GeoIDM: Action-Reprojected Geometric Supervision for Generalizable Inverse Dynamics

Anonymous authors
Paper under double-blind review

Abstract

Video-based robot policies can decouple visual planning from control by translating predicted future observations into actions with an inverse dynamics model (IDM). However, IDMs supervised solely by absolute end-effector poses and gripper states can overfit appearance cues and generalize poorly to unseen trajectories and tasks. We present **GeoIDM**, a **Geo**metry-supervised **IDM** that grounds action prediction in temporally aligned visual correspondences. A frozen VGGT encoder extracts visual tokens with geometric information from current and future observations, and a lightweight query compressor conditions an action DiT. During training, GeoIDM applies action-reprojected geometric supervision by accumulating predicted translations to estimate the future end-effector position, reprojecting it into the future view using camera calibration, and aligning the sampled feature with the corresponding feature in the current frame through a local contrastive objective. A gate suppresses static or invalid projections. The geometric branch is removed at inference, so the model still maps visual observations to actions without an additional geometry module. We evaluate GeoIDM through offline action prediction on AgiBot, replay of reference videos in LIBERO and RoboTwin 2.0, and replay of opening, clicking, and grasping tasks on a physical Franka robot. Relative to the DreamGen baseline, GeoIDM reduces position error by 63.9% and 29.3% on the in-domain and out-of-domain AgiBot splits, respectively. It further improves out-of-domain replay success by 20.5 and 18.0 percentage points on LIBERO and RoboTwin 2.0, respectively. These results show that geometric supervision during training improves IDM generalization while preserving the standard interface at inference.

1 Introduction

Decoupling visual planning from physical control has emerged as a promising paradigm for generalist robot manipulation (Feng et al., 2025a; Li et al., 2026b; Zhang et al., 2026c). Under this paradigm, a video generative model leverages broad visual priors learned from Internet-scale data to predict the visual progression of a task, while an *inverse dynamics model* (IDM) translates the imagined visual trajectory into executable robot actions (Feng et al., 2025a;b; Li et al., 2026b). By delegating high-level semantic reasoning to video generation, the IDM focuses specifically on physical robot control: given a current observation and a planned future observation, it infers the end-effector actions required to realize this visual transition in the real world.

The bottleneck of this decoupled paradigm, however, lies in the generalizability of the IDM across novel scenes and tasks. Direct image-to-action IDMs offer a minimal deployment interface (Feng et al., 2025a; Jang et al., 2025), yet supervision on action labels alone does not explicitly enforce agreement between predicted robot motion and the observed visual transition (Figure 1a). Without such a constraint, the model has no direct incentive to ground its predictions in spatial correspondence, leaving open the possibility of relying on appearance cues that may not transfer across scenes or tasks. Conversely, methods that incorporate explicit geometric representations, such as 3D point tracks, optical flow, or depth maps (Tong et al., 2026; Mi et al., 2026; Zhang et al., 2026a; Li et al., 2026b), can introduce additional estimation stages at inference and propagate tracking noise to downstream control (Figure 1b). This dilemma raises a fundamental question: *Can we ground in-*

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.

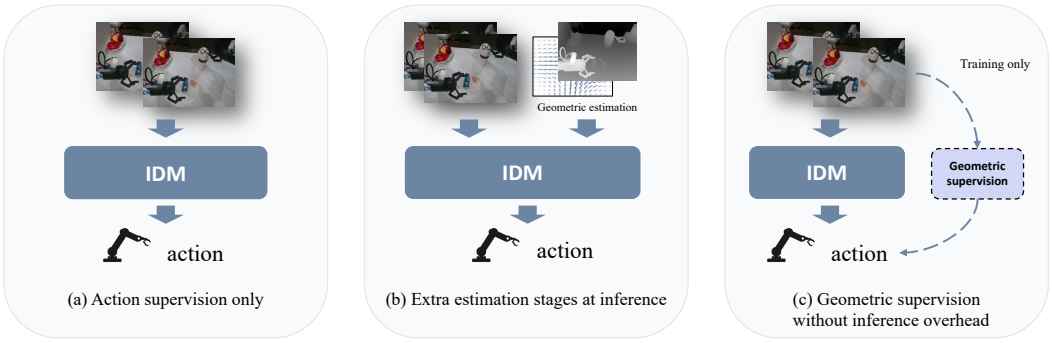


Figure 1: Comparison of IDM paradigms. (a) Direct IDM lacks explicit geometric grounding. (b) Geometric-Interface IDM introduces auxiliary estimation stages at inference. (c) GeoIDM (Ours) enforces training-time geometric consistency while retaining direct image-to-action inference.

verse dynamics in visual geometry during training, while preserving a simple, direct image-to-action mapping at inference?

To address this challenge, we introduce **GeoIDM**, a **Geo**metry-supervised **IDM** that connects predicted end-effector motion to its image-space projections and uses frozen dense geometric features (Wang et al., 2025) to assess consistency with the observed visual transition (Figure 1c). Specifically, during training, predicted actions are accumulated into a 3D displacement and reprojected into the future image via camera calibration. A local contrastive loss then matches the geometric feature at the projected location against its reference correspondence, providing a spatial learning signal that flows back through the predicted actions. A reliability-aware gate suppresses static or invalid projections. At inference, the geometric supervision branch is removed entirely, and the model retains the standard direct mapping from images to actions. We evaluate action prediction on AgiBot World (Bu et al., 2025) and reference-video replay in LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2025), complemented by an ablation of geometric supervision on a Franka robot.

