LEARNING END-TO-END DRIVING POLICIES WITHOUT EXPERT DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end autonomous driving has achieved state-of-the-art performance on benchmarks and real-world deployments. Its standard training recipe, however, is expensive across all existing paradigms: imitation learning requires fleet-scale driving logs whose behavior distribution only thinly covers the safety-critical tail; closed-loop reinforcement learning in a photorealistic simulator, while offering diverse behaviors, is bottlenecked by sample-efficiency, per-step cost of rendering, and a forward pass through a large vision backbone. Self-play in vectorized simulators changes the economics: at millions of rollout steps per second, one can obtain a state distribution with natural diversity in collisions, near-misses, and recoveries that no driving log contains, and a highly reactive policy emerges from a basic driving-quality reward. Our approach exploits this asymmetry by decoupling *learning to drive* from *learning to see*. We pretrain a single policy by self-play, then align a pretrained vision backbone to its latent space through the action Kullback-Leibler divergence and a batch-relational low-rank structural loss. The action target comes from the self-play policy, so alignment never supervises against a logged trajectory: a paired dataset of (image, scene-state) frames suffices, with no need for the curated expert demonstrations that imitation pretraining is built on. The proposed framework establishes a novel learning paradigm that substantially reduces the cost of end-to-end driving policy learning, while the policy learned significantly outperforms every published imitation-based end-to-end method on photorealistic 3D Gaussian splatting closed-loop benchmarks.

1 INTRODUCTION

End-to-end autonomous driving maps raw sensor inputs (*e.g.*, camera images) directly to control commands. Deep learning and large-scale driving data have made this approach widely adopted and continuously improved across benchmarks (Bojarski et al., 2016; Codevilla et al., 2019; Hu et al., 2023; Jiang et al., 2023; Chitta et al., 2023; Dauner et al., 2024; Zhou et al., 2024). However, this progress relies on expensive standard pipelines that follow either of the two routes.

The open-loop strategy draws its behavioral supervision from fleet logs: imitation copies logged human driving (Codevilla et al., 2019; Hu et al., 2023; Jiang et al., 2023; Chitta et al., 2023), and open-loop reinforcement learning (RL) optimizes against logged trajectories (Karkus et al., 2025). These logs provide millions of multi-camera hours (Caesar et al., 2020; Sun et al., 2020; Caesar et al., 2021), yet safety-critical events (collisions, near-misses, recoveries, *etc.*) remain vanishingly rare in natural driving: at NHTSA crash rates, a collision occurs only once every $\sim$16k driving hours (National Highway Traffic Safety Administration, 2025), so logging a single rare event can easily cost over $1M in fleet operation alone (Kelkar et al., 2025). Each sub-route inherits this scarcity in its own way: imitation can only copy the actions the recorded drivers took, and open-loop RL never sees how other vehicles would have responded to the alternative actions it explores. Broadening interaction coverage with human data therefore demands ever more collection and curation.

The closed-loop framework instead trains inside a photorealistic, sensor-rendering simulator (Dosovitskiy et al., 2017; Chen et al., 2019; Zhang et al., 2021; Wu et al., 2022; Chitta et al., 2023; Karkus et al., 2025), where the policy experiences the consequences of its own actions. This interaction, however, costs rendering and a heavy vision-backbone forward pass at every step. Co-training multiple interacting agents multiplies these per-step costs, so most image-based closed-loop work instead

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.

