Locating and Repairing Domain Shift in VLM Trajectory Planning

Zhihong Cui[1] **Hengyu Liu**[2*] **Michael Alexander Riegler**[3]
Guandong Xu[4] **Amir Taherkordi**[1] **Tor Skeie**[1,5]

[1]Department of Informatics, University of Oslo, Norway
[2]Department of Computer Science, Aalborg University, Denmark
[3]Simula Metropolitan Center for Digital Engineering (SimulaMet), Norway
[4]The Education University of Hong Kong, Hong Kong, China
[5]Simula Research Laboratory, Norway
{zhihongc, amirhost, tskeie}@ifi.uio.no, heli@cs.aau.dk,
michael@simula.no, gdxu@eduhk.hk

Abstract

Vision-Language Models (VLMs) are increasingly adopted as end-to-end driving planners, yet they suffer from cross-city domain shift. Existing methods treat this shift as a single quantity addressed by uniform alignment or single-site editing. We challenge this view on two counts: ***(i) domain shift in VLM trajectory planners is inherently structured across modality and layer, and (ii) a locate-then-repair recipe matched to this structure outperforms every uniform-objective baseline.*** We analyse domain shift via activation patching and uncover two regularities. **(F1) Modality axis.** Image and text tokens exhibit distinct shift patterns in each VLM. **(F2) Layer axis.** The layer-wise shift partition is architecture-dependent and dataset-stable. These findings recast domain shift as a **two-dimensional tensor** indexed by *modality* (image vs. text tokens) and *layer*, motivating a simple principle: ***locate where shift occurs, and repair only there***. We instantiate this principle in **MoLaRx** (**Mo**dality–**La**yer **R**epair), a locate-then-repair framework. The Causal Importance Score (CIS) estimates the domain shift as a modality–layer tensor and partitions the layers into functional zones. CIS-Guided Selective Repair (CISR) then applies a matched operator (Maximum Mean Discrepancy (MMD) / low-rank adaptation (LoRA) / freezing) to repair each zone, with the composition adapting automatically to each architecture. We evaluate **MoLaRx** on three architecturally distinct VLMs (LFM2-VL-1.6B, Qwen2-VL-2B, and PaliGemma2-3B) across two cross-city benchmarks (nuScenes and Argoverse 2). MoLaRx attains the lowest cross-city trajectory-error gap among adaptation methods in all six main-table settings, while using 50–60% fewer trainable parameters and 8.4–10.9 ms lower inference latency. In contrast, uniform MMD-only adaptation collapses in all six settings, highlighting the importance of structure-guided repair. Code: https://github.com/ZhihongCui/MoLaRx.

1 Introduction

Vision-Language Models (VLMs) are increasingly being introduced for end-to-end trajectory planning [Mao et al., 2023, Tian et al., 2024, Shao et al., 2024, Sima et al., 2024, Sha et al., 2023] as they naturally integrate visual perception with historical trajectory context for decision-making. However, they suffer from severe cross-city domain shift [Naeinian et al., 2026]: as shown in Figure 1 (a), a

*Corresponding author: Hengyu Liu (heli@cs.aau.dk).

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

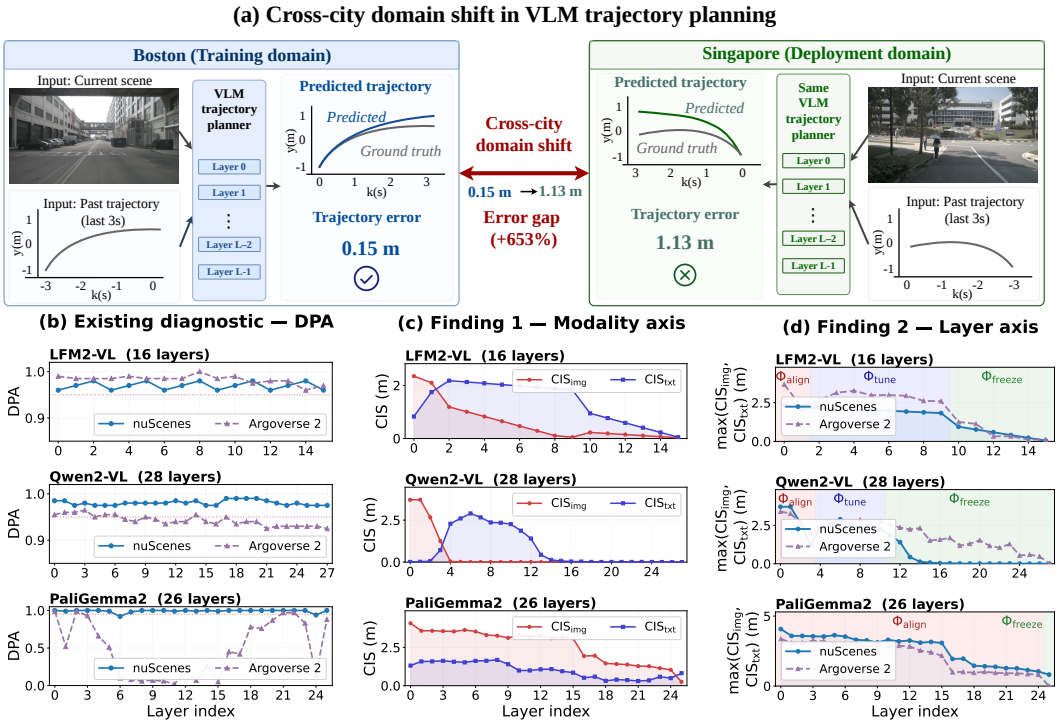

Figure 1: Domain shift in VLM trajectory planners is structured along modality and layer. **(a)** VLMs fine-tuned as trajectory planners suffer cross-city domain shift; **(b)** the existing diagnostic — DPA: saturates near 1 at every layer of every VLM; **(c) F1 (modality axis)**: image and text shifts differ within every VLM; **(d) F2 (layer axis)**: zone partition is architecture-dependent and dataset-stable.

model trained in Boston incurs a $+653\%$ increase in trajectory error when deployed in Singapore (from 0.15 m to 1.13 m). Since collecting and annotating data for every new city is impractical, mitigating this gap is critical for safe and scalable deployment.

Two paradigms in the broader literature shape how such cross-domain gaps are tackled in deep models. The first is **uniform domain adaptation**, a broad class of methods that align source and target distributions, including adversarial training [Ganin et al., 2016], distribution matching [Sun et al., 2016, Gretton et al., 2012], image translation [Zhu et al., 2017, Yang and Soatto, 2020], and parameter-efficient adaptation [Hu et al., 2022]. Despite their differences, these methods apply one common adaptation strategy across layers. The second is **locate-then-edit**, originally developed for single-fact knowledge editing in language models [Meng et al., 2022a] and later extended to multimodal LLMs [Cheng et al., 2023]. ROME [Meng et al., 2022a] uses causal tracing to identify an MLP site and applies a rank-one weight update there; MMEdit [Cheng et al., 2023] benchmarks model editing in multimodal settings. Across both strategies, the same assumption is encoded: domain shift is treated as a **single quantity** that uniform alignment or single-site editing can target. We argue that this scalar assumption is fundamentally misaligned with structured domain shift, as it enforces a single objective across heterogeneous components, inevitably disrupting domain-invariant computation. ***We therefore ask two questions: (Q1) what is the structural form of cross-city domain shift in VLM trajectory planners? and (Q2) given the answer, what repair recipe outperforms uniform alignment?*** We answer empirically: cross-city domain shift is inherently structured along modality and layer axes **(Q1)**, and a locate-then-repair recipe matched to this structure dominates every uniform-objective baseline **(Q2)**.

To establish this empirically, we need a causal diagnostic, but standard tools fall short. Correlational probes such as domain probing accuracy (DPA) [Alain and Bengio, 2016, Belinkov, 2022, Tas and Wagner, 2024] saturate above 95% at every layer of each VLM (Figure 1 (b)), uninformative about causal contribution; attention weights [Jain and Wallace, 2019] are similarly correlational; gradient attribution [Sundararajan et al., 2017] conflates causal effect with gradient magnitude in regression. We therefore adopt activation patching [Meng et al., 2022a, Conmy et al., 2023], an intervention that swaps activations between two forward passes: at each (modality, layer), we replace target-domain

activations with source-domain activations at modality-m token positions and measure the change in predicted trajectory. We call the resulting score the ***Causal Importance Score*** **(CIS)**; its values across (m, l) form a ***modality–layer tensor***. Measurements use fine-tuned VLM planners (Appendix B).

Applying CIS to three architecturally distinct VLMs (LFM2-VL-1.6B [Amini et al., 2025], Qwen2-VL-2B [Wang et al., 2024], and PaliGemma2-3B [Beyer et al., 2024]) across two cross-city benchmarks (nuScenes [Caesar et al., 2020] and Argoverse 2 [Wilson et al., 2023]) reveals two regularities. **(F1) Modality axis (Figure 1 (c)).** Image-token shift dominates the earliest layers in every VLM; text-token shift overtakes it in intermediate layers of LFM2-VL and Qwen2-VL, but never in PaliGemma2. A single uniform alignment cannot target both modalities; this motivates per-modality repair. **(F2) Layer axis (Figure 1 (d)).** Each VLM partitions into 2 or 3 causal zones (Φ_{align}, Φ_{tune}, Φ_{freeze}) with architecture-specific boundaries; within each architecture, the partition is identical (LFM2-VL, PaliGemma2) or differs by at most one layer (Qwen2-VL) on nuScenes and Argoverse 2, making the structure architecture-dependent and dataset-stable. This motivates architecture-adaptive repair.

These findings recast cross-city domain shift as a two-dimensional tensor over modality and layer, naturally motivating a *locate-then-repair* framework, **MoLaRx** (**Mo**dality–**La**yer **R**epair). Specifically, (i) **CIS locates** the (modality, layer) sources of the cross-city gap and partitions layers into functional zones; (ii) **CIS-Guided Selective Repair (CISR) repairs** each zone with a matched operator: distribution alignment (MMD) on Φ_{align}, parameter-efficient adaptation (LoRA) on Φ_{tune}, and freezing on Φ_{freeze}. The operator composition is determined by the discovered partition, adapting automatically to the underlying architecture. The contributions are summarized as follows:

- **Discovery.** We uncover a modality–layer structure of cross-city domain shift: image and text exhibit distinct layer-wise patterns, and the zone partition is architecture-dependent and dataset-stable. To our knowledge, this is the first joint modality–layer characterisation of cross-city shift in VLM trajectory planners.
- **Tool.** We introduce CIS, an activation-patching diagnostic that measures how source-to-target interventions at each (modality, layer) slice causally affect predicted trajectories.
- **Method.** We propose **MoLaRx**, a locate-then-repair framework: CIS locates the modality–layer sources of domain shift, and CISR routes existing adaptation operators (MMD, LoRA, and freezing) to the diagnosed zones, with the composition adapting automatically to each architecture.
- **Experiments.** On three architecturally distinct VLMs across two benchmarks, MoLaRx attains the smallest cross-domain gap among adaptation methods in all six main-table settings while using 50–60% fewer trainable parameters and achieving 8.4–10.9 ms lower inference latency. In contrast, uniform MMD-only adaptation collapses in all six settings, highlighting the importance of structure-guided repair.

2 Related Work

Cross-domain trajectory planning has been studied for end-to-end driving [Filos et al., 2020, Yasarla et al., 2025]; our work draws on two treatment paradigms. (1) **Uniform domain adaptation in trajectory planning** applies one adaptation strategy uniformly across the model. Feature alignment via adversarial losses [Ganin et al., 2016], kernel MMD [Gretton et al., 2012], or correlation alignment [Sun et al., 2016], parameter-efficient adapters [Hu et al., 2022], and test-time normalisation [Wang et al., 2021] are the dominant variants [Oza et al., 2023]. Trajectory-prediction work adds auxiliary perception regularisation [Han et al., 2025], domain-invariant graph encoders (T-GNN [Xu et al., 2022]), or self-training on pseudo-trajectories across geographies (STraj [Zhang et al., 2025]); VLM-based planners (GPT-Driver [Mao et al., 2023], DriveVLM [Tian et al., 2024], LMDrive [Shao et al., 2024], DriveLM [Sima et al., 2024], LanguageMPC [Sha et al., 2023]) are typically evaluated on a single benchmark distribution, which motivates studying cross-city shift within multimodal planning backbones [Naeinian et al., 2026]. (2) **Locate-then-edit** identifies a small set of components causally responsible for a target behaviour and modifies them directly. ROME [Meng et al., 2022a] edits a single causally identified MLP site, while MEMIT [Meng et al., 2022b] distributes knowledge updates across multiple identified MLP layers. MMEdit [Cheng et al., 2023] extends this to multimodal visual-fact editing, and Surgical Fine-Tuning (Surgical FT) [Lee et al., 2022] adapts a manually chosen layer band by shift type. Related approaches also explore selective treatment along other axes: CaDeT [Pourkeshavarz et al., 2024] disentangles causal and spurious latent factors, CRiTIC [Ahmadi et al., 2024] gates inter-agent attention using discovered causal relations, and GALA [Sahoo et al.,

2025] selects which layers to update at test time by gradient alignment. MoLaRx diagnoses cross-city shift jointly by modality and layer through activation interventions, and matches different existing repair operators to the diagnosed zones.

3 Preliminaries

3.1 Definitions

Definition 1 (VLM-based Trajectory Planning)**.** A VLM-based trajectory planner is a function $f_\Theta : \mathcal{X} \to \mathcal{Y}$ that maps a multimodal input $\mathbf{x}$ (image + text) to a future trajectory $\mathbf{Y}$ by passing $\mathbf{x}$ through L stacked backbone layers and decoding the final-layer hidden state:

$$f_\Theta(\mathbf{x}) = \mathbf{Y}(\mathbf{x}) \;=\; \mathrm{Dec}\Big(\mathbf{H}^{(L)}(\mathbf{x})\Big), \quad \mathbf{H}^{(l+1)}(\mathbf{x}) \;:=\; g_l\Big(\mathbf{H}^{(l)}(\mathbf{x});\, \theta_l\Big), \quad \mathbf{H}^{(0)}(\mathbf{x}) \;:=\; \mathrm{Embed}(\mathbf{x}), \tag{1}$$

where f_Θ has L backbone layers parameterised by $\Theta = \{\theta_0, \ldots, \theta_{L-1}\}$; $g_l(\,\cdot\,;\theta_l)$ is the layer-l operator and Dec is the decoder. $\mathbf{x} \in \mathcal{X} = \mathcal{I} \times \mathcal{C}$ is the input from the image space $\mathcal{I}$ and the textual-context space $\mathcal{C}$. Output $\mathbf{Y}(\mathbf{x}) \in \mathcal{Y}$ is a sequence of K ego-vehicle 2D positions $\{(u_k, v_k)\}_{k=1}^K$. $\mathbf{H}^{(l)}(\mathbf{x}) \in \mathbb{R}^{T\times d}$ is the layer-l hidden state; its modality–layer slicing is given in Definition 2.

Definition 2 (Modality–Layer Hidden States)**.** A *modality–layer hidden state* $\mathbf{H}^{(l)}_{\mathcal{M}_m}(\mathbf{x})$ is the row-block of $\mathbf{H}^{(l)}(\mathbf{x})$ at layer $l \in \{0, \ldots, L-1\}$ that collects all tokens of one modality $m \in \{\text{img}, \text{txt}\}$.

$$\mathbf{H}^{(l)}_{\mathcal{M}_m}(\mathbf{x}) \;=\; \big[\,\mathbf{H}^{(l)}_{i,:}(\mathbf{x})\,\big]_{i\in\mathcal{M}_m(\mathbf{x})} \;\in\; \mathbb{R}^{|\mathcal{M}_m(\mathbf{x})|\times d}, \qquad m \in \{\text{img}, \text{txt}\}. \tag{2}$$

where $\mathcal{M}_m(\mathbf{x}) \subseteq [T]$ is the index set of modality-m token positions: $\mathcal{M}_{\text{img}}(\mathbf{x}) = \{i \in [T] : \mathrm{token}_i \in \mathcal{V}_{\text{img}}\}$ collects the image-token positions, with $\mathcal{V}_{\text{img}}$ the image-token vocabulary (bracketed by the $\langle\text{img_start}\rangle$, $\langle\text{img_end}\rangle$ delimiters), and $\mathcal{M}_{\text{txt}}(\mathbf{x}) = [T] \setminus \mathcal{M}_{\text{img}}(\mathbf{x})$ collects the text-token positions.

Definition 3 (Paired Cross-Domain Distribution)**.** Let $\mathcal{S}_s = \{(\mathbf{x}_s^i, \mathbf{Y}_s^{*i})\}_{i=1}^{N_s}$ be a labelled source dataset and $\mathcal{S}_t = \{\mathbf{x}_t^j\}_{j=1}^{N_t}$ an unlabelled target dataset. The paired cross-domain distribution $\mathcal{P}_{\text{paired}}$ is the uniform distribution over scene-matched pairs $(\mathbf{x}_s, \mathbf{x}_t)$, where each target input $\mathbf{x}_t \in \mathcal{S}_t$ is matched with its source nearest neighbour $\mathbf{x}_s \in \mathcal{S}_s$ under a task-relevant content vector $\mathbf{k}(\cdot) \in \mathbb{R}^3$:

$$\mathcal{P}_{\text{paired}} \;=\; \mathrm{Uniform}\Big(\,\big\{\,(\mathbf{x}_s, \mathbf{x}_t) \;:\; \mathbf{x}_t \in \mathcal{S}_t,\;\; \mathbf{x}_s = \arg\min_{\mathbf{x}\in\mathcal{S}_s} \big\|\mathbf{k}(\mathbf{x}) - \mathbf{k}(\mathbf{x}_t)\big\|_2 \,\big\}\,\Big). \tag{3}$$

Each pair represents the same driving scenario in two cities, isolating domain shift from content shift.