We summarize our contributions as follows:

- We propose GeoIDM, an IDM that uses geometric supervision during training to improve out-of-domain generalization while preserving its standard visual inputs and action outputs at inference.
- We introduce an action-reprojected geometric objective that connects predicted robot motion to dense geometric features through a local contrastive loss, complemented by a reliability-aware gate to filter out static or invalid projections.
- Experiments across multiple datasets demonstrate lower translation errors and higher replay success on unseen tasks. A physical robot ablation further supports the benefit of geometric supervision.

2 Related Work

2.1 Video Models for Robot Control

Video generation models are a promising basis for robot policies because they capture motion, interaction, and task progression from large-scale video data. Joint video–action models predict future observations and actions within a shared framework (Ye et al., 2026; MotuBrain Team et al., 2026; Zhang et al., 2026b; Wang et al., 2026; Guo et al., 2026). These methods differ in visual-action tokenization and causal prediction, but all use future modeling to support action generation. Although this may align imagined dynamics with control, action adaptation remains coupled to the backbone and ultimately requires paired robot data for action grounding. Moreover, joint models may trade predictive generalization for precise reactive control under distribution shift (Feng et al., 2026).

Decoupled approaches instead pair an embodiment-agnostic video planner with an embodiment-specific IDM, preserving visual priors while allowing independent controller adaptation (Feng et al.,

2025a;b; Li et al., 2026b). Existing work improves task-relevant planning (Huang et al., 2024), real-time closed-loop prediction (Feng et al., 2025b), and latent-space action interfaces (Pai et al., 2025; Li et al., 2026c). Masked visual actions provides a related pixel-space interface for forward and inverse modeling (Alzayer et al., 2026). Hybrid methods distill generated futures and IDM pseudo-actions into a compact WAM (Qiu et al., 2026), or combine video-based planning with a reactive controller (Zhang et al., 2026c).

This modularity shifts much of the generalization burden to the IDM, which is usually trained on limited embodiment-specific data. It must recover actions from transitions affected by novel scenes, viewpoints, occlusion, and generation artifacts. Thus, a semantically plausible visual future may still yield inaccurate commands, motivating closer study of how IDMs represent and recover motion.

2.2 Inverse Dynamics Models

Within decoupled video-based policies, action recovery is commonly learned as direct regression from visual transitions. The central difficulty is not merely detecting motion, but isolating embodiment-induced motion from nuisance appearance changes: large background variations may be action-irrelevant, whereas subtle end-effector displacements can imply substantially different controls. Existing methods reduce this ambiguity by suppressing irrelevant regions and strengthening spatiotemporal or direction-sensitive features (Feng et al., 2025a;b; Huang et al., 2024; Li et al., 2026a), while task-agnostic exploration broadens coverage of the embodiment's feasible action space (Tan et al., 2025). These strategies improve robustness, but leave the correspondence between predicted robot motion and observed visual changes largely implicit.

A complementary line introduces structured motion representations to bridge visual prediction and control. Some methods require explicit geometric interfaces at inference, encoding motion through 2D flow or 3D point trajectories (Li et al., 2026b; Mi et al., 2026; Tong et al., 2026); errors in tracking, depth, or reconstruction can consequently propagate to control. Others use geometric modalities as pretraining supervision but remove the corresponding decoders at inference, retaining a direct image-to-action interface (Li et al., 2026c). A related approach uses affordance and depth supervision to shape internal video features before injecting them into the action branch, keeping the lightweight decoders during inference (Zhang et al., 2026a). GeoIDM differs from these representation-level strategies in where geometric supervision is applied: rather than predicting or consuming explicit geometric representations as an action interface, GeoIDM places the predicted action itself inside a differentiable geometric consistency loop during training, comparing reprojected motion against frozen geometric features (Wang et al., 2025) through a local contrastive objective, and removes the supervision branch entirely at inference.

3 Method

We introduce a latent geometric consistency objective that grounds predicted robot actions in temporally aligned visual representations. As illustrated in Figure 2, given two observations separated by an action horizon, I_t and I_{t+H}, our model extracts geometry-aware visual tokens via a frozen visual geometry encoder. The predicted action chunk is converted into a 3D end-effector displacement and reprojected into the future image, where a local contrastive objective matches the sampled feature against its reference correspondence. All equations apply independently to both robot arms, omitting batch and arm indices for clarity.

