MIRROR LEARNING: LEARNING EGOCENTRIC POLICIES FROM THIRD-PERSON OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate imitation learning through the lens of third-person observation and propose a method for mirror learning: acquiring actionable policies from passive observation. While behavior cloning (BC) excels under dense, well-aligned first-person data, it fundamentally fails to leverage the rich observational signals arising from third-person demonstrations that humans and animals routinely exploit. We introduce a method that composes (i) a learned perspective transformation that places learners in demonstrators' shoes using a fine-tuned video diffusion model and (ii) an inverse dynamics model that infers action trajectories in the learners' control space. Once trained, this enables the synthesis of mirror data, pseudo first-person expert data generated from third-person observations of demonstrator behavior. Empirically, we show that mirror data alone can train effective policies, and that augmenting first-person BC training with mirror data further improves downstream policy performance. Our results suggest that modern generative world models implicitly encode sufficient structure to enable a scalable alternative to teleoperation-heavy data collection.

1 INTRODUCTION

Behavior cloning (BC) (Pomerleau, 1988; Bojarski et al., 2016) is supervised learning of policies from first-person observation-action trajectories. It is conceptually straightforward and performs well when demonstrations are dense, diverse, low-noise, and near-optimal with sufficient state coverage. Recent industrial successes, particularly in autonomous driving (Bojarski et al., 2016; Gao et al., 2025), have reinforced BC as a dominant paradigm, further amplified by increasing computational scale (Brown et al., 2020). Why behavior cloning? First, supervised learning is robust and scalable (LeCun et al., 2015). Second, reinforcement learning (RL), the principal alternative, suffers from sample inefficiency (Mnih et al., 2015) and requires reward specification, which can be ill-posed in real-world settings (Ng et al., 1999). Moreover, RL training is inherently first-person and experiential, raising safety and feasibility concerns in physical systems (Garcıa & Fernández, 2015).

In contrast, humans and animals learn from observing others. Extensive evidence from cognitive science and neuroscience shows that demonstration accelerates skill acquisition (Bandura & Walters, 1977; Heyes, 2011). Observational learning provides strong priors, particularly in tasks with high failure costs, such as locomotion or tool use (Winstein, 1991). However, this creates a fundamental challenge: how can agents learn from demonstrations when only their own embodiment is available for interaction? We refer to this broader goal as mirror learning. Neuroscience offers partial insights into this question through the discovery of mirror neurons, which respond during both action execution and observation (Rizzolatti et al., 1996; Iacoboni et al., 1999). These findings suggest a shared representation between perception and action, though their computational role remains debated (Hickok, 2009). Related phenomena, such as vicarious responses to pain and touch, further support the existence of cross-agent representational alignment (Keysers et al., 2010).

In machine learning, practical solutions for obtaining demonstrations for BC have focused on scaling first-person data collection via teleoperation or simulation (Levine et al., 2016). However, teleoperation is often infeasible for complex embodiments or tasks. Prior work on third-person imitation learning attempts to address this gap by learning domain-invariant representations (Sermanet et al., 2018) or performing viewpoint transformations (Liu et al., 2018). These approaches often rely on known camera geometry or constrained environments. In principle, combining perspective trans-

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.

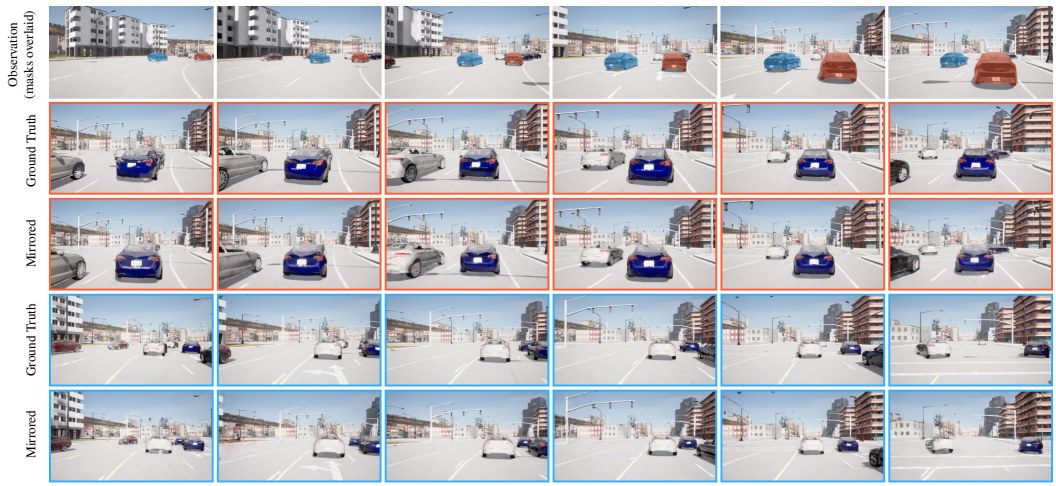


Figure 1: A demonstration of the mirror video model's cross-agent perspective transformation on a driving video. Top row: learner egocentric view with two demonstrator attention masks overlaid in orange and blue. For each demonstrator, we show the ground-truth egocentric video and one sample from our mirror video model. These mirror video model results demonstrate target-specific egocentric generation and temporal consistency that leverages context from throughout the learner video. Specifically the tan building on the right is not visible to the learner until the fifth shown frame, yet the model begins rendering it in the first mirrored frame. Car types, colors, light states, and lane markings are faithfully, though imperfectly, rendered. We show a single sample here because the learner is following both demonstrators and there is high overlap in their viewing frustums over time and therefore little variability between samples. Examples with greater sample variability arising from the learner and demonstrator crossing or opposing each other in motion appear in App. A.5.2.

formation with inverse dynamics models (IDMs) allows recovery of action labels from observation (Pathak et al., 2017). IDMs have been applied to infer missing actions in first-person data (Agrawal et al., 2016; Baker et al., 2022), but their application to cross-perspective learning remains limited.

Recent advances in generative video modeling suggest a new direction. Video diffusion and world models learn rich latent representations of physical environments (Ho et al., 2022; Brooks et al., 2024). These models enable planning (Hafner et al., 2019), inpainting (Lugmayr et al., 2022), and representation learning (Vélez et al., 2025), indicating strong implicit world modeling capabilities. Motivated by this, we develop a novel approach to adapting pretrained video diffusion to perform cross-agent perspective transformation, resulting in what we call a *mirror video model*.

Moreover, our mirror video model enables our method for mirror learning, which uses egocentric observations and actions inferred from third-person demonstrations for behavioral cloning. The mirror video model generates a demonstrator's egocentric video from the learner's observations, and the inverse dynamics model infers corresponding learner-executable actions. These predicted observation–action trajectories form mirror data. We show that learning a policy from mirror data alone is possible, and that combining mirror data with first-person experience can further improve policy performance. These results show that our method turns observations of other agents into a scalable source of policy supervision, offering an alternative to collecting first-person demonstrations.

2 Background

We consider multi-agent settings with K agents, indexed by $i \in \{0, \dots, K-1\}$, interacting in a shared environment with latent state $s_t \in \mathcal{S}$, shared action space $\mathcal{A}$, and shared observation space $\mathcal{O}$. At each timestep t, agent i takes action a_t^i according to its video-conditioned policy $\pi^i(a_t^i \mid o_{t-h:t}^i)$, where $o_t^i \sim p(\cdot \mid s_t, i) \in \mathcal{O}$ is agent i's partial observation of the latent state s_t, which we define as a single video frame from the perspective of agent i. Following prior art (Van Hasselt et al., 2016), we assume policies are conditioned on $o_{t-h:t}^i := (o_{t-h}^i, \dots, o_t^i)$, a stack of the $h+1$ most recent observations up to and including t.

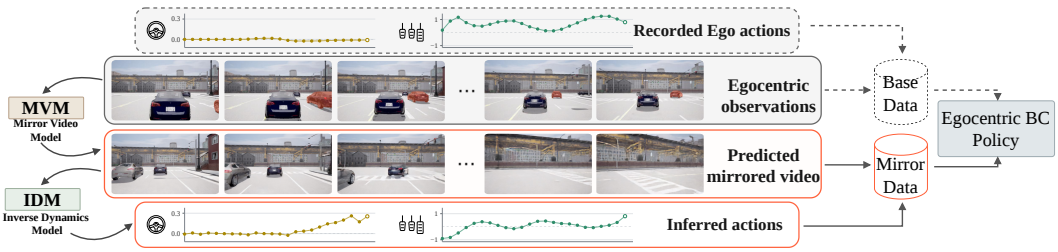


Figure 2: **Our mirror learning approach.** Our mirror video model (MVM) converts observations of a demonstrator (highlighted in red) into a predicted ego-view video in the learner's observation space, effectively putting the learner "in the shoes" of the demonstrator. An inverse dynamics model (IDM) infers learner-executable actions from this mirrored video, creating observation–action pairs for behavioral cloning. This mirror data provides supervision for behavioral cloning, optionally alongside learner's recorded observations and actions (base data), as indicated by dashed arrows. MVM and IDM remain frozen during BC.

We focus on the problem of policy learning from previously collected trajectories, without access to a reward function or additional environment interaction during training. We designate agent 0 as the learner and consider trajectories recorded from its perspective under the policy $\pi_{\text{base}} \coloneqq \pi^0$. For brevity, we omit the superscript for this agent, defining recorded observations $o_t \coloneqq o_t^0$ and actions $a_t \coloneqq a_t^0$. We assume access to a dataset of N trajectories, each consisting of $T+1$ observations and T actions, $\mathcal{D}_{\text{base}} = \{\tau_n\}_{n=1}^N$, $\tau_n = (o_{n,0}, a_{n,0}, \ldots, o_{n,T})$. In particular, we focus on policy learning through behavioral cloning, training a policy π_θ to approximate π_{base} by minimizing the behavioral cloning objective $\mathcal{L}_{\text{BC}}(\theta; \mathcal{D}) = \frac{1}{(T-h)\cdot|\mathcal{D}|} \sum_{n=1}^{|\mathcal{D}|} \sum_{t=h}^{T-1} -\log \pi_\theta(a_{n,t} \mid o_{n,t-h:t})$ over $\mathcal{D} = \mathcal{D}_{\text{base}}$.

3 Methodology

Behavioral cloning requires a dataset of trajectories $\mathcal{D}_{\text{base}}$ collected by interacting with the environment under policy π_{base}. However, collecting this data may be time-intensive and expensive, limiting the scale of $\mathcal{D}_{\text{base}}$. We note that in the multi-agent setting described by Section 2, video observations of the learner may capture the behavior of another agent $j \neq 0$ acting in the environment. We refer to such an agent as a demonstrator, since the learner's video provides a third-person demonstration of its behavior. On their own, these third-person demonstrations are not immediately useful for behavioral cloning of an egocentric video-conditional policy, which requires egocentric observations paired with actions expressed in the learner's control space. However, if these quantities could be inferred from the observations of the learner, we hypothesize that the resulting data can be a valuable source of additional training demonstrations, either for augmenting $\mathcal{D}_{\text{base}}$ by assuming $\pi^j = \pi_{\text{base}}$, or as a standalone source to train the learner's policy without requiring additional data collection.

To test this hypothesis, our method for mirror learning uses egocentric observations and actions inferred from third-person demonstrations for behavioral cloning. We refer to these inferred egocentric observations and actions as mirror data. Generation of mirror data requires two components: a mirror video model (MVM) g_ϕ, which estimates egocentric video frames $\hat{o}^j_{0:T}$ from third-person observations $o_{0:T}$, and an inverse dynamics model (IDM) f_ψ, which estimates learner-executable actions $\hat{a}^j_{0:T-1}$ consistent with the mirrored observations $\hat{o}^j_{0:T}$. As neither component requires the action sequence of the learner, mirror data can be generated from observation sequences alone. We assume access to a dataset $\mathcal{D}_{\text{source}} = \{o_{m,0:T}\}_{m=1}^M$ of recorded video observations which may be derived from $\mathcal{D}_{\text{base}}$ or from a separate data collection process. For a video $o_{0:T} \in \mathcal{D}_{\text{source}}$ we can generate a mirror trajectory $\hat{\tau}^j$ for visible demonstrator agent j via

$$\hat{\tau}^j(o_{0:T}) = \left(\hat{o}^j_0, \hat{a}^j_0, \ldots, \hat{o}^j_T\right), \tag{1}$$

$$\hat{o}^j_{0:T} \sim g_\phi\left(\cdot \mid o_{0:T}, j\right), \tag{2}$$

$$\hat{a}^j_{0:T-1} \sim f_\psi\left(\cdot \mid \hat{o}^j_{0:T}\right). \tag{3}$$



Figure 3: Zero-shot mirror video model predictions on real-world data (MVM trained only on CARLA synthetic data w/o pedestrians). Ground truth extracted from data (Li et al., 2024) in which a "learner" May robotaxi follows and observes a "demonstrator" May robotaxi.