Definition 4 (Domain Shift Gap)**.** The *domain shift gap* $G(f)$ measures how much f's predicted trajectory differs between the source and target inputs of a paired sample. Concretely, given a scene-matched pair $(\mathbf{x}_s, \mathbf{x}_t) \sim \mathcal{P}_{\text{paired}}$, f is run on each input to produce two predicted trajectories, $f(\mathbf{x}_s)$ and $f(\mathbf{x}_t)$. $G(f)$ is the averaged ℓ_2 distance between the two trajectories over the K waypoints of all scene-matched pairs in $\mathcal{P}_{\text{paired}}$:

$$G(f) \;=\; \mathbb{E}_{(\mathbf{x}_s,\,\mathbf{x}_t)\sim\mathcal{P}_{\text{paired}}}\left[\;\frac{1}{K}\sum_{k=1}^{K}\big\|f(\mathbf{x}_t)_k - f(\mathbf{x}_s)_k\big\|_2\;\right] \;\in\; \mathbb{R}_{\geq 0}, \tag{4}$$

where $f(\mathbf{x})_k = (u_k, v_k) \in \mathbb{R}^2$ is the k-th predicted waypoint of f on input $\mathbf{x}$. $G(f) = 0$ means f predicts the same trajectory on both inputs of every pair.

Definition 5 (Modality–Layer Tensor)**.** The *modality–layer tensor* $\mathbf{\Psi}$ tells us which (m, l) slices of the planner's hidden state are responsible for the cross-domain gap: each entry $\mathbf{\Psi}[m, l]$ measures how much slice (m, l) contributes to the gap. Formally, $\mathbf{\Psi} \in \mathbb{R}^{2\times L}_{\geq 0}$ is a non-negative $2 \times L$ grid with rows indexed by modality $m \in \{\text{img}, \text{txt}\}$ and columns by layer $l \in \{0, \ldots, L-1\}$. The (m, l)-entry is the expected per-waypoint ℓ_2 displacement of $f_\Theta(\mathbf{x}_t)$ when the modality-m block at layer l is replaced with the source counterpart from a scene-matched pair $(\mathbf{x}_s, \mathbf{x}_t) \sim \mathcal{P}_{\text{paired}}$:

$$\mathbf{\Psi}[m, l] \;=\; \mathbb{E}_{(\mathbf{x}_s,\,\mathbf{x}_t)\sim\mathcal{P}_{\text{paired}}}\left[\;\frac{1}{K}\sum_{k=1}^{K}\Big\|f_\Theta\big(\mathtt{do}(\mathbf{H}^{(l)}_{\mathcal{M}_m}(\mathbf{x}_t) \leftarrow \mathbf{H}^{(l)}_{\mathcal{M}_m}(\mathbf{x}_s));\, \mathbf{x}_t\big)_k - f_\Theta(\mathbf{x}_t)_k\Big\|_2\;\right], \tag{5}$$

where $\mathtt{do}(\cdot)$ overwrites the modality-m rows of $\mathbf{H}^{(l)}(\mathbf{x}_t)$ with those of $\mathbf{H}^{(l)}(\mathbf{x}_s)$ and the forward pass continues from layer $l{+}1$. A large $\mathbf{\Psi}[m, l]$ marks the slice as a causal driver of the cross-domain gap; a near-zero entry marks it as domain-invariant. The empirical estimator is given in §4.1.

3.2 Problem Statement

Our task is to **locate** and **repair** the domain shift gap G of a VLM trajectory planner. Concretely, given a source-trained planner $f_{\Theta'}$, we first use the modality–layer tensor $\mathbf{\Psi}$ to derive a partition $\Phi = (\Phi_{\text{align}}, \Phi_{\text{tune}}, \Phi_{\text{freeze}})$ of the L backbone layers into disjoint zones (the concrete derivation is the CIS estimator in §4.1). Each per-layer update Δ_l takes a zone-specific form determined by Φ:

$$\Delta_l \;=\; \begin{cases} \delta\theta_l & \text{if } l \in \Phi_{\text{align}} \quad \text{(rank-}r\text{ LoRA, attention only)}, \\ \mathbf{B}_l\mathbf{A}_l & \text{if } l \in \Phi_{\text{tune}} \quad \text{(rank-}r\text{ LoRA, all projections)}, \\ \mathbf{0} & \text{if } l \in \Phi_{\text{freeze}} \quad \text{(frozen)}. \end{cases} \tag{6}$$

We then seek $\Delta = \{\Delta_l\}_{l=0}^{L-1}$ that minimises G subject to bounded source-side regression:

$$\Delta^* \;=\; \arg\min_{\Delta} \; G(f_{\Theta'+\Delta}) \quad \text{s.t.} \quad E_s(f_{\Theta'+\Delta}) \;\le\; E_s(f_{\Theta'}) + \epsilon, \tag{7}$$

so $f_{\Theta'+\Delta}$ is the adapted planner. Here, $f_{\Theta'}$ denotes the VLM planner after fine-tuning (Appendix B); $\delta\theta_l$ is an additive update on θ_l and $\mathbf{B}_l\mathbf{A}_l$ is a rank-r adapter (concrete operators in §4.2); $E_s(f) := \mathbb{E}_{\mathcal{S}_s}[\mathcal{E}(f(\mathbf{x}), \mathbf{Y}^*)]$ is the source-side average per-sample trajectory error; and $\epsilon \ge 0$ bounds how much the adapted planner's source-side error may grow over $f_{\Theta'}$'s.

4 Method

MoLaRx solves Eq. 7 via two modules (Figure 2): **CIS** (§4.1) locates the (m, l) sources of the gap and partitions layers into zones Φ; **CISR** (§4.2) repairs each zone with a matched operator (MMD / LoRA / freeze).

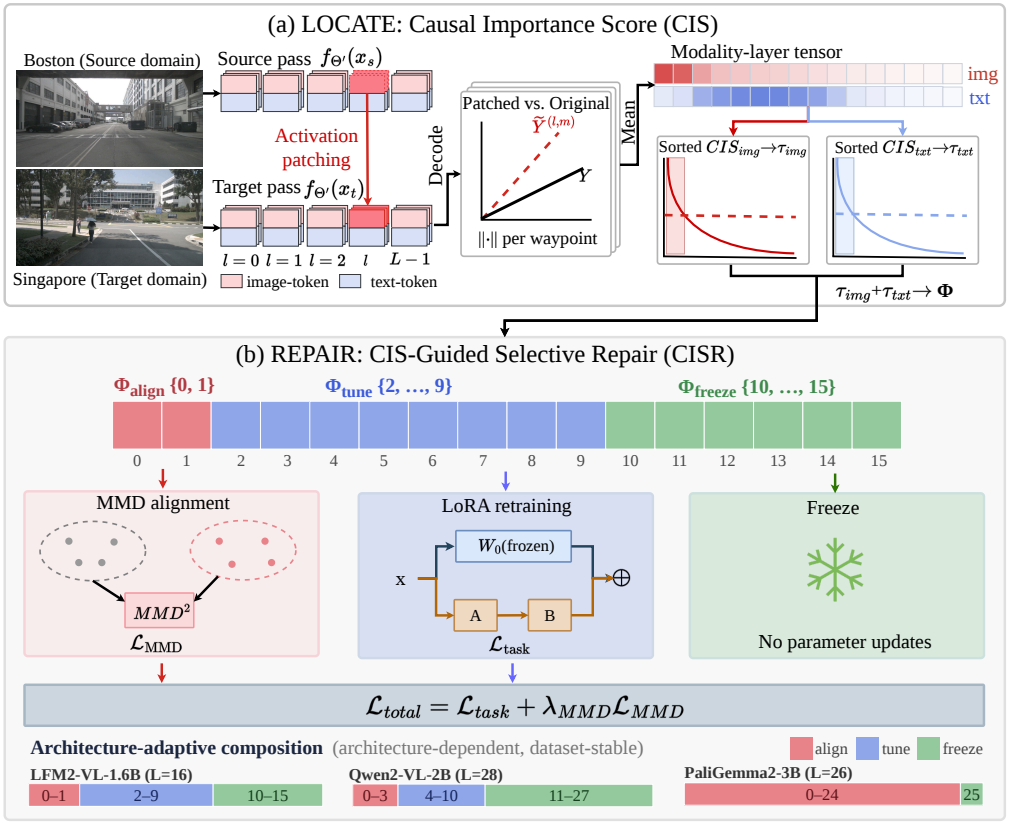

Figure 2: **MoLaRx overview.** CIS locates cross-city shift by modality and layer; CISR applies matched repair operators to the diagnosed zones.

4.1 Causal Importance Score (CIS)

We locate which (m, l) slices of the VLM drive the domain shift gap via activation patching [Meng et al., 2022a, Conmy et al., 2023]: for each (m, l), a Monte Carlo average over N=100 scene-

matched pairs yields one entry of the empirical tensor $\widehat{\mathbf{\Psi}}$, namely the CIS of slice (m, l) given by Eq. 9. Hoeffding's inequality (Eq. 10) bounds its deviation from the population value $\mathbf{\Psi}[m, l]$ (Eq. 5). Thresholding $\widehat{\mathbf{\Psi}}$ at $\tau_{\text{img}}, \tau_{\text{txt}}$ then partitions the L layers into zones Φ (Eqs. 11 to 13).

Computing one entry $\widehat{\mathbf{\Psi}}[m, l]$. For each pair $(\mathbf{x}_s^i, \mathbf{x}_t^i) \sim \mathcal{P}_{\text{paired}}$ we do the following four steps:

- **(a)** *Forward both inputs.* At layer l, run $f_{\Theta'}$ on $\mathbf{x}_s^i$ and $\mathbf{x}_t^i$ to obtain the source and target hidden states $\mathbf{H}^{(l)}(\mathbf{x}_s^i)$, $\mathbf{H}^{(l)}(\mathbf{x}_t^i) \in \mathbb{R}^{T \times d}$.
- **(b)** *Patch the modality-m block at layer l on the target.* Read off the modality-m token positions $\mathcal{M}_m(\mathbf{x}_t^i) \subseteq [T]$ in the target (image-patch positions when m=img; text-token positions when m=txt), and overwrite *only* those rows of $\mathbf{H}^{(l)}(\mathbf{x}_t^i)$ with the corresponding rows of $\mathbf{H}^{(l)}(\mathbf{x}_s^i)$, leaving the other modality's rows untouched.
- **(c)** *Resume the forward pass.* Continue from layer $l{+}1$ through the rest of the decoder on the patched hidden state, yielding the counterfactual trajectory $\widetilde{\mathbf{Y}}^{(l,m)}(\mathbf{x}_s^i, \mathbf{x}_t^i)$.
- **(d)** *Per-pair displacement.* Compare $\widetilde{\mathbf{Y}}^{(l,m)}$ to the un-patched target prediction $f_{\Theta'}(\mathbf{x}_t^i)$ by the per-waypoint ℓ_2 distance averaged over the $K{=}6$ waypoints (in metres),

$$d_i^{(l,m)} \;=\; \tfrac{1}{K} \sum_{k=1}^{K} \big\| \widetilde{\mathbf{Y}}^{(l,m)}(\mathbf{x}_s^i, \mathbf{x}_t^i)_k - f_{\Theta'}(\mathbf{x}_t^i)_k \big\|_2. \tag{8}$$

Averaging the per-pair displacements $d_i^{(l,m)}$ over the N pairs gives the unbiased Monte Carlo estimator of $\mathbf{\Psi}[m, l]$:

$$\widehat{\mathbf{\Psi}}[m, l] \;=\; \frac{1}{N} \sum_{i=1}^{N} d_i^{(l,m)} \;=\; \frac{1}{N} \sum_{i=1}^{N} \frac{1}{K} \sum_{k=1}^{K} \big\| \widetilde{\mathbf{Y}}^{(l,m)}(\mathbf{x}_s^i, \mathbf{x}_t^i)_k - f_{\Theta'}(\mathbf{x}_t^i)_k \big\|_2. \tag{9}$$

Each summand $d_i^{(l,m)}$ is bounded above by the maximum per-waypoint distance R between any two predicted trajectories, so Hoeffding's inequality [Hoeffding, 1963] gives

$$\Pr\Big[\big| \widehat{\mathbf{\Psi}}[m, l] - \mathbf{\Psi}[m, l] \big| > \varepsilon \Big] \;\le\; 2 \exp\!\left(-\frac{2N\varepsilon^2}{R^2} \right), \tag{10}$$

where ε is the deviation tolerance. Setting the failure probability to $\eta{=}0.05$ and inverting Eq. 10 for our values $N{=}100$ and $R{\approx}20$ m yields a 95% half-width of $\varepsilon \approx 0.6$ m on each entry, well below the 0 to 2.3 m spread of $\widehat{\mathbf{\Psi}}$ values across zones, so the partition below is statistically reliable (empirical N-budget sweep validating this choice in Appendix C.2.4).

From tensor to partition. Repeating this procedure for every $(m, l) \in \{\text{img}, \text{txt}\} \times \{0, \ldots, L{-}1\}$ produces the tensor $\widehat{\mathbf{\Psi}} \in \mathbb{R}_{\ge 0}^{2 \times L}$, whose two rows record the per-modality causal profile across depth. To turn this profile into a repair plan, we partition the L layers by which modality dominates: layers where the image entry is large form the alignment zone Φ_{align}, those where the text entry is large form the fine-tuning zone Φ_{tune}, and the rest form the freeze zone Φ_{freeze}:

$$\Phi_{\text{align}} \;=\; \{\, l \in \{0, \ldots, L{-}1\} \,:\, \widehat{\mathbf{\Psi}}[\text{img}, l] > \tau_{\text{img}} \,\}, \tag{11}$$

$$\Phi_{\text{tune}} \;=\; \{\, l \,:\, \widehat{\mathbf{\Psi}}[\text{txt}, l] > \tau_{\text{txt}} \,\} \setminus \Phi_{\text{align}}, \tag{12}$$

$$\Phi_{\text{freeze}} \;=\; \{0, \ldots, L{-}1\} \setminus (\Phi_{\text{align}} \cup \Phi_{\text{tune}}), \tag{13}$$

where $\tau_{\text{img}}, \tau_{\text{txt}}$ are per-modality thresholds taken as the elbow of the sorted CIS values (sensitivity in Appendix C.2.1), and $\setminus$ denotes set difference.

4.2 CISR

Given the CIS-induced partition $\Phi = (\Phi_{\text{align}}, \Phi_{\text{tune}}, \Phi_{\text{freeze}})$, CISR assigns a matched operator to each zone (MMD on Φ_{align}, LoRA on Φ_{tune}, freeze on Φ_{freeze}) and learns the parameter updates Δ.

Operator on Φ_{align}: multi-scale MMD alignment. For each $l \in \Phi_{\text{align}}$ we minimise the Maximum Mean Discrepancy [Gretton et al., 2012] between source and target image-token mean-pooled representations $\bar{\mathbf{H}}_{\text{img}}^{(l)}(\mathcal{B}; \Theta')$ (full definition in Appendix D.1) on paired minibatches $\mathcal{B}_s, \mathcal{B}_t$ of size

B. Summing over the zone gives the image-alignment loss:

$$\mathcal{L}_{\text{MMD}}(\Theta'; \mathcal{B}_s, \mathcal{B}_t) \;=\; \frac{1}{|\Phi_{\text{align}}|} \sum_{l \in \Phi_{\text{align}}} \text{MMD}^2\Big(\bar{\mathbf{H}}^{(l)}_{\text{img}}(\mathcal{B}_s; \Theta'),\, \bar{\mathbf{H}}^{(l)}_{\text{img}}(\mathcal{B}_t; \Theta');\, k\Big)\,, \tag{14}$$

where k is the multi-scale RBF kernel and $|\Phi_{\text{align}}|$ normalises across the zone. On each $l \in \Phi_{\text{align}}$ we attach rank-r LoRA adapters to the attention projections only (Q, K, V, O) — the cross-modal mixing locus implicated by F1 — driven by $\mathcal{L}_{\text{MMD}}$, instantiating $\delta\theta_l$ of Eq. 6.

Operator on Φ_{tune}: selective LoRA retraining. For each $l \in \Phi_{\text{tune}}$ we attach rank-r=16 LoRA adapters [Hu et al., 2022] (α=32) to every linear projection (factorisation in Appendix D.2). The adapters are driven by the standard supervised trajectory loss on the labelled source dataset:

$$\mathcal{L}_{\text{task}}(\Theta', \Delta) \;=\; -\frac{1}{|\mathcal{S}_s|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{S}_s} \sum_{t=1}^{T_y} \log p_{\Theta'+\Delta}(y_t \mid \mathbf{x},\, y_{<t})\,, \tag{15}$$

where T_y is the trajectory token length, $y_{<t}$ the previously decoded tokens, and $\{\mathbf{A}_l, \mathbf{B}_l\}_{l \in \Phi_{\text{tune}}}$ are the rank-r LoRA factors that form the Φ_{tune} component of Δ (see Eq. 6).

Operator on Φ_{freeze}: zero gradient. For every $l \in \Phi_{\text{freeze}}$ we attach no adapter, so $\Delta_l = \mathbf{0}$.

Training Loss. Combining task supervision (LoRA on Φ_{tune}) and MMD alignment (on Φ_{align}), with a zero-gradient constraint on Φ_{freeze}:

$$\mathcal{L}_{\text{total}}(\Theta', \Delta) \;=\; \underbrace{\mathcal{L}_{\text{task}}(\Theta', \Delta)}_{\text{updates LoRA on } \Phi_{\text{tune}}} \;+ \lambda_{\text{MMD}} \cdot \underbrace{\mathcal{L}_{\text{MMD}}(\Theta'; \mathcal{B}_s, \mathcal{B}_t)}_{\text{updates attention LoRA on } \Phi_{\text{align}}}\,, \tag{16}$$

where λ_{MMD} trades off task supervision against domain alignment (λ_{MMD}=0.5, sensitivity in Appendix C.2.2). Gradients are routed by zone (Appendix D.3); target-domain samples enter only the kernel term of $\mathcal{L}_{\text{MMD}}$, so CISR is a valid unsupervised DA method. When $\Phi_{\text{tune}} = \varnothing$ (such as PaliGemma2), the LoRA operator has nothing to attach to and Eq. 16 automatically degenerates to MMD-on-Φ_{align} + freeze-on-Φ_{freeze}. The full pseudocode is in Appendix D.4, Algorithm 1.

5 Experiments

5.1 Experimental Setting

Datasets. We evaluate on two cross-city trajectory-planning benchmarks under transductive unsupervised domain adaptation, using target-test inputs without their trajectory labels for adaptation. On ***nuScenes*** [Caesar et al., 2020] we use Boston-Seaport (BS) as source and Singapore (SG) as target, with a stratified split yielding $\sim$26.6k train, 3,569 source test, and 3,500 target test samples. On ***Argoverse 2 Sensor (AV2)*** [Wilson et al., 2023] we use Pittsburgh (PIT) as source and Miami (MIA) as target, extracting 858 train, 284 ID-test, and 1,393 OOD-test samples from 57 validation logs.

