NudgeVAD: Language-Nudged End-to-End Driving via FiLM Residuals

Chieh-Chi Yang* Yu-Hsiang Chen* Yi-Ting Chen
National Yang Ming Chiao Tung University
*Equal contribution

Abstract

*Natural-language instructions promise controllable end-to-end driving, but their benefit can be hidden when planners already receive reliable high-level commands. We propose **NudgeVAD**, a frozen-planner residual framework that uses language as a calibrated* nudge *to a VAD trajectory. With identity-initialized FiLM and a zero-initialized residual head, NudgeVAD is equivalent to the frozen planner at initialization, so learned deviations arise only from language-conditioned residuals.*

We evaluate NudgeVAD along a command-reliability axis. *With reliable commands, language improves the initial planner but becomes nearly redundant once compared against* VAD-FT (UNCOND)*, a compute-matched VAD model fine-tuned without language. With random commands, however, language becomes essential: detaching text degrades* ADE_{6s} *to* 3.166 *m, while NudgeVAD with text recovers* $\mathbf{2.806}$ *m and outperforms* VAD-FT (UNCOND) *by* 0.312 *m. These results show that language is not universally additive; it is most valuable when the categorical command channel is unreliable.*

1. Introduction

End-to-end autonomous driving has made strong progress with BEV-based and query-based planners that encode sensor-map context, reason over agents and lanes, and regress future ego trajectories [2, 4, 5, 7]. However, these planners typically infer the ego future from history and scene context alone, treating the ego vehicle like any other traffic participant. This ignores a key distinction: the ego vehicle may have access to privileged intent, such as a route command or a human instruction. Natural language offers a flexible way to express this intent, e.g., *"turn left at the next intersection"* or *"follow the white truck"*. The doScenes benchmark therefore asks whether language can improve trajectory prediction beyond a strong driving planner [1, 9].

Surprisingly, language is not always helpful in practice. The public instruction-conditioned doScenes result reports a negative conditioning gain, and we observe similar failures when naively fine-tuning a large language encoder with a strong planner. We argue that the issue is not merely model capacity or fusion design, but a structural redundancy between language and the categorical command channel. In VAD-style planners, the future decoder predicts multiple maneuver-conditioned trajectories, and a command selects the left, right, or straight mode [5]. When this command is reliable, it already provides much of the maneuver information that language would otherwise add. As a result, language can improve an initial checkpoint, yet become nearly redundant once compared against a compute-matched unconditional VAD model fine-tuned without language.

We propose **NudgeVAD**, a frozen-planner residual framework for studying when language is actually useful. Instead of jointly fine-tuning the planner, NudgeVAD keeps the VAD planner frozen and uses language only to predict a trajectory residual. A frozen LLaMA encoder maps the instruction to a sentence embedding, which FiLM-modulates the planner ego feature [8]; a lightweight MLP then predicts a residual added to the VAD trajectory. With identity-initialized FiLM and a zero-initialized residual head, NudgeVAD is exactly equivalent to the frozen planner at initialization, so any later deviation is learned through the language residual.

Our key finding is that the value of language depends on command reliability. With reliable commands, NudgeVAD substantially improves the initial planner, but adding the same language adapter on top of VAD-FT (UNCOND)—a compute-matched VAD model fine-tuned without language—yields only $+0.003$ m ADE gain. When commands are unreliable, the conclusion reverses: detaching text degrades ADE to 3.166 m, while NudgeVAD with text recovers $\mathbf{2.806}$ m and outperforms VAD-FT (UNCOND) by 0.312 m. Language is therefore not universally additive; it becomes most valuable when the categorical command channel is unreliable.

Contributions. We make three contributions:

- We identify command-language redundancy as a key reason why instruction conditioning may fail to improve strong VAD-style planners.