Applying Eqs. (1) to (3) to each video in $\mathcal{D}_{\text{source}}$ yields a mirrored dataset $\hat{\mathcal{D}}_{\text{mirror}} = \{\hat{\tau}^j(o_{0:T}) \mid o_{0:T} \in \mathcal{D}_{\text{source}}, j \in \mathcal{V}(o_{0:T})\}$ of the same form as $\mathcal{D}_{\text{base}}$ where $\mathcal{V}(o_{0:T}) \subseteq \{1, \ldots, K-1\}$ is the set of visible demonstrator agents. Using $\hat{\mathcal{D}}_{\text{mirror}}$ and $\mathcal{D}_{\text{base}}$, we define our policy-training objective

$$\mathcal{L}_{\text{policy}}\left(\theta; \mathcal{D}_{\text{base}}, \hat{\mathcal{D}}_{\text{mirror}}\right) = \lambda \mathcal{L}_{\text{BC}}\left(\theta; \mathcal{D}_{\text{base}}\right) + (1-\lambda)\, \mathcal{L}_{\text{BC}}\left(\theta; \hat{\mathcal{D}}_{\text{mirror}}\right), \tag{4}$$

where $\lambda \in [0, 1]$ is a weighting hyperparameter controlling the relative contribution of examples from $\mathcal{D}_{\text{base}}$ and $\hat{\mathcal{D}}_{\text{mirror}}$. We highlight three choices of λ. Setting $\lambda = N/(N + N_{\text{mirror}})$, where $N_{\text{mirror}} = |\hat{\mathcal{D}}_{\text{mirror}}|$, evenly weights samples from both datasets, whereas $\lambda = 0$ yields an objective which learns from $\hat{\mathcal{D}}_{\text{mirror}}$ exclusively, and $\lambda = 1$ recovers behavioral cloning on $\mathcal{D}_{\text{base}}$.

Mirror Video Model The central challenge of our method is generating egocentric observations $\hat{o}^j_{0:T}$ of agent j from the learner's observations $o_{0:T}$. We address this problem with a conditional video flow-matching model, without assuming knowledge of the demonstrator's relative camera pose or scene geometry. This avoids explicitly extrapolating latent geometry beyond the scene coverage directly observed by the learner, a common challenge for geometry-based novel-view synthesis methods (Fu et al., 2024; Shih et al., 2024).

We decompose the problem of generating mirrored observations into two subproblems. First, we identify demonstrator agents within $o_{0:T}$, and then generating their mirrored egocentric observations. For the first subproblem, we use SAM3 (Carion et al., 2025) with a text prompt to segment the relevant object class across $o_{0:T}$. We apply visibility and observation-length heuristics to obtain candidate tracks, then manually select the demonstrator using SAM3's temporally consistent object IDs to form the visible agent set $\mathcal{V}(o_{n,0:T})$. For each $j \in \mathcal{V}(o_{n,0:T})$, we produce binary instance segmentation mask videos $m^j_{0:T}$ which segments agent j. Our mirror video model (MVM) then learns $g_\phi(o^j_{0:T} \mid o_{0:T}, m^j_{0:T})$, the distribution of demonstrator j's ego-view video given the learner's video and the demonstrator mask. We note that this task is substantially more challenging than standard conditional video inpainting (Green et al., 2024; Li et al., 2025), due to limited spatial and temporal overlap between the recorded and demonstrator observations, and the required inference of camera pose, scene geometry, and unobserved scene content.

Training our MVM from scratch would require a large paired dataset of temporally aligned learner and demonstrator ego-view videos. Due to the difficulty of collecting such a dataset, we lessen the data requirement by instead fine-tuning our MVM from a pretrained video foundation model.

We select Cosmos-Predict2.5 (Cosmos)(Ali et al., 2025) as the basis of our MVM. As Cosmos is trained on the video completion distribution $p(o_{t+1:T} \mid o_{0:t}, c_{\text{text}})$, where c_{text} is the optional text prompt, we make the following changes to adapt it to our perspective transfer task. We encode the learner's ego-view video $o_{0:T}$ and demonstrator mask sequence $m^j_{0:T}$ with Cosmos' pretrained Wan2.1 VAE encoder (Wan et al., 2025), tiling the binary mask input over the three color channels.



Figure 4: Minecraft mirror video model samples. Demonstrator instance segmentation mask indicated by orange highlighting. For the learner and demonstrator both we show time-aligned frames from their ground-truth egocentric videos. The two samples are MVM predictions of the demonstrator view given the learner and instance segmentation mask videos. The learner cannot see its own "skin," so the MVM samples two different plausible skins. The learner can see that the demonstrator is holding an axe; this is reflected in all samples. The learner can see that the demonstrator is looking at and past the learner in a mountain biome; the MVM inpaints varied plausible mountain biome physical context. Many more examples appear in App. A.5.4

To form the MVM input, we temporally concatenate three encoded video streams: the mask sequence, the learner observation sequence, and the noise corrupted demonstrator observation sequence used for flow matching. Following the structure of Cosmos, we use a binary mask to differentiate the observed learner video and demonstrator mask from the noised demonstrator video which should be generated. We additionally modify the the 3D rotary positional embeddings (RoPE) (Su et al., 2024), duplicating the embeddings across the encoded videos to match their temporal alignment. We outline our MVM architecture in Fig. 9, and provide additional architectural and training details in App. A.4.

We fine-tune our MVM using the Cosmos' latent flow-matching training objective (Ali et al., 2025, Eq. (3)) over demonstrator observations. We optimize all layers of the denoising transformer, while freezing the text encoder and the spatial-temporal latent encoder and decoder. During inference, the MVM generates $\hat{o}^j_{0:T}$ conditioned on the learner's ego video and the demonstrator-identifying attention mask sequence, using Cosmos' default sampling algorithm.

Inverse Dynamics Model In addition to mirror video model outputs, generating a mirror dataset requires estimating corresponding learner-executable action labels. Recovering actions from egocentric observations with an inverse dynamics model (IDM) (Pathak et al., 2017; Torabi et al., 2018; Baker et al., 2022) is a well studied problem. We train the IDM on the learner's ego-view videos and corresponding actions, modeling $f_\psi(a_{0:T-1} \mid o_{0:T})$. As IDM implementation depends on the action space $\mathcal{A}$ of the chosen environment, we specify IDM model details for each of our experimental settings in Apps. A.6.1 and A.6.3.

4 Experiments

We evaluate our method through three main questions:

1. Can mirror data provide useful additional supervision for behavioral cloning?
2. Can our MVM, and an IDM, enable downstream policy learning from mirror data alone?
3. How well do the MVM and IDM components perform, and can an MVM fine-tuned only on simulated data transfer zero-shot to real-world video?

We address the first two questions in autonomous driving by training policies with mirror data, either alongside first-person demonstrations or alone. We also address the first question in Minecraft by training policies with additional mirror data. For the third question, we evaluate the actions predicted by the IDM and show that fine-tuned Cosmos models can perform the mirror video task in CARLA

and Minecraft. We test the MVM trained in CARLA on a held-out town and use the same model to qualitatively evaluate zero-shot transfer on the real-world robotaxi (May) dataset (Li et al., 2024), where occasionally one robotaxi sees the other giving us ground-truth demonstrator views.

4.1 Experimental Setup

We evaluate our method in autonomous driving and Minecraft. Autonomous driving offers readily available policies (Wu et al., 2022; Bartoccioni et al., 2025; Renz et al., 2025) and a two-dimensional control space (steering and acceleration), making inverse dynamics learning practical within our computational budget. Large datasets containing paired learner–demonstrator videos with action labels are not readily available, so we collect approximately 22k paired videos in CARLA (Dosovitskiy et al., 2017) using a shared data-driven object level policy (Lioutas et al., 2025) for all agents. Demonstrator masks come from CARLA's instance segmentation sensor. We fine-tune the MVM on 14,500 segments and extract 27.9k ground-truth (GT) policy-training windows from the learner's observations and actions in these segments for the 83% GT conditions in Table 1. We partition the data into training, validation, and test splits that support intra- and inter-town[1] evaluation (details in App. A.2.1). Minecraft provides a perceptually rich setting with more varied body and head poses than driving. We extract approximately 7.2k paired clips from real human multiplayer Minecraft gameplay (He et al., 2025), spanning years of continuous environment development and diverse user-built structures. We obtain demonstrator masks using SAM 3 (Carion et al., 2025) and use 6k clips for MVM training. The policy baseline uses 5,798 GT windows from the learner's data in the MVM training split. An additional 1,904 GT windows provide observations for mirror-data generation (details in App. A.3). In both domains, we train IDMs on learner observations and actions from the MVM training split.

Policy training and evaluation. To evaluate the value of mirror data for policy learning in the driving domain, we train VaVAM (Bartoccioni et al., 2025) policies, conditioned on six history frames at 2 Hz and a driving command, to predict three seconds of future actions. Our driving IDM uses a VaViM backbone with a head that predicts relative displacements (details in App. A.6.1). We evaluate policies open-loop using minimum average predicted displacement error over five samples, minADE_5. The data used to train the MVM and IDM, the third-person demonstrations used to generate mirror data, and the policy evaluation data come from separate collections (details in App. A.6). For the experiments in Table 1 and Fig. 5a, we generate mirror data from third-person demonstrations collected in the same physical environments used to train the MVM and IDM, but from disjoint traffic scenarios. Same-town policy evaluation uses held-out scenarios with different weather, time of day, vehicle placement, and traffic configurations. For Minecraft, we adapt a pretrained VPT IDM (Baker et al., 2022) and fine-tune a pretrained VPT policy, evaluating action prediction on held-out gameplay sessions using action negative log-likelihood (NLL). Training and evaluation details are provided in App. A.6.3.

4.2 Mirror Data Evaluation

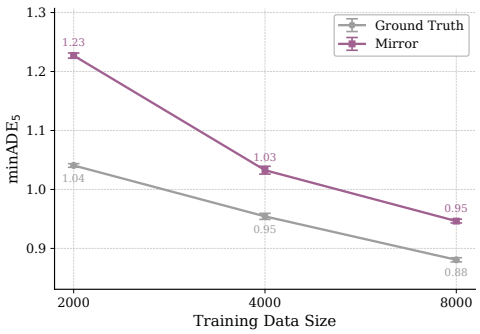
Learning from real and mirror data. We investigate the impact of additional mirror data on policy learning from recorded demonstrations, denoted GT in the tables. We optimize policies under Eq. (4) with $\lambda = N/(N + N_{\text{mirror}})$ across five training conditions, varying the amounts of GT and mirror data. The GT-plus-mirror conditions match the full-GT reference in total sample count. From Table 1, we find that adding mirror data improves policy performance over using GT data alone. Both GT-plus-mirror conditions achieve comparable performance to the full-GT reference in the training towns, indicating that our mirror data is a beneficial approach for expanding data scale. For completeness, we report a closed-loop evaluation showing higher mean driving scores and fewer mean collisions with additional mirror data (Table 3). In the Minecraft setting, we also see improved action prediction when policies are trained with additional mirror data.

A promising setting for augmenting recorded trajectories with mirror data is transfer learning in new geographic regions. A small fleet records limited first-person data in a new location, while other vehicles provide additional third-person demonstrations.

[1] CARLA towns are unreachable from each other and generally do not look the same.

Table 1: Open-loop policy evaluation (minADE_5) results for various ground-truth plus mirror-generated dataset sizes. GT and mirror amounts are expressed as percentages of the full reference dataset of 33.6k samples. Within each paired group, we compare GT-only training with the same GT data plus additional mirror data. The policy was evaluated both in the same towns in which the training data was collected and zero-shot in a new, held-out town. Adding mirror data improves performance over the corresponding GT-only baseline both in the training environments and zero-shot in the held-out environment.