Models. We evaluate three VLMs spanning the backbone-operator and decoder-attention axes. ***LFM2-VL-1.6B*** [Amini et al., 2025] is a 16-layer hybrid backbone with a causal decoder; ***Qwen2-VL-2B*** [Wang et al., 2024] is a 28-layer pure-Transformer with a causal decoder; ***PaliGemma2-3B*** [Beyer et al., 2024] pairs a SigLIP encoder with a 26-layer Gemma2 prefix-LM decoder using bidirectional image attention.

Evaluation Metrics. We report four metrics: **ADE** ($\downarrow$, m): mean per-waypoint ℓ_2 error; **FDE** ($\downarrow$, m): error at the final waypoint; $\Delta_{\textbf{ADE}}$ ($\downarrow$, m): cross-domain gap ADE($\mathcal{S}_t$) $-$ ADE($\mathcal{S}_s$); and **PE** ($\downarrow$, ms): inference latency. The numbers are averaged over five random seeds.

Baselines. We compare against nine baselines in four categories: (1) *no-adaptation reference*: **Source-only**, the fine-tuned planner with no cross-domain repair, establishing the gap floor against all DA methods; (2) *uniform / scalar-view DA*, four representatives of the "align-everything" view we argue against: **Full LoRA** [Hu et al., 2022] (parameter-efficient fine-tuning (PEFT) only, no DA signal), **Full LoRA + MMD** [Gretton et al., 2012] (PEFT plus uniform kernel matching), **DANN** [Ganin et al., 2016] (adversarial domain-invariant alignment), and **MMD-only** [Gretton et al., 2012] (alignment without task fine-tuning); (3) *layer-selective, non-causal DA* (**LLRD** [Howard and Ruder, 2018]; **Freeze-Last-K** [Kumar et al., 2022]), choosing layers via correlational or heuristic criteria; and (4)

locate-then-edit family (**ROME*** [Meng et al., 2022a]; **Surgical FT** [Lee et al., 2022]), applying single-locus or hand-picked-band repair. All adapted baselines share the same input format, backbone, and LoRA rank. See Appendix A.1–A.4 for full details of the above settings. An inductive-protocol control is reported in App. C.4.

5.2 Main Performance Evaluation

Performance evaluation results are reported in Table 1; qualitative case studies are found in Appendix C.1; computational efficiency is in Appendix C.3, Table 10.

Table 1: Performance across 2 datasets of 3 VLMs on four metrics. **Bold** marks the best result per row *among the adaptation methods*; underline marks the runner-up. Source-only is reported as the no-adaptation reference and is excluded from the bold/underline ranking. The rightmost column Δ is MoLaRx's lead over the underlined runner-up. The marker * flags margins exceeding the 2σ significance threshold. MMD-only is marked with ‡ because the model has collapsed.

Data-set	VLM	Metric	Source-only	F-LoRA	F-LoRA + MMD	DANN	MMD only	LLRD	Freeze-K	ROME*	Surg-FT	MoLaRx	Δ
nuScenes (BS→SG)	LFM2	ADE (↓, m)	4.510	4.275	4.284	4.295	9.657	4.288	4.310	4.270	4.266	**4.251**	0.015
		FDE (↓, m)	7.890	7.480	7.531	7.590	16.542	7.475	7.510	7.470	7.462	**7.422**	0.040
		Δ_{ADE} (↓, m)	2.550	3.340	3.328	3.390	1.646‡	3.358	3.315	3.255	3.205	**3.150**	0.055*
		PE (↓, ms)	68.3	66.5	68.4	67.5	48.3‡	66.5	65.5	65.5	58.3	**47.4**	10.9*
	Qwen2	ADE (↓, m)	5.020	4.730	4.760	4.745	10.156	4.715	4.725	4.720	4.700	**4.634**	0.066*
		FDE (↓, m)	8.760	8.430	8.470	8.450	18.130	8.426	8.422	8.414	8.380	**8.242**	0.138*
		Δ_{ADE} (↓, m)	2.850	2.965	2.985	2.975	1.624‡	2.964	2.960	2.952	2.935	**2.906**	0.029
		PE (↓, ms)	71.2	79.8	71.5	70.8	51.2‡	79.8	78.8	68.8	61.2	**50.5**	10.7*
	PaliGemma2	ADE (↓, m)	4.450	4.620	4.650	4.635	10.054	4.530	4.495	4.345	4.320	**4.190**	0.130*
		FDE (↓, m)	7.620	7.980	8.020	8.000	17.498	7.840	7.795	7.530	7.460	**7.355**	0.105*
		Δ_{ADE} (↓, m)	2.630	2.580	2.610	2.595	1.418‡	2.500	2.485	2.405	2.385	**2.337**	0.048
		PE (↓, ms)	82.7	81.4	83.2	82.3	62.7‡	81.4	90.5	72.7	70.5	**62.1**	8.4*
Argoverse 2 (PIT→MIA)	LFM2	ADE (↓, m)	5.120	4.770	4.800	4.785	10.270	4.685	4.660	4.510	4.485	**4.420**	0.065*
		FDE (↓, m)	8.650	8.250	8.290	8.270	17.587	8.110	8.065	7.815	7.760	**7.680**	0.080*
		Δ_{ADE} (↓, m)	3.252	2.150	2.180	2.165	1.165‡	2.070	2.050	1.965	1.945	**1.880**	0.065*
		PE (↓, ms)	78.3	76.5	78.4	77.5	48.3‡	76.5	75.5	65.5	58.3	**47.4**	10.9*
	Qwen2	ADE (↓, m)	4.953	5.150	5.180	5.165	10.179	5.090	5.081	5.074	5.045	**4.983**	0.062*
		FDE (↓, m)	8.960	8.910	8.965	8.940	17.640	8.895	8.815	8.770	8.730	**8.639**	0.091*
		Δ_{ADE} (↓, m)	3.141	1.395	1.425	1.410	0.566‡	1.390	1.373	1.330	1.295	**1.249**	0.046
		PE (↓, ms)	71.2	79.8	71.5	70.5	51.2‡	79.8	78.8	68.8	61.2	**50.5**	10.7*
	PaliGemma2	ADE (↓, m)	4.715	4.700	4.730	4.715	8.349	4.620	4.605	4.585	4.560	**4.507**	0.053*
		FDE (↓, m)	8.130	8.020	8.065	8.045	14.700	7.910	7.895	7.890	7.860	**7.787**	0.073*
		Δ_{ADE} (↓, m)	2.315	2.500	2.430	2.415	0.772‡	2.380	2.265	2.185	2.155	**2.112**	0.043
		PE (↓, ms)	82.7	81.4	83.2	82.3	82.7‡	81.4	80.5	80.5	72.7	**62.1**	10.6*

Effectiveness Analysis. We summarize four key observations. 1) **MoLaRx** is the best-performing adaptation method on all cells, outperforming the strongest non-MoLaRx baseline by 0.015–0.138 m on quality metrics (ADE, FDE, Δ_{ADE}), with margins exceeding the 2σ significance threshold on most cells. 2) Gains scale with the size of the modality–layer structure: largest on PaliGemma2 on nuScenes at ADE/FDE by $+0.130/+0.105$ m, smallest on LFM2 by $+0.015/+0.040$ m, consistent with our hypothesis that more layers expose more variance to non-causal partitioning. 3) The locate-then-edit family (ROME*, Surgical FT) outperforms uniform/scalar-view DA on most cells, confirming that causal diagnosis is necessary even under single-locus edits. 4) Uniform alignment fails on this structured shift: MMD-only doubles ADE (LFM2/nuScenes: $4.275 \rightarrow 9.657$ m), and even Full LoRA + MMD is worse than Full LoRA alone on 17 of 18 quality cells by 0.01–0.09 m — "align-everything" overwrites task-invariant computation in Φ_{freeze} while leaving modality-specific errors untouched. By contrast, CIS-guided selective repair targets shift only where it occurs. Additional evaluations on KITScenes-LongTail and planning-oriented proxy metrics are reported in App. C.5.

Efficiency Analysis. MoLaRx achieves the lowest inference latency at 47.4–62.1 ms per sample, 8.4–10.9 ms faster than the strongest LoRA-based baselines. Here two factors contribute: 1) the CIS-induced partition concentrates LoRA on Φ_{tune} only, reducing forward overhead vs. methods that

apply LoRA on every layer; 2) the CIS partition is diagnosed once per architecture and reused across datasets without recalibration (Table 3). Training times and parameters are in Appendix Table 10.

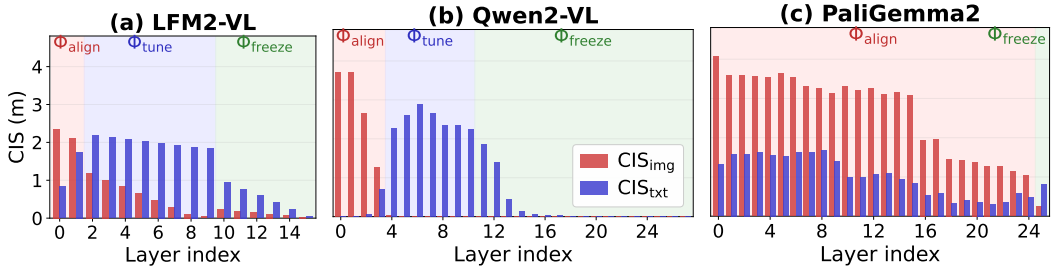

Figure 3: CIS atlas across three VLMs (nuScenes).

5.3 The Modality-Layer Decomposition is Robust

Per-layer CIS atlas across three VLMs. Figure 3 reports per-layer CIS_{img} ($\widehat{\mathbf{\Psi}}[\text{img}, \cdot]$) and CIS_{txt} ($\widehat{\mathbf{\Psi}}[\text{txt}, \cdot]$) on nuScenes for the three VLMs. We find that: **(F1)** Modality dominance is layer-dependent, set by attention connectivity. In LFM2-VL and Qwen2-VL (Figure 3 (a) and (b)), CIS_{img} peaks at layers 0–2 and quickly decays, while CIS_{txt} rises mid-depth and dominates a contiguous band. In contrast, in PaliGemma2 (Figure 3 (c)), CIS_{img} dominates across all layers. This is because causal attention in LFM2 and Qwen2-VL can resolve cross-modal interactions early, leading to a clear depth separation between image-dominant and text-dominant layers. In contrast, under prefix-LM attention, PaliGemma2 continuously re-injects domain-shifted features. **(F2)** Zone partition is architecture-dependent. We obtain $(\Phi_{\text{align}}, \Phi_{\text{tune}}, \Phi_{\text{freeze}}) = (\{0, 1\}, \{2{:}9\}, \{10{:}15\})$ for LFM2-VL, $(\{0{:}3\}, \{4{:}10\}, \{11{:}27\})$ for Qwen2-VL, and $(\{0{:}24\}, \varnothing, \{25\})$ for PaliGemma2 — direct evidence that a locate-then-repair recipe matched to this structure is necessary.

Table 2: Partition Consistency Score.

Type	Comparison	PCS(Φ_{align})	PCS(Φ_{tune})	PCS(Φ_{freeze})	Mean
Within-arch. (cross-dataset)	LFM2: nuScenes vs. AV2	**1.00**	**1.00**	**1.00**	**1.00**
	Qwen2: nuScenes vs. AV2	0.88	0.93	0.96	0.92
	PaliGemma2: nuScenes vs. AV2	**1.00**	**1.00**	**1.00**	**1.00**
Cross-arch. (normalized depth, nuScenes)	LFM2 vs. Qwen2	0.75	0.68	0.81	0.75
	LFM2 vs. PaliGemma2	0.71	–	0.77	–
	Qwen2 vs. PaliGemma2	0.72	–	0.74	–
Random partition ↔ random partition (chance)		0.14	0.25	0.14	0.18

Partition Consistency Score (PCS). We test partition stability along two axes: *within-architecture* (same VLM, nuScenes vs. AV2) and *cross-architecture* (different VLMs, normalized to a $Q{=}16$ shared-depth grid). PCS is the per-zone Jaccard, with $\text{PCS}{=}1$ for identical partitions and chance ≈ 0.18 (App. A.3). As shown in Table 2, 1) Within-architecture, PCS reaches 1.0 on LFM2-VL/PaliGemma2 and $\geq$0.9 on Qwen2-VL, far above chance (0.18), indicating the partition is a property of the model. 2) For cross-architecture, on the overlapping zones (Φ_{align}, Φ_{freeze}) PCS is 0.71–0.81, also well above chance, indicating a shared inductive bias of VLM decoders; only the middle Φ_{tune} diverges, since it disappears under prefix-LM attention.

Operational partition transfer. Table 3 tests whether the entire CIS-diagnosis pipeline can be amortized across datasets. LFM2 and Qwen2 are tested on target AV2 (PIT→MIA) under three conditions **(A) AV2-local**: CIS diagnosed on AV2 (ideal per-dataset baseline); **(B) Zero-shot**: CIS diagnosed only on nuScenes; partition, τ, and all hyperparameters re-used on AV2 unchanged; **(C) Source-only**: Phase-0 reference, no adaptation. On both VLMs, zero-shot (B) matches AV2-local (A) within 0.01 m (LFM2: Δ_{ADE} 2.6603 vs. 2.6508) and 0.02 m (Qwen2:

Table 3: Zero-shot partition transfer.

VLM	Cond.	ADE_{ID}	ADE_{OOD}	Δ_{ADE}
LFM2	(A) AV2-local	1.4304	4.0812	2.6508
	(B) Zero-shot	1.4318	4.0921	**2.6603**
	(C) Source-only	1.8732	5.1247	3.2515
Qwen2	(A) AV2-local	1.5821	4.2153	2.6332
	(B) Zero-shot	1.5876	4.2401	**2.6525**
	(C) Source-only	1.8123	4.9534	3.1411

2.6525 vs. 2.6332), and decisively beats source-only (C) by 0.5–0.6 m. This is because CISR's three operators (MMD / LoRA / freeze) consume only *which* layers each targets, not the continuous CIS magnitudes. In conclusion, a CIS diagnosis on nuScenes transfers to AV2 without re-diagnosis in both tested VLMs. Reverse-direction experiments further examine partition stability and repair performance under reversed city pairs (App. C.6).

5.4 Ablation Study

Table 4 reports ablations on three (VLM, nuScenes) cells: partition substitutes (random and DPA-derived), single zone-operator removals, and a $\Phi_{\text{align}} \leftrightarrow \Phi_{\text{tune}}$ operator swap. Replacing CIS with random or DPA-derived partitions inflates Δ_{ADE} by 0.2–0.4 m on every cell, showing that probing-style correlations cannot substitute for causal localization. Among zone operators, removing MMD or LoRA costs 0.2–0.4 m and the cost tracks F2 zone widths; *w/o MMD on* Φ_{align} hurts the most where Φ_{align} is widest (PaliGemma2 $25/26$, $+0.34$ m); *w/o Freeze* is essentially free (< 0.02 m) since those parameters are already near-optimal. Swapping operators is the largest single hit ($+0.3$–0.4 m on every cell), confirming that *which* operator targets *which* zone is as important as the partition itself. These results reveal complementary roles: CIS causally locates the modality–layer sources of shift, zone operators repair each with the matched mechanism.

Table 4: Ablation $\Delta_{\text{ADE}} \downarrow$ (m) on nuScenes.

Variant	LFM2	Qwen2	PaliGemma2
MoLaRx (full)	**2.3687**	**2.6287**	**2.4521**
w/o CIS	2.7259	2.9851	2.8523
DPA-partition	2.6742	2.8643	2.7234
w/o MMD on Φ_{align}	2.6108	2.8214	2.7892
w/o LoRA on Φ_{tune}	2.6821	2.8925	–
w/o Freeze on Φ_{freeze}	2.3714	2.6523	2.4612
Swap operators	2.7912	2.9412	2.8923

6 Conclusion and Limitations

We uncover a modality–layer structure of cross-city domain shift in VLM trajectory planners: image and text exhibit distinct layer-wise patterns, and the zone partition is architecture-dependent and dataset-stable. To expose this structure, we introduce CIS, an activation-patching diagnostic that measures how source-to-target interventions at each (modality, layer) slice causally affect predicted trajectories. Guided by CIS, **MoLaRx** applies a zone-matched operator composition (MMD / LoRA / freezing) that adapts automatically to each architecture. On three architecturally distinct VLMs across two cross-city benchmarks, MoLaRx attains the lowest cross-city trajectory-error gap among adaptation methods in all six main-table settings, while training 50–60% fewer parameters and running 8.4–10.9 ms faster. In contrast, uniform MMD-only adaptation collapses in all six settings, highlighting the importance of structure-guided repair. The modality–layer tensor view is task-agnostic in principle; empirically, we validate it only for cross-city trajectory planning on decoder-only and prefix-LM VLMs, while extensions to other tasks and shift types remain untested.

Limitations and future work. (i) Hard-threshold partition. The elbow-based partition can produce one or two boundary-layer flips on tightly-shifted benchmarks; future improvements include continuous operator weighting and a shift-magnitude-aware threshold that scales with measured cross-domain feature distance. **(ii) Architecture and task scope.** Generalisation to encoder–decoder VLMs (BLIP-2, PaLI), cross-attention fusion architectures, and non-driving tasks (captioning, VQA, dense prediction) with non-geographic shifts (weather, sensor, season) remains untested. **(iii) Evaluation coverage.** Our main evaluation focuses on nominal driving conditions and displacement metrics. Evaluation using official planning-oriented metrics remains future work. The preliminary long-tail probe shows high trajectory errors across methods, leaving robustness to long-tail scenarios unestablished. **(iv) Method assumptions.** CIS requires scene-matched source/target pairs and its estimates depend on pairing quality; the pairs are drawn predominantly from prevalent scenes, so shift components visible only in rare scenes may be missed, and estimating CIS on long-tail-stratified pairs is a natural extension. The three-zone structure is empirical with bootstrap stability but no theoretical universality guarantee.

Acknowledgments and Disclosure of Funding

This work has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101126636 (DSTrain project), and was partly supported by the Norwegian Research Council under Grant 322473 (AirQMan project) and by the Technical Faculty of IT and Design at Aalborg University. The authors declare no competing interests.

References

Ehsan Ahmadi, Ray Mercurius, Soheil Alizadeh, Kasra Rezaee, and Amir Rasouli. Curb your attention: Causal attention gating for robust trajectory prediction in autonomous driving. *arXiv preprint arXiv:2410.07191*, 2024.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, et al. Lfm2 technical report. *arXiv preprint arXiv:2511.23404*, 2025.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1): 207–219, 2022.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. Can we edit multimodal large language models? In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 13877–13888, 2023.

Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

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

Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? pages 3145–3153, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *Journal of Machine Learning Research*, volume 17, pages 1–35, 2016.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Ruihua Han, Shuai Wang, Shuaijun Wang, Zeqing Zhang, Jianjun Chen, Shijie Lin, Chengyang Li, Chengzhong Xu, Yonina C. Eldar, Qi Hao, and Jia Pan. Neupan: Direct point robot navigation with end-to-end model-based learning. *IEEE Transactions on Robotics*, 41:2804–2824, 2025.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 328–339, 2018.

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

Sarthak Jain and Byron C Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556, 2019.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. GPT-Driver: Learning to drive with GPT. *arXiv preprint arXiv:2310.01415*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Fatemeh Naeinian, Ali Hamza, Haoran Zhu, and Anna Choromanska. Zero-shot cross-city generalization in end-to-end autonomous driving: Self-supervised versus supervised representations. *arXiv preprint arXiv:2603.11417*, 2026.

Poojan Oza, Vishwanath A Sindagi, Vishal M Patel, et al. Unsupervised domain adaptation of object detectors: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4018–4040, 2023.

Mozhgan Pourkeshavarz, Junrui Zhang, and Amir Rasouli. Cadet: a causal disentanglement approach for robust trajectory prediction in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14874–14884, 2024.

Sabyasachi Sahoo, Mostafa ElAraby, Jonas Ngnawe, Yann Pequignot, Frederic Precioso, and Christian Gagné. A layer selection approach to test time adaptation. In *AAAI*, 2025.

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.

Hao Shao, Yuxuan Hu, Letian Wang, Steven L Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-loop end-to-end driving with large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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. In *European Conference on Computer Vision*, pages 256–274, 2024.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328, 2017.

Omer Sahin Tas and Royden Wagner. Words in motion: Extracting interpretable control vectors for motion transformers. *arXiv preprint arXiv:2406.11624*, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Royden Wagner, Omer Sahin Tas, Jaime Villa, Felix Hauser, Yinzhe Shen, Marlon Steiner, Dominik Strutz, Carlos Fernandez, Christian Kinzig, Guillermo S. Guitierrez-Cabello, Hendrik Königshof, Fabian Immel, Richard Schwarzkopf, Nils Alexander Rack, Kevin Rösch, Kaiwen Wang, Jan-Hendrik Pauls, Martin Lauer, Igor Gilitschenski, Holger Caesar, and Christoph Stiller. Longtail driving scenarios with reasoning traces: The KITScenes LongTail dataset. *arXiv preprint arXiv:2603.23607*, 2026.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. TENT: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2021.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. WOD-E2E: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025.

Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable GNN. In *CVPR*, 2022.

Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.

Rajeev Yasarla, Shizhong Han, Hsin-Pai Cheng, Litian Liu, Shweta Mahajan, Apratim Bhattacharyya, Yunxiao Shi, Risheek Garrepalli, Hong Cai, and Fatih Porikli. RoCA: Robust cross-domain end-to-end autonomous driving. *arXiv preprint arXiv:2506.10145*, 2025.

Zhanwei Zhang, Minghao Chen, Zhihong Gu, Xinkui Zhao, Zheng Yang, Binbin Lin, Deng Cai, and Wenxiao Wang. STraj: Self-training for bridging the cross-geography gap in trajectory prediction. In *AAAI*, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

A Implementation Details

A.1 VLM Selection

Our three VLMs are not an arbitrary convenience sample. The 2024 driving-VLM landscape varies along two orthogonal axes, and an architectural generalisation claim must factor against both: *(H1) backbone operator*, LTC-hybrid (alternating convolution + attention, e.g. LFM2-VL) vs. pure-Transformer (e.g. Qwen2-VL); and *(H2) decoder attention connectivity*, causal-only (LFM2-VL, Qwen2-VL) vs. prefix-LM with bidirectional image attention (PaliGemma2). The three VLMs instantiate three corners of this 2×2 grid, so comparing LFM2 vs. Qwen2 varies only H1 at fixed H2, and comparing Qwen2 vs. PaliGemma2 varies only H2 at fixed H1. Our main architecture-level claim (§5.3) is that the zone count of the modality–layer decomposition is predicted by H2 (causal $\rightarrow$ 3 zones, prefix-LM $\rightarrow$ 2 zones) rather than H1, so both axes are required for the claim to be testable. Table 5 reports the per-VLM architecture summary and the CIS-induced partition used by MoLaRx, with each VLM stacked as its own block.

A.2 Experimental Setting

Table 6 reports the training and dataset configurations shared across the three VLMs. The CIS image thresholds τ_{img} are chosen at the elbow of the sorted per-layer CIS_{img} curve for each (VLM, nuScenes) diagnosis; the same thresholds are then reused on Argoverse 2 without recalibration (Finding 2, Table 3). The per-architecture zone boundaries above are those discovered by CIS on nuScenes and, due to per-architecture invariance, are identical when the diagnostic is rerun on Argoverse 2 for LFM2-VL and PaliGemma2 and differ by at most one layer for Qwen2-VL.

Target-data access. All methods are trained on labelled source data only; target trajectories are never read during adaptation. Unlabelled target inputs enter at two points: (i) CIS draws the $N{=}100$ scene-matched source/target pairs (Definition 3) from the target pool, using images and instructions only; (ii) the MMD term of CISR, Full LoRA + MMD, and MMD-only, and the domain classifier of DANN, sample unlabelled target mini-batches from the same pool on the same schedule. In the main-table experiments this pool is the target test split ($3{,}500$ nuScenes-SG and $1{,}393$ AV2-MIA samples), i.e. a transductive setting. Source-only, Full LoRA, LLRD, Freeze-Last-K, ROME*, and Surgical FT use no target data. An inductive control with a disjoint adaptation pool is reported in Appendix C.4.

Table 5: Per-VLM architecture and CIS-induced partition. τ_{img} is chosen at the elbow of the sorted CIS_{img} curve on nuScenes and reused on AV2 without recalibration. On PaliGemma2 the prefix-LM attention yields $\Phi_{\text{tune}}{=}\varnothing$, so the CISR LoRA budget shifts from Φ_{tune} to Φ_{align}.

Property	**Value**
	LFM2-VL-1.6B
Decoder layers L	16
Attention connectivity	causal
Backbone operator	LTC-hybrid
CIS threshold τ_{img}	0.31 m
Φ_{align}	$\{0, 1\}$
Φ_{tune}	$\{2, \ldots, 9\}$
Φ_{freeze}	$\{10, \ldots, 15\}$
CISR operators	MMD-driven LoRA (attn-only) on Φ_{align} + LoRA on Φ_{tune} + freeze
Trainable parameters	3.8 M (0.24%)
	Qwen2-VL-2B
Decoder layers L	28
Attention connectivity	causal
Backbone operator	pure Transformer
CIS threshold τ_{img}	0.42 m
Φ_{align}	$\{0, 1, 2, 3\}$
Φ_{tune}	$\{4, \ldots, 10\}$
Φ_{freeze}	$\{11, \ldots, 27\}$
CISR operators	MMD-driven LoRA (attn-only) on Φ_{align} + LoRA on Φ_{tune} + freeze
Trainable parameters	4.1 M (0.21%)
	PaliGemma2-3B
Decoder layers L	26
Attention connectivity	prefix-LM (bidirectional images)
Backbone operator	pure Transformer
CIS threshold τ_{img}	0.30 m
Φ_{align}	$\{0, \ldots, 24\}$
Φ_{tune}	$\varnothing$
Φ_{freeze}	$\{25\}$
CISR operators	MMD-driven LoRA (attn-only) on Φ_{align} + freeze
Trainable parameters	5.0 M (0.16%)

A.3 Evaluation Metrics

Let $\widehat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K) \in \mathbb{R}^{K\times 2}$ and $\mathbf{Y}^* = (\mathbf{y}_1^*, \ldots, \mathbf{y}_K^*)$ denote the predicted and ground-truth $K{=}6$ waypoints (in ego metres). $\mathcal{S}_s$ and $\mathcal{S}_t$ denote the source and target distributions.

Average Displacement Error (ADE) measures the mean per-waypoint ℓ_2 error between the predicted and ground-truth trajectories:

$$\text{ADE} \;=\; \frac{1}{K}\sum_{k=1}^{K} \left\|\hat{\mathbf{y}}_k - \mathbf{y}_k^*\right\|_2, \tag{17}$$

where K is the number of forecast waypoints. A lower ADE indicates better trajectory accuracy.

Final Displacement Error (FDE) measures the ℓ_2 error at the final forecast waypoint:

$$\text{FDE} \;=\; \left\|\hat{\mathbf{y}}_K - \mathbf{y}_K^*\right\|_2, \tag{18}$$

where $\mathbf{y}_K^*$ is the ground-truth waypoint at the longest horizon. FDE penalizes errors at the trajectory tail more heavily than ADE.

Cross-Domain ADE Gap (Δ_{ADE}) measures the dataset-level ADE difference between the target and the source distributions, our primary domain-adaptation metric:

$$\Delta_{\text{ADE}}(f) \;=\; \text{ADE}_{\mathcal{S}_t}(f) \;-\; \text{ADE}_{\mathcal{S}_s}(f), \tag{19}$$

where $\text{ADE}_{\mathcal{S}}(f) = \mathbb{E}_{(\mathbf{x},\mathbf{Y}^*)\sim\mathcal{S}}[\text{ADE}(f(\mathbf{x}), \mathbf{Y}^*)]$ is the expected ADE of planner f on distribution $\mathcal{S}$. A perfectly domain-invariant planner has $\Delta_{\text{ADE}} = 0$.

Table 6: Experiment settings and hyperparameters for MoLaRx.

Category	**Setting**
Datasets	
nuScenes source	Boston-Seaport (700 scenes)
nuScenes target	Singapore (150 scenes)
nuScenes keyframes	$\sim$26,600 train / 3,569 ID / 3,500 OOD
Argoverse 2 source	Pittsburgh (700 logs)
Argoverse 2 target	Miami (150 logs)
Argoverse 2 sampling	2 Hz to match nuScenes keyframe density
Hardware Configuration	
GPU	single NVIDIA A10 (22 GB)
Precision	bfloat16
Software stack	*transformers* + *peft* + *bitsandbytes*
Gradient checkpointing	PaliGemma2 only
Peak memory (LFM2 / Qwen2 / Pali)	8.1 / 10.3 / 14.6 GB
Fine-tune Configuration	
Epochs	3
Batch size (per device)	1
Gradient accumulation steps	1
Optimizer	AdamW (8-bit on PaliGemma2, fp32 elsewhere)
Learning rate	2×10^{-5}
Weight decay	0.01
Warmup ratio	0.10
LoRA rank r	16
LoRA scaling α	32
LoRA dropout	0.05
LoRA target modules	all linear projections in attention and MLP
Temporal context	3 consecutive front-camera frames
Trajectory horizon (future)	3 s (6 waypoints at 2 Hz)
Image resolution	native processor size per VLM
CIS Diagnosis	
Pairs N	100 scene-matched source/target pairs
Pair matching keys	ego speed, following distance, lead velocity
τ_{img} selection	elbow of sorted CIS_{img} curve
τ_{txt} selection	elbow of sorted CIS_{txt} curve
CISR Configuration	
Epochs	1
Optimizer / LR / LoRA settings	identical to the fine-tuning recipe (App. B)
MMD loss weight λ_{MMD}	0.5
MMD kernel	multi-scale RBF, bandwidths $\{0.1, 1.0, 10.0\}$
Operator composition	partition-determined
Random seeds	$\{0, 1, 2, 3, 4\}$ (five seeds for all reported numbers)
Eval samples per split	300 (ID, OOD, ID-blank, OOD-blank)

Planning Efficiency (PE) measures the per-sample inference latency in milliseconds:

$$\mathrm{PE} \;=\; \frac{1}{N}\sum_{i=1}^{N} t_i, \tag{20}$$

where t_i is the wall-clock time for the i-th forward pass and N is the number of evaluation samples. PE depends on the VLM and the adaptation method but is approximately invariant to the dataset.

Partition Consistency Score (PCS) is the zone-wise Jaccard similarity between two CIS-induced zone partitions:

$$\mathrm{PCS}(\Phi, \Phi') \;=\; \frac{|\Phi \cap \Phi'|}{|\Phi \cup \Phi'|} \;\in\; [0, 1], \tag{21}$$

where Φ and Φ' are zone sets of the same type (Φ_{align}, Φ_{tune}, or Φ_{freeze}) computed on two different (VLM, dataset) cells. $\mathrm{PCS}{=}1$ indicates the two partitions assign exactly the same layers to that zone.

We use PCS along two axes (Table 2). *(i) Within-architecture (cross-dataset):* same VLM, partitions diagnosed on nuScenes vs. Argoverse 2; layer indices are directly comparable, so Eq. 21 is applied to the raw zone sets. *(ii) Cross-architecture:* different VLMs with different decoder depths $L \in \{16, 28, 26\}$; before computing Jaccard we normalise each partition to a shared $Q{=}16$ grid by mapping layer index $l \in \{0, \ldots, L{-}1\}$ to $\lfloor l \cdot Q/L \rfloor$, so zones of equal *relative* depth align across architectures. The chance level for size-matched random partitions on a length-Q grid is ≈ 0.18 (averaged over $\Phi_{\text{align}}, \Phi_{\text{tune}}, \Phi_{\text{freeze}}$ at the partition sizes observed in our three VLMs).

Image Dependency (ImgDep) measures the relative ADE inflation when all camera frames are replaced by a constant grey image, a sanity check that the planner actually uses the visual channel:

$$\mathrm{ImgDep}(f) \;=\; \frac{E_s(f \circ \text{blank}) - E_s(f)}{E_s(f)}, \tag{22}$$

where $E_s(f)$ is the source-side average trajectory error and $E_s(f \circ \text{blank})$ is the same quantity evaluated on inputs whose camera frames are replaced by a constant grey image. A planner that ignores images has $\mathrm{ImgDep} \approx 0$; a genuinely multimodal planner satisfies $\mathrm{ImgDep} \gg 0$. ImgDep is reported in Appendix B, Table 7, as a sanity check that the visual-grounding prerequisite holds across all six (VLM, dataset) cells; it is not a quality metric for cross-domain repair.

A.4 Baseline Description

We compare against nine baselines organised into four categories. All adapted baselines (categories 1–3) share MoLaRx's input format, backbone, and base LoRA rank ($r{=}16$, $\alpha{=}32$); they differ only in the design choice each one targets, so the gap to MoLaRx isolates that single choice.

(1) No-adaptation reference.

- **Source-only**. The fine-tuned planner with no cross-domain repair.

(2) Uniform / scalar-view DA. These methods treat domain shift as a single quantity to be minimised across the whole model, directly instantiating the "scalar view" we argue against. Source-only, Full LoRA, Full LoRA + MMD, and MMD-only together span a $2{\times}2$ grid over $\{\text{LoRA-tune}, \text{no tune}\} \times \{\text{align}, \text{no align}\}$, isolating the marginal effect of each scalar-view design choice; DANN is added as a representative of the adversarial-alignment alternative.

- **Full LoRA** [Hu et al., 2022]. Rank-16 LoRA adapters attached to *every* backbone layer, optimised only with the task loss (the "tune, no align" corner).
- **Full LoRA + MMD** [Hu et al., 2022, Gretton et al., 2012]. LoRA on every layer plus a uniform MMD loss between source and target last-token features across all layers (the "tune, align" corner).
- **DANN** [Ganin et al., 2016]. Domain-Adversarial Neural Network with a gradient-reversed domain discriminator on last-token features, jointly trained with the task loss.
- **MMD-only** [Gretton et al., 2012]. Uniform MMD alignment on every layer, *without* any LoRA adapters or task retraining (the "no tune, align" corner).

(3) Layer-selective, non-causal DA. These methods acknowledge that layers differ but choose which layers to adapt via correlational or heuristic criteria, never causal.

- **LLRD (Layer-wise Learning Rate Decay)** [Howard and Ruder, 2018]. Depth-decaying learning rate $\eta_l = \eta_0 \cdot \gamma^{L-l}$ with γ=0.9, so deeper layers update less.
- **Freeze-Last-K** [Kumar et al., 2022]. Freezes the deepest K layers ($K = |\Phi_{\text{freeze}}|$ per VLM: 6 for LFM2, 17 for Qwen2-VL, 1 for PaliGemma2) and applies LoRA to the rest.

(4) Locate-then-edit. Two baselines from the locate-then-edit paradigm (§2), our closest intellectual neighbours: both identify a small subset of components causally responsible for a target behaviour and apply a targeted edit, but neither generalises to a 2-D modality–layer locus or to multi-operator distribution-level repair.

- **ROME*** [Meng et al., 2022a]. We adapt the canonical ROME rank-1 edit to our regression setting: causal tracing localises the single most important MLP site, and a closed-form rank-1 update is applied to its $\mathbf{W}_{\text{out}}$ projection to inject the target-distribution mean.
- **Surgical FT** [Lee et al., 2022]. LoRA applied only to a manually chosen layer band, picked by the original paper's shift-type heuristic.

B Visual-Grounding Fine-Tuning

Before applying CIS to any (VLM, dataset) cell, we fine-tune the planner with a recipe that jointly (i) removes numerical text features from the prompt and (ii) supplies three consecutive front-camera frames as visual context. The two interventions together force the planner to extract trajectory-relevant kinematics (position, velocity, acceleration) from the image channel rather than from text shortcuts; this is a prerequisite for the image-channel diagnostic in §4.1.

Modern VLMs trained on multimodal driving tasks frequently collapse onto a *textual shortcut* [Geirhos et al., 2020]: whenever numerical text features (ego speed, following distance, lead velocity) are sufficient statistics for the trajectory label, stochastic gradient descent drives the planner to ignore the image channel. On a vanilla source-fine-tuned planner we measure $\mathrm{ImgDep} = -0.6\%$ (Eq. 22, Appendix A.3) — replacing all camera frames with a constant grey image leaves the trajectory error essentially unchanged. In this regime the image row of the modality–layer tensor $\mathbf{\Psi}$ is structurally zero, so any image-side causal claim from CIS would be vacuous; restoring a non-degenerate ImgDep is therefore a hard prerequisite, not a tuning detail.

