Beyond Raw Observations: Distilling and Storing Invariant Driving Memories for Generalizable Autonomous Driving

Xue Zhao[1], Cewu Lu[1], Xinbing Wang[1], Nanyang Ye[1, 2, 3]*
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3]Shanghai Innovation Institute, Shanghai, China

Abstract

Modern end-to-end autonomous driving systems often suffer from fragile out-of-distribution (OOD) generalization, largely due to their reliance on raw sensory observations and susceptibility to spurious visual correlations. Inspired by human drivers, who abstract recurrent interaction patterns rather than memorizing pixel-level details, we argue that robust generalization requires invariant, decision-relevant representations. To this end, we propose **DriveMem**, a framework that moves beyond raw observations by distilling and storing invariant driving memories for generalizable driving systems. DriveMem integrates two core components: *Invariant Feature Abstraction* that first distills decision-relevant representations from visual tokens, and *Prototype Memory Bank* that stores and retrieves prototypical driving experiences grounded in invariant representations. Extensive open-loop and closed-loop experiments demonstrate that DriveMem improves OOD generalization and safety-related planning metrics across diverse unseen scenarios. Notably, under severe sensor disruptions, DriveMem achieves a 39.5% relative reduction in Collision Rate compared with a strong generalization baseline. These results suggest that building prototype memory in an invariant representation space is a promising approach for improving robustness, while offering prototype-grounded traceability for model behavior.

1 Introduction

Human drivers are able to generalize to previously unseen traffic scenarios not by memorizing pixel-level visual appearances, but by abstracting recurrent interaction patterns and reusing them as structured experience [1, 2]. For instance, yielding at an intersection under extreme weather conditions may involve significantly different visual cues compared to normal weather, yet the underlying decision logic could be similar. This observation motivates our core hypothesis: enhancing the generalization capability of autonomous driving models requires learning feature representations of invariant interaction patterns, rather than superficial visual statistics.

Modern end-to-end (E2E) autonomous driving systems are typically trained to map sensor observations to future trajectories. While such systems have achieved strong performance, their heavy reliance on raw visual cues introduces spurious correlations between environmental appearance (e.g., weather) and driving actions, resulting in significant performance degradation under distribution shifts [3]. Moreover, these black-box mappings lack traceable decision rationale, making it difficult to diagnose failure modes and limiting their reliable deployment in safety-critical scenarios [4].

*Corresponding author: ynylincoln@sjtu.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

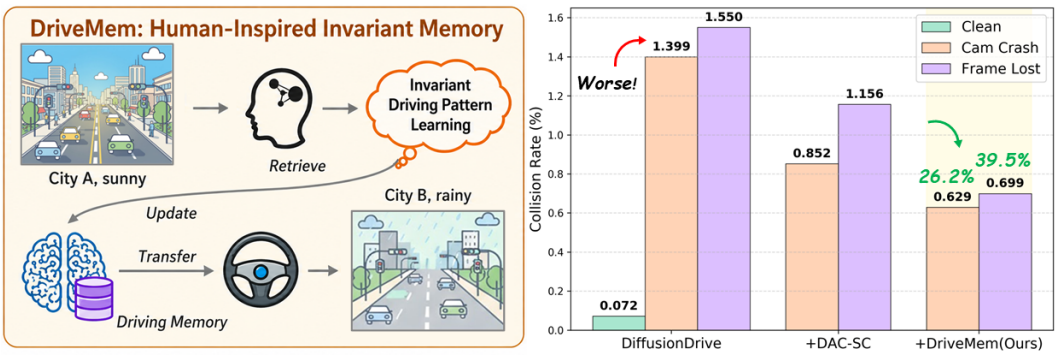

Figure 1: **Human-inspired invariant driving memory for robust OOD generalization.** Human drivers can transfer prior driving experience across visually different but semantically related scenarios. Inspired by this mechanism, DriveMem distills visual observations into invariant representations and stores them as prototype-level memories, enabling retrieval of relevant driving patterns under shifted conditions. Under severe sensor degradation, DriveMem reduces Collision Rate relative to DAC-SC by 26.2% under Camera Crash and 39.5% under Frame Lost, indicating that invariant prototype memory provides robust priors when visual observations become unreliable.

Existing generalization research has primarily focused on domain-invariant representation learning [5, 6, 7], aiming to mitigate distribution shifts via feature alignment. However, this static alignment is inherently limited: in complex, long-tail driving scenarios, alignment alone fails to support dynamic decision-making, which requires active retrieval and reuse of specific historical interaction experiences. The implicit memory encoded in neural network weights can capture general patterns but struggles to handle rare situations. To address this bottleneck, the autonomous driving community has begun integrating explicit memory mechanisms [8, 9] that store driving experiences. Yet, existing approaches often construct retrieval keys directly from raw observations or shallow semantic features to recall memory, making "appearance-biased retrieval" a critical issue: distribution shifts can disrupt matching based on superficial features, causing semantically similar but visually different critical scenarios to be missed. Therefore, achieving reliable experience reuse in dynamic, cross-domain scenarios requires memory mechanisms to operate on decision-relevant representations that remain stable across domains. Specifically, it is necessary to abstract domain-invariant key patterns, and use them as robust retrieval anchors to drive accurate matching in independent memory banks.

To this end, we propose **DriveMem**, a novel framework that moves beyond raw observations by distilling and storing invariant driving memories for generalizable autonomous driving. In DriveMem, **Invariant Feature Abstraction** first encourages latent visual tokens to retain decision-relevant information while reducing sensitivity to appearance-level perturbations. Subsequently, **Prototype Memory Bank** stores compact prototypes over the invariant representations and retrieves semantically related patterns to augment downstream trajectory prediction. As illustrated in Figure 1, analogous to human drivers, DriveMem enables generalization across scenes that differ in appearance but are semantically related. By retrieving prototype priors aligned with decision-relevant driving structures, DriveMem improves robustness. Additionally, it provides prototype-grounded traceability: model predictions can be inspected together with representative retrieved experiences, offering example-level evidence for downstream analysis. Our contributions are summarized as follows:

- We propose **DriveMem**, a novel memory-augmented framework that improves OOD generalization of vision-based E2E driving systems.
- We introduce an **invariant prototype memory** mechanism that first abstracts visual tokens into stable decision-relevant representations and then deposits these representations into a compact Prototype Memory Bank for structured retrieval and experience reuse.
- Extensive open-loop and closed-loop experiments demonstrate that DriveMem improves OOD generalization and safety-related planning metrics across diverse distribution shifts.

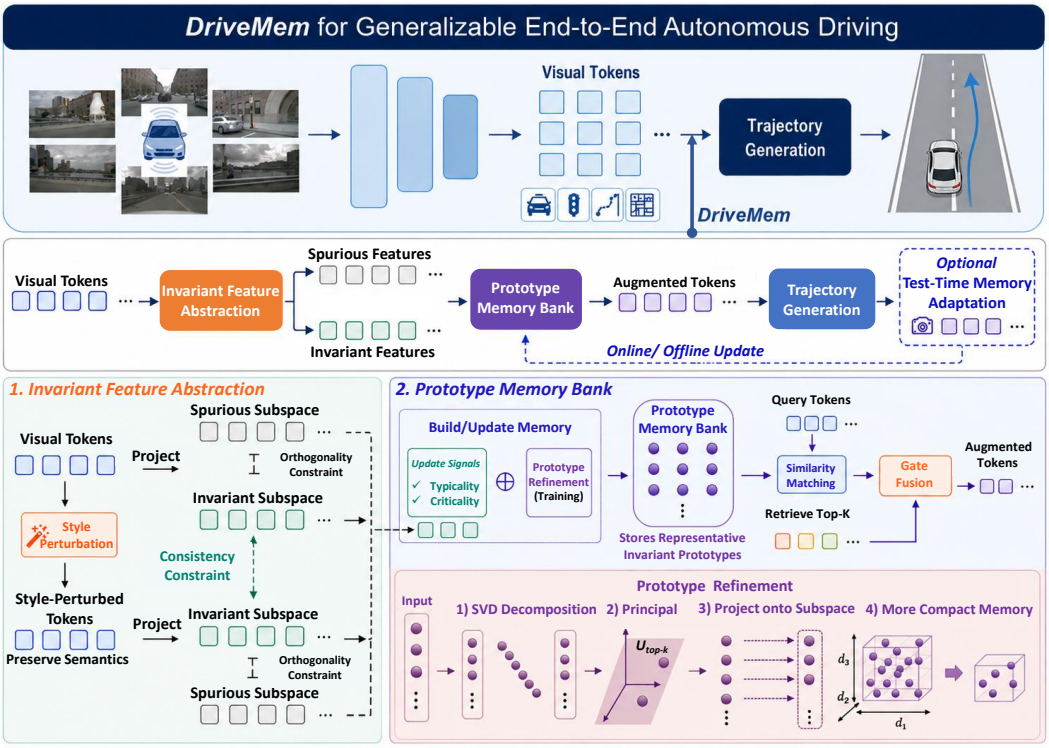

Figure 2: **Overview of DriveMem for End-to-End Autonomous Driving.**

2 Methodology

2.1 Problem Definition

We consider vision-based end-to-end autonomous driving, where a model predicts a future ego trajectory from multi-view images. At time step t, given images I_t, the model predicts

$$\hat{\tau}_t = f_\theta(I_t), \qquad \min_\theta \mathbb{E}_{(I_t,\tau_t)}\big[\mathcal{L}_{\text{E2E}}(\hat{\tau}_t, \tau_t)\big], \tag{1}$$

where $\hat{\tau}_t$ and τ_t denote the predicted and ground-truth trajectories. Despite strong in-distribution performance, E2E driving models often degrade under out-of-distribution (OOD) conditions, especially when appearance-level factors such as lighting, weather, or sensor quality change. This suggests that predictions may rely on domain-specific visual shortcuts rather than stable interaction patterns.

To improve generalization, DriveMem abstracts visual tokens into invariant, decision-relevant representations and stores them as compact prototype-level driving memories. Unlike memory mechanisms that store raw observations or heuristic rules, DriveMem builds memory in an invariant representation space, enabling structured retrieval under visual shifts.

2.2 Overview of DriveMem

As shown in Figure 2, DriveMem is a novel module for existing vision-based E2E driving pipelines. It first uses **Invariant Feature Abstraction (IFA)** to regularize latent visual tokens into stable decision-relevant representations, and then uses **Prototype Memory Bank (PMB)** to store these invariant representations as compact driving prototypes and retrieve them as structured priors for trajectory prediction. This design aims to ensure that memory retrieval is guided less by surface-level appearance similarity and more by planning-relevant structure.

2.3 Invariant Feature Abstraction

Given visual tokens $\mathbf{Z} \in \mathbb{R}^{B\times N\times D}$ from the perception backbone, IFA aims to reduce nuisance visual variations while preserving decision-relevant information. It serves as an invariance-regularized projection mechanism that encourages perturbation-stable representations and suppresses feature leakage through orthogonality.

Invariant-Spurious Projection. We first expand the latent tokens and project them into invariant and spurious subspaces:

$$\mathbf{Z}_{\text{exp}} = \phi_{\text{exp}}(\mathbf{Z}), \qquad \mathbf{Z}_{\text{inv}} = \mathbf{Z}_{\text{exp}}\mathbf{W}_{\text{inv}}, \qquad \mathbf{Z}_{\text{spu}} = \mathbf{Z}_{\text{exp}}\mathbf{W}_{\text{spu}}, \tag{2}$$

where $\phi_{\text{exp}}(\cdot)$ is an MLP, and $\mathbf{W}_{\text{inv}} \in \mathbb{R}^{D_h\times d_{\text{inv}}}$, $\mathbf{W}_{\text{spu}} \in \mathbb{R}^{D_h\times d_{\text{spu}}}$ are learnable projections. Only $\mathbf{Z}_{\text{inv}}$ is fed to downstream prediction, encouraging the planner to rely primarily on stable decision-relevant information.

