ACTIONS NEED NOT ALIGN: PREDICTIVE CONSEQUENCE-AWARE TRANSFER ACROSS EMBODIMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from heterogeneous robot experience promises more data-efficient and reusable manipulation policies, yet incompatible native action spaces make cross-embodiment sharing difficult. Existing approaches often address this heterogeneity by making actions comparable through shared or explicitly aligned action representations. We instead explore whether transfer can be organized around the predictive consequences of native actions without explicitly aligning their control representations. We formulate cross-embodiment learning as an **abstraction-dynamics-grounding** problem and introduce **Predictive Dynamics with Embodiment Grounding (PDEG)**. PDEG preserves each robot's native action space: embodiment-specific motion adapters expose heterogeneous trajectories to shared temporal abstraction and predictive modeling, while embodiment-conditioned queries retrieve control-relevant information from the shared predictive representation and ground it back into native controls. By sharing predictive computation rather than an explicitly aligned low-level action code, PDEG requires neither predefined joint correspondence, paired cross-embodiment demonstrations, cross-robot retargeting, nor a hand-designed common geometric action space. Experiments across three embodiments and three real-world manipulation tasks show positive cross-embodiment transfer under fixed target-robot data budgets.

1 INTRODUCTION

Pooling demonstrations from heterogeneous robots promises more data-efficient and reusable manipulation policies, but raises a fundamental question: *what should be shared when robots accomplish similar tasks through incompatible native action spaces?* Generalist policies and datasets have demonstrated the potential of sharing model capacity across diverse platforms (Open X-Embodiment Collaboration, 2024; Ghosh et al., 2024; Doshi et al., 2025). However, their control interfaces may differ in terms of dimensionality, joint semantics, kinematic coupling, actuation range, and end-effector articulation. These differences are particularly consequential in dexterous manipulation, where a coarse common control interface may discard the fine-grained finger coordination required for precise contacts.

Existing cross-embodiment methods address this challenge differently. Shared-policy approaches reuse model capacity across heterogeneous robots while retaining embodiment-specific interfaces (Doshi et al., 2025). Others construct universal, latent, kinematic, geometric, or functionally aligned action representations to facilitate sharing (Zheng et al., 2025; Bu et al., 2025; Bauer et al., 2026; Jiang et al., 2026; Zhang et al., 2026). This raises a question: *Can cross-embodiment transfer be organized around the predicted consequences of native actions without explicitly aligning their representations?*

We hypothesize that transferable action information can be characterized by task-relevant world changes that actions induce, rather than by correspondence between native control coordinates. Different robots may execute distinct joint trajectories while participating in similar interaction progressions, such as approaching an object, establishing contact, transporting it toward a goal, and releasing it. Neither their trajectories nor their latent representations need to be morphology-invariant.

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.

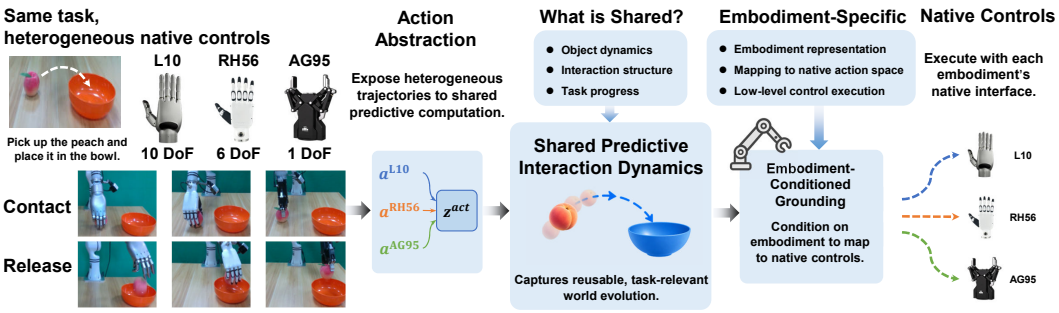


Figure 1: **Where should cross-embodiment sharing occur?** Different end effectors accomplish the same task using heterogeneous native controls. PDEG organizes sharing around predictive interaction computation rather than explicit action correspondence, while retaining embodiment-specific grounding for native execution. RH56 abbreviates RH56DFX in all figures.

Instead, we seek reusable predictive computation that relates heterogeneous native actions to subsequent visual world evolution.

This motivates an abstraction-dynamics-grounding formulation (Fig. 1). Native trajectories are exposed to shared predictive modeling through temporal abstraction, whereas the embodiment-specific physical context is retained to ground the predictive information into executable controls. Rather than eliminating embodiment information, we separate shared computation from the interfaces required for native execution.

We introduce **Predictive Dynamics with Embodiment Grounding (PDEG)** to instantiate this principle. PDEG processes native joint-space trajectories through two complementary routes. A Morphology Encoder combines embodiment-specific motion adapters with shared temporal abstraction to expose trajectory information to predictive modeling. In parallel, an Embodiment Encoder retains private physical context for the executing robot. A shared masked autoregressive (MAR) backbone (Li et al., 2024), together with video diffusion, models future visual evolution, while an embodiment-conditioned World-to-Control Adapter (WCA) retrieves control-relevant information for native action generation. PDEG imposes no explicit cross-robot alignment objective on its action tokens and requires neither predefined joint correspondence nor a common geometric action space.

This distinction matters empirically: supporting multiple robots in one network does not establish useful transfer from heterogeneous experiences. We therefore treat *positive cross-embodiment transfer* as a first-class evaluation target. Holding the target robot's Stage-II demonstration budget fixed, we ask whether adding demonstrations from other embodiments improves target-robot performance. We further compare the corresponding Single-to-Multi gains against the Unified Video Action Model (UVA) (Li et al., 2025) and ablate the interfaces on both sides of the shared predictive core to examine how shared temporal abstraction and embodiment-conditioned grounding contribute to cross-embodiment learning.

In summary, we made the following three contributions:

- **Predictive consequence-based cross-embodiment formulation.** We formulate cross-embodiment imitation learning as an abstraction-dynamics-grounding problem, organizing transfer around shared predictive computation without explicit correspondence between native action spaces.
- **Shared temporal abstraction with private embodiment context.** We introduce Morphology and Embodiment Encoders to separate shared trajectory abstraction from embodiment-specific physical context. This exposes heterogeneous native trajectories to shared predictive modeling while retaining information required for native execution.
- **Embodiment-conditioned world-to-control grounding.** We introduce the World-to-Control Adapter (WCA), which uses private embodiment representations to retrieve control-relevant information from shared predictive world tokens. With shared high-capacity control computation and private native interfaces, heterogeneous robots can reuse predictive information while generating controls in their respective action spaces.

2 Related Work

Heterogeneous robot data and shared policies. Large-scale robot datasets have motivated policies that reuse model capacity across heterogeneous embodiments. Octo (Ghosh et al., 2024) trains a generalist manipulation policy on multi-robot data and adapts across observation and action interfaces. CrossFormer (Doshi et al., 2025) demonstrates shared-policy learning across manipulation, navigation, locomotion, and aerial platforms without manually aligning observation or action spaces. RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2025), and π_0 (Black et al., 2025) further scale language-conditioned robot policies, with the latter two trained on multi-robot data. Cross-domain methods such as XSkill (Xu et al., 2023), EgoMimic (Kareer et al., 2025), and Universal Manipulation Interface (UMI) (Chi et al., 2024) bridge human demonstrations and robot control. These works demonstrate the feasibility of heterogeneous data sharing, while PDEG focuses specifically on how predictive interaction computation can be shared when each embodiment retains its native control interface.

Action abstraction and cross-embodiment correspondence. A major line of work facilitates transfer through intermediate action representations. UniAct (Zheng et al., 2025) learns a Universal Action Space, UniVLA (Bu et al., 2025) derives task-centric latent actions, Latent Action Pretraining for general Action models (LAPA) (Ye et al., 2025) learns discrete latent actions from unlabeled videos, and LAD (Bauer et al., 2026) learns an aligned latent action space before policy learning. Recent dexterous-hand systems introduce stronger structural priors: XL-VLA (Jiang et al., 2026) uses forward kinematics and fingertip-geometry retargeting, while UniDex (Zhang et al., 2026) employs human-in-the-loop retargeting and a Function-Actuator-Aligned Space. PDEG takes a different route: its abstract action tokens are not required to form a globally equivalent control code, and no explicit cross-robot alignment objective is imposed on them. Instead, trajectory abstraction serves as an interface through which heterogeneous native actions participate in shared predictive computation. Detailed method-level distinctions are summarized in Appendix A.

