An Integrated Perception-to-Planning Framework for Autonomous Driving Using YOLOv11 and Frenet Coordinate Waypoint Generation

Nguyen-Vo Thanh*, Giang-Nguyen Trinh Tra*, Khoi-Le Lu Anh, Duc-Bui Ha
Dept. of Mechanical Engineering
HCMC University of Technology and Education
Ho Chi Minh City, Vietnam
ducbh@hcmute.edu.vn

***Abstract*—Autonomous driving systems traditionally decompose the task into separate perception and planning modules, leading to redundant feature extraction. In this work, we propose a dual-branch multi-task architecture built on YOLO11. After a shared stem, the network splits into independent pathways: a Driving Branch fusing navigation commands via cross-attention Command State Modulation (CSM) to produce a semantic heatmap and Bézier waypoints; and a Perception Branch for anchor-free object detection. Branch separation eliminates feature-level gradient conflicts. We introduce the Frenet Decomposition Anisotropic Temporal (FDAT) loss, which decomposes trajectory error into lateral (cross-track) and longitudinal (along-track) components in the Frenet frame, penalizing lane departure more heavily than timing offset for safety-critical lane keeping. Uncertainty weighting and Perception-Gated Planning (PGP) stabilize joint training. The predicted waypoints are passed to a classical vehicle controller for direct steering actuation. Evaluated on a 1:10 scale vehicle (NVIDIA Jetson Orin), the system successfully navigates complex scenarios (intersections, traffic lights, pedestrians) with low computational overhead.**

***Index Terms*—Autonomous driving, dual-branch architecture, YOLOv11, object detection, Frenet coordinates, waypoint prediction, multi-task learning, embedded systems.**

I. Introduction

In recent years, the field of autonomous driving has seen rapid development, driven largely by advances in deep learning, embedded computing, and sensor technologies. Most autonomous driving systems use modular pipelines that split perception, prediction, and planning into separately optimized networks, increasing engineering complexity and data-labeling cost. A detection model may focus exclusively on bounding box coordinates, while a separate planning model must independently learn to interpret spatial geometry to forecast movement [1]–[4]. This disjointed optimization leads to inefficient feature utilization and increased computational burden, as multiple networks redundantly extract similar representations from identical input frames. Furthermore, inaccuracies in upstream perception modules propagate and degrade downstream prediction and planning performance [5].

It is therefore imperative to explore unified frameworks that integrate the aforementioned modules, reducing redundancy, improving efficiency, and enabling tighter coupling between scene understanding and motion generation [6]. However, naive feature sharing between detection and planning creates a fundamental tension: detection requires spatially precise features for object localization, while planning requires holistic features sensitive to navigational context. When both tasks share a single feature pathway, competing gradient signals often cause one task to suppress the other [7]. Existing solutions fail to satisfy real-time and memory constraints of resource-limited embedded platforms [8].

In this work, we propose a dual-branch architecture built on the native YOLO11 convolutional backbone, augmented with a **Command State Modulation (CSM)** mechanism that fuses navigation intent into the driving features via cross-attention with learned residual scaling and gating, allowing the vehicle to transition from straight driving to a commanded turn at an intersection. After a lightweight shared stem, the network splits into two fully independent branches: a **Driving Branch** for heatmap prediction, Frenet trajectory generation, and command classification, and a **Perception Branch** dedicated to anchor-free object detection. Each branch develops specialized feature representations without task-level gradient interference. A novel *Frenet Decomposition Anisotropic Temporal* (FDAT) loss provides physically motivated trajectory supervision, with uncertainty-based task weighting [7] and a Perception-Gated Planning (PGP) mechanism stabilizing joint training through the shared stem.

The main contributions of this work are:

- A **dual-branch YOLO11 architecture** with a shared convolutional stem that splits into an independent Driving Branch (heatmap, trajectory, classification) and Perception Branch (detection), eliminating feature-level gradient conflict between the two objectives.
- A **Command State Modulation** (CSM) module that uses cross-attention to modulate the deep driving feature state according to the discrete navigation command (left, right, straight, follow), enabling all driving heads to be conditioned on navigational intent.
- The **FDAT loss**, which decomposes trajectory error into lateral and longitudinal components in the Frenet frame,

*Equal contribution.