Style-Perturbation Consistency & Orthogonality Regularization. To simulate appearance-level domain shifts, we apply style perturbations to latent visual tokens:

$$\tilde{\mathbf{Z}} = \text{StylePerturb}(\mathbf{Z}), \qquad \tilde{\mathbf{Z}}_{\text{inv}} = \phi_{\text{exp}}(\tilde{\mathbf{Z}})\mathbf{W}_{\text{inv}}, \tag{3}$$

where the perturbation mainly changes appearance statistics while approximately preserving driving semantics [10]. Let $g_\theta(\cdot)$ denote the downstream prediction head. We impose prediction-level consistency under perturbation:

$$\mathcal{L}_{\text{style}} = \mathbb{E}\left[\left\| g_\theta(\tilde{\mathbf{Z}}_{\text{inv}}) - g_\theta(\mathbf{Z}_{\text{inv}})\right\|_2^2\right]. \tag{4}$$

To further reduce leakage between the two branches, we use

$$\mathcal{L}_{\text{orth}} = \mathbb{E}_i\left[\left(\frac{\mathbf{z}_{\text{inv},i}^\top \mathbf{z}_{\text{spu},i}}{\|\mathbf{z}_{\text{inv},i}\|_2\|\mathbf{z}_{\text{spu},i}\|_2 + \epsilon}\right)^2\right]. \tag{5}$$

This term serves as a lightweight anti-leakage regularizer rather than a guarantee of statistical independence or causal disentanglement.

The overall IFA objective combines both terms:

$$\mathcal{L}_{\text{IFA}} = \lambda_{\text{style}}\mathcal{L}_{\text{style}} + \mathcal{L}_{\text{orth}}. \tag{6}$$

Without such invariant regularization, memory retrieval may be dominated by appearance-level similarity, causing prototypes to encode domain-specific shortcuts rather than stable driving patterns.

2.4 Prototype Memory Bank

After IFA, DriveMem stores invariant features in a Prototype Memory Bank (PMB):

$$\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_K] \in \mathbb{R}^{K\times d_{\text{inv}}}, \tag{7}$$

where K is the memory size. PMB aims to learn compact prototypes that cover representative and difficult driving patterns in the invariant space, rather than memorizing all features uniformly. Prototype-level memory is computationally efficient and encourages reuse of structured experience rather than memorization of individual observations.

Weighted Reconstruction Loss. Memory updates are driven by a coverage-hardness trade-off over invariant features. We use two practical criteria: *typicality*, which captures confidently matched recurring patterns, and *criticality*, which highlights difficult or potentially failure-prone samples.

(1) Prototype assignment. For each invariant feature $\mathbf{z}_i \in \mathbb{R}^{d_{\text{inv}}}$, we compute soft assignment weights over the prototype bank $\mathbf{M} = \{\mathbf{m}_k\}_{k=1}^K$:

$$w_{ik} = \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{m}_k)/\tau)}{\sum_{j=1}^K \exp(\text{sim}(\mathbf{z}_i, \mathbf{m}_j)/\tau)}, \qquad \hat{\mathbf{z}}_i = \sum_{k=1}^K w_{ik}\mathbf{m}_k, \tag{8}$$

where $\mathrm{sim}(\cdot,\cdot)$ measures feature-prototype similarity, τ controls assignment sharpness, and $\hat{\mathbf{z}}_i$ is the prototype-reconstructed feature.

(2) Typicality score. We use assignment entropy to measure how confidently a sample aligns with an existing prototype:

$$t_i = 1 - \frac{H(\mathbf{w}_i)}{\log K}, \qquad H(\mathbf{w}_i) = -\sum_{k=1}^{K} w_{ik} \log w_{ik}. \tag{9}$$

Here $\log K$ is the maximum entropy under a uniform assignment. A high t_i indicates low assignment entropy, meaning that the sample is confidently explained by a dominant prototype. We use t_i as a proxy for prototype-level typicality, encouraging the memory to learn recurring driving patterns.

(3) Criticality score. To emphasize difficult samples, we compute a standardized task-loss score:

$$h_i = \max\left(0, \frac{\ell_i - \mu_\ell}{\sigma_\ell + \epsilon}\right), \tag{10}$$

where ℓ_i is the per-sample E2E prediction loss, and μ_ℓ and σ_ℓ are the batch-wise mean and standard deviation. This score selects samples with above-average loss, which serve as proxies for challenging driving cases.

To make the selection scale-invariant, we convert t_i and h_i into batch-wise percentile ranks π_i^t and π_i^h. A sample is selected for memory optimization if it ranks highly in either dimension:

$$\alpha_i = \mathbb{I}\left[\max(\pi_i^t, \pi_i^h) \geq \gamma\right], \tag{11}$$

where γ is a percentile threshold. The weighted reconstruction loss is then defined as

$$\mathcal{L}_{\mathrm{recon}} = \frac{1}{\sum_i \alpha_i + \epsilon} \sum_i \alpha_i \left\|\hat{\mathbf{z}}_i - \mathbf{z}_i\right\|_2^2. \tag{12}$$

This dual-criterion selection encourages prototypes to cover both recurring dominant patterns and difficult high-loss samples, mitigating memory collapse toward trivial modes.

Prototype Refinement. To reduce redundancy, we align prototypes with the dominant variation directions of selected invariant features. Let $\mathbf{U}_r \in \mathbb{R}^{d_{\mathrm{inv}} \times r}$ be the top-r singular vectors computed from their covariance. We regularize

$$\mathcal{L}_{\mathrm{refine}} = \left\|\mathbf{M} - \mathbf{M}\mathbf{U}_r\mathbf{U}_r^\top\right\|_F^2. \tag{13}$$

This refinement is used only during training and encourages more compact memory space.

Prototype Retrieval and Adaptive Fusion. Given flattened invariant features $\mathbf{Z}_{\mathrm{inv}}^{\mathrm{flat}} \in \mathbb{R}^{BN \times d_{\mathrm{inv}}}$, PMB retrieves prototype priors and fuses them with current invariant features:

$$\begin{aligned} &\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{Z}_{\mathrm{inv}}^{\mathrm{flat}}\mathbf{M}^\top}{\tau}\right), \qquad \hat{\mathbf{Z}}_{\mathrm{inv}}^{\mathrm{flat}} = \mathbf{A}\mathbf{M}, \\ &\mathbf{G} = \sigma\Big(\mathbf{W}_{\mathrm{gate}}[\mathbf{Z}_{\mathrm{inv}}^{\mathrm{flat}}, \hat{\mathbf{Z}}_{\mathrm{inv}}^{\mathrm{flat}}] + \mathbf{b}_{\mathrm{gate}}\Big), \qquad \mathbf{Z}_{\mathrm{fused}}^{\mathrm{flat}} = \mathbf{G} \odot \mathbf{Z}_{\mathrm{inv}}^{\mathrm{flat}} + (1 - \mathbf{G}) \odot \hat{\mathbf{Z}}_{\mathrm{inv}}^{\mathrm{flat}}. \end{aligned} \tag{14}$$

The fused features are reshaped to $\mathbf{Z}_{\mathrm{fused}} \in \mathbb{R}^{B \times N \times d_{\mathrm{inv}}}$ and fed to the downstream planner. The PMB objective is

$$\mathcal{L}_{\mathrm{PMB}} = \mathcal{L}_{\mathrm{recon}} + \lambda_{\mathrm{svd}}\mathcal{L}_{\mathrm{refine}}. \tag{15}$$

Through retrieval and gated fusion, PMB adaptively balances the current invariant feature and the retrieved prototype prior, enabling structured experience reuse without storing raw observations.

2.5 Training and Inference

The overall training objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{E2E}} + \mathcal{L}_{\mathrm{IFA}} + \mathcal{L}_{\mathrm{PMB}}. \tag{16}$$

During training, the perception backbone, IFA, PMB, and downstream planner are jointly optimized end-to-end. During standard inference, DriveMem performs retrieval-only memory augmentation with fixed prototypes. When human corrective feedback is available, DriveMem supports **optional** feedback-guided memory adaptation (**Appendix E.1**), a lightweight, supervised update of prototype memory and coupled layers to rapidly align with new scenarios.

3 Theoretical Insight

We provide a concise upper-bound analysis to interpret the empirical improvements of DriveMem. Rather than aiming for a distribution-free guarantee, the analysis serves as an interpretable decomposition of the target-domain risk. Let $\phi(x) = (u(x), v(x))$ denote the learned representation, where u captures invariant decision-relevant factors and v captures spurious factors. DriveMem predicts via

$$h_M(x) = g(r_M(u(x))).$$

Theorem 3.1 (Informal)**.** *Under standard boundedness, Lipschitzness, and approximate conditional invariance assumptions, with probability at least* $1 - \eta$*,*

$$\begin{aligned} R_{D_t}(h_M) \le \widehat{R}_{D_s}(h_M) + 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}} \\ + (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\mathrm{cond}} \\ + 2L_\ell L_g \epsilon_M. \end{aligned} \tag{17}$$

Here R_{D_t} denotes target population risk, $\widehat{R}_{D_s}$ denotes empirical source risk, $\mathfrak{R}_n$ is the empirical Rademacher complexity, $W_1(P_s^u, P_t^u)$ measures invariant-space discrepancy, λ_{cond} denotes residual conditional mismatch, and ϵ_M denotes memory approximation error.

The bound shows that target risk is governed by source fitting, hypothesis complexity, invariant-space discrepancy, residual conditional mismatch, and memory approximation error. Within this decomposition, IFA mainly reduces the invariant discrepancy term by enforcing perturbation consistency and suppressing spurious leakage, while PMB mainly reduces the memory approximation term and can also reduce effective complexity through prototype-based representation compression. **A formal version of the theorem and its proof are provided in Appendix C.**

4 Experiments

4.1 Implementation Details

We use two representative end-to-end autonomous driving models as backbones: VAD [11] and DiffusionDrive [12]. VAD represents vectorized deterministic planning, while DiffusionDrive represents generative multi-modal trajectory planning. For fair comparison, we follow their official architectural designs and training pipelines, and integrate the proposed *DriveMem* module without modifying the core perception-planning backbone. We compare DriveMem with several representative domain generalization methods, including DAC-SC [13], DSU [14], FCVL [10], and the causality-inspired invariant representation learning method CIRL [7]. We also include Pin [15] as a representative memory-augmented baseline, which stores prototypical or historical features and recalls them during inference. For DriveMem, we set the default loss weights to $\lambda_{\mathrm{orth}} = \lambda_{\mathrm{svd}} = \lambda_{\mathrm{style}} = 1.0$, the temperature to $\tau = 0.1$, and the percentile threshold to $\gamma = 0.8$ for selecting typical and critical samples. **Detailed evaluation protocols, dataset splits, and metrics are provided in Appendix D.**

4.2 Main Results

Open-Loop Evaluation: nuScenes → nuScenes-C. We first evaluate DriveMem on nuScenes and nuScenes-C using DiffusionDrive [12] as the base model. As shown in Table 1, DriveMem achieves the best OOD average performance, reducing the L2 error from 0.9080 to 0.7458 and the collision rate from 0.662 to 0.270 compared with the base DiffusionDrive model. This corresponds to a 17.9% reduction in trajectory error and a 59.2% reduction in collision rate. Compared with the strongest competing baseline, DriveMem further reduces the OOD average collision rate by 22.4% and the OOD average L2 error by 4.1%. The gains are most pronounced under severe sensor degradation. In Camera Crash and Frame Lost, DriveMem reduces the collision rate from 1.399/1.550 to 0.629/0.699 compared with the base model, and achieves relative reductions of 26.2% and 39.5% compared with DAC-SC. These results suggest that invariant prototype memory is especially beneficial when instantaneous visual observations become unreliable. Meanwhile, DriveMem maintains competitive clean-domain performance (0.069 collision rate and 0.5950 L2 error), indicating that the robustness improvement does not come at the cost of nominal performance. Although DriveMem is not uniformly

Table 1: Comparison with baseline methods on nuScenes and nuScenes-C. "Clean" denotes the normal validation set of nuScenes. "OOD Avg." is the average performance of eight testing domains unseen during training. **Note that lower values are better for both Collision Rate and L2 Error.** All results reported below are obtained under the standard inference setting, without Test-Time Memory Adaptation (TTMA).

