E2M-DISTILL: DISTILLING END-TO-END DRIVING MODELS INTO MOTION-ONLY SIM AGENTS FOR LARGE-SCALE AUTONOMOUS DRIVING SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Widespread deployment of autonomous vehicles (AVs) may introduce unexpected system-level effects that require evaluation of interactions among multiple vehicles, so large-scale autonomous driving simulation is needed to understand their urban-mobility impact. Existing approaches face a trade-off: faithfully simulating real autonomous driving stacks from perception to planning is prohibitively expensive, while motion-only simulation bypasses the perceptual limitations of a target AV stack. We formulate this gap as *AV-Stack-to-Sim-Agent Conversion*: given a target AV stack, derive a motion-only sim agent (consuming only privileged simulator state, without rendering) whose closed-loop rollouts preserve the target stack's behavior. This task exhibits an *Inverted Learner-Expert Asymmetry* (ILEA): the teacher acts on partial sensor perception while the student sees privileged state, so conversion must account for the teacher's perceptual limitations. We propose **E2M-Distill** (E2E-to-Motion-only sim agent Distillation), a distillation framework that addresses ILEA. E2M-Distill places a Perception Contraction Module (PCM) ahead of the retained teacher planning head: PCM contracts the student's privileged simulator-state features toward the teacher's perception-conditioned representation. An interface-alignment loss anchors the contracted features to a teacher-side activation selected by the Perception–Planning Score (PPS). We introduce three closed-loop consistency metrics for this task and demonstrate across the evaluated target stacks that E2M-Distill improves behavioral fidelity while reducing the per-agent computational bottleneck for large-scale AV simulation. Code: `https://anonymous.4open.science/r/E2M-Distill`.

1 INTRODUCTION

Autonomous-driving (AV) technology is rapidly maturing, moving from isolated single-vehicle prototypes toward city-scale deployment. This shift raises system-level questions—how AVs reshape urban traffic efficiency and safety, and how future traffic-management policies should be designed around them—that call for controlled, repeatable evaluation before deployment. Simulation is therefore indispensable for anticipating such impacts before deployment, enabling repeatable evaluation, stress testing, and algorithm and policy optimization at a fraction of the cost of real-world testing (Hu et al., 2024; Jia et al., 2024; Yang et al., 2024; Yu et al., 2025a; Humnabadkar et al., 2026). Prior AV simulation has largely centered on single-vehicle (ego) testing, where the deployed AV stack can be dropped directly into a simulator to evaluate ego-level driving performance under controlled scenarios. Such ego-centric setups, however, provide limited evidence about interactions and aggregate outcomes under multi-AV deployment, motivating large-scale AV simulation. For these studies, simulated agents should retain behavior associated with the target stacks. However, existing choices expose a trade-off. Running the full target AV stack for every controlled vehicle preserves the target policy, but sensor-based stacks require rendering observations and executing the perception–prediction–planning loop at every step, making simulation computationally prohibitive (Noguchi & Yamamoto, 2026; Gao et al., 2026). Motion-only simulators (Igl et al., 2022; Gulino et al., 2023; Feng et al., 2023; Zhou et al., 2024; Wu et al., 2024), which act directly on privileged simulator state, sidestep this rendering loop at much lower per-step cost, but in doing so they discard both the perceptual limitations and the stack-specific planning logic that govern how a deployed AV actually behaves. This leaves a practical bottleneck between *behaviorally faithful* and *computationally scalable* AV simulation.

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

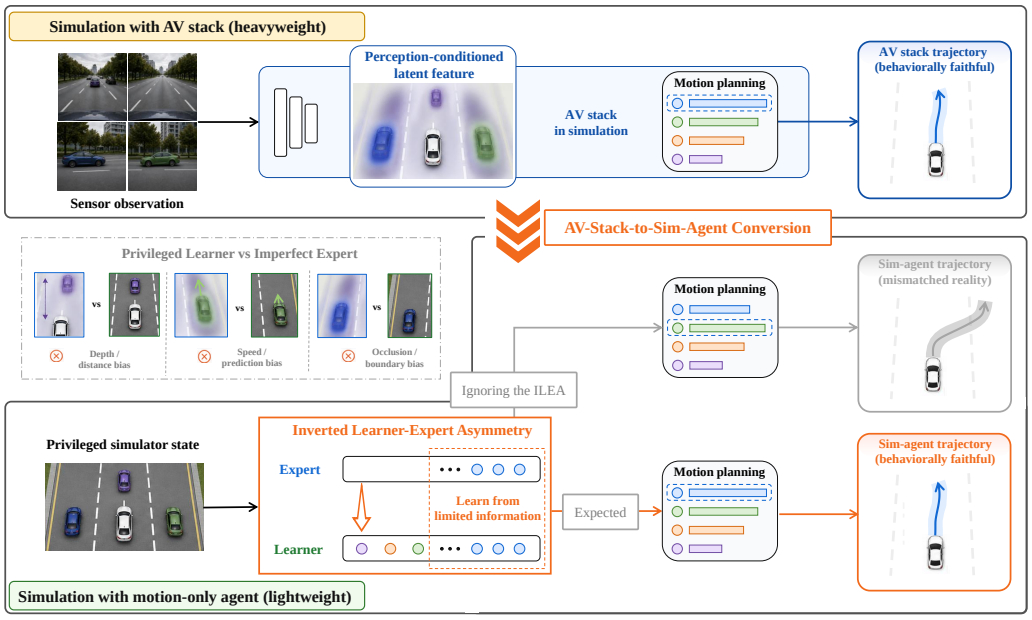


Figure 1: Overview of AV-Stack-to-Sim-Agent Conversion: derive a sim agent that preserves the target stack's perception-shaped closed-loop behavior while running at motion-only simulation cost.

We accordingly formulate this bottleneck as a conversion problem that decouples the cost of perception-faithful decision making from the per-step cost of large-scale rollout: the expensive perception–planning behavior of the target stack is captured once, offline, by a learned surrogate that is then deployed many times in cheap simulation. In this formulation, which we call *AV-Stack-to-Sim-Agent Conversion*, we are given a target AV stack—the full perception–planning pipeline of a driving algorithm to be studied at scale—and we seek a corresponding sim agent: a simulator-native motion policy that operates only on privileged simulator state, without running the full-stack inference loop. We assess fidelity through paired closed-loop teacher–student rollouts.

In this task, the difficulty is an *Inverted Learner-Expert Asymmetry* (ILEA): the teacher is the sensor-based AV stack, acting through noisy sensor observations and perception-conditioned latent features—intermediate features that inherit the noise and biases of the teacher's perception modules—while the student receives privileged simulator state. The student has direct access to scene geometry, while the teacher's behavior also depends on sensor-specific cues. With naïve distillation, the student may rely on state cues that the teacher could not reliably perceive, obtain low open-loop trajectory error, and nevertheless diverge from the teacher in closed-loop interaction. Therefore, we constrain the student to reproduce the teacher's perception-conditioned behavior during conversion.

We introduce **E2M-Distill** (E2E-to-Motion-only sim agent Distillation), a distillation framework that addresses ILEA when performing AV-Stack-to-Sim-Agent Conversion in the sensor-based E2E setting. E2M-Distill follows a shared conversion recipe with model-specific adapters (*privileged simulator state* $\rightarrow$ *MLP embeddings* $\rightarrow$ *Perception Contraction Module (PCM)* $\rightarrow$ *retained teacher planning head*) that removes the perception front end while preserving the teacher's planning interface. Within PCM, a Privileged Perception Degradation (PPD) applies input-dependent gating and learned noise to the student's privileged simulator-state features, an Interface Adapter (IA) maps the contracted features into the planning head's expected distribution, and an interface-alignment loss anchors PCM to a teacher-side activation at the perception–planning interface. To localize this interface, we introduce the *Perception–Planning Score* (PPS) for selecting the teacher target activation that PCM should align to. We instantiate this recipe on two architecturally distinct teachers, SparseDrive (Sun et al., 2025) and VAD (Jiang et al., 2023), to validate E2M-Distill's conversion fidelity and efficiency across different E2E designs.

For this task, we propose three closed-loop consistency metrics (scenario outcome agreement, per-route driving-score correlation, and infraction-type Jaccard similarity), complemented by open-loop error and latency.

Contributions. First, we formulate AV-Stack-to-Sim-Agent Conversion and characterize the observation mismatch between sensor-based teachers and privileged-state students as Inverted Learner–Expert Asymmetry (ILEA). Second, we propose E2M-Distill, combining privileged-state embeddings, PCM, and PPS-guided interface alignment with a retained teacher planning head. Third, we evaluate conversion on SparseDrive, VAD, and SparseDriveV2 using a teacher-relative closed-loop evaluation protocol, alongside efficiency measurements and controlled component comparisons.

2 Related work

Appendix A gives an expanded version of this section.

End-to-end autonomous driving. Sensor-based E2E driving models unify perception, prediction, and planning, such as UniAD (Hu et al., 2023), VAD/VADv2 (Jiang et al., 2023; 2026), SparseDrive (Sun et al., 2025). E2M-Distill converts pretrained sensor-based driving stacks into motion-only sim agents that target their perception-conditioned behavior.

Closed-loop traffic simulation. Motion-only simulators, which act directly on state or trajectories, support scalable rollouts without rendering or full-stack inference (Igl et al., 2022; Gulino et al., 2023; Feng et al., 2023; Zhou et al., 2024; Wu et al., 2024), while recent methods leverage cheap closed-loop rollouts either to mitigate covariate shift in learned traffic agents (Zhang et al., 2025; Chen et al., 2025) or to enable rendering-free closed-loop training of E2E driving policies (Gao et al., 2026). E2M-Distill focuses on a simulator-native agent for a designated AV stack and evaluates its closed-loop consistency with that stack.

Knowledge distillation and learner-expert asymmetry. Knowledge distillation (Hinton et al., 2015), learning using privileged information (Vapnik & Vashist, 2009), generalized distillation (Lopez-Paz et al., 2016), and driving distillation (Chen et al., 2020; Zhang et al., 2021; Chen et al., 2023; Zhou et al., 2023; Feng et al., 2024; Yu et al., 2025b) usually transfer richer supervision to a smaller or sensor-limited policy. LEAD (Nguyen et al., 2026) names this regime the *Learner-Expert Asymmetry* (LEA): the privileged expert holds visibility, uncertainty, or intent advantages over a sensor-based student, and these asymmetries measurably limit imitation-learning performance. The sensor-based instantiation of AV-Stack-to-Sim-Agent Conversion exhibits the dual regime, which we label *Inverted LEA* (ILEA): the teacher is the sensor-based E2E model while the student observes privileged simulator state, so the privileged side is reversed relative to LEA. In our conversion setting, the student receives privileged simulator state, and PCM aligns its features with the teacher's sensor-conditioned representation.

Representation analysis and layer-role identification. Linear probing (Alain & Bengio, 2016), CKA representation similarity (Kornblith et al., 2019), and activation-spectral diagnostics (Roy & Vetterli, 2007; Gao et al., 2017) are standard tools for inspecting what a network has consolidated at which layer. These signals have driven intermediate-layer selection for knowledge distillation through random or attention-based mappings (Haidar et al., 2022; Ji et al., 2021), with recent analyses revisiting how much such selection matters (Yu et al., 2025c). PPS combines semantic probes and representation change to select the teacher activation used for interface alignment.