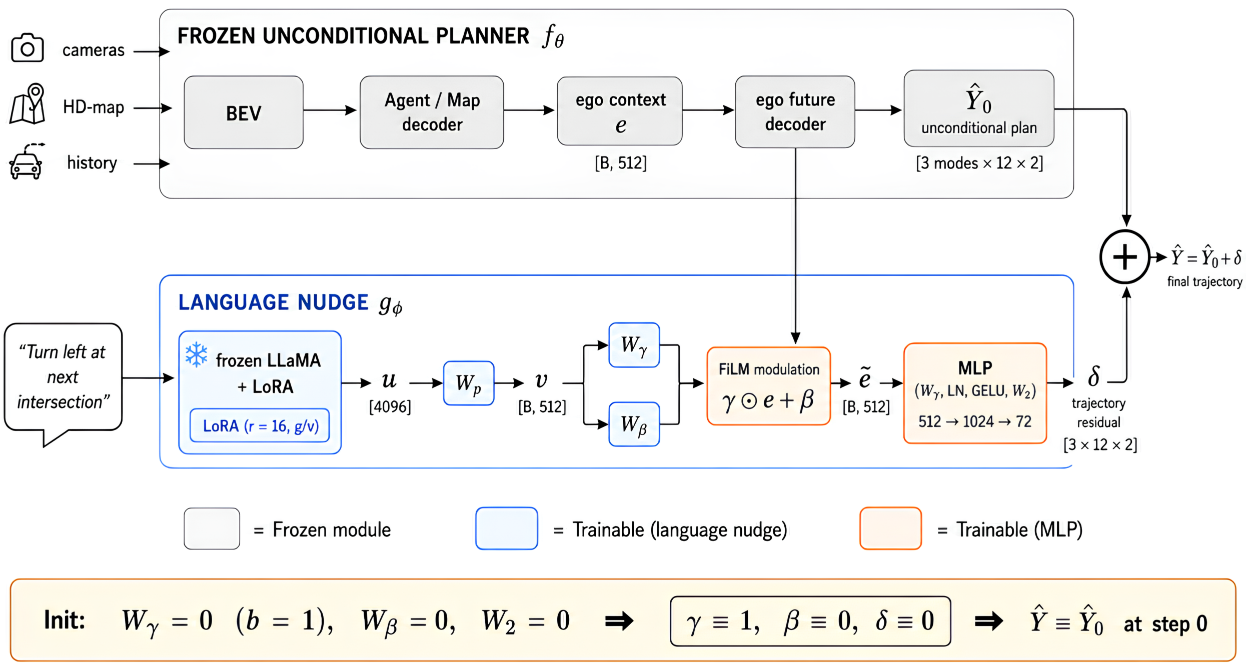

Figure 1. Overview of **NudgeVAD**. A frozen VAD planner predicts an unconditional trajectory $\hat{\mathbf{Y}}_0$ from sensor-map context, ego history, and a command $\mathbf{c}$ inferred from past-only lanelet geometry. The instruction is encoded by a frozen language model with lightweight adapters, FiLM-modulates the planner ego feature $\mathbf{e}$, and predicts a residual $\boldsymbol{\Delta}$. The final output is $\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_0 + \boldsymbol{\Delta}$. Identity initialization makes $\boldsymbol{\Delta} = 0$ at step zero, so NudgeVAD is initially identical to the frozen planner.

- We introduce **NudgeVAD**, a frozen-planner residual framework with identity initialization that isolates language-conditioned trajectory corrections without perturbing the pretrained planner.
- We evaluate language conditioning along a command-reliability axis and show that, under unreliable commands, NudgeVAD outperforms a compute-matched unconditional VAD fine-tuning baseline by 0.312 m.

2. Method

We propose **NudgeVAD**, a frozen-planner residual framework for instruction-conditioned driving. Given a pretrained VAD planner, NudgeVAD keeps the planner fixed and trains only a lightweight language branch to predict a trajectory correction. This isolates language from planner fine-tuning and preserves the planner's learned geometric priors.

A key protocol detail is that we do *not* use ground-truth commands. VAD-style planners often select the planning mode using a future-derived command or oracle navigation input [5, 6, 10]. Under doScenes, however, the command must be available at inference and must not leak future ego motion. We therefore infer $\mathbf{c}$ only from past ego history and HD-map lanelet geometry.