	Model	Clean	Cam Crash	Frame Lost	Color Quant	Motion Blur	Bright	Low Light	Fog	Snow	OOD Avg.
Collision Rate(%)↓	DiffusionDrive [12]	0.072	1.399	1.550	0.231	0.481	0.110	0.407	0.106	1.015	0.662
	+DSU [14]	0.129	0.960	1.492	0.188	0.157	0.094	0.230	0.095	0.278	0.437
	+DAC-SC [13]	0.078	0.852	1.156	0.145	0.121	**0.081**	**0.149**	0.084	**0.196**	0.348
	+CIRL [7]	0.080	1.206	1.357	0.200	0.315	0.102	0.265	0.121	0.452	0.502
	+Pin [15]	0.075	1.029	0.959	0.161	0.256	0.085	0.169	**0.080**	0.523	0.407
	+FCVL [10]	0.088	0.896	0.983	0.223	0.162	0.101	0.225	0.144	0.222	0.369
	+DriveMem (Ours)	**0.069**	**0.629**	**0.699**	**0.145**	**0.107**	0.097	0.184	0.098	0.199	**0.270** (**22.4%**)
L2(m)↓	DiffusionDrive [12]	0.5995	1.1371	1.3357	0.7697	0.8835	0.6508	0.8432	0.6583	0.9860	0.9080
	+DSU [14]	**0.5866**	0.9009	1.4636	0.7160	0.6984	**0.6190**	0.7329	0.6319	0.7308	0.8117
	+DAC-SC [13]	0.6313	0.9928	1.1598	0.6893	0.6683	0.6426	**0.7097**	0.6400	**0.7157**	0.7773
	+CIRL [7]	0.6261	0.9603	1.2169	0.7210	0.7117	0.6564	0.7341	0.6681	0.7782	0.8058
	+Pin [15]	0.6019	0.9942	1.1014	0.7503	0.6781	0.6355	0.7312	0.6425	0.8034	0.7921
	+FCVL [10]	0.6043	1.2188	1.2446	0.8182	0.7113	0.6712	0.7599	0.6681	0.8023	0.8618
	+DriveMem (Ours)	0.5950	**0.8941**	**1.0159**	**0.6851**	**0.6667**	0.6231	0.7103	**0.6216**	0.7495	**0.7458** (**4.1%**)

Table 2: Comparison with baseline methods on the NAVSIM benchmark. ↑ indicates higher is better. Due to the partial absence of LiDAR point clouds in `navhard_two_stage`, all methods are evaluated in an image-only setting for fair comparison†. All results reported below are obtained under the standard inference setting, without Test-Time Memory Adaptation (TTMA).

Method	**Input**	**navtest (ID)**						**navhard_two_stage (OOD)**		
		NC↑	**DAC↑**	**TTC↑**	**Comf↑**	**EP↑**	**PDMS↑**	**EPDMS (S1)↑**	**EPDMS (S2)↑**	**EPDMS ↑**
ResAD [16]	C+L	98.0	**97.5**	94.1	**100.0**	**83.3**	**88.8**	–	–	–
DiffusionDrive [12]	C+L	98.2	96.2	94.7	100.0	82.2	88.1	–	–	–
+DriveMem (Ours)	C+L	**98.4**	96.7	**95.0**	**100.0**	82.9	**88.8**	–	–	–
Image-only baselines										
DiffusionDrive† [12]	C	97.93	94.52	93.71	100.00	80.51	86.01	71.66	37.88	26.95
+DSU [14]	C	98.11	95.61	94.12	99.99	81.52	87.26	73.65	39.28	28.81
+CIRL [7]	C	98.25	95.14	94.42	99.98	81.42	87.02	38.05	30.41	12.49
+Pin [15]	C	97.22	92.65	92.49	99.98	77.81	83.19	35.47	27.91	10.98
+FCVL [10]	C	98.00	95.05	93.76	99.99	81.04	86.53	71.91	37.82	27.98
+DriveMem (Ours)	**C**	**98.32**	**96.27**	**94.66**	**100.00**	**82.43**	**88.12**	**75.26**	**39.38**	**30.53(+5.97%)**

best on every individual corruption type, its strong OOD average and tail-risk improvements indicate that the proposed memory mechanism mainly strengthens failure-prone scenarios, which are critical for robust autonomous driving.

Closed-Loop Evaluation: NAVSIM navtrain → navhard. We further evaluate our method on NAVSIM, a non-reactive pseudo-simulation benchmark, using navtest as the in-distribution split and navhard_two_stage as the OOD split. As shown in Table 2, DriveMem outperforms the image-only DiffusionDrive baseline across all metrics, improving EP from 80.51 to 82.43 and PDMS from 86.01 to 88.12 while preserving a perfect comfort score. Notably, despite using only camera inputs, DriveMem achieves performance comparable to the original LiDAR-camera DiffusionDrive on navtest, and even slightly surpasses it on NC, DAC, EP, and PDMS. This suggests that invariant prototype memory can provide useful structural priors that partially compensate for the absence of explicit LiDAR geometry. Furthermore, our plug-and-play DriveMem elevates DiffusionDrive to performance levels comparable to ResAD [16], a recent SOTA method, without requiring architectural changes. This demonstrates the effectiveness of our memory mechanism as a lightweight enhancement module.

On the OOD navhard_two_stage split, DriveMem obtains the best overall EPDMS score among all image-only methods, improving over the image-only DiffusionDrive baseline from 26.95 to 30.53, corresponding to a 13.3% relative gain. Compared with the strongest image-only baseline, DriveMem still improves the final EPDMS by 5.97%. These closed-loop results support the open-loop findings: DriveMem improves generalization not by simply increasing model capacity, but by retrieving invariant driving prototypes that provide robust priors under shifted or challenging scenarios.

Table 3: Ablation Study of DriveMem's Different Components. The metric is the average performance of OOD domains. SPCL: Style-Perturbation Consistency Loss; OR: Orthogonality Regularization; WRL: Weighted Reconstruction Loss; PR: Prototype Refinement.

Method	IFA		PMB		L2 Error↓	Collision Rate↓
	SPCL	OR	WRL	PR		
#0	✗	✗	✗	✗	0.9080	0.662
#1	✓	✗	✗	✗	0.7923	0.421
#2	✓	✓	✗	✗	0.7827	0.397
#3	✓	✓	✓	✗	0.7702	0.310
#4	✓	✓	✓	✓	0.7458	0.270

Table 4: Ablation Study on Different Memory Sizes. Ego memory size: 64 vs 128. Agent memory size: 1024 vs 2048 vs 4096. The performance is the average metric of eight testing domains from nuScenes-C unseen during training.

Ego Size	Agent Size	L2 Error↓	Collision Rate↓
64	1024	0.8012	0.367
64	2048	0.7731	0.374
64	4096	0.7990	0.294
128	1024	0.7949	0.280
128	2048	0.7702	0.310
128	4096	0.7722	0.291

4.3 Ablation Study

4.3.1 Effects of DriveMem's Different Components

We systematically evaluate the effects of DriveMem's two core modules, Invariant Feature Abstraction (IFA) and Prototype Memory Bank (PMB) on model generalization across eight unseen test domains on DiffusionDrive [12]. The ablation study is shown in Table 3.

First, baseline method (#0) exhibits the worst generalization performance (L2 Error: 0.9080, Collision Rate: 0.662), highlighting the limitations of relying solely on raw visual features for OOD prediction. Introducing the IFA module (#1 and #2) leads to significant performance gains, demonstrating that IFA effectively separates task-relevant features from confounding noise, enhancing model robustness under domain shift. Then, we add Weighted Reconstruction Loss (WRL, #3) to the IFA, indicating that the memory learning mechanism based on "typical and critical" samples strengthens the model's ability to handle long-tail and high-risk scenarios. The introduction of Prototype Refinement (PR, #4) is key to another performance leap. By enforcing memory prototypes to lie within the principal subspace, PR improves the information density and representativeness of the memory, providing the model with high-quality priors of interactive driving experience. Finally, the full DriveMem model achieves the best results (L2 Error: 0.7458, Collision Rate: 0.270).

This ablation study fully validates the collaborative effects of IFA and PMB. IFA delivers clean and stable invariant representations, while PMB builds an optimizable experience bank on top of them. Together, the model learns core driving principles that generalize beyond the training distribution, enabling exceptional performance in OOD challenges.

4.3.2 Effects of Different Memory Sizes

We conduct an ablation study on the impact of memory capacity, with results shown in Table 4. The experiments reveal distinct roles of Ego Memory and Agent Memory in generalization performance. When Ego Memory is expanded from 64 to 128, the model's L2 Error and Collision Rate both decrease, albeit to different extents, with Agent Memory size fixed. This demonstrates that a larger Ego Memory effectively stores a richer repertoire of ego-vehicle driving strategies, improving both prediction accuracy and interaction safety simultaneously. In contrast, expanding Agent Memory (Ego Memory = 128) reveals a clear trade-off between accuracy and safety. Increasing Agent Memory from 1024 to 2048 reduces L2 Error but leads to a higher Collision Rate, whereas further expanding it from 2048 to 4096 lowers the Collision Rate at the cost of increased L2 Error. This pattern suggests that while a larger memory captures a broader range of typical and critical interaction patterns, it may also introduce redundant or less informative samples. Such redundancy can bias retrieval toward less smooth interaction responses. In particular, when both Ego and Agent Memory are scaled to 128 and 2048, respectively, the model achieves the best balance between performance and capacity. These findings strongly indicate that memory module design should not blindly pursue larger capacity but should instead prioritize prototype quality.

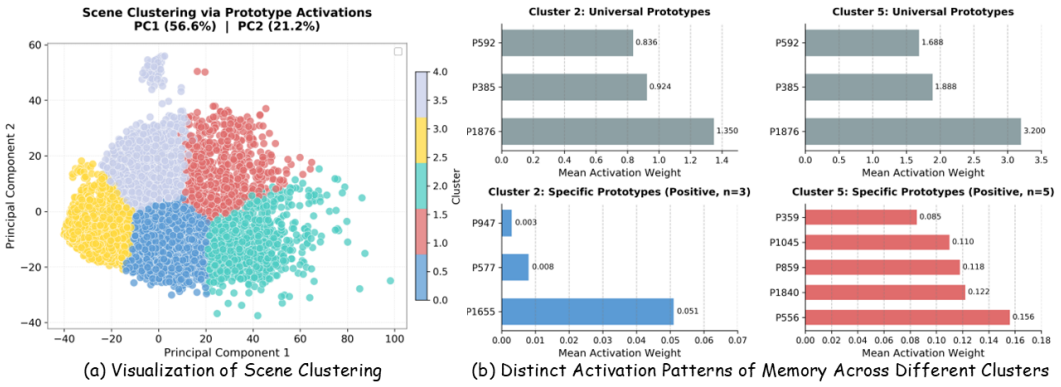

Figure 3: **Prototype activation clustering and functional specialization.** (a) PCA visualization of scene clusters derived from prototype activation weights. Scenes with similar activation patterns exhibit coherent driving semantics, such as routine cruising or obstacle negotiation. (b) Prototype activation statistics in representative clusters. Some prototypes are broadly activated across common scenes, while others are selectively associated with specific interaction contexts, suggesting compositional reuse of learned driving memories.

4.4 Prototype Memory Visualization and Semantic Retrieval

To examine what is encoded in DriveMem, we analyze learned prototype activations from two perspectives: semantic organization and retrieval-based traceability. For each scene, we extract its prototype activation vector and perform K-Means clustering over the test set. As shown in Figure 3(a), scenes assigned to the same cluster tend to share coherent driving semantics, such as routine lane following, and obstacle negotiation. This suggests that DriveMem organizes scenes according to high-level driving structure rather than low-level visual appearance.

We further inspect prototype activation frequencies within representative clusters. Figure 3(b) shows that some prototypes are frequently activated across diverse scenes, behaving as general prototypes that capture common driving priors such as lane following or road layout. In contrast, other prototypes are selectively activated in more specific contexts, such as obstacle avoidance. Rather than assigning a complete semantic meaning to each individual prototype, DriveMem represents a scene through the composition of multiple activated prototypes. This compositional behavior enables flexible reuse of learned memory patterns when encountering new but semantically related driving scenarios.

We next evaluate whether prototype activations support meaningful retrieval. Given a query scene, we retrieve nearest scenes from the memory database using cosine similarity over prototype activation vectors. As shown in Figure 4, retrieved scenes often share similar driving intents, even under different weathers. This indicates that retrieval is guided by invariant driving structure rather than surface-level similarity. Such retrieval traces provide prototype-based interpretability: each prediction can be inspected through representative stored experiences instead of opaque latent activations alone.

These observations are consistent with the memory update strategy. Typicality-driven updates reinforce samples confidently assigned to existing prototypes, consolidating recurring driving patterns, while criticality-driven updates emphasize high-loss samples to improve coverage of difficult interactions. Overall, the visualization results show that DriveMem learns a structured invariant memory: prototype activations organize scenes by driving semantics, support both general and scenario-specific patterns, enable meaningful retrieval under visual shifts, and provide example-level evidence for interpreting model predictions.