3 E2M-Distill

E2M-Distill addresses the ILEA setting (Section 1) to perform AV-Stack-to-Sim-Agent Conversion. The method has three components: a privileged simulator-state student that preserves the teacher's planning interface, the Perception Contraction Module (PCM) for contracting privileged simulator-state features toward teacher perception-conditioned features, and the Perception–Planning Score (PPS) for selecting the teacher-side activation that defines the alignment target.

3.1 PROBLEM FORMULATION

Let a target AV stack π_T produce planning output a_t from its sensor observation o_t,

$$a_t \sim \pi_T(\cdot \mid o_t), \tag{1}$$

where, for the sensor-based E2E teachers studied here, o_t is a multi-view image observation. The simulator exposes only privileged simulator state

$$s_t = \Big(\{ (x_t^{(i)},\ v_t^{(i)},\ \ell_t^{(i)}) \}_{i=1}^{N_t},\ m_t \Big), \tag{2}$$

where $x_t^{(i)}, v_t^{(i)}, \ell_t^{(i)}$ denote agent pose, velocity, and lane-context features, and m_t denotes map context queryable without rendering.

Definition 3.1 (AV-Stack-to-Sim-Agent Conversion)**.** Given a target AV stack π_T and privileged simulator state s_t, learn a sim agent $\pi_S(\cdot \mid s_t)$ whose closed-loop rollouts preserve the motion-level behavioral characteristics of π_T under a fixed simulator environment.

The conversion task creates an ILEA perception gap. The student observes privileged simulator state, but the teacher acts through noisy sensor observations and latent features $f_T = \phi_T(o_t)$ shaped by imperfect perception. The simulator state provides explicit agent geometry, velocity, and map context. Sensor observations additionally contain appearance, lighting, and visibility cues that shape the teacher's perception. The two inputs therefore expose different aspects of the scene. PCM learns a state-conditioned approximation to the teacher's perception features for the retained planning head. Appendix B.1 formalizes one aspect of this asymmetry: under a latent-scene model, $I(s_t; f_T) \le I(o_t; f_T)$. This motivates treating privileged simulator state as an imperfect substitute for the teacher's observation-conditioned interface.

Feature approximation and planning error. Let S collect the student's available state, history, and navigation inputs, and let F be a square-integrable feature produced by a fixed teacher. Writing $m(S) = \mathbb{E}[F \mid S]$, any square-integrable state-based predictor g satisfies

$$\mathbb{E}\|g(S) - F\|_2^2 = \mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(F \mid S) + \mathbb{E}\|g(S) - m(S)\|_2^2. \tag{3}$$

The first term measures teacher-feature variability left unresolved by the student's inputs; the second measures approximation to the conditional mean. Thus, precise simulator geometry can coexist with a positive feature-matching error floor. For example, suppose a given state input admits two teacher features f_1, f_2 with probabilities p and $1-p$, due to observation factors absent from that input. The optimal squared-error prediction is $pf_1 + (1-p)f_2$, with residual error $p(1-p)\|f_1 - f_2\|_2^2$. The conversion target therefore depends on how the teacher represents the scene, in addition to the accuracy of the simulator state. The retained planning head connects representation alignment to planning outputs. Let H_T, H_S denote its complete teacher and student inputs, with H_S taken after IA and including anchors, geometry, ego features, and temporal context. If the shared head π_{head} is L-Lipschitz on the comparison domain, then

$$\mathbb{E}\|a_S - a_T\|_2^2 \le L^2 \mathbb{E}\|H_S - H_T\|_2^2. \tag{4}$$

Here the output comparison uses continuous planning outputs under a fixed mode-selection branch. The bound concerns the complete head input; the interface loss supervises an aggregated post-PPD, pre-IA feature. Propositions B.3 and B.5 give the statements and proofs.

Mode agreement. For a multimodal head, feature alignment also affects which candidate trajectory is selected. Let $q(H)$ be its mode logits for a fixed navigation command, and let $k^\star$ be the teacher's highest-scoring mode, with score margin

$$\Delta_T = q_{k^\star}(H_T) - \max_{j \neq k^\star} q_j(H_T) > 0. \tag{5}$$

If q is L_q-Lipschitz from the input Euclidean norm to the logit maximum norm, then

$$2L_q\|H_S - H_T\|_2 < \Delta_T \quad \Longrightarrow \quad \arg\max_k q_k(H_S) = k^\star. \tag{6}$$

Within this neighborhood, a Lipschitz bound for the selected candidate also bounds the difference between the selected planning trajectories (Corollary B.6). The score margin determines how much representation error preserves the teacher's mode choice.

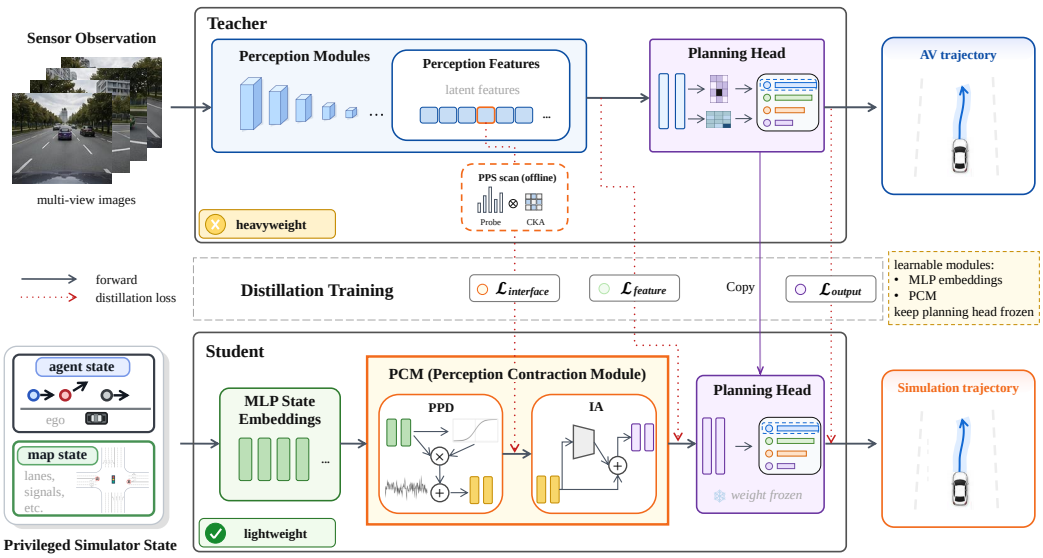


Figure 2: E2M-Distill pipeline. The frozen teacher supplies intermediate token features, planning outputs, and a PPS-selected interface activation during conversion. The deployed student replaces sensor perception with privileged simulator-state embeddings, applies PCM before the copied frozen planning head, and trains the embedding–PCM pathway with feature, output, and interface-alignment losses plus a lightweight gate-range regularizer.

3.2 Framework Overview

E2M-Distill uses the same conversion recipe across teachers: privileged simulator state $s_t \rightarrow$ lightweight agent/map/ego MLP embeddings $\rightarrow$ PCM $\rightarrow$ retained teacher planning head $\rightarrow$ planning output a_t. As shown in Figure 2, the frozen teacher processes its sensor observation o_t and supplies the supervision signals consumed by the distillation losses below. The deployed student instead embeds s_t, applies PCM, and runs the frozen planning head. Model-specific adapters implement token budgets, anchor formats, temporal queues, and output terms within this pipeline.

Training uses paired image–state samples. Feature, output, and interface-alignment losses keep student tokens compatible with the teacher interface, preserve teacher motion behavior, and align the post-PPD feature f_S^{pre} with f_T^{itf}. Implementation details are deferred to Appendices B.2 and B.4.

This construction removes the two dominant costs of full E2E simulation, multi-view rendering and vision-backbone inference, and replaces them with MLP embeddings ($\sim$0.5M parameters) plus the retained planning head. The latency benefit is measured directly in Table 1.

3.3 Perception Contraction Module

PCM maps privileged-state embeddings to features for the retained planning head. It combines (i) a learnable *Privileged Perception Degradation* (PPD), (ii) a lightweight *Interface Adapter* (IA), and (iii) an interface-alignment loss. PPD is followed by IA, and the alignment loss supervises the post-PPD feature using the PPS-selected teacher activation.

Privileged Perception Degradation. For an embedded feature h, PPD applies an input-dependent gate and learned noise,

$$\text{PPD}(h) = h \odot g(h)/c(\bar{g}) + n(h) \cdot s, \quad g(h) = \sigma(\text{MLP}_{\text{gate}}(h)), \tag{7}$$

where $n(h) = \text{Softplus}(\text{MLP}_{\text{noise}}(h)) \cdot \alpha_{\text{eff}} \cdot \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, $\alpha_{\text{eff}} = \max(\alpha, \alpha_{\min})$ enforces a nonzero lower bound on the learnable noise-scale parameter α, $\bar{g}$ is the EMA gate value, $c(u) = \max(u, 0.1)$ is a detached inverted-gating normalizer, and $s \in [0, 1]$ is a curriculum strength. PPD is applied to instance features, anchor embeddings, and raw geometric fields. It further uses object-

level dropout and classification recomputation to close object- and confidence-level shortcuts. The five-level mechanism, gate-range regularizer, and processing order are in Appendix B.3.

Interface Adapter. After PPD, IA maps degraded features back to the distribution expected by the retained head using a zero-initialized residual adapter followed by LayerNorm:

$$f_{\text{proj}} = \mathrm{LN}\big(f + \mathrm{MLP}_{\text{proj}}(f)\big). \tag{8}$$

Interface-alignment loss. Let f_S^{pre} be the post-PPD, pre-IA student feature and f_T^{itf} the teacher activation at the PPS-selected interface. PCM aligns the student to this interface with

$$\mathcal{L}_{\text{interface}} = w_{\text{mse}} \| f_S^{\text{pre}} - P(f_T^{\text{itf}}) \|_2^2 + w_{\cos}\big(1 - \cos(f_S^{\text{pre}}, P(f_T^{\text{itf}}))\big), \tag{9}$$

where P is a trainable linear interface projector that absorbs teacher–student dimension mismatch. We use w_{mse}=1.0 and $w_{\cos}$=0.5. The interface loss supervises the pre-IA feature, while IA is trained through the downstream distillation losses. At inference, the student retains PPD gating and IA, with noise and object dropout disabled.