Predictive world models and action-world interaction. Predictive models connect perception, dynamics, and control through latent planning (Hafner et al., 2019), text-guided video policies (Du et al., 2023), and learned latent actions (Bruce et al., 2024). DINO-WM (Zhou et al., 2025) predicts future visual representations for planning, while Unified World Models (Zhu et al., 2025) couples video and action diffusion and mimic-video (Pai et al., 2026) decodes actions from video-space predictive representations. UVA (Li et al., 2025) further uses masked video-action modeling to support multiple predictive and control objectives within a shared framework. Closely related cross-embodiment work includes LAC-WM (Huang et al., 2026), which performs world-model pretraining through a unified latent action space, and particle-based predictive modeling (He et al., 2026), which constructs a common geometric action interface from hand/object displacement. XEWorld (Chen et al., 2026) further shows that predictive accuracy does not necessarily imply morphology-independent world representations. PDEG therefore makes a narrower but distinct claim: neither native actions nor world representations are required to become cross-embodiment equivalent; instead, transfer is organized around shared predictive interaction computation together with embodiment-conditioned native control grounding.

3 Predictive Dynamics with Embodiment Grounding

PDEG preserves each embodiment's native control space while organizing cross-embodiment learning around shared predictive computation and embodiment-specific grounding. As shown in Fig. 2, native trajectories are processed through two complementary routes: the Morphology Encoder exposes heterogeneous trajectories to shared temporal and predictive computation, while the Embodiment Encoder retains private physical context. A shared MAR backbone integrates visual and robot representations into predictive world tokens, from which the World-to-Control Adapter (WCA) retrieves control-relevant information using private embodiment representations to guide native action generation.

PDEG shares temporal, predictive, and control computation while preserving private physical context and coordinate-dependent interfaces, without requiring a common action space.

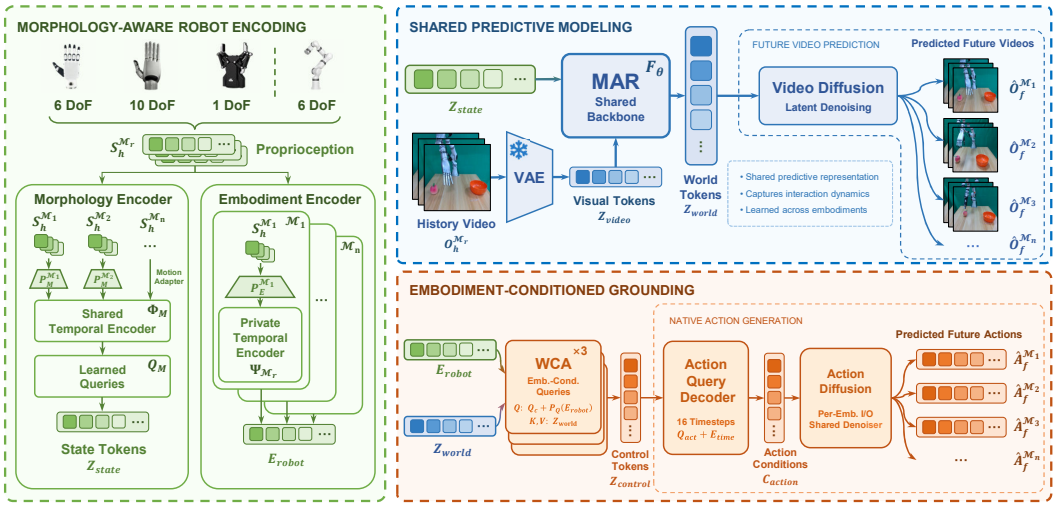


Figure 2: **Overview of PDEG.** Native proprioception is processed by complementary Morphology and Embodiment Encoders. Shared temporal abstraction supplies state tokens to the predictive backbone, while private embodiment context conditions WCA's control queries. WCA progressively retrieves control-relevant information from shared world tokens, and native action generation uses a shared denoiser with embodiment-specific interfaces.

3.1 Problem Formulation

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_R\}$ denote the set of robot embodiments. Embodiment $\mathcal{M}_r$ has a native proprioceptive space $\mathcal{S}_{\mathcal{M}_r} \subseteq \mathbb{R}^{d_r^s}$ and native action space $\mathcal{A}_{\mathcal{M}_r} \subseteq \mathbb{R}^{d_r^a}$, where dimensionality and coordinate semantics may differ across robots. We assume no explicit mapping between $\mathcal{A}_{\mathcal{M}_i}$ and $\mathcal{A}_{\mathcal{M}_j}$ and do not require paired trajectories across embodiments.

For embodiment $\mathcal{M}_r$, let $O_h^{\mathcal{M}_r}$ and $S_h^{\mathcal{M}_r}$ denote the historical visual observations and proprioceptive trajectory, respectively. As in action-chunking methods (Zhao et al., 2023), the policy predicts a future action chunk directly in the embodiment's native joint coordinates,

$$\pi^{\mathcal{M}_r} : \left(O_h^{\mathcal{M}_r}, S_h^{\mathcal{M}_r}\right) \mapsto A_f^{\mathcal{M}_r}, \qquad A_f^{\mathcal{M}_r} \in \mathbb{R}^{H_a \times d_r^a}, \tag{1}$$

where H_a denotes the action horizon.

We use native joint positions as proprioceptive context and anchor-relative joint displacements as prediction targets. Given the current joint configuration $q_t^{\mathcal{M}_r}$,

$$A_f^{\mathcal{M}_r} = \left[q_{t+1}^{\mathcal{M}_r} - q_t^{\mathcal{M}_r}, \ldots, q_{t+H_a}^{\mathcal{M}_r} - q_t^{\mathcal{M}_r}\right]. \tag{2}$$

Both arm and end-effector actions retain their native joint coordinates and dimensionality without geometric retargeting.

3.2 Shared Abstraction and Private Embodiment Context

Robot encoding combines shared trajectory abstraction with private embodiment context, enabling shared prediction without cross-robot coordinate alignment.

Morphology Encoder. For a native trajectory $X^{\mathcal{M}_r}$, the embodiment-specific Motion Adapter $P_M^{\mathcal{M}_r}$ first projects native coordinates into a common hidden width. The resulting sequence is processed by the temporal encoder Φ_M, whose parameters are shared across embodiments. A set of

learned queries Q_M then retrieves trajectory information from the temporal features:

$$\begin{aligned} \bar{X}^{\mathcal{M}_r} &= P_M^{\mathcal{M}_r}\big(X^{\mathcal{M}_r}\big), \\ H_M^{\mathcal{M}_r} &= \Phi_M\big(\bar{X}^{\mathcal{M}_r}\big), \\ \mathcal{Z}_M\big(X^{\mathcal{M}_r}\big) &= \mathrm{CrossAttn}\Big(Q_M, H_M^{\mathcal{M}_r}, H_M^{\mathcal{M}_r}\Big), \end{aligned} \tag{3}$$

where Q_M serves as the query and $H_M^{\mathcal{M}_r}$ provides both keys and values. The adapter $P_M^{\mathcal{M}_r}$ is embodiment-specific, whereas the temporal encoder Φ_M is shared.

Historical proprioception is encoded as $Z_{\mathrm{state}}^{\mathcal{M}_r} = \mathcal{Z}_M(S_h^{\mathcal{M}_r})$. When true future actions are available as predictive conditions during training, the same encoder produces $Z_{\mathrm{action}}^{\mathcal{M}_r} = \mathcal{Z}_M(A_f^{\mathcal{M}_r})$. No explicit cross-embodiment alignment objective is imposed on $Z_{\mathrm{action}}^{\mathcal{M}_r}$; the action tokens are learned only through their role in shared predictive computation.

Embodiment Encoder. Shared predictive modeling does not require embodiment-specific physical information to be discarded. PDEG therefore maintains a complementary private route. For embodiment $\mathcal{M}_r$, a private state adapter $P_E^{\mathcal{M}_r}$ followed by a private temporal encoder $\Psi_{\mathcal{M}_r}$ produces

$$E_{\mathrm{robot}}^{\mathcal{M}_r} = \Psi_{\mathcal{M}_r}\Big(P_E^{\mathcal{M}_r}\Big(S_h^{\mathcal{M}_r}\Big)\Big). \tag{4}$$

Unlike the shared temporal encoder Φ_M, both the adapter and temporal processing in this route are embodiment-specific. The two encoder outputs therefore serve complementary roles: $Z_{\mathrm{state}}^{\mathcal{M}_r}$ participates in shared predictive modeling, whereas $E_{\mathrm{robot}}^{\mathcal{M}_r}$ retains private physical context for embodiment-conditioned control grounding. Architectural details are provided in Appendix B.