5 Related Works

Modern end-to-end driving systems have advanced significantly but remain constrained by fragile out-of-distribution generalization and opaque decision processes. While memory-augmented frameworks and domain-invariant representation learning independently offer pathways to improve robustness and feature stability, existing methods fall short of unifying them: memory retrieval is often dominated

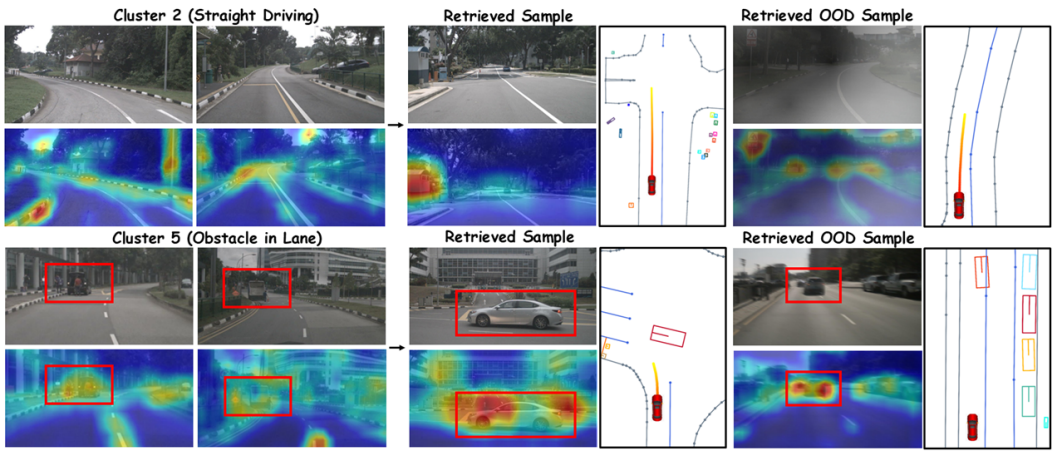

Figure 4: **Qualitative validation of semantic fidelity and retrieval traceability.**

by superficial appearance rather than decision-critical semantics, and invariant features are rarely structured for explicit experience reuse. DriveMem bridges this gap by grounding prototype-based memory in an invariant representation space, enabling reliable, traceable retrieval under distribution shifts. We kindly refer the reader to **Appendix A** for a detailed discussion of related works.

6 Conclusion

In this paper, we introduce DriveMem, a framework that moves end-to-end autonomous driving beyond raw observation fitting by distilling and storing invariant driving memories. Extensive results suggest that structured invariant memory is a promising direction for building more generalizable and interpretable E2E driving systems.

7 Limitations

While DriveMem improves robustness under the distribution shifts studied in this paper, it has several limitations. First, while our Bench2Drive evaluation includes preliminary exploration of behavioral variations across different towns and traffic scenarios, systematic benchmarking of higher-level semantic shifts, such as diverse traffic rule interpretations, complex social interactions, and entirely novel road topologies, remains an important direction for future work. Our current focus has been on establishing robustness to appearance and sensor-level shifts, which form the foundation for real-world deployment. Second, prototype retrieval relies on latent-space proximity, which may diverge from fine-grained semantic alignment under extreme shifts or rare scenarios. When query features fall outside the support region of the learned prototype manifold, retrieval quality can degrade substantially. Future work may incorporate richer semantic priors, dynamic memory expansion, and uncertainty-aware retrieval to address these issues.

Broader Impacts

This work advances robust and generalizable autonomous driving through a memory mechanism grounded in invariant representation learning. By organizing and reusing driving experiences, DriveMem improves generalization to unseen and rare scenarios.

Acknowledgments and Disclosure of Funding

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0122901) and the National Natural Science Foundation of China (No. 62622215 and 62572313). The authors declare no competing interests.

References

[1] John R Anderson. *Cognitive psychology and its implications*. Macmillan, 2005.

[2] Sandra P Marshall. *Schemas in problem solving*. Cambridge University Press, 1995.

[3] Yuping Wang, Shuo Xing, Cui Can, Renjie Li, Hongyuan Hua, Kexin Tian, Zhaobin Mo, Xiangbo Gao, Keshu Wu, Sulong Zhou, Hengxu You, Juntong Peng, Junge Zhang, Zehao Wang, Rui Song, Mingxuan Yan, Walter Zimmer, Xingcheng Zhou, Peiran Li, Zhaohan Lu, Chia-Ju Chen, Yue Huang, Ryan A. Rossi, Lichao Sun, Hongkai Yu, Zhiwen Fan, Frank Hao Yang, Yuhao Kang, Ross Greer, Chenxi Liu, Eun Hak Lee, Xuan Di, Xinyue Ye, Liu Ren, Alois Knoll, Xiaopeng Li, Shuiwang Ji, Masayoshi Tomizuka, Marco Pavone, Tianbao Yang, Jing Du, Ming-Hsuan Yang, Hua Wei, Ziran Wang, Yang Zhou, Jiachen Li, and Zhengzhong Tu. Generative ai for autonomous driving: Frontiers and opportunities, 2025.

[4] Shahin Atakishiyev, Mohammad Salameh, and Randy Goebel. Safety implications of explainable artificial intelligence in end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[5] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7313–7324. PMLR, 18–24 Jul 2021.

[6] Yunqi Wang, Furui Liu, Zhitang Chen, Yik-Chung Wu, Jianye Hao, Guangyong Chen, and Pheng-Ann Heng. Contrastive-ace: Domain generalization through alignment of causal mechanisms. *IEEE Transactions on Image Processing*, 32:235–250, 2023.

[7] Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8046–8056, 2022.

[8] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive Like a Human: Rethinking Autonomous Driving with Large Language Models, July 2023. arXiv:2307.07162 [cs].

[9] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models, February 2024. arXiv:2309.16292 [cs].

[10] Xue Zhao, Qinying Gu, Xinbing Wang, Chenghu Zhou, and Nanyang Ye. Generalizable multi-camera 3d object detection from a single source via fourier cross-view learning. In *Forty-second International Conference on Machine Learning*, 2025.

[11] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

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

[13] Sangrok Lee, Jongseong Bae, and Ha Young Kim. Decompose, adjust, compose: Effective normalization by playing with frequency for domain generalization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11776–11785, 2023.

[14] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling Yu Duan. Uncertainty modeling for out-of-distribution generalization. *arXiv e-prints*, 2022.

[15] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4360, 2022.

[16] Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving, 2025.

[17] Jianbiao Mei, Yukai Ma, Xuemeng Yang, Licheng Wen, Xinyu Cai, Xin Li, Daocheng Fu, Bo Zhang, Pinlong Cai, Min Dou, et al. Continuously learning, adapting, and improving: a dual-process approach to autonomous driving. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 123261–123290, 2024.

[18] Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning, 2025.

[19] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14093–14100. IEEE, 2024.

[20] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

[21] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

[22] Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023.

[23] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

[24] 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*, pages 533–549. Springer, 2022.

[25] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023.

[26] 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.

[27] 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.

[28] 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*, 2025.

[29] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *International Journal of Computer Vision*, pages 1–23, 2024.

[30] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024.

[31] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[32] Xuefeng Jiang, Yuan Ma, Pengxiang Li, Leimeng Xu, Xin Wen, Kun Zhan, Zhongpu Xia, Peng Jia, XianPeng Lang, and Sheng Sun. Transdiffuser: End-to-end trajectory generation with decorrelated multi-modal representation for autonomous driving. *arXiv preprint arXiv:2505.09315*, 2025.

[33] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

[34] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[35] Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning, 2025.

[36] Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, Yining Shi, He Zhe Lim, Li Liu, Tianbao Zhou, Huang Yu, Yifei Hu, Guang Li, Guang Chen, Hao Ye, Lijun Sun, and Diange Yang. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving, 2025.

[37] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2023.

[38] Changjian Shui, Boyu Wang, and Christian Gagné. On the benefits of representation regularization in invariance based domain generalization. *Machine Learning*, 111(3):895–915, 2022.

[39] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[40] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[41] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robobev: Towards robust bird's eye view perception under corruptions, 2023.

[42] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020.

[43] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025.

[44] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[45] 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 Datasets and Benchmarks Track*, 2024.

[46] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Appendix Contents

A. Related Works
- **A.1** End-to-End Autonomous Driving: Evolution and Bottlenecks
- **A.2** Memory-Augmented Autonomous Driving
- **A.3** Domain-Invariant Representation Learning

B. Algorithm

C. Theoretical Analysis

D. Evaluation Protocol

E. More Experimental Results
- **E.1** Test-Time Memory Adaptation
- **E.2** Statistical Consistency Analysis
- **E.3** Complementary Roles of IFA and PMB
- **E.4** Comparison with Alternative Memory Designs

F. Visualization Analysis
- **F.1** Latent Space Evolution
- **F.2** Qualitative Analysis: Robustness and Failure Modes

G. Efficiency Analysis

A Related Works

A.1 End-to-End Autonomous Driving: Evolution and Bottlenecks

The transition from modular pipelines to end-to-end (E2E) autonomous driving has been driven by the availability of large-scale driving datasets and the increasing demand for scalable, jointly optimized autonomy systems [30]. By integrating perception, prediction, and planning into a unified differentiable framework, E2E models reduce hand-crafted interfaces and alleviate error propagation across separately optimized modules. Existing E2E driving methods can be broadly grouped into three representative paradigms:

(1) Modular-style E2E models, such as UniAD [31] and VAD [11], preserve structured intermediate representations while enabling joint optimization. UniAD unifies perception, prediction, and planning within a task-aligned framework, whereas VAD represents driving scenes through vectorized agents and map elements for trajectory prediction and planning.

(2) Generative trajectory models, such as DiffusionDrive [12] and TransDiffuser [32], formulate planning as a generative process and leverage diffusion-based decoding to model multi-modal future trajectories. In particular, DiffusionDrive introduces an efficient transformer-based diffusion decoder that interacts with conditional scene information in a cascaded manner while maintaining real-time efficiency.

(3) Vision-Language Model (VLM) and Vision-Language-Action (VLA) based methods aim to inject semantic reasoning into driving. For example, VLM-based frameworks [33] directly predict trajectories from visual-language representations, while ORION [34], AutoVLA [35], and AgentThink [36] incorporate LLMs or VLMs to enhance high-level decision-making and action generation.

Despite their progress, current E2E systems still face two major limitations that hinder reliable deployment in open-world environments. First, they often exhibit fragile out-of-distribution (OOD) generalization. Since most E2E models are trained from observational correlations, they can become

sensitive to incidental visual or contextual patterns and suffer substantial performance degradation under environmental shifts, rare behaviors, or novel urban layouts [3]. Second, their decision process remains largely opaque. As black-box function approximators, they provide limited traceability for safety-critical decisions, making failure diagnosis, regulatory inspection, and human-AV interaction challenging [4]. These limitations motivate architectures that preserve the scalability of E2E learning while improving robustness and decision traceability under distribution shifts.

A.2 Memory-Augmented Autonomous Driving

To improve robustness and traceability, recent studies have explored memory-augmented driving systems that store and retrieve past driving experiences. A representative line of work uses symbolic or language-based memory. For example, Drive Like a Human [8] introduces natural language reasoning chains for long-tail driving scenarios, while DiLu [9] constructs a structured experience bank through LLM-based self-refinement. These methods provide explicit reasoning traces and improve interpretability at the semantic level. However, they are often weakly grounded in raw visual perception, depend heavily on prompt engineering, and are difficult to directly scale to continuous low-level control.

Another line of work stores prototype-level or scenario-level driving experiences for retrieval. Such memory mechanisms provide a natural way to reuse previous experience when facing similar driving situations. Nevertheless, existing memory representations are often entangled with the statistics of the training domain. When memory is built from raw or weakly regularized features, retrieval may be dominated by superficial appearance similarity rather than stable driving semantics. Under OOD conditions, this can lead to unreliable retrieval: the model may fail to find relevant prototypes due to feature mismatch, or retrieve visually similar but decision-irrelevant experiences.