3.4 Perception–planning interface localization

Interface alignment needs a teacher target whose semantics match the post-PPD, pre-IA student feature: an exposed activation inside the teacher's perception-semantic stack, after perception-specific processing has consolidated but before planning-specific transformations dominate. The alignment target is selected from exposed teacher activations according to their semantic and geometric properties. We therefore localize the closest exposed teacher activation with an offline *Perception–Planning Score* (PPS) from calibration samples without retraining the teacher. For each ordered candidate activation H_l, linear probes produce held-out scores for a perception proxy (number of valid surrounding agents) and a planning proxy (2D ego endpoint). These scores are converted to percentile ranks $r_l^{\text{perc}}, r_l^{\text{plan}} \in [0,1]$. For each edge $b : H_{b-1} \to H_b$ between adjacent candidates, let $A_b \in [0,1]$ be the linear-CKA dissimilarity between adjacent centered activations. PPS is

$$\mathrm{PPS}_b = \big(r_{b-1}^{\text{perc}} - r_{b-1}^{\text{plan}}\big) A_b, \qquad b^\star = \arg\max_b \mathrm{PPS}_b, \qquad f_T^{\text{itf}} = H_{b^\star}. \tag{10}$$

The signed product is the design choice: a high score requires both an upstream layer that is perception-dominant and an edge across which the representation undergoes large geometric reorganization. PPS evaluates the asymmetry on the upstream layer because the score asks whether perception-specific semantics have consolidated *by* edge b. Given the selected edge $b^\star$, the interface target hooked by Equation (9) is the downstream activation $H_{b^\star}$, i.e., the exposed post-transition activation inside the perception-semantic stack. The product combines upstream probe-rank asymmetry with adjacent-activation dissimilarity. PPS uses a common scoring rule for different models, with model-specific candidate activations and no tunable fusion weights or score thresholds. Appendix C gives the detailed procedure, including the CKA computation, the side convention shared across all interface localizers reported in Table 3, and the alternative localizers compared there.

3.5 Distillation training strategy

We train E2M-Distill from paired image–state samples. At each time step, the frozen teacher exposes three supervision signals from o_t: token-level intermediate features for $\mathcal{L}_{\text{feat}}$, the planning target a_t^T for $\mathcal{L}_{\text{out}}$, and the PPS-selected interface activation f_T^{itf} for $\mathcal{L}_{\text{interface}}$, while the student consumes only privileged simulator state s_t. The student maps s_t into teacher-compatible tokens, applies PCM, and feeds the result to the frozen planning head.

The training objective combines feature, output, and interface alignment. Feature distillation aligns student tokens with teacher tokens, output distillation matches the teacher's motion predictions, and interface alignment matches the post-PPD, pre-IA feature to the projected PPS-selected teacher activation. The main training objective is

$$\mathcal{L} = \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{out}}\mathcal{L}_{\text{out}} + \lambda_{\text{itf}}\mathcal{L}_{\text{interface}}. \tag{11}$$

At training time we additionally add a lightweight PPD gate-range regularizer $\mathcal{L}_{\text{gate}}$. The gate-range regularizer penalizes mean gate activations outside a prescribed interval. Its precise form is given in Appendix B.3. The planning head remains frozen throughout training, and only the privileged simulator-state MLP embeddings and PCM-related parameters are optimized. Details on token matching, output terms, optimization, PPD curriculum, and loss weights are in Appendix B.4.

4 EXPERIMENTS

We evaluate E2M-Distill on two architecturally distinct primary E2E teachers: **(a) transfer fidelity and efficiency** against Direct-GT and distillation baselines (Section 4.2), **(b) perception–planning interface localization** against single-signal interface-localization baselines (Section 4.3), and **(c) ablations** of PPD, IA, and the distillation losses (Section 4.4).

4.1 SETUP AND EVALUATION PROTOCOL

Models and protocol. We use the SparseDrive (Sun et al., 2025) checkpoint (sparse scene representation, multi-mode motion-planning head) released by RIFT (Chen et al., 2025) for Bench2Drive evaluation, and the VAD (Jiang et al., 2023) checkpoint (vectorized scene representation, transformer decoder) from the official Bench2DriveZoo release (Jia et al., 2024). All E2M-Distill results use the trained PCM pathway described in Section 3.3. Closed-loop evaluation uses the official Bench2Drive protocol with 220 routes in CARLA. Training and baseline details are in Appendix B, evaluation details in Appendix D, and latency-protocol details in Appendix F.

Teacher–student consistency. We evaluate teacher–student consistency with paired measurements on the same scenes and routes. Closed-loop consistency measures route-level behavioral agreement, and teacher-relative open-loop error measures instantaneous planning agreement. Full metric definitions are provided in Appendix D.

Closed-loop consistency. Teacher and student are rolled out over the same 220 Bench2Drive routes under the same CARLA seed, weather, and traffic configuration. Route indices are aligned across the teacher, E2M-Distill, and all baselines before aggregation, and the paired per-route results are compared along three complementary facets. **Scenario Outcome Agreement** (SOA), $\mathrm{SOA} = \frac{1}{n}\sum_i \mathbb{1}[s_i^T = s_i^S] \times 100\%$, is the percentage of routes on which teacher and student agree on the binary success indicator, capturing *scenario-level* fidelity. **Driving-Score Correlation** ($\mathrm{DS}_{\mathrm{corr}}$), $\mathrm{DS}_{\mathrm{corr}} = \rho(\mathrm{DS}^T_{1:n}, \mathrm{DS}^S_{1:n})$, is the Pearson correlation between paired per-route driving scores, capturing route-score association. A uniformly stronger or weaker student can still obtain high $\mathrm{DS}_{\mathrm{corr}}$ when the route scores remain linearly associated. **Infraction-Type Jaccard** ($\mathrm{Inf}_{\mathrm{Jaccard}}$) is the mean route-wise Jaccard similarity of triggered infraction types, with the empty-empty case assigned similarity 1. It measures overlap in triggered infraction types. Together they cover distinct failure modes.

Teacher-relative open-loop error. For each frame in the Bench2Drive validation split, teacher and student plan from the same logged scene state. We report the per-frame Euclidean distance OL_δ between their planned ego waypoints at horizons $\delta \in \{0.5, 1.5, 3.0\}$ s, averaged over frames. Lower is better, and the full definition is in Appendix D.3. The reference trajectory is the teacher's planned trajectory. It captures instantaneous planning agreement and diagnoses a possible *snapshot-level shortcut*: under ILEA, a privileged simulator-state student may reduce OL_δ on logged snapshots by exploiting state cues the teacher cannot reliably perceive, without preserving teacher behavioral fidelity in closed-loop rollouts.

Latency. We report per-step inference latency (ms/step) at $V \in \{1, 10, 100\}$ simultaneously simulated agents to characterize cost scaling.

4.2 TRANSFER FIDELITY AND EFFICIENCY

Protocol. All students in Table 1 share the same MLP-embedding architecture, retained planning head, and training schedule. Differences therefore reflect training objectives and degradation choices.

Closed-loop consistency. E2M-Distill achieves the highest SOA, $\mathrm{DS}_{\mathrm{corr}}$, and $\mathrm{Inf}_{\mathrm{Jaccard}}$ among the compared methods for both teachers. Relative to the strongest baseline for SOA, the gains are 0.5 percentage points for SparseDrive and 2.8 points for VAD. These results demonstrate improved teacher-relative consistency across the two primary conversion settings.

Table 1: Main transfer-fidelity and efficiency results. Teacher-relative open-loop errors are lower-is-better; closed-loop consistency metrics follow Section 4.1; latency is reported in ms/step. Official Bench2Drive DS/RC/IP/SR metrics are reported separately in Table 6.

		Open-Loop Error (vs Teacher)			*Closed-Loop Consistency*			*Efficiency*		
Method	Input	0.5s↓	1.5s↓	3.0s↓	SOA↑	$\mathrm{DS}_{\mathrm{corr}}$ ↑	$\mathrm{Inf}_{\mathrm{Jaccard}}$ ↑	1-veh (ms/step)	10-veh (ms/step)	100-veh (ms/step)
SparseDrive teacher										
Teacher	Images	—	—	—	—	—	—	268	2,736	≈32,800†
Direct-GT	State	1.6953	5.1948	10.3674	50.50	−0.0200	0.3337			
Output-only KD	State	1.3330	3.5923	6.5337	64.10	0.5398	0.5669			
FitNets KD	State	**0.8697**	**2.6717**	**5.3924**	64.50	0.3632	0.5093			
Hand-crafted noise + KD	State	1.2968	3.7021	7.0577	66.80	0.5680	0.5462			
E2M-Distill (Ours)	State	1.0489	2.9715	5.7000	**67.30**	**0.5863**	**0.5768**	18	141	1,468
VAD teacher										
Teacher	Images	—	—	—	—	—	—	444	4,480	≈50,900†
Direct-GT	State	**11.8591**	**21.1765**	45.8033	63.60	0.5228	0.4986			
Output-only KD	State	13.2457	26.2042	**28.8249**	60.00	0.5083	0.4854			
FitNets KD	State	12.5516	34.5990	53.1776	60.00	0.4759	0.4868			
Hand-crafted noise + KD	State	12.0629	21.6032	53.8428	58.20	0.4855	0.4851			
E2M-Distill (Ours)	State	17.9205	38.6938	35.4211	**66.40**	**0.5251**	**0.5097**	17	137	1,373

Notes. **State** denotes privileged simulator-state vectors. Student latency is reported for E2M-Distill. Matched baseline forward timings are in Appendix F.2. †The teacher completes only $V{=}20 < 100$ in our setting; the 100-vehicle cost is extrapolated from $V{=}1$ to 20.

Short-horizon open-loop trade-off. The rankings under open-loop error and closed-loop consistency differ. FitNets KD gives the best SparseDrive open-loop errors but much lower $\mathrm{DS}_{\mathrm{corr}}$ (0.36 vs 0.59) than E2M-Distill. The reversal is sharper on VAD: E2M-Distill has the worst 0.5 s open-loop error among all five methods (17.92 vs 11.86 for the best baseline) yet the best closed-loop consistency on every facet.

Efficiency. Replacing rendering and vision inference with privileged simulator-state embeddings yields $14.9\times$ speedup over SparseDrive and $26.1\times$ over VAD at $V{=}1$. At $V{=}100$, the students require 1,468 ms (SparseDrive) and 1,373 ms (VAD), compared with extrapolated teacher costs of $\approx$32,800 ms and $\approx$50,900 ms. The full scaling plot is in Figure 3.

Additional target stacks and repeated evaluation. To assess conversion of a higher-performing teacher, we include SparseDriveV2 (DS 81.42). To examine a different pretraining source, we use a nuScenes-trained SparseDrive checkpoint, converted and evaluated in CARLA. We compare learned PPD with a fixed-noise control retaining IA and interface alignment, and assess repeatability over three simulator seeds with fixed checkpoints. E2M-Distill achieves higher closed-loop consistency than the corresponding controls and higher mean consistency across repeated evaluations. Table 4 reports the additional target-stack and controlled PPD results; detailed settings and repeated-evaluation results are in Appendices E.1 and E.2.

4.3 Perception–planning interface localization

We compare PPS-selected interface targets with CKA-change, downstream-semantic, and spectral-transition targets under the same student architecture and training schedule. For SparseDrive, PPS selects `img_neck.fpn_convs.0`, achieving 67.30% SOA versus 65.50% for CKA-change. For VAD, PPS and CKA-change both select `pts_bbox_head.transformer` and reach 66.40% SOA; the downstream-semantic and spectral-transition targets yield 56.80% and 59.10%, respectively. The selected PPS scores are 0.8017 for SparseDrive and 0.9255 for VAD, with margins of 0.0974 and 0.3084 over the respective runners-up. Candidate activations, scoring procedures, and detailed results are provided in Appendix C.

