Hydra-NeXt: Robust Closed-Loop Driving with Open-Loop Training

Supplementary Material

1. Details on Training Hydra-NeXt

In this section, we provide the details on training Hydra-NeXt such as loss functions and the formulation of the diffusion policy π_{dp}. The three policies used in Hydra-NeXt (π_{traj}, π_{ctrl}, and π_{dp}) are all trained end-to-end simultaneously on the Bench2Drive dataset [11], which is collected by the RL-based expert Think2Drive [12]. External data from different experts such as PDMLite [2] are not used.

1.1. π_{traj}: Trajectory Decoder

The loss of π_{traj} is consistent with Hydra-MDP [13], which consists of two terms: an imitation loss $\mathcal{L}_{im}$ and a knowledge distillation loss $\mathcal{L}_{kd}$. We denote k trajectory anchors used in π_{traj} as $\{T_i\}_{i=1}^k$. The imitation loss is calculated as a cross entropy based on the expert trajectory $\hat{T}$ and the predicted imitation scores for each trajectory anchor $\{\mathcal{S}_i^{im}\}_{i=1}^k$:

$$\begin{aligned} y_i &= \frac{e^{-(\hat{T}-T_i)^2}}{\sum_{j=1}^k e^{-(\hat{T}-T_j)^2}} \\ \mathcal{L}_{im} &= -\sum_{i=1}^k y_i \log(\mathcal{S}_i^{im}). \end{aligned} \tag{1}$$

y_i measures the similarity between the expert trajectory $\hat{T}$ and each trajectory anchor T_i. The knowledge distillation loss aims to learn open-loop objectives via binary cross-entropy between predicted metric scores $\{\mathcal{S}_i^m\}_{i=1}^k$ and the ground-truth metric scores $\{\hat{\mathcal{S}}_i^m\}_{i=1}^k$:

$$\begin{aligned} \mathcal{L}_{kd} = -\sum_{m,i} \hat{\mathcal{S}}_i^m \log \mathcal{S}_i^m + (1-\hat{\mathcal{S}}_i^m)\log(1-\mathcal{S}_i^m) \\ \text{where } m \in \{COL, SLK, EP\}. \end{aligned} \tag{2}$$

COL, SLK, EP correspond to the Collision, Soft Lane Keeping, and Ego Progress metrics defined in Sec. 3. The ground-truth metric scores are binary values, derived from the privileged information and each trajectory anchor. The overall loss of π_{traj} is formulated as:

$$\mathcal{L}_{traj} = \mathcal{L}_{im} + \mathcal{L}_{kd}. \tag{3}$$

1.2. π_{ctrl}: Control Decoder

The loss of π_{ctrl} computes a classification-based loss for each control signal (*i.e.* brake, throttle, and steer) across t_{ctrl} timesteps. For simplicity, a control signal tuple $C = (brake, throttle, steer)$ is abbreviated as (b, th, s), while the expert demonstration is denoted as $(\hat{b}, \hat{th}, \hat{s})$. Specifically, these loss functions are formulated as:

$$\begin{aligned} \mathcal{L}_{brake} &= \sum_{t=1}^{t_{ctrl}} Focal(b^t, \hat{b}^t) \\ \mathcal{L}_{throttle} &= \sum_{t=1}^{t_{ctrl}} CE(th^t, \hat{th}^t) \\ \mathcal{L}_{steer} &= \sum_{t=1}^{t_{ctrl}} CE(s^t, \hat{s}^t) \\ \mathcal{L}_{ctrl} &= \mathcal{L}_{brake} + \mathcal{L}_{throttle} + \mathcal{L}_{steer}, \end{aligned} \tag{4}$$

where $Focal$ stands for the focal loss [15] and CE is the cross entropy loss. We empirically find that applying a focal loss to throttle and steer predictions leads to a performance degradation, possibly due to their more balanced distribution in the dataset.

1.3. π_{dp}: Diffusion Policy

π_{dp} is based on the standard Diffusion Policy [3] trained on continuous data. Given expert control signals $(\hat{C^1}, ..., \hat{C^{t_{dp}}})$ across t_{dp} frames, π_{dp} is trained to predict the noise ε^j added to the expert control signals, where j is the denoising iteration. MSE loss is applied for noise prediction:

$$\mathcal{L}_{dp} = MSE(\varepsilon^j, \pi_{dp}((\hat{C^1}, ..., \hat{C^{t_{dp}}}) + \varepsilon^j, j)). \tag{5}$$

Finally, the overall loss of Hydra-NeXt becomes

$$\mathcal{L} = \mathcal{L}_{traj} + \mathcal{L}_{ctrl} + \mathcal{L}_{dp}. \tag{6}$$

2. Performance of Individual Policies

We conduct experiments on how each individual policy performs on the Bench2Drive Benchmark. As shown in Tab. 2, using π_{ctrl} alone leads to serious performance degradation compared with the full version of Hydra-NeXt (-16.56 DS and -29.33 SR). Moreover, π_{dp} achieves better results when its control candidate follows the predicted trajectory from π_{traj} rather than being randomly selected (+5.36 DS and +4.28 SR), highlighting the importance of trajectory guidance. Finally, the full version of Hydra-NeXt achieves substantial improvements (+13.09 DS and +17.47 SR) compared with the baseline Hydra-MDP (π_{traj}).

