CONDENSEDRIVE: KNOWLEDGE-DENSE, TOKEN-SPARSE VISUAL ENCODER FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end driving requires rich spatiotemporal representations to support scene understanding and trajectory planning under strict latency constraints. However, most systems initialize their visual encoders from generic image pretraining and densely process all patch tokens for each camera view and input frame, limiting both representation quality and inference efficiency. To address these challenges, we present ConDenseDrive, a knowledge-dense and token-sparse visual encoder for end-to-end autonomous driving. Our three-stage framework first makes the encoder knowledge-dense by distilling complementary visual, semantic, and geometric knowledge from three vision foundation models. It then adapts the distilled encoder to multi-frame end-to-end driving and integrates spatiotemporal context into register tokens for planning. Finally, it makes the encoder token-sparse by applying register-guided token compression, reducing redundant patch computation and yielding a $2.0\times$ speedup while improving planning performance. ConDenseDrive achieves 94.5 PDMS on NAVSIM v1 and 90.8 EPDMS on NAVSIM v2, and reaches 37.4 HD-Score in zero-shot evaluation on HUGSIM. Extensive experiments and qualitative analyses demonstrate that visual encoder design provides an effective route to reliable and efficient end-to-end driving.

1 INTRODUCTION

End-to-end autonomous driving predicts future ego trajectories from sensor observations and navigation conditions (Chitta et al., 2022; Hu et al., 2023). Recent end-to-end autonomous driving systems have advanced structured scene representations (Hu et al., 2023; Jiang et al., 2023), trajectory generation (Chen et al., 2024; Liao et al., 2025), and trajectory scoring and selection (Yao et al., 2026; Li et al., 2025b; Gao et al., 2026b), as illustrated in Fig. 1. These methods rely on visual encoders, which supply scene information to the planner and can contribute substantially to inference latency (Fig. 3(b)). Despite this central role, visual encoders are typically initialized from generic image pretraining. Their representation quality and computational cost are rarely considered jointly.

A straightforward solution is to scale the visual backbone to accommodate richer representations, but the added computation increases visual encoding latency. Recent research has developed driving-oriented pretraining methods for visual encoders (Wu et al., 2023; Yang et al., 2024; Min et al., 2024; Wang et al., 2026a; Strong et al., 2026). These results show that driving benefits from representations beyond generic image recognition. However, visual structure, semantic context, and scene geometry are often learned through different objectives. Learning a comprehensive representation of driving scenes without increasing encoder size therefore remains a central challenge.

Beyond stronger features from individual images, driving representations should capture the surrounding scene across cameras and its evolution over time. Recent work has shown that multi-camera observations and historical frames can improve trajectory planning (Hu et al., 2022; 2023; Xie et al., 2025). However, encoding each image separately does not model spatiotemporal relationships. Dense processing of every camera frame also increases inference latency. A further challenge is to adapt a visual encoder to provide spatiotemporal representations for planning while limiting the cost of processing multiple cameras and frames.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

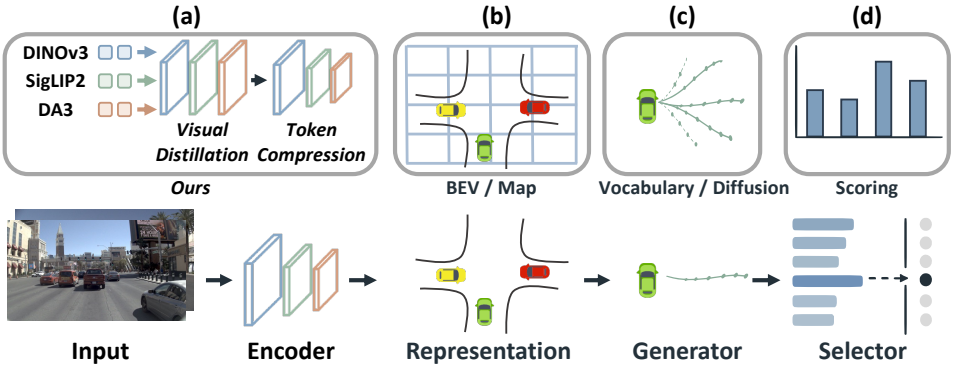


Figure 1: **Design focuses in end-to-end driving.** The bottom row outlines an end-to-end driving pipeline. Representative methods focus on (b) scene representation (Hu et al., 2023; Jiang et al., 2023), (c) trajectory generation (Chen et al., 2024; Liao et al., 2025), and (d) trajectory selection (Yao et al., 2026; Li et al., 2025b). ConDenseDrive instead targets (a) visual encoding through multi-teacher distillation and token compression.

To address these challenges, we propose ConDenseDrive, a three-stage framework that makes a visual encoder knowledge-dense and token-sparse. First, multi-teacher visual distillation transfers complementary visual, semantic, and geometric knowledge into the encoder, improving representation quality without increasing encoder size. Second, multi-frame register adaptation uses the distilled encoder to capture spatiotemporal context in register tokens for planning. Finally, register-guided token compression uses these driving-oriented register tokens to retain informative patch tokens and fuse the remaining content, reducing the visual encoding cost of multiple cameras and frames without an auxiliary routing network.

ConDenseDrive achieves 94.5 PDMS on NAVSIM v1 and 90.8 EPDMS on NAVSIM v2, outperforming strong approaches centered on scene representation and planner design. It further reaches 37.4 HD-Score in zero-shot closed-loop evaluation on HUGSIM. These gains arise from encoder-centered optimization without redesigning the downstream planner. Its knowledge-dense and token-sparse design delivers a $2.0\times$ speedup while improving planning performance. These findings demonstrate that visual encoder design provides an effective route to reliable and efficient end-to-end driving.

Our main contributions are summarized as follows:

- We introduce ConDenseDrive, a three-stage framework that makes the visual encoder knowledge-dense and token-sparse for end-to-end driving.
- We propose register-guided token compression, which uses driving-oriented register tokens to retain informative patch tokens without an auxiliary routing network.
- We achieve state-of-the-art performance on NAVSIM v1 and v2 with a $2.0\times$ end-to-end speedup and demonstrate zero-shot closed-loop generalization on HUGSIM.

2 Related Work

End-to-End Autonomous Driving. Recent progress has largely come from redesigning scene representations and planners. UniAD unifies perception, prediction, and planning through a BEV representation, while VAD represents the scene with vectorized elements (Hu et al., 2023; Jiang et al., 2023). Planning has evolved from trajectory vocabularies in VADv2 to generative distributions in DiffusionDrive and learned proposal selection in DriveSuprim (Chen et al., 2024; Liao et al., 2025; Yao et al., 2026). VLA systems further introduce semantic reasoning into end-to-end driving. ReCogDrive transfers VLM-derived driving knowledge to a diffusion planner, while UniDriveVLA coordinates specialized experts for understanding, perception, and action planning (Xiong et al., 2026; Li et al., 2026). RAD and RAD-2 instead improve driving policies through large-scale closed-

loop reinforcement learning (Gao et al., 2026a;b). DrivoR connects a ViT encoder to lightweight planning modules through camera-specific register tokens, but still processes all patch tokens within each camera image (Kirby et al., 2026). We improve representation quality while reducing patch token computation inside the encoder.