4.4 Further ablations

We next decompose the contribution of PPD, IA, and the distillation loss. Table 2 reports module-removal and loss-ablation results for both teachers.

For module removal, the full model is the only configuration that achieves the best $\mathrm{DS}_{\mathrm{corr}}$ and $\mathrm{Inf}_{\mathrm{Jaccard}}$ for both teachers. Removing IA causes the largest SparseDrive SOA drop (67.30→61.40),

Table 2: Further ablations on SparseDrive and VAD. Bold marks the best value within each group and teacher, with ties bolded.

	SparseDrive teacher			*VAD teacher*		
Configuration	SOA↑	DS_{corr} ↑	$Inf_{Jaccard}$ ↑	SOA↑	DS_{corr} ↑	$Inf_{Jaccard}$ ↑
Group I: Module removal						
E2M-Distill	**67.30**	**0.5863**	**0.5768**	**66.40**	**0.5251**	**0.5097**
w/o IA	61.40	0.5675	0.5633	57.70	0.5124	0.4758
w/o PPD	**67.30**	0.5626	0.5731	57.70	0.5152	0.4733
w/o PCM	64.50	0.5465	0.5543	58.20	0.5129	0.4775
Group II: Loss ablation						
E2M-Distill	**67.30**	**0.5863**	**0.5768**	**66.40**	**0.5251**	**0.5097**
w/o Feature loss	64.10	0.5581	0.5551	64.50	0.5153	0.5002
w/o Output loss	52.70	0.2010	0.4241	61.80	0.5092	0.4919
w/o Interface loss	64.50	0.5398	0.5641	58.20	0.5127	0.4957

while every module removal drops VAD SOA to 57.70–58.20. These configurations show a larger decrease in outcome agreement for VAD.

For loss removal, $\mathcal{L}_{out}$ is most important for SparseDrive (SOA 52.70 without it), whereas interface alignment matters most for VAD (SOA 58.20 without it). Removing feature loss reduces SOA by 3.2 percentage points for SparseDrive and 1.9 points for VAD.

SparseDrive's SOA remains 67.30 when PPD is removed, while DS_{corr} drops from 0.5863 to 0.5626. Agreement on binary route outcomes can therefore coexist with lower route-score association. The controlled comparison in Table 4 retains IA, interface alignment, and fixed input noise. With learned PPD, SparseDrive SOA increases from 64.1 to 67.3 and DS_{corr} from 0.5655 to 0.5863, complementing the module-removal results.

5 Discussion and conclusion

E2M-Distill makes teacher-relative behavior preservation the objective of AV-stack conversion. The framework aligns privileged-state features with the teacher's perception-conditioned interface and retains its planning head to produce motion-only simulation agents. Experiments across SparseDrive, VAD, and SparseDriveV2 show improved closed-loop consistency over the compared conversion baselines. The SparseDrive and VAD conversions also reduce per-step simulation latency.

Scope and limitations. The ILEA proposition characterizes information asymmetry under a latent-scene model; closed-loop behavior preservation is assessed empirically. Evaluation uses Bench2Drive/CARLA with the target stacks described in Section 4.1 and Appendix E.1. PCM learns an average perception contraction and may miss rare-class errors, hallucinated objects, weather-dependent correlations, or other long-tail perception effects. PPS assumes an identifiable perception–planning interface and may be less suitable for heavily interleaved or multi-transition architectures.

Future directions. Future work will develop interaction-focused datasets and benchmarks for assessing teacher–student behavior consistency in city-scale scenarios, using time-to-collision (TTC) distributions and emergent traffic statistics. We will study jointly learned, multi-interface or scenario-conditioned PCM/PPS together with student-side planning-head compression to improve conversion fidelity. We also plan system-level studies using rendering-free surrogates, including rare-event safety validation, mixed-fleet traffic-impact analysis, and city-scale traffic policies over populations of converted sim agents.

AI USE STATEMENT

Generative AI tools assisted with LaTeX formatting, editing and organizing author-reported experimental results, and reviewing the manuscript's mathematical arguments and experimental methodology. The authors manually reviewed the AI-assisted content and take responsibility for the final manuscript.

REFERENCES

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

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2020.

Keyu Chen, Wenchao Sun, Hao Cheng, and Sifa Zheng. RIFT: Group-relative RL fine-tuning for realistic and controllable traffic simulation. *arXiv preprint arXiv:2505.03344*, 2025. doi: 10.48550/arXiv.2505.03344.

Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. BEVDistill: Cross-modal BEV distillation for multi-view 3D object detection. In *International Conference on Learning Representations (ICLR)*, 2023.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245.

Kaituo Feng, Changsheng Li, Dongchun Ren, Ye Yuan, and Guoren Wang. On the road to portability: Compressing end-to-end motion planner for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15099–15108, 2024. doi: 10.1109/CVPR52733.2024.01430.

Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. TrafficGen: Learning to generate diverse and realistic traffic scenarios. In *International Conference on Robotics and Automation (ICRA)*, pp. 3567–3575, 2023.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, 2017. doi: 10.1101/214262.

Yinfeng Gao, Qichao Zhang, Deqing Liu, Zhongpu Xia, Guang Li, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, Da-Wei Ding, and Dongbin Zhao. PerlAD: Towards enhanced closed-loop end-to-end autonomous driving with pseudo-simulation-based reinforcement learning. *IEEE Robotics and Automation Letters*, 2026. doi: 10.1109/LRA.2026.3675928.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John D. Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2310.08710.

Md Akmal Haidar, Nithin Anchuri, Mehdi Rezagholizadeh, Abbas Ghaddar, Philippe Langlais, and Pascal Poupart. RAIL-KD: Random intermediate layer mapping for knowledge distillation. In *Findings of the Association for Computational Linguistics: NAACL*, pp. 1389–1400, 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Xuemin Hu, Shen Li, Tingyu Huang, Bo Tang, Rouxing Huai, and Long Chen. How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence. *IEEE Transactions on Intelligent Vehicles*, 9(1):593–612, 2024. doi: 10.1109/TIV.2023.3312777.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023.

Zhiyu Huang, Haochen Liu, and Chen Lv. GameFormer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3903–3913, 2023.

Aditya Humnabadkar, Arindam Sikdar, Benjamin Cave, Huaizhong Zhang, Nik Bessis, and Ardhendu Behera. From virtual environments to real-world trials: Emerging trends in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–21, 2026. doi: 10.1109/TITS.2026.3679248.

Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2445–2451, 2022. doi: 10.1109/ICRA46639.2022.9811990.

Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/017761f94a1cd66d01c041aff85492c4-Paper-Datasets_and_Benchmarks_Track.pdf.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023. doi: 10.1109/ICCV51070.2023.00766.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=0a4dA6eUHN.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Charles H. Martin, Tongsu Peng, and Michael W. Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, 2021.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. LEAD: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2512.20563.

Chihiro Noguchi and Takaki Yamamoto. Pseudo-expert regularized offline RL for end-to-end autonomous driving in photorealistic closed-loop environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026. URL https://arxiv.org/abs/2512.18662. arXiv preprint arXiv:2512.18662.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, pp. 606–610, 2007.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801, 2025. doi: 10.1109/icra55743.2025.11128800.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026. URL https://arxiv.org/abs/2603.29163.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009. doi: 10.1016/j.neunet.2009.06.042.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xuemeng Yang, Licheng Wen, Yukai Ma, Jianbiao Mei, Xin Li, Tiantian Wei, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Botian Shi, Liang He, Yong Liu, and Yu Qiao. DriveArena: A closed-loop generative simulation platform for autonomous driving. *arXiv preprint arXiv:2408.00415*, 2024. URL https://arxiv.org/abs/2408.00415.

Haibao Yu, Wenxian Yang, Ruiyang Hao, Chuanye Wang, Jiaru Zhong, Ping Luo, and Zaiqing Nie. DriveE2E: Closed-loop benchmark for end-to-end autonomous driving through real-to-simulation. *arXiv preprint arXiv:2509.23922*, 2025a. URL https://arxiv.org/abs/2509.23922.

Rui Yu, Xianghang Zhang, Runkai Zhao, Huaicheng Yan, and Meng Wang. DistillDrive: End-to-end multi-mode autonomous driving distillation by isomorphic hetero-source planning model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26188–26197, 2025b.

Zony Yu, Yuqiao Wen, and Lili Mou. Revisiting intermediate-layer matching in knowledge distillation: Layer-selection strategy doesn't matter (much). In *Findings of the Association for Computational Linguistics: IJCNLP-AACL*, pp. 1686–1694, 2025c. doi: 10.18653/v1/2025.findings-ijcnlp.105.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15222–15232, 2021.

Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5422–5432, 2025.

Shengchao Zhou, Weizhou Liu, Chen Hu, Shuchang Zhou, and Chao Ma. UniDistill: A universal cross-modality knowledge distillation framework for 3D object detection in bird's-eye view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5116–5125, 2023.

Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. BehaviorGPT: Smart agent simulation for autonomous driving with next-patch prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-2527.

A Related Work

This appendix expands Section 2 and situates E2M-Distill among adjacent work.

A.1 End-to-end autonomous driving

End-to-end (E2E) systems unify perception, prediction, and planning into a single differentiable pipeline. UniAD (Hu et al., 2023) cascades perception, prediction, and planning within a shared transformer backbone. VAD (Jiang et al., 2023) vectorizes scene elements and reasons about planning in a token space, and VADv2 (Jiang et al., 2026) extends this line with probabilistic action planning for closed-loop driving. SparseDrive (Sun et al., 2025) maintains an explicit motion-planning head atop a sparse scene representation, while Transfuser (Chitta et al., 2023) fuses camera and LiDAR features through transformer-based attention for waypoint prediction. GameFormer (Huang et al., 2023) uses vectorized scene inputs and assumes upstream perception. These exposed representations provide interfaces for stack-to-agent conversion.

A.2 Closed-loop traffic simulation

Learned traffic-agent models improve behavioral diversity and realism in closed-loop rollouts. Symphony (Igl et al., 2022) trains agents with imitation and reinforcement learning. BehaviorGPT (Zhou et al., 2024) casts agent simulation as next-patch prediction, Waymax (Gulino et al., 2023) emphasizes accelerator-friendly large-scale rollout, TrafficGen (Feng et al., 2023) generates controllable traffic scenarios, and SMART (Wu et al., 2024) uses discrete tokenization to model multi-agent trajectories autoregressively. Recent closed-loop fine-tuning methods such as CAT-K (Zhang et al., 2025) and RIFT (Chen et al., 2025) mitigate covariate shift for learned traffic policies, while PerlAD (Gao et al., 2026) uses vector-space pseudo-simulation for rendering-free closed-loop E2E training. Our evaluation centers on transferring a designated teacher's behavior and measuring closed-loop consistency.