979-8-3195-0496-8/26/$31.00 ©2026 IEEE

applies direction-aware penalization conditioned on a learned CommandGate, and uses temporal weighting that emphasizes the first and last waypoints.

- **Uncertainty-based multi-task weighting** with a **Perception-Gated Planning** (PGP) gate to stabilize joint training of detection and planning objectives.

II. Proposed Method

A. System Overview

The proposed network processes two inputs: a monocular RGB image (320×320 pixels) and a discrete navigation command $c \in \{\text{left, right, straight, follow}\}$. A shared YOLO11 convolutional stem extracts low-level features up to stride 4. The stem output branches into two fully independent pathways. The *Driving Branch* applies its own deep YOLO11 encoder, fuses the navigation command via cross-attention CSM, refines features through a PAN (Path Aggregation Network) neck, and outputs a semantic heatmap, n_w=10 Frenet waypoints, and a command prediction. The *Perception Branch* applies its own separate YOLO11 encoder and PAN neck to predict n_c=14-class bounding boxes. Predicted waypoints are consumed by a classical vehicle controller for real-time actuation. The full architecture is shown in Fig. 1.

B. Shared YOLO11 Stem

The shared stem uses the three lightest YOLO11 layers: a strided `Conv`(64, 3×3, stride 2) that extracts low-level edge and texture features (P_1), a strided `Conv`(128, 3×3, stride 2) that produces stride-4 features (P_2), and a lightweight `C3k2`(256, `shortcut=False`, r=0.25) block that enriches P_2 with short-range context. The P_2 output is the branching point, routed simultaneously to both branch encoders and retained as a high-resolution skip connection for the Heatmap-Head.

C. Driving Branch

Driving Encoder: Starting from P_2, three YOLO11 Conv + C3k2 stage pairs progressively downsample the feature map: `Conv`(256, stride 2) + `C3k2`(512) at stride 8 (P_3^D), `Conv`(512, stride 2) + `C3k2`(512, `shortcut=True`) at stride 16 (P_4^D), and `Conv`(1024, stride 2) + `C3k2`(1024, `shortcut=True`) at stride 32 (P_5^D). P_5^D is further processed by `SPPF`(1024, k=5) and `C2PSA`(1024) to aggregate multi-scale context.

Command State Modulation: A *CommandEncoder* projects the discrete command c into a 256-d embedding $\mathbf{z}_c$. P_5^D is compressed to 256 channels, flattened to spatial tokens $\mathbf{X}$, and fused with $\mathbf{z}_c$ via multi-head cross-attention (visual tokens as queries, command as key-value), scaled by a learned gate g and residual gain r, followed by LayerNorm. Naively concatenating the command to the feature vector allows the high-dimensional visual context to overpower the command signal; cross-attention forces each spatial token to explicitly attend to $\mathbf{z}_c$, ensuring navigational intent actively gates the driving representation rather than being diluted. A learned *CommandGate* $g \in [0,1]$ switches between lane-keeping (g≈0) and intersection (g≈1) mode, reducing failures caused by commands applied too early or too late.