Visual Knowledge Distillation. Knowledge distillation transfers capabilities from large teachers to compact students (Hinton et al., 2015; Touvron et al., 2021). Recent methods extend feature distillation across diverse vision foundation models. AM-RADIO and UNIC consolidate complementary visual teachers within a shared encoder, while DUNE aligns representations from 2D and 3D teachers (Ranzinger et al., 2024; Sarıyıldız et al., 2024; 2025). Theia distills diverse vision foundation models into a compact encoder for robot policy learning (Shang et al., 2024). DriveZero further applies multi-teacher visual distillation to learn a visual backbone for end-to-end driving (He et al., 2026). We jointly address visual knowledge transfer and encoder efficiency for end-to-end driving.

Token Compression. Token compression reduces the cost of Vision Transformers by changing how patch tokens are processed (Dosovitskiy et al., 2021). DynamicViT learns to prune less informative tokens, while A-ViT allocates a different computation depth to each token (Rao et al., 2021; Yin et al., 2022). EViT ranks patch tokens using CLS attention and fuses the low-score tokens into a single representation (Liang et al., 2022). ToMe merges similar tokens, while TPS combines token pruning and squeezing to preserve discarded information (Bolya et al., 2022; Wei et al., 2023). Driving-specific methods operate on different compression targets. SeeLess stochastically masks extracted foundation-model patch features, while COMPACT-VA compresses long temporal context into a bounded working memory (Mallak et al., 2026; Liang et al., 2026). Our method instead uses attention from driving-oriented register tokens to retain informative patch tokens and fuse the remaining low-score tokens into a single token within selected encoder blocks.

3 PRELIMINARIES

Problem Definition. Let $\mathcal{I} = \{I_{t,v}\}$ denote the set of synchronized images from V camera views over T observed frames, where t and v index time and camera view. Let $\mathbf{u}$ denote the auxiliary inputs, including the ego state and navigation information. The predicted trajectory $\boldsymbol{\tau} = \{(x_q, y_q, \psi_q)\}_{q=1}^{Q}$ contains Q future ego poses. We formulate the task as

$$\boldsymbol{\tau} = \mathcal{P}(\mathcal{E}(\mathcal{I}), \mathbf{u}), \tag{1}$$

where $\mathcal{E}$ denotes the visual encoding module and $\mathcal{P}$ denotes the downstream planner. We optimize $\mathcal{E}$ while retaining the interface of $\mathcal{P}$. Since every view and frame contributes a patch token sequence, visual encoding is costly. Our goal is to learn driving-oriented spatiotemporal representations while reducing computation on less relevant patch tokens without degrading trajectory prediction.

4 METHOD

Overview. Fig. 2 presents the three-stage framework of ConDenseDrive. First, multi-teacher visual distillation enriches the representations of a compact encoder without increasing its size. Second, multi-frame register adaptation uses the distilled encoder to aggregate spatial evidence and temporal context in register tokens for planning. Finally, register-guided token compression uses attention from these register tokens to retain informative patch tokens and fuse the rest, reducing visual encoding cost.

4.1 MULTI-TEACHER VISUAL DISTILLATION

End-to-end driving relies on diverse visual cues, including appearance, semantic context, and scene geometry. Vision foundation models trained with distinct objectives encode these cues in complementary ways. To make a compact encoder knowledge-dense without increasing its size, Stage 1 distills their features into a single student encoder.

Teacher Supervision. We use DINOv3 (Siméoni et al., 2025), SigLIP2 (Tschannen et al., 2025), and Depth Anything 3 (DA3) (Lin et al., 2025) as frozen teachers for visual, semantic, and geometric supervision, respectively. We extract final-layer patch features from DINOv3 and SigLIP2, and the final DA3 backbone output before its depth decoder.

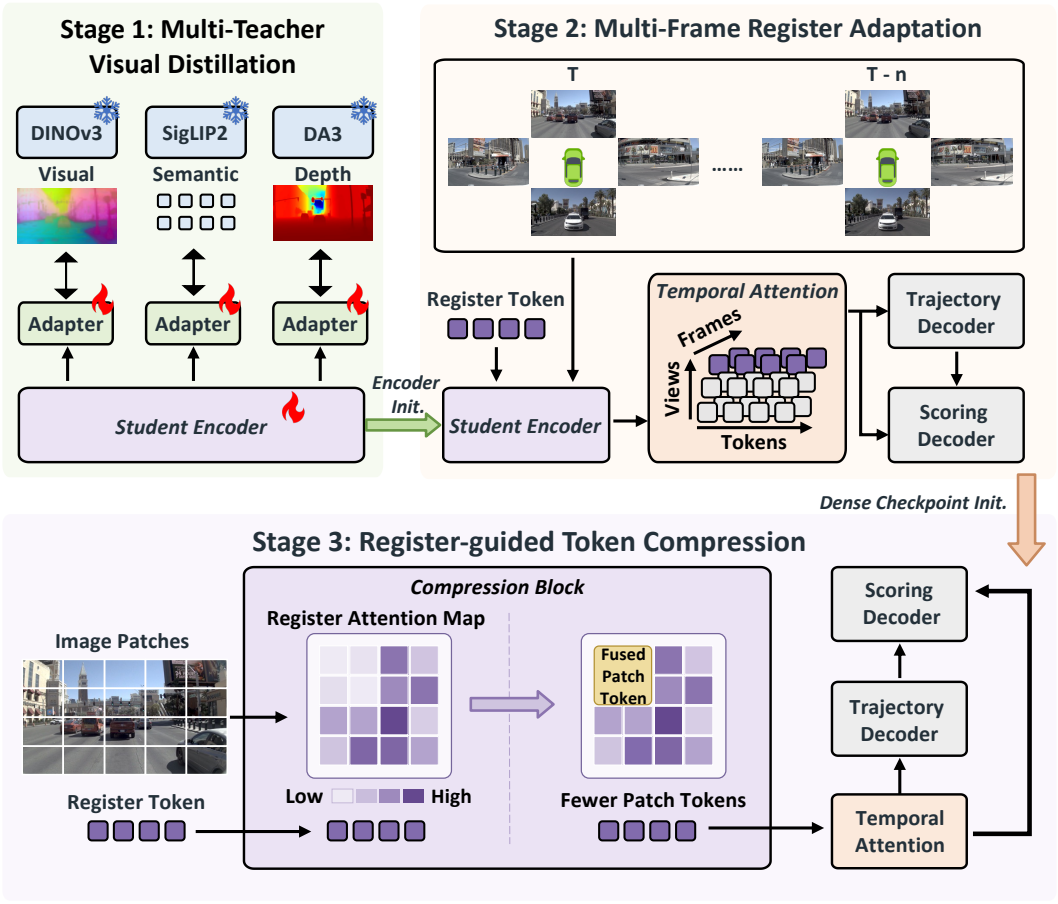


Figure 2: **Three-stage framework of ConDenseDrive.** Multi-teacher visual distillation enriches the student encoder. Multi-frame register adaptation incorporates spatiotemporal context into register tokens for planning. Register-guided token compression then uses the resulting register-to-patch attention to retain informative patch tokens and fuse the rest.