3.1 Geometry-Aware Visual Representation

We jointly encode observations with the pretrained VGGT aggregator (Wang et al., 2025) $\mathcal{A}$, yielding patch tokens $[\mathbf{Z}_t, \mathbf{Z}_{t+H}] = \mathcal{A}(I_t, I_{t+H})$ with $\mathbf{Z}_s \in \mathbb{R}^{N_p \times C_v}$ for $s \in \{t, t+H\}$. Discarding auxiliary camera and register tokens, the frozen encoder establishes cross-frame correspondences through alternating frame-wise and global attention. To reduce the token budget for the action model, a lightweight compressor maps patch tokens to a compact representation: $\mathbf{P}_s = \mathcal{C}(\mathbf{Z}_s; \mathbf{Q}_{\text{learn}}) \in \mathbb{R}^{Q \times d}$, where $\mathbf{Q}_{\text{learn}} \in \mathbb{R}^{Q \times d}$ is a set of Q learnable queries and $\mathcal{C}$ applies cross-attention followed by an FFN. To capture the visual transition between observations, we compute the difference between compressed tokens:

$$\mathbf{F} = (\mathbf{P}_{t+H} - \mathbf{P}_t)\, \mathbf{W}_{\text{out}}, \qquad \mathbf{F} \in \mathbb{R}^{Q \times d_{\text{hid}}}. \tag{1}$$

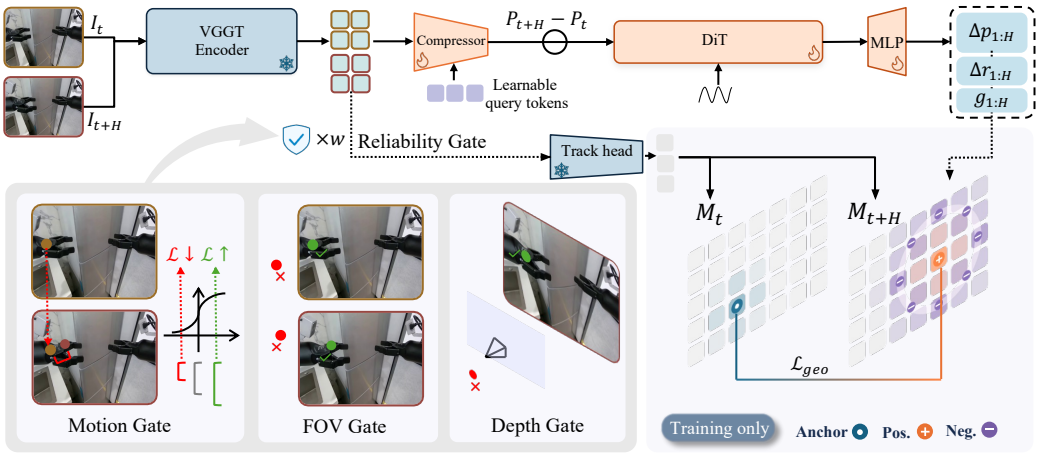


Figure 2: Overview of GeoIDM. P_t, P_{t+H} denote compressed visual tokens; M_t, M_{t+H} denote dense geometric features. Δp, Δr, and g are translation increments, rotation increments, and gripper states. Snowflakes and flames indicate frozen and trainable modules. Blue, orange, and purple markers denote the anchor, positive, and negative samples in Eq. (3). The reliability weight w in Eq. (2) modulates the training-only geometric loss in Eq. (5).

The difference in Eq. (1) can attenuate shared static content and emphasize motion. It conditions the action DiT along the upper path in Figure 2.

For the geometric supervision branch, we also reuse the pretrained DPT feature extractor within the VGGT tracking head. The original head combines this decoder with an iterative point tracker; GeoIDM uses only the decoder and skips the tracker. It consumes intermediate outputs from the same frozen encoder pass used for action conditioning, producing dense maps $\mathbf{M}_s \in \mathbb{R}^{C_f \times H_f \times W_f}$ for $s \in \{t, t+H\}$, with C_f channels and spatial resolution $H_f \times W_f$. The Track head block in Figure 2 denotes this frozen extractor. It requires no second backbone pass and supplies features rather than predicted tracking coordinates.

3.2 Action Reprojection and Reliability Gating

Conditioned on the visual difference in Eq. (1), the action DiT predicts an H-step chunk of translation increments, rotation increments, and gripper states. Let $\mathbf{p}_t^{\text{world}}$ denote the current end-effector position in world coordinates. We accumulate the predicted translations to obtain its future position: $\hat{\mathbf{p}}_{t+H}^{\text{world}} = \mathbf{p}_t^{\text{world}} + \sum_{k=1}^{H} \Delta\mathbf{p}_k$.

Using the calibrated camera at each time step, we project the current and predicted future positions into their respective images and rescale the coordinates to the $H_f \times W_f$ feature maps, yielding $\mathbf{u}_t$ and $\hat{\mathbf{u}}_{t+H}$. Projecting the ground-truth future position in the same way gives $\mathbf{u}_{t+H}^{\text{gt}}$, with image displacement $\Delta\mathbf{u}_{\text{gt}} = \mathbf{u}_{t+H}^{\text{gt}} - \mathbf{u}_t$. We denote the predicted future point's depth in the future camera frame by $\hat{Z}_c$ and the valid coordinate domain of the feature maps by Ω. Using the camera calibration at each time step accounts for camera motion.

Small image displacements provide limited motion information, while points behind the camera or outside Ω cannot provide valid feature samples. The Motion, FOV, and Depth gates in Figure 2 address these cases. With a motion threshold τ_g measured in feature-map pixels and a positive depth threshold ϵ measured in meters, they define the reliability weight:

$$w = \sigma\left(\|\Delta\mathbf{u}_{\text{gt}}\|_2 - \tau_g\right) \cdot \mathbb{1}[\hat{Z}_c > \epsilon] \cdot \mathbb{1}[\hat{\mathbf{u}}_{t+H} \in \Omega]. \tag{2}$$

Here, σ is the sigmoid function and $\mathbb{1}$ is an indicator. The first term smoothly downweights small motions, and the remaining terms exclude invalid projections. Computing the motion term from ground-truth displacement keeps it independent of the predicted action magnitude and requires no external tracking annotations.