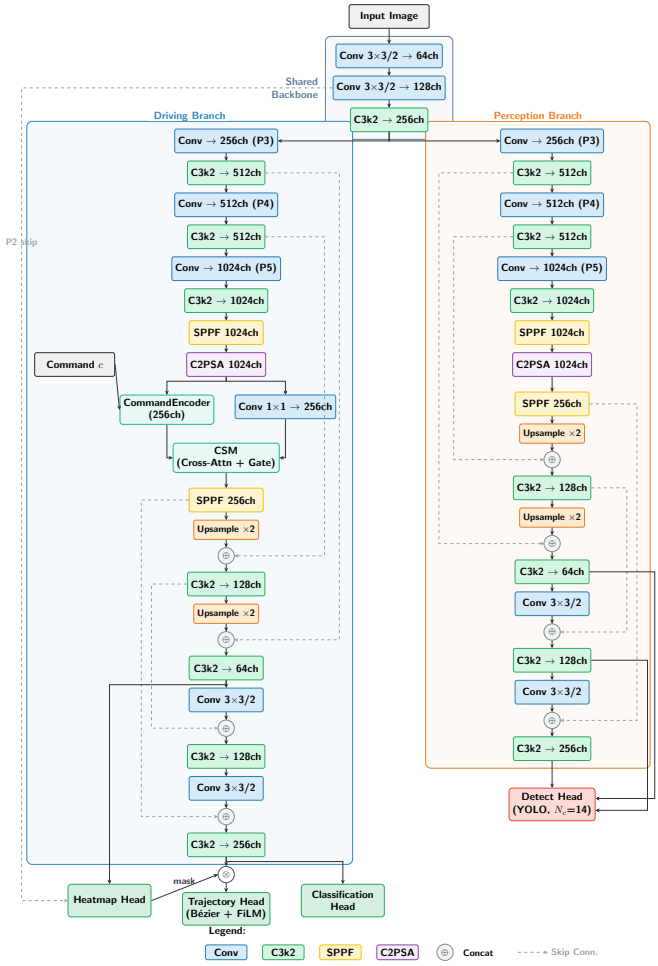


Fig. 1. Overview of the proposed dual-branch architecture with a shared low-level backbone and two independent streams: a **Driving branch** for CSM-based trajectory planning, heatmap prediction, and command classification, and a **Perception branch** for multi-scale object detection. Dashed lines denote skip connections.

Path Aggregation Network (PAN) Neck (Driving): The fused feature enters a top-down and bottom-up PAN, producing three command-conditioned neck outputs $P_3^{D\prime}$ (stride 8), $P_4^{D\prime\prime}$ (stride 16), $P_5^{D\prime\prime}$ (stride 32).

Heatmap Head: This module fuses $P_3^{D\prime}$ with the high-resolution shared P_2 via an Attention Gate to produce a trajectory heatmap $\mathcal{H}$, supervised with MSE + Dice Loss against on-the-fly Gaussian polylines. The heatmap serves a dual purpose: it provides an explicit 2D spatial path prior that the Trajectory Head cannot recover from its 4×4 pooled representation alone, and its sharper Dice-driven boundaries yield a cleaner gating signal via $1+\sigma(\mathcal{H})$ (Fig. 2).

Trajectory Head: The head receives $P_5^{D\prime\prime}$ and the heatmap $\mathcal{H}$ as inputs. First, $\mathcal{H}$ is resized to match $P_5^{D\prime\prime}$ and applied as a multiplicative spatial attention mask:

$$\mathbf{F} = P_5^{D\prime\prime} \odot \big(1+\sigma(\mathcal{H})\big), \tag{1}$$

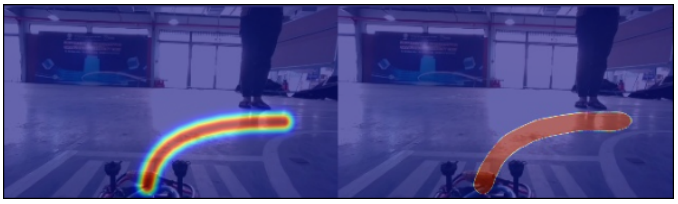


Fig. 2. Ground-truth heatmap (left) vs. predicted output (right). Sharper boundaries arise from Dice Loss and serve as a stronger attention prior for trajectory planning.

where $\sigma(\cdot)$ is the sigmoid function. $\mathbf{F}$ is then spatially downsampled to 4×4 via bilinear interpolation and flattened to a vector. The navigation command c is encoded by a learned $\texttt{Embedding}(n_m{=}4,\ 64)$ layer into a 64-dimensional vector $\mathbf{z}_c$, $\mathbf{h}_0 = [\mathbf{F}_{\text{flat}};\ \mathbf{z}_c]$. A two-layer MLP with BatchNorm and ReLU maps $\mathbf{h}_0$ to a 512-dimensional state $\mathbf{h}$, which is then modulated by a *Feature-wise Linear Modulation* [10] (FiLM) layer. A linear output layer regresses $4\times2{=}8$ values that are reshaped and passed through $\tanh$ to produce four Bézier control points $\{\mathbf{c}_j\}_{j=0}^{3} \subset [-1,1]^2$. The predicted trajectory is obtained by sampling $n_w{=}10$ points from the cubic Bézier curve:

$$\hat{\mathbf{p}}_i = \sum_{j=0}^{3} \underbrace{\binom{3}{j} t_i^j (1{-}t_i)^{3-j}}_{B_{j,3}(t_i)} \mathbf{c}_j, \quad t_i = \frac{i}{n_w{-}1}, \tag{2}$$

where the Bernstein matrix $\mathbf{M} \in \mathbb{R}^{n_w\times4}$ is precomputed and fixed. The final waypoints are obtained as $\hat{\mathbf{P}} = \mathbf{M}\,\mathbf{C}$, where $\mathbf{C} \in \mathbb{R}^{4\times2}$ is the control-point matrix. This parameterization guarantees C^1 path continuity and reduces the effective output space from $2n_w{=}20$ to $2\times4{=}8$ values.