Feature Alignment. The teacher features differ in spatial resolution, channel width, and scale. Given an image I, the DeiT-S student encoder (Touvron et al., 2021) produces patch features $\mathbf{F}_s$. For each teacher m, we normalize its features per channel to obtain $\bar{\mathbf{F}}_m$, while a teacher-specific convolutional adapter A_m maps $\mathbf{F}_s$ to match $\bar{\mathbf{F}}_m$ in spatial resolution and channel width. All branches supervise dense patch features only, while CLS token features are excluded.

Distillation Objective. Once aligned, each adapted student feature $A_m(\mathbf{F}_s)$ is matched to its normalized teacher target $\bar{\mathbf{F}}_m$ using cosine and Smooth L1 losses. With M teachers and loss weights $\lambda_{\cos}$ and λ_{L1}, each branch receives equal weight:

$$\mathcal{L}_{\text{distill}} = \frac{1}{M} \sum_{m=1}^{M} \left[\lambda_{\cos} \mathcal{L}_{\cos}\big(A_m(\mathbf{F}_s), \bar{\mathbf{F}}_m\big) + \lambda_{\mathrm{L1}} \mathcal{L}_{\mathrm{SmoothL1}}\big(A_m(\mathbf{F}_s), \bar{\mathbf{F}}_m\big) \right]. \tag{2}$$

The cosine term aligns feature direction, while Smooth L1 encourages element-wise agreement. Only the student and adapters are optimized in this stage. After distillation, we discard the adapters and use the student encoder to initialize Stage 2, where register tokens aggregate its enriched patch features for planning.

4.2 Multi-Frame Register Adaptation

Multi-teacher visual distillation enriches the encoder's image features, whereas planning requires spatiotemporal representations across camera views and frames. Stage 2 places the distilled encoder

within the planning architecture of DrivoR (Kirby et al., 2026) and adapts it through end-to-end training. All patch tokens are retained during image encoding at this stage.

Spatial Attention. For each image $I_{t,v}$, the shared encoder processes camera-specific register tokens together with its patch tokens. Through self-attention, the registers attend to patches across the image and aggregate spatial information into $\mathbf{R}_{t,v}$.

Temporal Attention. Each image is initially encoded independently, so its register tokens do not interact across time. To capture scene evolution, we arrange the frame-wise register tokens of each camera into one sequence and add learned temporal embeddings and ego-motion embeddings derived from relative planar motion. We denote the combined embeddings by $\mathbf{E}_v$:

$$\mathbf{R}_v = \mathrm{TempAttn}([\mathbf{R}_{1,v}; \ldots; \mathbf{R}_{T,v}] + \mathbf{E}_v)\,. \tag{3}$$

Temporal attention operates independently within each camera stream. The register tokens from all views and observed frames are passed to the trajectory generator and scorer together with the ego state and navigation information $\mathbf{u}$.

End-to-End Objective. We retain the planner's trajectory generation and scoring objectives during end-to-end training:

$$\mathcal{L}_{\mathrm{E2E}} = \mathcal{L}_{\mathrm{traj}} + \mathcal{L}_{\mathrm{score}}, \tag{4}$$

where $\mathcal{L}_{\mathrm{traj}}$ is the minimum L1 loss over the trajectory candidates and $\mathcal{L}_{\mathrm{score}}$ is the binary cross-entropy loss for driving-quality scores. This objective adapts the spatiotemporal register representations to planning and shapes the register-to-patch attention used for token routing in Stage 3.

4.3 Register-Guided Token Compression

Multi-camera, multi-frame inputs provide richer visual evidence but increase the number of patch tokens processed by the encoder. Stage 3 therefore uses attention from the registers adapted for planning to reduce redundant patch computation. It starts from the complete Stage 2 checkpoint while retaining the planner and end-to-end objective.

Register-Guided Routing. We apply token compression in selected Transformer blocks. At each compression block, let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ denote the compressible sequence, consisting of unfused patch tokens and fused tokens carried forward from earlier blocks. We score each token by averaging the attention it receives from R register tokens over H attention heads:

$$s_i = \frac{1}{HR} \sum_{h=1}^{H} \sum_{j=1}^{R} w_{h,j,i}, \tag{5}$$

where $w_{h,j,i}$ is the attention weight from register token j to token $\mathbf{x}_i$ in head h. A higher score indicates stronger attention from the registers trained for planning. No auxiliary routing network or loss is introduced. The register attention used to compute the routing scores is shaped by the end-to-end driving objective in Equation (4).

Token Retention and Fusion. We always preserve the register and CLS tokens. At each compression block, among the current compressible tokens, we retain the fraction ρ with the highest routing scores and merge all lower-scored tokens into a single fused token $\mathbf{x}_{\mathrm{fuse}}$. We compute it as $\mathbf{x}_{\mathrm{fuse}} = \sum_i s_i \mathbf{x}_i$, where the sum is taken over the lower-scored tokens. The routing scores are not renormalized, preserving the original attention mass of the lower-scored set. The fused token proceeds through subsequent blocks and remains eligible for later compression.

5 Experiments

5.1 Experimental Setup

Benchmarks and Metrics. We train on the NAVSIM `navtrain` split with 103,288 scenes and evaluate on `navtest` with 12,146 scenes, using PDMS for NAVSIM v1 and EPDMS for NAVSIM v2 (Dauner et al., 2024; Cao et al., 2025). We further evaluate zero-shot closed-loop generalization on all 436 HUGSIM scenarios using HD-Score (Zhou et al., 2025).

Table 1: **Comparison on NAVSIM v1 `navtest` split.** ConDenseDrive uses $T = 2$ and $\rho = 0.4$. Bold values denote the best result among the listed learned methods.

Method	Image Encoder	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Human driver (Dauner et al., 2024)	–	100	100	100	99.9	87.5	94.8
UniAD (Hu et al., 2023)	ResNet-50	97.8	91.9	92.9	**100**	78.8	83.4
TransFuser (Chitta et al., 2022)	ResNet-34	97.7	92.8	92.8	**100**	79.2	84.0
DiffusionDrive (Liao et al., 2025)	ResNet-34	98.2	96.2	94.7	**100**	82.2	88.1
Hydra-MDP (Li et al., 2024)	V2-99	98.4	97.8	93.9	**100**	86.5	90.3
GoalFlow (Xing et al., 2025)	V2-99	98.4	98.3	94.6	**100**	85.0	90.3
DiffusionDriveV2 (Zou et al., 2025)	ResNet-34	98.3	97.9	94.8	99.9	87.5	91.2
DriveSuprim (Yao et al., 2026)	ViT-L	98.6	98.6	95.5	**100**	91.3	93.5
DrivoR (Kirby et al., 2026)	DINOv2-Small	99.0	**98.9**	96.7	**100**	90.0	93.7
RAP-DINO (Feng et al., 2026)	DINOv3-H+	99.1	**98.9**	96.7	**100**	90.3	93.8
ConDenseDrive (ours)	**Distilled DeiT-S**	**99.3**	98.8	**96.8**	**100**	**91.9**	**94.5**

Table 2: **Comparison on NAVSIM v2 `navtest` split.** ConDenseDrive uses $T = 2$ and $\rho = 0.4$. EPDMS* and EPDMS denote scores before and after the official evaluator correction, respectively.