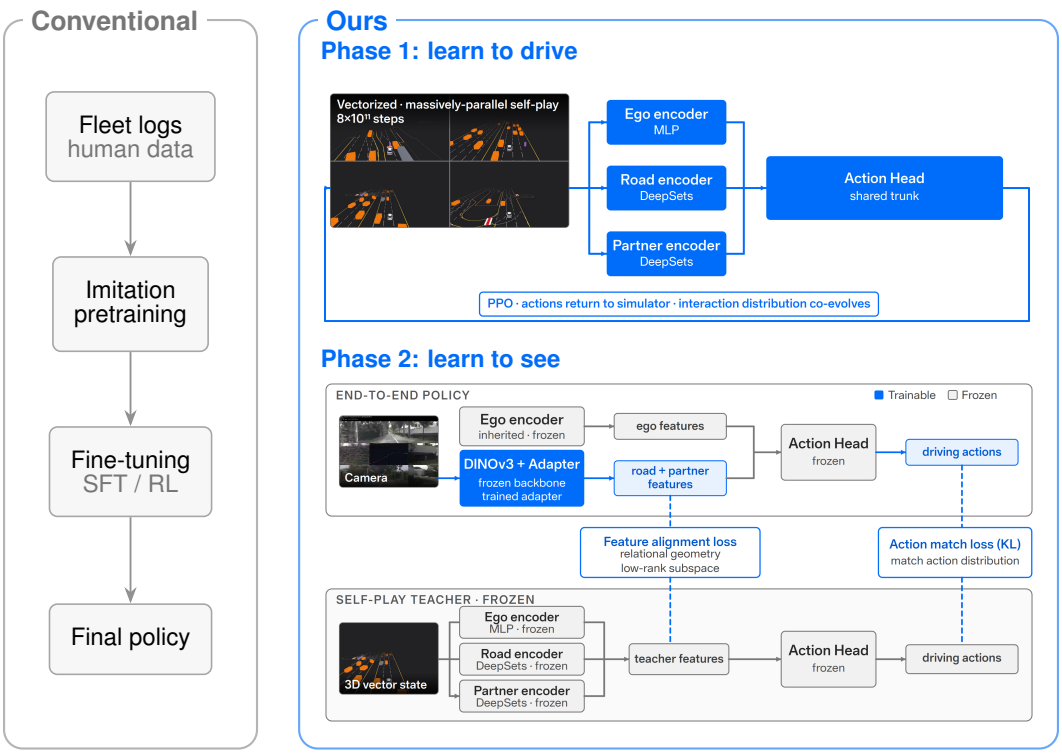


Figure 1: **Conventional training pipeline vs. our reinforcement-learning pretraining pipeline.** Conventional recipes imitate fleet logs, then fine-tune on curated demonstrations, logged trajectories, or rendered simulations. Each conventional stage requires human data or rendering. *Phase 1* (top right) uses Proximal Policy Optimization (PPO) to train a planning head through self-play, where one policy controls all agents in a multi-agent vectorized simulator. *Phase 2* (bottom right) freezes this teacher. The student then matches the teacher's action distributions and pairwise scene-feature similarities on paired (image, scene-state) frames, producing the head's scene features from images. The action and structural losses require no demonstrations or rendering during training. Deep blue modules are trainable.

trains the ego policy against scripted or log-replayed traffic, and the behavior distribution stays close to the logged or hand-designed content (Dosovitskiy et al., 2017; Chen et al., 2019; Zhang et al., 2021; Wu et al., 2022). These costs confine this route to fine-tuning a pretrained imitation policy, as in Figure 1, rather than pretraining driving competence from scratch.

Recently, multi-agent self-play has risen as a promising alternative to offer the missing interaction: one shared policy controls every agent, so each policy improvement reshapes the traffic every other agent faces, and the interactive behavior co-evolves with the policy. Vectorized simulation makes this co-training highly efficient, executing millions of steps per second on commodity hardware (Cusumano-Towner et al., 2025; Chang et al., 2026; Seong et al., 2025; Guo et al., 2026). Training requires only a basic definition of driving quality as per-step rewards: goal reaching, collision avoidance, comfort, and simple rule compliance. These rewards specify general driving objectives rather than separate rules for corner cases. The resulting interactions naturally produce the collisions, recoveries, yielding, negotiation, and overtaking that logged traffic seldom contains, all without expert trajectories or behavior-specific rewards.

Despite its promise, however, self-play does not yield deployable end-to-end driving policies *per se*, as they observe vectorized scene state rather than raw sensors. Deploying these policies would require a two-stage perception stack that recovers their HD-map-style inputs from images. Such a stack demands high-quality perception, is sensitive to perception noise, and forfeits the co-optimization of a fully differentiable end-to-end model. As a result, it has mostly remained confined to traffic simulation rather than an end-to-end solution (Cusumano-Towner et al., 2025; Chang et al., 2026; Seong et al., 2025; Guo et al., 2026; Konstantinidis et al., 2026; Ahmadi et al., 2026).

In this work, we unite the strengths of self-play and end-to-end driving in a paradigm that learns driving policies without expert demonstrations. Self-play learns driving competence without human demonstrations but consumes vector state; end-to-end recipes consume raw sensors but depend on human demonstrations, whose coverage of rare interactions is the very gap identified above. We bridge this gap by decoupling *learning to drive* from *learning to see* and never use logged demonstration trajectories as supervision. Figure 1 contrasts its two stages with the conventional workflow.

In the first stage, we train a planning head *tabula rasa* through multi-agent self-play in a vectorized driving simulator, in the spirit of GigaFlow (Cusumano-Towner et al., 2025). One parameter set controls every agent in every scene. The head learns from the long-tail distribution of its own behavior and never imitates a human trajectory. The resulting policy is a strong planner, scoring 94.19 on nuPlan val14, 70.87 on the interactive long-tail InterPlan suite, and 0.632 human-similarity on WOSAC 2023. These scores exceed those of every imitation-based planner we are aware of. In stage two, we freeze the planning head and use it as a teacher. The student retains this head unchanged and learns encoders that produce its scene features for road elements and surrounding agents from images instead of vector states; these encoders pair a pretrained vision backbone (Siméoni et al., 2025) with learned adapters supervised by paired (image, scene-state) frames of the same scene. The teacher's per-state action distribution trains the student to choose the same actions. A feature-level *structural* loss preserves the teacher's pairwise similarities among scene features, following the empirical finding that the teacher's features are low-rank and redundant (§4.3).

Our framework preserves the naturalistic driving competence learned through self-play. On photorealistic closed-loop benchmarks, our end-to-end student attains an aggregate HD-Score of 0.548, exceeding every published end-to-end baseline (Table 1).

We make the following contributions:

- **A novel demonstration-free paradigm for learning end-to-end driving policies.** A planning head first learns to drive through multi-agent self-play in a vectorized simulator. Its per-state action distributions then supervise encoders that produce the head's scene features from images, with no logged-trajectory supervision at any stage.
- **A feature-level structural alignment mechanism.** We align the encoder directly with the teacher's pooled scene features instead of cascading reconstruction and detection. A batch-relational low-rank *structural* loss matches the pairwise scene similarities induced by the teacher.
- **Self-play driving performance beyond published imitation-based end-to-end methods.** On photorealistic closed-loop scenarios in HUGSIM, the teacher outperforms every published imitation-based end-to-end method without observing a single human demonstration.