2.1. Residual Instruction Conditioning

Given ego history $\mathbf{H}$, sensor-map context $\mathbf{S}$, command $\mathbf{c} \in \{0, 1\}^3$, and instruction $\mathbf{t}$, the task is to predict the future ego trajectory $\mathbf{Y} \in \mathbb{R}^{T_f \times 2}$. We use $T_f = 12$ waypoints, corresponding to 6 seconds at 2 Hz. As shown in Fig. 1, NudgeVAD decomposes prediction into a frozen VAD trajectory and a language residual:

$$\hat{\mathbf{Y}} = \underbrace{f_\theta(\mathbf{H}, \mathbf{S}, \mathbf{c})}_{\text{frozen VAD}} + \underbrace{g_\phi(\mathbf{e}, \mathbf{t})}_{\text{language residual}}, \tag{1}$$

where $\mathbf{e}$ is the VAD ego planning feature. During NudgeVAD training, θ is fixed and only the residual branch g_ϕ is optimized.

2.2. Frozen 3-Mode VAD Planner

We instantiate f_θ with VAD-Tiny [5]. Given camera and map inputs, VAD encodes the scene into an ego planning feature $\mathbf{e} \in \mathbb{R}^{D_e}$ and predicts three command-conditioned trajectories:

$$\hat{\mathbf{Y}}_0 = \mathrm{EgoFutDec}_\theta(\mathbf{e}) \in \mathbb{R}^{3 \times T_f \times 2}. \tag{2}$$

The command selects the final trajectory, $\hat{\mathbf{y}}_0 = \hat{\mathbf{Y}}_0[\mathbf{c}]$. This routing structure is central to our analysis: when $\mathbf{c}$ is reliable, it already carries the maneuver class and can make language redundant; when $\mathbf{c}$ is unreliable, the instruction becomes the remaining maneuver-level signal.

2.3. Language Nudge

The residual branch encodes the instruction with a frozen LLaMA encoder and trainable LoRA adapters [3]. Given final token states $\mathbf{H}_t \in \mathbb{R}^{L \times D_t}$ and attention mask $\mathbf{m}$, we form a sentence embedding by masked mean pooling and project it to the planner feature space:

$$\mathbf{u} = \frac{1}{\sum_{\ell=1}^{L} m_\ell} \sum_{\ell=1}^{L} m_\ell \mathbf{H}_t[\ell], \qquad \mathbf{v} = W_p \mathbf{u}. \tag{3}$$

The projected instruction feature generates FiLM parameters [8] for the ego feature:

$$\boldsymbol{\gamma} = W_\gamma \mathbf{v} + \mathbf{1}, \qquad \boldsymbol{\beta} = W_\beta \mathbf{v}, \qquad \tilde{\mathbf{e}} = \boldsymbol{\gamma} \odot \mathbf{e} + \boldsymbol{\beta}. \tag{4}$$

A lightweight MLP then predicts a command-conditioned residual:

$$\boldsymbol{\Delta} = W_2 \,\mathrm{GELU}\left(\mathrm{LN}(W_1 \tilde{\mathbf{e}})\right) \in \mathbb{R}^{3 \times T_f \times 2}. \tag{5}$$

The final prediction is

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_0 + \boldsymbol{\Delta}, \qquad \hat{\mathbf{y}} = \hat{\mathbf{Y}}[\mathbf{c}]. \tag{6}$$

Thus, language does not replace the planner; it nudges the VAD trajectory through an additive residual.

2.4. Identity Initialization

To prevent the language branch from perturbing the frozen planner before learning, we initialize

$$\begin{gathered} W_\gamma = 0, \quad W_\beta = 0, \quad b_\gamma = \mathbf{1}, \quad b_\beta = 0, \quad W_2 = 0, \\ \boldsymbol{\gamma} = \mathbf{1}, \quad \boldsymbol{\beta} = \mathbf{0}, \quad \boldsymbol{\Delta} = \mathbf{0} \quad \Rightarrow \quad \hat{\mathbf{Y}} = \hat{\mathbf{Y}}_0. \end{gathered} \tag{7}$$

Therefore, NudgeVAD is exactly equivalent to the frozen VAD planner at initialization. Any later deviation is learned through the language residual.