Therefore, existing memory-augmented driving systems still struggle to simultaneously achieve robust OOD retrieval and traceable decision support. A desirable memory mechanism should not merely store past observations, but should organize driving experiences in a representation space that emphasizes domain-invariant and decision-relevant patterns. This motivates our design of a prototype memory bank built upon invariant driving representations, enabling memory retrieval to be less sensitive to appearance-level shifts and better aligned with recurring interaction patterns.

A.3 Domain-Invariant Representation Learning

Domain-invariant representation learning provides an important foundation for improving model robustness under distribution shifts. Its central goal is to learn representations that preserve task-relevant information while reducing sensitivity to nuisance variations across environments [37, 38]. A foundational formulation is Invariant Risk Minimization (IRM) [39], which encourages a shared optimal predictor across multiple environments to discourage reliance on spurious correlations. Although theoretically appealing, IRM usually requires explicit environment annotations and can be difficult to optimize in large-scale vision systems.

To relax these requirements, practical approaches often pursue invariance through data augmentation, consistency regularization, or distribution alignment. Consistency-based methods [14, 10, 7] encourage stable features or predictions under perturbations that preserve task semantics, thereby reducing sensitivity to nuisance factors such as appearance, lighting, or weather. Another family of methods uses adversarial domain alignment, exemplified by Domain-Adversarial Neural Networks (DANN) [40], which align feature distributions across domains through a domain discriminator. These methods have been widely used to improve robustness in perception and prediction tasks, but they are rarely connected with structured experience memory for E2E autonomous driving.

Our work connects invariant representation learning with memory-augmented E2E driving. Instead of storing raw observations or language-level experiences, DriveMem first abstracts visual features into an invariant representation space and then stores compact prototype-level driving memories. This design enables the model to retrieve experiences based on stable driving semantics rather than domain-specific appearance patterns. As a result, DriveMem complements existing E2E driving models by introducing an invariance-driven memory mechanism that improves OOD robustness while offering prototype-based traceability for downstream trajectory generation.

B Algorithm

The overall training pipeline of DriveMem is outlined in Algorithm 1.

Algorithm 1 Training and Inference Pipeline of DriveMem

Require: Training set D_{train}, test sample x
Ensure: Planned trajectory $\hat{\tau}$

Training
for each batch $(I, \tau) \sim D_{\text{train}}$ **do**
  $\mathbf{Z} \leftarrow \text{Backbone}(I)$
  $(\mathbf{Z}_{\text{inv}}, \mathbf{Z}_{\text{spu}}) \leftarrow \text{IFA}(\mathbf{Z})$
  $\hat{\mathbf{Z}}_{\text{inv}} \leftarrow \text{PMB}(\mathbf{Z}_{\text{inv}})$
  Predict trajectory $\hat{\tau} = f_\theta(\hat{\mathbf{Z}}_{\text{inv}})$
  Compute $\mathcal{L}_{\text{E2E}}, \mathcal{L}_{\text{IFA}}, \mathcal{L}_{\text{PMB}}$
  $\mathcal{L} = \mathcal{L}_{\text{E2E}} + \mathcal{L}_{\text{IFA}} + \mathcal{L}_{\text{PMB}}$
  Update backbone, IFA, PMB and planner.
end for
Inference
$\mathbf{Z} \leftarrow \text{Backbone}(x)$
$\mathbf{Z}_{\text{inv}} \leftarrow \text{IFA}(\mathbf{Z})$
Retrieve prototype memory $\hat{\mathbf{Z}}_{\text{inv}} \leftarrow \text{PMB}(\mathbf{Z}_{\text{inv}})$
Predict $\hat{\tau} = f_\theta(\hat{\mathbf{Z}}_{\text{inv}})$
return $\hat{\tau}$

C Theoretical Analysis: Invariant Prototype Memory and Generalization

Theoretical sketch. We provide an upper-bound analysis to interpret why DriveMem can improve out-of-domain generalization. Let the learned representation be decomposed as $\phi(x) = (u(x), v(x))$, where u denotes decision-relevant invariant factors and v denotes spurious factors. DriveMem predicts through a memory-augmented invariant representation:

$$h_M(x) = g(r_M(u(x))),$$

where $r_M(\cdot)$ is the prototype-memory retrieval operator and g is the downstream prediction head. Under standard boundedness, Lipschitz, and approximate invariant-mechanism assumptions, the target risk of DriveMem admits the following bound:

$$\begin{aligned} R_{D_t}(h_M) \le \widehat{R}_{D_s}(h_M) + 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}} \\ + (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\text{cond}} \\ + 2L_\ell L_g \epsilon_M, \end{aligned} \tag{18}$$

where $W_1(P_s^u, P_t^u)$ measures the discrepancy between source and target invariant representations, λ_{cond} measures the residual conditional mismatch after conditioning on u, and

$$\epsilon_M = \sup_{u \in \mathcal{Z}_{\text{inv}}} \|u - r_M(u)\|_2$$

is the memory approximation error. This decomposition suggests complementary roles of the two key modules. Invariant Feature Abstraction (IFA) can reduce the invariant discrepancy term by enforcing style consistency and suppressing spurious leakage, while the Prototype Memory Bank (PMB) can reduce the approximation and effective complexity terms by replacing unstructured latent features with compact prototype-composed representations. We emphasize that this analysis is intended as an explanatory upper-bound argument rather than a distribution-free guarantee.

C.1 Preliminaries

Let D_s and D_t denote the source and target distributions over input-label pairs (x, y). For any predictor h, the population risk on domain D is

$$R_D(h) = \mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)]. \tag{19}$$

Table 5: Main notation used in the theoretical analysis.

Symbol	Meaning
D_s, D_t	Source and target distributions over (x, y)
$R_D(h)$	Population risk on domain D: $\mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)]$
$\widehat{R}_{D_s}(h)$	Empirical risk of h on n source samples drawn from D_s
n	Number of source training samples
η	Confidence parameter in the generalization bound
$\ell(\cdot, \cdot)$	Task loss function
B	Uniform upper bound of the loss
L_ℓ	Lipschitz constant of the loss with respect to the prediction
L_g	Lipschitz constant of the prediction head g
$\phi(x)$	Learned representation
$u(x)$	Invariant / decision-relevant representation
$v(x)$	Spurious / nuisance representation
$\mathcal{Z}_{\mathrm{inv}}, \mathcal{Z}_{\mathrm{spu}}$	Invariant and spurious representation spaces
$r_M(\cdot)$	Prototype-memory retrieval operator mapping invariant features to memory-composed representations
$g(\cdot)$	Downstream prediction head
$h_M(x)$	DriveMem predictor, $g(r_M(u(x)))$
$h_{\mathrm{inv}}(x)$	Ideal invariant predictor, $g(u(x))$
$\mathcal{H}_M$	Hypothesis class of predictors of the form $x \mapsto g(r_M(u(x)))$
$\widetilde{\mathcal{H}}_M$	Auxiliary surrogate class used in the PMB complexity analysis
$\mathfrak{R}_n(\ell \circ \mathcal{H}_M)$	Empirical Rademacher complexity of the loss-composed class on the source sample
P_s^u, P_t^u	Source and target distributions of the invariant representation $u(x)$
$P_s(y\|u), P_t(y\|u)$	Source and target conditional label mechanisms given u
$W_1(P_s^u, P_t^u)$	Wasserstein-1 discrepancy in invariant space
$\mathrm{TV}(P, Q)$	Total variation distance between distributions P and Q
λ_{cond}	Residual conditional mismatch after conditioning on u
ϵ_M	Memory approximation error: $\sup_{u\in\mathcal{Z}_{\mathrm{inv}}} \|u - r_M(u)\|_2$
ϵ_K	Covering radius induced by K prototypes
$M = \{m_k\}_{k=1}^K$	Set of K prototype vectors in the memory bank
K	Number of prototypes in the memory bank
$\mathcal{A}$	Family of appearance transformations
a	A label-preserving appearance transformation
c	Ideal invariant factor in the leakage model
A	Linear leakage operator from spurious factors to the learned invariant feature
Σ_v	Second-moment matrix / covariance proxy of spurious factors
μ	Positive lower bound such that $\Sigma_v \succeq \mu I$
P_s^c, P_t^c	Source and target distributions of the ideal invariant factor c
P_s^v, P_t^v	Source and target distributions of the spurious factor v
W	Linear prediction matrix in the surrogate PMB model
C_W	Upper bound on $\|W\|_F$ in the surrogate PMB model
R_M	Uniform bound on $\|r_M(u)\|_2$
C_Z	Constant in the covering-number / quantization bound
d_{eff}	Effective dimension of the invariant manifold

Given n source samples, the empirical source risk is denoted by

$$\widehat{R}_{D_s}(h). \tag{20}$$

We decompose the learned representation as

$$\phi(x) = (u(x), v(x)), \tag{21}$$

where $u(x) \in \mathcal{Z}_{\mathrm{inv}}$ denotes invariant, decision-relevant factors and $v(x) \in \mathcal{Z}_{\mathrm{spu}}$ denotes nuisance or spurious factors. DriveMem predicts through a memory-augmented invariant representation:

$$h_M(x) = g(r_M(u(x))), \tag{22}$$

where $r_M(\cdot)$ is the prototype-memory retrieval operator and g is the downstream prediction head. Throughout the analysis, we adopt the convention

$$\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)| \tag{23}$$

for total variation distance.

C.2 Assumptions

Assumption C.1 (Boundedness and Lipschitz continuity)**.** The loss $\ell(\cdot,\cdot)$ is bounded in $[0, B]$ and is L_ℓ-Lipschitz with respect to the prediction. The downstream head g is L_g-Lipschitz with respect to the invariant representation.

Assumption C.2 (Approximate invariant mechanism)**.** The representation admits a decomposition $\phi(x) = (u(x), v(x))$, where u captures the dominant decision-relevant mechanism and v absorbs nuisance variations. Define the residual conditional mismatch:

$$\lambda_{\mathrm{cond}} = \mathbb{E}_{u\sim P_t^u}\Big[\mathrm{TV}(P_t(y|u), P_s(y|u))\Big]. \tag{24}$$

A small λ_{cond} indicates that conditioning on u largely aligns the source and target predictive mechanisms.

Assumption C.3 (Smooth source conditional mechanism)**.** For representations within the support corresponding to the evaluated label-preserving appearance shifts,

$$\mathrm{TV}(P_s(y|u), P_s(y|u')) \le L_y\|u - u'\|_2. \tag{25}$$

This assumption formalizes that small perturbations in the invariant representation induce only limited changes in the source conditional label distribution.

C.3 How IFA Can Reduce an Upper Bound on Invariant Discrepancy

Lemma C.4 (IFA can reduce an upper bound on invariant discrepancy)**.** *Assume the target domain is generated from the source domain by a label-preserving appearance transformation $a \in \mathcal{A}$:*

$$(x_t, y_t) = (a(x_s), y_s). \tag{26}$$

Then

$$W_1(P_s^u, P_t^u) \le \mathbb{E}_{x_s,a}\|u(x_s) - u(a(x_s))\|_2. \tag{27}$$

Thus, minimizing style consistency reduces an upper bound on the invariant discrepancy term.

Furthermore, suppose the invariant feature follows a linear leakage model

$$u = c + Av, \tag{28}$$

where c is the ideal invariant factor, v is the spurious factor, and A measures leakage. If

$$\mathbb{E}[cv^\top] = 0, \qquad \Sigma_v = \mathbb{E}[vv^\top] \succeq \mu I,$$

then

$$\|A\|_F \le \frac{1}{\mu}\|\mathbb{E}[uv^\top]\|_F, \tag{29}$$

and

$$W_1(P_s^u, P_t^u) \le W_1(P_s^c, P_t^c) + \|A\|_{\mathrm{op}} W_1(P_s^v, P_t^v). \tag{30}$$

Hence, under the stated linear-leakage assumptions, a smaller invariant–spurious cross-moment implies a smaller upper bound on the leakage contribution to invariant discrepancy.

Proof. The assumption in Eq. (26) should be viewed as an analytical abstraction for style-like shifts, rather than a literal model for every real-world corruption.