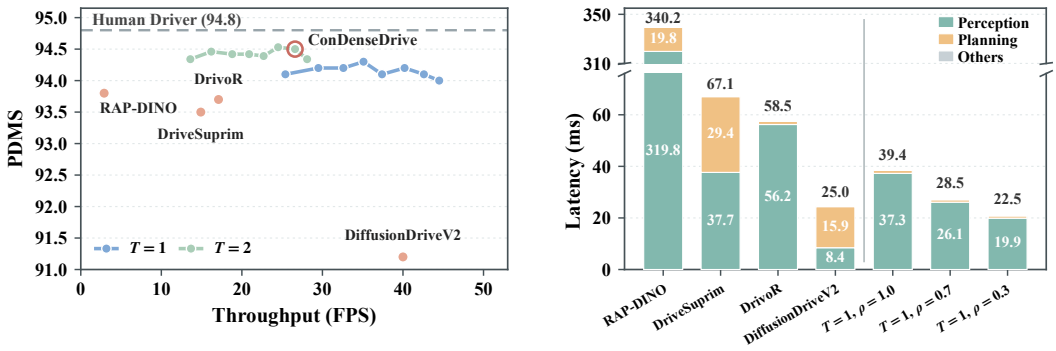
Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS* ↑	EPDMS↑
TransFuser (Chitta et al., 2022)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7	–
DiffusionDriveV2 (Zou et al., 2025)	97.7	96.6	99.2	**99.8**	88.9	97.2	96.0	97.8	**91.0**	85.5	87.5
Hydra-MDP++ (Li et al., 2025a)	98.4	98.0	99.4	**99.8**	87.5	97.7	95.3	98.3	77.4	85.1	–
Senna-2 (Song et al., 2026)	98.5	97.8	99.5	**99.8**	88.1	97.5	**97.0**	**98.6**	88.4	86.6	–
SparseDriveV2 (Sun et al., 2026)	98.1	98.1	99.6	**99.8**	**91.1**	97.3	96.9	98.2	78.4	86.7	90.1
DriveSuprim (Yao et al., 2026)	98.4	98.6	99.6	**99.8**	90.5	97.8	**97.0**	98.3	78.6	87.1	–
CLOVER (Ang et al., 2026)	**99.4**	**99.3**	99.5	**99.8**	86.9	**98.9**	95.2	98.2	75.5	**87.2**	90.4
ConDenseDrive (ours)	**99.4**	99.1	**99.7**	99.5	90.3	98.8	96.1	97.8	71.0	87.1	**90.8**

Implementation Details. We distill a DeiT-S student on front-view images from NAVSIM `navtrain` using frozen DINOv3 ViT-L/16, SigLIP2-SO400M NaFlex, and Depth Anything 3-Large teachers. End-to-end training uses front, rear, left, and right camera images resized to 1136×672, with $R = 16$ register tokens per view per frame. The distilled backbone weights remain frozen and are adapted with LoRA (Hu et al., 2021), while the downstream planner is trained jointly. Training uses four A100 GPUs, and inference latency is measured on one. Further training and evaluation details are in Appendix A.

5.2 Main Results

Result on NAVSIM v1. ConDenseDrive achieves 94.5 PDMS with $T = 2$ and $\rho = 0.4$, remaining only 0.3 below the human-driver reference (Tab. 1). Its NC, TTC, and EP scores of 99.3, 96.8, and 91.9 indicate strong collision avoidance while maintaining ego progress. With a 22M-parameter DeiT-S encoder, ConDenseDrive outperforms GoalFlow (V2-99, 70M), RAP-DINO (DINOv3-H+, 840M), and DriveSuprim (ViT-L, 304M) (Xing et al., 2025; Feng et al., 2026; Yao et al., 2026). DriveZero also uses a multi-teacher distilled ViT-S encoder with a DrivoR-style planner, but relies on broader distillation data and additional RL trajectory supervision (He et al., 2026). ConDenseDrive achieves higher PDMS through encoder optimization with less distillation data. These comparisons show that stronger visual representations can improve planning without increasing encoder capacity or redesigning the downstream planner.

Result on NAVSIM v2. Using the same $T = 2$ and $\rho = 0.4$ configuration, ConDenseDrive achieves 90.8 EPDMS with the updated official NAVSIM v2 evaluator and 87.1 EPDMS* with the original evaluation code (Tab. 2). CLOVER provides a relevant architectural comparison because it adopts a comparable DrivoR-style planner while focusing on proposal generation and trajectory ranking (Ang et al., 2026). ConDenseDrive instead focuses on visual encoder design and outperforms CLOVER by 0.4 EPDMS. It also improves EP by 3.4 points while matching CLOVER in NC and maintaining a comparable TTC score. This result supports visual encoder optimization as another effective direction for improving end-to-end planning.



(a) Accuracy–efficiency trade-off. **(b) Module-wise latency.**

Figure 3: **Accuracy–efficiency trade-off and module-wise latency.** (a) PDMS versus throughput for $T = 1$ and $T = 2$. Markers follow $\rho = 1.0, 0.9, \ldots, 0.3$ from left to right, and the outlined marker denotes the selected $T = 2$, $\rho = 0.4$ model. (b) End-to-end latency decomposed into perception, planning, and other costs.

Table 3: **Zero-shot closed-loop evaluation on HUGSIM.** ConDenseDrive uses $T = 1$ and $\rho = 0.7$.

Method	All 436 Scenarios						HD-Score				
	NC↑	DAC↑	TTC↑	COM↑	PDMS↑	RC↑	Easy↑	Medium↑	Hard↑	Extreme↑	Average↑
VAD (Jiang et al., 2023)	41.2	90.3	28.0	**95.3**	28.3	30.1	42.0	8.5	7.7	6.3	13.9
UniAD (Hu et al., 2023)	**65.6**	93.2	51.6	66.3	49.4	43.8	64.0	37.2	21.0	6.3	31.2
LTF (Chitta et al., 2022)	44.3	92.8	37.5	94.8	36.5	38.0	66.1	15.5	12.0	11.7	23.1
DrivoR (Kirby et al., 2026)	52.2	95.6	46.2	93.9	44.8	47.2	78.0	29.1	20.0	**14.1**	32.5
ConDenseDrive (ours)	60.3	**96.8**	**54.1**	93.9	**53.3**	**49.9**	**79.7**	**41.2**	**21.8**	13.3	**37.4**

Zero-Shot on HUGSIM. Our three-stage training pipeline uses no data from any of HUGSIM's four source datasets. Under this zero-shot setting, ConDenseDrive with $T = 1$ and $\rho = 0.7$ achieves 37.4 HD-Score on all 436 scenarios, outperforming DrivoR by 4.9 points (Tab. 3). The improvement is particularly pronounced on medium-difficulty scenarios, where the HD-Score increases from 29.1 to 41.2. NC, TTC, and RC also improve over DrivoR, indicating gains in collision avoidance and route completion. Since the downstream planner remains unchanged, this gain is achieved through encoder-centered optimization.