A.3 Distillation and learner-expert asymmetry

Knowledge distillation (Hinton et al., 2015) transfers learned representations from a large teacher to a smaller student. Learning using privileged information (Vapnik & Vashist, 2009) and generalized distillation (Lopez-Paz et al., 2016) similarly use information available only during training to guide a deployment-time learner. In autonomous driving, privileged, cross-modal, or planner-compression distillation typically trains a sensor-input or smaller policy to match an oracle, richer, or heavier teacher that observes privileged state, BEV annotations, structured scene annotations, or carries a larger planner (Chen et al., 2020; Zhang et al., 2021; Chen et al., 2023; Zhou et al., 2023; Feng et al., 2024; Yu et al., 2025b). LEAD (Nguyen et al., 2026) names this regime the *Learner-Expert Asymmetry* (LEA): the privileged expert holds visibility, uncertainty, or intent advantages over the sensor-based student, and these asymmetries measurably limit imitation-learning performance. E2M-Distill operates under the dual regime, which we label *Inverted LEA* (ILEA): the teacher is the sensor-based E2E model, and the student uses privileged simulator-state input. PCM is trained with teacher features and outputs to account for the different observation structures.

A.4 Representation analysis and layer-role identification

Our PPS-based interface-localization procedure builds on representation-analysis tools. CKA (Kornblith et al., 2019) measures representational similarity between layers, linear probing (Alain & Bengio, 2016) diagnoses what information is decodable at each intermediate layer, and spectral analysis (Martin et al., 2021) characterizes representation geometry via eigenvalue statistics. Automatic layer selection for intermediate-layer distillation has been studied under random mappings, as in RAIL-KD (Haidar et al., 2022), and attention-based matching (Ji et al., 2021). Recent analysis (Yu et al., 2025c) further suggests that many generic layer-selection strategies perform comparably. PPS combines probe-rank asymmetry and CKA dissimilarity through a signed product to select the teacher activation used for interface alignment. The localized interface is then validated by downstream closed-loop transfer, linking representation evidence to preserved behavior.

B Architecture and Training

B.1 Inverted Learner-Expert Asymmetry (ILEA): formalization

This section relates the observation mismatch in Section 3.1 to feature approximation and planning error. Entropy and mutual information refer to the finite-precision variables actually used by the simulator and network. For this analysis, o_t includes the sensor history and any teacher-side context used to compute f_T.

Assumption B.1 (Latent-scene model)**.** There exists a scene latent z such that the privileged simulator state is a deterministic function of the scene, $s_t = \psi(z)$. The teacher's sensor observation is generated by a stochastic perception channel $o_t \sim p(o_t \mid z)$, equivalently $o_t = \rho(z, \xi_t)$ for an unmodeled observation nuisance ξ_t, with $H(o_t \mid z) > 0$. This conditional randomness represents sensor noise, randomized rendering, and weather, lighting, or occlusion factors not encoded deterministically by z. The frozen teacher in evaluation mode produces intermediate features through a fixed measurable map $f_T = \phi_T(o_t)$.

Proposition B.2 (ILEA: perception gap)**.** *Under Assumption B.1, $H(s_t \mid z) = 0$ and*

$$I(o_t; f_T) - I(s_t; f_T) = H(f_T \mid s_t) \geq 0. \tag{12}$$

The inequality is strict exactly when no measurable function of s_t equals f_T almost surely. A sufficient condition is that no measurable function of z equals f_T almost surely.

Proof. Since $s_t = \psi(z)$, $H(s_t \mid z) = 0$. Since $f_T = \phi_T(o_t)$ is deterministic, $I(o_t; f_T) = H(f_T)$. Subtracting $I(s_t; f_T) = H(f_T) - H(f_T \mid s_t)$ proves Equation (12). For finite-precision variables, $H(f_T \mid s_t) = 0$ exactly when f_T is determined by s_t almost surely. Moreover, data processing along $s_t \leftarrow z \rightarrow f_T$ yields $I(s_t; f_T) \leq I(z; f_T)$, or $H(f_T \mid s_t) \geq H(f_T \mid z)$. If no measurable function of z equals f_T almost surely, the latter conditional entropy is positive, giving the stated sufficient condition. □

Approximation from the available student inputs. For the following results, S includes all state, history, and navigation information available to the student. The target F is produced by a fixed vector-valued teacher feature map. This formulation also applies to a fixed projection of teacher features. When evaluating a learned projector, its parameters are held fixed in the definition of F.

Proposition B.3 (Conditional feature approximation)**.** *Suppose $\mathbb{E}\|F\|_2^2 < \infty$, and let $m(S) = \mathbb{E}[F \mid S]$. For every square-integrable measurable predictor $g(S)$,*

$$\begin{aligned} \mathbb{E}\|g(S) - F\|_2^2 = \mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(F \mid S) \\ + \mathbb{E}\|g(S) - m(S)\|_2^2. \end{aligned} \tag{13}$$

Over all such predictors, the minimum error is $\mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(F \mid S)$, attained exactly by $g(S) = m(S)$ almost surely.

Proof. Write $g(S) - F = (g(S) - m(S)) + (m(S) - F)$ and expand the squared norm. The cross term has expectation zero because

$$\mathbb{E}[\langle g(S) - m(S), m(S) - F\rangle \mid S] = \langle g(S) - m(S), m(S) - \mathbb{E}[F \mid S]\rangle = 0.$$

The conditional expectation of $\|F - m(S)\|_2^2$ is $\mathrm{tr}\,\mathrm{Cov}(F \mid S)$. Taking expectations gives Equation (13). Its second term is nonnegative and equals zero precisely when $g(S) = m(S)$ almost surely. □

Corollary B.4 (Randomized feature approximation)**.** *Let $G = g(S, U)$ have finite second moment, where U and F are conditionally independent given S. Then*

$$\begin{aligned} \mathbb{E}\|G - F\|_2^2 = \mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(F \mid S) + \mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(G \mid S) \\ + \mathbb{E}\|\mathbb{E}[G \mid S] - m(S)\|_2^2. \end{aligned} \tag{14}$$

Proof. Condition on S and center F and G at their respective conditional means. Conditional independence makes the cross covariance zero. The two centered squared norms yield the two covariance traces, and the remaining term is the squared distance between conditional means. Averaging over S gives the identity. □

These identities separate teacher-feature variability, approximation error, and student randomization. The conditional mean minimizes paired squared error, and independent student noise contributes the conditional-variance term in Equation (14).

Propagation through the retained planning head. Let H_T and H_S concatenate all corresponding inputs read by the shared frozen head, using a common token layout. These inputs include instance and anchor features, geometric fields, ego features, and temporal context; student features are taken after IA.

Proposition B.5 (Planning-output error)**.** *Suppose the shared head π_{head} is L-Lipschitz on a domain containing H_T and H_S, with outputs $a_T = \pi_{\mathrm{head}}(H_T)$ and $a_S = \pi_{\mathrm{head}}(H_S)$. Then*

$$\|a_S - a_T\|_2 \le L\|H_S - H_T\|_2. \tag{15}$$

If $\mathbb{E}\|H_S - H_T\|_2^2 < \infty$, then

$$\mathbb{E}\|a_S - a_T\|_2^2 \le L^2\mathbb{E}\|H_S - H_T\|_2^2. \tag{16}$$

Proof. Apply the Lipschitz property to the pair (H_S, H_T) to obtain Equation (15). Squaring both sides and taking expectations gives Equation (16). □

For multimodal planners, the proposition applies to continuous candidate outputs or to a selected trajectory within a fixed mode-selection branch. The feature error in this bound is measured at the actual head input. The interface loss in Equation (9) instead supervises an aggregated post-PPD, pre-IA representation; the feature and output losses provide the additional supervision described in Appendix B.4.

Corollary B.6 (Preservation of the selected planning mode)**.** *For a fixed navigation command, let the shared head output mode logits $q(H)$ and candidate trajectories $\tau_k(H)$. Suppose*

$$\|q(H) - q(H')\|_\infty \le L_q\|H - H'\|_2$$

on the comparison domain. Let $k^\star = \arg\max_k q_k(H_T)$ be the unique teacher-selected mode, with margin $\Delta_T > 0$ as in Equation (5). If $\delta = \|H_S - H_T\|_2$ satisfies $2L_q\delta < \Delta_T$, the student selects the same mode. If $\tau_{k^\star}$ is L_τ-Lipschitz on that domain, the selected trajectories satisfy

$$\|\tau_{k^\star}(H_S) - \tau_{k^\star}(H_T)\|_2 \le L_\tau\delta. \tag{17}$$

Proof. For every $j \neq k^\star$, the logit bound gives

$$\begin{aligned} q_{k^\star}(H_S) - q_j(H_S) &\ge q_{k^\star}(H_T) - q_j(H_T) - 2L_q\delta \\ &\ge \Delta_T - 2L_q\delta > 0. \end{aligned}$$

Thus $k^\star$ remains the unique maximizing mode at H_S. Applying the Lipschitz property of $\tau_{k^\star}$ to the two head inputs proves Equation (17). □

B.2 Student embedding pathways

The student has three embedding pathways into the teacher feature dimension d=256.

Agent embedding. Each surrounding agent is represented by a 7-dimensional state vector $x_{\text{agent}} = [x, y, w, l, \psi, v_x, v_y]$ (position, bounding-box size, heading, velocity). A two-layer MLP maps each agent state to an *instance feature* $f_i \in \mathbb{R}^d$. The agent state is also lifted to a teacher-specific *anchor* representation and passed through a separate MLP to produce an *anchor embedding* $a_i \in \mathbb{R}^d$. SparseDrive uses an 11-dimensional log-space anchor

$$x^{\text{SD}}_{\text{anchor}} = [X, Y, Z, \log w, \log l, \log h, \sin\psi, \cos\psi, v_x, v_y, v_z], \tag{18}$$

with Z, h, and v_z set to ground-plane defaults. VAD uses a 10-dimensional linear anchor (no log transform and no v_z). Both architectures use the same MLP-based embedding structure, with only the anchor lift differing. Up to N=50 agents per frame are emitted and zero-padded. The student also produces per-agent classification logits $c_i \in \mathbb{R}^C$ (C=9) and maintains a validity mask. During training, agent state vectors receive Gaussian noise with σ_{pos}=0.3 m, σ_{size}=0.1 m, σ_{yaw}=0.05 rad, and σ_{vel}=0.2 m/s.

Map embedding. Lane polylines are represented as $x_{\text{map}} \in \mathbb{R}^{M\times P\times 2}$ with M=100 and P=20. A two-layer MLP maps each flattened segment to $\mathbb{R}^d$ instance features and anchors. During training, polyline coordinates additionally receive Gaussian noise with σ_{poly}=0.2 m, applied jointly with the agent-state noise above.

Ego-state encoding. The ego state $x_{\text{ego}} \in \mathbb{R}^{10}$ (local acceleration, angular velocity, speed, steering, validity flags) is encoded by a three-layer MLP with ReLU and LayerNorm:

$$e_S = \text{MLP}_{\text{ego}}(x_{\text{ego}}) \in \mathbb{R}^d. \tag{19}$$

A learnable ego anchor provides a fixed bounding-box reference.

Temporal instance queue. Models that require temporal context (e.g., SparseDrive) maintain a queue of instance features across up to T=4 consecutive frames, matched by track ID with threshold 0.2. Scene boundaries reset the queue. The queue exposes stacked temporal features of shape (B, N, T, d) to the downstream head.

B.3 Perception Contraction Module — full details

This appendix expands the main-body PCM description (Section 3.3) with the full five-level mechanism, gate-range regularizer, classification recomputation, processing order, and curriculum.