Part 1: Style consistency controls Wasserstein discrepancy. Let P_s^u denote the distribution of $u(x_s)$ and P_t^u denote the distribution of $u(a(x_s))$. Sampling $x_s \sim D_s$ and $a \sim \mathcal{A}$ induces a valid coupling

$$\pi(u, u') = \mathrm{Law}(u(x_s), u(a(x_s))). \tag{31}$$

By the primal definition of Wasserstein-1 distance,

$$W_1(P_s^u, P_t^u) = \inf_{\gamma\in\Pi(P_s^u, P_t^u)} \mathbb{E}_{(u,u')\sim\gamma}\|u - u'\|_2. \tag{32}$$

Evaluating the infimum at the coupling π yields

$$W_1(P_s^u, P_t^u) \le \mathbb{E}_{x_s,a}\|u(x_s) - u(a(x_s))\|_2. \tag{33}$$

Part 2: Cross-moment control under a linear leakage model. Assume $u = c + Av$ with $\mathbb{E}[cv^\top] = 0$. Then

$$\mathbb{E}[uv^\top] = \mathbb{E}[(c + Av)v^\top] = A\Sigma_v. \tag{34}$$

Since $\Sigma_v \succeq \mu I$, we have $\|\Sigma_v^{-1}\|_{\mathrm{op}} \le 1/\mu$, hence

$$A = \mathbb{E}[uv^\top]\Sigma_v^{-1}, \qquad \|A\|_F \le \frac{1}{\mu}\|\mathbb{E}[uv^\top]\|_F. \tag{35}$$

This proves Eq. (29).

Now take any couplings between (c_s, c_t) and (v_s, v_t). The induced coupling between

$$u_s = c_s + Av_s, \qquad u_t = c_t + Av_t$$

satisfies

$$\begin{aligned}\|u_s - u_t\|_2 &= \|c_s - c_t + A(v_s - v_t)\|_2 \\ &\le \|c_s - c_t\|_2 + \|A\|_{\mathrm{op}}\|v_s - v_t\|_2.\end{aligned} \tag{36}$$

Taking expectations and then the infimum over valid couplings yields Eq. (30).

We emphasize that this result is conditional on the stated linear-leakage assumptions and on the magnitude of the cross-moment $\mathbb{E}[uv^\top]$. It does not imply that the sample-wise orthogonality objective directly minimizes the full cross-moment. Rather, it provides a conditional interpretation of how limited invariant–spurious dependence can reduce the leakage contribution to invariant discrepancy. □

C.4 How PMB Controls Approximation Error and Effective Complexity

Lemma C.5 (PMB controls approximation error and can reduce effective complexity under a surrogate model)**.** *Let $M = \{m_k\}_{k=1}^K$ be a prototype set in $\mathcal{Z}_{\mathrm{inv}}$, and define its covering radius*

$$\epsilon_K = \sup_{u \in \mathcal{Z}_{\mathrm{inv}}} \min_{m_k \in M} \|u - m_k\|_2. \tag{37}$$

If the retrieval operator satisfies the uniform approximation property

$$\sup_{u \in \mathcal{Z}_{\mathrm{inv}}} \|u - r_M(u)\|_2 \le \epsilon_K, \tag{38}$$

then for any distribution over (u, y),

$$\left|\mathbb{E}_{(u,y)}[\ell(g(r_M(u)), y)] - \mathbb{E}_{(u,y)}[\ell(g(u), y)]\right| \le L_\ell L_g \epsilon_K. \tag{39}$$

Furthermore, suppose the prediction head is linear over the memory-composed representation, with

$$\|W\|_F \le C_W, \qquad \|r_M(u)\|_2 \le R_M \quad \textit{for all } u.$$

Define the auxiliary surrogate class

$$\widetilde{\mathcal{H}}_M = \{u \mapsto W r_M(u) : \|W\|_F \le C_W\}.$$

Then the empirical Rademacher complexity satisfies

$$\mathfrak{R}_n(\ell \circ \widetilde{\mathcal{H}}_M) \le \frac{L_\ell C_W R_M}{\sqrt{n}}. \tag{40}$$

Proof. This lemma analyzes a generic retrieval operator satisfying Eq. (38). It is intended as an abstraction of the role of PMB rather than an exact characterization of the full gated retrieval used in implementation.

Part 1: Approximation error. For any (u, y),

$$\begin{aligned}|\ell(g(r_M(u)), y) - \ell(g(u), y)| &\le L_\ell \|g(r_M(u)) - g(u)\|_2 \\ &\le L_\ell L_g \|r_M(u) - u\|_2 \\ &\le L_\ell L_g \epsilon_K.\end{aligned} \tag{41}$$

Taking expectation over (u, y) proves Eq. (39).

Part 2: Complexity reduction under a surrogate linear head. Consider

$$\widetilde{\mathcal{H}}_M = \{u \mapsto W r_M(u) : \|W\|_F \le C_W\}.$$

Its empirical Rademacher complexity is

$$\mathfrak{R}_n(\widetilde{\mathcal{H}}_M) = \mathbb{E}_\sigma \left[\sup_{\|W\|_F \le C_W} \frac{1}{n} \sum_{i=1}^n \sigma_i \langle W, r_M(u_i) \rangle \right]. \tag{42}$$

By Frobenius duality,

$$\mathfrak{R}_n(\widetilde{\mathcal{H}}_M) \le \frac{C_W}{n} \mathbb{E}_\sigma \left\| \sum_{i=1}^n \sigma_i r_M(u_i) \right\|_2. \tag{43}$$

By Jensen's inequality,

$$\begin{aligned} \mathbb{E}_\sigma \left\| \sum_{i=1}^n \sigma_i r_M(u_i) \right\|_2 &\le \left(\mathbb{E}_\sigma \left\| \sum_{i=1}^n \sigma_i r_M(u_i) \right\|_2^2 \right)^{1/2} \\ &= \left(\sum_{i=1}^n \|r_M(u_i)\|_2^2 \right)^{1/2} \le \sqrt{n} R_M. \end{aligned} \tag{44}$$

Hence

$$\mathfrak{R}_n(\widetilde{\mathcal{H}}_M) \le \frac{C_W R_M}{\sqrt{n}}. \tag{45}$$

Applying contraction for Lipschitz losses proves Eq. (40). □

C.5 Main Target-Risk Upper Bound

Theorem C.6 (Target-risk upper bound for DriveMem)**.** *Let $\mathcal{H}_M$ be the hypothesis class induced by memory-augmented predictors of the form*

$$h_M(x) = g(r_M(u(x))).$$

Under Assumptions C.1, C.2, and C.3, with probability at least $1 - \eta$ over n source samples,

$$\begin{aligned} R_{D_t}(h_M) \le \widehat{R}_{D_s}(h_M) &+ 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}} \\ &+ (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\text{cond}} \\ &+ 2L_\ell L_g \epsilon_M. \end{aligned} \tag{46}$$

where

$$\epsilon_M = \sup_{u \in \mathcal{Z}_{\text{inv}}} \|u - r_M(u)\|_2 \tag{47}$$

is the memory approximation error.

Moreover, if the prototype set $M = \{m_k\}_{k=1}^K$ forms a K-point cover of a compact invariant manifold with effective dimension d_{eff}, then

$$\epsilon_M \le C_Z \operatorname{diam}(\mathcal{Z}_{\text{inv}}) K^{-1/d_{\text{eff}}}, \tag{48}$$

which yields

$$\begin{aligned} R_{D_t}(h_M) \le \widehat{R}_{D_s}(h_M) &+ 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}} \\ &+ (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\text{cond}} \\ &+ 2L_\ell L_g C_Z \operatorname{diam}(\mathcal{Z}_{\text{inv}}) K^{-1/d_{\text{eff}}}. \end{aligned} \tag{49}$$

Proof. We derive the theorem by combining three ingredients: a standard source-domain generalization bound, a transfer bound in invariant space, and the memory approximation control provided by PMB.

Step 1: Source-domain generalization. By a standard Rademacher complexity bound for bounded losses, with probability at least $1-\eta$,

$$R_{D_s}(h_M) \le \widehat{R}_{D_s}(h_M) + 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}}. \tag{50}$$

Step 2: Compare the memory predictor with the ideal invariant predictor. Define the ideal invariant predictor

$$h_{\mathrm{inv}}(x) = g(u(x)). \tag{51}$$

For any (x, y), by Assumption C.1,

$$\begin{aligned} |\ell(h_M(x), y) - \ell(h_{\mathrm{inv}}(x), y)| &= |\ell(g(r_M(u(x))), y) - \ell(g(u(x)), y)| \\ &\le L_\ell \|g(r_M(u(x))) - g(u(x))\|_2 \\ &\le L_\ell L_g \|r_M(u(x)) - u(x)\|_2 \\ &\le L_\ell L_g \epsilon_M. \end{aligned} \tag{52}$$

Taking expectation over either D_s or D_t gives

$$|R_D(h_M) - R_D(h_{\mathrm{inv}})| \le L_\ell L_g \epsilon_M, \qquad D \in \{D_s, D_t\}. \tag{53}$$

This is exactly the type of approximation control formalized abstractly in Lemma C.5.

Step 3: Transfer bound for the ideal invariant predictor. Let

$$\bar{h}_{\mathrm{inv}}(u) = g(u). \tag{54}$$

Then

$$R_{D_t}(h_{\mathrm{inv}}) = \mathbb{E}_{u\sim P_t^u} \mathbb{E}_{y\sim P_t(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)]. \tag{55}$$

Add and subtract the source conditional expectation:

$$\begin{aligned} R_{D_t}(h_{\mathrm{inv}}) =& \mathbb{E}_{u\sim P_t^u} \mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)] \\ &+ \mathbb{E}_{u\sim P_t^u}\Big(\mathbb{E}_{y\sim P_t(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)] - \mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)]\Big). \end{aligned} \tag{56}$$

Step 4: Bound the conditional mismatch term. For each fixed u, since $\ell \in [0, B]$ and using the convention in Eq. (23),

$$\Big|\mathbb{E}_{y\sim P_t(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)] - \mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)]\Big| \le 2B\,\mathrm{TV}(P_t(y|u), P_s(y|u)). \tag{57}$$

Taking expectation over $u \sim P_t^u$ yields

$$\text{conditional mismatch} \le 2B\lambda_{\mathrm{cond}}. \tag{58}$$

Step 5: Bound the invariant marginal discrepancy term. Define

$$F(u) = \mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u), y)]. \tag{59}$$

For any u, u', adding and subtracting $\mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u'), y)]$ gives

$$\begin{aligned} |F(u) - F(u')| \le& \left|\mathbb{E}_{y\sim P_s(y|u)}\left[\ell(\bar{h}_{\mathrm{inv}}(u), y) - \ell(\bar{h}_{\mathrm{inv}}(u'), y)\right]\right| \\ &+ \left|\mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u'), y)] - \mathbb{E}_{y\sim P_s(y|u')}[\ell(\bar{h}_{\mathrm{inv}}(u'), y)]\right|. \end{aligned} \tag{60}$$

By Assumption C.1, the first term satisfies

$$\left|\mathbb{E}_{y\sim P_s(y|u)}\left[\ell(\bar{h}_{\mathrm{inv}}(u), y) - \ell(\bar{h}_{\mathrm{inv}}(u'), y)\right]\right| \le L_\ell L_g \|u - u'\|_2. \tag{61}$$

For the second term, boundedness of the loss and the total-variation convention in Eq. (23) give

$$\begin{aligned} &\left|\mathbb{E}_{y\sim P_s(y|u)}[\ell(\bar{h}_{\mathrm{inv}}(u'), y)] - \mathbb{E}_{y\sim P_s(y|u')}[\ell(\bar{h}_{\mathrm{inv}}(u'), y)]\right| \\ &\quad\le 2B\,\mathrm{TV}(P_s(y|u), P_s(y|u')) \\ &\quad\le 2BL_y\|u - u'\|_2, \end{aligned} \tag{62}$$

where the last inequality follows from Assumption C.3. Combining Eqs. (61) and (62) yields

$$|F(u) - F(u')| \le (L_\ell L_g + 2BL_y)\|u - u'\|_2. \tag{63}$$

Hence, F is $(L_\ell L_g + 2BL_y)$-Lipschitz. By Kantorovich–Rubinstein duality,

$$\mathbb{E}_{u\sim P_t^u} F(u) - \mathbb{E}_{u\sim P_s^u} F(u) \le (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u). \tag{64}$$

Combining this result with the conditional mismatch bound gives

$$R_{D_t}(h_{\mathrm{inv}}) \le R_{D_s}(h_{\mathrm{inv}}) + (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\mathrm{cond}}. \tag{65}$$

At this point, Lemma C.4 provides two complementary interpretations of the invariant discrepancy term. First, style consistency directly controls an upper bound on $W_1(P_s^u, P_t^u)$ through Eq. (27). Second, under the stated linear-leakage assumptions, a small invariant–spurious cross-moment implies a limited leakage contribution through Eq. (30). The latter is a conditional interpretation and is not claimed to follow directly from the sample-wise orthogonality objective.