Setup	GT (%)	Mirror (%)	Same towns	Held-out town
Full GT reference	100	0	0.74±0.01	**0.73**±0.00
GT only	70	0	0.77±0.01	0.85±0.00
GT + mirror	70	30	0.75±0.01	0.76±0.00
GT only	83	0	0.76±0.01	0.80±0.05
GT + mirror	83	17	**0.73**±0.01	0.76±0.01



(a)

GT	Augmentation	Aug. ratio	minADE_5
Pretrained	–	–	0.80±0.05
1k	None	–	0.69±0.01
2k		–	0.64±0.00
1k	Style transfer	1×	0.68±0.00
1k		2×	0.70±0.01
1k	Mirror data	1×	0.68±0.00
1k		2×	0.65±0.00
2k		1×	0.63±0.01
2k		2×	**0.62**±**0.00**

(b)

Figure 5: *(a)* BC from mirror data alone suffices for policy learning. With more mirror data, it can match the performance of BC trained on recorded data. *(b)* Geographic expansion into a new town. Here we fine-tune the 83% GT-only policy from Table 1 with 1k or 2k new ground-truth samples and augment with style-transfer or mirror data (ratio relative to GT). Mirror data outperforms style transfer and improves with higher ratios. Lower minADE_5 is better; ± is std. over three seeds.

We evaluate the efficacy of mirror data augmentation in this setting against two baselines: no augmentation and augmentation via Cosmos-Transfer2.5-2B (Ali et al., 2025) style transfer (examples in App. A.6.4). Using the MVM and IDM from Table 1, we fine-tune the IDM on new-town GT data and generate mirror data from demonstrators in the new town while keeping the MVM fixed. In each case, we fine-tune the 83% GT policy from Table 1 on the resulting dataset. Figure 5b demonstrates the value of mirror data for transfer learning. Compared to no fine-tuning or fine-tuning on GT data alone, mirror data improves policy performance and remains effective at augmentation ratios up to twice the GT data size. Furthermore, mirror augmentation is more effective than style transfer, which changes scene appearance without introducing new trajectories into the fine-tuning corpus.

Learning from mirror data alone. We next evaluate the viability of policy learning from mirror data alone using our trained MVM and IDM. Figure 5a compares the open-loop performance of policies trained exclusively on mirror data ($\lambda = 0$) to those trained exclusively on recorded trajectories ($\lambda = 1$) across three data scales. While recorded data produces better policies at matched dataset sizes, the gap between the curves suggests that comparable performance can be achieved using approximately twice as much mirror data. These findings present a promising direction for learning exclusively from demonstration when collecting first-person demonstrations is expensive or difficult.

Table 2: Minecraft policy evaluation (action negative log-likelihood, NLL) results for various ground-truth plus mirror-generated dataset sizes. Policies are evaluated on held-out data. Adding mirror data improves action prediction over the corresponding GT-only baseline at both GT dataset sizes.

Training data	5.8k GT	7.7k GT	5.8k GT + 1.9k mirror	7.7k GT + 1.9k mirror
Action NLL ↓	1.10±0.01	1.02±0.01	1.05±0.01	**1.00**±0.01

4.3 Component Analysis and MVM Generalization

Held-out-town generalization. Figure 1 exemplifies MVM performance on unseen test locations. These experimental results demonstrate that Cosmos, even after fine-tuning, possesses rich enough features to solve the video mirroring task as defined, and that the instance segmentation mask is a sufficient signal for the model to identify and implicitly infer the pose of a demonstrator. Finally it demonstrates that the MVM has not memorized the training set, and is able to perform video mirroring even on unseen test town data. We provide additional examples in App. A.5.1.

Zero-shot real-world transfer. Figure 3 shows that our MVM is able to generate realistic data even when fine-tuned exclusively on synthetic data. The May dataset is entirely withheld and Cosmos has not been trained with any paired, real-world video recordings. Moreover, the synthetic CARLA dataset contained no pedestrians. The MVM nearly perfectly renders the environment from the demonstrator perspective, partially infers the presence and motion of pedestrians, and makes sensible inferences about occluded traffic presence and behavior. This zero-shot real-world performance suggests that training on synthetic data is not only sufficient to elicit mirror video model capabilities but that real world objects, behaviors, and more "known" to the pretrained video diffusion model may remain present in the MVM after fine-tuning. Access to paired real-world driving data would likely enable further gains, as fine-tuning the video mirroring model on even a small real dataset could reduce sim-to-real gaps arising from missing pedestrians, vehicle classes, and behaviors.

Inverse dynamics model. To help interpret the policy performance gap between mirror and GT data, we evaluate the IDM on held-out recorded demonstrator videos, obtaining an average displacement error (ADE) of 0.20 m against recorded trajectories. Beyond video-quality metrics (Table 5), we also assess how faithfully mirror videos preserve the demonstrated behavior. Applying the same IDM to mirror videos generated from the corresponding learner observations yields an ADE of 0.42 m against the same recorded trajectories. This evaluates the behavioral fidelity of the combined MVM–IDM pipeline. Implementation and evaluation details appear in App. A.6.1.

4.4 Minecraft Experiments

Minecraft agents perceive the world through time-varying body and head poses with many degrees of freedom. Fig. 4 shows that MVMs can nevertheless perform this more challenging mirroring task. Compared to CARLA, the samples exhibit much higher levels of variability reflecting the fact that the learner does not see or otherwise know its own appearance which varies across videos, and it does not have access to latent demonstrator state such as inventory contents, chat window status, or health. Encouraged by these MVM results, we evaluate whether mirror data provides useful additional supervision for policy learning in Minecraft. Table 2 shows that additional mirror data improves action prediction on held-out Minecraft data at both GT dataset sizes.

5 Related Work

Learning from demonstrations by observation must address two missing pieces, demonstrator actions and demonstrator egocentric observations. Prior work recovers missing actions with inverse dynamics models trained on the learner's own observations (Torabi et al., 2018; Agrawal et al., 2016), but this requires demonstrations to already lie in the learner's observation space. To handle viewpoint mismatch, other methods learn view-invariant representations or explicit cross-view translation (Sermanet et al., 2018; Stadie et al., 2017; Liu et al., 2018; Shang & Ryoo, 2021) mostly in constrained environments. Most relevant is Liu et al. (2018), who learn a context translation

model conditioned on a single learner-context frame to translate demonstrations into the learner's observation context. However, they use the learned intermediate translation features as perceptual rewards for reinforcement learning rather than as synthetic supervision for behavior cloning.

Prior autonomous driving work performs learner-frame transfer in structured intermediate representations rather than generating mirrored RGB video (Zhang & Ohn-Bar, 2021; Chen & Krähenbühl, 2022; Feng et al., 2025). Learning by Watching (Zhang & Ohn-Bar, 2021) uses occluded birdview rasterization, Learning from All Vehicles (Chen & Krähenbühl, 2022) fuses RGB, segmentation, and LiDAR into a shared 2D representation, and RAP (Feng et al., 2025) uses lightweight 3D rasterization with raster-to-real alignment. These methods expand training data through structured observation transfer, but do not produce realistic first-person RGB supervision for end-to-end driving policies.

Third-to-first video transfer synthesizes first-person videos from temporally synchronized third-person observations and is closely related to controllable video generation. Methods typically rely on strong geometric or multi-view supervision, either by decoupling structure transfer from appearance synthesis or by injecting explicit 3D priors (Liu et al., 2024; Mahdi et al., 2025; Kang et al., 2025). EgoX (Kang et al., 2025), for example, builds on pretrained video diffusion models but is not geometry-free. It requires target egocentric camera poses, point-cloud derived egocentric priors, and fixed exocentric camera poses at inference. WorldWander (Song et al., 2025) is also particularly relevant because it frames the problem as conditional video-to-video generation with a video diffusion model, but its EgoExo-8k dataset focuses on a single continuously tracked target that is typically centered in the frame. By contrast, our approach does not require explicit 3D scene reconstruction, allows both learner and demonstrator to move throughout the scene in both CARLA and Minecraft, and operates in a more challenging multi-agent setting in which multiple candidate demonstrators may be present. Unlike WorldWander, we also show that synthesized demonstrator ego-view videos can be paired with IDM-inferred actions to form mirror data for behavior cloning.

End-to-End Driving Policies. Our choice, VaVAM Bartoccioni et al. (2025), pretrains an autoregressive driving video model, VaViM, on large-scale unannotated driving videos then uses those representations in a lightweight imitation-learning stage. Alternatives policies such as SimLingo Renz et al. (2025) and Alpamayo-R1 Wang et al. (2025) may be considered in future work.

6 Conclusions

For the autonomous vehicle setting, we showed that (i) an effective driving policy can be learned using only mirror data generated by our learned MVM and IDM, (ii) augmenting existing ground-truth data with mirror data consistently improves open-loop policy evaluation (minADE5) both in training environments and in held-out geographic locations, and (iii) mirror data augmentation outperforms style transfer at matched augmentation ratios in the geographic expansion setting. In Minecraft, additional mirror data also improves held-out policy action prediction when added to the GT dataset.

These policy learning results build on our finding that pretrained video diffusion models possess sufficiently rich internal representations of physical scene structure to support mirror video synthesis, without requiring explicit camera parameters, pose estimation, or scene reconstruction. We demonstrated this capacity across two visually and structurally diverse domains (CARLA autonomous driving and multiplayer Minecraft), and showed zero-shot generalization from synthetic training data to real-world robotaxi footage.

These results suggest a path toward scalable policy acquisition in settings where teleoperation is costly, dangerous, or infeasible. Our method can reduce reliance on first-person policy-training data and leverage the rich observational signals that are already present in any multi-agent environment. Improvements in video world model capacity will further increase the fidelity and downstream value of synthesized mirror data. Training the MVM currently requires paired learner–demonstrator videos, which can be difficult to obtain. Our preliminary sim-to-real results suggest that transferring a trained MVM could reduce this burden. Our method also inherits limitations from behavior cloning, such as compounding errors.

Code, datasets, and model checkpoints will be released upon acceptance to facilitate reproducibility and to provide baseline metrics for future work on mirror video modeling.

7 AI USE STATEMENT

Generative AI tools were used to polish the presentation of this manuscript, including language editing, consistency checks, and figure formatting. LLMs were used also generated text prompts for the style-transfer augmentation baseline. The authors reviewed all AI-assisted changes and take responsibility for the final content of this work, including its text, claims, and figures.

REPRODUCIBILITY STATEMENT

Code, datasets, and model checkpoints will be released upon acceptance to facilitate reproducibility. Data collection, preprocessing, and splits are described in Apps. A.2.1 and A.3. Model training configurations and computational resources are provided in App. A.4. IDM implementation and evaluation details appear in App. A.6.1, while policy-training and evaluation protocols are detailed in Apps. A.6 and A.6.3. The style-transfer procedure and example prompts are provided in App. A.6.4.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Albert Bandura and Richard H Walters. *Social learning theory*, volume 1. Prentice-hall Englewood Cliffs, NJ, 1977.

Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loick Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, et al. Vavim and vavam: Autonomous driving through video generative modeling. *arXiv preprint arXiv:2502.15672*, 2025.

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3(1):3, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17222–17231, 2022.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 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. *arXiv preprint arXiv:2510.04333*, 2025.

Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *arXiv preprint arXiv:2109.03805*, 2021.

Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20796–20805, June 2024.

Haoxiang Gao, Zhongruo Wang, Yaqian Li, Kaiwen Long, Ming Yang, and Yiqing Shen. A survey for foundation models in autonomous driving. In *2025 6th International Conference on Computer Vision and Data Mining (ICCVDM)*, pp. 63–71. IEEE, 2025.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Dylan Green, William Harvey, Saeid Naderiparizi, Matthew Niedoba, Yunpeng Liu, Xiaoxuan Liang, Jonathan Lavington, Ke Zhang, Vasileios Lioutas, Setareh Dabiri, et al. Semantically consistent video inpainting with conditional diffusion models. *arXiv preprint arXiv:2405.00251*, 2024.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Yingchen He, Christian D Weilbach, Martyna E Wojciechowska, Yuxuan Zhang, and Frank Wood. Plaicraft: Large-scale time-aligned vision-speech-action dataset for embodied ai. *arXiv preprint arXiv:2505.12707*, 2025.