Level 1: Input-dependent gating with inverted normalization. A learned gate suppresses feature dimensions via $g(h) = \sigma(\text{MLP}_{\text{gate}}(h))$, where MLP_{gate} is $\text{Linear}(d, d/r) \to \text{ReLU} \to \text{Linear}(d/r, d) \to \text{Sigmoid}$ with reduction ratio r=4. The pre-sigmoid bias is initialized to 1.0 (sigmoid ≈ 0.73). To keep PPD's feature scale stable from training to inference, an EMA tracks per-dimension mean gate activation $\bar{g}_d \leftarrow \mu \cdot \bar{g}_d + (1-\mu)\cdot \mathbb{E}[g_d(h)]$ with $\mu = 0.99$. Writing $c(u) = \max(u, 0.1)$, the gated output is normalized by the lower-bounded EMA:

$$h_{\text{gated}} = h \odot g(h) \;/\; c(\bar{g}). \tag{20}$$

Level 2: Structured noise injection. A learned noise generator outputs $n(h)$=$\text{Softplus}(\text{MLP}_{\text{noise}}(h)) \cdot \alpha_{\text{eff}} \cdot \epsilon$ with $\alpha_{\text{eff}} = \max(\alpha, \alpha_{\min})$, $\epsilon \sim \mathcal{N}(0, I)$, a learnable scalar α initialized to 0.1, and $\alpha_{\min}$=0.05. Noise is modulated by the curriculum strength $s \in [0, 1]$. The noise MLP shares the gate-MLP architecture with its output bias initialized to -3.0 (softplus ≈ 0.05). The main-body PPD equation Equation (7) summarizes Levels 1–2.

Level 3: Anchor degradation. Anchor embeddings a_i carry geometric information that the planning head can use to bypass PPD applied to instance features. PPD applies independent gate and noise networks to anchor embeddings with separate EMA tracking:

$$\text{PPD}_{\text{anchor}}(a) = a \odot g_a(a)/c(\bar{g}_a) + n_a(a)\cdot s. \tag{21}$$

Level 4: Object-level dropout. To simulate *missed detections*, PPD zeros all fields of each object (instance feature, anchor embedding, query position) with probability $p_{\text{drop}} \cdot s$ (p_{drop}=0.1). Classification logits for dropped objects are filled with -10 so that sigmoid activation yields near-zero confidences. The dropout mask is propagated to the feature distillation and interface-alignment losses to exclude dropped agents from Hungarian matching.

Level 5: Geometric-field degradation. Raw geometric fields (`query_pos`, `prediction`) can otherwise bypass embedding-level degradation. PPD applies a learnable scalar gate $g_{\text{geo}} = \sigma(\theta_{\text{geo}})$, active at training and inference (via the same inverted-normalization pattern):

$$x'_{\text{geo}} = x_{\text{geo}} \cdot g_{\text{geo}}/c(\bar{g}_{\text{geo}}) + \epsilon_{\text{geo}} \cdot s, \tag{22}$$

with $\epsilon_{\text{geo}} \sim \mathcal{N}(0, \sigma^2_{\text{geo}} I)$, σ_{geo}=0.15 (training only). A *padding re-masking* step following all degradations restores zeros (features) and -10 (classification) at invalid positions.

Classification recomputation. Because classification logits emitted by the embedding layer reflect pre-PPD features, PPD recomputes them from degraded instance features, closing the bypass.

Two-sided gate-range regularizer. For each gate network $q \in \{\text{inst}, \text{anchor}, \text{geo}\}$, let $m_g^{(q)} = \text{mean}(g_q(\cdot))$ be the scalar mean of its current activations. The gate-range regularizer penalizes mean gate activations outside a prescribed interval:

$$\mathcal{L}_{\text{gate}} = \sum_{q \in \{\text{inst,anchor,geo}\}} \Big(\lambda_g \, [m_g^{(q)} - g_{\max}]_+^2 + \lambda_\ell \, [g_{\min} - m_g^{(q)}]_+^2 \Big), \tag{23}$$

where $[u]_+ = \max(0, u)$, and we use $g_{\max}$=0.85, $g_{\min}$=0.3, λ_g=0.1, λ_ℓ=0.05. The full training loss including this gate-range regularizer is

$$\mathcal{L}_{\text{full}} = \mathcal{L} + \mathcal{L}_{\text{gate}}, \tag{24}$$

where $\mathcal{L}$ is the main objective in Equation (11).

Processing order. The PPD forward pass inside PCM is: (1) gate + noise on instance features, (2) classification recomputation, (3) gate + noise on anchor embeddings, (4) geometric-field degradation, (5) object-level dropout (training only), (6) padding re-masking. The resulting features are then passed through the Interface Adapter (Equation (8)) and finally into the retained planning head.

Curriculum. PPD strength is ramped from zero to one over a configurable number of warmup epochs via $s \in [0, 1]$. The curriculum scales feature noise, object dropout, and geometric noise by s. Deterministic gates remain active throughout training and are regularized by Equation (23).

Teacher-conditioned feature regularization. The gating, noise-amplitude, and adaptation networks are optimized jointly with the embeddings. The inverted normalization (Equation (20)) rescales gated features using the running mean gate activation. PCM uses information-bottleneck-inspired degradation (Tishby et al., 1999) to align privileged simulator-state features with the teacher's perception-conditioned representation.

B.4 Distillation training details

Feature distillation. Feature distillation aligns the three pathways through which the student exposes the scene to the retained planning head: agent-instance tokens, map polyline tokens, and the ego token. Agent and map tokens are matched to their teacher counterparts by Hungarian assignment under cosine distance on the combined representation $f_S^{(i)} + a_S^{(i)}$, while the ego token is paired one-to-one. The bundled feature objective is

$$\mathcal{L}_{\text{feat}} = \frac{1}{\sigma_f} \sum_{c \in \{\text{a,m}\}} \frac{1}{\max\{1, |\mathcal{M}_c|\}} \sum_{(i,j) \in \mathcal{M}_c} \big\| h_S^{c,(i)} - h_T^{c,(j)} \big\|_2^2 \; + \; w_{\text{ego}} \big\| e_S - e_T \big\|_2^2, \tag{25}$$

where $h^{\text{a},(i)} = f^{(i)} + a^{(i)}$ for agent tokens, $h^{\text{m},(i)} = f^{\text{m},(i)}$ for map tokens, σ_f=5 (chosen to bring the feature term into the same dynamic range as the output and interface terms), and w_{ego}=1.0. Agents dropped by PPD object dropout are excluded from the matching set.

Output-loss sub-components. The output loss sums six sub-terms,

$$\begin{aligned} \mathcal{L}_{\text{out}} = \lambda_{\text{traj}} \, \mathcal{L}_{\text{traj}} + \lambda_{\text{head}} \, \mathcal{L}_{\text{heading}} + \lambda_{\text{plan}} \, \text{KL}_{\text{plan}} \\ + \lambda_{\text{mot}} \, \mathcal{L}_{\text{mot-traj}} + \lambda_{\text{mkl}} \, \text{KL}_{\text{mot}} + \lambda_{\text{st}} \, \mathcal{L}_{\text{status}}, \end{aligned} \tag{26}$$

with defaults $(\lambda_{\text{traj}}, \lambda_{\text{head}}, \lambda_{\text{plan}}, \lambda_{\text{mot}}, \lambda_{\text{mkl}}, \lambda_{\text{st}}) = (1.0, 0.5, 1.0, 0.5, 0.5, 0.5)$:

- *Trajectory regression*: planning modes matched by Hungarian assignment on per-command L2 distance, then Smooth-L1 on matched pairs.
- *Heading distillation*: per-step heading is derived from consecutive waypoint displacements, with L1 on $\sin\theta$ and $\cos\theta$ separately.
- *Planning-mode KL*: for SparseDrive, classification logits are reshaped to $(B, C_{\text{cmd}}, M_{\text{ego}})$ and aligned via KL with temperature T=2.0 and a T^2 scale. VAD skips this term because it lacks per-command classification.

- *Motion trajectory*: per-agent Hungarian matching and Smooth-L1, excluding PPD-dropped agents.
- *Motion-mode KL*: per-agent mode distributions aligned via KL (T=2.0) using the same matching as motion-trajectory.
- *Status alignment*: SparseDrive-only L1 on planning-head status output (speed, brake).

Single-phase schedule. The planning head is kept *frozen throughout training*. Gradient updates are restricted to the MLP embeddings, the Privileged Perception Degradation (PPD), the Interface Adapter (IA), and the trainable interface projector P. Freezing the planning head holds its parameters constant across these comparisons. The dynamic weight scheduler is therefore reduced to its constant regime: $(\lambda_{\text{feat}}, \lambda_{\text{out}}, \lambda_{\text{itf}}) = (1.0, 1.0, 1.0)$ throughout, so feature alignment, output distillation, and interface alignment are jointly optimized. Optimization uses AdamW with cosine annealing (initial learning rate 4×10^{-4}, weight decay 10^{-3}, $\eta_{\min}$ at 1% of the initial learning rate) over E_1 epochs, where E_1 denotes the single-phase training epoch count specified in Appendix B.5. Per-iteration gradients are clipped at $\|g\|_2 \leq 25$. The PPD strength s is linearly ramped from 0 to 1 over the first E_{warm}=5 epochs (matching the warmup specified in Appendix B.5); thereafter $s \equiv 1$ for the remaining epochs.

Interface hook. $\mathcal{L}_{\text{interface}}$ (Equation (9)) captures f_T^{itf} at the downstream activation of the selected edge via a forward hook at training time. The teacher activation is reduced to (B, d_T) before the linear projector P: conv-style teachers use global spatial mean, transformer-style teachers use a leading-axis pool folded back into the batch dimension. When the student's interface-feature batch differs from the teacher's (e.g., teacher produces $B \cdot N_{\text{cam}}$ samples), we fold extra leading axes into the student batch via averaging. The student aggregates the post-PPD, pre-IA feature by per-sample top-K (K=32) selection on feature norm, skipping samples with no valid instances.

B.5 Training and baseline settings

The student embeds privileged simulator-state vectors $x_{\text{agent}} \in \mathbb{R}^{N\times7}$ (position, size, yaw, velocity, with N=50), $x_{\text{map}} \in \mathbb{R}^{M\times P\times2}$ (lane polylines, with M=100 and P=20), and $x_{\text{ego}} \in \mathbb{R}^{10}$ (ego state). All feature dimensions are d=256. The planning head is copied from the teacher and frozen throughout training.

All baselines share the MLP-embedding student backbone, the retained (frozen) planning head, the optimizer, and the training schedule. They differ only in which distillation terms are active and whether the PPD and IA modules are attached. **Direct-GT** trains directly against ground-truth motion labels, with PPD and IA both disabled and no teacher-distillation terms used. **Output-only KD** keeps output distillation only (λ_{feat}=0, λ_{itf}=0). Both PPD and IA are disabled, so the student receives the clean teacher feature path with no perturbation or interface alignment. **FitNets KD** (Romero et al., 2015) matches intermediate features through a linear hint projector. PPD and IA are both disabled and the interface-alignment loss is off. **Hand-crafted noise + KD** is exactly E2M-Distill with PPD, IA, and the interface-alignment loss removed, while keeping the embedding-level Gaussian noise on the privileged simulator state described in Appendix B.2 (the same σ's on both agent and map inputs: σ_{pos}=0.3 m, σ_{size}=0.1 m, σ_{yaw}=0.05 rad, σ_{vel}=0.2 m/s, σ_{poly}=0.2 m). The noise is applied only at training time, only to GT entries selected by the validity mask, and is identical for SparseDrive and VAD students.