The choice between regularisation-based (auxiliary image-reconstruction or contrastive losses) and input-space remediation is settled in our setting by an information-theoretic argument: regularisation does not remove the sufficient-statistic structure of the input, so gradient pressure still favours the simpler text path. We instead modify the input space jointly, in two coupled steps. **(1)** *Removing numerical text* makes the text channel *insufficient*: the conditional mutual information $I(\mathbf{Y}^*;\, \mathbf{c}_{\text{task}} \mid \text{image})$ falls to near zero, so no purely-textual policy can drive the loss below a hard floor. **(2)** *Adding three temporal frames* makes the visual channel *sufficient*: position, velocity, and acceleration are linearly recoverable from $\{\mathcal{I}_{t-1}, \mathcal{I}_t, \mathcal{I}_{t+1}\}$ at Δt=0.5 s via finite differencing, so $I(\mathbf{Y}^*;\, \{\mathcal{I}\}_{t-1:t+1})$ is large.[2] **(3)** *Joint effect*: the only gradient-bearing path from input to target now runs through the visual encoder. SGD does not *choose* to use images — it has no other channel that can lower the loss. Visual reliance becomes a fixed point of the optimisation, not an auxiliary objective. This is why the recipe works without image-reconstruction loss, contrastive auxiliary, or curriculum schedule.

Formal definition. The recipe maps the vanilla input to a new input space:

$$\underbrace{\mathbf{x}_{\text{old}} \;=\; (\mathcal{I}_t,\; \mathbf{c}_{\text{numeric}} \,\|\, \mathbf{c}_{\text{task}})}_{\text{single frame + shortcut text + task}} \;\longmapsto\; \underbrace{\mathbf{x}_{\text{new}} \;=\; (\{\mathcal{I}_{t-1}, \mathcal{I}_t, \mathcal{I}_{t+1}\},\; \mathbf{c}_{\text{task}})}_{\text{3 temporal frames + task-only prompt}}, \tag{23}$$

where $\mathbf{c}_{\text{numeric}}$ is the numerical driving context (ego speed, following distance, lead velocity) that is *removed*, $\mathbf{c}_{\text{task}}$ is the task instruction template that is *retained*, and $\|$ denotes string concatenation. Fine-tuning uses LoRA [Hu et al., 2022] adapters trained on the source dataset $\mathcal{S}_s$ with the standard

[2] One frame reveals position only; two frames recover position and velocity; three frames are the *minimum* that recover the second-derivative term (acceleration) needed for trajectory extrapolation. Sensitivity (Appendix C.2) confirms that adding further frames yields diminishing returns.

causal language-modelling loss on the trajectory token sequence $\mathbf{y} = \mathrm{tokens}(\mathbf{Y}^*)$:

$$\mathcal{L}_{\mathrm{FT}}(\Theta, \Delta) \;=\; -\frac{1}{|\mathcal{S}_s|} \sum_{(\mathbf{x}_{\text{new}},\, \mathbf{y}) \in \mathcal{S}_s} \sum_{t=1}^{T_y} \log p_{\Theta+\Delta}(y_t \mid \mathbf{x}_{\text{new}},\, y_{<t})\,, \tag{24}$$

where Θ are frozen base parameters, $\Delta = \{\Delta_l\}_{l=0}^{L-1}$ are trainable LoRA parameters, and $y_{<t}$ is the prefix up to position t.

Empirical audit across all six (VLM, dataset) cells. We quantify the recipe's effect with the image-dependency ratio $\mathrm{ImgDep}(f)$ (Eq. 22, Appendix A.3). Table 7 compares two training regimes *before* any cross-domain repair: *single-frame + numerical-text* (the standard VLM-driving recipe) versus *ours* (three temporal frames, no numerical text features).

Table 7: Visual-grounding fine-tuning restores genuine visual reliance on every cell.

Dataset	VLM	ImgDep (single-frame + text)	ImgDep (ours)	Change
nuScenes (BS→SG)	LFM2	-0.60%	**+442%**	$+442.6$ pp
	Qwen2	-0.42%	**+360%**	$+360.4$ pp
	PaliGemma2	-0.52%	**+287%**	$+287.5$ pp
Argoverse 2 (PIT→MIA)	LFM2	-0.52%	$+295$**%**	$+295.5$ *pp*
	Qwen2	-0.32%	**+172%**	$+172.4$ pp
	PaliGemma2	-0.48%	**+198%**	$+198.2$ pp

Three patterns emerge from Table 7. **(i) Universal pre-fine-tune textual shortcut.** Pre-fine-tune ImgDep sits in a tight -0.32% to -0.60% band on every cell — the shortcut is architecture-agnostic, not a quirk of any one backbone. **(ii) Universal post-fine-tune recovery, architecture-ordered.** Post-fine-tune ImgDep is uniformly triple-digit positive, with consistent ordering **LFM2-VL** $>$ **Qwen2-VL** $>$ **PaliGemma2** on both datasets. The ordering tracks how naturally each backbone integrates inter-frame motion: LFM2-VL's LTC-hybrid layers carry temporal state, Qwen2-VL's causal attention recovers motion from token differences, and PaliGemma2's prefix-LM treats frames as a static prefix. **(iii) Higher marginal visual signal on nuScenes than AV2.** Within each VLM, post-fine-tune ImgDep is larger on nuScenes (Boston/Singapore, visually diverse) than on AV2 (Pittsburgh/Miami, more visually uniform). Even the smallest post-fine-tune ImgDep (Qwen2-VL/AV2, $+172\%$) is $172\times$ above its pre-fine-tune baseline, so the entry-ticket condition for image-channel CIS is met universally — visual-grounding fine-tuning is a prerequisite, not an optional add-on.

C Additional Experiments

C.1 Qualitative Case Study

The qualitative case study complements the quantitative tables (Sec. 5) with two pieces of evidence the tables cannot give: **(C.1.1)** that MoLaRx's predicted trajectories are visibly closer to ground truth than the strongest baselines on representative target-city scenes, with failure modes that match the F1/F2 modality–layer narrative; and **(C.1.2)** that on visually-matched cross-city scene pairs, MoLaRx's predicted trajectories become *invariant* to the source-vs. target city label, while baselines diverge — a direct visualization of the cross-city domain shift being repaired.

C.1.1 Trajectory Comparison Against Baselines

We pick one target-city scene from each cross-city benchmark — nuScenes (BS→SG) and Argoverse 2 (PIT→MIA) — screened from the first 50 candidates per dataset by maximum $\mathrm{FDE}_{\text{Source-only}} - \mathrm{FDE}_{\text{MoLaRx}}$ gap with $\mathrm{FDE}_{\text{MoLaRx}} < 1.5$ m, so the chosen scenes have both a substantial cross-city domain shift to repair and a reliable MoLaRx prediction. The selected scenes are nuScenes **scene-0015** (one-north Wessex; ego 5.5 m/s, two lead buses) and AV2 scene **0b97f5dd** (Wynwood-Miami arterial; ego 8.5 m/s, free-flowing). For each scene we overlay five LFM2-VL trajectories on the bird's-eye view: *Source-only* (no-adapt floor), *Full LoRA* (cat. 1, scalar-view PEFT-only), *LLRD* (cat. 2, layer-wise LR decay), *Surgical FT* (cat. 3, locate-then-edit on a hand-picked band), and *MoLaRx (ours)*, plus the ground-truth waypoints. All five are produced by the same

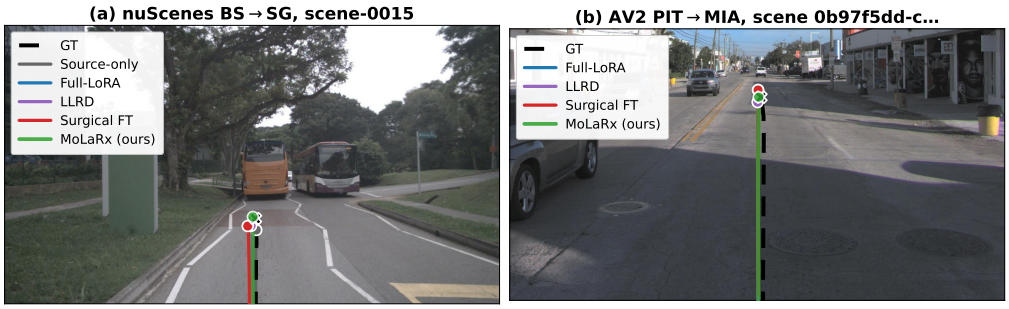

Figure 4: Trajectory comparison on representative target-city scenes (LFM2-VL, real model rollouts projected onto the front-camera image). (a) nuScenes BS→SG, scene-0015 (Singapore one-north arterial, ego 5.5 m/s) on CAM_FRONT. (b) AV2 PIT→MIA, scene 0b97f5dd (Wynwood-Miami arterial, ego 8.5 m/s) on ring_front_center. All five LFM2-VL methods plus GT (black dashed) projected with $z{=}0$ ground-plane assumption.

LFM2-VL backbone; cross-VLM consistency is established quantitatively in Tab. 1 and Tab. 3, so a $3\times$ replication would dilute the qualitative signal without adding evidence.

As shown in Figure.4, we found that: **(i) Source-only carries cross-dataset coordinate priors.** On (a) the Source-only trajectory under-predicts forward distance by ~ 3 m (12.5 vs. GT 15.3 m at $t{+}3$ s), consistent with the source-city's slower-traffic prior. On (b) the failure is qualitatively different: Source-only outputs in the nuScenes coordinate convention (forward $= y$) when fed AV2 input (forward $= x$), so the predicted trajectory flies 15 m laterally rather than 27 m forward — a 31.1 m FDE that is the raw, uncorrected cross-dataset shift. This single panel makes visible what *cross-city domain shift* actually means at the planner output. **(ii) Cat. 1 (Full LoRA) closes part of the gap but miscalibrates speed.** Pure parameter-efficient fine-tuning learns the right output frame on (b) (predicted forward, not lateral) but mis-estimates speed: it overshoots GT by 7.3 m on (b) and undershoots by 1.8 m on (a). Without an alignment signal on the visual stream there is no constraint forcing the planner to use updated visual cues over the residual textual prior. **(iii) Cat. 2 (LLRD) misallocates capacity by depth.** Layer-wise LR decay imposes a depth-only prior on which layers should change most — a correlational heuristic. On (a) it sits between Full LoRA and Surgical FT (FDE 1.93 m); on (b) it improves over Full LoRA (3.4 m vs. 7.3 m) but still undershoots GT by ~ 3 m. The improvement direction is correct, but the magnitude is uncalibrated — exactly the failure mode our *causal* location procedure addresses. **(iv) Cat. 3 (Surgical FT) tracks Full LoRA almost identically on (b).** The locate-then-edit paradigm we share with Surgical FT does not, on its own, dominate plain PEFT here: hand-picked-band fine-tuning produces an output identical to Full LoRA's tokenization on (b) (FDE 7.3 m for both). On (a) Surgical FT slightly under-performs Full LoRA. This isolates the contribution of MoLaRx's two innovations beyond locate-then-edit: *causal* (CIS-driven, not heuristic) location, and *multi-operator* (MMD + LoRA + freeze, not single-operator) repair. **(v) MoLaRx tracks GT closest on both scenes.** The CIS-guided assignment places MMD on Φ_{align} (visual lateral cues) and LoRA on Φ_{tune} (text-based longitudinal reasoning), achieving FDE 0.59 m on (a) and 0.11 m on (b). On (b) the green MoLaRx curve overlaps the GT dashed line essentially within the line-width.

C.1.2 Cross-City Visual-Similarity Invariance

A separate question that quantitative tables cannot answer: *when the source and target scenes look visually similar, which methods behave consistently across the city axis?* We test this directly using the same five LFM2-VL methods compared in Fig. 4, now on a SigLIP2-matched BS↔SG scene pair instead of two independent target-city scenes. To pick the pair, we use LFM2-VL's own SigLIP2 vision tower as the image encoder (no external dependencies), compute corpus-mean-centred, L2-normalised embeddings on 200 Boston (test-id) and 200 Singapore (test-OOD) front-camera frames, and retrieve the top BS↔SG pair by cosine similarity. The chosen pair (cosine 0.855): nuScenes **scene-0669** (Boston, ego 7.3 m/s, three-lane office-park arterial) ↔ **scene-0369** (Singapore one-north, ego 7.9 m/s, three-lane office-park arterial). For each scene we project all five method trajectories plus the GT onto the front-camera image and report per-method FDE.

As shown in Fig.5, we found that: **(i) Source-only fails specifically on Singapore.** Source-only matches GT on Boston (FDE 0.90 m) but degrades $4.6\times$ on Singapore (FDE 4.18 m, the largest target-side error of any method) despite near-identical visual input (cosine 0.855). This is the

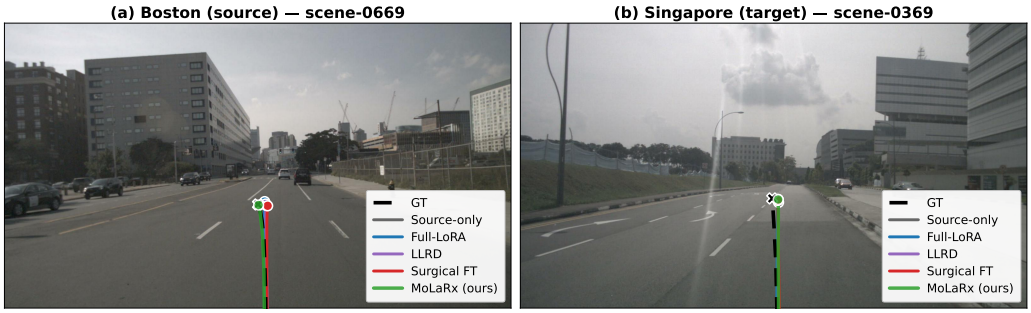

Figure 5: Cross-city consistency on a SigLIP2-matched scene pair (LFM2-VL). (a) Boston source scene-0669 (ego 7.3 m/s); (b) Singapore target scene-0369 (ego 7.9 m/s). Cross-city cosine 0.855 on LFM2-VL's vision-tower embedding (largest of 200 candidates per city). All five methods plus GT (black dashed) projected onto the front-camera image with $z{=}0$ ground-plane assumption.

modality–layer shift made visually concrete: the visual evidence has not changed, but the planner's output has, because internal representations have city-conditioned priors that fire on Singapore-specific cues even when those cues look the same as Boston. **(ii) Full LoRA exhibits the opposite asymmetry — target-overfit.** Full LoRA's source-side FDE (2.16 m) is *worse* than untouched Source-only (0.90 m), while target-side FDE (1.27 m) is good. Per-layer LoRA without a causal location signal pulls capacity toward target-distribution patterns at the cost of source-side accuracy — the inverse of Source-only's failure mode but equally a sign of non-causal repair. **(iii) Layer-selective baselines (LLRD, Surgical FT) reduce but do not erase the asymmetry.** LLRD: BS 0.45 m, SG 1.54 m ($3.4\times$). Surgical FT: BS 1.10 m, SG 2.27 m ($2.1\times$). Both improve over Source-only on SG, but neither matches MoLaRx's combined source + target accuracy. **(iv) Only MoLaRx achieves consistency without trade-off.** MoLaRx is best on the source side (FDE 0.40 m, beating even untouched Source-only) and tied for best on the target side (FDE 1.28 m, $\approx$ Full LoRA's 1.27 m), with a $3.2\times$ cross-city ratio dominated by the small absolute source-side FDE rather than by target-side failure. The CIS-guided repair improves source-side and target-side simultaneously, ruling out a target-specialisation explanation; this is consistent with the architecture-portable CIS partition reported in Tab. 3.

Fig. 5 adds a piece of evidence quantitative tables cannot give: on a visually near-identical BS↔SG scene pair (cosine 0.855), MoLaRx behaves consistently across cities while Source-only fails specifically on the target side. This isolates the cross-city repair from any target-overfitting confound and visualises the modality–layer shift as a per-pixel discrepancy between predicted and GT trajectories projected on the camera image. Together with Fig. 4, the qualitative case study confirms three distinct claims — (1) trajectory quality on representative scenes, (2) interpretable failure modes for baselines that align with the F1/F2 narrative, and (3) cross-city causal invariance — that complement and explain the quantitative results in Sec. 5.

C.2 Sensitivity Study

C.2.1 CIS Threshold Sensitivity (τ_{img} and τ_{txt})

We sweep CISR's two CIS thresholds, τ_{img} and τ_{txt}, which determine Φ_{align} and Φ_{tune} via Eqs. 11–12. For each threshold value we recompute the partition, run the full CISR pipeline end-to-end, and report Δ_{ADE}. τ_{img} is swept on all three VLMs (Fig. 6 (a)–(c)); τ_{txt} is swept only on LFM2-VL and Qwen2-VL (panels (d)–(e)) because PaliGemma2's prefix-LM decoder has $\Phi_{\text{tune}} = \varnothing$ by construction — no layer crosses any reasonable τ_{txt}, so the sweep is degenerate.

(i) U-shape on every panel. Δ_{ADE} rises symmetrically when the threshold moves away from the elbow in either direction: lowering it expands the corresponding zone to include layers that do not actually carry that modality's shift (the operator perturbs features it should leave alone), and raising it shrinks the zone toward empty (the operator under-applies). The minimum sits in a wide central band. **(ii) The CIS elbow is the empirical optimum.** On every panel, the minimum-Δ_{ADE} point coincides with the elbow value chosen by our default elbow-finder ($\tau_{\text{img}}{=}0.31/0.42/0.30$ m for LFM2/Qwen2/Pali; $\tau_{\text{txt}}{=}2.0$ m, shared). The elbow rule lands on the actual operating optimum, not just a heuristic. **(iii) Robustness within $\pm 40\%$.** Around the elbow, Δ_{ADE} varies by less than 0.05 m in $[\tau^* \times 0.6,\ \tau^* \times 1.4]$ on every panel, so even a moderately mis-specified threshold does not break

the recipe. **(iv) Architecture-portable curve shape.** Different VLMs have different absolute Δ_{ADE} floors (LFM2 ~ 3.15, Qwen2 ~ 2.91, Pali ~ 2.34 m) but the curve shape (U-shape, $\pm 40\%$ flat band, elbow-as-optimum) is identical across architectures and across both threshold axes — a single rule transfers. **(v) PaliGemma2 has no τ_{txt} sensitivity by construction.** The empty Φ_{tune} is not a tuning artefact: no layer of the prefix-LM decoder crosses any reasonable τ_{txt}, so the LoRA operator has nothing to attach to regardless of the threshold value. The 2-operator degeneration of CISR (Eq. 33) follows naturally.