Cecilia Heyes. Automatic imitation. *Psychological bulletin*, 137(3):463, 2011.

Gregory Hickok. Eight problems for the mirror neuron theory of action understanding in monkeys and humans. *Journal of cognitive neuroscience*, 21(7):1229–1243, 2009.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Marco Iacoboni, Roger P Woods, Marcel Brass, Harold Bekkering, John C Mazziotta, and Giacomo Rizzolatti. Cortical mechanisms of human imitation. *science*, 286(5449):2526–2528, 1999.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Taewoong Kang, Kinam Kim, Dohyeon Kim, Minho Park, Junha Hyung, and Jaegul Choo. Egox: Egocentric video generation from a single exocentric video. *arXiv preprint arXiv:2512.08269*, 2025.

Christian Keysers, Jon H Kaas, and Valeria Gazzola. Somatosensation in social perception. *Nature Reviews Neuroscience*, 11(6):417–428, 2010.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025.

Yiming Li, Zhiheng Li, Nuo Chen, Moonjun Gong, Zonglin Lyu, Zehong Wang, Peili Jiang, and Chen Feng. Multiagent multitraversal multimodal self-driving: Open mars dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22041–22051, 2024.

Vasileios Lioutas, Adam Scibior, Matthew Niedoba, Berend Zwartsenberg, and Frank Wood. Controlitra: Controlling the behavior of a driving model. *arXiv preprint arXiv:2501.12408*, 2025.

Jia-Wei Liu, Weijia Mao, Zhongcong Xu, Jussi Keppo, and Mike Zheng Shou. Exocentric-to-egocentric video generation. *Advances in Neural Information Processing Systems*, 37:136149–136172, 2024.

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1118–1125. IEEE, 2018.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Mohammad Mahdi, Yuqian Fu, Nedko Savov, Jiancheng Pan, Danda Pani Paudel, and Luc Van Gool. Exo2egosyn: Unlocking foundation video generation models for exocentric-to-egocentric video synthesis. *arXiv preprint arXiv:2511.20186*, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

NVIDIA. Physicalai autonomous vehicles, 2025. URL `https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles`. Dataset card, created October 28, 2025; accessed April 8, 2026.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Giacomo Rizzolatti, Luciano Fadiga, Vittorio Gallese, and Leonardo Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive brain research*, 3(2):131–141, 1996.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Jinghuan Shang and Michael S Ryoo. Self-supervised disentangled representation learning for third-person imitation learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 214–221. IEEE, 2021.

Meng-Li Shih, Wei-Chiu Ma, Lorenzo Boyice, Aleksander Holynski, Forrester Cole, Brian Curless, and Janne Kontkanen. Extranerf: Visibility-aware view extrapolation of neural radiance fields with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20385–20395, 2024.

Quanjian Song, Yiren Song, Kelly Peng, Yuan Gao, and Mike Zheng Shou. Worldwander: Bridging egocentric and exocentric worlds in video generation. *arXiv preprint arXiv:2511.22098*, 2025.

Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third person imitation learning. In *International Conference on Learning Representations*, 2017.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Pedro Vélez, Luisa F Polanía, Yi Yang, Chuhan Zhang, Rishabh Kabra, Anurag Arnab, and Mehdi SM Sajjadi. From image to video: An empirical study of diffusion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16948–16958, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Carolee J Winstein. Knowledge of results and motor learning—implications for physical therapy. *Physical therapy*, 71(2):140–149, 1991.

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 *NeurIPS*, 2022.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025.

Jimuyang Zhang and Eshed Ohn-Bar. Learning by watching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12711–12721, 2021.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Berend Zwartsenberg, Adam Ścibior, Matthew Niedoba, Vasileios Lioutas, Yunpeng Liu, Justice Sefas, Setareh Dabiri, Jonathan Wilder Lavington, Trevor Campbell, and Frank Wood. Conditional permutation invariant flows. *arXiv preprint arXiv:2206.09021*, 2022.

A APPENDIX

We first present additional experiments and analysis, followed by datasets, shared training details, and experiment-specific settings and results.

A.1 ADDITIONAL EXPERIMENTS AND ANALYSIS

Closed-loop policy evaluation. Our central question is whether supervision recovered from third-person demonstrations is sufficiently accurate to train and improve an egocentric behavior-cloning policy. Open-loop minADE_5 directly evaluates expert-trajectory imitation accuracy, providing a controlled comparison between policies trained with ground-truth and mirror-generated supervision. For completeness, we additionally evaluate the trained policies in closed loop to examine whether the same trend holds during policy execution.

We evaluate policies from three training conditions in Table 1 on 20 routes of up to 50 m in Town 10, with balanced straight and turning scenarios, three policy seeds, and a seeded Traffic Manager. Following the per-route scoring rule of Bench2Drive (Jia et al., 2024), the composed score is $S_i = \mathrm{RC}_i \prod_j p_{i,j}$, where RC_i is route completion in percent and $p_{i,j}$ is the penalty for infraction j. We use its infraction categories and penalty values. Per seed, we average scores and completion over routes, count total collision events, and report the percentage of routes with an outside-lane violation.

Table 3: Closed-loop evaluation in Town 10. Results are mean $\pm$ standard deviation over three seeds. GT and mirror amounts are expressed as percentages of the full reference dataset of 33.6k samples, as in Table 1.

GT (%)	Mirror (%)	Composed score ↑	Collisions ↓	Route completion (%) ↑	Lane violations (%) ↓
70	0	80.9 ± 3.7	3.7 ± 1.9	97.2 ± 2.0	13.3 ± 2.4
70	30	84.6 ± 1.9	3.0 ± 1.6	97.4 ± 3.6	8.3 ± 2.4
100	0	84.8 ± 1.2	2.3 ± 1.7	94.3 ± 5.3	8.3 ± 2.4

Adding mirror data increases the mean composed score from 80.9 to 84.6 and reduces mean collisions and lane violations (Table 3). The 100% GT reference achieves a similar mean score of 84.8.

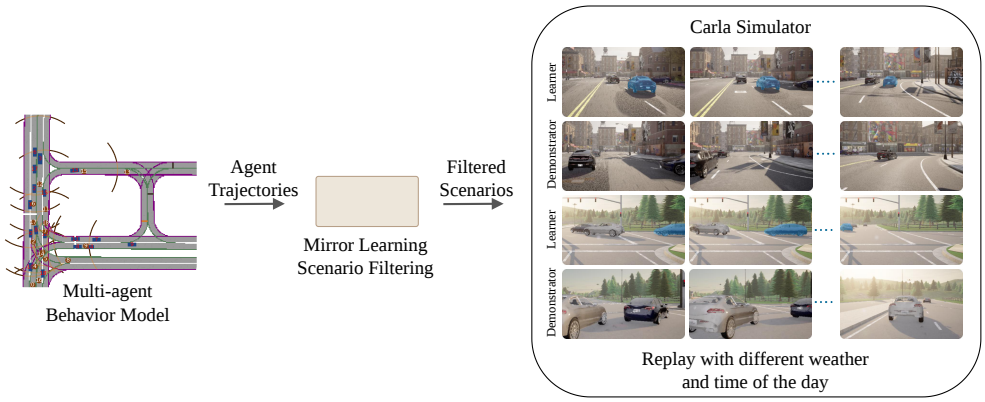


Figure 6: **Data generation pipeline.** Diverse multi-agent traffic scenarios are generated in Bird's-eye-view CARLA maps using a realistic scenario generator trained on real-world trajectories (Lioutas et al., 2025). After task-specific filtering for our mirror learning setting, the selected scenarios are replayed in the simulator to record candidate paired observations and the agents' executed trajectories, while demonstrator instance masks are obtained from CARLA's instance segmentation sensor.

Table 4: Counts of paired demonstrator–learner clips in the CARLA dataset by Town. Each clip spans 34 frames at 4 fps (8.5 s), for a total of approximately 52 hours of paired data.

Town	# of paired clips
Town04	3,096
Town06	7,286
Town10	6,571
Town03 (Held-out town)	5,013
Total	**21,966**

A.2 Self-Driving Datasets

A.2.1 CARLA Dataset

Existing open-source end-to-end driving datasets are predominantly egocentric (first-person) and generally do not provide paired third-person and first-person observations of the same scene Fong et al. (2021); Xu et al. (2025); NVIDIA (2025). In contrast, this kind of paired data exists naturally in proprietary robotaxi fleets, where multiple instrumented vehicles can observe the same scene from different viewpoints, but such data is not publicly available. Moreover, the scale of open driving data remains orders of magnitude smaller than what industrial systems are believed to leverage. For example, NVIDIA's open driving release contains on the order of 1,700 hours of real-world data, whereas Tesla has publicly stated that it trains on over 10 billion miles of fleet data. Motivated by the lack of publicly available paired data with action labels at sufficient scale, we study our approach in the CARLA simulator.

A potential concern is that simulated data, especially agent behavior produced by CARLA's default Traffic Manager, may be insufficiently realistic. To mitigate this sim-to-real gap in paired-data mining, we use a realistic multi-agent scenario generation model (Lioutas et al., 2025) trained on a large corpus of real-world trajectories to generate diverse traffic scenarios in bird's-eye-view CARLA maps. We further apply task-specific filtering to retain only scenarios that satisfy the requirements of our third-to-first video transfer setting (e.g., adequate third-person visibility of the demonstrator). These scenarios are then replayed in the simulator to identify candidate paired observations for recording. Instance masks are obtained through CARLA's instance segmentation sensor. The full data creation pipeline is shown in the Figure 6. We will release the dataset and the collection pipeline upon acceptance.

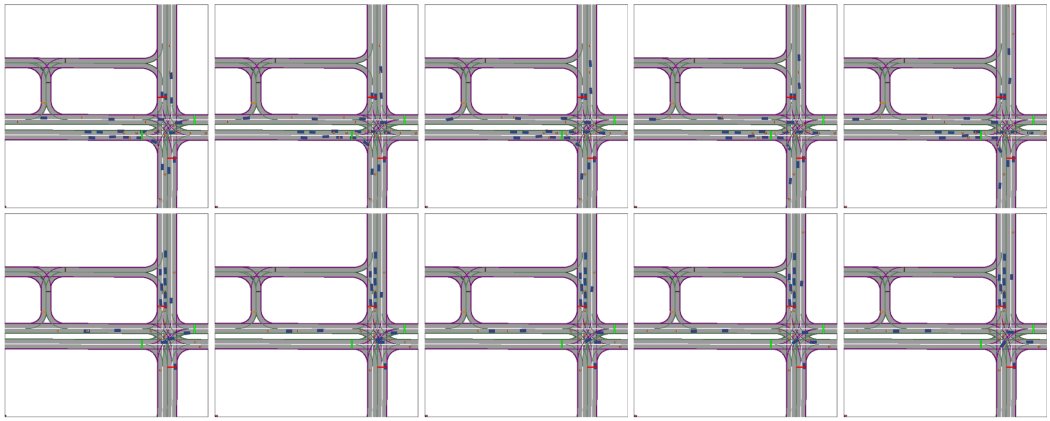


Figure 7: Bird's-eye-view traffic scenarios generated by a multi-agent driving behavior model (Lioutas et al., 2025) in CARLA Town10 and used in our experiments. The road geometry is shown in gray, blue rectangular boxes indicate vehicles, and traffic lights are annotated at their corresponding map locations. This visualization provides spatial context for the scene layout, vehicle distribution, and signalized intersections in the Town10 environment. Brown circles labeled with different numbers indicate the goal positions for each agent.

CARLA Dataset Collection. We generate diverse and realistic traffic scenarios in CARLA by using a traffic scenario initialization model (Zwartsenberg et al., 2022) to sample agent starting positions and speeds, and then roll out a data-driven object-level driving policy (Lioutas et al., 2025). Both the initialization model and the policy are trained on large-scale real-world multi-agent driving behavior data, enabling realistic traffic scenes and interactions. This process yields 21,966 paired demonstrator–learner source videos, each containing 34 frames rendered at four frames per second (8.5 seconds); the number of source videos per town is reported in Table 4. We then extract approximately 37,000 paired six-second clips using a sliding window and filter out clips in which the demonstrator is not visible in the learner's observation. The underlying simulation trajectories are generated at 10 Hz. The scenarios are generated by rolling out realistic vehicle-only traffic in four distinct CARLA towns. Samples of the traffic scenarios are shown in Fig. 7.