Inference Efficiency. Fig. 3(a) compares driving performance and throughput under the same hardware setting. Our $T = 2$, $\rho = 0.4$ model achieves 94.5 PDMS at 26.6 FPS. At $T = 2$, reducing ρ from 1.0 to 0.4 provides a 2.0× speedup while improving driving performance. ConDenseDrive is also faster and more accurate than RAP-DINO, DriveSuprim, and DrivoR. DiffusionDriveV2 is faster than the selected two-frame model but scores 3.3 points lower in PDMS. With $T = 1$ and $\rho = 0.3$, ConDenseDrive reaches 44.5 FPS at 94.0 PDMS, only 0.1 below its uncompressed counterpart. Across the $T = 1$ and $T = 2$ settings, throughput increases as ρ decreases, while PDMS varies by no more than 0.3. This stability suggests that a substantial share of dense patch-token computation can be avoided.

Fig. 3(b) shows that visual encoding dominates latency in most end-to-end models. For ViT-based systems, dense patch-token sequences are processed across successive Transformer blocks for each camera view and input frame, amplifying this latency bottleneck. At $T = 1$, token compression reduces perception latency by 30.0% at $\rho = 0.7$ and by 46.6% at $\rho = 0.3$, relative to $\rho = 1.0$. These reductions show that optimizing the visual encoder alone can substantially increase end-to-end system throughput.

5.3 Ablation Studies

We evaluate the complete three-stage pipeline to quantify how each stage affects planning performance and inference efficiency. We then isolate the key design choices in multi-teacher distillation, temporal modeling, and register-guided token compression.

Table 4: **Contribution of the three-stage framework.** Distill., Temporal, and Compress. denote visual distillation, temporal attention, and token compression, respectively. ID 1 uses the original pretrained DeiT-S.

ID	Encoder	Distill.	Temporal	Compress.	Frames	Rate	FPS↑	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
1	DeiT-S	×	×	×	$T=1$	$\rho=1.0$	25.4	98.7	97.9	95.6	**100**	88.0	91.8
2		√	×	×	$T=1$	$\rho=1.0$	25.4	99.1	98.8	96.6	**100**	91.2	94.1 $_{+2.3}$
3		√	√	×	$T=2$	$\rho=1.0$	13.6	99.2	**98.9**	**97.0**	**100**	91.2	94.3 $_{+2.5}$
4		√	√	√	$T=2$	$\rho=0.4$	**26.6**	**99.3**	98.8	96.8	**100**	**91.9**	**94.5** $_{+2.7}$

Table 5: **Routing token** ($T=1$, $\rho=0.7$).

Routing Token	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
CLS	**98.9**	98.9	96.1	**100**	91.5	94.0
Register	**98.9**	**99.0**	**96.3**	**100**	**91.8**	**94.3**

Table 6: **Compression init.** ($T=1$, $\rho=0.7$).

Initialization	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Scratch	**98.9**	98.7	**96.3**	**100**	90.1	93.6
Dense checkpoint	**98.9**	**99.0**	**96.3**	**100**	**91.8**	**94.3**

Contribution of Each Stage. Tab. 4 separates the contribution of each stage. ID 1 directly uses the original pretrained DeiT-S encoder and reaches 91.8 PDMS. Multi-teacher distillation increases PDMS by 2.3 points to 94.1 without changing throughput. Extending the input to two frames with temporal attention increases PDMS from 94.1 to 94.3, although using two frames reduces throughput to 13.6 FPS. Register-guided token compression at $\rho = 0.4$ restores throughput to 26.6 FPS and reaches 94.5 PDMS.

Table 7: **Keep rate** ($T=2$).

Rate	Encoder GFLOPs↓	FPS↑	PDMS↑
$\rho=1.0$	2363.62	13.6	94.3
$\rho=0.9$	1951.77	16.2	94.5
$\rho=0.8$	1637.24	18.8	94.4
$\rho=0.7$	1398.12	20.9	94.4
$\rho=0.6$	1215.52	22.7	94.4
$\rho=0.5$	1075.51	24.5	94.5
$\rho=0.4$	968.67	26.6	**94.5**
$\rho=0.3$	**887.02**	**28.1**	94.3

Routing Token. With $T = 1$ and $\rho = 0.7$ fixed, routing with the CLS token reaches 94.0 PDMS, while routing with register tokens reaches 94.3 (Tab. 5). Since register tokens form the visual interface to the planner, their attention is more directly aligned with planning-relevant information.

Compression Initialization. Tab. 6 compares two initialization strategies with $T = 1$ and $\rho = 0.7$. Scratch initialization loads the Stage 1 checkpoint only for the visual encoder and initializes the remaining components from scratch. Dense-checkpoint initialization loads all weights from Stage 2 before enabling token compression. This increases PDMS from 93.6 to 94.3, showing that driving adaptation before compression provides a stronger initialization.

Keep Rate. Tab. 7 evaluates different keep rates with $T = 2$. All compressed variants use the same training hyperparameters without rate-specific tuning. At $\rho = 0.4$, encoder computation decreases by 59.0% and throughput reaches 26.6 FPS, while PDMS improves from 94.3 to 94.5. Reducing ρ to 0.3 further raises throughput to 28.1 FPS while matching the $\rho = 1.0$ result. These results show that patch token computation can be substantially reduced without degrading driving performance.

5.4 Qualitative Analysis

Feature Visualizations. Fig. 4 compares adapter-aligned student features with their teacher references across three scenes. Compared with DINOv3, the student PCA projections more clearly delineate small objects and fine-grained road surface structure, while separating vehicle windows from surrounding body regions that DINOv3 tends to merge. For depth visualization, we decode the DA3-aligned student features using a lightweight convolutional head because the original DA3 head aggregates multiple feature layers. The resulting depth maps preserve the broad scene geometry of the DA3 predictions. These results show that the shared student representation captures complementary appearance and geometric cues from multiple teachers.

Token Compression Visualizations. Fig. 5 presents token compression patterns across the compression blocks. As compression proceeds, retained tokens increasingly concentrate on vehicles, pedestrians, and nearby obstacles, while tokens over the sky and distant buildings are fused. The allocation also varies across camera views. The front view preserves more road-surface tokens near

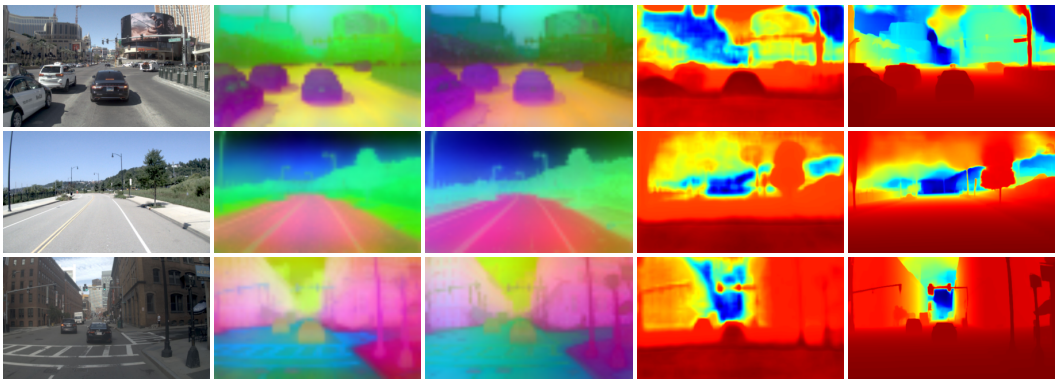


Figure 4: **PCA and depth visualizations.** From left to right, columns show the input image, PCA projections of the DeiT-S student and DINOv3 ViT-L/16 teacher, and depth maps from the student and Depth Anything 3-Large teacher.