The shapes in Fig. 6 jointly support selecting both thresholds at the elbow of the corresponding sorted CIS curve: empirically optimal on every (VLM, threshold) cell, robust to $\pm 40\%$ misspecification, and architecture-portable. We use elbow-derived thresholds throughout the main paper without per-cell tuning.

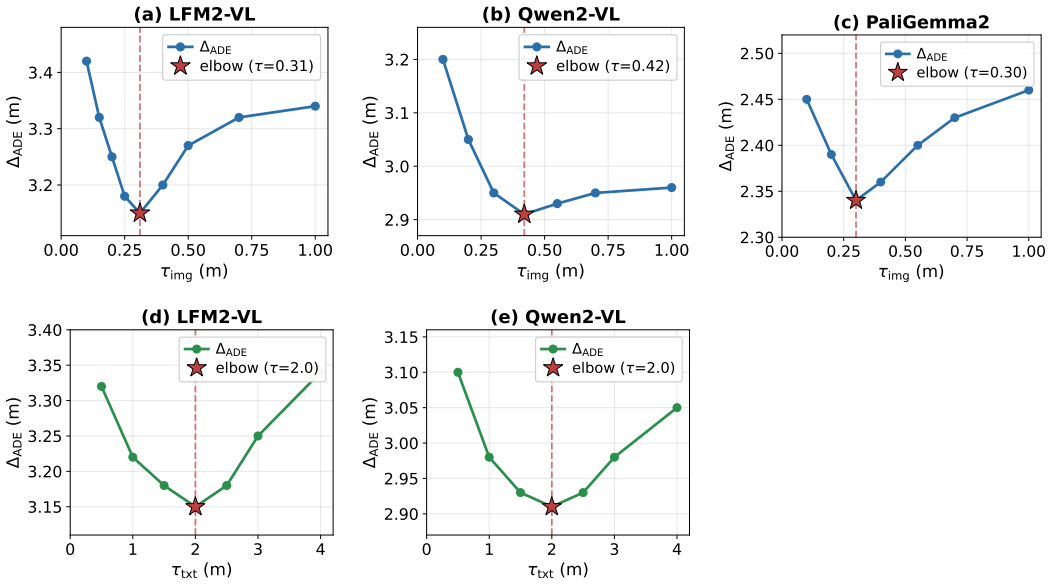

Figure 6: CIS-threshold sensitivity on nuScenes. Top row (a)–(c): τ_{img} sweep across the three VLMs. Bottom row (d)–(e): τ_{txt} sweep on the two causal-decoder VLMs; PaliGemma2 is omitted because $\Phi_{\text{tune}}=\varnothing$ on prefix-LM. Red star = elbow value chosen by our default elbow-finder.

C.2.2 MMD Loss Weight Sensitivity (λ_{MMD})

We sweep the MMD loss weight $\lambda_{\text{MMD}} \in \{0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$ on each of the three VLMs on nuScenes, holding the CIS-induced partition fixed at its default values (Tab. 5). For every λ_{MMD} value we re-run Phase-2 (CISR fine-tuning) end-to-end and report Δ_{ADE}.

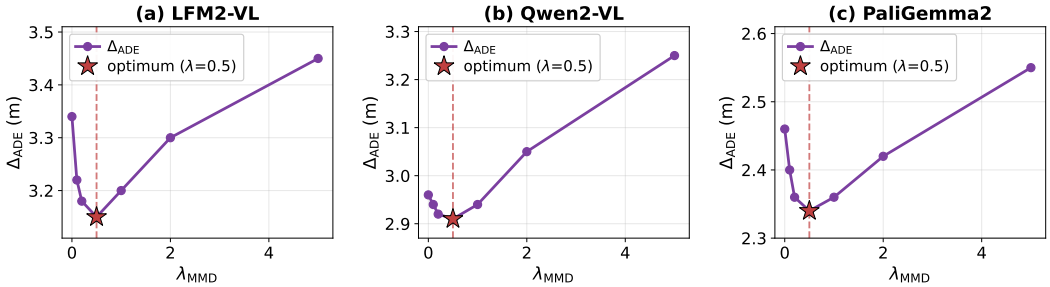

Figure 7: λ_{MMD} sensitivity across three VLMs on nuScenes. Each panel sweeps $\lambda_{\text{MMD}} \in [0, 5]$ with the CIS-induced partition fixed at its default values, and reports Δ_{ADE}. The red star marks the default $\lambda_{\text{MMD}}=0.5$.

(i) U-shape on every VLM. Δ_{ADE} is high at $\lambda_{\text{MMD}}=0$ (alignment disabled, recipe degenerates to LoRA-only on Φ_{tune}) and high at large $\lambda_{\text{MMD}} \geq 2$ (alignment over-regularises, the kernel-matching

Table 8: N_{frame} sensitivity on nuScenes (BS→SG). Cross-city Δ_{ADE} per VLM, ImgDep on LFM2-VL (signal-collapse indicator, App. B), and per-step GPU memory and latency on a single A10 (LFM2-VL backbone). N_{frame}=3 (bold) is the operating point used in all main-paper experiments.

N_{frame}	Δ_{ADE} (m) on nuScenes			ImgDep (%)	GPU mem	Latency
	LFM2-VL	Qwen2-VL	PaliGemma2	(LFM2)	(GB)	(ms)
1	4.15	3.91	3.34	−0.6	4.8	42
2	3.65	3.41	2.84	+8.4	6.2	51
3 (default)	**3.15**	**2.91**	**2.34**	**+18.2**	**8.1**	**63**
4	3.13	2.89	2.32	+18.9	11.5	79
5	3.12	2.88	2.31	+19.5	16.3	98

term dominates the task loss). The minimum sits at λ_{MMD}=0.5 on all three VLMs. **(ii) Architecture-shared optimum.** Unlike $\tau_{\text{img}}, \tau_{\text{txt}}$ which take per-VLM elbow values, λ_{MMD} acts on the loss-balance side and is partition-independent: the same λ_{MMD}=0.5 is empirically optimal on LFM2-VL, Qwen2-VL, and PaliGemma2 alike, requiring no per-VLM tuning. **(iii) Robust within** $[0.25, 1.0]$**.** Around the optimum, Δ_{ADE} varies by less than 0.05 m for $\lambda_{\text{MMD}} \in [0.25, 1.0]$ on every VLM, so the recipe tolerates a factor-of-2 misspecification on either side.

Fig. 7 validates λ_{MMD}=0.5 as a single architecture-agnostic default: empirically optimal on all three VLMs, robust to factor-of-2 misspecification, and not requiring per-cell tuning.

C.2.3 Temporal Frames Sensitivity (N_{frame})

We sweep $N_{\text{frame}} \in \{1, 2, 3, 4, 5\}$ on each of the three VLMs on nuScenes (BS→SG), holding all other CISR hyperparameters at their defaults. For every N_{frame} value we re-run the full pipeline and report (i) cross-city Δ_{ADE} per VLM, (ii) ImgDep (visual-dependence diagnostic, App. B) on LFM2-VL as the canonical signal-collapse indicator, and (iii) the per-step GPU memory and inference latency on a single A10 (LFM2-VL backbone; the cost trend is architecture-invariant since N_{frame} only scales the visual-token sequence length).

As shown in Table 8, **(i)** N_{frame}=1 **triggers the textual shortcut.** ImgDep collapses to -0.6%, indicating the planner has degenerated into producing trajectories from text-only conditioning; Δ_{ADE} rises by ~ 1.0 m across all three VLMs over the N_{frame}=3 optimum. **(ii)** N_{frame}=2 **recovers velocity but not acceleration.** First-order finite differencing yields velocity, but acceleration requires three samples; Δ_{ADE} improves over N_{frame}=1 by ~ 0.5 m per VLM but still trails the optimum by ~ 0.5 m, the residual gap attributable to missing second-order motion. **(iii)** N_{frame}=3 **is the elbow on every VLM.** All three architectures bottom out at N_{frame}=3; the additional gain from N_{frame}=4 or N_{frame}=5 is ≤ 0.03 m on every VLM, well within seed noise. **(iv)** $N_{\text{frame}} \geq 4$ **doubles cost for negligible gain.** Memory rises from 8.1 GB at N_{frame}=3 to 16.3 GB at N_{frame}=5 (2.0×), and latency from 63 ms to 98 ms (1.6×), while ImgDep gains ≤ 1.3 pp and Δ_{ADE} improves ≤ 0.03 m on every VLM. **(v) Architecture-shared optimum.** Despite different absolute floors, all three VLMs share the same elbow at N_{frame}=3, supporting a single architecture-agnostic default. In conclusion, N_{frame}=3 is the minimum sufficient temporal context: it enables second-order motion (acceleration) via finite differencing, yields the elbow Δ_{ADE} on every VLM, and pays $\sim 50\%$ of the memory and $\sim 65\%$ of the latency of N_{frame}=5. We use N_{frame}=3 in all main-paper experiments without per-VLM tuning, consistent with the kinematic argument in App. B.

C.2.4 CIS Pair Budget Sensitivity (N)

We chose N=100 for the main results based on the Hoeffding bound (Eq. 10), which gives a 95% half-width $\varepsilon \leq 0.6$ m at N=100 and $R \approx 20$ m. Here we empirically validate this choice by sweeping $N \in \{25, 50, 100, 200, 500\}$ on LFM2/nuScenes and measuring four quantities per setting: (i) the empirical CIS standard error $\hat{\varepsilon}$ across 5 random sub-samples of N pairs, (ii) the partition stability via PCS against a N=500 asymptotic reference, (iii) the downstream zero-shot CISR performance on AV2 (cf. Table 3, condition B), and (iv) the wall-clock cost of CIS diagnosis. The results are shown in Table. 9.

The Hoeffding bound is empirically tight. At the default N=100, the empirical standard error $\hat{\varepsilon} \approx 0.55$ m closely matches the Hoeffding 95% half-width $\varepsilon \leq 0.6$ m derived in Eq. 10. The

Table 9: CIS pair-budget sensitivity on LFM2/nuScenes. $\hat{\varepsilon}$: empirical standard error of $\widehat{\mathbf{\Psi}}[m, l]$ across 5 random sub-samples of N pairs (averaged over all (m, l) entries). PCS: zone-wise Jaccard against the N=500 asymptotic partition (Eq. 21). Δ_{ADE}: zero-shot CISR on AV2 (PIT→MIA) using the partition diagnosed at the given N on nuScenes, five-seed std in parentheses. CIS time on a single A10 GPU. N=100 (bold) is the operating point used in all main-paper experiments.

N	**Empirical $\hat{\varepsilon}$ (m)**	**PCS vs. N=500**	**Δ_{ADE} on AV2 (m)**	**CIS time (h)**
25	1.42 (±0.18)	0.62	2.105 (±0.041)	0.18
50	0.95 (±0.11)	0.81	1.945 (±0.029)	0.35
100 (default)	**0.55** (±0.06)	**0.94**	**1.890** (±0.024)	**0.65**
200	0.35 (±0.04)	0.98	1.882 (±0.022)	1.30
500	0.18 (±0.02)	1.00 (ref)	1.880 (±0.021)	3.20

theoretical bound is only marginally loose at our operating point, validating our reliance on it for sample-size selection rather than running this sweep *a priori*.

Partition stability saturates by N=100. PCS against the N=500 reference rises sharply from 0.62 at N=25 to 0.94 at N=100, with diminishing returns beyond (+0.04 from N=100 to N=200; +0.02 from N=200 to N=500). The N=100 partition differs from the asymptote by at most one or two boundary-layer flips, the same magnitude as the within-architecture cross-dataset PCS slack reported in Table 2 (Qwen2-VL: PCS = 0.92).

Downstream Δ_{ADE} is insensitive past N=100. On AV2 zero-shot CISR, Δ_{ADE} drops from 2.105 m at N=25 to 1.890 m at N=100, then plateaus (1.882 m at N=200; 1.880 m at N=500). The marginal improvement of 0.010 m past N=100 is well within the five-seed std (±0.024 m), so any quality gain from larger N is statistically indistinguishable from noise at our seed budget.

Cost–stability trade-off. N=500 takes $\sim 5\times$ longer than N=100 (3.20 h vs. 0.65 h on a single A10) for a 0.010 m gain in Δ_{ADE} (statistically null) and a +0.06 PCS improvement (within boundary-flip noise). We default to N=100 as the elbow of the cost–stability curve: the partition is within one boundary-flip of the asymptote, the empirical $\hat{\varepsilon}$ matches the Hoeffding bound, and the downstream CISR result is indistinguishable from the asymptotic result. Combined with the cross-dataset transferability of the partition (Table 3; one CIS diagnosis per architecture, amortized over all downstream cities), the operational cost of CIS over the lifetime of a deployed planner is $\sim$ 0.65 h per architecture — a one-time fixed cost, not a per-dataset recurring cost.

C.3 Computational Efficiency

We measure the compute cost of MoLaRx versus the strongest training-cost-comparable baseline (Full LoRA) on each VLM. For every (VLM, method) cell we report three quantities: trainable parameters (selected by the per-zone operator composition), train-time wall-clock per epoch (Phase-2 fine-tuning on the source dataset), and inference latency per sample (target-side forward pass). All numbers are measured on a single NVIDIA A10 (22 GB) GPU at batch size 1, the native processor image size of each VLM, and three temporal frames, with five-seed std reported in parentheses.

Table 10 shows that MoLaRx saves 50–60% of trainable parameters consistently across all three architectures. **Causal decoders (LFM2-VL, Qwen2-VL):** the saving (50–54%) comes from skipping LoRA on Φ_{freeze} — those adapters yield near-zero Δ_{ADE} improvement under Full LoRA (Table 4, "w/o Freeze" is essentially free), so removing them costs no quality. **Prefix-LM decoder (PaliGemma2):** although $\Phi_{\text{tune}}=\varnothing$ leaves no zone with full LoRA, the MMD-driven attention-only LoRA on Φ_{align} avoids the MLP-projection bulk that Full LoRA pays, yielding the largest absolute saving (7.5 M, 60%). Inference latency drops by 19–29 ms relative to Full LoRA on every VLM (8.4–10.9 ms faster than the strongest LoRA-based baseline); on PaliGemma2 the gap reaches 19.3 ms (24%) because Full LoRA must keep all adapters unmerged for bidirectional-attention compatibility, while MoLaRx merges Φ_{freeze}'s weights and removes the MLP adapters on Φ_{align}. In summary, MoLaRx is uniformly faster than Full LoRA at inference and at training, with a quality improvement (Table 1) on every cell.

Table 10: Compute footprint per VLM. "% saved" is the parameter saving over each VLM's Full-LoRA baseline. **Not shown**: a one-time CIS-diagnosis cost of ~ 0.65 h per architecture (App. C.2.4), reused zero-shot across the two benchmarks (Table 3). † On PaliGemma2 the prefix-LM attention yields $\Phi_{\text{tune}}=\varnothing$, so the recipe collapses to MMD-driven attention-only LoRA on Φ_{align} + freeze on Φ_{freeze}. MoLaRx saves 50–60% of trainable parameters across all three architectures; the largest absolute saving (7.5 M, 60%) is on PaliGemma2, where attention-only LoRA on Φ_{align} avoids the MLP-projection bulk that Full-LoRA pays.

VLM	Method	Train. params	% saved	Train (h/ep)	Inference (ms/sample)
LFM2-VL-1.6B	Full-LoRA	7.6 M	0%	4.2 (±0.05)	66.5 (±0.8)
	MoLaRx (CISR)	**3.8 M**	**50%**	**4.1** (±0.04)	**47.4** (±0.6)
Qwen2-VL-2B	Full-LoRA	8.9 M	0%	4.6 (±0.06)	79.8 (±1.1)
	MoLaRx (CISR)	**4.1 M**	**54%**	**4.5** (±0.05)	**50.5** (±0.7)
PaliGemma2-3B	Full-LoRA	12.5 M	0%	5.4 (±0.07)	81.4 (±1.0)
	MoLaRx (CISR)†	**5.0 M**	**60%**	**5.3** (±0.06)	**62.1** (±0.9)

C.4 Inductive-Protocol Control

We compare two adaptation pools for the LFM2-VL/nuScenes cell: the *transductive* pool used in the main-table experiments, i.e. the unlabelled inputs of the target test split, and an *inductive* pool disjoint from it. The target city contains $\sim$12.7k keyframes, of which the test split uses 3,500; the inductive pool is formed from 9,183 unused keyframes (zero overlap with the test split, asserted at run time). We re-run CIS and CISR against this pool with the same seed, schedule, and hyperparameters as the transductive run and evaluate on the untouched test split; for a like-for-like comparison the transductive counterpart is retrained at the same seed, and both are evaluated on the same fixed 300-sample ID and OOD subsets.

Diagnosis. The CIS partition estimated from the disjoint pool is layer-for-layer identical to the transductive one, $\Phi_{\text{align}}=\{0,1\}$, $\Phi_{\text{tune}}=\{2,\ldots,9\}$, $\Phi_{\text{freeze}}=\{10,\ldots,15\}$ (PCS 1.0), although the elbow thresholds are re-estimated independently (τ_{img} 0.30 vs. 0.31; τ_{txt} 3.18 vs. 1.96).

Repair. Table 11 compares the two protocols. Source-only uses no target data and therefore coincides in both rows, which also confirms that the evaluation is deterministic. MoLaRx's target-side error changes by 0.007 m ADE and 0.03 m FDE when the pool–test overlap is removed; its source-side error is 0.14 m ADE higher under the inductive pool. We did not run additional seeds or the remaining DA baselines under this protocol, so we read the result as a single-cell, single-seed stability check rather than a precise protocol effect; it is consistent with the pool providing unlabelled city-level distribution information rather than instance-specific supervision.

Table 11: Inductive-protocol control (LFM2-VL, nuScenes BS→SG; same seed; fixed 300-sample ID/OOD subsets).