Step 6: Transfer back from the ideal invariant predictor to the memory predictor. Using Eq. (53) on target and source domains,

$$R_{D_t}(h_M) \le R_{D_t}(h_{\mathrm{inv}}) + L_\ell L_g \epsilon_M, \tag{66}$$

$$R_{D_s}(h_{\mathrm{inv}}) \le R_{D_s}(h_M) + L_\ell L_g \epsilon_M. \tag{67}$$

Substituting Eq. (65) into Eq. (66), and then using Eq. (67), gives

$$R_{D_t}(h_M) \le R_{D_s}(h_M) + (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\mathrm{cond}} + 2L_\ell L_g \epsilon_M. \tag{68}$$

This is the population transfer inequality, where the IFA-related improvement acts on the discrepancy term and the PMB-related improvement acts on the approximation term.

Step 7: Combine with source generalization. Finally, combining Eq. (68) with the source generalization bound Eq. (50) yields

$$\begin{aligned} R_{D_t}(h_M) \le & \widehat{R}_{D_s}(h_M) + 2\mathfrak{R}_n(\ell \circ \mathcal{H}_M) + B\sqrt{\frac{\log(1/\eta)}{2n}} \\ & + (L_\ell L_g + 2BL_y) W_1(P_s^u, P_t^u) + 2B\lambda_{\mathrm{cond}} + 2L_\ell L_g \epsilon_M, \end{aligned} \tag{69}$$

which proves Eq. (46).

Step 8: Prototype covering specialization. If the prototype memory forms a K-point cover on a compact invariant manifold with effective dimension d_{eff}, then standard covering/quantization arguments give

$$\epsilon_M \le C_Z \,\mathrm{diam}(\mathcal{Z}_{\mathrm{inv}})\, K^{-1/d_{\mathrm{eff}}}. \tag{70}$$

Substituting this into Eq. (46) yields Eq. (49).

We note that Lemma C.5 also provides, under an additional surrogate linear-head assumption, the complexity estimate in Eq. (40). This supports the interpretation that PMB can reduce not only the approximation term but also the effective complexity term. However, the theorem itself only requires the generic quantity $\mathfrak{R}_n(\ell \circ \mathcal{H}_M)$ and does not depend on the surrogate linear-head model. □

Remark. The residual conditional mismatch can also be related to conditional mutual information. We use this only as an interpretive auxiliary perspective, not as the main discrepancy term in the bound.

Overall, Theorem C.6 shows that the target risk is controlled by five terms:

$$\widehat{R}_{D_s}(h_M), \quad \mathfrak{R}_n(\ell \circ \mathcal{H}_M), \quad W_1(P_s^u, P_t^u), \quad \lambda_{\mathrm{cond}}, \quad \epsilon_M.$$

These correspond to source fitting, hypothesis complexity, invariant-domain discrepancy, residual conditional mismatch, and memory approximation error.

This decomposition suggests complementary roles of IFA and PMB. Lemma C.4 shows that perturbation consistency can directly control an upper bound on the invariant discrepancy. It further provides

a conditional interpretation of spurious leakage under the stated linear model, without claiming that the sample-wise orthogonality objective directly minimizes the full invariant–spurious cross-moment. The practical role of orthogonality regularization is instead supported empirically by the component ablation. Lemma C.5 shows that PMB controls the approximation gap induced by prototype retrieval and, under an additional surrogate model, can also reduce effective complexity. The theorem follows by incorporating these mechanisms into a standard source-to-target risk decomposition. Thus, the theory provides one plausible explanation for the empirical gains of DriveMem: it transforms raw, domain-sensitive visual features into structured invariant memory representations whose retrieval behavior is less sensitive to appearance-level shifts.

D Evaluation Protocol

Our evaluation protocol assesses the robustness and generalization capability of DriveMem under three complementary settings: open-loop robustness evaluation, closed-loop planning evaluation, and optional test-time memory adaptation.

Open-Loop Evaluation: nuScenes → nuScenes-C. We first evaluate DriveMem in open-loop trajectory prediction using the nuScenes-C benchmark [41], which applies eight types of out-of-distribution perceptual corruptions, including Camera Crash, Frame Lost, Motion Blur, Fog, and Snow, to the standard nuScenes validation set [42]. These corruptions cover common appearance shifts, weather variations, and sensor degradation that may occur in real-world deployment. Each corruption domain contains three severity levels: easy, medium, and hard. To stress-test OOD robustness, we report results on the *hard* subset across all eight corruption domains.

We use *L2 Error* and *Collision Rate* as the primary open-loop metrics. L2 Error measures the distance between the predicted trajectory and the human-driven ground-truth trajectory, while Collision Rate evaluates whether the predicted trajectory would collide with surrounding agents under the recorded scene annotations.

Closed-Loop Evaluation: NAVSIM navtrain → navhard. We further evaluate closed-loop planning performance on NAVSIM [43, 44], a benchmark based on non-reactive pseudo-simulation. Models are trained on the `navtrain` split and evaluated on `navtest` as the in-distribution setting. For OOD evaluation, we test on the challenging `navhard_two_stage` split, which contains more difficult and shifted driving scenarios. This setting examines whether DriveMem can maintain safe and efficient planning behavior under long-tail conditions.

Following the official NAVSIM protocol, we report *PDMS* for in-distribution evaluation and *EPDMS* for the `navhard_two_stage` split. EPDMS extends the PDM score by combining rule-compliance multipliers and weighted planning quality subscores through a two-stage evaluation procedure, providing a more stringent measure of planning robustness.

Test-Time Memory Adaptation: Bench2Drive. We evaluate optional test-time memory adaptation in interactive CARLA environments using the Bench2Drive benchmark [45]. Models are trained on Town 1-5 and adapted/evaluated on unseen Town 6-15. Specifically, the Town 6-15 data is split into two strictly disjoint subsets: one subset is used as the adaptation set D_{adapt} for updating the memory, and the remaining subset is used as the evaluation set D_{eval}. The two subsets contain no overlapping samples, routes, or scenarios, ensuring that no evaluation data is used during adaptation.

In this setting, D_{adapt} represents a small amount of feedback available in a new environment. In practical deployment, such data can be collected from human-driver takeover or safety-intervention events, where corrective human actions or trajectories naturally provide ground-truth supervision for memory adaptation.

We report closed-loop driving performance using *Success Rate* (SR) and *Driving Score* (DS). SR measures the percentage of routes completed within the time limit, while DS combines route completion with safety penalties, where infractions discount the final score. This protocol evaluates whether DriveMem can incorporate limited new-environment feedback without sample leakage between adaptation and evaluation.

E More Experimental Results

E.1 Test-Time Memory Adaptation

The standard inference mode of DriveMem is retrieval-only: the prototype memory is fixed after training and no test-time update is required. When human corrective feedback is available, DriveMem can be **optionally** extended with test-time memory adaptation, a lightweight, supervised update of prototype memory and coupled layers to rapidly align with new scenarios.

Given disjoint adaptation and evaluation subsets,

$$D_{\text{test}} = D_{\text{adapt}} \cup D_{\text{eval}}, \quad D_{\text{adapt}} \cap D_{\text{eval}} = \emptyset, \tag{71}$$

we update only the memory while keeping the backbone and planner fixed:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \mathcal{L}_{\text{adapt}}(D_{\text{adapt}}; \mathbf{M}) + \lambda_{\text{reg}} \|\mathbf{M} - \mathbf{M}_0\|_F^2. \tag{72}$$

Here $\mathcal{L}_{\text{adapt}}$ can be instantiated as an imitation loss on expert-corrected takeover samples. To avoid global memory drift, each adaptation sample updates only its top-k relevant prototypes:

$$\mathcal{N}_k(x) = \text{TopK}_i \, \text{sim}(\mathbf{u}(x), \mathbf{m}_i), \qquad \mathbf{m}_i \leftarrow \mathbf{m}_i - \eta \nabla_{\mathbf{m}_i} \mathcal{L}_{\text{adapt}}, \quad i \in \mathcal{N}_k(x). \tag{73}$$

This optional setting is evaluated separately from retrieval-only OOD generalization to avoid conflating standard generalization with feedback-based deployment adaptation.

Results of Test-Time Memory Adaptation. Table 6 compares **DriveMem** under three adaptation protocols on Bench2Drive: (1) *BN-only* (update BN statistics only), (2) *Full Finetune* (end-to-end update on limited target data), and (3) *TTMA* (our parameter-efficient memory adaptation).

TTMA achieves the highest Driving Score in both ID (**39.99**) and OOD (**33.07**) settings. Most notably, it yields substantial gains in **Comfortness**: in OOD scenarios, the score rises to **79.08** (+81% over baseline VAD, +15.77 over zero-shot DriveMem), while maintaining robust ID performance (64.57 vs. baseline 47.43). This indicates that memory-augmented adaptation promotes smoother, more conservative driving policies in unseen environments, a critical attribute for safety-critical deployment.

Table 6: Test-Time Memory Adaptation on Bench2Drive.

Method	**Bench2Drive Town 1-5** (ID)				**Bench2Drive Town 6-15** (OOD)			
	DS↑	**SR(%)↑**	**Efficiency↑**	**Comfortness↑**	**DS↑**	**SR(%)↑**	**Efficiency↑**	**Comfortness↑**
VAD [11]	32.57	7.5	133.82	47.43	27.09	0.00	158.08	43.65
+DriveMem	37.22	12.5	130.43	64.81	32.52	3.33	147.50	63.31
+DriveMem (w/ BN-only adaptation)	35.26	7.5	133.64	54.60	30.56	3.33	141.14	74.00
+DriveMem (w/ Full Finetune)	33.55	5.0	129.10	60.80	25.94	0.00	124.66	74.35
+DriveMem (w/ SFT)	39.99	7.5	152.74	64.57	33.07	3.33	137.88	79.08

In contrast, Full Finetune with limited data causes severe performance degradation, confirming the risk of catastrophic forgetting when updating all parameters on small adaptation sets. TTMA avoids this pitfall: despite modest OOD DS gains (+0.55 over zero-shot DriveMem), it preserves ID capability while improving OOD robustness, demonstrating effective balance between generalization and preservation.

This offline study primarily validates the adaptation mechanism under controlled conditions. Future efforts will transition to online closed-loop environments, where dynamic takeover triggers and incremental memory updates will be deployed for real-time, low-latency adaptation.

Discussion. The primary results reported in this paper are obtained under the standard zero-shot inference setting, without TTMA. The evaluations on nuScenes-C, NAVSIM, and the zero-shot Bench2Drive benchmark collectively validate the robustness of DriveMem under diverse distribution shifts. TTMA is evaluated separately as an optional deployment-oriented extension. It leverages labeled corrective feedback to incrementally refine the prototype memory during deployment and demonstrates that the proposed memory framework can be further adapted online when additional supervision is available.

Table 7: Statistical consistency analysis on the nuScenes/nuScenes-C benchmark. All methods are trained using the same five random seeds ({0,1,2,3,4}). We report the performance of each seed together with the mean and standard deviation.

Method	OOD L2 ↓						OOD Collision ↓					
	s0	s1	s2	s3	s4	Mean ± Std	s0	s1	s2	s3	s4	Mean ± Std
DiffusionDrive	0.9080	0.9030	0.9001	0.9102	0.9017	0.9046 ± 0.0043	0.662	0.659	0.647	0.641	0.668	0.6554 ± 0.0111
DAC-SC	0.7773	0.7761	0.7887	0.7746	0.7891	0.7812 ± 0.0071	0.348	0.356	0.346	0.351	0.360	0.3522 ± 0.0058
DriveMem	**0.7458**	**0.7441**	**0.7432**	**0.7501**	**0.7598**	$\mathbf{0.7486 \pm 0.0068}$	**0.270**	**0.277**	**0.300**	**0.298**	**0.278**	$\mathbf{0.2846 \pm 0.0135}$

E.2 Statistical Consistency Analysis

To evaluate the statistical robustness of the proposed method, we conduct multi-seed experiments on the **nuScenes/nuScenes-C** benchmark. All methods are trained using the same five random seeds ({0,1,2,3,4}), ensuring a matched comparison across different random initializations. We compare the direct planning baseline **DiffusionDrive**, the strongest domain-generalization baseline **DAC-SC**, and the proposed **DriveMem**.