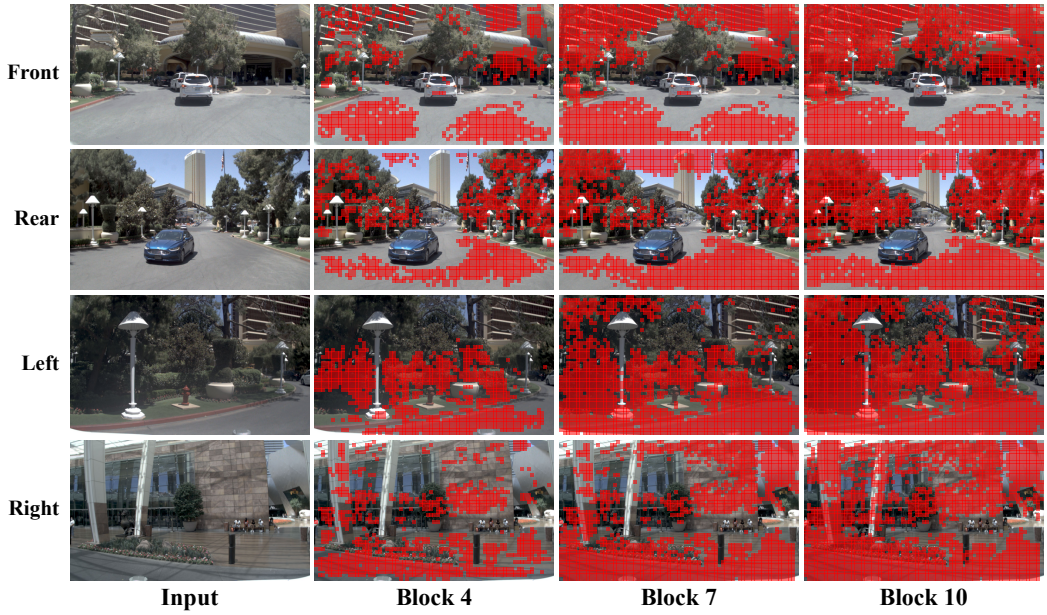


Figure 5: **Progressive register-guided token compression.** Rows show front, rear, left, and right camera views of the same scene. Columns show the input image and patch token states at Transformer blocks 4, 7, and 10 with $\rho = 0.7$. Red regions denote patch tokens assigned to fusion, while unmasked regions indicate retained patch tokens.

the driving path, whereas side views prioritize nearby obstacles. These observations indicate that register-guided routing adjusts token retention according to the visual context of each view.

6 Conclusion

We present ConDenseDrive, which revisits the visual encoder as a central design target for reliable and efficient end-to-end driving. The key idea is to enrich a compact encoder with complementary visual knowledge and concentrate computation on planning-relevant image patches. Our three-stage framework realizes this idea through multi-teacher distillation, multi-frame register adaptation, and register-guided token compression. Extensive experiments and qualitative analyses demonstrate that ConDenseDrive improves planning performance and inference efficiency. We hope this work encourages further research on driving-oriented visual encoders.

AI USE STATEMENT

Large Language Models (LLMs) were used solely to assist with manuscript drafting, improve clarity, grammar, and readability, and identify relevant literature.

REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. Clover: Closed-loop value estimation and ranking for end-to-end autonomous driving planning. *arXiv preprint arXiv:2605.15120*, 2026.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 2026.

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Xiangyu Li, Wenyu Liu, Qian Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *Advances in Neural Information Processing Systems*, 38:32551–32576, 2026a.

Hao Gao, Shaoyu Chen, Yifan Zhu, Yuehao Song, Wenyu Liu, Qian Zhang, and Xinggang Wang. Rad-2: Scaling reinforcement learning in a generator-discriminator framework. *arXiv preprint arXiv:2604.15308*, 2026b.

Hao He, Chengcheng Hu, Zirun Su, Heng Zhang, Haisong Liu, Jinke Li, Haochen Tian, Zhenwei Shen, Hongyang Li, Zhichao Li, et al. Drivezero: End-to-end driving beyond human demonstrations. *arXiv preprint arXiv:2609.06055*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pp. 533–549. Springer, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025a.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025b.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Zhixuan Liang, Yuxiao Chen, Yurong You, Peter Karkus, Wenhao Ding, Boyi Li, Alexander Popov, Yan Wang, Maximilian Igl, Yiming Li, et al. Planning-aligned token compression for long-context autonomous driving. *arXiv preprint arXiv:2606.07464*, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Amir Mallak, Erfan Aasi, Shiva Sreeram, Tsun-Hsuan Wang, Daniela Rus, and Alaa Maalouf. See less, drive better: Generalizable end-to-end autonomous driving via foundation models stochastic patch selection. *arXiv preprint arXiv:2601.10707*, 2026.

Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15522–15533. IEEE, 2024.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12490–12500. IEEE, 2024.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Unic: Universal classification models via multi-teacher distillation. In *European Conference on Computer Vision*, pp. 353–371. Springer, 2024.

Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Pau De Jorge, Diane Larlus, and Yannis Kalantidis. Dune: Distilling a universal encoder from heterogeneous 2d and 3d teachers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30084–30094. IEEE, 2025.

Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yuehao Song, Shaoyu Chen, Hao Gao, Yifan Zhu, Weixiang Yue, Jialv Zou, Bo Jiang, Zihao Lu, Yu Wang, Qian Zhang, et al. Senna-2: Aligning vlm and end-to-end driving policy for consistent decision making and planning. *arXiv preprint arXiv:2603.11219*, 2026.

Matthew Strong, Wei-Jer Chang, Quentin Herau, Jiezhi Yang, Yihan Hu, Chensheng Peng, and Wei Zhan. Learning to drive is a free gift: Large-scale label-free autonomy pretraining from unposed in-the-wild videos. *arXiv preprint arXiv:2602.22091*, 2026.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 446–463. Springer, 2026.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Linhan Wang, Zichong Yang, Chen Bai, Guoxiang Zhang, Xiaotong Liu, Xiaoyin Zheng, Xiao-Xiao Long, Chang-Tien Lu, and Cheng Lu. Drive-jepa: Video jepa meets multimodal trajectory distillation for end-to-end driving. *arXiv preprint arXiv:2601.22032*, 2026a.

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, et al. Wa-jepa: Rethinking the video jepa paradigm for world-action modeling in autonomous driving. *arXiv preprint arXiv:2608.20974*, 2026b.

Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2092–2101. IEEE, 2023.

Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. *arXiv preprint arXiv:2301.01006*, 2023.

Yichen Xie, Runsheng Xu, Tong He, Jyh-Jing Hwang, Katie Luo, Jingwei Ji, Hubert Lin, Letian Chen, Yiren Lu, Zhaoqi Leng, et al. S4-driver: Scalable self-supervised driving multimodal large language model with spatio-temporal visual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1622–1632, 2025.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611. IEEE, 2025.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14673–14684. IEEE, 2024.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10799–10808. IEEE, 2022.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

A Implementation Details

This section provides the evaluation metrics, three-stage training hyperparameters, and efficiency profiling protocol. Tab. 8 summarizes the settings used for the main ConDenseDrive model.

