DriveVLM: The Convergence of Autonomous Driving and Large Vision-language Models. https://arxiv.org/abs/2402.12289

Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H., 2023. Occ3D: A large-scale 3D occupancy prediction benchmark for autonomous driving. *Adv Neural Inf Process Syst*, **36**, 64318–64330.

Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., et al., 2023. Scene as occupancy. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 8372–8381.

Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., et al., 2025a. OmniDrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 22442–22452.

Wang, T., He, C., Li, H., Li, Y., Xu, Y., Wang, Y., Jiao, J., 2025b. Hlcg: A hierarchical lane-changing gaming decision model for heterogeneous traffic flow on two-lane highways. *Transp Res Rec*, **2679**, 449–469.

Wang, T. H., Maalouf, A., Xiao, W., Ban, Y., Amini, A., Rosman, G., et al., 2024. Drive anywhere: Generalizable end-to-end autonomous driving with multi-modal foundation models. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), 6687–6694.

Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., et al., 2023. OpenOccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 17804–17813.

Wang, Y., Liu, Q., Jiang, Z., Wang, T., Jiao, J., Chu, H., et al., 2025c. RAD: Retrieval-augmented decision-making of meta-actions with vision-language models in autonomous driving. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3838–3848.

Wang, Y., Wang, T., 2024. Research on dual-clutch intelligent vehicle infrastructure cooperative control based on system delay prediction of two-lane highway on-ramp merging area. *Automot Innov*, **7**, 588–601.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al., 2022. Chain-of-thought prompting elicits reasoning in large language models. *Adv Neural Inf Process Syst*, **35**, 24824–24837.

Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J., 2023. SurroundOcc: Multi-camera 3D occupancy prediction for autonomous driving. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 21672–21683.

Wen, L., Fu, D., Li, X., Cai, X., Ma, T., Cai, P., et al., 2024. DiLu: A knowledge-driven Approach to Autonomous Driving with Large Language Models. https://arxiv.org/abs/2309.16292

Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M., 2024. PARA-drive: Parallelized architecture for real-time autonomous driving. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15449–15458.

Wu, J., Gao, B., Gao, J., Yu, J., Chu, H., Yu, Q., et al., 2024. Prospective role of foundation models in advancing autonomous vehicles. *Research*, **7**, 399.

Wu, P., Chen, S., Metaxas, D. N., 2020. MotionNet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11382–11392.

Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y., 2022. Trajectory-guided Control Prediction for End-to-end Autonomous Driving: A Simple yet Strong Baseline. https://arxiv.org/abs/2206.08129

Xing, S., Qian, C., Wang, Y., Hua, H., Tian, K., Zhou, Y., et al., 2025. OpenEMMA: Open-source multimodal model for end-to-end autonomous driving. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 911–919.

Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K. K., et al., 2024. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robot Autom Lett*, **9**, 8186–8193.

Yang, K., Guo, Z., Lin, G., Dong, H., Huang, Z., Wu, Y., et al., 2025a. Tra-

jectory-llm: A Language-based Data generator for Trajectory Prediction in Autonomous Driving. https://openreview.net/forum?id=UapxTvxB3N

Yang, Y., Mei, J., Ma, Y., Du, S., Chen, W., Qian, Y., et al., 2025b. Driving in the occupancy world: Vision-centric 4D occupancy forecasting and planning via world models for autonomous driving. *Proc AAAI Conf Artif Intell*, **39**, 9327–9335.

Yu, L., Wang, T., Jiao, J., Shan, F., Chu, H., Gao, B., 2025. BIDA: A bi-level interaction decision-making algorithm for autonomous vehicles in dynamic traffic scenarios. In: 2025 IEEE Intelligent Vehicles Symposium (IV), 1209–1214.

Yuan, J., Sun, S., Omeiza, D., Zhao, B., Newman, P., Kunze, L., et al, 2024. Rag-driver: Generalisable Driving Explanations with Retrieval-augmented In-context Learning in Multimodal Large Language Model. https://doi.org/10.48550/arXiv.2402.10828

Zhang, M., Fang, Z., Wang, T., Lu, S., Wang, X., Shi, T., 2025. CCMA: A framework for cascading cooperative multi-agent in autonomous driving merging using Large Language Models. *Expert Syst Appl*, **282**, 127717.

Zheng, J., Liang, M., Yu, Y., Li, Y., Xue, Z., 2024. Knowledge graph enhanced multimodal transformer for image-text retrieval. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE), 70–82.