3.3 Shared Predictive Interaction Dynamics

Historical visual observations are encoded by a frozen variational autoencoder (VAE) (Kingma & Welling, 2014; Rombach et al., 2022) into visual tokens $Z_{\mathrm{video}}^{\mathcal{M}_r}$. A shared MAR backbone F_θ (Li et al., 2024) integrates visual and robot-side tokens into predictive world tokens. To write the different training configurations compactly, let $\widetilde{Z}_{\mathrm{action}}^{\mathcal{M}_r}$ denote the action-slot input to MAR: it equals the encoded native action tokens $Z_{\mathrm{action}}^{\mathcal{M}_r}$ when true future actions are provided as predictive conditions, and otherwise consists of learned fake-action tokens. Then

$$Z_{\mathrm{world}} = F_\theta\Big(Z_{\mathrm{video}}^{\mathcal{M}_r}, Z_{\mathrm{state}}^{\mathcal{M}_r}, \widetilde{Z}_{\mathrm{action}}^{\mathcal{M}_r}\Big). \tag{5}$$

The resulting Z_{world} is shared by the predictive and control pathways. When future-video prediction is active, the same world tokens condition the video-diffusion branch to reconstruct masked future visual content, providing direct predictive supervision on subsequent visual world evolution.

PDEG shares predictive computation rather than enforcing latent invariance. World tokens may retain embodiment-specific information while modeling future task-relevant evolution from visual, proprioceptive, and native action inputs.

3.4 Embodiment-Conditioned World-to-Control Grounding

Shared predictive information must ultimately be interpreted according to the executing embodiment. WCA therefore performs embodiment-conditioned retrieval through a sequence of three control-query refinement blocks.

Let Q_c denote the learned base control queries and $R^{\mathcal{M}_r} = P_Q(E_{\mathrm{robot}}^{\mathcal{M}_r})$ the projected embodiment representation. Starting from $Q_0^{\mathcal{M}_r} = Q_c$, each WCA block injects the same embodiment representation into the current control queries:

$$\widetilde{Q}_i^{\mathcal{M}_r} = Q_i^{\mathcal{M}_r} + \alpha_i R^{\mathcal{M}_r}. \tag{6}$$

The conditioned queries are then refined by a WCA block that attends to the shared world tokens:

$$Q_{i+1}^{\mathcal{M}_r} = \mathrm{WCA}_i\left(\widetilde{Q}_i^{\mathcal{M}_r}, Z_{\mathrm{world}}\right), \qquad i = 0, 1, 2. \tag{7}$$

Conditioned queries attend to the unchanged Z_{world} as keys and values and are successively refined across WCA blocks.

The final representation is $Z_{\text{control}}^{\mathcal{M}_r} = \text{LN}_{\text{out}}(Q_3^{\mathcal{M}_r})$. WCA uses 256 control queries with hidden width 768. All three blocks have independent attention, feed-forward, and normalization parameters. Their detailed computations and initialization are provided in Appendix B.

3.5 Native Action Generation

The retrieved control tokens are converted into temporally organized conditions for the future action horizon by an Action Query Decoder (AQD),

$$C_{\text{action}}^{\mathcal{M}_r} = G_{\text{AQD}}\left(Q_{\text{act}} + E_{\text{time}}, Z_{\text{control}}^{\mathcal{M}_r}\right),$$

where Q_{act} contains H_a learned action queries and E_{time} provides explicit temporal embeddings. Consequently, each future action position receives a dedicated condition rather than sharing a single pooled control feature.

Native action generation follows conditional diffusion policy (Chi et al., 2023). Let $A_k^{\mathcal{M}_r}$ denote the noised native action chunk at diffusion step k, and let e_k denote the corresponding diffusion-step embedding. Each embodiment uses private input and output projections around a shared denoising trunk:

$$\hat{\epsilon}^{\mathcal{M}_r} = P_{\mathcal{M}_r}^{\text{out}}\left[\Omega_{\text{shared}}\left(P_{\mathcal{M}_r}^{\text{in}}\left(A_k^{\mathcal{M}_r}\right), e_k, C_{\text{action}}^{\mathcal{M}_r}\right)\right]. \tag{8}$$

The shared denoising trunk contains six adaptive layer normalization (AdaLN) (Peebles & Xie, 2023) residual blocks. Diffusion is performed only over the valid native action dimensions of the executing embodiment; padded dimensions are excluded from the physical action interpretation. Thus, high-capacity control computation is shared while coordinate-dependent input and output interfaces remain private.

3.6 Training Objectives and Inference

For optimization, we adopt the five masked video-action configurations introduced by UVA (Li et al., 2025): *Video Model*, *Dynamic Model*, *Policy Model*, *Inverse Model*, and *Full Dynamic Model*. These configurations determine which future video and action slots are available as conditions and which modality is predicted; historical proprioception is available throughout. Among them, only *Dynamic Model* provides the true future native action trajectory $A_f^{\mathcal{M}_r}$ as an input condition. The trajectory is encoded by the Morphology Encoder into $Z_{\text{action}}^{\mathcal{M}_r}$ before entering MAR. In the action-prediction configurations, future actions serve as prediction targets rather than encoder inputs. Therefore, no explicit cross-embodiment alignment objective is applied to the action tokens.

Stage I pretrains MAR and video diffusion using only *Video Model*, without action prediction or future-action conditioning. Stage II jointly trains prediction and native action generation across all five configurations, keeping both components trainable.

For mixed-embodiment batches, action losses are first averaged within each robot and then across the robots present in the batch,

$$\mathcal{L}_{\text{action}} = \frac{1}{|\mathcal{R}_{\mathcal{B}}|} \sum_{r \in \mathcal{R}_{\mathcal{B}}} \frac{1}{|\mathcal{B}_r|} \sum_{i \in \mathcal{B}_r} \ell_i^{\mathcal{M}_r}, \tag{9}$$

which weights each embodiment equally, regardless of its sample count.

At deployment, PDEG receives only historical visual observations $O_h^{\mathcal{M}_r}$ and proprioception $S_h^{\mathcal{M}_r}$. The Morphology Encoder produces $Z_{\text{state}}^{\mathcal{M}_r}$, the Embodiment Encoder produces $E_{\text{robot}}^{\mathcal{M}_r}$, and MAR forms Z_{world} using learned fake-action tokens in place of unavailable future actions. WCA retrieves $Z_{\text{control}}^{\mathcal{M}_r}$, AQD produces $C_{\text{action}}^{\mathcal{M}_r}$, and iterative diffusion generates $\hat{A}_f^{\mathcal{M}_r}$ in the target embodiment's native action space. Neither future video nor true future action is available as conditioning information, and the video-diffusion branch is not required for action execution.

4 EXPERIMENTS

We evaluate PDEG along four dimensions: real-world manipulation performance, cross-embodiment transfer under fixed target-robot data budgets, parameter scalability, and the contribution of its encoding and grounding interfaces. Quantitative comparisons are complemented by qualitative execution examples and future-video prediction visualizations.

4.1 EXPERIMENTAL SETUP

Robots and tasks. We evaluate three heterogeneous end effectors mounted on the same 6-degree-of-freedom (DoF) RealMan RM65 arm: LinkerHand L10 (10 controllable hand DoFs), Inspire Robots RH56DFX (abbreviated as RH56 in figures; 6 DoFs), and DH-Robotics AG95 (1 DoF). All three embodiments perform **Peach Placement**, **Cup Stacking**, and **Ball Pouring** under comparable conditions (nine settings; Fig. 3a), isolating end-effector and control-interface differences.

Data and observation pipeline. Each task-embodiment pair contains approximately 300 successful demonstrations, yielding approximately 2700 trajectories across the nine combinations. RGB observations are captured using a front-view Intel RealSense D435 and resized to 256×256. The policy uses four visual conditioning frames, a 16-step proprioceptive history, and a 16-step future action horizon. Further dataset and observation details are provided in Appendix D.

Baselines and controlled comparisons. We compare PDEG with UVA (Li et al., 2025) and an adapted multi-embodiment EgoMimic (Kareer et al., 2025). For each task, all UVA and PDEG variants share the same three-embodiment Stage-I checkpoint for MAR and video diffusion.

Single variants use only target-embodiment demonstrations during Stage II, whereas Multi variants additionally use data from the other two embodiments. Under matched target-robot budgets and initialization, these comparisons assess the benefit of heterogeneous experience, with UVA as an alternative shared-policy reference.

Parameter scalability. We compare active core parameter growth across embodiments, excluding the frozen shared VAE and inactive modules. Figure 3b shows that UVA adds 52.26M parameters per embodiment versus 8.32M for PDEG, a $6.29\times$ reduction in marginal growth through shared computation and lightweight native interfaces. Detailed accounting is provided in Appendix D.3.