2 Related Work

Most end-to-end driving policies imitate logged human driving (Codevilla et al., 2019; Hu et al., 2023; Jiang et al., 2023). However, driving competence can also be learned without human demonstrations through multi-agent self-play in vector-state simulators such as GigaFlow (Cusumano-Towner et al., 2025), TerraZero (Wu et al., 2026), and PufferDrive (Cornelisse et al., 2025). This raises the question of how to bring the resulting competence to camera-based policies.

Several systems pursue this goal along two routes. The first route learns to drive in vector state and then transfers the behavior to images. Concurrent with our work, GigaPixel (Rowe et al., 2026) distills its vector-state teacher into a pixel-based student through DAgger in a bounding-box renderer, then freezes that student's head to adapt perception to real images. DriveZero (He et al., 2026) instead trains a separate student planner, supervised by trajectories that a PPO teacher rolls out from logged frames. Although their transfer procedures differ, neither reuses the teacher's planning head. The second route learns the behavior directly in image space, skipping the vector-state teacher. Pictura (Yin et al., 2026), also concurrent, renders each agent's egocentric view at every self-play step and trains policies from these non-photorealistic perspective images. Cheap rendering scales the interaction, but the resulting policies remain in the renderer's domain; a policy that consumes real camera images still requires a further training stage. Photorealistic 3D Gaussian splatting simulators close this visual gap, but rendering every rollout at photorealistic quality is costly, so works like

RAD (Gao et al., 2025) and World Engine (Li et al., 2026) use them only to post-train imitation-pretrained image policies rather than to learn driving from scratch.

We take the first route, but retain the planning head learned through vector-state self-play and train encoders that produce its scene features from images instead of vector state. RAP (Feng et al., 2026) likewise pairs lightweight rasterization with feature alignment, but applies them to imitation learning: it rasterizes counterfactual views from annotated logs and aligns raster features to real ones, whereas our alignment transfers the decisions of a self-play teacher. Concretely, on fixed paired (image, scene-state) frames we compute action-distribution targets per state and feature-relation targets across the batch. Because these frames are fixed, alignment requires neither online rendering nor teacher rollouts.

3 METHOD

We first train a vector-state teacher through self-play, then align a vision student to it with structural and action losses.

3.1 SELF-PLAY PRETRAINING

Setup. We pretrain a driving policy π_θ by self-play in a vectorized driving simulator. A single set of parameters θ controls all agents in every scene. We pool their transitions into one on-policy buffer. Each agent's "surrounding traffic" consists of other instances of the current policy. The multi-agent interaction distribution therefore evolves with the policy itself.

Policy architecture. At each step, an agent observes its own ego state o_t^e. This state contains its kinematics and the per-episode reward-weight vector $w^{(n)}$ introduced below. The agent also observes a variable-sized set of road elements $\mathcal{O}_t^r = \{o_{t,i}^r\}_{i=1}^{M_r}$ and a variable-sized set of partner agents $\mathcal{O}_t^p = \{o_{t,j}^p\}_{j=1}^{M_p}$. Three encoders process these inputs in parallel:

$$\hat{f}_t^e = E_{\text{ego}}(o_t^e), \quad \hat{f}_t^r = E_r(\mathcal{O}_t^r), \quad \hat{f}_t^p = E_p(\mathcal{O}_t^p). \tag{1}$$

The ego encoder E_{ego} is a multilayer perceptron (MLP), while E_r, E_p are set-processing DeepSets (Zaheer et al., 2017) encoders. Each DeepSets encoder max-pools its element embeddings. For the road encoder, the pooling rule is $E_r(\mathcal{O}_t^r) = \text{maxpool}_i\, \phi_r(o_{t,i}^r) + b_r$; E_p uses the same rule. The learned biases b_r, b_p provide a well-defined output when the corresponding set is empty. We concatenate the three features and pass them through a shared MLP. An actor head then produces a categorical distribution over a discrete action set $\mathcal{A}$ (see Appendix A.4):

$$\pi_\theta(\cdot \mid o_t) = \text{softmax}\Big(W_a \cdot \text{MLP}_{\text{shared}}\big([\hat{f}_t^e,\, \hat{f}_t^r,\, \hat{f}_t^p]\big)\Big). \tag{2}$$

Reward. Self-play supplies the experience; the reward defines what counts as good behavior within it. We define the per-step reward $R_t^{(n)}$ for agent n as a weighted sum of K driving-quality terms:

$$R_t^{(n)} \;=\; \sum\nolimits_{k=1}^{K} w_k^{(n)} \cdot r_t^{(k)}\big(o_t^{(n)},\, a_t^{(n)}\big). \tag{3}$$

The terms measure goal-reaching, collision, comfort, lane alignment, lane centering, velocity, traffic-rule compliance, and related driving qualities (see Appendix A.3).

Following the domain-randomization scheme of GigaFlow (Cusumano-Towner et al., 2025), we resample the weights $w_k^{(n)} \sim \mathcal{U}(w_k^{\min}, w_k^{\max})$ for each agent and episode. We include the sampled weights in the policy's ego observation o_t^e. This conditioning trains a single π_θ to act consistently across a continuum of driver preferences.

Objective. We optimize the clipped proximal policy optimization (PPO) surrogate (Schulman et al., 2017) with a value loss and entropy bonus:

$$\mathcal{L}(\theta) = -\mathbb{E}_t\Big[\min\big(\rho_t \hat{A}_t,\; \text{clip}(\rho_t,\, 1{-}\epsilon,\, 1{+}\epsilon)\hat{A}_t\big)\Big] \;+\; c_v\, \mathcal{L}_V(\theta) \;-\; c_H\, \mathcal{H}[\pi_\theta(\cdot \mid o_t)]\,. \tag{4}$$

Here, $\rho_t = \pi_\theta(a_t \mid o_t)/\pi_{\theta_{\text{old}}}(a_t \mid o_t)$ is the policy probability ratio. The term $\hat{A}_t$ is the generalized advantage estimation (GAE) advantage (Schulman et al., 2015). The clipping threshold is ϵ. The coefficients c_v and c_H weight the value loss and entropy bonus, respectively.