3.3 GEOMETRIC CONSISTENCY OBJECTIVE

Using the dense maps and projected coordinates defined above, we construct the features for geometric supervision. Let $\tilde{\phi}(\mathbf{M}, \mathbf{u})$ denote ℓ_2-normalized bilinear sampling from $\mathbf{M}$ at continuous coordinate $\mathbf{u}$. Although the feature maps are frozen, differentiable sampling allows the loss to update predicted actions through their projected coordinates. The blue, orange, and purple markers in Figure 2 correspond to the anchor, positive, and local negatives below:

$$\mathbf{z}_{\text{anc}} = \tilde{\phi}\left(\mathbf{M}_t, \text{sg}[\mathbf{u}_t]\right), \quad \mathbf{z}_{+} = \tilde{\phi}\left(\mathbf{M}_{t+H}, \hat{\mathbf{u}}_{t+H}\right), \quad \mathbf{z}_{-,j} = \tilde{\phi}\left(\mathbf{M}_{t+H}, \text{sg}[\hat{\mathbf{u}}_{t+H}] + \boldsymbol{\delta}_j\right), \tag{3}$$

where $\text{sg}[\cdot]$ denotes stop-gradient, and negatives are sampled along a deterministic ring $\boldsymbol{\delta}_j = \rho[\cos\theta_j, \sin\theta_j]^\top$ with $\theta_j = 2\pi j / N_{\text{neg}}$ for $j = 0, \ldots, N_{\text{neg}} - 1$ where ρ is the radius in feature-map pixels and N_{neg} is the number of negative samples. Applying stop-gradient to the anchor and negative ring centers isolates loss gradient flow exclusively through the positive sampling coordinate $\hat{\mathbf{u}}_{t+H}$, directly optimizing predicted robot motion.

Using these representations, we compute a local InfoNCE contrastive loss:

$$\ell_{\text{NCE}} = -\log \frac{\exp\left(\mathbf{z}_{\text{anc}}^\top \mathbf{z}_{+}/\tau\right)}{\exp\left(\mathbf{z}_{\text{anc}}^\top \mathbf{z}_{+}/\tau\right) + \sum_{j=0}^{N_{\text{neg}}-1} \exp\left(\mathbf{z}_{\text{anc}}^\top \mathbf{z}_{-,j}/\tau\right)}, \tag{4}$$

where τ is the temperature parameter. Local negatives encourage the feature at the predicted projection to match the anchor more closely than nearby alternatives. Appendix B reports a spatial similarity probe of the frozen features that motivates this relative comparison.

We combine the local loss in Eq. (4) with the reliability gate in Eq. (2):

$$\mathcal{L}_{\text{geo}} = \lambda_{\text{geo}} \frac{\sum_{b=1}^{B} w_b \, \ell_{\text{NCE}}^b}{\max\left(\sum_{b=1}^{B} w_b, \, 1\right)}, \tag{5}$$

where λ_{geo} controls the geometric loss weight, B is the mini-batch size, and b indexes its samples. The denominator normalizes by the total reliability weight and is clamped below by one for numerical stability. As shown in the lower branch of Figure 2, w weights only the auxiliary geometric loss; it does not gate the action features or predictions.

During training, $\mathcal{L}_{\text{geo}}$ is optimized jointly with the flow-matching action loss and cumulative displacement regression, grounding coordinate predictions in dense visual geometry without requiring extra geometric annotations or tracking at inference. Parameter values and training configurations are reported in Section 4.1 and Appendix A.1.

4 EXPERIMENTS

We first compare GeoIDM with existing methods on action prediction accuracy and simulation replay. We then examine individual components through ablations on AgiBot, validate geometric supervision on a physical robot, and analyze the geometric objective.

4.1 EXPERIMENTAL SETUP

Datasets and evaluation splits. For action prediction and simulation replay, we define the evaluation domain by task identity. In-domain evaluation uses held-out episodes of tasks seen during training, whereas out-of-domain evaluation uses tasks excluded entirely from training.

For action prediction, we use AgiBot World (Bu et al., 2025). The training split contains 22,824 episodes across 29 tasks. In-domain evaluation draws from a pool of 7,021 held-out episodes from four of these tasks, while out-of-domain evaluation covers six additional tasks. To balance tasks, we sample 40 of the dataset's predefined segments per task from each evaluation split.

For simulation replay, we use LIBERO-90 (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2025). LIBERO-90 comprises 90 tasks, of which 72 are used for training with 40 episodes per task. In-domain evaluation uses 10 held-out episodes from each training task, while out-of-domain evaluation covers the remaining 18 tasks. For RoboTwin 2.0, we train on 40 of the 50 tasks using 40

episodes per task. In-domain evaluation uses 10 held-out episodes from each of 20 selected training tasks, while out-of-domain evaluation covers the remaining 10 tasks.

For the physical ablation of geometric supervision, we collect 1,000 demonstration trajectories on a Franka Emika robot for training. To evaluate open-loop replay capability, we test on three representative manipulation tasks: opening, clicking, and grasping. For each task, we conduct 10 evaluation trials from reset initial configurations, yielding a total of 30 physical trials per evaluated model.