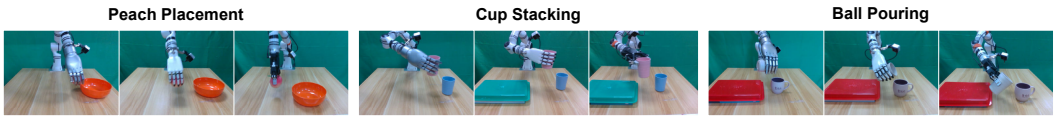
4.2 COMPARATIVE REAL-WORLD EVALUATION

We evaluate real-world manipulation performance across all nine task-embodiment settings and examine the incremental benefit of heterogeneous Stage-II demonstrations under fixed target-robot data budgets.

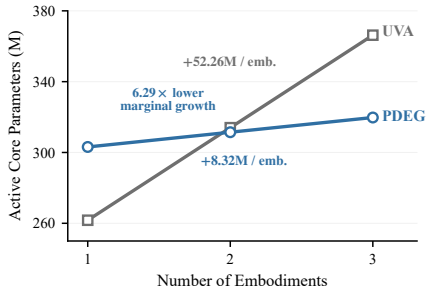
Table 1: **Real-world task success rate (%);** 50 physical rollouts per entry. Best results in each task-embodiment setting are bold.

	Peach Placement			Cup Stacking			Ball Pouring			Average
Method	L10	RH56DFX	AG95	L10	RH56DFX	AG95	L10	RH56DFX	AG95	Avg.
UVA-Single	78	82	82	64	60	54	70	68	64	69.1
UVA-Multi	86	84	90	66	64	60	72	76	66	73.8
EgoMimic-Multi	66	62	72	46	42	38	52	48	46	52.4
PDEG-Single	84	86	82	70	72	62	74	76	68	74.9
PDEG-Multi	**92**	**88**	**94**	**80**	**78**	**70**	**86**	**82**	**80**	**83.3**

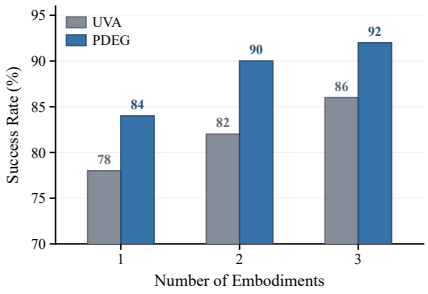
As shown in Table 1, PDEG-Multi leads all nine settings, averaging 83.3% versus 73.8% for UVA-Multi and 52.4% for EgoMimic-Multi. Its task-wise averages are 91.3%, 76.0%, and 82.7%, exceeding UVA-Multi by 4.7, 12.7, and 11.3 percentage points on Peach Placement, Cup Stacking, and Ball Pouring, respectively. The larger margins on the latter two tasks indicate that the advantage extends beyond Peach Placement. PDEG-Single also exceeds UVA-Single (74.9% versus 69.1%) under target-only Stage-II training.



(a) Real-world task-embodiment settings.



(b) Parameter scalability.



(c) Representative performance scaling.

Figure 3: **Real-world evaluation and embodiment scaling. (a)** The nine task-embodiment settings. **(b)** Active core parameter counts as the number of supported embodiments increases. **(c)** Peach Placement success rates with L10 as the target embodiment under a fixed target-robot Stage-II data budget. The two-embodiment setting additionally includes RH56DFX, while the three-embodiment setting includes all three robots.

Cross-embodiment control transfer. To quantify the benefit of heterogeneous Stage-II experience, we define the transfer gain for embodiment $\mathcal{M}_r$ and task t as

$$\Delta^{(\mathcal{M}_r,t)}_{\text{transfer}} = \text{SR}^{(\mathcal{M}_r,t)}_{\text{PDEG-Multi}} - \text{SR}^{(\mathcal{M}_r,t)}_{\text{PDEG-Single}}. \tag{10}$$

Under matched predictive initialization and target-robot Stage-II budgets, PDEG exhibits positive transfer in all nine settings, with average gains of 7.3, 8.0, and 10.0 percentage points on Peach Placement, Cup Stacking, and Ball Pouring, respectively. The corresponding UVA gains are 6.0, 4.0, and 4.0 points. Across all settings, PDEG improves average success by 8.4 points, compared with 4.7 points for UVA, with larger task-level transfer gains particularly evident on Cup Stacking and Ball Pouring. Although individual gains vary across embodiments, these results demonstrate a greater aggregate benefit from heterogeneous Stage-II experience under the evaluated protocol. The comparisons do not isolate the effects of additional training data and computation. Detailed success counts and confidence intervals are provided in Appendix D.

4.3 Qualitative Evaluation

We visualize future-video predictions on a held-out Cup Stacking trajectory collected with RH56DFX. Figure 4 compares ground-truth future observations with predictions from PDEG-Single and PDEG-Multi.

Both models receive identical visual and proprioceptive histories and predict fully masked future video under *Full Dynamic Model*, without ground-truth future observations or actions.

These examples illustrate the predicted visual evolution of the manipulation process. Additional qualitative predictions across all three embodiments are provided in Appendix D.5.

4.4 Ablation Studies

We examine the contribution of three components on Peach Placement across all three embodiments: shared temporal trajectory abstraction, WCA-based predictive information retrieval, and embodiment-conditioned queries. UVA-Multi is included as a reference multi-embodiment policy. All ablation variants are evaluated under the same task and embodiment settings.

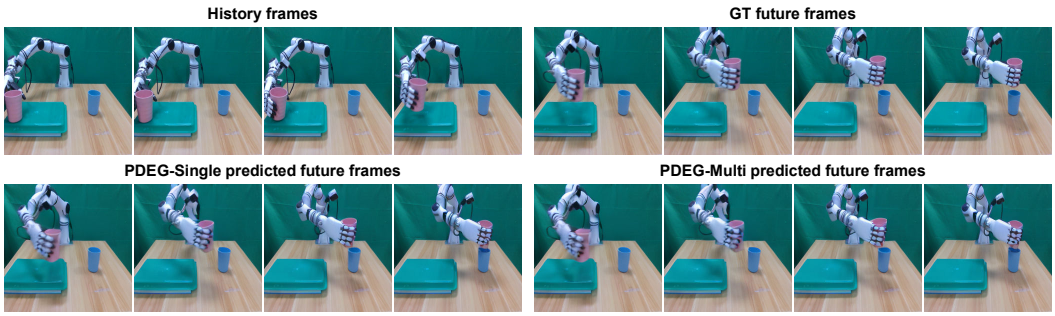


Figure 4: **Qualitative future-video prediction.** A held-out Cup Stacking trajectory with RH56DFX. The four groups show historical observations, ground-truth future frames, PDEG-Single predictions, and PDEG-Multi predictions. Both models receive identical historical inputs under the *Full Dynamic Model* configuration with fully masked future video.

Table 2: **Ablation results on Peach Placement.** Real-world success rates (%) across the three embodiments.

Variant	L10	RH56DFX	AG95	Avg.
Full PDEG	92	88	94	**91.3**
w/o WCA	88	84	90	87.3
w/o Shared Temporal Abstraction	88	86	86	86.7
w/o Emb. Conditioning	74	70	76	73.3
UVA-Multi	86	84	90	86.7

Shared temporal abstraction. Replacing the shared temporal encoder with embodiment-specific linear projection reduces average success by 4.7 percentage points, with decreases of 4, 2, and 8 points for L10, RH56DFX, and AG95, respectively. This result supports the contribution of shared temporal processing beyond dimensional compatibility alone.

World-to-control grounding. Removing WCA reduces average success by 4.0 percentage points, while disabling embodiment conditioning leads to a larger reduction of 18.0 points. These results indicate that both dedicated retrieval from predictive world tokens and embodiment-dependent query modulation contribute to native action generation, with the latter having a particularly pronounced effect in the evaluated settings.

5 Conclusion

We introduced Predictive Dynamics with Embodiment Grounding (PDEG), a cross-embodiment imitation-learning framework that organizes knowledge sharing around predictive interaction computation rather than explicit action-space alignment. By combining shared temporal abstraction, private embodiment context, and embodiment-conditioned world-to-control grounding, PDEG shares predictive and control computation while preserving native action spaces, without predefined joint correspondence, paired cross-embodiment demonstrations, or a hand-designed common geometric action space.

Across three embodiments and three real-world manipulation tasks, PDEG-Multi achieves an average success rate of 83.3%, compared with 73.8% for UVA-Multi and 52.4% for EgoMimic-Multi. Under fixed target-robot Stage-II data budgets, PDEG improves average success from 74.9% to 83.3%, yielding an 8.4-point transfer gain, compared with 4.7 points for UVA. Ablations on Peach Placement support the roles of shared temporal abstraction, WCA retrieval, and embodiment conditioning, while marginal parameter growth is 8.32M per embodiment, compared with 52.26M for UVA. Our evaluation is limited to heterogeneous end effectors on a shared arm; transfer across different arm kinematics, sensing configurations, workspace geometries, and unseen embodiments remains to be investigated.