3.2 VISION ALIGNMENT

Setup. After self-play, we freeze π_θ as the teacher π_θ^T and construct an image-conditioned student. The teacher supplies per-frame supervision. The student keeps the teacher's frozen planning head, comprising the shared MLP and actor head. It also keeps the frozen ego encoder E_{ego}. The student replaces only its own road and partner encoders with image-conditioned counterparts $E_r^{\text{vis}}, E_p^{\text{vis}}$; the frozen teacher remains unchanged. Given an image I_t paired with the teacher's vector observation o_t, the student and teacher features are

$$\big(f_t^r,\, f_t^p\big) = \big(E_r^{\text{vis}}(I_t),\, E_p^{\text{vis}}(I_t)\big), \qquad \big(\hat{f}_t^r,\, \hat{f}_t^p\big) = \big(E_r(\mathcal{O}_t^r),\, E_p(\mathcal{O}_t^p)\big). \tag{5}$$

The frozen DeepSets encoders E_r, E_p from Eq. 1 produce the teacher features $\hat{f}_t^r, \hat{f}_t^p$, exactly as during self-play. The two vision encoders share a single frozen DINOv3 (Siméoni et al., 2025) backbone and differ only in their adapters: E_r^{vis} and E_p^{vis} apply separate linear adapters to the same backbone features, producing the road and partner features f_t^r, f_t^p (Fig. 1). Under a fixed alignment recipe, this backbone gives the strongest closed-loop score among the alternatives we ablate (Appendix C).

The student receives camera images together with the ego vector, whereas the teacher additionally receives vector road and partner state. A possible bridge is a *cascade*: a perception module recovers the teacher's vector inputs (bounding boxes, lane polylines, and partner kinematics) from the image, and the frozen teacher then consumes these recovered inputs. We avoid this cascade for two reasons. First, it is not end-to-end differentiable: the perception module trains on surrogate detection objectives, so no gradient couples its errors to the driving quality that the frozen head serves, and the perception stage makes any future end-to-end post-training of the policy harder. Second, the teacher trains on clean simulator state and is sensitive to input quality; perception errors at its input perturb the action distribution in ways the frozen head never saw during self-play. Our ablation confirms this empirically: a modular cascade under the same data and compute budget scores below our direct alignment (§4.3). We therefore adopt the pooled feature as the alignment target and train image-conditioned encoders to produce it directly.

We supervise the student with two complementary objectives. The action loss aligns policy distributions, but a softmax is coarse: different feature configurations can yield the same action logits. The structural loss matches feature relations in the informative low-rank subspace of §4.3. It preserves how the teacher arranges scenes relative to one another, but its pairwise angles are invariant to rotations and shifts, so it alone cannot ensure that the frozen head maps features to the right actions. Together, the action loss anchors actions and the structural loss shapes scene geometry.

Action alignment. Following classical distillation (Hinton et al., 2015), we match the student's policy distribution to the teacher's by forward KL divergence,

$$\mathcal{L}_{\text{act}} \;=\; \mathrm{KL}\big(\pi^T(\cdot \mid o_t) \,\|\, \pi^S(\cdot \mid I_t, o_t^e)\big). \tag{6}$$

We match the full distribution rather than the sampled action: the distribution carries the teacher's uncertainty across the action set, so near-ambiguous scenes teach the student the same ambiguity instead of a single arbitrary choice. This policy-level objective provides a global consistency signal. Minimizing it encourages the student's policy distribution to approach the teacher's, regardless of how the student represents road and partner features.

Structural alignment. Action matching can leave different feature configurations indistinguishable. We therefore also match scene geometry: scenes the teacher treats as alike should remain alike for the student. This relational target complements the action loss rather than adding a second coordinate-regression objective.

The informative variation in the teacher's features is sharply low-rank, as shown in §4.3. Matching every coordinate would also force the student to fit the structureless tail. An optimal low-rank denoiser discards these low-variance directions as noise (Gavish & Donoho, 2014). We call the span of the informative directions the *teacher principal subspace* and align features only within it.

For a batch of size B, we process each encoder $* \in \{r, p\}$ separately. We first mean-center the student and teacher features, obtaining matrices $\bar{F}_*, \hat{\bar{F}}_* \in \mathbb{R}^{B\times d}$. To extract the teacher principal

subspace, we compute a per-batch basis from the top-k_* right singular vectors of the stop-gradient teacher matrix:

$$\bar{\bar{F}}_* \overset{\text{SVD}}{=} U_* \Sigma_* V_*^\top, \qquad V_*^{(k_*)} = \left[\, v_{*,1}, \ldots, v_{*,k_*} \,\right] \in \mathbb{R}^{d \times k_*}.$$

The basis $V_*^{(k_*)}$ spans the dominant axes of variation across the current scenes. We then project both feature matrices onto the teacher principal subspace. The projections are $Z_* = \bar{F}_* V_*^{(k_*)}$ and $\hat{Z}_* = \bar{\hat{F}}_* V_*^{(k_*)} \in \mathbb{R}^{B \times k_*}$.