For each replayed scenario, we select learner and demonstrator vehicle pairs that satisfy visibility constraints using the underlying 10 Hz trajectories. The demonstrator must lie within the learner's camera field of view, using a horizontal field of view of 70° and a range of 24 m, and must have an unoccluded line of sight. We retain pairs with at least one continuous visible window of 30 or more simulation timesteps. We discard low-motion windows in which the target is stalled, defined as speed below 0.1 m/s for at least 10 consecutive simulation timesteps, since these examples provide limited interaction signal.

In all recordings, RGB cameras use the shared first-person-view preset, which places the camera approximately 2.2 m forward and 1.5 m above the vehicle origin with default rotation. This transform is used for both learner and demonstrator RGB streams. The instance segmentation camera used to obtain demonstrator masks is mounted on the learner vehicle with the same transform. During replay, the learner vehicle is fixed to Tesla Model 3, while other replayed agents, including target vehicles, are sampled from a small passenger vehicle pool.

Because all replayed demonstrator and learner vehicles are passenger cars, their front-view camera placements are similar across vehicle types. We therefore use a shared egocentric camera convention. The mirror video model predicts the demonstrator's front-view video in this common observation space. Since the inverse dynamics model is trained only on our ego vehicle type, decoded actions are interpreted as ego-executable actions that would reproduce the predicted mirror video, rather than as the target demonstrator's original controls.

Split Rationale. We split the CARLA data at the traffic-scenario level, where a traffic scenario denotes a single recorded multi-agent interaction rollout generated by the multi-agent trajectory prediction model and a recording denotes one paired demonstrator–learner clip extracted from that

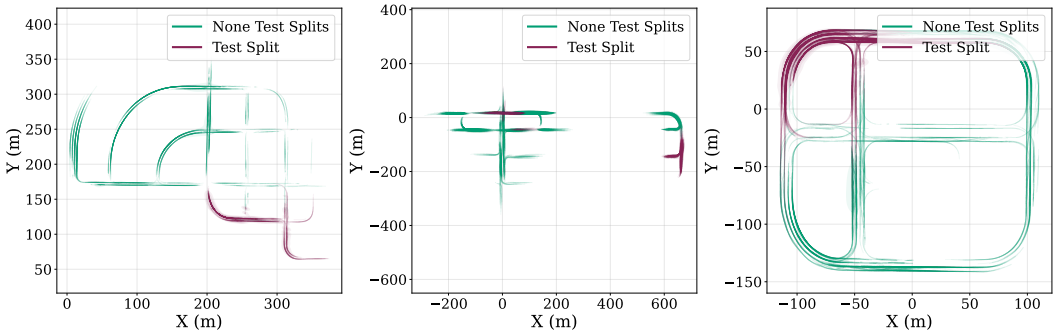


Figure 8: Spatial trajectory density comparison between the test split and the remaining data from Town 04, Town 06, and Town 10. We partition the data from Town 04, Town 06, and Town 10 into five disjoint splits: a training split for MVM training, a validation split for MVM model selection, a mirror-data generation split, a policy-validation split, and a test split. Importantly, the learner observations used to generate mirror data are excluded from MVM training. The test split contains different scenarios and spatial regions that are underrepresented in the other splits, as shown in this figure.

rollout. All recordings from the same scenario are assigned to the same split. Among the four towns, we reserve Town 03 as a held-out town for evaluation. We partition the data from Town 04, Town 06, and Town 10 into five disjoint splits: a training split for MVM training, a validation split for MVM model selection, a mirror-data-generation split, a policy-validation split, and a test split. Importantly the learner observations used to generate mirror data are excluded from MVM training. The test split uses different scenarios and spatial regions that are underrepresented to the rest of the splits. Figure 8 compares trajectory densities from the test split and the remaining data from Town 04, Town 06, and Town 10.

We do not enforce geographic separation among the MVM training, MVM validation, mirror-data-generation, and policy-validation splits within the non-held-out towns (Town 04, Town 06, and Town 10). This choice reflects two considerations. First, in practical autonomous driving deployment, additional fleet data is often collected from locations that have already been traversed. Second, even within a fixed town, variation in traffic scenarios, weather, and times of day provides a meaningful setting for evaluating whether the mirror video model generalizes beyond the exact conditions seen during training. We reserve stronger geographic generalization for evaluation on Town 03.

A.2.2 May Mobility Multi-Agent Dataset

We also evaluate zero-shot transfer on the multi-agent subset of MARS Li et al. (2024), a real-world multimodal autonomous-driving dataset collected from May Mobility vehicles. It contains 53 synchronized 30-second scenes with over 15,000 frames, capturing nearby vehicles with overlapping viewpoints for collaborative perception. Each vehicle records six surround-view RGB cameras, a 128-channel LiDAR, IMU, and GPS at 10 Hz in a NuScenes-compatible format. Only a small subset of scenes is suitable for our zero-shot setting, where one fleet vehicle observes another and provides ground-truth ego-view for the demonstrator. After manually filtering for paired learner-demonstrator data in which the demonstrator is visible in the learner's view, fewer than 20 scenes meet these requirements.

Mask Extraction. In general, we obtain the demonstrator's mask sequence using SAM 3 (Carion et al., 2025) in video mode with a text prompt such as "car," which yields per-frame instance masks together with temporally consistent object identifiers. In many settings, this is sufficient to track the target agent directly through its persistent identifier. For the May Mobility data, we use an additional dataset-specific selection step only for convenience, since ground-truth ego-view video is only available for a single demonstrator robotaxi and we therefore require a reliable way to recover

its corresponding track in the learner view for evaluation. Specifically, we use ego-pose annotations, camera intrinsics, and a fixed 3D vehicle cuboid to project an approximate 2D target bounding box into each frame, and then select the SAM 3 instance mask that best matches this box using a score based on box overlap and mask coverage, with a small temporal consistency bonus to reduce flicker. This yields an occlusion-robust target mask sequence without manual per-frame annotation.

A.3 MINECRAFT DATASET

Training and Validation Split for the Minecraft Dataset The Minecraft data used in this study is derived from the PLAICraft dataset (He et al., 2025), which comprises multiple gameplay sessions from a diverse set of participants. To make this data suitable for model training, we implement a systematic preprocessing pipeline. Our pipeline identifies visibility events between pairs of players through a three-stage refinement process. First, we use tracked player locations and viewing angles to identify candidate pairs satisfying geometric constraints on distance, field of view, and vertical alignment. Second, we employ OpenCV-based heuristics to remove cases where the learner view is obstructed by GUI elements using edge, color, and shape cues. Finally, SAM3 (Carion et al., 2025) based instance segmentation verifies true visibility by generating and propagating player masks across short video clips. Candidate clips are retained only if sufficiently large masks persist across most frames (more than 80%).

In our Minecraft clips, 40% contain at least one frame without a visible demonstrator, and 11% have the demonstrator absent in at least one-sixth of frames. The MVM conditions on the full sequence, allowing visible frames to provide context when the demonstrator is temporarily absent or occluded.

The extracted clips are fixed-length temporal segments of 3 seconds, sampled at 10 frames per second (FPS), resulting in 30 frames per clip at a spatial resolution of 640x360 pixels. The dataset consists of a total of 712 gameplay sessions, which are partitioned at the session level into 676 training sessions and 36 validation sessions. We assign entire gameplay sessions exclusively to either the training or validation split to eliminate temporal and contextual overlap, preventing data leakage and ensuring that validation samples come from unseen gameplay trajectories.

To further increase the number of training samples and capture temporal dynamics, we construct windowed segments using a sliding window strategy within each clip. Specifically, each 30-frame clip is subdivided into sequences of length 25 frames with a stride of 2, producing 3 overlapping segments per clip. This results in a total of 6,003 clips (18,009 segments) for training and 253 clips (759 segments) for validation, yielding 6,256 clips and 18,768 windowed segments overall.

A.4 SHARED TRAINING DETAILS

Unless otherwise noted, the following settings are shared across experiments in the paper.

Mirror Video Model. We fine-tune NVIDIA Cosmos with a learning rate of 3×10^{-4}. For the CARLA dataset, the model takes as input a 25-frame video clip at 512×288 resolution sampled at 4 Hz together with the corresponding instance mask at each frame, and predicts a 25-frame video clip at the same frame rate. For the Minecraft dataset, we use the same resolution at 10 fps. At inference time, we use Cosmos's built-in multistep flow-matching solver Zhao et al. (2023) with 35 denoising steps. We train MVM on the CARLA dataset using four H100 GPUs for 15,000 steps with a batch size of 10 per GPU, which takes approximately five days. The Minecraft model was trained on eight L40S GPUs for 9800 steps with a batch size of 5 per GPU, which took approximately 42 hours. Cosmos also takes a text prompt as input. In this work, we use a fixed generic prompt for each dataset. For CARLA, we use: *"Photoreal CARLA simulation, single continuous take. Imagine the first-person perspective of the target car from a third-person observation, with consistent vehicle and lane geometry. Follow kinematics: no jumps, cuts, or teleports."* We leave the exploration of more informative text prompts to future work.

CARLA Policy Learning. For downstream policy learning, we adopt VaViM as the visual backbone, initialize from the pretrained model, and add an action prediction head as in VaVAM, without any architecture modifications. The backbone hidden size is 768. Unless otherwise noted, all policy variants share the same architecture and training protocol. Following VaVAM, we evaluate forecasting accuracy using minimal average displacement error, minADE_5, which has been used for end-to-end

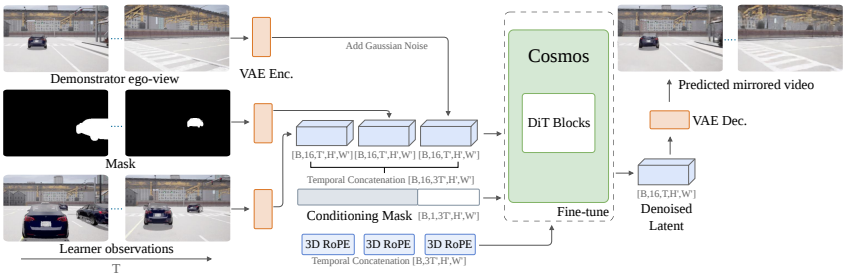


Figure 9: **Architecture of our mirror video model (MVM).** Given a learner ego-view video and a binary attention-mask video that identifies a demonstrator, the MVM predicts the corresponding demonstrator ego-view video. During training, we fine-tune COSMOS-PREDICT 2.5-2B to denoise a noised latent encoding of the target demonstrator ego-view conditioned on the learner video and the demonstrator mask. The temporally aligned learner video, demonstrator mask, and noised target videos are encoded with a spatio-temporal VAE encoder, and the resulting latents are concatenated along the temporal dimension. To adapt the pretrained video backbone to this multi-stream input, we assign the same 3D RoPE to tokens with aligned spatiotemporal positions across the three encoded videos and use a conditioning mask to distinguish observed from denoised tokens. The denoised latent is then decoded to produce the mirrored video, i.e. the inferred first-person video of the demonstrator.

policy evaluation in works such as Alpamayo-R1 (Wang et al., 2025). The policy is conditioned on a fixed history window of 6 observations and predicts 6 future actions at 2 Hz. We tune the learning rate with a hyperparameter sweep and use 10^{-3} for CARLA policy training unless otherwise specified.

CARLA Inverse Dynamics Model. For IDM training, we use a learning rate of 0.0194 with a batch size of 16. The model receives a sequence of 7 observations as input and predicts the 6 actions between consecutive observations. Driving commands are heuristically classified from the output actions using the heuristic provided in the VaVaM code. We train for 10 epochs and observe that the validation average displacement error (ADE) has converged by the end of training. Training is performed using 4 NVIDIA RTX A5000 GPUs. We train the IDM on learner observations and actions from the MVM training split in the non-held-out towns, while keeping the mirror-data generation split disjoint from the IDM training data.