Zhu, Y., Wang, S., Zhong, W., Shen, N., Li, Y., Wang, S., et al., 2025. A Survey on Large Language Model-powered Autonomous Driving. Engineering. https://doi.org/10.1016/j.eng.2025.07.038

Yujin Wang is currently a Ph.D. student at the College of Automotive and Energy Engineering, Tongji University, China. He received the B.E. degree in vehicle engineering from Tongji University, China, in 2023. His research interests include vision-language models, deep learning, and foundation intelligence in autonomous driving.

Tianyi Wang is currently a Ph.D. student at the Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, USA. He received the M.S. degree in mechanical engineering and materials science from Yale University, USA, in 2025. He earned the B.E. degree in vehicle engineering from Tongji University, China, in 2024. His research interests include intelligent transportation systems and connected and automated vehicles, particularly in decision-making, trajectory planning and cooperative control.

Quanfeng Liu is currently a M.S. student at the College of Automotive and Energy Engineering, Tongji University, China. He received the B.E. degree in vehicle engineering from Tongji University, China, in 2025. His research interests include vision-language models, deep learning, and foundation intelligence in autonomous driving.

Wenxian Fan is currently an undergraduate student at the College of Automotive and Energy Engineering, Tongji University, China. Her research interests include vision-language models, deep learning, and foundation intelligence in autonomous driving.

Junfeng Jiao received the Ph.D. degree in urban planning from the University of Washington in 2010. He is currently an Associate Professor in the Community and Regional Planning Program at The University of Texas at Austin. He is the Founding Director of Urban Information Lab, Director of Texas Smart Cities, Director of UT Ethical AI program, and a Founding Member of UT Austin's Good Systems Grand Challenge. His research focuses on smart cities, urban informatics, and ethical/generative AI.

Christian Claudel received the Ph.D. degree in electrical engineering from the University of California Berkeley in 2010. He is currently an Associate Professor in the Department of Civil, Architectural, and Environmental Engineering at The University of Texas at Austin. His research interests include control and estimation of distributed parameter systems, cyberphysical systems monitoring, and the use of wireless sensor networks for environmental applications.

Yunbing Yan received the B.S. degree in vehicle engineering from Chongqing University in 1990, and the M.S. and Ph.D. degrees in vehicle engineering from the Wuhan University of Technology in 1995 and 2008, respectively, where he was an Assistant Professor with the School of Machinery and Automation from 1998 to 2008. Since 2008, he has been a Professor with the School of Vehicle and Traffic Engineering. He is the author of three books, more than 100 articles, and more than 30 inventions. His main research interests include vehicle dynamics and its control, electric vehicle energy management, and intelligent vehicle technology.

Bingzhao Gao received the B.E. and M.E. degrees in vehicle engineering from Jilin University, China, in 1998 and 2002, respectively, and the Ph.D. degree in mechanical engineering from Yokohama National University, Japan, and in control engineering from Jilin University, in 2009. He is currently a Professor with the College of Automotive and Energy Engineering, Tongji University, China. His research interests include vehicle powertrain control and vehicle stability control.

Jianqiang Wang received the B.E. and M.E. degrees from Jilin University of Technology, China, in 1994 and 1997, respectively, and the Ph.D. degree from Jilin University, in 2002. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University, China. He has authored more than 150 papers and is a coinventor of more than 140 patent applications. He was involved in more than ten sponsored projects. His research interests include intelligent vehicles, driving assistance systems, and driver behavior. He was a recipient of the Best Paper Award in the 2014 IEEE Intelligent Vehicle Symposium, the Best Paper Award in the 14th ITS Asia–Pacific Forum, the Best Paper Award in the 2017 IEEE Intelligent Vehicle Symposium, the Changjiang Scholar Program Professor in 2017, the Distinguished Young Scientists of NSF China in 2016, and the New Century Excellent Talents in 2008.

Hong Chen (Fellow, IEEE) received the B.E. and M.E. degrees in process control from Zhejiang University, China, in 1983 and 1986, respectively, and the Ph.D. degree in system dynamics and control engineering from the University of Stuttgart, Germany, in 1997. In 1986, she joined the Jilin University of Technology, China. From 1993 to 1997, she was a Wissenschaftlicher Mitarbeiter with the Institut fuer Systemdynamik und Regelungstechnik, University of Stuttgart. Since 1999, she has been a Professor with Jilin University, and hereafter a Tang Aoqing Professor. From 2015 to 2019, she served as the Director of the State Key Laboratory of Automotive Simulation and Control, Jilin University. She currently joins Tongji University, China, as a Distinguished Professor. Her current research interests include model predictive control, nonlinear control, and applications in mechatronic systems focusing on automotive systems.