Classification Head: A lightweight MLP operating on $P_5^{D\prime\prime}$ predicts the navigation command ($n_m{=}4$ classes) as an auxiliary signal, enforcing that the driving representation retains discriminative command-level information.

D. Perception Branch

The Perception Branch is fully independent from the Driving Branch, sharing only the stem output P_2. It applies its own three Conv + C3k2 encoder stages (P_3^P, P_4^P, P_5^P at strides 8, 16, 32), followed by $\texttt{SPPF}(1024, k{=}5)$ and $\texttt{C2PSA}(1024)$, and a dedicated PAN neck producing $P_3^{P\prime}$, $P_4^{P\prime\prime}$, $P_5^{P\prime\prime}$. An anchor-free YOLO11-style *Detect* head [9] operating on all three scales predicts bounding boxes, class probabilities ($n_c{=}14$), and confidence scores. Box regression uses Distribution Focal Loss (DFL) and CIoU loss. Since no driving-related gradients flow through the Perception Branch, its encoder and neck can specialize fully for detection accuracy.

E. FDAT Loss

Standard waypoint regression in Cartesian space penalizes x- and y-axis errors symmetrically, yet their driving consequences differ: lateral (y) deviation causes lane departure, while longitudinal (x) error only affects timing. The Frenet frame of the ground-truth curve explicitly exposes this asymmetry, enabling direction-aware penalization. We therefore propose the *Frenet Decomposition Anisotropic Temporal* (FDAT) loss, which decomposes trajectory error into cross-track and along-track components and applies temporally weighted, anisotropic penalties.

1) Frenet Decomposition: At each step i, the unit tangent $\mathbf{T}_i$ is estimated from the ground-truth curve via central finite differences (one-sided at boundaries), with unit normal $\mathbf{N}_i = [-T_{i,y},\ T_{i,x}]^\top$. The prediction error $\mathbf{e}_i = \hat{\mathbf{p}}_i - \mathbf{p}_i^*$ decomposes as:

$$e_i^s = \mathbf{e}_i \cdot \mathbf{T}_i \quad \text{(along-track)}, \tag{3}$$

$$e_i^d = \mathbf{e}_i \cdot \mathbf{N}_i \quad \text{(cross-track)}. \tag{4}$$

e_i^d measures lateral miss-distance (lane departure); e_i^s measures longitudinal timing error.

2) Dual-Mode Anisotropic Penalty: A learned *CommandGate* (GAP $\rightarrow$ two-layer MLP $\rightarrow$ sigmoid) produces $g \in [0,1]$ selecting lane-following ($g{\approx}0$) or intersection ($g{\approx}1$) mode. Each mode applies Smooth ℓ_1 ($\beta{=}0.1$) as $\rho(\cdot)$ with direction-dependent coefficients:

$$\mathcal{L}_i^{\text{lane}} = \tfrac{\alpha_\ell}{\bar{\alpha}_\ell}\,\rho(e_i^d) + \tfrac{\beta_\ell}{\bar{\alpha}_\ell}\,\rho(e_i^s), \tag{5}$$

$$\mathcal{L}_i^{\text{inter}} = \tfrac{\alpha_\iota}{\bar{\alpha}_\iota}\,\rho(e_i^d) + \tfrac{\beta_\iota}{\bar{\alpha}_\iota}\,\rho(e_i^s), \tag{6}$$

where $\bar{\alpha}_\ell{=}(\alpha_\ell{+}\beta_\ell)/2$, $\bar{\alpha}_\iota{=}(\alpha_\iota{+}\beta_\iota)/2$ keep loss magnitude comparable to standard 2D Smooth ℓ_1. Defaults: $\alpha_\ell{=}5, \beta_\ell{=}2, \alpha_\iota{=}5, \beta_\iota{=}3$; the ratio $\alpha_\ell/\beta_\ell{=}2.5$ means lateral errors receive $2.5\times$ the penalty of longitudinal ones, reflecting the higher criticality of lane departure. In intersection mode, an endpoint anchor $\lambda_{\text{ep}}\|\hat{\mathbf{p}}_{n_w} - \mathbf{p}^*_{n_w}\|_2^2$ ($\lambda_{\text{ep}}{=}1$) is added. The gate blends modes:

$$\mathcal{L}_{\text{pos},i} = (1{-}g)\,\mathcal{L}_i^{\text{lane}} + g\,\mathcal{L}_i^{\text{inter}}. \tag{7}$$