Table 7 reports the individual results for each random seed together with the mean and standard deviation. Across all five matched runs, DriveMem consistently achieves the lowest OOD trajectory error and collision rate. Specifically, DriveMem reduces the OOD L2 error from 0.7812 ± 0.0071 (DAC-SC) to 0.7486 ± 0.0068, and the OOD collision rate from 0.3522 ± 0.0058 to 0.2846 ± 0.0135. Similar improvements are also observed over DiffusionDrive.

To further assess whether these improvements are statistically reliable, we perform two-sided paired t-tests across the five matched seeds. Compared with DAC-SC, DriveMem achieves statistically significant improvements in both OOD L2 ($t(4) = -9.313$, $p = 7.40 \times 10^{-4}$) and OOD collision rate ($t(4) = -9.013$, $p = 8.39 \times 10^{-4}$). The improvements over DiffusionDrive are also highly significant ($p = 1.75 \times 10^{-6}$ for OOD L2 and $p = 4.11 \times 10^{-6}$ for collision rate).

These results demonstrate that the performance gains of DriveMem are consistently observed across different random initializations rather than arising from favorable single-run training, providing additional evidence for the robustness and reproducibility of the proposed method.

E.3 Complementary Roles of IFA and PMB

To better understand the respective roles of the proposed Invariant Feature Abstraction (IFA) and Prototype Memory Bank (PMB), we conduct a controlled ablation study by independently enabling each component. The results are summarized in Table 8.

The baseline DiffusionDrive directly predicts trajectories from latent visual features without invariant abstraction or prototype retrieval. Enabling only IFA substantially improves OOD robustness, reducing the OOD L2 error from 0.9080 to 0.7827 and the collision rate from 0.662 to 0.397. This indicates that learning appearance-invariant representations alone already improves robustness under distribution shifts.

By contrast, directly introducing PMB into the original latent feature space without IFA does not improve trajectory accuracy (0.9148 vs. 0.9080), although it reduces the collision rate to 0.431. This suggests that prototype retrieval alone cannot reliably support planning when the query representations remain sensitive to appearance changes, leading to unstable query-prototype matching under OOD conditions.

When IFA and PMB are jointly employed, DriveMem further improves OOD L2 from 0.7827 to 0.7458 and reduces the collision rate from 0.397 to 0.270 compared with IFA-only. These results indicate that IFA and PMB play complementary roles. Rather than serving as an independent robustness module, IFA provides a stable representation space in which prototype retrieval becomes substantially more reliable. Consequently, the full benefit of PMB is realized only when prototype retrieval is performed in the invariant representation space learned by IFA.

Table 8: Complementary roles of IFA and PMB.

Method	IFA	PMB	OOD L2 ↓	Collision ↓
DiffusionDrive	✗	✗	0.9080	0.662
IFA-only	✓	✗	0.7827	0.397
PMB-only	✗	✓	0.9148	0.431
DriveMem	✓	✓	**0.7458**	**0.270**

Table 9: Comparison with alternative memory designs under the same IFA representation space.

Method	OOD L2 ↓	Collision ↓
IFA-only	0.7827	0.397
IFA + Matched-capacity	0.7802	0.412
IFA + Frozen/Random Prototype	0.8215	0.496
IFA + Simpler Memory	0.7790	0.363
IFA + PMB (DriveMem)	**0.7458**	**0.270**

E.4 Comparison with Alternative Memory Designs

To verify that the performance gain originates from the proposed prototype learning strategy rather than simply introducing additional parameters or a generic memory module, we compare PMB with several alternative memory designs while keeping the IFA representation unchanged. The results are shown in Table 9.

The matched-capacity variant replaces PMB with a trainable module of comparable parameter count while preserving the same fusion location. Its performance is nearly identical to IFA-only, indicating that the improvement cannot be explained by additional model capacity alone.

The frozen/random prototype variant keeps the retrieval and fusion pipeline unchanged but uses randomly initialized prototypes that remain fixed throughout training. This substantially degrades both OOD trajectory accuracy and collision rate, demonstrating that unstructured prototype retrieval may even introduce harmful guidance under distribution shifts.

The simpler-memory variant updates prototypes uniformly using all training samples without the proposed typical-critical sample selection or prototype refinement. Although this generic memory mechanism provides a moderate improvement over IFA-only, it remains consistently inferior to the proposed PMB.

Overall, these comparisons demonstrate that the effectiveness of DriveMem does not arise from introducing an arbitrary memory interface or additional parameters. Instead, the combination of representative prototype learning, typical-critical sample selection, and prototype refinement enables PMB to provide substantially more reliable planning guidance within the invariant representation space learned by IFA.

F Visualization Analysis

F.1 Latent Space Evolution

To clearly illustrate the evolution of latent space, we make feature visualization on cross-domain test sets. Figure 5 presents a UMAP-3D [46] comparison of three stages in the latent space. (a) Latent space of baseline model. Features from different domains form distinct clusters, indicating strong domain-specificity and limited generalization. (b) Invariant latent space. Via IFA module, features from diverse domains begin converging toward a shared semantic subspace, partially mitigating domain gaps. (c) DriveMem-enhanced latent space. With PMB further applied, features from all target domains align more tightly with those from the source domain, forming more compact semantic clusters. The comparison reveals that the DriveMem framework induces strong domain invariance: semantically equivalent features across different distributions are mapped into highly proximate regions, maintaining tight alignment with the source distribution. This shows that DriveMem

effectively learns essential semantic representations, significantly improving generalization to unseen domains.

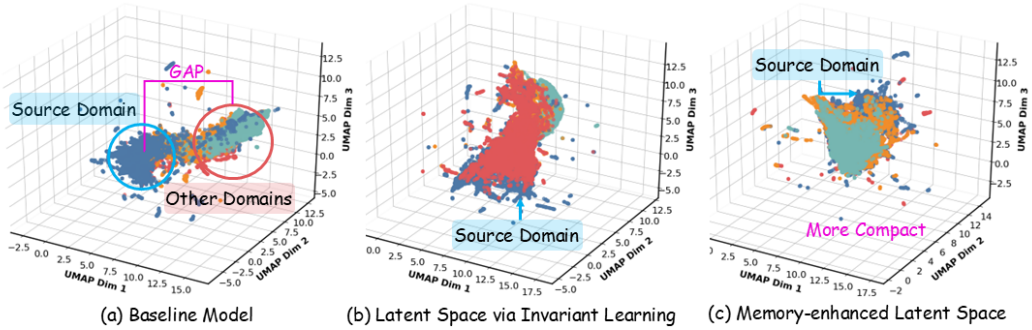

Figure 5: Latent Space Visualization via UMAP-3D.

F.2 Qualitative Analysis: Robustness and Failure Modes

We first evaluate the strengths of DriveMem in challenging perception conditions. As shown in Figure 6, when visual inputs are severely degraded or partially missing, the baseline model relies heavily on instantaneous features and produces rigid trajectories that deviate significantly from the ground truth. In contrast, DriveMem effectively compensates for perceptual gaps by retrieving decision-relevant driving primitives from memory. This mechanism enables the planner to maintain adaptive, human-like path planning that closely follows real-world driving behaviors, demonstrating that our memory-augmented reasoning successfully decouples decision-making from fragile pixel-level dependencies.

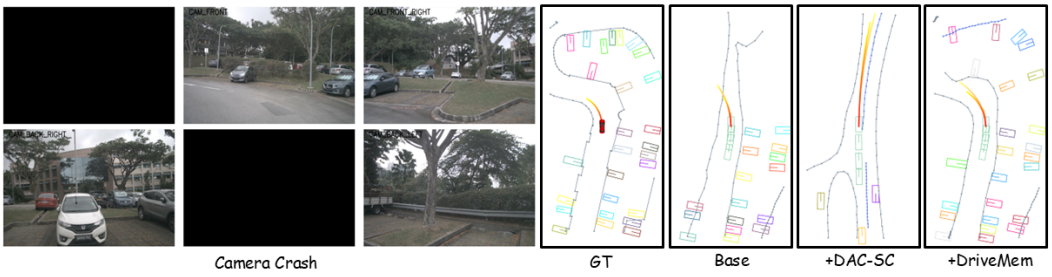

Figure 6: Qualitative comparison under camera degradation. The baseline produces rigid trajectories that deviate from ground truth due to corrupted inputs. DriveMem compensates via memory retrieval, maintaining adaptive and safe path planning consistent with human driving behavior.

Despite these advantages, we observe occasional limitations under extreme environmental conditions, as illustrated in Figure 7. We attribute these failures to degraded memory retrieval under severe distributional shifts. Specifically, heavily corrupted or out-of-distribution inputs induce feature distortion, which compromises the fidelity of query-prototype matching. Consequently, retrieval quality degrades when test samples lie outside the support region of the learned prototype manifold.

Overall, while DriveMem significantly enhances robustness across most challenging scenarios, its performance under extreme shifts naturally depends on the stability of invariant features and prototype alignment fidelity. This behavior reflects an inherent characteristic of memory-driven reasoning frameworks rather than a structural flaw, and points to a clear direction for future work: strengthening invariant feature learning to produce more compact memory representations, thereby boosting retrieval consistency under severe corruptions.

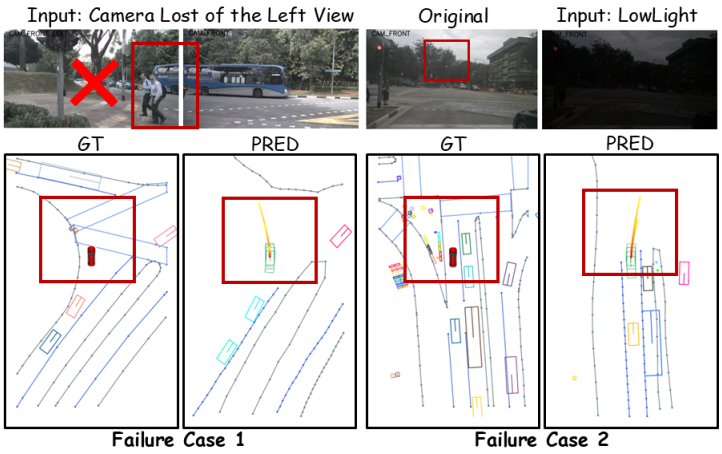

Figure 7: Failure cases under extreme distribution shifts. Performance degrades when severe input corruption pushes query features outside the learned prototype manifold, leading to suboptimal retrieval and trajectory deviation.

G Efficiency Analysis

As summarized in Table 10, DriveMem introduces marginal computational overhead: training iteration time increases modestly from 0.466 to 0.664 s. More importantly, inference latency rises only slightly from 0.175 to 0.188 s, while GPU memory consumption remains nearly identical (+6 MiB). The SVD-based prototype refinement is only used during training and incurs negligible cost (0.007 s/iter).

Table 10: **Efficiency Analysis.** All measurements are conducted on a single NVIDIA RTX 3090 GPU (24GB VRAM) with batch size=1. "E2E Inference" is the end-to-end latency for trajectory prediction. SVD refers to Prototype Refinement, executed offline during memory update with negligible inference overhead.

Model	Train Iter (s)	E2E Inference (s)	GPU Memory (MiB)
DiffusionDrive	0.466 ± 0.024	0.175 ± 0.006	3788
DriveMem (Ours)	0.664 ± 0.012	0.188 ± 0.014	3794
SVD	0.007 ± 0.001	–	–

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction sections offer a comprehensive discussion of the manuscript's context, intuition, and ambitions, as well as its contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed by authors at the end of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For each theoretical result, the paper provides the full set of assumptions and a complete proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The pipeline of the methods and the details of experiments are presented with corresponding reproducible credentials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All utilized data are sourced from open-access platforms. The code will be uploaded to GitHub and made publicly available once this paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The pipeline of the methods and the details of experiments are presented with corresponding reproducible credentials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Key experimental results are obtained by running over different random seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details of experiments are presented with corresponding reproducible credentials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the broader impacts of the paper after the conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any models or datasets that have a high risk for misuse or dual-use. The released code and models are intended solely for academic research purposes and do not pose safety or privacy risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original owners of assets, including data and models, used in the paper, are properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release the code and model weights upon acceptance. The repository includes a comprehensive README detailing the training procedures, dependencies, license, and known limitations of the proposed method.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.