Reproducibility Statement

We provide detailed architecture specifications in Appendix B, training configurations and optimization details in Appendix C, and dataset construction, evaluation protocols, parameter accounting, and additional experimental results in Appendix D. Anonymous implementation code is provided in the supplementary material.

AI Use Statement

Generative AI tools were used as auxiliary research tools for critical review, implementation support, and language and presentation refinement. The research problem, core methodological ideas, model architecture, and experimental objectives were conceived by the authors. AI tools were used to examine the feasibility and internal consistency of author-developed ideas, identify potential omissions or ambiguities, suggest additional validation checks, and assist with code debugging, literature organization, figure refinement, and language editing. All suggestions were critically evaluated by the authors, and all implementations, references, experimental results, and scientific conclusions were independently verified by the authors. The authors take full responsibility for the final work.

References

Erik Bauer, Elvis Nava, and Robert K. Katzschmann. Latent action diffusion for cross-embodiment manipulation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014.

Yixiang Chen, Jiabing Yang, Yuan Xu, Qisen Ma, Keji He, Peiyan Li, et al. XEWorld: Can action-conditioned world models generalize to unseen robot embodiments?, 2026. arXiv preprint arXiv:2608.05799.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.045.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 496–512. PMLR, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023. doi: 10.52202/075280-0403.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Zihao He, Bo Ai, Tongzhou Mu, Yulin Liu, Weikang Wan, Jiawei Fu, et al. Scaling cross-embodiment world models for dexterous manipulation. In *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. Accepted; proceedings details forthcoming.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Huang Huang, Sriram Yenamandra, Arjun Majumdar, Elie Aljalbout, Tushar Nagarajan, Tsung-Yen Yang, et al. Cross-embodiment robot foundation world models with latent actions. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-hand latent representation for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13496–13507, June 2026.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233, 2025. doi: 10.1109/ICRA55743.2025.11127989.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.074.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 56424–56445, 2024. doi: 10.52202/079017-1797.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Jonas Pai, Liam Achenbach, Oliver Sanchez, Stefanos Charalambous, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.077.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2018. arXiv preprint arXiv:1812.01717.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3536–3555. PMLR, 2023.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Gu Zhang, Qicheng Xu, Haozhe Zhang, Jianhan Ma, Long He, Yiming Bao, et al. UniDex: A robot foundation suite for universal dexterous hand control from egocentric human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1841–1852, June 2026.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22508–22519, June 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.015.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.

A Extended Related Work

Cross-embodiment policies and heterogeneous robot data. Large-scale robot datasets have motivated policies that reuse model capacity across heterogeneous embodiments. Octo (Ghosh et al., 2024) trains a generalist manipulation policy on multi-robot data and supports adaptation across observation and action interfaces. CrossFormer (Doshi et al., 2025) demonstrates that a single policy can be trained across manipulation, navigation, locomotion, and aerial platforms without manually aligning observation or action spaces. Cross-domain approaches further exploit heterogeneous experience. XSkill (Xu et al., 2023) discovers transferable skill prototypes from unaligned human and robot videos and maps them to executable robot actions. EgoMimic (Kareer et al., 2025) jointly trains on egocentric human and robot demonstrations through coordinated observation and action interfaces.

These approaches demonstrate the potential of sharing experience across heterogeneous agents. PDEG addresses a complementary question: how to organize shared predictive computation and native control grounding when different embodiments retain incompatible action spaces. Rather than requiring a universal control interface, PDEG separates coordinate-dependent interfaces and private physical context from the computation shared across embodiments.

Action abstraction and correspondence. Intermediate action representations offer another route to heterogeneous experience sharing. UniAct (Zheng et al., 2025) learns a Universal Action Space and translates abstract actions into embodiment-specific executable controls. UniVLA (Bu et al., 2025) derives task-centric latent actions from heterogeneous robot and human videos and connects them to robot-specific action decoders. LAD (Bauer et al., 2026) learns an aligned latent action space for different hand embodiments through contrastive supervision before training a latent diffusion policy.

Recent dexterous-hand systems incorporate additional structural constraints. XL-VLA (Jiang et al., 2026) employs differentiable forward kinematics and fingertip-geometry retargeting to construct a common latent representation without requiring paired cross-hand demonstration trajectories. UniDex (Zhang et al., 2026) uses human-in-the-loop retargeting and explicit 3D representations to construct multi-hand training data, together with a Function-Actuator-Aligned Space that maps functionally related actuators into shared coordinates. These methods employ different forms of latent, kinematic, geometric, or functional correspondence.

PDEG takes a different approach. It does not explicitly enforce equality between abstract action representations from different embodiments, such as $z_a^{(i)} \approx z_a^{(j)}$. Instead, native trajectories are abstracted to provide information for predictive computation shared across robots. The resulting action tokens are not required to form a globally equivalent control code; their utility is determined by their contribution to predicting subsequent visual world evolution. Embodiment-specific physical context is separately retained to ground predictive information into native control.

Predictive world models and cross-embodiment interaction. Predictive world models connect perception, dynamics, and action generation. DINO-WM (Zhou et al., 2025) predicts future representations in a pretrained visual feature space and uses the learned dynamics for planning. Unified World Models (Zhu et al., 2025) couples video and action diffusion, enabling multiple predictive and control functions within a common architecture. The Unified Video Action Model (UVA) (Li et al., 2025) employs masked video-action modeling to support multiple predictive and action-generation configurations. mimic-video (Pai et al., 2026) uses representations from a pretrained video model as video-space plans and learns an action decoder that maps these representations to robot controls.

Closely related cross-embodiment predictive methods differ in how they represent heterogeneous actions. LAC-WM (Huang et al., 2026) performs cross-embodiment world-model pretraining through a unified latent action space. It uses an inverse dynamics model to infer latent actions from observation transitions and embodiment-specific projections to connect native actions to the learned space. A particle-based approach represents hands and objects as 3D particles and defines actions through particle displacement, thereby constructing a common geometric action interface for predictive modeling and planning (He et al., 2026). XEWorld (Chen et al., 2026) highlights the influence of robot appearance on the cross-embodiment generalization of visual world models.

PDEG does not assume that shared predictive computation produces morphology-invariant world representations. Instead, it investigates whether heterogeneous native actions can participate in reusable predictive computation without explicit cross-robot action alignment. Embodiment-conditioned retrieval subsequently converts predictive information into native control.

B PDEG Architecture Details

PDEG separates shared predictive and control computation from embodiment-specific coordinate interfaces and physical context. The Morphology Encoder generates 1024 learned-query tokens, the World-to-Control Adapter (WCA) uses 256 control queries, and the Action Query Decoder (AQD) produces conditions for a 16-step future action horizon. The principal architecture configurations are summarized in Table 3.

B.1 Morphology Encoder

For an input trajectory $X^{\mathcal{M}_r}$ from embodiment $\mathcal{M}_r$, an embodiment-specific Motion Adapter $P_M^{\mathcal{M}_r}$ first projects native coordinates into a common hidden width:

$$\bar{X}^{\mathcal{M}_r} = P_M^{\mathcal{M}_r}\left(X^{\mathcal{M}_r}\right). \tag{11}$$

Table 3: Main PDEG architecture configuration. Depth denotes the number of blocks in each module.

Module	Depth	Width	Heads
Shared temporal encoder	6	768	–
Embodiment temporal encoder	2	768	–
MAR encoder	12	768	12
MAR decoder	12	768	12
WCA	3	768	12
Action Query Decoder	1	768	12

The projected sequence is processed by a temporal encoder Φ_M whose parameters are shared across embodiments:

$$H_M^{\mathcal{M}_r} = \Phi_M\left(\bar{X}^{\mathcal{M}_r}\right). \tag{12}$$

In our implementation, Φ_M is a six-block temporal residual network. A set of learned queries Q_M then retrieves trajectory information from the temporal features:

$$\mathcal{Z}_M\left(X^{\mathcal{M}_r}\right) = \mathrm{CrossAttn}\left(Q_M, H_M^{\mathcal{M}_r}, H_M^{\mathcal{M}_r}\right). \tag{13}$$

Here, Q_M provides the queries, whereas $H_M^{\mathcal{M}_r}$ supplies both keys and values. The resulting tokens expose native trajectory information to shared predictive computation without requiring explicit correspondence between different embodiments.