Baselines. We compare GeoIDM with RoboDreamer (Zhou et al., 2024), Vidar (Feng et al., 2025a), AnyPos (Tan et al., 2025), VERA (Li et al., 2026b), and DreamGen (Jang et al., 2025). We retrain all compared methods on the training splits defined above, since their original training datasets differ from ours. We follow each method's original architecture and training configuration as closely as possible, making only the adaptations required by the target datasets, such as adjusting the action output dimensionality.

Implementation details and controls. GeoIDM uses an action horizon of $H = 16$, $Q = 16$ compressed queries, and a geometric loss weight of $\lambda_{\text{geo}} = 0.1$ by default. The pretrained VGGT encoder and the DPT feature extractor from its tracking head remain frozen; only the query compressor and action DiT are optimized. On AgiBot, we train for 80,000 optimization steps with a batch size of 128 per GPU on eight NVIDIA H200 GPUs, requiring approximately 46 hours. Geometric supervision hyperparameters and dataset-specific training configurations are collected in Appendix A.1.

Evaluation protocols. For action prediction evaluation, predictions are converted back to physical units and aggregated over 16 steps according to each model's native action representation. Delta actions are accumulated before computing the error, whereas errors for absolute actions are averaged over the horizon. For bimanual models, the two arm errors are computed separately and then averaged. We report position error in millimeters, geodesic rotation error in degrees, and gripper error. Lower is better for all metrics.

For simulation replay, each method predicts action chunks from the same successful reference videos. The predicted chunks are concatenated and executed open-loop after resetting the simulator to the corresponding demonstration's initial state. All methods use the same action-chunk boundaries and controller settings. We report success rates averaged uniformly across tasks on the in-domain and out-of-domain splits.

For the physical ablation, the Franka robot replays the predicted action chunks open-loop given the reference observations. We report task-level success rates alongside the aggregate success across all 30 trials. Success is determined by task-specific physical criteria: *Opening* requires unlatching and pulling a cabinet or drawer open by at least $5\,\text{cm}$; *Clicking* requires depressing a target switch or button to trigger mechanical or electrical state actuation; and *Grasping* requires lifting the target object stably at least $5\,\text{cm}$ above the surface for more than two seconds without slippage.

4.2 Main Results

4.2.1 Action Prediction Accuracy

We first evaluate how accurately the models recover actions from visual observations on the two AgiBot splits.

Table 1 shows that GeoIDM achieves the lowest reported position and rotation errors among the evaluated methods on both splits. Relative to DreamGen, position error decreases from 53.70 to 19.39 mm on in-domain evaluation and from 105.96 to 74.95 mm on out-of-domain evaluation, corresponding to reductions of 63.9% and 29.3%. Rotation error falls to 3.16 and 3.58 degrees, respectively, a reduction of 24.0% and 57.1% from DreamGen. Interestingly, rotation error also decreases despite L_{geo} directly supervising only translational displacement. We hypothesize that the shared Action DiT representation allows geometric supervision on translation to indirectly regularize orientation prediction. Position error also remains substantially higher on unseen tasks than on unseen episodes of training tasks, highlighting the remaining difficulty of task transfer. The comparisons follow the protocol in Section 4.1, which evaluates accumulated delta actions at the end of the

Table 1: Action prediction errors on AgiBot ↓. In-domain evaluation uses unseen trajectories of training tasks; out-of-domain evaluation uses tasks excluded from training. Errors follow the aggregation protocol in Section 4.1.

	In-Domain			Out-of-Domain		
Method	Pos. (mm)	Rot. (deg)	Grip	Pos. (mm)	Rot. (deg)	Grip
RoboDreamer	53.04	5.77	0.121	101.71	13.53	0.151
Vidar	62.10	4.58	0.094	88.97	11.49	0.124
AnyPos	49.34	4.42	0.087	90.20	10.63	0.119
VERA	71.11	7.75	0.102	94.64	16.61	0.121
DreamGen	53.70	4.16	0.028	105.96	8.34	0.034
Ours	19.39	3.16	0.115	74.95	3.58	0.080

Table 2: Task execution success rate (%) in simulation under reference-video guidance. Every method receives the same successful reference trajectories and is evaluated on paired simulator resets. In-Domain and Out-of-Domain denote evaluation on seen and unseen tasks, respectively.

	LIBERO		RoboTwin2.0	
Method	In-Domain	Out-of-Domain	In-Domain	Out-of-Domain
GT replay	89.7	90.0	96.0	100.0
RoboDreamer	57.5	3.9	14.0	1.0
Vidar	10.0	0.6	5.0	0.0
AnyPos	75.7	23.3	31.0	2.0
VERA	67.9	8.9	24.0	6.0
DreamGen	69.7	27.8	35.0	0.0
Ours	79.9	48.3	28.5	18.0

16-step horizon but averages absolute-action errors across the horizon. The ablations in Section 4.3 examine individual design choices separately.

Qualitative correspondence. Figure 3 compares the wrist projections induced by the predicted actions. Both methods are visualized using the same cosine-similarity maps from frozen VGGT features and identical image crops. In all three selected examples, GeoIDM produces projections closer to the ground truth and lower translation errors than the baseline. The negative-sampling rings shown for the baseline are included only for visualization. Additional visualization details are provided in Appendix A.2.