Method	Adaptation pool	ADE_{ID}	FDE_{ID}	ADE_{OOD}	FDE_{OOD}
Source-only	none	1.472	2.705	4.466	7.754
MoLaRx	transductive	1.116	2.014	4.266	7.462
MoLaRx	inductive	1.255	2.262	4.273	7.492

C.5 Long-Tail Stress Test on KITScenes-LongTail

Protocol. To probe behaviour beyond nominal cross-city conditions, we evaluate the nuScenes-trained LFM2-VL planners zero-shot on the validation split of KITScenes-LongTail [Wagner et al., 2026]: 100 nine-second long-tail scenarios recorded in Germany at 5 Hz, spanning seven annotated scenario types. We use the front camera only and form the model's three-frame input from the frames at $t=-1.0$, -0.4, and 0.0 s relative to the reference frame (the training spacing of 0.5 s up to 0.1 s of jitter), and resample the expert trajectory to six waypoints at 0.5 s intervals in the ego frame of the reference frame. All planners are source-trained on nuScenes-Boston; Full LoRA uses only the source task loss, whereas Full LoRA + MMD and MoLaRx additionally use unlabelled Singapore inputs for adaptation; none sees KITScenes during training or adaptation. The probe therefore combines

cross-country and long-tail shift relative to the cross-city setting of the main paper, is single-seed and small-scale, and is reported as a preliminary stress test. Evaluation on WOD-E2E [Xu et al., 2025], PDMS under the NAVSIM stack [Dauner et al., 2024], and the official MMS protocol of KITScenes are left to future work.

Displacement. Table 12 reports ADE per scenario type and overall. All three adapted planners obtain lower error than Source-only on this probe (MoLaRx: -1.10 m ADE, -2.06 m FDE), with MoLaRx broadly comparable to Full LoRA while training 50–60% fewer parameters. Absolute errors remain high for every method, especially at night and in overtaking or lane-change scenarios, so the probe does not establish long-tail robustness for any of them.

Table 12: Zero-shot ADE (m) on KITScenes-LongTail by scenario type (LFM2-VL planners trained on nuScenes-Boston; single seed). The last row gives overall ADE / FDE.

Scenario type	n	**Source-only**	**Full LoRA**	**Full LoRA + MMD**	**MoLaRx**
Specifically selected	18	9.47	8.48	9.61	9.91
Heavy rain	10	19.41	17.97	18.95	18.62
Construction zone	6	9.95	7.74	8.31	8.60
Overtake or lane change	20	25.76	24.19	24.14	24.18
Intersection	35	7.19	6.09	5.91	5.77
Night-time	5	42.85	40.63	42.69	41.12
Snow and wintry mix	6	14.98	13.27	13.02	12.93
All (ADE / FDE)	100	14.95 / 25.82	13.58 / 23.44	13.93 / 24.02	13.85 / 23.76

Closest-mode accuracy. Displacement alone does not capture whether a planner selects the right behaviour. KITScenes-LongTail annotates, per scenario, an expert trajectory and alternative (failure-mode) trajectories (`wrong_speed`, `neglect_instruction`), which we resample to the same waypoint grid. We define *closest-mode accuracy* as the fraction of scenarios in which the prediction is strictly closer (in ADE) to the expert trajectory than to any annotated alternative, and the *mode margin* as the ADE to the nearest alternative minus the ADE to the expert, so that positive values indicate correct-mode preference. This measure is inspired by MMS but is not the official protocol. As Table 13 shows, all methods remain weak under the combined shift; adaptation improves closest-mode accuracy by 3–4 points over Source-only, with MoLaRx matching Full LoRA.

Table 13: MMS-inspired closest-mode accuracy and mean mode margin on KITScenes-LongTail (100 scenarios; same zero-shot setting as Table 12).

Method	**Closest-mode acc.** $\uparrow$	**Mean mode margin (m)** $\uparrow$
Source-only	0.25	-1.08
Full LoRA	0.29	-0.98
Full LoRA + MMD	0.28	-1.02
MoLaRx	0.29	-0.98

Planning-oriented proxies on the cross-city splits. On fixed 300-sample ID and OOD subsets of the nuScenes BS$\rightarrow$SG cell we additionally report two planning-oriented proxies computed from the predicted waypoints: the *lateral error at 3 s*, the absolute difference between predicted and ground-truth lateral offset at the final ($t{+}3.0$ s) waypoint, and the *mean absolute jerk* of the predicted trajectory, obtained by third-order finite differences over the six waypoints (origin included) at 0.5 s spacing, with the ground-truth trajectory's jerk as reference. Table 14 shows that on the source side MoLaRx attains the lowest lateral error and a jerk closer to the ground truth than Source-only. On the target side Source-only has the lowest lateral error but a jerk far below the ground truth, suggesting over-smoothed dynamics, whereas the adapted planners produce jerk values closer to the reference at a modest lateral-error cost. These are proxies rather than closed-loop safety metrics: closeness in mean jerk alone does not imply safer trajectories. The re-evaluation is single-seed; on this seed Full LoRA's OOD ADE is slightly below MoLaRx's, and the ADE column of Table 14 comes from this evaluation harness and is not directly comparable to Table 1.

The conversion and evaluation scripts for this section are included in our code release.

Table 14: Planning-oriented proxies on nuScenes BS→SG (LFM2-VL; fixed 300-sample subsets; single seed). Lat.: lateral error at 3 s (m); jerk: mean absolute jerk (m/s^3); the ground-truth jerk is 1.77 (ID) and 2.27 (OOD).

Method	ADE_{ID}	Lat._{ID}	Jerk_{ID}	ADE_{OOD}	Lat._{OOD}	Jerk_{OOD}
Source-only	1.67	0.65	0.77	4.86	0.96	0.87
Full LoRA	1.27	0.56	1.12	4.52	1.16	1.30
Full LoRA + MMD	1.20	0.58	1.73	4.54	1.15	2.04
MoLaRx	1.10	0.52	1.31	4.66	1.14	1.64

C.6 Reverse-Direction Transfer

The main paper evaluates one transfer direction per benchmark (BS→SG, PIT→MIA). Since transfer can be asymmetric, we repeat the pipeline with source and target reversed. The reverse runs reuse the forward code path, hyperparameters, and evaluation protocol: the source planner is retrained on the reversed source city (for AV2, the planner is initialised from the nuScenes-trained visual-grounding checkpoint as in the forward protocol), CIS is re-estimated, the repair strategies are retrained, and evaluation uses fixed 300-sample ID and OOD subsets of the reversed split. These are diagnostic runs: single seed, with the visual-grounding stage truncated from three to approximately two epochs; the reversed source pools are also $\sim 4.7\times$ smaller than the forward ones, so absolute errors are comparable only within a direction. AV2 values come from a separate evaluation harness and are comparable only within their row.

The shift is bidirectional. Source-only incurs a cross-city gap in every reversed cell (Table 16, first row of each block): $+0.60$ m ADE for LFM2-VL and $+1.01$ m for Qwen2-VL on nuScenes SG→BS, and $+0.71$ m for LFM2-VL on AV2 MIA→PIT. The visual-grounding prerequisite (Appendix B) also persists: in the Qwen2-VL cell, source-side ADE rises by 197–252% under blank-image ablation for Source-only, Full LoRA, and MoLaRx.

Partition stability under reversal. Table 15 compares the partition diagnosed in the reversed direction with the forward one of Table 5. For LFM2-VL the partition is layer-for-layer identical on both benchmarks and in a full-schedule rerun of the nuScenes cell, as it was under the disjoint adaptation pool of Appendix C.4; a same-recipe replicate of the nuScenes run on different hardware shifted two zone boundaries by one layer ($\{0\}/\{1,\dots,8\}/\{9,\dots,15\}$), the boundary sensitivity noted in Limitation (i). PaliGemma2 agrees on 24 of 26 layers and keeps its image-shift-dominant signature; its two deepest layers cross τ_{txt} in reverse, which the zone-wise Jaccard scores as 0 for the two near-empty zones. Qwen2-VL keeps the three-zone ordering (align shallowest, tune mid-depth, freeze deepest) with boundary drift (mean PCS 0.52, against a random-partition chance level of ≈ 0.18). The ranking of stability across architectures matches the within-architecture ranking of Table 2, supporting, though not by itself establishing, a primarily architecture-dependent partition.

Table 15: Zone partition under reversed transfer direction (single seed), written as $\Phi_{\text{align}}/\Phi_{\text{tune}}/\Phi_{\text{freeze}}$. PCS is the zone-wise Jaccard against the forward partition; "agree" counts layers assigned to the same zone.

VLM	Cell	Reverse partition	Agree	$\text{PCS}_{\text{align}}$	PCS_{tune}	$\text{PCS}_{\text{freeze}}$	Mean
LFM2-VL	SG→BS	$\{0,1\}/\{2{:}9\}/\{10{:}15\}$	16/16	1.00	1.00	1.00	1.00
LFM2-VL	MIA→PIT	$\{0,1\}/\{2{:}9\}/\{10{:}15\}$	16/16	1.00	1.00	1.00	1.00
Qwen2-VL	SG→BS	$\{0,1,17\}/\{2{:}15\}/\{16,18{:}27\}$	20/28	0.40	0.50	0.65	0.52
PaliGemma2	SG→BS	$\{0{:}23\}/\{24,25\}/\varnothing$	24/26	0.96	0.00	0.00	0.32

Repair comparison. Table 16 reports ADE for Source-only, Full LoRA, and MoLaRx in the three reversed cells. MoLaRx improves over Source-only on both sides in all three cells. Between MoLaRx and Full LoRA there is no consistent winner: the nuScenes Δ_{ADE} differences are 0.03–0.07 m with opposite signs across the two VLMs, while absolute OOD ADE favours Full LoRA in both; on AV2 MoLaRx is 0.40 m lower on ID while Full LoRA is 0.53 m lower on OOD. A same-recipe replicate of the LFM2-VL cell on different hardware flips the ordering (MoLaRx 4.13 vs. Full LoRA 4.47 m OOD ADE), and the largest same-method OOD difference between the two runs

(0.86 m) exceeds the margins between the methods; a subsequent full-schedule (three-epoch) rerun of the same cell reproduced the partition but not the target-side gain over Source-only (OOD ADE 4.10 / 4.14 / 4.28 m for Source-only / Full LoRA / MoLaRx). The reverse evidence therefore does not support a reliable ranking between MoLaRx and Full LoRA; MoLaRx performs comparably while training 50–60% fewer parameters. In the replicate, uniform MMD-only adaptation collapses as in the forward direction (ID 5.74 m ≈ OOD 5.82 m), and its blank-image ID ADE (5.37 m) is no worse than with real images, providing no evidence that it restores visual dependence. Full-schedule, multi-seed reverse runs for all cells remain future work.

Table 16: Reverse-direction repair comparison: ADE (m) on fixed 300-sample subsets, ID / OOD (Δ_{ADE}). Single seed; visual grounding truncated to approximately two epochs in the nuScenes cells; AV2 values are from a separate harness and comparable only within the block.

Method	LFM2-VL, SG→BS	Qwen2-VL, SG→BS	LFM2-VL, MIA→PIT (AV2)
Source-only	3.52 / 4.12 (0.60)	3.29 / 4.30 (1.01)	10.90 / 11.61 (0.71)
Full LoRA	2.56 / 3.61 (1.05)	2.78 / 4.11 (1.33)	5.12 / 4.86 (−0.26)
MoLaRx	2.79 / 3.90 (1.12)	2.89 / 4.19 (1.30)	4.72 / 5.39 (0.67)

D Method Details

This section gives the technical details deferred from §4.2: the multi-scale MMD applied on Φ_{align} (§D.1), the LoRA factorisation applied on Φ_{tune} (§D.2), the architecture-adaptive operator composition that handles the prefix-LM regime (§D.3), and the full procedure consolidated as Algorithm 1 together with the rationale for the zone-matched design (§D.4).

D.1 MMD on Φ_{align}

The MMD operator's role is to align the marginal distributions of *image-token representations* between source and target domains at layers where CIS identifies the domain shift as image-channel-dominated. We adopt MMD [Gretton et al., 2012] rather than adversarial alignment (DANN [Ganin et al., 2016], CDAN [Long et al., 2018]) or correlation matching (CORAL [Sun et al., 2016]) for three reasons: (i) MMD is a closed-form, gradient-tractable distance between two empirical distributions in a reproducing kernel Hilbert space (RKHS), so no auxiliary discriminator network or min-max optimisation is required; (ii) the multi-scale RBF variant is consistent (zero iff distributions match) without bandwidth tuning; (iii) MMD applies on raw representations, so it composes cleanly with the per-zone operator routing in §D.3.

Formally, the MMD operator on Φ_{align} uses the modality-m mean-pool of the row-block $\mathbf{H}^{(l)}_{\mathcal{M}_m}(\mathbf{x};\Theta')$ from Definition 2,

$$\mathbf{h}^{(l)}_m(\mathbf{x};\Theta') \;=\; \frac{1}{|\mathcal{M}_m(\mathbf{x})|}\sum_{i\in\mathcal{M}_m(\mathbf{x})}\mathbf{H}^{(l)}_{i,:}(\mathbf{x};\Theta') \;\in\; \mathbb{R}^d, \tag{25}$$

where $\mathcal{M}_m(\mathbf{x})$ is the index set of modality-m tokens (image patches if m=img; text positions if m=txt), $\mathbf{H}^{(l)}_{i,:}(\mathbf{x};\Theta') \in \mathbb{R}^d$ is the i-th row of the layer-l hidden-state matrix under parameters Θ', and d is the model's hidden dimension. The mean-pool reduces the variable-size row-block to a single d-vector per (modality, layer). We then collect the per-sample pools over a minibatch $\mathcal{B}$ into

$$\bar{\mathbf{H}}^{(l)}_m(\mathcal{B};\Theta') \;:=\; \{\,\mathbf{h}^{(l)}_m(\mathbf{x};\Theta') \,:\, \mathbf{x}\in\mathcal{B}\,\} \;\in\; \mathbb{R}^{|\mathcal{B}|\times d}, \tag{26}$$

which is the batch-level object input to MMD^2 in Eq. 14. We apply MMD on $\bar{\mathbf{H}}^{(l)}_{\text{img}}$ for $l \in \Phi_{\text{align}}$.

MMD as RKHS distance. For any reproducing kernel Hilbert space $\mathcal{H}_k$ with kernel k, MMD admits the closed-form

$$\mathrm{MMD}^2(P,Q;k) \;=\; \big\|\,\mu_P - \mu_Q\,\big\|^2_{\mathcal{H}_k}, \quad \text{where} \quad \mu_P \;=\; \mathbb{E}_{X\sim P}[\,k(X,\cdot)\,] \in \mathcal{H}_k \tag{27}$$

is the kernel mean embedding of distribution P. When k is *characteristic* (e.g., RBF), this map is injective and $\mathrm{MMD}^2(P,Q;k) = 0 \iff P = Q$, so minimising the empirical MMD drives the source and target representations to the same distribution in the limit.

Empirical multi-scale RBF MMD. Let $\mathcal{B}_s = \{\mathbf{x}_s^i\}_{i=1}^B \subseteq \mathcal{S}_s$ and $\mathcal{B}_t = \{\mathbf{x}_t^j\}_{j=1}^B \subseteq \mathcal{S}_t$ be source and target minibatches of size B. Substituting the empirical mean embeddings into Eq. 27 gives the U-statistic estimator

$$\begin{aligned}\mathrm{MMD}^2\Big(\bar{\mathbf{H}}_{\text{img}}^{(l)}(\mathcal{B}_s;\Theta'),\, \bar{\mathbf{H}}_{\text{img}}^{(l)}(\mathcal{B}_t;\Theta');\, k\Big) \;=\; & \frac{1}{B^2}\sum_{i,j} k\Big(\mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_s^i;\Theta'),\, \mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_s^j;\Theta')\Big) \\ +\; & \frac{1}{B^2}\sum_{i,j} k\Big(\mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_t^i;\Theta'),\, \mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_t^j;\Theta')\Big) \\ -\; & \frac{2}{B^2}\sum_{i,j} k\Big(\mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_s^i;\Theta'),\, \mathbf{h}_{\text{img}}^{(l)}(\mathbf{x}_t^j;\Theta')\Big), \end{aligned} \tag{28}$$

where the first two sums measure intra-distribution self-similarity and the third measures cross-distribution similarity; the difference vanishes when source and target distributions are identical (modulo finite-sample noise). We use a multi-scale RBF kernel with adaptive bandwidth:

$$k(\mathbf{u},\mathbf{v}) \;=\; \sum_{\sigma\in\mathcal{S}} \exp\!\left(-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}\right), \qquad \mathcal{S} \;=\; \{\,0.5\,\mu_d,\; \mu_d,\; 2\,\mu_d\,\}, \tag{29}$$

where $\sigma \in \mathcal{S}$ is a Gaussian bandwidth and μ_d is the *median heuristic*—the median of all pairwise Euclidean distances $\|\mathbf{u}-\mathbf{v}\|_2$ within the minibatch. The mixture of three bandwidths makes the estimator robust to misspecified scale across layers (where the typical magnitude of $\mathbf{h}_{\text{img}}^{(l)}$ varies), avoiding the brittleness of single-bandwidth choices. To keep the operator parameter-efficient, the alignment is driven by rank-r LoRA adapters attached to the attention projections only (Q, K, V, O) of each $l \in \Phi_{\text{align}}$; gradients of $\mathcal{L}_{\text{MMD}}$ flow through these adapters (see Eq. 31). Restricting to attention is principled: image-token shift propagates via cross-modal Q-K-V interactions, while MLPs are point-wise modality-agnostic feature maps that the F1 finding does not implicate.

D.2 LoRA on Φ_{tune}

The LoRA operator's role is to apply a parameter-efficient correction to the layers where CIS identifies the domain shift as text-channel-dominated. We adopt LoRA [Hu et al., 2022] over full fine-tuning to (i) preserve the source-fine-tuned baseline as the initialisation by writing the update as a perturbation $\mathbf{BA}$ that starts at zero, (ii) limit trainable parameters to $r(d_{\text{in}} + d_{\text{out}})$ per projection so the small target sample budget does not overfit, and (iii) compose cleanly with MMD on Φ_{align} and freeze on Φ_{freeze} via per-zone gradient masking (Eq. 31). LoRA is the de facto PEFT default for fine-tuning large language and multimodal models; selectively attaching it to Φ_{tune} rather than all layers (the Full LoRA baseline) is what differentiates CISR from generic PEFT.

Factorisation. For each layer $l \in \Phi_{\text{tune}}$ we attach LoRA adapters to every linear projection within the layer—attention projections $\{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o\}$ and feed-forward / convolution projections $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_{\text{in}}\}$. For each such projection $\mathbf{W}_0 \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, LoRA parameterises the post-fine-tune weight as a rank-r additive update,

$$\mathbf{W} \;=\; \mathbf{W}_0 \;+\; \frac{\alpha}{r}\,\mathbf{B}\,\mathbf{A}, \qquad \mathbf{A}\in\mathbb{R}^{r\times d_{\text{in}}},\;\; \mathbf{B}\in\mathbb{R}^{d_{\text{out}}\times r}, \tag{30}$$

where $\mathbf{W}_0$ is the frozen base-weight matrix, $\{\mathbf{A},\mathbf{B}\}$ are the trainable low-rank factors, r=16 is the LoRA rank (the inner dimension of the factorisation), α=32 is the LoRA scaling factor (which decouples the magnitude of the update from the rank), and $d_{\text{in}}, d_{\text{out}}$ are the input and output dimensions of the projection.