Historical proprioception produces

$$Z_{\mathrm{state}}^{\mathcal{M}_r} = \mathcal{Z}_M\left(S_h^{\mathcal{M}_r}\right). \tag{14}$$

When true future actions are available as predictive conditions during training, the same encoding route produces

$$Z_{\mathrm{action}}^{\mathcal{M}_r} = \mathcal{Z}_M\left(A_f^{\mathcal{M}_r}\right). \tag{15}$$

No explicit cross-embodiment alignment objective is imposed on these action tokens. They are optimized through their participation in shared predictive modeling rather than through independently specified action correspondence constraints.

B.2 Embodiment Encoder

The Embodiment Encoder provides a complementary private representation of the executing robot. For embodiment $\mathcal{M}_r$, a private state adapter $P_E^{\mathcal{M}_r}$ and private temporal encoder $\Psi_{\mathcal{M}_r}$ produce

$$E_{\mathrm{robot}}^{\mathcal{M}_r} = \Psi_{\mathcal{M}_r}\left(P_E^{\mathcal{M}_r}\left(S_h^{\mathcal{M}_r}\right)\right). \tag{16}$$

Unlike Φ_M, the temporal processing in this route is not shared across embodiments. Both the state adapter and temporal encoder are embodiment-specific. The private temporal encoder contains two residual blocks with hidden width 768.

The two robot-side representations serve complementary purposes. $Z_{\mathrm{state}}^{\mathcal{M}_r}$ participates in predictive computation shared across embodiments, while $E_{\mathrm{robot}}^{\mathcal{M}_r}$ retains private physical context for embodiment-conditioned control grounding. PDEG therefore does not require the shared trajectory representation to discard all embodiment-related information.

B.3 Shared MAR and Video Prediction

Historical visual observations are encoded by the frozen VAE:

$$Z_{\mathrm{video}}^{\mathcal{M}_r} = \mathrm{VAE}\left(O_h^{\mathcal{M}_r}\right). \tag{17}$$

The shared masked autoregressive (MAR) backbone F_θ integrates visual, state, and action-slot tokens into a predictive world representation.

Let $\widetilde{Z}^{\mathcal{M}_r}_{\text{action}}$ denote the action-slot input. When true future native actions are provided as predictive conditions, it contains $Z^{\mathcal{M}_r}_{\text{action}}$; otherwise, unavailable action information is represented by learned fake-action tokens. The shared predictive computation is written as

$$Z_{\text{world}} = F_\theta\left(Z^{\mathcal{M}_r}_{\text{video}}, Z^{\mathcal{M}_r}_{\text{state}}, \widetilde{Z}^{\mathcal{M}_r}_{\text{action}}\right). \tag{18}$$

Future-video slots are additionally observed, partially masked, or fully masked according to the active training configuration, as detailed in Appendix C. Equation 18 summarizes the principal robot and visual inputs rather than enumerating every future-video masking configuration.

The MAR backbone contains a 12-layer encoder and a 12-layer decoder with hidden dimension 768. It produces dense predictive world tokens,

$$Z_{\text{world}} \in \mathbb{R}^{B\times 1024\times 768}.$$

When future-video prediction is active, these world tokens condition the video-diffusion branch to reconstruct masked future visual content. This predictive objective supervises the shared world representation through subsequent visual evolution.

The MAR parameters are shared across embodiments, but the resulting world tokens are not required to be morphology-invariant. Their values may retain robot appearance and other embodiment-related information useful for prediction. The video-diffusion branch provides predictive training supervision and is not required for online action execution.

B.4 World-to-Control Adapter and Action Query Decoder

The World-to-Control Adapter (WCA) comprises three serial refinement blocks. Each block operates on the same set of control queries and cross-attends to the same shared world representation.

Let $Q^{\mathcal{M}_r}_0 = Q_c \in \mathbb{R}^{B\times K\times D}$, where $K = 256$ and $D = 768$. The projected embodiment representation is

$$R^{\mathcal{M}_r} = P_Q(E^{\mathcal{M}_r}_{\text{robot}}) \in \mathbb{R}^{B\times 1\times D}. \tag{19}$$

At block i, for $i \in \{0, 1, 2\}$, the embodiment representation is broadcast across the control-query dimension:

$$\widetilde{Q}^{\mathcal{M}_r}_i = Q^{\mathcal{M}_r}_i + \alpha_i R^{\mathcal{M}_r}. \tag{20}$$

The block then computes

$$\begin{aligned} A^{\mathcal{M}_r}_i &= \text{CrossAttn}_i\left(\widetilde{Q}^{\mathcal{M}_r}_i, Z_{\text{world}}, Z_{\text{world}}\right), \\ H^{\mathcal{M}_r}_i &= \text{LN}_{i,1}\left(\widetilde{Q}^{\mathcal{M}_r}_i + A^{\mathcal{M}_r}_i\right), \\ Q^{\mathcal{M}_r}_{i+1} &= \text{LN}_{i,2}\left(H^{\mathcal{M}_r}_i + \text{FFN}_i(H^{\mathcal{M}_r}_i)\right). \end{aligned} \tag{21}$$

Each block has independent attention, feed-forward, and LayerNorm parameters. The world representation remains unchanged across blocks, whereas the updated control queries are passed sequentially to the next block.

The final control tokens are

$$Z^{\mathcal{M}_r}_{\text{control}} = \text{LN}_{\text{out}}(Q^{\mathcal{M}_r}_3) \in \mathbb{R}^{B\times 256\times 768}. \tag{22}$$

WCA initialization. The three WCA blocks are initialized jointly. The first block uses Xavier-uniform initialization for its attention and feed-forward layers, with zero biases and unit-scale LayerNorm. The second and third blocks are initialized similarly, except that their attention output projections and final feed-forward projections are zero-initialized to suppress the initial residual updates.

The shared control queries are initialized from a truncated normal distribution with standard deviation 0.02. The embodiment projection is zero-initialized, with injection coefficients $\alpha_0 = 0.1$ and $\alpha_1 = \alpha_2 = 0$. This initialization makes the additional refinement blocks approximate identity mappings while allowing them to learn during subsequent optimization.

Table 4: Masked video-action training configurations adopted from UVA.

Mode	History video	Future video	Future action	Prediction
Video Model	Observed	Partial mask	Fake	Video
Dynamic Model	Observed	Partial mask	Observed	Video
Policy Model	Observed	Full mask	Fake	Action
Inverse Model	Full mask	Observed	Fake	Action
Full Dynamic Model	Observed	Partial mask	Fake	Video + Action

The Action Query Decoder converts the retrieved control tokens into temporally organized conditions for the future action horizon:

$$C_{\text{action}}^{\mathcal{M}_r} = G_{\text{AQD}}\left(Q_{\text{act}} + E_{\text{time}}, Z_{\text{control}}^{\mathcal{M}_r}\right). \tag{23}$$

Here, Q_{act} contains H_a learned action queries and E_{time} provides explicit temporal embeddings. The decoder output is

$$C_{\text{action}}^{\mathcal{M}_r} = \left[c_1^{\mathcal{M}_r}, \ldots, c_{H_a}^{\mathcal{M}_r}\right],$$

where each condition corresponds to one position in the future native action chunk. This temporal decoding avoids using a single pooled world feature as the condition for the entire action horizon.

B.5 Native Action Diffusion

The final action generator follows conditional diffusion policy (Chi et al., 2023). Following the standard noise parameterization of Ho et al. (2020), a native action chunk is corrupted at diffusion step k as

$$A_k^{\mathcal{M}_r} = \sqrt{\bar{\alpha}_k} A_f^{\mathcal{M}_r} + \sqrt{1-\bar{\alpha}_k}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{24}$$

Here, $A_f^{\mathcal{M}_r}$ is the ground-truth anchor-relative native action chunk defined in Eq. 2, and $\bar{\alpha}_k$ denotes the cumulative noise-schedule coefficient.

Each embodiment uses private input and output projections around a shared denoising trunk, as defined in Eq. 8. The shared denoiser contains six AdaLN residual blocks and receives diffusion-step embeddings together with the action conditions produced by AQD.

Noise prediction and iterative denoising operate only over the valid native action dimensions of the executing robot. Any padded dimensions used for batch compatibility are excluded from the physical action interpretation. At inference, iterative denoising produces $\hat{A}_f^{\mathcal{M}_r}$ in the target embodiment's native joint space.

C Training Details

C.1 Masked Multi-Function Modeling

For optimization, PDEG adopts the five masked video-action training configurations introduced by UVA (Li et al., 2025). Historical proprioception $S_h^{\mathcal{M}_r}$ is available in all configurations, while historical actions are not used.

The training configurations determine which future-video and future-action slots are available as predictive conditions and which modality is reconstructed. When a modality is unavailable as conditioning information, its corresponding token slots are filled with learned fake tokens rather than numerical zeros.

For future video, partial masking replaces 70–100% of spatial latent positions with fake tokens. The sampled spatial mask is shared across the four future frames. The five configurations are summarized in Table 4.