3) Temporal Weighting and Auxiliary Terms: Temporal weights follow a double-exponential bathtub curve (higher weight at the first and last waypoints, lower in the middle) $\tilde{w}_i = e^{-i/\tau_s} + e^{-(n_w-1-i)/\tau_e} + 0.5$, normalized as $w_i = \tilde{w}_i/\bar{\tilde{w}}$ so that magnitude is invariant to n_w ($\tau_s{=}\tau_e{=}2$). Two auxiliary terms complete the loss: a cosine-similarity *heading loss* $\mathcal{L}_{\text{head}} = \frac{1}{n_w-1}\sum_i(1 - \cos(\Delta\hat{\mathbf{p}}_i, \Delta\mathbf{p}_i^*))$ where $\Delta\hat{\mathbf{p}}_i{=}\hat{\mathbf{p}}_{i+1}{-}\hat{\mathbf{p}}_i$, and a second-order finite-difference *smoothness loss* $\mathcal{L}_{\text{smooth}} = \frac{1}{(n_w-2)\cdot2}\sum_{i=1}^{n_w-2}\|(\hat{\mathbf{p}}_{i+1}{-}\hat{\mathbf{p}}_i){-}(\hat{\mathbf{p}}_i{-}\hat{\mathbf{p}}_{i-1})\|^2$. The total FDAT loss per sample is:

$$\mathcal{L}_{\text{FDAT}} = \frac{1}{n_w}\sum_i w_i\,\mathcal{L}_{\text{pos},i} + \lambda_h\,\mathcal{L}_{\text{head}} + \lambda_s\,\mathcal{L}_{\text{smooth}}, \tag{8}$$

with $\lambda_h{=}0.2$ and $\lambda_s{=}0.1$.

F. Multi-Task Loss Balancing

Although branch separation eliminates feature-level gradient conflicts after P_2, the shared YOLO11 stem still receives gradient signals from all four task heads. We stabilize this with two complementary mechanisms applied to the total training loss $\mathcal{L}_{\text{total}} = \sum_k \mathcal{W}_k$.

1) Uncertainty-Based Task Weighting: Following Kendall et al. [7], each task loss $\mathcal{L}_k$ is weighted by a learned scalar $s_k = \log \sigma_k^2$, interpreted as the homoscedastic log-variance of task k. The per-task contribution to the total loss is:

$$\mathcal{W}_k = e^{-s_k} \lambda_k \mathcal{L}_k + s_k, \tag{9}$$

where λ_k is a fixed task-specific scale and s_k is clamped to $[-5, 5]$ for numerical stability, ensuring that tasks with inherently small magnitudes (such as normalized trajectory errors) are not starved of gradients during shared representation learning. Tasks with inherently high variance (large s_k) are automatically down-weighted, allowing the network to self-calibrate task importance without manual retuning. Four s_k parameters are learned: one each for the heatmap, trajectory, detection, and classification losses.

2) Perception-Gated Planning: To prevent corrupted gradients from reaching the driving heads when detection is poorly initialized, a stop-gradient soft gate $g_{\text{pgp}} \in [0.7, 1]$ scales $\mathcal{L}_{\text{heatmap}}$ and $\mathcal{L}_{\text{FDAT}}$. The gate decays exponentially when $\mathcal{L}_{\text{det}}$ exceeds a threshold of 8, allowing the backbone to stabilize perceptual features first; it recovers to 1 as detection converges, restoring full gradient flow to the shared stem.