4.2.2 Task Execution in Simulation

We next assess whether the predicted actions can reproduce successful reference demonstrations under open-loop execution.

Table 2 shows that GeoIDM achieves the highest out-of-domain success rate among the learned methods on both benchmarks. On LIBERO, it reaches 79.9% in-domain success and 48.3% out-of-domain success, exceeding DreamGen by 10.2 and 20.5 percentage points, respectively. GeoIDM also outperforms the other evaluated methods on both LIBERO splits, with the larger gain over DreamGen occurring on tasks excluded from training.

RoboTwin 2.0 reveals a distinct contrast between in-domain and out-of-domain performance. On out-of-domain execution, GeoIDM achieves 18.0% success, substantially outperforming all competing baselines. In comparison, DreamGen collapses from 35.0% in-domain success to 0.0%, and other baselines drop below 6.0%. On in-domain evaluation, GeoIDM achieves 28.5% compared to DreamGen's 35.0%. This difference stems from how the models utilize training environments. Under in-domain evaluation on tasks seen during training, baseline models can exploit memorized scene appearance and task-specific priors to achieve higher success rates. However, when evaluated on novel tasks lacking these environmental priors, baseline policies suffer a precipitous drop to near-zero success. In contrast, by grounding action learning in physical geometric correspon-



Figure 3: Geometric correspondence in three selected AgiBot examples. Baseline denotes VGGT without $\mathcal{L}_{\text{geo}}$. Crosses and open diamonds mark current wrist anchors and future ground-truth projections, respectively. Values below the predictions report 2D reprojection error in pixels and translation error in millimeters at the end of the action horizon.

dences, GeoIDM is less susceptible to such appearance-dependent shortcuts, securing robust out-of-distribution transfer across unseen tasks. Nonetheless, out-of-domain success remains below ground-truth execution across both benchmarks, leaving room for further improvement under open-loop execution.

4.3 Ablation and Analysis

We examine component contributions to action prediction accuracy, test the effect of geometric supervision on physical execution, and probe the choice of contrastive objective.

Component ablation on AgiBot. Table 3 evaluates action differencing (Δa), latent feature differencing (Δz), geometric supervision ($\mathcal{L}_{\text{geo}}$), and reliability gating. Action differencing is present in all configurations, including the baseline; the remaining components are added incrementally under the same training configuration. Subscripts report relative error changes with respect to the baseline, computed as $100(E - E_{\text{base}})/E_{\text{base}}$, with arrows indicating decreases or increases.

Component contributions. Latent feature differencing reduces position, rotation, and gripper errors by 12.5%, 3.8%, and 15.9%, respectively, relative to the baseline. Adding ungated geometric supervision yields the lowest rotation and gripper errors, with reductions of 9.3% and 26.8% from the baseline. Its position error, however, rises from 74.33 to 77.25 mm compared with feature differencing alone, although it remains 9.1% below the baseline. Reliability gating brings position error down to 61.09 mm, the lowest among the evaluated configurations and a 28.1% reduction from the baseline. The full model also reduces gripper error by 17.1% and rotation error from 4.19 to 4.10 degrees, a 2.1% reduction relative to the baseline. These results indicate that the components do not benefit all action dimensions uniformly: gating improves translation accuracy but trades off some of the rotation and gripper gains of the ungated variant.

Table 3: Component ablation on a balanced 50% subset of AgiBot under a fixed computation budget. Checkmarks and crosses indicate enabled and disabled components. Subscripts show percentage error changes relative to the baseline; ↓ denotes a reduction and ↑ an increase. Lower errors are better, and best values are bold.

	Components				Prediction error ↓		
Configuration	Δa	Δz	$\mathcal{L}_{\text{geo}}$	Gate	Pos. (mm)	Rot. (deg)	Grip
Baseline	✓	×	×	×	84.99	4.19	0.082
+ Feature diff.	✓	✓	×	×	$74.33_{\downarrow 12.5\%}$	$4.03_{\downarrow 3.8\%}$	$0.069_{\downarrow 15.9\%}$
+ Geo. loss	✓	✓	✓	×	$77.25_{\downarrow 9.1\%}$	$\mathbf{3.80}_{\downarrow 9.3\%}$	$\mathbf{0.060}_{\downarrow 26.8\%}$
Full model	✓	✓	✓	✓	$\mathbf{61.09}_{\downarrow 28.1\%}$	$4.10_{\downarrow 2.1\%}$	$0.068_{\downarrow 17.1\%}$

Physical validation of geometric supervision. To complement the component ablation on AgiBot, we test whether geometric supervision improves open-loop execution on the Franka robot. Table 4 compares the full model with a variant trained without $\mathcal{L}_{\text{geo}}$ across opening, clicking, and grasping tasks. This variant differs from the first-row baseline in Table 3, which also omits latent feature differencing. Overall success increases from 3.3% to 20.0%, a gain of 16.7 percentage points. The full model succeeds in 6 of 30 trials, compared with 1 of 30 without geometric supervision, and improves success on all three tasks. These small-scale physical trials provide complementary evidence for the benefit of geometric supervision, rather than a comparison with external methods.

Table 4: Ablation of geometric supervision on the Franka robot. Success rates are reported in %, with 10 trials per task and 30 trials per model overall. Higher is better.