π_{traj}	π_{ctrl}	π_{dp}	Driving Score ↑	Success Rate ↑
✓			52.80	30.73
	✓		49.33	18.87
		Rand.	51.18	27.98
		Traj.	56.54	32.26
✓	✓	✓	**65.89**	**48.20**

Table 2. **Performance of Individual Policies on Bench2Drive.**

3. Efficient Diffusion Policy π_{dp}

We found that the efficiency of π_{dp} can be greatly enhanced without sacrificing too much performance. By replacing the DDPM with a DDIM scheduler [17], we can reduce the latency by approximately 53%, decreasing the number of denoising steps from 100 to 20. Additionally, by combining this modification with flash attention, we achieve latencies

Method	Perception Network	Grid Search	NC ↑	DAC ↑	TTC ↑	EP ↑	C ↑	PDMS ↑
Transfuser [4]	Transfuser [4]	-	97.7	92.8	92.8	79.2	**100**	84.0
DRAMA [18]	Transfuser [4]*	-	98.0	93.1	94.8	80.1	**100**	85.5
Hydra-MDP [13]	Transfuser [4]	✗	97.9	91.7	92.9	77.6	**100**	83.0
Hydra-MDP [13]	Transfuser [4]	✓	98.3	96.0	94.6	78.7	**100**	86.5
DiffusionDrive [14]	Transfuser [4]	-	98.2	96.2	94.7	**82.2**	**100**	88.1
Hydra-NeXt	Transfuser [4]	✗	**98.4**	95.9	**94.8**	80.6	**100**	87.2
Hydra-NeXt	Transfuser [4]	✓	98.1	**97.7**	94.6	81.8	**100**	**88.6**

Table 1. **Performance of E2E-AD Methods on NAVSIM.** All methods above use Transfuser [4] with ResNet34 [9] as the perception backbone. *DRAMA [18] uses Mamba [5] for multi-modal interaction. Hydra-MDP [13] uses grid search to obtain the optimal hyper-parameters for weighting different predicted metric scores.

Method	Latency (ms) ↓	D.S. ↑	S.R. ↑
VAD	224.3	39.42	10.00
Hydra-NeXt (w/o π_{dp})	250.6	60.40	48.10
Hydra-NeXt* (DDPM)	528.3	65.89	48.20
Hydra-NeXt* (DDIM)	243.9	64.87	46.63

Table 3. **Efficiency of Different Diffusion Schedulers.** * denotes Flash-attention [6].

Method	Expert	D.S. ↑	S.R. ↑
Hydra-NeXt	Think2Drive	73.86	50.00
TF++† [10]	PDM-Lite	84.21	67.27
SimLingo [16]	PDM-Lite	85.94	66.82
Hydra-NeXt*†	PDM-Lite	86.00	68.18

Table 4. **Performance of Models Trained with Different Experts.** †Ensemble of three models trained with different seeds.

comparable to VAD while maintaining strong closed-loop performance, with only a marginal 1% drop in the Driving Score. Therefore, we conclude that π_{dp} can be both effective and efficient with the right design choices.

4. Implementation on NAVSIM

Our implementation of Hydra-NeXt on NAVSIM uses a different perception network Transfuser [4] following Hydra-MDP [13]. Transfuser [4] features two backbones for camera and lidar feature extraction, a BEV segmentation head, a 3D object detection head, and transformer layers for multi-modal feature interaction. This setting helps to make a fair comparison to baselines such as Hydra-MDP and DiffusionDrive [14]. For π_{ctrl} and π_{dp}, we incorporate the same transformer architectures used on Bench2Drive for acceleration and steering rate predictions since NAVSIM [7] does not utilize control signals like CARLA [1, 8] and only evaluates the trajectory. Therefore, these auxiliary predictions only act as extra learning targets. As a result, Hydra-NeXt surpasses the state-of-the-art planner DiffusionDrive by 0.5 PDMS (see Tab. 1) when adopting the grid search trick [13] among different metric scores in π_{traj}.

5. Implementation on CARLA-Garage

We provide additional results on the CARLA-Garage dataset [10] collected by the PDM-Lite expert [2]. Compared to the Bench2Drive dataset [11], this dataset features smooth driving with no jittery behavior. We experiment with π_{traj} and π_{dp} within the TF++ framework [10]. Specifically, we replace the longitudinal control head of TF++ with π_{traj} (imitation + collision head) and fuse the outputs of π_{dp} with longitudinal and lateral controls. We then ensemble three models trained with different seeds. This variant, Hydra-NeXt*, achieves a Driving Score of 86%, surpassing existing methods.

6. Visualization Results

Fig. 1 shows more visualization results in interactive scenarios (Merging, Overtaking, and Give Way). The Diffusion Policy π_{dp} can capture multiple planning modes such as following other agents or overtaking them (see the second row and the fourth row of the figure).