III. Experimental Setup

A. Hardware Platform

The system is deployed on a 1:10 scale autonomous vehicle equipped with a front-facing monocular camera and an NVIDIA Jetson Orin as the onboard compute unit. The fully convolutional design exports cleanly to ONNX and TensorRT, achieving sub-10 ms end-to-end inference latency at the full 320×320 input resolution. A classical vehicle controller converts the n_w=10 predicted waypoints into steering angle commands, decoupling the learned component from low-level actuation.

B. Dataset Collection and Annotation

The dataset consists of approximately 5,000 RGB image frames collected in a controlled 10×10 m laboratory environment encompassing straight roads, curved segments, intersections, and shared zones with pedestrians and functional traffic lights. Trajectory labels are manually annotated in the 2D image plane as cubic Bézier control points (Fig. 3), normalized to $[-1, 1]$ relative to the image frame, requiring no 3D lifting. Bounding boxes and behavioral states are labeled simultaneously through the same interface. We acknowledge that this single controlled environment limits out-of-distribution generalization; large-scale outdoor validation is left for future work.

The complete dataset was randomly partitioned into an 85% training set and a 15% validation set.

C. Training Details

All models are trained for 60 epochs using AdamW with momentum 0.937 and weight decay 5×10^{-4}. The initial learning rate is 10^{-3}, decayed via cosine annealing. Input images are resized to 320×320 with HSV jitter, color jitter, and Gaussian noise augmentations. Training is conducted on a single GPU with batch size 32 and mixed-precision (FP16).

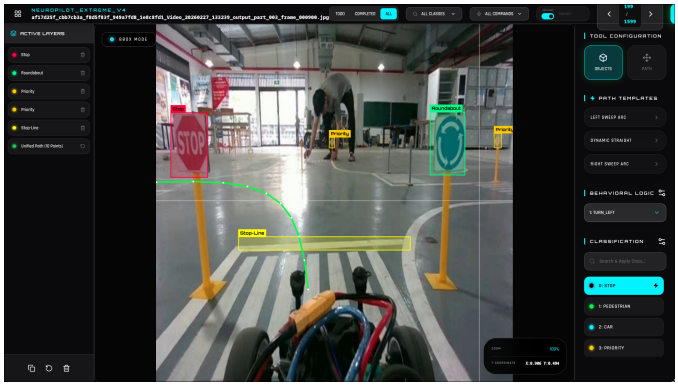


Fig. 3. The integrated labeling panel used for dataset generation. The interface facilitates the simultaneous annotation of perceptual bounding boxes, behavioral state classification, and the manual placement of cubic Bézier control points (visible as the green curve) to define smooth, expert-level ground truth waypoints.



Fig. 4. The 10×10 m laboratory map used for closed-loop evaluation, featuring high-curvature turns and standardized traffic infrastructure.

D. Methodology and Evaluation Scenarios

The system is evaluated through two distinct modalities designed to test visual generalization and real-time control stability.

First, we evaluate offline on an **unseen video** from a novel track to test generalization. Second, we conduct closed-loop testing on a physical 10×10 **m map** (Fig. 4).

1) Quantitative Results: We report four validation metrics. *ADE* and *FDE* are the mean and final ℓ_2 trajectory errors. *CTE* and *ATE* are the mean absolute cross-track and along-track errors: CTE=mean($|e_i^d|$), ATE=mean($|e_i^s|$) (Eq. 4–3). Table I summarizes the results.

Smooth ℓ_1+FDAT achieves the lowest trajectory errors in Table I, reducing ADE, FDE, CTE, and ATE by 11.1%, 17.3%, 11.4%, and 10.1%, respectively, relative to baseline Smooth ℓ_1. This confirms that the α_ℓ/β_ℓ=2.5 anisotropy targets lane departure while preserving along-track gradient flow through the $(\alpha+\beta)/2$ normalization. In contrast, Wing+FDAT degrades because Wing loss retains a non-vanishing near-zero gradient that is amplified by FDAT's lateral weighting, destabilizing training.

TABLE I
TRAJECTORY PREDICTION BENCHMARK. CTE AND ATE MEASURE LATERAL AND LONGITUDINAL DEVIATIONS IN THE FRENET FRAME, RESPECTIVELY.