A.5 MIRROR VIDEO MODEL EXPERIMENTS

This section provides supplementary results for the mirror video model. We begin with CARLA evaluation, including both quantitative results and qualitative generalization examples, then present zero-shot transfer to the real-world May Mobility dataset, and finally show qualitative results on Minecraft. We also include additional mirror video qualitative results in Section A.7.

A.5.1 CARLA EVALUATION

Unless otherwise noted, for the CARLA dataset we train the mirror video model on the MVM training split mentioned in App. A.2.1. After careful filtering, we obtain 14,500 training segments constructed with a sliding window stride of 4. Quantitative evaluation is performed on 1,024 samples from each town. For Town 04, Town 06, and Town 10, we use samples from the MVM validation split. We also evaluate on 1,024 samples from Town 03, which is excluded from MVM training, to assess generalization to an unseen environment.

Quantitative Results. To facilitate reproducibility and provide baseline metrics for future work on mirror video modeling, we report quantitative results for the mirror video model in Table 5. Metrics are computed over 1,024 samples from each of the four towns, using the evaluation sets described above. We report FVD, PSNR, and SSIM, where lower FVD indicates better video distribution fidelity and higher PSNR/SSIM indicate better frame-level reconstruction quality.

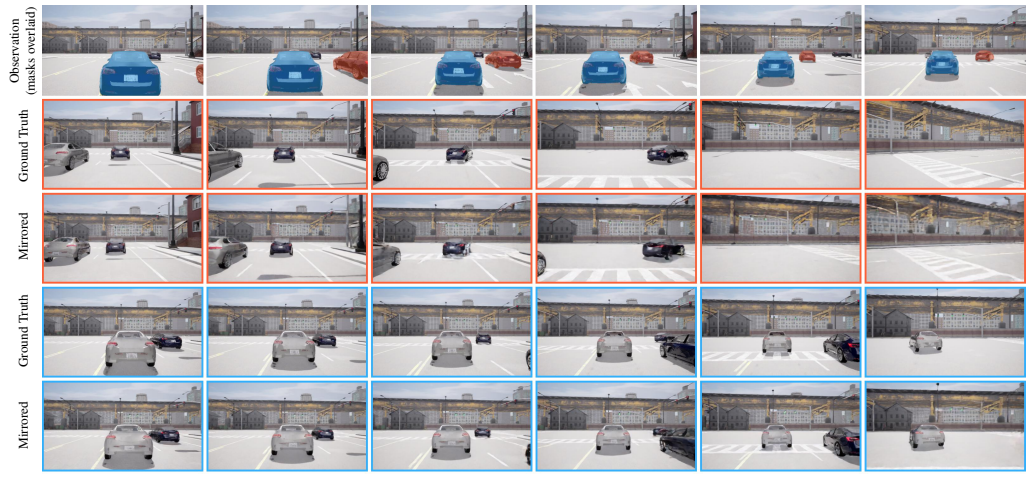
Table 5: Quantitative evaluation on CARLA towns for the mirror video model. Town 04, Town 06, and Town 10 are training towns and Town 03 is held out. Lower FVD is better; higher PSNR and SSIM are better.

Town	**FVD** ↓	**PSNR** ↑	**SSIM** ↑
Town 04	56.27	19.12	0.59
Town 06	39.88	23.51	0.76
Town 10	39.63	19.61	0.54
Town 03	58.82	19.98	0.66

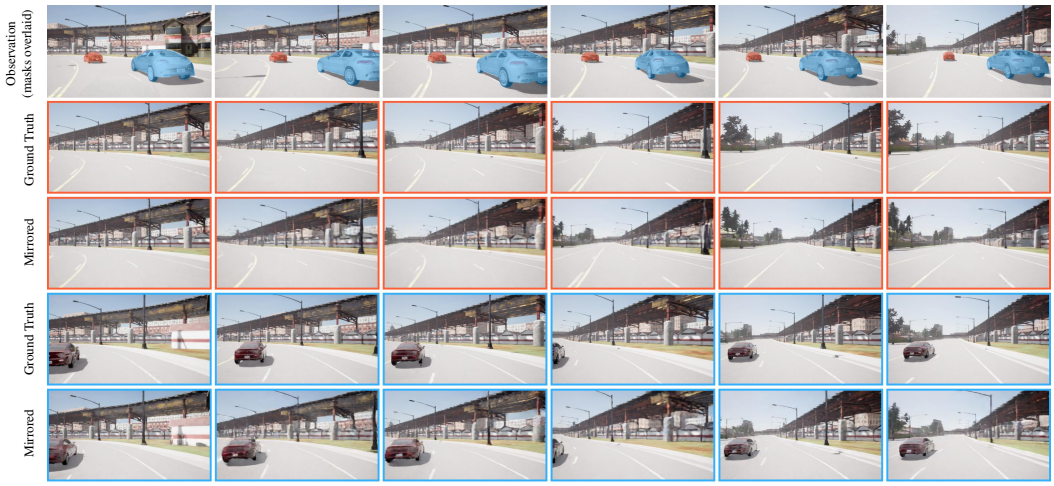
Qualitative Results. We show qualitative results beginning with the held-out Town 03 split, followed by results from the training towns in the additional results section.

A.5.2 CARLA HELD-OUT TOWN

We next present qualitative results on CARLA Town 03, a held-out urban environment with large junctions and a roundabout. This map provides a challenging test of whether the mirror video model generalizes beyond the towns used for training. Figure 10 shows that MVM generates target-specific and temporally consistent egocentric videos in the held-out town, correctly capturing distinct agent behaviors and anticipating scene content from future observation context. Figure 11 further shows that MVM captures uncertainty under partial observability by generating diverse but plausible egocentric samples, including variations in occluded vehicles and surrounding scene structure.



(a)



(b)

Figure 10: Qualitative mirroring results in the held-out town for MVM. In each example, the top row shows the learner egocentric view with two demonstrator attention masks overlaid in orange and blue. For each demonstrator, we show the ground-truth egocentric video and one sample from our mirror video model. (a) The orange agent takes a right turn while the blue agent drives straight, demonstrating target-specific egocentric generation. (b) MVM produces temporally consistent scene predictions by leveraging future context from the observation sequence: although the distant building is not visible until the sixth observation frame, the model anticipates its presence in the last generated frame for both agents.

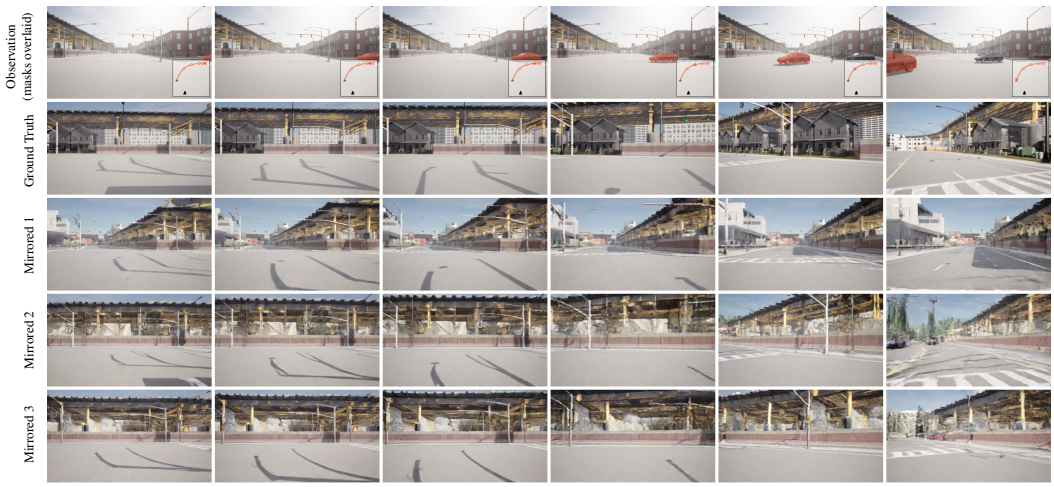


(a)

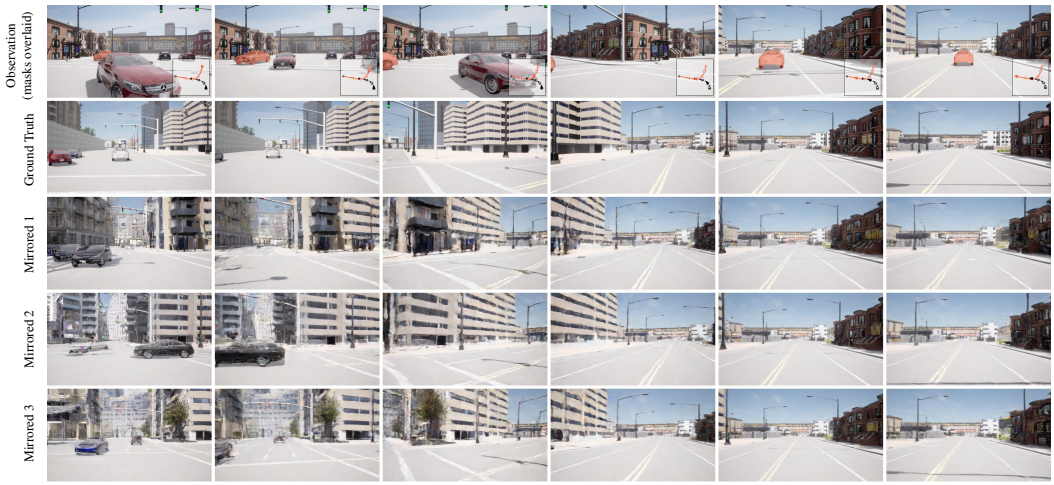


(b)

Figure 11: Qualitative diversity results in the held-out town for MVM. In each example, the top row shows the learner egocentric view with two demonstrator attention masks overlaid in orange and blue. For each demonstrator, we show the ground-truth egocentric video and two samples from our MVM, demonstrating target-specific egocentric generation from a shared observation. (a) MVM captures uncertainty in occluded scenes by generating diverse plausible outcomes for the blue agent: one sample predicts a car ahead, whereas another does not. (b) MVM also captures uncertainty in the surrounding environment by generating diverse but plausible structures, such as buildings appearing on the left side of the frame when this information is unobserved by the learner.



(a)

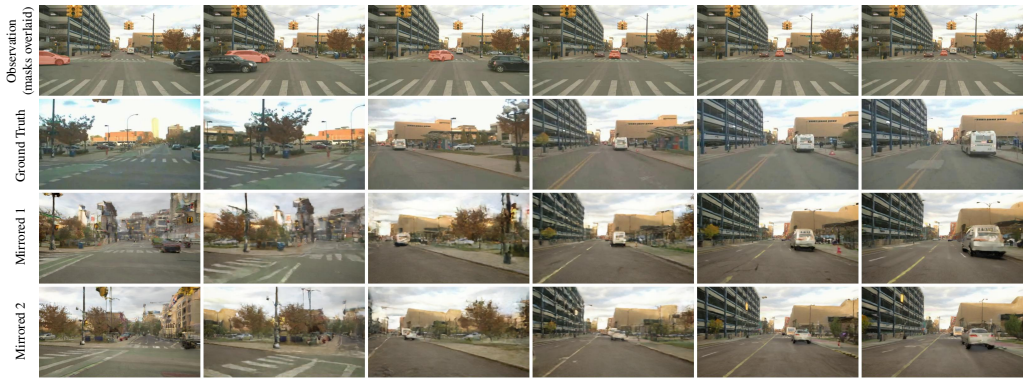


(b)

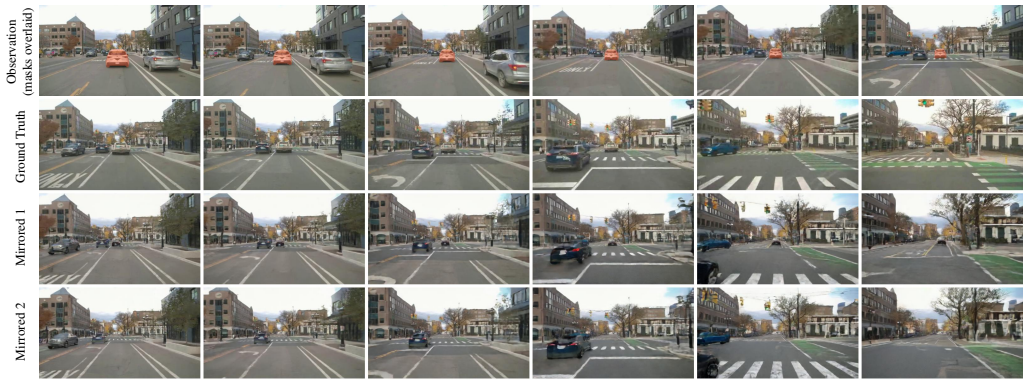
Figure 12: Qualitative passing-diversity results in the held-out town. In these challenging pass-by scenarios, the learner observes the demonstrator from a different viewpoint, with limited overlap with the demonstrator's front-facing view. To clarify the interaction geometry, we additionally show a bird's-eye-view trajectory subplot alongside the learner's observations, oriented to the learner's heading in the first frame. The dark triangle denotes the learner, and the orange triangle denotes the demonstrator. (a) Given the learner's observation and the demonstrator's instance mask, the mirror video model predicts the correct left-turn behavior across multiple sampled egocentric videos. (b) In this scenario, the learner makes a left turn while the demonstrator turns right in the opposite direction. The two agents have limited observation overlap at the beginning of the scene, leading to diverse predicted backgrounds; once the learner completes the turn and the views become more aligned, the predicted scene becomes consistent.