7. Limitations.

Although Hydra-NeXt shows outstanding closed-loop driving performance compared with E2E methods, it still falls behind RL-based experts using privileged input. The runtime efficiency of the Diffusion Policy also deserves optimization. We expect these to be addressed in future research.

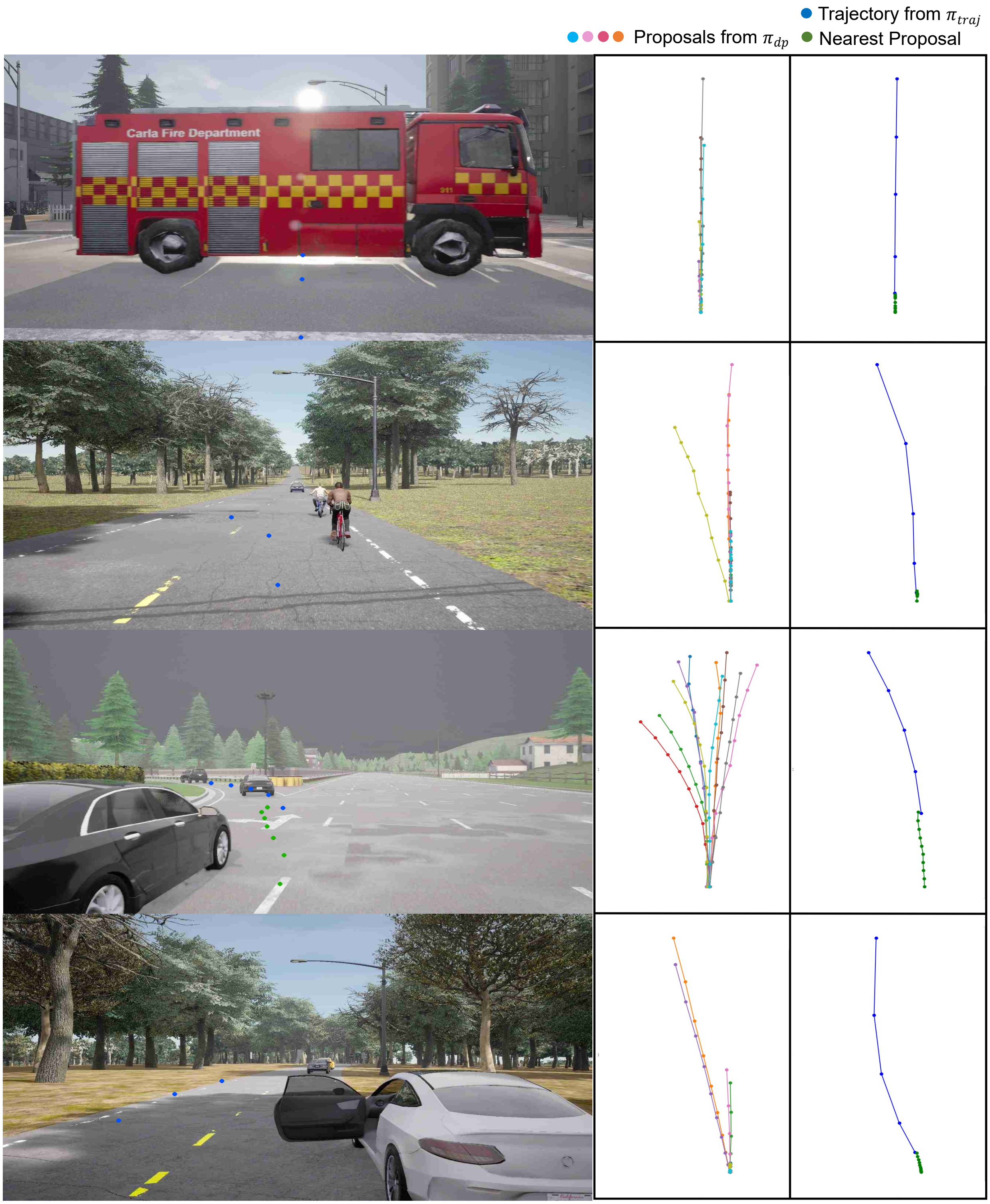

Figure 1. **More Visualizations of Trajectory Refinement.** The figure shows the front-view image, trajectories transformed from control proposals, and the selected nearest proposal to the predicted trajectory.

References

[1] Carla: Carla autonomous driving leaderboard (2022). https://leaderboard.carla.org/, 2022. 2

[2] Jens Beißwenger. PDM-Lite: A rule-based planner for carla leaderboard 2.0. https://github.com/

OpenDriveLab/DriveLM/blob/DriveLM-CARLA/docs/report.pdf, 2024. 1, 2

[3] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 1

[4] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE TPAMI*, 2022. 2

[5] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 2

[6] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[7] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, 2024. 2

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16. PMLR, 2017. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[10] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 2

[11] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS*, 2024. 1, 2

[12] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking in latent world model for quasi-realistic autonomous driving (in carla-v2). In *ECCV*, 2024. 1

[13] 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. 1, 2

[14] 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. *arXiv preprint arXiv:2411.15139*, 2024. 2

[15] T Lin. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1

[16] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025. 2

[17] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. 1

[18] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024. 2