2.5. Command-Reliability Probe

To separate language from the categorical command channel, we evaluate two command regimes. Let $\mathbf{c}_{\text{lanelet}}$ be the past-only lanelet command. We define

$$\mathbf{c} = \mathcal{N}_\rho(\mathbf{c}_{\text{lanelet}}) = \begin{cases} \mathbf{c}_{\text{lanelet}}, & \rho = \text{reliable}, \\ \mathrm{OneHot}(\mathrm{Uniform}\{0,1,2\}), & \rho = \text{random}. \end{cases} \tag{8}$$

The reliable regime tests whether language adds information beyond a strong command channel. The random regime removes this categorical shortcut at both training and inference, forcing maneuver information to enter through language. In both regimes, with-text and no-text passes use the same $\mathbf{c}$, so $\Delta\mathrm{ADE}$ isolates the contribution of the language residual.

2.6. Training Objective

NudgeVAD trains only the language residual branch. Given ground truth $\mathbf{Y}$, we minimize the weighted L1 trajectory loss

$$\mathcal{L}_{\text{traj}} = \sum_{t=1}^{T_f} w_t \left\|\hat{\mathbf{y}}_t - \mathbf{Y}_t\right\|_1 + \lambda_{\text{end}} \left\|\hat{\mathbf{y}}_{T_f} - \mathbf{Y}_{T_f}\right\|_1. \tag{9}$$

No auxiliary language loss is used. We use three stabilizing choices: removing the scalar residual gate, increasing residual-MLP capacity, and keeping frozen perception modules in evaluation mode to prevent batch-normalization drift.

3. Experiments

We evaluate whether language provides information beyond a categorical command channel. We distinguish two baselines: VAD-Init, the initial VAD checkpoint used by NudgeVAD, and VAD-FT (Uncond), a compute-matched VAD model fine-tuned on the same data without language. This distinction is important: language may improve an initial planner, yet become redundant once the unconditional planner is further optimized.

3.1. Setup

We evaluate on doScenes [9], built on nuScenes [1]. All methods predict one open-loop 6 s ego trajectory with $T_f = 12$ waypoints at 2 Hz. We report ADE, FDE, and

$$\Delta\mathrm{ADE} = \mathrm{ADE}_{\text{no-text}} - \mathrm{ADE}_{\text{with-text}},$$

computed using the same model and the same command $\mathbf{c}$.

We compare two command regimes. In the *reliable* regime, $\mathbf{c}$ is the past-only lanelet command. In the *random* regime, $\mathbf{c}$ is replaced by a uniformly sampled one-hot command at both training and inference. To control for optimization budget, VAD-FT (Uncond) continues training the VAD trunk for the same additional 60 epochs as NudgeVAD, but without language.

Stop override. During evaluation, we apply a conservative stop override only when both text and motion indicate stopping: a hard-stop cue, no conflicting action or stop-related noun phrase, word count ≤ 12, and history speed $\leq 2\,\mathrm{m/s}$.

3.2. Command Reliability Determines the Value of Language

Table 1 shows two regimes. With reliable commands, NudgeVAD substantially improves the initial VAD checkpoint, reducing ADE from 3.146 to 2.957. However, once the unconditional planner is compute-matched through additional fine-tuning, adding the same language adapter gives

Table 1. Command-reliability comparison. Under reliable commands, language improves VAD-INIT but becomes nearly redundant on top of VAD-FT (UNCOND). Under random commands, language provides a large same-model gain and outperforms the unconditional baseline.

Cmd	Method	ADE↓	FDE↓	ΔADE↑	Gain over VAD-FT
Reliable	VAD-INIT	3.146	7.238	—	—
	NudgeVAD on VAD-INIT	**2.957**	**6.763**	+0.293	−0.228
	VAD-FT (UNCOND)	2.729	6.396	—	—
	NudgeVAD on VAD-FT	2.726	6.376	+0.003	+0.003
Random	VAD-STAGE1	3.590	7.333	—	—
	VAD-FT (UNCOND)	3.118	6.517	—	—
	NudgeVAD w/o text	3.166	6.722	—	—
	NudgeVAD w/ text	**2.806**	**6.148**	**+0.36**	**+0.312**

Table 2. Adapter progression under unreliable command routing. A plain text residual is not enough; residual capacity and FiLM are the main drivers of the language gain. The conservative stop override has only a marginal effect.