Loss	FDAT	ADE (↓)	FDE (↓)	CTE (↓)	ATE (↓)
MSE	No	0.0882	0.1035	0.0344	0.0706
Smooth ℓ_1	No	0.0865	0.1085	0.0316	0.0704
Wing	No	0.0874	0.1116	0.0322	0.0698
Wing	Yes	0.1383	0.1791	0.0674	0.0970
Smooth ℓ_1	Yes	**0.0769**	**0.0897**	**0.0280**	**0.0633**

TABLE II
DETECTION BENCHMARK ON THE VALIDATION SET. F1 @CONF AND P=1.0 @CONF DENOTE THE CONFIDENCE THRESHOLDS FOR BEST F1 AND PERFECT PRECISION, RESPECTIVELY.

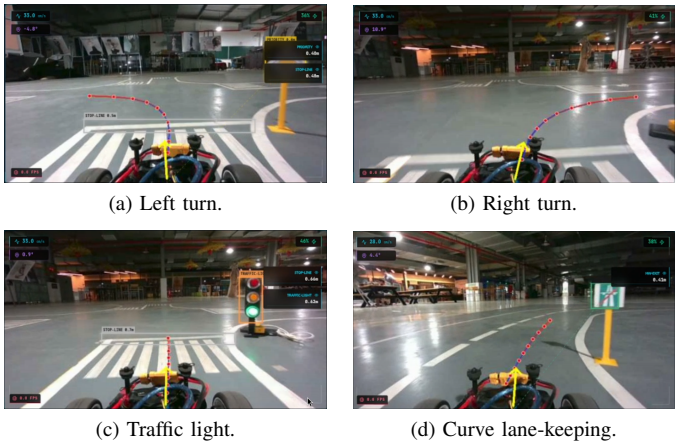
Loss	FDAT	mAP@0.5 (↑)	Best F1 (↑)	F1 @conf	P=1.0 @conf	Max Recall (↑)
MSE	No	0.860	0.83	0.435	0.982	0.89
Smooth ℓ_1	No	0.865	**0.86**	0.365	0.972	**0.92**
Wing	No	**0.875**	**0.86**	0.495	0.973	**0.92**
Wing	Yes	0.853	0.83	0.409	**0.988**	0.87
Smooth ℓ_1	Yes	0.858	0.84	0.360	0.950	0.90

Table II shows that detection accuracy remains competitive across loss variants despite the trajectory-focused FDAT modification. The best mAP@0.5 is obtained by Wing without FDAT, while Smooth ℓ_1 without FDAT ties its best F1 and max recall. The FDAT variants incur only modest detection changes, indicating that branch separation limits negative transfer from the driving loss to the perception head.

2) Qualitative Results: Figure 5 presents the system's performance on an **unseen test video**. The Heads-Up Displays overlays predicted waypoints (red) and steering vector (yellow). These smooth, C^1-continuous trajectories validate the CSM module's intent interpretation and FDAT's precise lane adherence. In closed-loop evaluation, the system successfully completed 9 of 10 independent intersection runs (90%). A run was scored as successful if the vehicle completed the commanded maneuver without collision or track departure. The single failure occurred under extreme glare that saturated the camera input. The vehicle also responded correctly to stop signs and traffic lights throughout all runs.

IV. CONCLUSION

We presented a dual-branch architecture with Command State Modulation (CSM) and Frenet Decomposition Anisotropic Temporal (FDAT) loss for end-to-end autonomous driving. By decoupling perception and trajectory planning, the system eliminates gradient conflicts while maintaining high efficiency. The CSM module ensures robust directional intent fusion, while the FDAT loss rigorously penalizes lateral deviations for stable lane adherence. Real-world experiments on a 1:10 scale platform demonstrate successful navigation of complex scenarios—including intersections, traffic lights, and sharp curves—at sub-10 ms latency on a Jetson Orin. Future work will explore multi-camera inputs and temporal modeling.



(a) Left turn. (b) Right turn. (c) Traffic light. (d) Curve lane-keeping.

Fig. 5. Qualitative results on unseen video: (a)–(b) intersection traversal, (c) traffic light response, (d) curve lane-keeping.

REFERENCES

[1] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[2] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

[3] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "FIERY: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 15273–15282.

[4] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

[5] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024, doi: 10.1109/TPAMI.2024.3435937.

[6] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, arXiv:1604.07316. [Online]. Available: https://arxiv.org/abs/1604.07316

[7] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 7482–7491.

[8] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "TransFuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[9] G. Jocher and A. Chaurasia, "Ultralytics YOLO11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3942–3951.

[11] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2235–2245.