A.5.3 MAY MOBILITY ZERO-SHOT TRANSFER

We show qualitative zero-shot transfer results on the May Mobility dataset described above in Figures 13–14. Target-agent masks are obtained with SAM 3 (Carion et al., 2025) together with the dataset-specific selection procedure described in the dataset section.

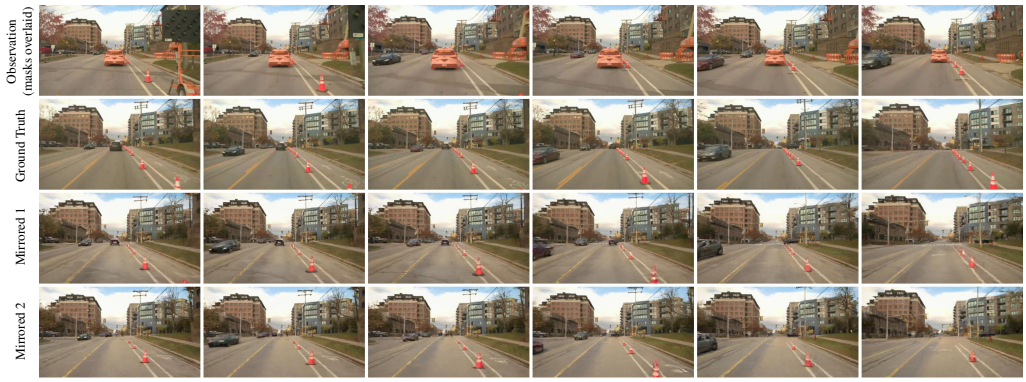


(a) The demonstrator makes a left turn while partially occluded by a black Mini, yet MVM predicts consistent demonstrator-centric egocentric views.



(b) MVM generates plausible real-world demonstrator-centric views from learner observations and SAM3 instance masks.

Figure 13: Qualitative zero-shot mirroring results on real-world data using our mirror video model trained only in the CARLA simulator. Given the learner's observations and the demonstrator instance mask generated by SAM3, we show the demonstrator's egocentric ground-truth video and two mirrored samples generated by our model.



(a) MVM transfers to a real-world scene containing traffic cones, which are absent from the CARLA training data.



(b) MVM generates diverse plausible road markings, including stop text and a straight-ahead arrow.



(c) MVM infers the demonstrator's right-turn behavior from real-world learner observations.

Figure 14: Qualitative zero-shot mirroring results on real-world data using our mirror video model trained only in the CARLA simulator. Given the learner's observations and the demonstrator instance mask generated by SAM3, we show the demonstrator's egocentric ground-truth video and two mirrored samples generated by our model.

A.5.4 MINECRAFT EVALUATION

Experiment-Specific Setup. Minecraft experiments use the Plaicraft training and validation splits described above. We train and evaluate the mirror video model on 25-frame windows extracted from 3-second clips. Below, we show quantitative results and qualitative results on the held-out validation split. Additional qualitative results on the training split in the additional results section.

Table 6: Quantitative evaluation on the held out Minecraft dataset for the mirror video model. Lower FVD is better; higher PSNR and SSIM are better.

FVD ↓	**PSNR** ↑	**SSIM** ↑
227.60	16.42	0.45

Quantitative Results To facilitate reproducibility and provide baseline metrics for future work on mirror video modeling, we report quantitative results for the mirror video model in Table 6. Results are computed over 750 validation samples from the validation set. We report FVD, PSNR, and SSIM, where lower FVD indicates better video distribution fidelity and higher PSNR/SSIM indicate better frame-level reconstruction quality.

Qualitative results on Validation-Set Figures 15 and 16 show qualitative mirroring results of our MVM model on Minecraft validation set. Compared to the CARLA dataset, Minecraft presents a more challenging setting because the relative camera poses between the learner and demonstrator are much more flexible and unconstrained. As a result, the learner and demonstrator often observe each other from viewpoints with limited perspective overlap, for example when the two players are simply facing one another. This partial observability leads to higher diversity across generated samples. Nevertheless, MVM can exploit visual cues in the learner observation, such as the demonstrator's pose and held weapon, to generate plausible demonstrator-centric egocentric views.



(a) Mirror video model generation captures how the learner is holding a purple block.

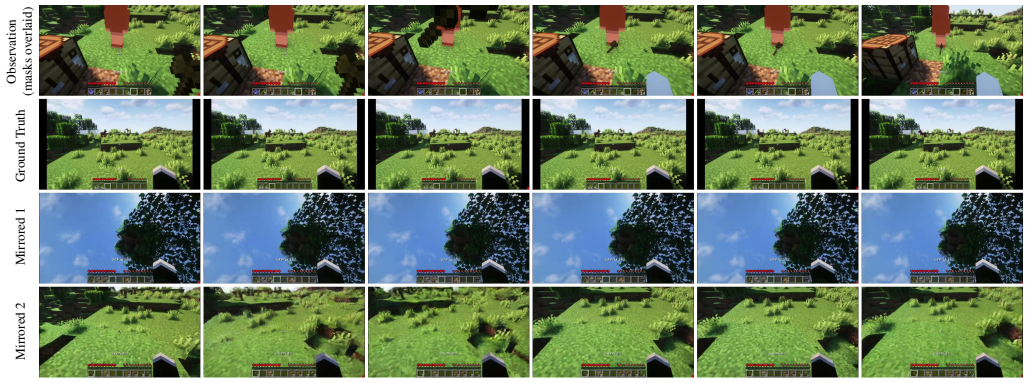


(b) Mirror video model is able to generate a first-person view of a player engaged in block placing action.



(c) Mirror video model captures first-person view of running through passable bamboo blocks.

Figure 15: Validation set qualitative mirroring results for a scenario the mirror network has not been trained on. We select a target agent using orange instance masks. For the target, we show the ground-truth egocentric video and two sampled predictions from the perspective transfer network, demonstrating egocentric generation from a shared observation.



(a) Mindful of the ambiguity over the head orientation of the observed player, the mirror video model generates first-person views that looks at different directions.



(b) Mirror video model renders fine-grained details like floating items.



(c) Mirror video model renders diverse observing player appearances in its generations.

Figure 16: Validation set qualitative mirroring results for a scenario the mirror network has not been trained on. We select a target agent using orange instance masks. For the target, we show the ground-truth egocentric video and two sampled predictions from the perspective transfer network, demonstrating egocentric generation from a shared observation.

A.6 Mirror Learning Experiments

This section provides supplementary details for the downstream mirror learning results in the main paper. Specifically, it documents the experimental setups used for Figure 5 and Table 1, and concludes with qualitative examples for the style-transfer baseline used in the geographic-expansion experiment. Details of our CARLA dataset split are provided in the "Split Rationale" paragraph of App. A.2.1.

A.6.1 Inverse Dynamics Model Implementation and Evaluation

Implementation. We base our AV IDM on the VaViM foundation model and its corresponding VaVAM video action model (Bartoccioni et al., 2025). Our IDM predicts timestamped relative displacements, which form the trajectory-based action labels used to train the downstream VaVAM policy. VaVAM is a stochastic flow-matching model of future actions conditioned on prior video observations. An IDM, however, must infer vehicle actions conditioned on the observations made while those actions were taken. To this end, we replace the flow-matching future-action sequence head with a head that predicts both the mean action and its heteroscedastic uncertainty in a single forward pass. We train on learner video–action pairs using the following objective:

$$-\log f_\psi(a_{0:T-1} \mid o_{0:T}) = \sum_{t=0}^{T-1} \left\| \frac{a_t - \mu_\psi(o_{0:T})_t}{2\sigma_\psi(o_{0:T})_t} \right\|_2^2 + \log \sigma_\psi(o_{0:T})_t. \tag{5}$$

Once trained, we use the point estimate $\hat{a}^j_{0:T-1} = \mu_\psi(\hat{o}^j_{0:T})$ as the learner-executable action labels for behavioral cloning. The IDM receives seven observations and predicts the six intervening actions. Training details are provided in App. A.4.

Evaluation and controlled comparison. Table 7 reports average displacement error on ground-truth validation videos and mirror videos. We compare our VaVAM-based IDM, denoted VaVAM-IDM, with a variant that retains the original flow-matching action head, denoted VaVAM-IDM-FM. Because the flow-matching head is stochastic, we average its predicted trajectories over three samples. The two variants use the same backbone and training data, providing a controlled comparison of the action heads. VaVAM-IDM achieves lower error on both video sources. These component metrics are distinct from the best-of-five downstream policy minADE_5.

A.6.2 Policy Experiments

Figure 5a: Mirror-Only Behavior Cloning. For the mirror-only behavior cloning experiment in Figure 5(a), we train the stock VaVAM architecture, conditioned on six history frames at 2 Hz, to predict three seconds of future actions given a driving command. We train policy variants at three dataset scales using two data sources: ground-truth first-person demonstrations and mirror data. Ground-truth first-person demonstrations are subsampled from the MVM training split. Mirror data is generated from the disjoint mirror-data-generation split in Town 04, Town 06, and Town 10. As described in the dataset split rationale in App. A.2.1, the CARLA dataset is split at the traffic-scenario level, so each scenario appears in exactly one split. Within each town, weather and other scene conditions vary across scenarios. Learner observations used for mirror-data generation are excluded from both MVM and IDM training. Across dataset scales, we subsample with a fixed random seed so that the town distribution is matched as closely as possible.

We evaluate all policies using open-loop minADE_5 on the test split in the non-held-out towns, which contains 10,294 segments constructed from scenarios and spatial regions that are separated from or underrepresented in the main training distribution. Unless otherwise noted, we use the same learning rate across all data scales and train each policy until convergence.

Contribution of video generation and action inference. To examine how each component affects policy supervision, we extend the 8k-sample setting of Fig. 5a, retaining its data splits and policy-training procedure, and vary the observation and action sources. Table 8 reports the four combinations over three seeds.

With ground-truth observations, replacing recorded actions with IDM predictions increases policy error, and using MVM-generated observations introduces further degradation, indicating that both

Table 7: IDM evaluation on ground-truth validation videos and mirror videos. Values are average displacement errors; lower is better. Both variants use the same backbone and training data. VaVAM-IDM-FM averages predicted trajectories over three samples.

IDM	GT validation videos	Mirror videos
VaVAM-IDM-FM	0.27	0.49
VaVAM-IDM	**0.20**	**0.42**

Table 8: Policy supervision ablation at 8k training samples. Results are mean $\pm$ standard deviation over three seeds. GT actions paired with MVM videos come from the original recorded trajectories.

Observation source	Action source	minADE_5 (m) $\downarrow$
Ground truth	Ground truth	0.88 ± 0.00
Ground truth	IDM	0.92 ± 0.00
MVM	Ground truth	0.97 ± 0.01
MVM	IDM	0.95 ± 0.00

components contribute to the downstream performance gap. For generated observations, IDM-inferred actions yield slightly lower policy error than recorded actions, suggesting that inferring actions from the generated video helps maintain coherent supervision under imperfect generation.