Method	a@1s	a@2s	a@3s	a@4s	a@5s	a@6s	FDE	Gain@6s
VAD-STAGE1 no-cmd base	0.360	1.100	1.605	2.275	2.923	3.590	7.333	—
VAD-FT (UNCOND)	**0.337**	1.088	1.561	2.031	2.526	3.118	6.517	+0.472
Plain text residual	0.377	1.008	1.589	2.066	2.577	3.170	6.547	+0.420
Large residual MLP	0.369	0.912	1.418	1.863	2.367	2.961	6.341	+0.629
NudgeVAD (FiLM)	0.351	0.846	1.305	1.731	2.221	2.806	6.148	+0.784
Stop override	0.348	**0.836**	**1.290**	**1.713**	**2.197**	**2.774**	**6.071**	**+0.816**

only $+0.003$ m ADE gain. Thus, reliable commands make language useful on a weaker initial planner, but nearly redundant on top of a stronger unconditional one.

With random commands, the conclusion changes sharply. The no-text NudgeVAD pass fails because the command no longer provides a reliable maneuver signal, while the with-text model recovers strong performance, reducing ADE from 3.166 to 2.086. It also outperforms VAD-FT (UNCOND) by 0.312 m. This supports our central claim: language is most valuable when the categorical command channel is unreliable.

3.3. What Makes the Adapter Work?

Table 2 explains why the random-command regime benefits from language. A plain text residual does not outperform the compute-matched unconditional baseline at 6 s. Scaling the residual MLP closes much of the gap, and FiLM provides the largest improvement, reducing a@6s to 2.806 and outperforming VAD-FT (UNCOND) by 0.312 m. The stop override changes a@6s by only 0.032 m, indicating that the gain comes from the learned language residual rather than post-processing.

Limitations. ADE is not a complete measure of instruction following. In supplementary analysis, we find that lower ADE does not always improve semantic compliance, suggesting that future instructed-driving benchmarks should report both trajectory error and instruction-compliance metrics.

Implementation. NudgeVAD trains for 60 epochs with AdamW and a cosine learning rate of 10^{-4}. The planner is VAD-Tiny [5]. VAD-FT (UNCOND) uses the same training budget as NudgeVAD but contains no language branch. The language encoder is LLaMA-7B with LoRA adapters. No auxiliary language loss is used.

4. Conclusion

We presented **NudgeVAD**, a frozen-planner residual framework for instruction-conditioned driving. Our study shows that the value of language is conditional rather than universal. With reliable commands, language can improve an initial planner but becomes nearly redundant once compared against a compute-matched unconditional VAD fine-tuning baseline. With unreliable commands, however, language provides maneuver intent that the unconditional planner cannot recover, outperforming VAD-FT (UNCOND) by 0.312 m.

These findings suggest that instruction-conditioned driving should not be evaluated only under oracle or near-oracle command channels. Future benchmarks should explicitly vary command reliability and measure not only trajectory accuracy, but also whether the predicted motion satisfies the instruction. NudgeVAD provides a simple baseline for studying when language is redundant and when it is genuinely necessary for planning.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1, 3

[2] 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. 1

[3] Edward J. Hu, Yalu Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[4] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi tugboat Chai, Senyao Du, Tianyuan Lin, Wenhai Wang, Lewei Geng, Hongyang Li, Jiyan He, Jifeng Yu, Jifeng Dai, Yu Wang, and Ping Luo. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14229–14238, 2023. 1

[5] Bo Jiang, Shaoyu Chen, Qing Wang, Wenyu Liu, and Xinggang Wang. Vad: Vectorized autonomous driving via spatial-temporal graph neural networks. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision (ICCV), pages 8642–8651, 2023. 1, 2, 4

[6] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025. 2

[7] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chong Chong, Jifeng Yu, Xiaohui Liang, Yu Qiao Shao, Ping Shen, Wenyu Liu, Jialin Yang, Jie Zhou, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–18, 2022. 1

[8] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1, 3

[9] Parthib Roy, Srinivasa Perisetla, Shashank Shriram, Harsha Krishnaswamy, Keskar Aryan, and Ross Greer. doscenes: An autonomous driving dataset with natural language instruction for human interaction and vision-language navigation. *arXiv preprint arXiv:2412.05893*, 2024. 1, 3

[10] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025. 2