After mean-centering and unit-normalizing the projected features to remove batch offset and per-scene magnitude, we compare cosine similarities among the B scenes:

$$S_* = \tilde{Z}_* \, \tilde{Z}_*^\top, \quad \hat{S}_* = \tilde{\hat{Z}}_* \, \tilde{\hat{Z}}_*^\top \in \mathbb{R}^{B \times B}, \qquad \tilde{Z}_{*,i} = \frac{Z_{*,i}}{\|Z_{*,i}\|}, \qquad \tilde{\hat{Z}}_{*,i} = \frac{\hat{Z}_{*,i}}{\|\hat{Z}_{*,i}\|}.$$

Here, $\hat{S}_{*,ij}$ encodes the angle between scenes i and j in the teacher principal subspace. This angle indicates which scenes the teacher treats as alike. The student learns to replicate this geometry by minimizing the Frobenius norm of the two matrices' difference:

$$\mathcal{L}_{\text{struct}}^{(*)} \;=\; \frac{1}{B^2} \left\| S_* - \hat{S}_* \right\|_F^2, \qquad \mathcal{L}_{\text{struct}} = \mathcal{L}_{\text{struct}}^{(r)} + \mathcal{L}_{\text{struct}}^{(p)}.$$

We recompute the basis $V_*^{(k_*)}$ and target matrix $\hat{S}_*$ for each batch under a stop-gradient. This procedure makes the student match pairwise scene relationships only within the teacher principal subspace. It leaves both the absolute coordinate frame and the orthogonal noise complement free. We use $k_p = 13$ and $k_r = 9$, the 80% cumulative-energy cutoffs of the partner and road teacher spectra (§4.3).

Total loss. The alignment phase optimizes: $\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{struct}}$. We set $\lambda = 0.5$; performance was largely insensitive to λ in our sweeps.

4 Experiments

We test one claim: a vision policy distilled from a self-play teacher can drive competitively in closed loop. Neither training stage touches a logged human trajectory. The sections below build the evidence in three steps. First, we define the setup and the two scorers (§4.1), then compare the student and teacher against end-to-end baselines on HUGSIM (§4.2); the teacher sets the reference level, and the student is the main result. Second, we examine the alignment mechanism (§4.3): the teacher's features are sharply low-rank, which supports the low-rank alignment target of §3.2, and a privileged probe attributes the student–teacher gap mostly to perception. Third, a training-paradigm ablation (§4.3) shows that no other supervision source produces a stronger student. The backbone ablation appears in Appendix C.

4.1 Setup

We evaluate on all 436 scenarios of HUGSIM (Zhou et al., 2024), a photorealistic 3D Gaussian splatting closed-loop benchmark, spanning its four dataset splits (nuScenes 88, PandaSet 127, KITTI-360 113, Waymo 108), each divided into Easy, Medium, Hard, and Extreme tiers. We report two scores throughout: the closed-loop (CL) HD-Score, which scores the realized rollout, and HUGSIM's native scorer, which scores a planned trajectory. Our single-step controller replans at each step, emitting one immediate jerk action that we Euler-integrate into HUGSIM's bicycle-model control signals in causal closed form, without a second dynamics model (Appendix A.5 specifies the conversion). The closed-loop scorer executes only this first action per step and scores the resulting rollout. The native scorer, in contrast, needs a full trajectory that our controller does not publish, so we give it the *decel* stand-in plan: the realized first step followed by a comfort-bounded stop. Appendix B defines both scorers in detail; Appendix D reports per-split, per-tier results for the teacher, the student, and the re-run baselines. Beyond the nuPlan alignment corpus, we sample 300K paired (image, scene-state) frames from the HUGSIM reconstructions of each dataset split, for domain adaptation, following the real-world-simulation practice of SimScale (Tian et al., 2026). These synthetic frames cover only 10% of the total route length across all evaluation scenarios.

Table 1: **Our aligned student outperforms published end-to-end baselines on HUGSIM.** We report closed-loop HD-Score and HUGSIM's own planned-trajectory scorer (*Native*) per difficulty tier (§4.1), aggregated over all four dataset splits (436 scenarios, weighted by per-tier scenario counts; per-split tables appear in Appendix D). We re-run the released checkpoints of UniAD, VAD, LTF, and ECO on the same scenes and score their own published plans. Single-step policies, our student and the teacher alike, publish no trajectory, so their native scores use the *decel* stand-in plan defined in §4.1. The *italicized* teacher is a reference, not a comparison entry: it drives from privileged vectorized scene state, reads maps that we generate for the reconstruction-derived splits (Appendix D), and is excluded from best-per-column counts. Higher is better; **bold** marks the best comparison entry per column.

	Closed-loop HD-Score					Native HD-Score				
Method	Easy	Med.	Hard	Ext.	All	Easy	Med.	Hard	Ext.	All
UniAD (Hu et al., 2023)	0.441	0.255	0.225	0.093	0.244	0.437	0.259	0.230	0.100	0.247
VAD (Jiang et al., 2023)	0.283	0.210	0.178	0.087	0.187	0.272	0.204	0.173	0.092	0.183
LTF (Chitta et al., 2023; Dauner et al., 2024)	0.527	0.301	0.194	0.078	0.266	0.529	0.309	0.202	0.089	0.273
ECO (Zhang et al., 2025)	0.726	0.527	0.242	**0.215**	0.427	0.722	0.416	0.143	0.131	0.344
Self-play teacher (ref.)	*0.833*	*0.742*	*0.476*	*0.158*	*0.562*	*0.817*	*0.729*	*0.442*	*0.136*	*0.542*
Ours (E2E student)	**0.804**	**0.724**	**0.463**	0.162	**0.548**	**0.792**	**0.709**	**0.435**	**0.137**	**0.528**