Figure 5b: Geographic Expansion. For the geographic-expansion experiment in Fig. 5b, we start from the pretrained policy trained on 27.9k first-person ground-truth samples, corresponding to the 83% GT-only condition in Table 1, and fine-tune it with a learning rate of 10^{-6} on either 1k or 2k additional ground-truth samples from the held-out town, Town 03. Within Town 03, the adaptation data and evaluation data are split at the traffic scenario level, so no scenario is shared between fine-tuning and evaluation. We compare three augmentation choices: no augmentation, style-transfer augmentation, and mirror-data augmentation. For augmentation with mirror data, we keep the mirror video model, pretrained on the MVM training split from the non-held-out towns, fixed without further training or fine-tuning on Town 03. We pair it with an IDM pretrained on the MVM training split and further fine-tuned on the additional first-person ground-truth data from Town 03 to generate extra behavior-cloning samples from learner videos in the held-out town. The fine-tuned IDM achieves an average displacement error of 0.67 when fine-tuned on 1k Town 03 first-person ground-truth samples and 0.66 when fine-tuned on 2k samples. For style transfer, we preserve the same scene layout and ego trajectory by conditioning on the original edge structure while varying the text prompt; examples are shown in App. A.6.4. At the $1\times$ augmentation ratio, mirror data achieves a lower mean minADE_5 than style transfer (0.678 versus 0.682); both round to 0.68 in Fig. 5b.

Table 1: Mirror-Data Augmentation. The 27.9k ground-truth policy-training windows used in the 83% GT conditions were extracted from the same 14,500 video segments used to fine-tune the MVM. We extracted policy-training windows with a one-frame stride and removed fully stationary windows. Thus, the 14.5k and 27.9k counts refer to MVM training segments and policy-training windows derived from those segments, respectively, rather than separate data collections.

Within each paired group in Table 1, we compare a ground-truth-only policy with a policy trained on the same first-person dataset plus mirror data. GT and mirror amounts are expressed as percentages of the full reference dataset of 33.6k samples. The GT-only baselines use 70% or 83% GT data, and the paired GT+mirror conditions add 30% or 17% mirror data, respectively. The mirror data is generated from learner observations in the disjoint mirror-data-generation split in Town 04, Town 06, and Town 10 using a mirror video model and IDM trained on the MVM training split. These components are shared across all table conditions. The IDM used for this experiment achieves average displacement errors of 0.20 on ground-truth validation videos and 0.42 on mirror videos (Table 7). Policies are evaluated both on the test split in the non-held-out towns and zero-shot on the held-out town, Town 03. The paired comparisons test whether adding mirror data improves policy performance at a fixed first-person policy-data budget. Separately, the full-GT reference uses 33.6k ground-truth samples, matching the total sample count of each GT+mirror condition.

Table 9: Minecraft policy evaluation using action negative log-likelihood (NLL) on 314 held-out evaluation windows of 25 frames each. Results are mean ± standard deviation over three seeds, in nats per frame. Lower is better. Counts denote training windows, and additional GT consists of recorded learner observation–action data. Bold marks the lowest unrounded mean. Policy checkpoints are selected by the lowest validation loss.

GT windows	Mirror windows	Action NLL ↓
5,798	0	1.10±0.01
7,702	0	1.02±0.01
5,798	1,904	1.05±0.01
7,702	1,904	**1.00**±0.01

Comparisons against this reference assess the performance of GT+mirror mixtures relative to using only ground-truth data at the same total sample count.

A.6.3 Minecraft Policy Evaluation

We evaluate policies offline on 314 held-out PLAICraft evaluation windows. The gameplay sessions from which these windows are drawn are excluded from MVM training, IDM adaptation, and policy training and validation. The baseline uses 5,798 ground-truth (GT) policy-training windows derived from the learner's observations and recorded actions in the Minecraft MVM training split. We additionally acquire 1,904 GT windows that are excluded from MVM training and use their observation sequences to generate 1,904 mirror windows, allowing us to evaluate mirror data as additional supervision.

Policy initialization and training. All runs start from the pretrained VPT $1\times$ foundation policy (Baker et al., 2022). Our Minecraft gameplay recordings were collected after the VPT policy was pretrained, so they were not included in its pretraining data. We freeze the visual encoder and projection, fine-tune the temporal backbone and existing action heads, and keep the unused value head frozen. For each run, we select the policy checkpoint with the lowest validation loss. Each evaluation window contains 25 frames. Results are reported over three seeds.

Mirror-action labeling. We adapt the pretrained VPT $4\times$ IDM using LoRA for three epochs. Both fine-tuning and validation use the learner's observations and recorded actions drawn exclusively from the Minecraft MVM training set, with disjoint gameplay sessions for the two splits. We use LoRA with rank $r = 8$, scaling factor $\alpha/r = 2$ ($\alpha = 16$), and dropout 0.05. We adapt the query, key, value, and output projections in both temporal-transformer blocks, together with the button and camera output linear layers, for a total of 10 layers. The adapted IDM infers action labels from the generated mirror videos.

Table 9 reports the action NLL results from Table 2 with the exact training-window counts.

Offline action likelihood. We evaluate action prediction on held-out demonstrations using VPT's discrete button actions, such as moving forward, jumping, and attacking. We report action negative log-likelihood (NLL):

$$\mathcal{L}_{\text{action}} = -\frac{1}{N}\sum_{t=1}^{N} \log \pi_{\text{button}}(b_t \mid h_t), \tag{6}$$

where h_t denotes the observation history, b_t is the ground-truth button action under VPT's action mapping, and N is the number of evaluated frames. Action NLL is averaged over all evaluated frames and reported in nats per frame. Lower values indicate better held-out action prediction.

A.6.4 Style-Transfer Baseline Examples

We apply simulation-to-real style transfer to CARLA clips using Cosmos-Transfer2.5-2B (Ali et al., 2025), a 2B-parameter video diffusion model conditioned on structural control signals. Edge maps extracted from simulator frames are used as layout conditioning, preserving road geometry, building placement, and vehicle layout while allowing photorealistic appearance changes. We run inference at

720p for 93-frame windows with 35 diffusion steps and guidance scale 3, cycling text prompts for diverse lighting and weather conditions. Prompts are generated by a large language model.



(a) *A first-person driving simulator scene in the sparkling light of a clear spring day with a gentle breeze, with an assortment of brightly colored cars in every direction. Maintaining a steady forward motion, the camera glides through the cityscape, keeping roads, intersections, and buildings in a clear and continuous frame.*



(b) *A first-person driving simulator scene in the sparkling light of a clear spring day with a gentle breeze, with an assortment of brightly colored cars in every direction. Maintaining a steady forward motion, the camera glides through the cityscape, keeping roads, intersections, and buildings in a clear and continuous frame.*



(c) *A first-person driving simulator scene on a snowy winter day with white-dusted streets and rooftops, alongside dark charcoal and deep navy luxury vehicles. Maintaining a steady forward motion, the camera glides through the cityscape, keeping roads, intersections, and buildings in a clear and continuous frame.*



(d) *A first-person driving simulator scene as a gentle rain mists the air beneath pale overcast skies, with an assortment of brightly colored cars in every direction. The perspective advances smoothly through urban intersections and streets, holding a stable forward-facing view that captures the road and the surrounding architecture.*

Figure 17: Style-transfer examples using text prompts and edge conditioning with Cosmos-Transfer2.5-2B. In each subfigure, the top row shows the ground-truth driving sequence and the bottom row shows the corresponding style-transferred output.

A.7 Additional Mirror Video Qualitative Results

In this final subsection, we provide additional qualitative mirror video results for completeness.

A.7.1 CARLA Seen-Town Test Set

We then show qualitative results on Test One, a seen-town benchmark built from scenarios and spatial regions that are separated from or underrepresented in the main CARLA training distribution. These examples complement the held-out-town results by isolating generalization to new situations within otherwise familiar towns.



(a)

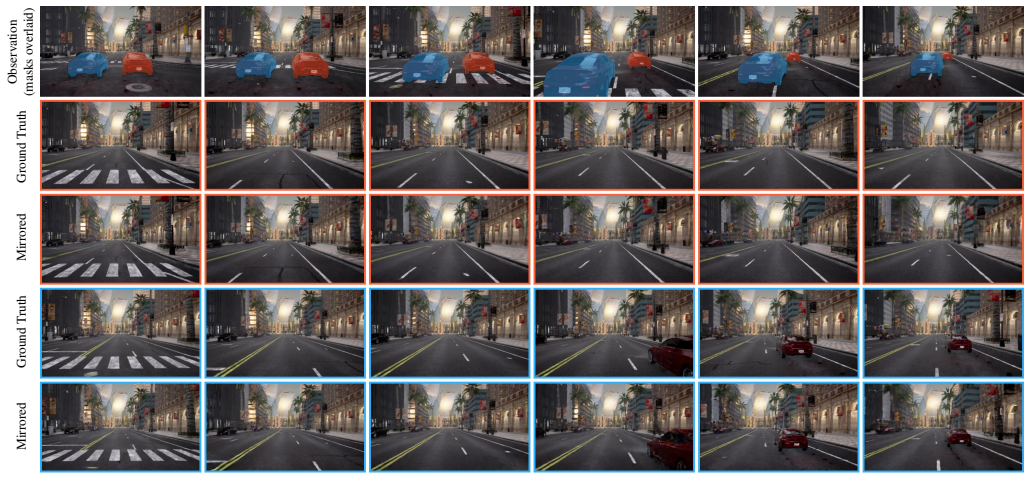


(b)

Figure 18: Demonstrator-specific view transformation in CARLA. In each example, MVM is conditioned on the same learner observation but different demonstrator masks, shown in orange and blue. For each selected demonstrator, we show the ground-truth demonstrator egocentric video and the corresponding mirrored sample generated by our method. Because only the demonstrator mask changes between the orange and blue samples, these results demonstrate that MVM generates target-specific egocentric observations.



(a)



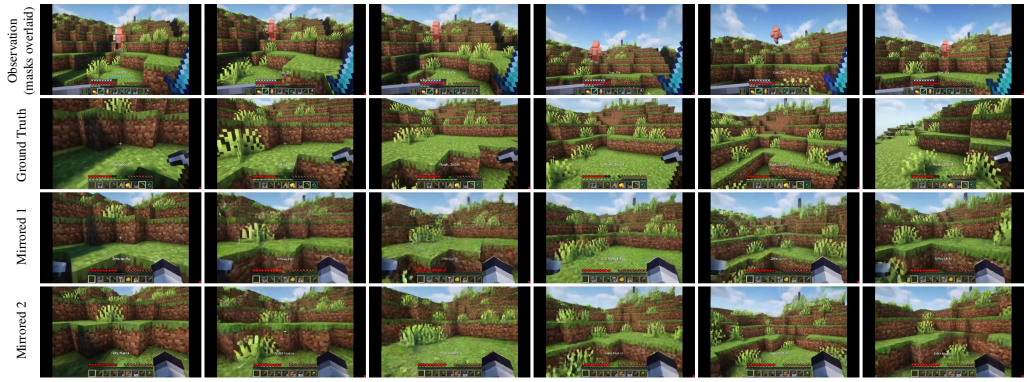
(b)

Figure 19: Additional demonstrator-specific view transformation results in CARLA Test locations Training town.

A.8 MINECRAFT TRAINING-SET EXAMPLES.



(a) Mirror video model's generates a cave's texture with plausible torch placements.



(b) Mirror video model effectively captures the player jump motion.



(c) Mirror video model faithfully renders observed player's camera movement.

Figure 20: Training set qualitative mirroring results for a scenario the mirror network has been trained on. We select a target agent using orange instance masks. For the target, we show the ground-truth egocentric video and two sampled predictions from the perspective transfer network, demonstrating egocentric generation from a shared observation.

A.9 Social Impact

Our work aims to reduce the cost of training embodied policies by reusing observations from a learner in an environment where multiple demonstrators exist and by releasing paired simulation data that can support future research. This could broaden access to mirror-video modeling and data-efficient policy learning for academic groups. At the same time, methods that improve autonomous decision making can create downstream safety and misuse risks if deployed without careful validation. We therefore view our approach as a research tool that should be paired with rigorous testing, uncertainty-aware evaluation.