The synchronized Bench2Drive corpus contains 234,769 image/state samples and is reused across target stacks. A frozen-teacher pass caches activations and outputs for conversion.

E2M-Distill and KD baselines train for E_1=10 epochs on the standard Bench2Drive base training split, using a single fixed random seed of 42. E2M-Distill uses $(\lambda_{\text{feat}}, \lambda_{\text{out}}, \lambda_{\text{itf}}) = (1.0, 1.0, 1.0)$. PPD uses a 5-epoch curriculum warmup. The two-sided gate-range regularizer (Equation (23)) uses $g_{\max}$=0.85, $g_{\min}$=0.3, λ_g=0.1, λ_ℓ=0.05. We use AdamW with cosine annealing applied from the first epoch (no linear warmup), learning rate 4×10^{-4}, weight decay 10^{-3}, and gradient clipping at norm 25.0. Training uses NVIDIA H200 GPUs: one H200 with global batch size 64 for SparseDrive students, and two H200s with per-GPU batch size 32 (global batch size 64) for VAD students, where the larger card budget compensates for the higher per-step cost of the VAD teacher. Under these configurations the full E_1=10-epoch training run takes approximately 48 wall-clock hours per student

for both SparseDrive and VAD. Across all reported runs in Tables 1, 2 and 3, including pilot studies and configurations that did not appear in the final paper, the full project consumed roughly 2–3$\times$ the compute of the experiments tabulated above.

C INTERFACE-TARGET SELECTION

This appendix expands the main-body PPS description (Section 3.4) with the shared calibration setup, probe protocol, and single-signal baselines used in Section 4.3.

C.1 SHARED CALIBRATION AND PROBE PROTOCOL

Module enumeration and calibration. Starting from the teacher's enumerable sub-module list, we apply an include/exclude regex filter to define the candidate hooks. Candidate hooks are ordered by their observed execution sequence. We push $K{=}200$ calibration samples through the frozen teacher and capture each module's output via forward hooks, flattening spatial dimensions and mean-pooling trailing axes where necessary to keep memory bounded. Feature dimensions above $d_{\max}{=}512$ are deterministically strided before kernel or spectral operations, keeping computation in a fixed budget across teacher widths.

Linear probes. The probes used by PPS (Equation (10)) and by the semantic-change baseline are trained with an 80/20 sample split for $E_p{=}10$ steps of Adam on the calibration batches, and evaluated on the held-out fold under the higher-is-better convention (negative MSE for regression targets). We use linear probes. The two proxy regression targets are described in Section 3.4: the perception proxy regresses the per-frame number of valid surrounding agents from H_l, and the planning proxy regresses the 2D ego endpoint of the future trajectory from H_l. Both labels are extracted from the same calibration batches. The PPS asymmetry factor in Equation (10) converts held-out scores to per-layer percentile ranks $r_l^{\text{perc}}, r_l^{\text{plan}} \in [0,1]$, absorbing MSE scale mismatch and bounding $r_l^{\text{perc}} - r_l^{\text{plan}}$ in $[-1,1]$ regardless of feature magnitudes.

CKA dissimilarity. For each adjacent edge $b : H_{b-1} \to H_b$, PPS uses the linear-CKA dissimilarity

$$A_b = 1 - \frac{\left\|\bar{H}_{b-1}^\top \bar{H}_b\right\|_F^2}{\left\|\bar{H}_{b-1}^\top \bar{H}_{b-1}\right\|_F \left\|\bar{H}_b^\top \bar{H}_b\right\|_F} \in [0,1], \tag{27}$$

where $\bar{H}$ denotes column-centered activations. The Frobenius form remains well defined when adjacent feature widths differ. If either denominator factor is zero, we set $A_b = 0$ so that constant activations cannot be selected by an undefined ratio.

C.2 SIDE CONVENTION FOR REPORTED INTERFACES

Each localizer selects an edge $b^\star$ between two adjacent candidate activations $H_{b^\star-1} \to H_{b^\star}$, and we report that edge by its downstream module $H_{b^\star}$ in Table 3—this is the activation hooked by the interface-alignment loss (Equation (9)) and is therefore the operationally meaningful name to attach to an interface. Among the probe-based scores, PPS (Equation (10)) evaluates the upstream-side perception–planning rank gap because it asks whether perception-specific semantics have consolidated *by* edge b, whereas the semantic-change baseline (Equation (29)) evaluates downstream planning-proxy decodability after that edge. CKA change and spectral transition score geometric changes across candidate activations. We therefore separate the scoring convention from the training convention: regardless of how a localizer scores an edge, the selected target for retraining is always the downstream activation $H_{b^\star}$.

C.3 SINGLE-SIGNAL LOCALIZATION BASELINES

The three interface-localization baselines compared against PPS in Table 3 share the same calibration setup but reduce the interface score to a single signal. All three select an edge by $\arg\max_b$ of their respective per-edge score, and the selected edge is reported and hooked on the downstream side $H_{b^\star}$ as described in Appendix C.2. We list each score below.

CKA-change baseline. With A_b as in Equation (27), the CKA-change baseline selects

$$b^\star_{\text{CKA}} \;=\; \arg\max_b A_b, \tag{28}$$

equivalently, the PPS edge score with its upstream probe-rank factor replaced by a constant. This baseline tests whether CKA-style geometric reorganization alone can identify that interface.

Semantic-change baseline. Reusing the per-layer rank-normalized probe scores $r_l^{\text{perc}}, r_l^{\text{plan}} \in [0,1]$ of Equation (10), the semantic-change baseline scores each edge by the downstream-side decodability gap

$$C_b \;=\; r_b^{\text{plan}} \;-\; r_b^{\text{perc}}, \qquad b^\star_{\text{sem}} \;=\; \arg\max_b C_b, \tag{29}$$

i.e. the edge whose downstream layer is most planning-dominant relative to perception under the linear probes. This probe-only baseline tests whether downstream planning-proxy decodability alone selects a useful interface target. PPS combines the upstream perception–planning probe-rank gap with CKA change; this baseline uses the downstream planning–perception rank gap.

Spectral-transition baseline. With singular values $\{\sigma_i^{(l)}\}$ of the centered H_l and total energy $E_l = \sum_j (\sigma_j^{(l)})^2$, define $p_i^{(l)} = (\sigma_i^{(l)})^2 / E_l$ when $E_l > 0$. The three spectral descriptors are the effective rank (Roy & Vetterli, 2007) $\rho_l = \exp(-\sum_i p_i^{(l)} \log p_i^{(l)})$, the spectral entropy $S_l = -\sum_i p_i^{(l)} \log p_i^{(l)}$, and the participation ratio (Gao et al., 2017) $R_l = (\sum_i (\sigma_i^{(l)})^2)^2 / \sum_i (\sigma_i^{(l)})^4$. If $E_l = 0$, the centered activation is constant. We set $\rho_l = S_l = R_l = 0$ to avoid an undefined normalization and to prevent a constant layer from contributing artificial spectral evidence. The spectral-transition baseline selects

$$b^\star_{\text{spec}} \;=\; \arg\max_b B_b, \qquad B_b \;=\; \big|\rho_b - \rho_{b-1}\big| \;+\; \big|S_b - S_{b-1}\big| \;+\; \big|R_b - R_{b-1}\big|, \tag{30}$$

matching the non-monotonic intrinsic-dimensionality curves of trained hierarchies (Ansuini et al., 2019). This single-signal geometric baseline provides an alternative to the CKA-style transition evidence used by PPS.

C.4 Interface-target validation and selected-edge scores

Table 3: PPS interface-target validation. Rows compare the PPS-selected teacher target activation with single-signal interface-localization baselines. For every localizer we report the downstream module $H_{b^\star}$ of the selected edge $b^\star$, which is the activation hooked by the interface-alignment loss (Equation (9)). Bold marks the best value within each teacher family.

Interface localization method	Teacher target activation	SOA↑	DS_{corr} ↑	$\text{Inf}_{\text{Jaccard}}$ ↑
SparseDrive teacher				
PPS (ours)	`img_neck.fpn_convs.0`	**67.30**	**0.5863**	**0.5768**
Downstream semantic	`img_neck.lateral_convs.0`	62.84	0.5558	0.5699
CKA change	`img_neck.lateral_convs.1`	65.50	0.5642	0.5671
Spectral transition	`img_neck.lateral_convs.2`	63.20	0.5598	0.5583
VAD teacher				
PPS (ours)	`pts_bbox_head.transformer`	**66.40**	**0.5251**	**0.5097**
Downstream semantic	`img_backbone.layer2.1`	56.80	0.4957	0.4697
CKA change	`pts_bbox_head.transformer`	**66.40**	**0.5251**	**0.5097**
Spectral transition	`img_neck.fpn_convs.0`	59.10	0.4958	0.4811

The all-edge PPS scan selects `img_neck.lateral_convs.3` → `img_neck.fpn_convs.0` for SparseDrive, with score 0.8017 and a margin of 0.0974 over the runner-up. For VAD, it selects `img_neck.fpn_convs.3` → `pts_bbox_head.transformer`, with score 0.9255 and a margin of 0.3084. The downstream activations are the alignment targets evaluated in Table 3.

D Evaluation Protocol and Metrics

This appendix gives the full formal definitions of the closed-loop teacher–student consistency metrics and the teacher-relative open-loop error introduced in Section 4.1.

Throughout, T and S denote teacher and student. We let $n{=}220$ index the Bench2Drive routes. For route i, DS_i^T and DS_i^S denote the per-route driving scores, $s_i^T, s_i^S \in \{0,1\}$ denote the per-route success indicators (the SR criterion of (Jia et al., 2024): route completed within the time budget without any rule infraction), and $\mathcal{I}_i^T, \mathcal{I}_i^S$ denote the CARLA infraction-type sets. Finally, $\mathcal{F}$ denotes the Bench2Drive validation-frame index set.

D.1 Evaluation setup

Closed-loop evaluation, open-loop evaluation, and per-step latency profiling all use one NVIDIA RTX 4090. Closed-loop rollouts run on CARLA 0.9.16 under the official Bench2Drive protocol.

D.2 Closed-loop consistency

For each route $i \in \{1, \dots, n\}$, the teacher and student are independently rolled out under the same CARLA seed, weather, and traffic configuration, producing paired per-route summaries $(\mathrm{DS}_i^T, s_i^T, \mathcal{I}_i^T)$ and $(\mathrm{DS}_i^S, s_i^S, \mathcal{I}_i^S)$.