Method	Open	Click	Grasp	Overall
Ours w/o $\mathcal{L}_{\text{geo}}$	0	10	0	3.3
Ours	10	30	20	20.0

Failure analysis. Despite this improvement, overall physical success remains low. In opening, failures were predominantly caused by slight end-effector offsets at the handle contact boundary, where sub-centimeter errors prevented proper engagement. In grasping, failures primarily arose from subtle timing mismatches in gripper closure relative to object clearance. These observations suggest that geometric supervision alone does not resolve contact and gripper-timing errors under open-loop execution, motivating future evaluation with closed-loop visual or tactile feedback.

Local feature similarity and loss design. A spatial probe across 40 unseen AgiBot episodes shows that frozen dense features vary little near the correct correspondence: average cosine similarity is 0.9681 at the ground-truth location and 0.9666 at a 1-pixel offset. This limited local discrimination motivates comparing the projected feature with nearby alternatives through the InfoNCE objective in Eq. (4), rather than optimizing positive similarity alone. Appendix B provides the probe protocol and quantitative similarity statistics.

5 Conclusion

This paper presents GeoIDM, an inverse dynamics modeling framework that reconciles out-of-domain generalization with trajectory stability in robot manipulation. While relative delta action formulations help policies discard scene-specific coordinate shortcuts, they inevitably suffer from compounding kinematic drift under open-loop execution. GeoIDM resolves this tension by reprojecting predicted end-effector trajectories into dense feature maps via camera geometry, enforcing physical visual correspondence during training as a spatial anchor. Action prediction and simulation comparisons show improved action accuracy and replay success on unseen tasks, while a complementary physical ablation shows higher success with geometric supervision than without it. Together, these results support geometric consistency as a useful training signal for inverse dynamics without adding an inference-time geometry module.

AI USE STATEMENT

In accordance with the ICLR Generative AI Policy, generative AI tools (LLMs) were employed solely for language polishing, improving syntactic clarity, and formatting candidate bibliographic entries retrieved from academic search engines. All citations were manually verified for existence, accurate metadata, and relevance. Generative AI was not used for scientific ideation, algorithm design, theoretical formulation, data collection, or experimental execution, and the authors take full responsibility for the contents of this paper.

REPRODUCIBILITY STATEMENT

Our algorithmic framework and mathematical formulations are fully specified in Section 3.1–3.3, with complete implementation details, architectural choices, hyperparameters, and training schedules documented in Appendix A (Table 5 and Table 6). All evaluations rely on open academic benchmarks (AgiBot-World, LIBERO, and RoboTwin 2.0) under standardized splits detailed in Section 4.1. Anonymized source code, training configurations, and pretrained model checkpoints will be publicly released upon publication.

ETHICS STATEMENT

This paper studies vision-based inverse dynamics for robot manipulation. All real-world experiments on the Franka robot followed strict safety protocols, including bounded physical workspaces, joint torque limits, and continuous operator supervision with an accessible emergency stop. The offline and simulation benchmarks used are public academic assets that contain no human subjects, sensitive personal data, or personally identifiable information. While generalizable robot manipulation offers significant societal benefits in manufacturing and assistive robotics, open-loop execution carries inherent drift risks upon physical contact, and safety-critical deployments should incorporate closed-loop reactive feedback.

REFERENCES

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, et al. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Qiuxuan Feng, Jiale Yu, Jiaming Liu, Yueru Jia, Zhuangzhe Wu, Hao Chen, Zezhong Qian, Shuo Gu, Peng Jia, Siwei Ma, et al. Harmowam: Harmonizing generalizable and precise manipulation via adaptive world action models. *arXiv preprint arXiv:2605.10942*, 2026.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025a.

Yao Feng, Chendong Xiang, Xinyi Mao, Hengkai Tan, Zuyue Zhang, Shuhe Huang, Kaiwen Zheng, Haitian Liu, Hang Su, and Jun Zhu. Vidarc: Embodied video diffusion model for closed-loop control. *arXiv preprint arXiv:2512.17661*, 2025b.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Shuaiyi Huang, Mara Levy, Zhenyu Jiang, Anima Anandkumar, Yuke Zhu, Linxi Fan, De-An Huang, and Abhinav Shrivastava. Ardup: Active region video diffusion for universal policies. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8465–8472. IEEE, 2024.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Kerui Li, Zhe Jing, Xiaofeng Wang, Zheng Zhu, Yukun Zhou, Guan Huang, Dongze Li, Qingkai Yang, and Huaibo Huang. Stableidm: Stabilizing inverse dynamics model against manipulator truncation via spatio-temporal refinement. *arXiv preprint arXiv:2604.17887*, 2026a.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning video models into generalist robot policies. *arXiv preprint arXiv:2605.27817*, 2026b.