4.2 Closed-Loop Comparison with End-to-End Baselines

Main Results. Table 1 aggregates HD-Score over all 436 HUGSIM scenarios: nuScenes, PandaSet, KITTI-360, and Waymo. The self-play teacher reaches 0.562 closed-loop and 0.542 under the native scorer, despite never observing a human demonstration; its exported rollouts also score 0.632 on the WOSAC human-likeness benchmark, within 0.09 of the 0.722 logged-oracle reference and on par with the strongest learned sim agents. The aligned student retains roughly 97.5% of the teacher's closed-loop performance (0.548 closed-loop, 0.528 native) and still exceeds every published baseline under both scorers. The strongest baseline, ECO, trails the teacher by 0.135 closed-loop and 0.198 native; the imitation-trained LTF, UniAD, and VAD reach only 0.266, 0.244, and 0.187 closed-loop. The lead holds on every dataset split, not just the aggregate (Appendix D).

Comparison with concurrent self-play-to-pixel systems. Two concurrent self-play-to-pixel systems report results on HUGSIM: GigaPixel (Rowe et al., 2026) trains in a dedicated bounding-box renderer and reaches a native HD-Score of 0.385; DriveZero (He et al., 2026) regresses teacher rollouts collected on real log-initialized worlds and reaches 0.466. Both were evaluated on the pre-challenge 345-scenario protocol of Kirby et al. (2026), which the current 436-scenario release later superseded; their checkpoints are not public, so we re-evaluate our system on the same 345 scenarios with HUGSIM's native scorer (Figure 2). The teacher reaches 0.527 and the student 0.518, exceeding both systems. The training loops differ as well: GigaPixel keeps a renderer in its loop, and DriveZero keeps per-scene teacher rollouts, whereas our alignment consumes only the teacher's per-state action distribution on paired frames, with no rendering or rollouts anywhere in training.

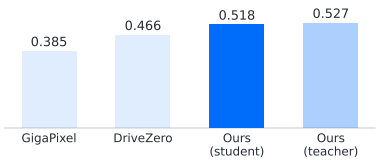


Figure 2: **Native HD-Score on the shared 345-scenario protocol.** Our student exceeds both concurrent self-play-to-pixel systems; the teacher upper-bounds the student.

Extreme-Tier Failure Mode. The student's weakest tier is Extreme (0.162 closed-loop, 0.137 native), where even the teacher reference manages only 0.158 and 0.136. The tier rewards the wrong behavior. Its adversarial scenarios are partly implausible (Figure 3): the other agents collide with the ego regardless of what the ego does, so a policy that just accelerates and drives straight, never reacting, already scores 0.227 closed-loop, above every reactive policy. The student, like the teacher it is aligned to, instead waits for the oncoming vehicle as a human driver would and sacrifices route completion. The tier thus punishes exactly the reactive behavior we want.



Figure 3: **Visually implausible extreme-tier scenarios.** Some HUGSIM Extreme scenarios begin from implausible configurations: other views are occluded, and vehicles appear to pass through one another. Configurations like these trigger the student's conservative yielding, which sacrifices route completion and pulls the Extreme scores of Table 1 down for every method.

4.3 INSIGHTS

The teacher's features are low-rank. Figure 4 shows where the teacher's structure lives. The map spectrum concentrates 80% of its energy in the first 9 directions, the partner spectrum in the first 13 of 64. The energy of direction i is its squared singular value σ_i^2, so the cumulative fraction $\sum_{j\le i}\sigma_j^2 / \sum_j \sigma_j^2$ is the variance explained by the first i directions. The structural loss therefore matches pairwise similarities only within these dominant subspaces, and the cutoffs are read from the spectrum rather than tuned. We expect two failure modes when the truncation moves either way. A rank set too low discards directions that carry real structure, so the loss stops constraining the student where constraint matters. A rank set too high pulls in the weakly structured tail, whose variation the image cannot explain, so the encoder is dragged toward a target it cannot identify. The ablation in Table 2 bears both expectations out: performance peaks at the spectrum-predicted cutoff and drops on either side, most at full coordinates.

Structural target	CL HD-Score ↑
Lower rank (k_p=7, k_r=5)	0.517
Low-rank (k_p=13, k_r=9, headline)	**0.548**
Higher rank (k_p=26, k_r=18)	0.533
Full coordinates	0.456

Table 2: **Rank ablation.** Closed-loop HD-Score on all 436 HUGSIM scenarios.

The losses are nearly lossless. The two losses are the only channel through which the teacher reaches the student. If they failed to express some of the teacher's competence, no encoder, however perfect, could recover the missing part. We therefore first quantify the cost of the objective itself. A *privileged probe* provides the measurement. The probe replaces the vision encoders with freshly initialized encoders of the teacher's own architecture, the two DeepSets of §3.2, which consume the privileged vector observation o_t instead of images. We train it with the same losses on the same paired frames, and the frozen head and the teacher's structural definition remain untouched. The probe therefore differs from the teacher in one variable only: its encoders are trained, not inherited. Two paired comparisons read the result. On the nuScenes split, the probe reaches 0.645 CL HD-Score against the teacher's 0.647: with identical vector inputs, the gap of 0.002 sits within evaluation noise, so the alignment objective costs nothing measurable. The camera student reaches 0.636 against the probe's 0.645: with an identical objective, the gap of 0.009 is the cost of perception. That perception gap is the asymmetric-observation problem LEAD (Nguyen et al., 2026) studies: the teacher decides from vectorized state that can include agents the camera cannot see.