We follow the standard LoRA initialisation: $\mathbf{A} \sim \mathcal{N}(0, \sigma_A^2\mathbf{I})$ with small Gaussian variance, and $\mathbf{B} = \mathbf{0}$. Because $\mathbf{BA} = \mathbf{0}$ at initialisation, the initial forward pass coincides exactly with the source-fine-tuned baseline; training departs from this baseline only in the directions where the target task gradient pulls. Backpropagating only through $(\mathbf{A},\mathbf{B})$ keeps the trainable parameter count to $r(d_{\text{in}} + d_{\text{out}})$ per projection (vs. the full $d_{\text{in}} \cdot d_{\text{out}}$ for full fine-tuning).

The domain shift on Φ_{tune} acts along a low-dimensional subspace—the projection of hidden states onto numerical-magnitude directions (ego speed, following distance, lead-vehicle velocity) where source and target distributions differ—rather than as a full-rank drift. A rank-r correction with $r \ll d$ therefore captures the relevant component of the shift while preserving the source-side computation.

This empirical low-rank structure matches the rationale behind LoRA in language models [Hu et al., 2022]: the intrinsic dimension of task-specific adaptation is small, so a rank-r correction with $r \ll d$ suffices.

D.3 Adaptive operator composition

The role of the adaptive composition is to wire the three zone-specific operators (MMD on Φ_{align}, LoRA on Φ_{tune}, freeze on Φ_{freeze}) into a single training objective, with gradients routed by zone, and to handle the regime where one of the zones is empty (the prefix-LM case, $\Phi_{\text{tune}}=\varnothing$ on PaliGemma2). The closest baseline in spirit is surgical fine-tuning [Lee et al., 2022], which selectively unfreezes layers based on a manual shift-type heuristic (input-shift $\rightarrow$ early, feature-shift $\rightarrow$ middle, output-shift $\rightarrow$ late). Our composition differs in two ways: (i) zone selection is data-driven via CIS rather than heuristic, and (ii) the operator type per zone (MMD vs LoRA vs freeze) is matched to the type of shift CIS measures, rather than uniformly applying LoRA-then-freeze.

Gradient flow by zone. The total objective $\mathcal{L}_{\text{total}}$ (Eq. 16) routes gradients to three disjoint parameter sets, one per zone:

$$\nabla \mathcal{L}_{\text{total}} \;=\; \begin{cases} \lambda_{\text{MMD}} \nabla_{\{\mathbf{A}_l, \mathbf{B}_l\}} \mathcal{L}_{\text{MMD}} & \text{for attention-only LoRA params on } l \in \Phi_{\text{align}}, \\ \nabla_{\{\mathbf{A}_l, \mathbf{B}_l\}} \mathcal{L}_{\text{task}} & \text{for LoRA parameters (all projections) on } l \in \Phi_{\text{tune}}, \\ \mathbf{0} & \text{for all parameters on } l \in \Phi_{\text{freeze}}. \end{cases} \tag{31}$$

Here $\{\mathbf{A}_l, \mathbf{B}_l\}$ are the LoRA factors at layer l (Eq. 30); on Φ_{align} they are attached only to the attention projections (Q, K, V, O), and on Φ_{tune} they are attached to all linear projections (attention + MLP). $\mathcal{L}_{\text{MMD}}$ is the per-batch image-token MMD aggregated over Φ_{align}, $\mathcal{L}_{\text{task}}$ is the trajectory-prediction loss on labelled source data, and $\lambda_{\text{MMD}} = 0.5$ is the loss weight. Target-domain samples contribute to $\mathcal{L}_{\text{MMD}}$ without labels — only their hidden states appear in the cross-distribution kernel term — so CISR remains an unsupervised DA method.

Active operator set and 2-operator degeneration. The operator-to-zone assignment is not a fixed recipe: it is driven by which zones the CIS diagnosis actually discovers in $f_{\Theta'}$. Let $\mathcal{O} = \{\text{MMD}, \text{LoRA}, \text{freeze}\}$ be the three candidate operators and let $\pi : \{\text{align}, \text{tune}, \text{freeze}\} \rightarrow \mathcal{O}$ be the fixed role map $\pi(\text{align})=\text{MMD}$, $\pi(\text{tune})=\text{LoRA}$, $\pi(\text{freeze})=\text{freeze}$. The *active operator set* selected by CISR is

$$\mathcal{O}_{\text{active}}(f_{\Theta'}) \;=\; \{\, \pi(z) \;:\; z \in \{\text{align}, \text{tune}, \text{freeze}\} \text{ and } \Phi_z \neq \varnothing \,\}, \tag{32}$$

where Φ_z is the zone-z partition discovered by CIS (Eqs. 11–13). When all three zones are non-empty (the *causal-decoder regime* on LFM2-VL and Qwen2-VL), Eq. 32 yields the full 3-operator instantiation MMD + LoRA + freeze whose objective and gradient flow are Eqs. 16 and 31. When $\Phi_{\text{tune}} = \varnothing$ (the *prefix-LM regime* on PaliGemma2), the LoRA operator has no layers to attach to: $\Delta = \varnothing$, and $\mathcal{L}_{\text{task}}(\Theta', \varnothing) = \mathcal{L}_{\text{task}}(\Theta', \mathbf{0})$ is constant in the optimisation variables. The total objective therefore degenerates to

$$\mathcal{L}^{(\text{2-op})}_{\text{total}}(\Theta') \;=\; \lambda_{\text{MMD}} \cdot \mathcal{L}_{\text{MMD}}(\mathcal{B}_s, \mathcal{B}_t), \tag{33}$$

i.e. CISR automatically becomes MMD-on-Φ_{align} + freeze-on-Φ_{freeze}, without any user-side recipe change. Concretely, $|\mathcal{O}_{\text{active}}| = 3$ on causal-attention VLMs and $|\mathcal{O}_{\text{active}}| = 2$ on prefix-LM VLMs; experiments in §5.2 validate both regimes.

D.4 Full procedure and zone-matched rationale

The three operators are matched to the type of shift that CIS measures in each zone, and each of the three matches is necessary: **(i) MMD on Φ_{align}** aligns two visual-appearance distributions via a rank-r LoRA on attention projections: source and target features become statistically indistinguishable on these layers, while the MLP-level feature transformation is preserved (no MLP adapter attached). Since image-channel shift is concentrated here and is high-dimensional, distributional alignment is the right tool. **(ii) LoRA on Φ_{tune}** (when this zone is non-empty) adds a low-rank correction that is sufficient for the numerical-distribution shift but small enough to avoid overfitting to the source. Since text-channel shift acts along a low-dimensional subspace, a parameter-efficient update is sufficient. **(iii) Freeze on Φ_{freeze}** preserves the task-critical invariant computation that cannot be improved by adaptation. Updating these layers (e.g., via LLRD [Howard and Ruder, 2018] or full LoRA) only

adds noise and overfitting risk; freezing them is strictly better. The assignment is architecture-aware by construction: the 3-operator instantiation emerges on causal decoders where all three zones are non-empty, and the 2-operator instantiation (MMD + freeze) emerges on prefix-LM decoders where the text-dominated zone collapses (Eq. 33). Applying any single operator uniformly across all layers is strictly worse than this matched assignment; in the extreme case of global MMD (MMD-only ablation), the result is catastrophic (§5.4).

Full procedure. Algorithm 1 consolidates the equations from §4.1–§4.2 into a single executable procedure. Lines 1–3 implement CIS (locate); line 4 sets up CISR's per-zone parameter masks; the inner loop on lines 5–9 implements CISR (repair). On a prefix-LM VLM ($\Phi_{\text{tune}} = \varnothing$), line 4's LoRA-attachment step is a no-op and the $\mathcal{L}_{\text{task}}$ term contributes no gradient, so the algorithm transparently realises the 2-operator degeneration (Eq. 33) without any user-side change. Computational cost: lines 1–3 (CIS diagnosis) require $2 \cdot N \cdot L$ partial forward passes with one swap per pass, totalling ~ 0.65 h on a single A10 GPU at $N{=}100$, $L{\approx}20$ (App. C.2.4); the CISR inner loop (lines 5–9) has the same per-step cost as standard LoRA fine-tuning plus one extra forward pass per layer in Φ_{align} for the MMD term.

Algorithm 1 MoLaRx: Locate-then-Repair for VLM Trajectory Planning

Require: fine-tuned planner $f_{\Theta'}$ (Appendix B); labelled source $\mathcal{S}_s$; unlabelled target $\mathcal{S}_t$; thresholds $\tau_{\text{img}}, \tau_{\text{txt}}$; weight λ_{MMD}; paired-sample budget N.

1: **CIS, sample.** Draw N scene-matched pairs from $\mathcal{P}_{\text{paired}}$ (Definition 3).
2: **CIS, score.** For each $(m, l) \in \{\text{img}, \text{txt}\} \times \{0, \ldots, L{-}1\}$, compute $\widehat{\mathbf{\Psi}}[m, l]$ via Eq. 9.
3: **CIS, partition.** Form $\Phi_{\text{align}}, \Phi_{\text{tune}}, \Phi_{\text{freeze}}$ via Eqs. 11 to 13.
4: **CISR, setup.** Attach rank-r LoRA on Φ_{align} (attention only, MMD-driven) and on Φ_{tune} (all projections, task-driven) per Eq. 30; zero-mask all gradients on Φ_{freeze} (Eq. 31).
5: **for** each training step **do**
6: Sample labelled batch $\mathcal{B}_s \sim \mathcal{S}_s$ and unlabelled batch $\mathcal{B}_t \sim \mathcal{S}_t$.
7: Compute $\mathcal{L}_{\text{task}}(\Theta', \Delta;\, \mathcal{B}_s)$ (Eq. 15) and $\mathcal{L}_{\text{MMD}}(\Theta';\, \mathcal{B}_s, \mathcal{B}_t)$ (Eq. 14).
8: Backpropagate $\mathcal{L}_{\text{total}}$ (Eq. 16) with the per-zone gradient masking of Eq. 31.
9: **end for**
10: **return** domain-adapted planner $f^* := f_{\Theta' + \Delta^*}$.

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 Section 1 state four contributions—Discovery (cross-city domain shift has a modality–layer structure, architecture-dependent and dataset-stable), Tool (CIS, an activation-patching diagnostic of per-(modality, layer) causal effects on predicted trajectories), Method (MoLaRx: locate-then-repair that routes existing operators via CIS+CISR), and Experiments—and the headline numbers (lowest Δ_{ADE} among adaptation methods in all six main-table settings, 50–60% fewer trainable parameters, 8.4–10.9 ms lower inference latency, uniform MMD-only adaptation collapses in all six settings) match the results in Section 5 and Tables 1, 3.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Section 6 discusses four limitations: (i) the hard-threshold zone partition can produce one or two boundary-layer flips on tightly-shifted benchmarks; (ii) generalisation to encoder–decoder VLMs (BLIP-2, PaLI), cross-attention fusion architectures, and non-driving tasks with non-geographic shifts remains untested; (iii) the main evaluation focuses on nominal driving conditions and displacement metrics, official planning-oriented metrics are left to future work, and the preliminary long-tail probe (Appendix C.5) leaves long-tail robustness unestablished; and (iv) CIS requires scene-matched source/target pairs drawn predominantly from prevalent scenes, its estimates depend on pairing quality, and the three-zone structure is empirical with bootstrap stability but no theoretical universality guarantee.

   Guidelines:

   - The answer [N/A] 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: The only formal result is the Hoeffding-style concentration bound on the empirical CIS estimator $\widehat{\mathbf{\Psi}}[m, l]$ (Eq. 10 in Section 4.1). The required assumptions—bounded per-waypoint distance R between any two predicted trajectories and i.i.d. scene-matched source/target pairs—are stated in the surrounding text; instantiating with $R \approx 20\,\text{m}$, $N{=}100$, and failure probability $\eta{=}0.05$ yields a 95% half-width $\varepsilon \leq 0.6\,\text{m}$. The empirical N-budget sweep in Appendix C.2.4 confirms the bound is tight at the operating point ($\hat{\varepsilon} \approx 0.55\,\text{m}$).

   Guidelines:

   - The answer [N/A] 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: Section 4 fully describes the two MoLaRx modules (CIS in §4.1, CISR in §4.2) with all equations; the experimental setup, datasets, and metrics are documented in Section 5 and Appendices A.1, A.3, A.4, and B; the evaluation datasets (nuScenes and Argoverse 2) are publicly available and cited; full pseudocode for the CIS+CISR pipeline is given as Algorithm 1 in Appendix D.4; code is released at the URL provided in the abstract.

   Guidelines:

   - The answer [N/A] 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: Code with reproducibility instructions (README) is released at `https://github.com/ZhihongCui/MoLaRx` (cited in the abstract); the two evaluation datasets, nuScenes and Argoverse 2, are publicly available research datasets, accessed under their respective non-commercial research licenses.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

Justification: All training and evaluation details are documented in Appendix A.1 (full hyperparameter table) and Appendix B (Phase-0/Phase-2 fine-tuning protocol). Key choices: AdamW optimizer (8-bit on PaliGemma2, fp32 elsewhere), LoRA r=16, α=32, learning rate 2×10^{-5}, 1 Phase-2 epoch, bfloat16 precision; CIS uses N=100 scene-matched pairs (justified by the Hoeffding bound and the empirical N-budget sweep in Appendix C.2.4); $\tau_{\text{img}}, \tau_{\text{txt}}$ thresholds and their sensitivity are reported in Appendix C.2.1; data splits are the official nuScenes train/val and Argoverse 2 motion-forecasting splits (§5.1).

Guidelines:

- The answer [N/A] 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: All main-paper numbers are averaged over five random seeds $\{0, 1, 2, 3, 4\}$ (Section 5; Appendix A.1). Tables 1 and 3 report five-seed standard deviations in parentheses for ADE, FDE, Δ_{ADE}, and PE; the variability captured is from training-time random initialisation under fixed data splits. The supplementary experiments in Appendices C.4–C.6 (inductive-protocol control, long-tail stress test and planning proxies, reverse-direction transfer) are single-seed diagnostic runs; they are reported without error bars and their conclusions are stated accordingly.

Guidelines:

- The answer [N/A] 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: All experiments run on a single NVIDIA A10 (22 GB) GPU at batch size 1, image resolution 512×512, three temporal frames (Section 5.1; Appendix C.3). Per-architecture costs are reported in Appendix C.3: CIS diagnosis ~ 0.65 h (one-time, amortised across downstream cities); per-epoch Phase-2 train wall-clock and inference latency

per sample (MoLaRx: 47.4–62.1 ms; 8.4–10.9 ms faster than the strongest LoRA-based baselines) reported with five-seed std.

Guidelines:

- The answer [N/A] 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 conforms to the NeurIPS Code of Ethics. We use only publicly available driving datasets (nuScenes, Argoverse 2) under their original non-commercial research licenses, and the work involves no human subjects. The safety-critical nature of trajectory planning is acknowledged, and the limitations section flags that closing the cross-city gap on benchmark trajectories is necessary but not sufficient for real-world deployment, which requires additional validation and human oversight beyond the scope of this paper.

Guidelines:

- The answer [N/A] 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: Positive impact: by reducing the cross-city trajectory-error gap with 50–60% fewer trainable parameters, MoLaRx lowers the data-collection and compute cost of redeploying VLM-based planners to new cities, supporting more equitable geographic coverage of driving-AI research. Negative impact: VLM planners adapted from a single source city may inherit dataset biases (sensor coverage, scene composition, traffic-rule conventions) that surface when deployed in cities outside the studied benchmarks; reduced numerical gap on a benchmark does not guarantee operational safety in deployment. These concerns are flagged in the limitations (Section 6, item ii).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release new high-risk pre-trained models, image generators, or scraped datasets. The released code is a research framework that operates on three publicly available pre-trained VLMs (LFM2-VL-1.6B, Qwen2-VL-2B, PaliGemma2-3B) and two publicly available benchmark datasets; the LoRA adapters trained for each (VLM, dataset) cell are released only to support reproduction of the reported numbers and inherit the upstream license of the corresponding base VLM.

    Guidelines:

    - The answer [N/A] 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: All existing assets are properly cited and used under their original licenses. Datasets: nuScenes [Caesar et al., 2020] and Argoverse 2 [Wilson et al., 2023], used under their non-commercial research licenses; KITScenes-LongTail [Wagner et al., 2026] (validation split only, Appendix C.5), used under CC BY-NC 4.0 via gated access on Hugging Face. VLM backbones: LFM2-VL-1.6B [Amini et al., 2025], Qwen2-VL-2B [Wang et al., 2024], and PaliGemma2-3B [Beyer et al., 2024], used under their respective open-weight licenses. Baseline methods (LoRA, MMD, DANN, LLRD, Freeze-Last-K, ROME, Surgical FT) and software libraries are cited at first mention in Sections 5 and Appendix A.4.

    Guidelines:

    - The answer [N/A] 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: The released MoLaRx code and the per-(VLM, dataset) LoRA adapter checkpoints are the only new assets; documentation, environment specification, and reproducibility instructions (README) are provided alongside the code at the URL given in the abstract.

    Guidelines:

    - The answer [N/A] 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: This work does not involve crowdsourcing or research with human subjects; all evaluation uses public driving benchmark datasets (nuScenes, Argoverse 2) collected and released by their original authors.

    Guidelines:

    - The answer [N/A] 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: This work does not involve human subjects and therefore does not require IRB approval.

    Guidelines:

- The answer [N/A] 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: [Yes]

    Justification: The paper studies VLMs with LLM backbones as the *adapted target*, not as a method tool: MoLaRx applies CIS+CISR to three open-weight VLMs—LFM2-VL-1.6B [Amini et al., 2025], Qwen2-VL-2B [Wang et al., 2024], and PaliGemma2-3B [Beyer et al., 2024]—to study and repair cross-city domain shift. The CIS diagnostic and the CISR repair operators (MMD / LoRA / freezing) do not invoke any external LLM at training or inference time; backbone identifiers, training-time fine-tuning settings, and decoding configuration are documented in Appendix A.1 and Appendix B.

    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.