NAVSIM evaluation. We use the official Predictive Driver Model Score (PDMS) for NAVSIM evaluation (Dauner et al., 2024). PDMS measures safety, progress, and comfort by combining no-at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), driving comfort (C), and ego progress (EP):

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C}}{12}. \tag{6}$$

NAVSIM v2 adds driving direction compliance (DDC), traffic light compliance (TLC), lane keeping (LK), and extended comfort (EC), and replaces C with history comfort (HC) (Cao et al., 2025). The resulting Extended PDM Score (EPDMS) is

$$\mathrm{EPDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \mathrm{DDC} \cdot \mathrm{TLC} \cdot \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{LK} + 2\,\mathrm{HC} + 2\,\mathrm{EC}}{16}. \tag{7}$$

We report scores from the original evaluator as EPDMS* and those from the corrected official evaluator as EPDMS. All component scores in Equation (7) follow the official human-behavior filtering rule. A component is set to 1 if its score for the human reference is 0 and otherwise retains the model's score. Higher PDMS and EPDMS values indicate better planning performance.

HUGSIM evaluation. We evaluate zero-shot closed-loop generalization on all 436 HUGSIM scenarios without HUGSIM-specific training or fine-tuning (Zhou et al., 2025). HUGSIM defines No Collision (NC), Drivable Area Compliance (DAC), Time to Collision (TTC), Comfort (COM), Route Completion (RC), and the HUGSIM Driving Score (HD-Score). HD-Score aggregates the safety and comfort metrics over a closed-loop rollout and scales them by route completion.

We follow the HUGSIM evaluation protocol of WA-JEPA and take the baselines in Tab. 3 from its unified re-evaluation (Wang et al., 2026b). We also report PDMS from the same evaluation protocol. The six metrics under All 436 Scenarios are averaged uniformly across scenarios. For each difficulty level, HD-Score is averaged within each source dataset and then uniformly across datasets. The overall HD-Score weights the Easy (80), Medium (157), Hard (96), and Extreme (103) difficulty levels according to their scenario counts.

Training configuration. We use AdamW for all three stages. Stage 1 distills DeiT-S from DINOv3 ViT-L/16, SigLIP2-SO400M NaFlex, and Depth Anything 3-Large (DA3) on NAVSIM front-view images. We weight the cosine and Smooth L1 feature losses by 0.9 and 0.1. Stage 2 initializes the encoder from Stage 1, while the register tokens, temporal attention, and planning modules are initialized from scratch. Stages 2 and 3 sample input frames at 2 Hz, corresponding to a temporal interval of 0.5 s. The backbone stays frozen and is adapted with rank-32 LoRA updates to the query and value projections. The trajectory and score losses have equal weights. The NC, DAC, DDC, TTC, EP, and comfort ranking weights are $1/1/0/5/5/2$. Stage 3 starts from the complete Stage 2 checkpoint and enables register-guided token compression. It uses the same optimizer settings as Stage 2 with reinitialized optimizer state.

Efficiency profiling. We profile all methods with a batch size of one and four camera views on a single NVIDIA A100 GPU. We perform 100 warm-up iterations followed by $1{,}000$ timed iterations under PyTorch inference mode. Each forward pass is measured with CUDA events and synchronized before its latency is recorded. The timed region covers model inference and excludes data loading and host-side preprocessing. We report mean end-to-end latency and compute throughput as $1{,}000$ divided by mean latency in milliseconds. Perception denotes the image encoder, planning denotes the trajectory generation and scoring modules, and remaining framework overhead is grouped under Others. Encoder FLOPs cover all camera views and frames, include both attention matrix multiplications, and use $1\,\mathrm{MAC} = 2\,\mathrm{FLOPs}$.

Table 8: **Hyperparameters for ConDenseDrive.**

Stage	Parameter	Value
1	Number of epochs	50
	Batch size	128
	Learning rate	5×10^{-4}
	Weight decay	10^{-2}
	Warmup ratio	0.10
	Learning rate schedule	Cosine
2	Number of epochs	25
	Batch size	64
	Learning rate	2×10^{-4}
	Weight decay	10^{-2}
	Warmup ratio	0.10
	Learning rate schedule	Cosine
3	Number of epochs	10
	Batch size	64
	Learning rate	2×10^{-4}
	Weight decay	10^{-2}
	Warmup ratio	0.10
	Learning rate schedule	Cosine
	Compression blocks	4, 7, 10

B Additional Stage 1 Studies

Encoder transferability. We further evaluate the distilled encoder with a camera-only DiffusionDrive planner. In Tab. 9, we replace its ResNet-34 backbone with our distilled DeiT-S while keeping all training settings and the remaining architecture unchanged. This improves PDMS from 86.0 to 87.9 without token compression. Both encoders have approximately 22M parameters.

Table 9: **Encoder transfer to camera-only DiffusionDrive.**

Method	Image Encoder	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
DiffusionDrive (Liao et al., 2025)	ResNet-34	**97.9**	94.6	93.6	**100**	80.7	86.0
DiffusionDrive + distilled encoder	Distilled DeiT-S	**97.9**	**96.4**	**94.0**	**100**	**82.4**	**87.9**

C Additional Stage 2 Studies

Number of input frames. Tab. 10 compares dense models with different temporal lengths. Using two frames gives the best PDMS, while four frames increase computation without further improvement. We therefore use $T = 2$ for the main model.

Table 10: **Effect of the number of input frames** ($\rho = 1.0$)**.**

Frames	FPS↑	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
$T = 1$	**25.4**	99.1	98.8	96.6	**100**	**91.2**	94.1
$T = 2$	13.6	**99.2**	**98.9**	**97.0**	**100**	**91.2**	**94.3**
$T = 4$	7.3	99.1	98.5	96.8	**100**	91.1	93.9

View attention. The independently encoded register tokens also lack interaction across camera views. To study cross-view context, we arrange the view-wise register tokens at each time step:

$$\mathbf{R}_t = \mathrm{ViewAttn}([\mathbf{R}_{t,1}; \ldots; \mathbf{R}_{t,V}])\,. \tag{8}$$

This mirrors temporal attention in Equation (3), but groups camera views at a fixed time. Temporal attention reaches 94.3 PDMS and performs best among the variants in Tab. 11. Adding view attention does not improve the temporal model. Our final model therefore uses temporal attention without view attention and retains the registers from all input frames.

Table 11: **View and temporal attention variants** ($\rho = 1.0$)**.**

Attention	Frames	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
None	$T = 1$	99.1	98.8	96.6	**100**	**91.2**	94.1
View	$T = 1$	98.9	98.5	96.9	**100**	90.0	93.5
None	$T = 2$	99.0	98.7	96.8	**100**	90.9	94.0
Temporal	$T = 2$	**99.2**	**98.9**	**97.0**	**100**	**91.2**	**94.3**
View & temporal	$T = 2$	98.9	98.7	96.2	**100**	90.1	93.7

Register token scope. Tab. 12 compares the register tokens passed to the downstream planner. Using only the current-frame registers reaches 93.9 PDMS, while retaining registers from all input frames reaches 94.3. This result supports preserving the complete temporal register sequence after temporal attention.