Only *Dynamic Model* provides the true future native action trajectory $A_f^{\mathcal{M}_r}$ as a predictive condition. It is encoded by the Morphology Encoder into $Z_{\text{action}}^{\mathcal{M}_r}$ before entering MAR.

In *Policy Model*, *Inverse Model*, and *Full Dynamic Model*, true future actions are supervision targets rather than inputs to the Morphology Encoder. Consequently, PDEG learns to incorporate native action information into shared predictive computation without imposing an explicit cross-embodiment alignment loss.

C.2 Two-Stage Optimization

Training follows a two-stage procedure, using the predictive and control configurations described above.

Stage I: Predictive pretraining. Stage I uses only *Video Model* to pretrain MAR and video diffusion through future-video prediction. No action prediction or future-action conditioning is performed. Although the three-block WCA is instantiated, it does not participate in forward computation or parameter updates during this stage.

Stage II: Joint optimization. Stage II inherits the Stage-I EMA checkpoint, including the initialized but previously untrained WCA parameters. All five training configurations are sampled, and the predictive and control components are jointly optimized. MAR and video diffusion remain trainable, while WCA, AQD, and native action diffusion are trained for action generation.

The five training modes are sampled with the following probabilities:

$$p(m) = \begin{cases} 0.40, & m = \textit{Policy Model}, \\ 0.15, & \text{otherwise}. \end{cases} \tag{25}$$

For mixed-embodiment batches, action losses are first averaged within each represented embodiment and then averaged across embodiments:

$$\mathcal{L}_{\text{action}} = \frac{1}{|\mathcal{R}_{\mathcal{B}}|} \sum_{r \in \mathcal{R}_{\mathcal{B}}} \frac{1}{|\mathcal{B}_r|} \sum_{i \in \mathcal{B}_r} \ell_i^{\mathcal{M}_r}, \tag{26}$$

where $\mathcal{B}_r$ is the set of samples from embodiment $\mathcal{M}_r$ and $\mathcal{R}_{\mathcal{B}}$ is the set of embodiments represented in the batch.

This robot-balanced reduction assigns equal weight to the embodiments present in a mixed batch, regardless of their individual sample counts. It prevents more frequently represented embodiments from directly dominating shared control optimization.

Optimization. Both stages use AdamW with a base learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 0.02. Bias and normalization parameters are excluded from weight decay. The learning rate follows cosine decay with 10,000 warmup optimizer steps. Gradient clipping is not applied.

We maintain an exponential moving average (EMA) of model parameters with a step-dependent decay

$$\lambda_s = \min\left(0.9999,\, 1 - (1+s)^{-0.75}\right), \tag{27}$$

where EMA updates begin at step $s = 0$.

Both video and action diffusion use 1,000 training timesteps and 100 inference sampling steps.

C.3 Inference

At deployment, PDEG receives only historical visual observations $O_h^{\mathcal{M}_r}$ and proprioception $S_h^{\mathcal{M}_r}$.

The Morphology Encoder produces $Z_{\text{state}}^{\mathcal{M}_r}$, the Embodiment Encoder produces $E_{\text{robot}}^{\mathcal{M}_r}$, and the VAE encodes historical observations into $Z_{\text{video}}^{\mathcal{M}_r}$.

MAR forms Z_{world} using learned fake-action tokens in place of unavailable future native actions. WCA retrieves $Z_{\text{control}}^{\mathcal{M}_r}$ conditioned on $E_{\text{robot}}^{\mathcal{M}_r}$, and AQD produces temporally organized action conditions $C_{\text{action}}^{\mathcal{M}_r}$. Iterative conditional denoising then generates $\hat{A}_f^{\mathcal{M}_r}$ in the target embodiment's native action space.

Neither true future video nor true future action is available as a condition during policy execution. The video-diffusion prediction branch is unnecessary for online action generation.

D Experimental Details

D.1 Dataset and Observation Pipeline

Demonstrations are collected through joystick teleoperation at 30 Hz. We evaluate three end effectors mounted on the same 6-DoF RealMan RM65 arm: LinkerHand L10, Inspire Robots RH56DFX, and DH-Robotics AG95.

All three embodiments perform Peach Placement, Cup Stacking, and Ball Pouring, yielding nine task-embodiment combinations. Each combination contains approximately 300 successful demonstrations, for approximately 2700 trajectories across the full dataset.

Individual demonstrations contain approximately 80–170 frames, corresponding to roughly 3–6 seconds. Each subset contains approximately 4×10^4–5×10^4 frames, giving approximately 3.6×10^5–4.5×10^5 frames over the full dataset.

RGB observations are captured with a front-view Intel RealSense D435 camera at 640×480 resolution and resized to 256×256 before entering the model. The camera configuration is kept approximately consistent across embodiments.

The policy uses four visual conditioning frames, a 16-step proprioceptive history, and a 16-step future action horizon. Native action targets follow the anchor-relative definition in Eq. 2, while historical proprioception uses native joint positions.

No manually constructed cross-embodiment action mapping or retargeted action representation is used. State and action dimensions are normalized independently for each embodiment using per-joint min-max statistics.

Baselines and controlled comparisons. We compare PDEG against UVA (Li et al., 2025) and an adapted multi-embodiment EgoMimic (Kareer et al., 2025) baseline.

For each task, all UVA and PDEG Stage-II variants initialize their MAR and video-diffusion components from the same task-specific, three-embodiment PDEG Stage-I predictive checkpoint. These shared predictive components remain trainable during Stage II.

UVA-Single uses only target-embodiment demonstrations during Stage II, whereas UVA-Multi jointly uses demonstrations from all three embodiments.

PDEG-Single retains the complete PDEG architecture but likewise uses only target-embodiment demonstrations during Stage II. PDEG-Multi additionally incorporates demonstrations from the other two embodiments.

Thus, each Single/Multi pair shares the same predictive initialization and target-robot Stage-II data budget. The controlled difference is the availability of additional source-embodiment Stage-II experience. The comparisons measure incremental transfer from heterogeneous Stage-II data under a common predictive initialization, rather than differences between independently pretrained single- and multi-embodiment models.

EgoMimic adaptation. We extend EgoMimic's robot branch to three embodiments without using human demonstrations. Only the embodiment-specific input and output interfaces are extended, with independent native action heads; the remaining architecture is unchanged. EgoMimic takes one historical front-view RGB frame and predicts 50 future steps using stepwise native joint displacements. It uses the same robot demonstrations and per-embodiment data budget and is evaluated under identical real-world conditions.

The UVA Single/Multi comparison provides a reference for evaluating incremental transfer under an alternative shared-policy architecture.

Table 5: **Parameter counts under different execution scopes (M).** Active core parameters are used for the main-text scalability comparison.

Model	Scope	1 Emb.	2 Emb.	3 Emb.
UVA	Active core	261.727	313.999	366.252
PDEG	Active core	303.116	311.441	319.747
UVA	Training-active	289.993	342.265	394.519
PDEG	Training-active	331.383	339.708	348.014
UVA	Full video lifecycle	328.183	380.455	432.708
PDEG	Full video lifecycle	369.572	377.897	386.203
UVA	Instantiated object	328.196	380.468	432.721
PDEG	Instantiated object	422.462	430.786	439.093

D.2 Training Setup

For each task, Stage I performs 500 epochs of three-embodiment predictive pretraining. All corresponding UVA and PDEG Stage-II variants inherit the same task-specific MAR and video-diffusion parameters.

PDEG additionally loads its three-block WCA parameters from the Stage-I EMA checkpoint. Since WCA receives no updates during Stage I, these parameters remain at their initialized values, as specified in Appendix B. Stage II then jointly trains predictive and control components for 500 epochs. All comparisons use the same task-specific data partitions and predictive initialization.

Training uses eight NVIDIA RTX 4090 GPUs with a per-GPU batch size of 12 and bfloat16 mixed precision.

Checkpoint selection. Stage-I predictive checkpoints are selected using validation Fréchet Video Distance (FVD) (Unterthiner et al., 2018), whereas Stage-II checkpoints are selected using validation action L2 loss. Real-world control and qualitative future-video predictions use the corresponding Stage-II exponential moving average (EMA) checkpoints.

D.3 Parameter Accounting and Embodiment Scalability

Counting protocol. The set of parameters used for training, video decoding, and online action execution is not identical. We therefore distinguish four parameter-counting scopes.

Active core parameters include parameters participating in at least one active video-action training path across the five training configurations and supported embodiments. The shared frozen VAE and modules retained solely for checkpoint compatibility are excluded. This is the counting scope used for the main-text parameter scalability comparison.

Training-active parameters add the frozen VAE encoder and `quant_conv` components used to encode visual observations during training.