The training paradigm sets the score. The first block of Table 3 measures how the choice of training paradigm shapes the camera student, evaluated closed-loop on the nuScenes split: each row holds the student architecture, the 1.83M nuPlan paired frames without the HUGSIM domain-adaptation frames, the perception labels, and the compute budget fixed, and varies only the supervision source. The cascade row recovers the teacher's vector inputs from the image with a SparseDrive perception module (Sun et al., 2024) that we finetune under the same compute budget, and feeds the recovered inputs to the frozen head. The cascade lands at 0.361. Our self-play alignment reaches 0.574. nuPlan-action imitation, trained on the same paired frames, scores 0.206: the training paradigm alone moves the student from 0.206 to 0.574. The training paradigm matters.

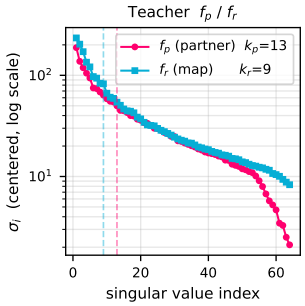


Figure 4: **Singular value spectra of the teacher features.** Centered SVD of teacher f_p (partner) and f_r (map) over the alignment manifest; dashed lines mark the 80% cumulative-energy index, the truncations used in §3.2.

Training paradigm	CL HD-Score
Imitation ($\mathcal{L}_{\text{act}}$ on nuPlan)	0.206
Modular cascade (SparseDrive)	0.361
Self-play pretraining + alignment (ours)	**0.574**
Action-only distillation	0.434
Structural-only distillation	0.373

Table 3: **Paradigm ablation under matched data and compute.** Closed-loop HD-Score on the nuScenes split. Each row uses the same architecture, the 1.83M nuPlan paired frames, perception labels, and a 1,616 A100-hour compute budget (1,536 self-play + 80 alignment) to match our setup exactly. The second block holds out one alignment loss term at a time, trained on the same data and compute budget.

Both losses are necessary. The second block of Table 3 holds out one alignment loss at a time. Action-only distillation reaches 0.434, structural-only 0.373, and both fall well short of the combined 0.574: each loss secures its own part of the teacher, the action distribution or the scene relations, and neither part suffices alone.

5 Conclusion and Future Work

Our photorealistic closed-loop results support a simple two-stage recipe: learn driving behavior through multi-agent self-play, then align a vision student to it. We train a planning head in a vectorized simulator, freeze it as the teacher, and train a vision frontend that produces the teacher's scene features from paired (image, scene-state) frames. Neither stage trains against logged ego trajectories: the teacher's behavior comes from self-play, and the alignment targets come from the frozen head the student reuses.

In the tested scenarios, the teacher outperforms imitation-trained pipelines, and the aligned vision policy retains most of that competence. Imitation can only replay maneuvers that a fleet log recorded, so its coverage of rare events is capped by what happened to occur; self-play generates collisions, recoveries, and negotiations on demand. We still use what the fleet log sees, but no longer what it does. Self-play is therefore a stronger alternative to imitation-based pretraining for learning driving behavior, while imitation remains useful for visual cues unavailable in bounding-box state, such as brake lights and hand signals.

Two extensions follow directly from this design. The teacher may base decisions on occluded agents that the camera cannot observe; paired-frame alignment does not remove this gap, and we have not isolated its contribution to the student's closed-loop performance. LEAD (Nguyen et al., 2026) studies the asymmetry directly, Pictura (Yin et al., 2026) sidesteps it by learning to drive from the rendered image itself, and belief-aware privileged distillation (Haklidir, 2026) adapts the teacher's guidance to what the student can observe. It is natural to extend the self-play training and distillation to these methods. Alignment also draws from nuPlan logs and reconstructed scenes, so it rarely encounters severe weather, unusual times of day, or rare scenario geometries. Controllable scene generation offers a way out: video world models such as DriveDreamer (Wang et al., 2024) generate the image side of our pairs, Scenario Dreamer (Rowe et al., 2025) produces the scene-state side as vectorized latents, and InfiniCube (Lu et al., 2025) and SCube (Huang et al., 2024) synthesize street scenes beyond the recorded weather and time of day. Pairing such generators with the self-play teacher would let both sides of the alignment come from procedurally controlled worlds.

AI Use Statement

We used generative AI tools to assist with code development and manuscript editing. We reviewed all AI-assisted material, verified the experimental code, and validated the reported results. We take responsibility for the final text, claims, code, and artifacts.

REFERENCES

Ehsan Ahmadi, Hunter Schofield, Behzad Khamidehi, Fazel Arasteh, Jinjun Shan, Lili Mou, Kasra Rezaee, and Dongfeng Bai. RLFTSim: Realistic and controllable multi-agent traffic simulation via reinforcement learning fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

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)*, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei-Jer Chang, Akshay Rangesh, Kevin Joseph, Matthew Strong, Masayoshi Tomizuka, Yihan Hu, and Wei Zhan. SPACeR: Self-play anchoring with centralized reference models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2019.

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.

Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Daphne Cornelisse, Spencer Cheng, Pragnay Mandavilli, Julian Hunt, Kevin Joseph, Waël Doulazmi, Valentin Charraut, Aditya Gupta, Joseph Suárez, and Eugene Vinitsky. PufferDrive: A fast and friendly driving simulator for training and evaluating RL agents. GitHub repository, 2025. URL `https://github.com/Emerge-Lab/PufferDrive`.

Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Krähenbühl, and Vladlen Koltun. Robust autonomy emerges from self-play. *arXiv preprint arXiv:2502.03349*, 2025.

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

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. RAP: 3d rasterization augmented end-to-end planning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, Ying Zhang, Wenyu Liu, Qian Zhang, and Xinggang Wang. RAD: Training an end-to-end driving policy via large-scale 3DGS-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

Yihong Guo, Dongqiangzi Ye, Sijia Chen, Anqi Liu, and Xianming Liu. CorrectionPlanner: Self-correction planner with reinforcement learning in autonomous driving. *arXiv preprint arXiv:2603.15771*, 2026.

Mehmet Haklidir. When does adaptive guidance help? Belief-Aware privileged distillation for autonomous driving under partial observability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2026.

Hao He, Chengcheng Hu, Zirun Su, Heng Zhang, Haisong Liu, Jinke Li, Haochen Tian, Zhenwei Shen, Hongyang Li, Zhichao Li, Yunchen Yang, Bochao Huang, Siyu Zhang, Kuangye Chen, Xiongjie Zhang, Wentao Dai, Hengchen Dai, Siyuan Liu, Zehao Huang, and Naiyan Wang. Drivezero: End-to-end driving beyond human demonstrations. *arXiv preprint arXiv:2609.06055*, 2026.

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

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)*, 2023.

Jiahui Huang, Zan Gojcic, Miao Wang, Francis Williams, Wenzheng Wu, Stefan Schneiders, Nan Zhou, Anurag Karpur, Shiju Jing, Lin Wang, Sharon X. Cui, Sanja Fidler, and Kaichang Ma. SCube: Instant large-scale scene reconstruction using VoxSplats. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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)*, 2023.

Peter Karkus, Maximilian Igl, Yuxiao Chen, Kashyap Chitta, Jef Packer, Bertrand Douillard, Thomas Tian, Alexander Naumann, Guillermo Garcia-Cobo, Shuhan Tan, Alperen Degirmenci, Alexander Popov, Nikolai Smolyanskiy, Urs Muller, Boris Ivanovic, and Marco Pavone. Beyond behavior cloning in autonomous driving: a survey of closed-loop training techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Ani Kelkar, Kersten Heineke, Martin Kellner, and Philipp Kampshoff. Getting on board with shared autonomous vehicles. McKinsey Center for Future Mobility, January 2025. URL `https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/getting-on-board-with-shared-autonomous-mobility`.

Eric Kirby, Alexandre Boulch, Yanis Xu, Yuhang Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. *arXiv preprint arXiv:2601.05083*, 2026.

Fabian Konstantinidis, Moritz Sackmann, Ulrich Hofmann, and Christoph Stiller. Toward efficient and robust behavior models for multi-agent driving simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Tianyu Li, Li Chen, Caojun Wang, Haochen Liu, Kashyap Chitta, Zhenjie Yang, Yuhang Lu, Naisheng Ye, Yihang Qiu, Yufei Wang, Luoxi Zou, Jiaxin Peng, Jin Pan, Zhaoyu Su, Andrei Bursuc, Shengbo Eben Li, Andreas Geiger, Peng Su, and Hongyang Li. World engine: Towards the era of post-training for autonomous driving. *arXiv preprint arXiv:2606.19836*, 2026.

Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, Francis Williams, and Jiahui Huang. InfiniCube: Unbounded and controllable dynamic 3D driving scene generation with world-guided video models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

National Highway Traffic Safety Administration. Overview of motor vehicle traffic crashes in 2023. Technical Report DOT HS 813 705, US Department of Transportation, 2025.

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.

Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario Dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Luke Rowe, Roger Girgis, Rodrigue de Schaetzen, Daphne Cornelisse, Alaap Grandhi, Felix Heide, Eugene Vinitsky, Christopher Pal, and Liam Paull. Scaling self-play for end-to-end driving. *arXiv preprint arXiv:2606.19641*, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hyunki Seong, Jeong-Kyun Lee, Heesoo Myeong, Yongho Shin, Hyun-Mook Cho, Duck Hoon Kim, Pranav Desai, and Monu Surana. Post-training and test-time scaling of generative agent behavior models for interactive autonomous driving. *arXiv preprint arXiv:2512.13262*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Wang, Kai Chen, and Xiaogang Jiang. SimScale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, Jiwen Lu, Chi Zhang, et al. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zhouchonghao Wu, Akshay Rangesh, Weixin Li, Wei-Jer Chang, Zachary Lee, Tim Wang, and Wei Zhan. TerraZero: Procedural driving simulation for zero-demonstration self-play at scale. *arXiv preprint arXiv:2607.13028*, 2026.

Yuan Yin, Elias Ramzi, Marc Lafon, Valentin Charraut, Victor Bares, Yihong Xu, Éloi Zablocki, Alexandre Boulch, Thibault Buhet, Andrei Bursuc, and Matthieu Cord. Pictura: Perspective-view self-play at scale for driving. *arXiv preprint arXiv:2607.26005*, 2026.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Brayden Zhang, Mahsa Golchoubian, Igor Gilitschenski, Boris Ivanovic, and Kashyap Chitta. End-point constrained trajectory optimization for driving foundation models. In *ICCV RealADSim Workshop*, 2025.

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)*, 2021.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv preprint arXiv:2412.01718*, 2024.