Table 12: **Register scope** ($T = 2$, $\rho = 1.0$)**.**

Register Scope	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Current frame	99.1	98.7	96.6	**100**	91.0	93.9
All frames	**99.2**	**98.9**	**97.0**	**100**	**91.2**	**94.3**

Table 13: **Token fusion** ($T = 1$, $\rho = 0.7$)**.**

Token Handling	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Discard	**98.9**	98.9	96.1	**100**	91.7	94.1
Fuse	**98.9**	**99.0**	**96.3**	**100**	**91.8**	**94.3**

D Additional Stage 3 Studies

Token fusion. Tab. 13 compares token fusion with direct token discarding. Fusion reaches 94.3 PDMS, compared with 94.1 for discarding. This result supports retaining a compact summary of low-scoring patch tokens during compression.

Keep rate. Tab. 14 reports the keep rate sweep in the single-frame setting. At $\rho = 0.7$, encoder computation falls by about 41% while PDMS improves from 94.1 to 94.3. At $\rho = 0.3$, throughput reaches 44.5 FPS with only a 0.1 PDMS reduction from the dense model.

Table 14: **Keep rate** ($T = 1$)**.**

Rate	Encoder GFLOPs↓	FPS↑	PDMS↑
$\rho = 1.0$	1181.77	25.4	94.1
$\rho = 0.9$	975.85	29.5	94.2
$\rho = 0.8$	818.59	32.6	94.2
$\rho = 0.7$	699.02	35.1	**94.3**
$\rho = 0.6$	607.73	37.4	94.1
$\rho = 0.5$	537.72	40.2	94.2
$\rho = 0.4$	484.30	42.6	94.1
$\rho = 0.3$	**443.48**	**44.5**	94.0

E Additional Qualitative Visualizations

Feature Visualizations. Fig. 6 compares the student with DINOv3 and DA3 across two additional driving scenes.

Token Compression Visualizations. Figs. 7–8 show register-guided token compression across four camera views in two additional scenes.

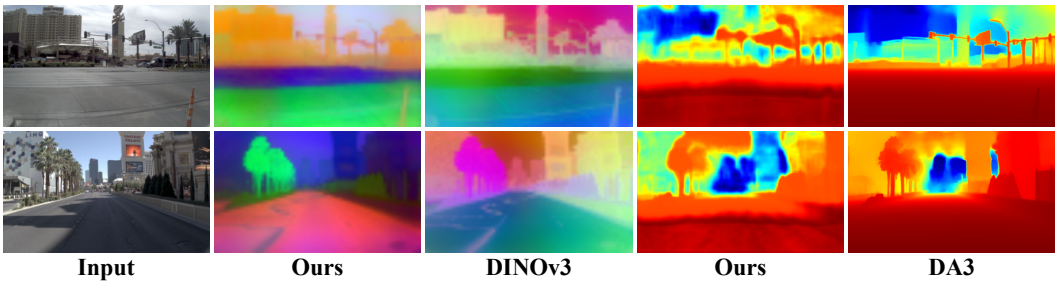


Figure 6: **PCA and depth visualizations.**

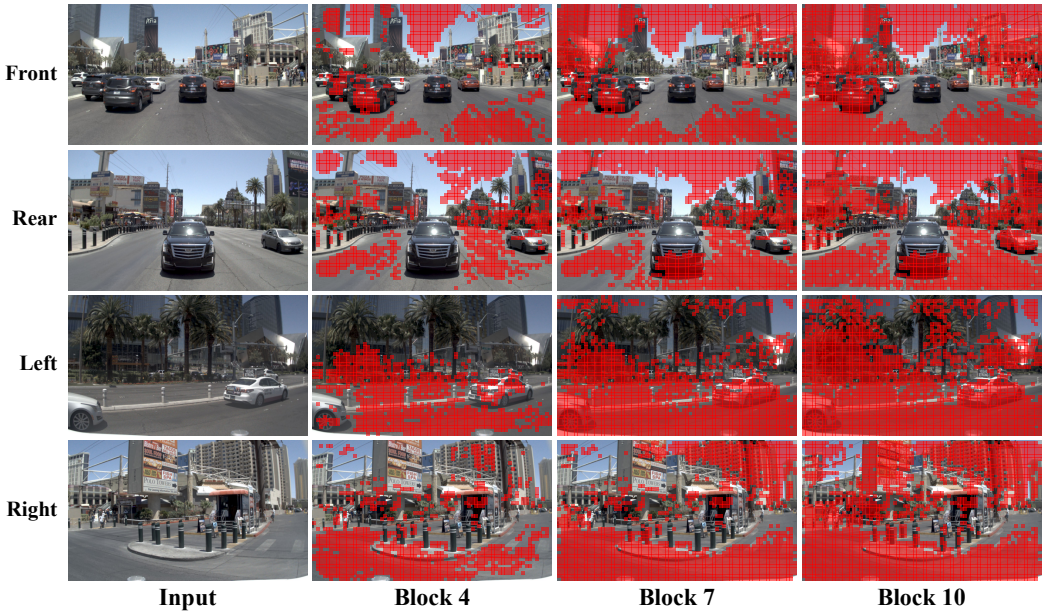


Figure 7: **Progressive register-guided token compression, Scene 1.** Results use $\rho = 0.7$.

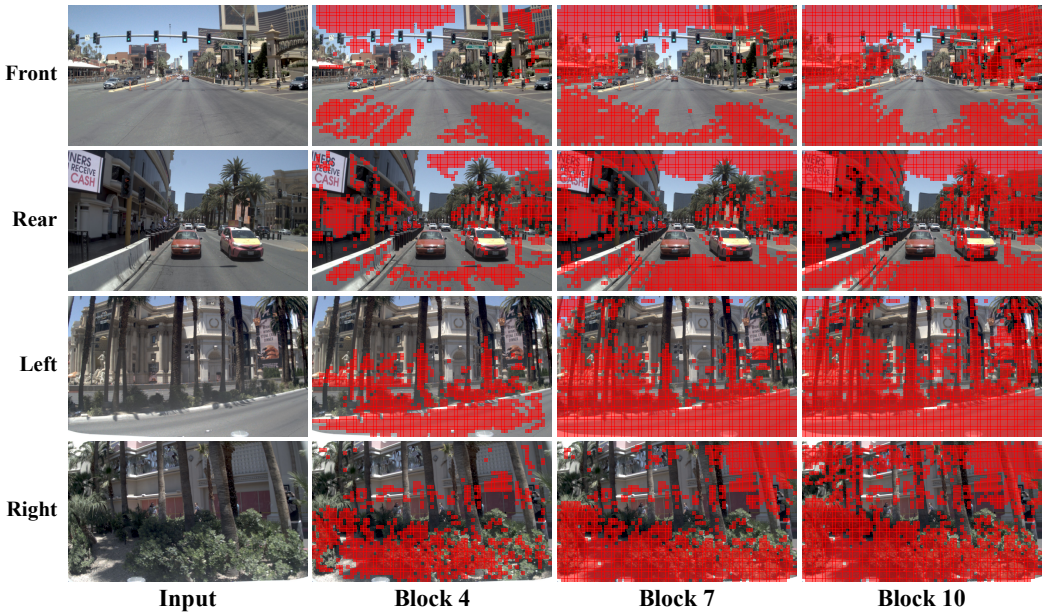


Figure 8: **Progressive register-guided token compression, Scene 2.** Results use $\rho = 0.7$.