Scenario Outcome Agreement (SOA, %). The fraction of routes on which teacher and student agree on the binary success indicator:

$$\mathrm{SOA} \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\big[s_i^T = s_i^S\big] \times 100\%. \tag{31}$$

Driving-Score Correlation ($\mathbf{DS}_{\mathrm{corr}}$, $\in [-1,1]$). Pearson correlation across routes between paired per-route driving scores:

$$\mathrm{DS}_{\mathrm{corr}} \;=\; \frac{\sum_{i=1}^{n}\big(\mathrm{DS}_i^T - \overline{\mathrm{DS}^T}\big)\big(\mathrm{DS}_i^S - \overline{\mathrm{DS}^S}\big)}{\sqrt{\sum_{i=1}^{n}\big(\mathrm{DS}_i^T - \overline{\mathrm{DS}^T}\big)^2 \sum_{i=1}^{n}\big(\mathrm{DS}_i^S - \overline{\mathrm{DS}^S}\big)^2}}, \tag{32}$$

where $\overline{\mathrm{DS}^T}$ and $\overline{\mathrm{DS}^S}$ are the route-average teacher and student driving scores. If either centered score vector has zero norm, we define $\mathrm{DS}_{\mathrm{corr}} = 1$ when both vectors are identical constants and $\mathrm{DS}_{\mathrm{corr}} = 0$ otherwise.

Infraction-Type Jaccard ($\mathbf{Inf}_{\mathrm{Jaccard}}$, $\in [0,1]$). Mean route-wise Jaccard similarity between the sets of triggered CARLA infraction types:

$$\mathrm{Inf}_{\mathrm{Jaccard}} \;=\; \frac{1}{n}\sum_{i=1}^{n} J_i, \qquad J_i \;=\; \begin{cases} 1 & \text{if } \mathcal{I}_i^T = \mathcal{I}_i^S = \varnothing, \\ \dfrac{\big|\mathcal{I}_i^T \cap \mathcal{I}_i^S\big|}{\big|\mathcal{I}_i^T \cup \mathcal{I}_i^S\big|} & \text{otherwise.} \end{cases} \tag{33}$$

The infraction-type vocabulary follows the official CARLA Leaderboard v2 categorization adopted by Bench2Drive (Jia et al., 2024) (e.g., `collision_pedestrian`, `collision_vehicle`, `collision_layout`, `red_light`, `stop_infraction`, `outside_route_lanes`, `route_deviation`, `vehicle_blocked`, `agent_blocked`, `timeout`, `min_speed_infraction`). Each triggered infraction type contributes once to the route-level set.

D.3 Teacher-relative open-loop error

For each frame $f \in \mathcal{F}$, the teacher and student are run in open-loop forward-pass mode on the same input snapshot, producing planned ego waypoints $\tau_{f,\delta}^T, \tau_{f,\delta}^S \in \mathbb{R}^2$ at prediction horizons $\delta \in \{0.5, 1.5, 3.0\}$ s. Both trajectories are expressed in the ego frame at time f, so the metric is

invariant to global localization. The teacher-relative open-loop error at horizon δ is the per-frame Euclidean distance between paired teacher and student waypoints, averaged over frames:

$$\mathrm{OL}_\delta \;=\; \frac{1}{|\mathcal{F}|}\sum_{f\in\mathcal{F}}\left\|\tau^T_{f,\delta}-\tau^S_{f,\delta}\right\|_2, \qquad \delta\in\{0.5,1.5,3.0\}\,\mathrm{s}. \tag{34}$$

Lower is better. For multi-modal planners (SparseDrive, with M_{ego}=6 ego modes), $\tau^S_{f,\delta}$ is the student mode that minimizes per-frame command-conditioned L_2 to the teacher's mode under Hungarian matching, matching the assignment used at training time (Appendix B.4). The same selection is applied to $\tau^T_{f,\delta}$ to ensure paired evaluation. Frames in which either teacher or student fails to produce a valid plan are excluded from $\mathcal{F}$.

The metric measures teacher–student planning agreement using the teacher trajectory as the reference. Reporting at three horizons exposes whether mismatch grows or stabilizes over the prediction window.

D.4 Complementarity of the three closed-loop metrics

The three closed-loop metrics are designed to be complementary and are reported jointly throughout Section 4. We summarize the three failure cases that motivate joint reporting:

- **High SOA, low $\mathbf{DS}_{\mathrm{corr}}$.** Teacher and student agree on success or failure, but their scores on failed routes vary differently: a route with a high teacher score can have a low student score, and vice versa. Binary outcomes agree while score correlation is low.
- **High $\mathbf{DS}_{\mathrm{corr}}$, low $\mathbf{Inf}_{\mathrm{Jaccard}}$.** The student matches per-route driving scores but achieves them through different infraction patterns — e.g., the teacher's score loss comes from minor red-light infractions while the student's comes from collisions of equivalent multiplicative penalty. The student matches scores but not failure modes.
- **High SOA, low $\mathbf{Inf}_{\mathrm{Jaccard}}$.** The student succeeds on the same routes the teacher succeeds on, but on the failed routes the failure modes are unrelated. The student matches binary outcomes but not the structure of failure.

For the AV-Stack-to-Sim-Agent Conversion task studied here, SOA, $\mathrm{DS}_{\mathrm{corr}}$, and $\mathrm{Inf}_{\mathrm{Jaccard}}$ are complementary teacher–student fidelity dimensions and should be read jointly.

D.5 Repeated-evaluation protocol

We repeat the complete 220-route simulation under three simulator-seed configurations, keeping the target AV stack, converted-agent checkpoint, route set, and evaluation protocol fixed. Table 5 reports the mean and sample standard deviation over the three complete runs.

E Additional Results

E.1 Additional target stacks and controlled PPD comparison

SparseDriveV2. We evaluate conversion of SparseDriveV2 (Sun et al., 2026), whose target-stack Bench2Drive DS is 81.42 in our evaluation. E2M-Distill and Hand-crafted noise + KD use the same SparseDriveV2-based student framework, a single 10-epoch training run with seed 42, and the complete 220-route evaluation.

Target stack trained on real data. We also convert the official SparseDrive checkpoint trained on nuScenes (Caesar et al., 2020). The frozen target stack supplies activations and outputs on synchronized CARLA conversion data, and the converted agent is evaluated on the full 220-route Bench2Drive suite. E2M-Distill and a fixed-noise control use the same conversion data and settings. The control bypasses learned PPD and retains fixed Gaussian input perturbations, IA, and interface alignment.

Controlled PPD comparison. For the SparseDrive teacher used in Table 1, we hold fixed the teacher checkpoint, MLP embeddings, frozen planning head, PPS target `img_neck.fpn_convs.0`, IA,

feature/output/interface-alignment losses, and fixed Gaussian perturbations to agent, map, and ego inputs. The fixed-noise control bypasses learned PPD and uses the same 10-epoch schedule, global batch size 64, optimizer, and checkpoint-selection rule, with training seed 42. Results are reported in Table 4.

Table 4: Additional target stacks and controlled PPD comparison on 220 routes. The fixed-noise control retains IA and interface alignment.

Target stack	Conversion method	SOA (%)	$\mathrm{DS}_{\mathrm{corr}}$	$\mathrm{Inf}_{\mathrm{Jaccard}}$
SparseDriveV2	E2M-Distill	73.2	0.4768	0.6298
	Hand-crafted noise + KD	71.4	0.4451	0.5769
SparseDrive (nuScenes)	E2M-Distill	66.4	0.6981	0.7510
	Fixed-noise control	63.2	0.6859	0.7170
SparseDrive (Table 1)	E2M-Distill	67.3	0.5863	0.5768
	Fixed-noise control	64.1	0.5655	0.5603

E.2 Repeated closed-loop evaluation

The repeated-evaluation protocol is given in Appendix D.5. E2M-Distill has higher mean consistency than the corresponding baseline on all three metrics for both target stacks.

Table 5: Closed-loop consistency over three simulator-seed configurations with fixed trained checkpoints (mean $\pm$ sample standard deviation).

Target stack	Method	SOA (%)	$\mathrm{DS}_{\mathrm{corr}}$	$\mathrm{Inf}_{\mathrm{Jaccard}}$
SparseDrive	E2M-Distill	67.57 ± 0.23	0.5839 ± 0.0045	0.5746 ± 0.0061
	Hand-crafted noise + KD	66.23 ± 0.67	0.5712 ± 0.0050	0.5464 ± 0.0008
VAD	E2M-Distill	65.93 ± 0.45	0.5219 ± 0.0047	0.5115 ± 0.0018
	Direct-GT	64.40 ± 0.52	0.5139 ± 0.0013	0.5031 ± 0.0029

E.3 Official Bench2Drive driving metrics

Table 6: Official Bench2Drive closed-loop driving performance (220 routes). DS = Driving Score, RC = Route Completion (%), IP = Infraction Penalty, SR = Success Rate (%).

Run	DS↑	RC↑	IP↑	SR↑
SparseDrive teacher				
Teacher	49.42	60.46	0.8466	47.27
E2M-Distill	64.08	82.90	0.7395	70.45
VAD teacher				
Teacher	59.53	71.36	0.8416	57.73
E2M-Distill	51.95	85.89	0.5892	74.09

F Computational Cost and Scaling

F.1 Latency scaling

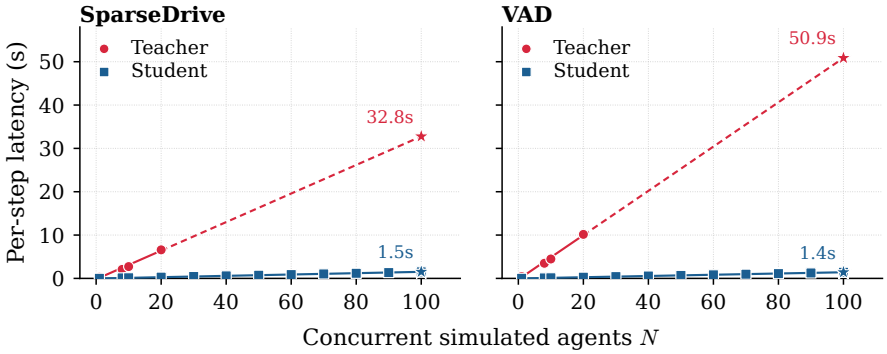


Figure 3: Per-step latency versus agent count V for both teacher families and their E2M-Distill students. Data points are the *Efficiency* columns of Table 1. Teacher points at $V=100$ are extrapolated because the full sensor-rendering pipeline cannot complete that run in our setting.

F.2 Student-forward latency

We profile each trained student on one NVIDIA H200 with batch size 1, using a fixed validation input within each teacher family, 50 warm-up iterations, and 200 synchronized timed iterations. The timed operation is the rendering-free student forward pass including planning-head post-processing. Data loading, host-to-device transfer, CARLA, and orchestration are outside this timing interval. Table 7 reports the median (p50) and 95th-percentile (p95) latency. The four baseline checkpoints share the MLP embedding pathway and retained planning head with PCM disabled; enabling PCM adds 791,558 active parameters.

Table 7: Student-forward latency on one H200 (ms, p50 / p95).

Student	SparseDrive	VAD
E2M-Distill	13.97 / 14.32	9.75 / 9.78
Direct-GT	12.97 / 13.01	8.63 / 8.65
Output-only KD	12.98 / 13.02	8.66 / 8.69
FitNets KD	12.99 / 13.04	8.63 / 8.67
Hand-crafted noise + KD	13.00 / 13.05	8.66 / 8.69