Full video lifecycle parameters additionally include the VAE decoder used to decode predicted video latents.

Instantiated object parameters count all parameters attached to the PyTorch policy object, including inactive compatibility modules.

The frozen VAE contains 66.456M parameters in total, of which 28.267M participate in the training forward path. The different counting scopes are summarized in Table 5.

Source of parameter growth. The difference in embodiment scaling primarily arises from the action generation architecture.

UVA instantiates an independent diffusion action module for each supported embodiment. Its action component therefore grows from 52.263M parameters for one embodiment to 156.758M for three embodiments.

Table 6: **Detailed real-world transfer results.** Success counts are reported over 50 physical rollouts per entry. Parentheses denote 95% Wilson confidence intervals (%); transfer gains are in percentage points.

Task	Embodiment	PDEG-Single	PDEG-Multi	Δ_{transfer}
Peach Placement	L10	42/50 (71.5–91.7)	46/50 (81.2–96.8)	+8
	RH56DFX	43/50 (73.8–93.0)	44/50 (76.2–94.4)	+2
	AG95	41/50 (69.2–90.2)	47/50 (83.8–97.9)	+12
Cup Stacking	L10	35/50 (56.2–80.9)	40/50 (67.0–88.8)	+10
	RH56DFX	36/50 (58.3–82.5)	39/50 (64.8–87.2)	+6
	AG95	31/50 (48.2–74.1)	35/50 (56.2–80.9)	+8
Ball Pouring	L10	37/50 (60.4–84.1)	43/50 (73.8–93.0)	+12
	RH56DFX	38/50 (62.6–85.7)	41/50 (69.2–90.2)	+6
	AG95	34/50 (54.2–79.2)	40/50 (67.0–88.8)	+12

PDEG instead shares WCA and the high-capacity control denoising trunk. Adding an embodiment primarily introduces a lightweight Motion Adapter, a private Embodiment Encoder, and small native input and output projections.

Consequently, the active core grows by approximately 52.26M parameters per additional embodiment for UVA, compared with 8.32M for PDEG. This corresponds to $6.29\times$ lower marginal parameter growth.

PDEG has a larger instantiated-object count because the current implementation retains legacy modules solely for checkpoint compatibility, including an unused UVA action-diffusion module and obsolete projections. These inactive modules do not participate in the current forward computation and are excluded from the active-core comparison.

At deployment, future-video prediction is unnecessary, so online action execution uses only a subset of the training-time model.

Interpretation. The parameter analysis quantifies the architectural scalability of PDEG's shared/private decomposition. Rather than replicating a high-capacity action generator for each new embodiment, PDEG amortizes predictive and control computation across supported robots.

The resulting reduction in marginal parameter growth is an architectural property. The controlled Single/Multi experiments separately examine whether adding heterogeneous demonstrations also improves target-robot performance.

D.4 Real-World Evaluation Protocol

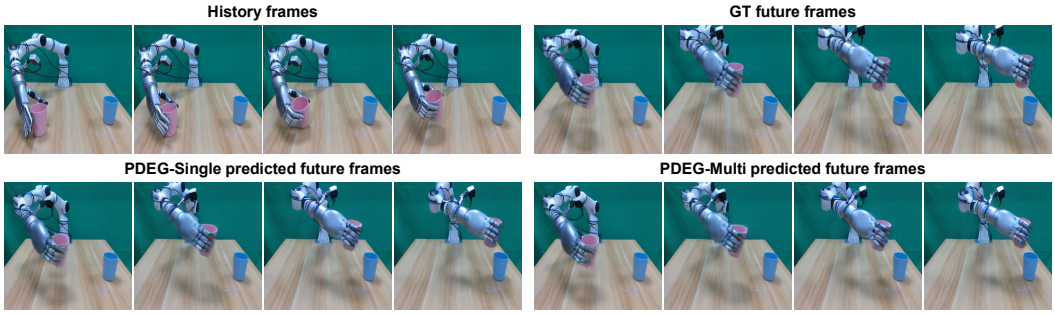
For each task, we define discrete marked initial object configurations within the valid manipulation workspace. The 50 evaluation trials are distributed approximately evenly across these configurations, and compared methods use the same configuration set whenever physically possible.

Objects are manually reset before each rollout. This introduces small unavoidable perturbations around the marked initial positions.

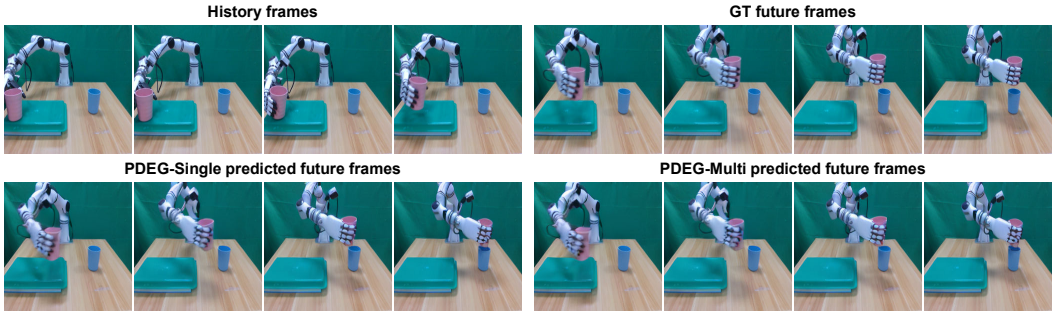
Unless otherwise stated, each method-embodiment-task setting is evaluated over 50 real-world rollouts. A rollout starts from a prescribed reset configuration and terminates upon task completion, an unrecoverable failure, or the evaluation time limit.

Success is determined according to task-specific criteria: the peach must be completely placed inside the bowl for Peach Placement; one cup must be stably stacked on the other for Cup Stacking; and the ball must be transferred into the target cup for Ball Pouring. All outcomes are manually verified.

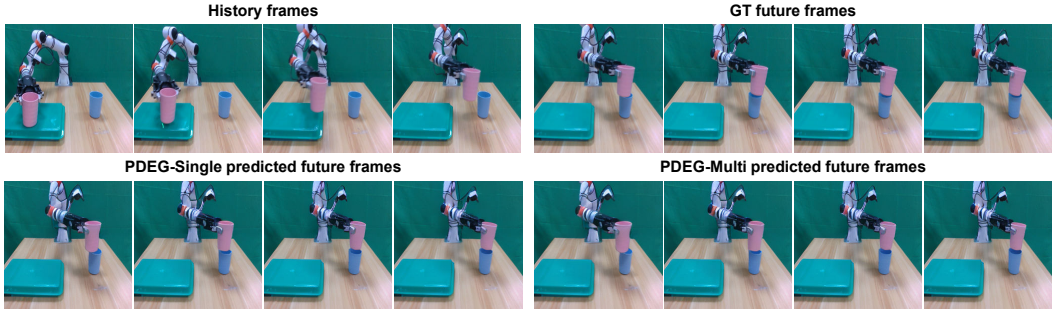
We report success rates, the corresponding numbers of successful trials, and 95% Wilson confidence intervals (Wilson, 1927). For the Single/Multi comparisons, we additionally report the transfer gain Δ_{transfer} defined in Eq. 10.



(a) L10.



(b) RH56DFX.



(c) AG95.

Figure 5: **Qualitative future-video prediction across three embodiments.** Held-out Cup Stacking trajectories for (a) L10, (b) RH56DFX, and (c) AG95. Each embodiment is shown with four groups of four frames: observed history, ground-truth future observations, PDEG-Single predictions, and PDEG-Multi predictions. Single and Multi use identical historical inputs within each embodiment, with all future-video tokens masked and no ground-truth future actions.

Table 6 provides the detailed task-embodiment transfer results. Confidence intervals are expressed as percentages, whereas Δ_{transfer} is measured in percentage points.

D.5 Qualitative Future-Video Prediction

We further examine the predictive behavior of PDEG on held-out Cup Stacking trajectories across L10, RH56DFX, and AG95. For each embodiment, we compare PDEG-Single and PDEG-Multi using identical historical visual observations and proprioceptive inputs. Both models generate four future frames through the *Full Dynamic Model* prediction pathway, with all future-video tokens masked and no ground-truth future-action conditioning.

Qualitative observations. Figure 5 provides a side-by-side comparison of future-video predictions across three heterogeneous end effectors. For each embodiment, the predicted sequences from

PDEG-Single and PDEG-Multi are compared with the same ground-truth future observations under matched historical inputs.

These examples illustrate the predictive behavior of both models across different robot-object interactions without assuming aligned action representations. They provide qualitative evidence of future visual evolution rather than a direct measure of closed-loop control performance. These examples complement the main-text visualization by covering all three embodiments.