Yajie Li, Bozhou Zhang, Chun Gu, Zipei Ma, Jiahui Zhang, Jiankang Deng, Xiatian Zhu, and Li Zhang. From imagined futures to executable actions: Mixture of latent actions for robot manipulation. *arXiv preprint arXiv:2605.12167*, 2026c.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Weishi Mi, Yong Bao, Xiaowei Chi, Xiaozhu Ju, Zhiyuan Qin, Kuangzhi Ge, Kai Tang, Peidong Jia, Shanghang Zhang, and Jian Tang. Tc-idm: Grounding video generation for executable zero-shot robot motion. *arXiv preprint arXiv:2601.18323*, 2026.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

Chenhao Qiu, Ruixiang Wang, Runyi Zhao, Sixu Lin, Songen Gu, Shufeng Nan, Guiliang Liu, Kui Jia, Yanwei Fu, and Simo Wu. Vid2wam: Distilling video diffusion priors into world action models. *arXiv preprint arXiv:2608.08558*, 2026.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. Anypos: Automated task-agnostic actions for bimanual manipulation. *arXiv preprint arXiv:2507.12768*, 2025.

Mutian Tong, Han Jiang, Qiao Feng, Lingjie Liu, and Jiatao Gu. Pointaction: 3d points as universal action representations for robot control. *arXiv preprint arXiv:2606.03943*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. Repwam: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Mengqi Zhang, Sahil Khose, Simar Kareer, Yuchen Song, Unnat Jain, and Judy Hoffman. Deva: Decoupled video-action model with physical guidance for robot policy learning. *arXiv preprint arXiv:2607.24159*, 2026a.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, et al. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026b.

Zhongru Zhang, Chenghan Yang, Qingzhou Lu, Yanjiang Guo, Jianke Zhang, Yucheng Hu, and Jianyu Chen. Veo-act: How far can frontier video models advance generalizable robot manipulation? *arXiv preprint arXiv:2604.04502*, 2026c.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhou24f.html.

A Additional Implementation Details

A.1 GeoIDM Training Configuration

Table 5 collects the default model and geometric supervision settings. Table 6 lists the training schedule and computational resources for each dataset. The pretrained VGGT encoder and its DPT feature extractor remain frozen, while the query compressor and action DiT are optimized.

Table 5: Default model and geometric supervision hyperparameters for GeoIDM. Pixel distances are measured in feature-map coordinates.

Setting	Symbol	Default value
Action horizon	H	16 steps
Compressed queries	Q	16
Geometric loss weight	λ_{geo}	0.1
Contrastive temperature	τ	0.07
Negative samples per anchor	N_{neg}	8
Negative sampling radius	ρ	12 pixels
Motion gate threshold	τ_g	2 pixels
Depth validity threshold	ϵ	10^{-2} m

Table 6: GeoIDM training configurations. Batch size is reported per GPU; the effective batch size per optimizer update also depends on gradient accumulation.

Setting	AgiBot	LIBERO-90	RoboTwin 2.0
GPU type	NVIDIA H200	NVIDIA H200	NVIDIA H200
Number of GPUs	8	8	8
Batch size per GPU	128	128	128
Effective batch per update	1,024	1,024	1,024
Optimization steps	80,000	40,000	15,000
Training time	$\approx$ 46 hours	$\approx$ 6.3 hours	$\approx$ 2.9 hours

A.2 Visualization Details

Pixel distances in Figure 3 are measured in the original 640×480 image coordinates. The rings visualize eight local negative samples at a radius of 30 display pixels. This radius is expressed in display coordinates, whereas the training radius $\rho = 12$ is defined in feature-map coordinates. The baseline rings are shown only for visual comparison.

B Spatial Similarity Probe of Geometric Features

Table 7: Spatial probe diagnostic of frozen dense feature similarity across 40 unseen episodes. Margins and finite-difference gradient magnitudes are measured relative to the ground-truth correspondence (Offset 0 px).

Displacement Offset (px)	Cosine Similarity	Margin	$\mathrm{d(sim)/dpx}$
0 (Ground Truth)	0.9681	–	–
1	0.9666	0.0014	0.0015
8	0.9156	0.0525	0.0091
32	0.6209	0.3472	0.0131
Random Background	0.2068	0.7613	–

Probe protocol and setup. To characterize the spatial sensitivity of the features used for geometric supervision, we conduct a probe on frozen dense feature maps across 40 unseen validation episodes from the AgiBot benchmark. For each ground-truth correspondence pair $(\mathbf{u}, \mathbf{u}')$, we evaluate the cosine similarity between the anchor feature $\mathbf{f}_t(\mathbf{u})$ and the candidate feature $\mathbf{f}_{t+H}(\mathbf{u}' + \boldsymbol{\delta})$ at

various displacement offsets $\boldsymbol{\delta} \in \mathbb{R}^2$ with $\|\boldsymbol{\delta}\|_2 \in \{0, 1, 8, 32\}$ pixels, alongside randomly sampled background tokens. We measure both the similarity margin relative to the ground truth, defined as $\Delta_{\mathrm{sim}} = \mathrm{sim}_0 - \mathrm{sim}_\delta$, and the empirical spatial gradient magnitude $\mathrm{d}(\mathrm{sim})/\mathrm{dpx}$ via finite differences. The diagnostic results are summarized in Table 7.

Local similarity plateau in dense features. Table 7 shows strong local spatial correlation in the frozen dense features. Cosine similarity is 0.9681 at the correct correspondence and 0.9666 at a 1-pixel offset, with a reported margin of 0.0014. The finite-difference gradient magnitude is 0.0015 at 1 px and 0.0131 at 32 px. These measurements suggest that raw cosine similarity has limited sensitivity to small image displacements near a correct match. The probe characterizes frozen feature similarity; it does not directly compare the performance of different training objectives.