TOKENS OF MOTION: PRIMITIVE MOTION LEARNING FOR HUMAN-HUMANOID SKILL TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Human videos provide a scalable source of manipulation experience, yet transferring them to robots remains challenging because human demonstrations lack robot-compatible actions and differ substantially from robots in embodiment and kinematics. We propose *Motion Primitives for Action Composition and Transfer* (M-PACT), a framework that addresses this challenge by factorizing primitive motion learning from embodiment-specific motion realization. This separation allows us to learn generalizable and reusable motion primitives from large-scale 3D human hand-pose trajectories, while requiring only limited embodiment-specific robot data to ground these primitives into executable motions. Specifically, we first discover local motion segments from human hand trajectories and learn a compact primitive-motion representation that captures recurring manipulation patterns independently of robot embodiment. Given task and observation context, these primitives are then selected and composed, while a lightweight embodiment-specific decoder maps them to robot motions. Crucially, transferring to a new embodiment preserves the learned primitive representation and requires adapting only the realization model, substantially reducing the dependence on costly robot demonstrations. Experiments across EgoDex, HoloAssist, and HumanoidEveryday demonstrate cross-dataset and cross-embodiment transferability of proposed M-PACT, with lightweight decoder adaptation achieving strong humanoid motion reconstruction using as few as $\sim$100 robot demonstrations. These results establish factorized primitive learning as a scalable bridge from large-scale human motion data to humanoid manipulation.

1 INTRODUCTION

Developing robots capable of diverse manipulation in unstructured environments remains a central goal in robotics and artificial intelligence. Most manipulation policies are learned from robot-centric experience, including demonstrations collected through teleoperation or kinesthetic teaching and interactions acquired through reinforcement learning or autonomous exploration. Although such data provide directly executable actions, collecting them at sufficient scale and diversity remains costly and difficult across tasks, environments, and embodiments Brohan et al. (2022); O'Neill et al. (2024). Human videos offer a compelling alternative: they are abundant and capture diverse object interactions, behavioral strategies, and long-horizon activities in natural settings Grauman et al. (2022). They can therefore complement limited robot demonstrations and broaden the task and behavioral coverage of robot learning systems Kareer et al. (2025). However, human demonstrations lack robot-compatible action supervision, while human and robot effectors differ substantially in morphology, kinematics, appearance, and feasible action spaces Kareer et al. (2025); Chen et al. (2025). Directly imitating human trajectories can consequently produce ambiguous or kinematically infeasible robot actions. Bridging this embodiment gap requires representations that abstract away embodiment-specific execution while preserving transferable motion structure, for effective and generalizable robot control.

Vision-language-action (VLA) models (Pertsch et al., 2025; Black et al., 2024; Intelligence et al., 2025; Liu et al., 2025; Brohan et al., 2023) offer a promising framework for general-purpose manipulation by mapping visual observations and language instructions directly to robot actions. Recent models such as Ψ_0 Wei et al. (2026), H-RDT Bi et al. (2026), and Being-H0 Luo et al. (2025) further pretrain on large-scale egocentric human videos before adapting to specific robot embodiments, demonstrating the potential of human data to broaden behavioral coverage. However, two limitations remain. First,

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.

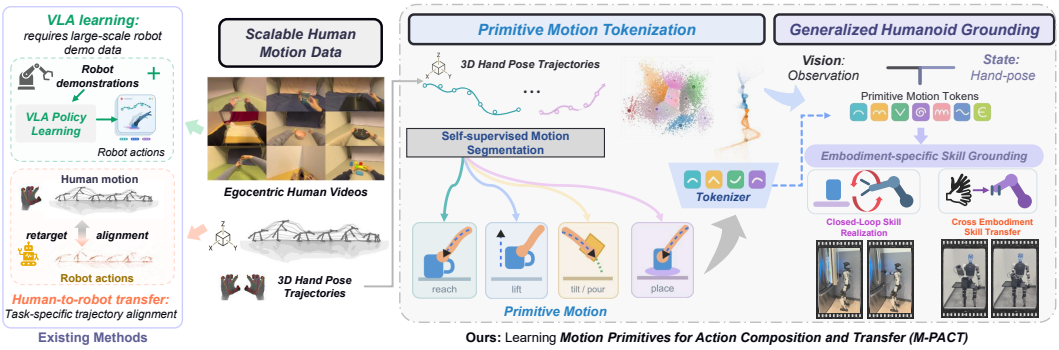

Figure 1: Proposed *Motion Primitives for Action Composition and Transfer* (M-PACT) pipeline.

human videos lack robot-executable actions, so converting them into control supervision still requires costly robot demonstrations or computationally intensive human-motion estimation. Second, existing pretraining paradigms often couple human motion with scene appearance and context rather than explicitly isolating transferable motion structure, limiting generalization across scenes, tasks, and embodiments.

Existing human-to-robot skill transfer approaches primarily bridge the embodiment gap by geometric alignment of human hand and robot trajectories Kareer et al. (2025); Chen et al. (2025); Shi et al. (2025); Haldar & Pinto (2025); Zhou et al. (2026); Li et al. (2024), visual transformation of human demonstrations into robot-centric observations Li et al. (2025); Lepert et al. (2025); Chen et al. (2026a), or reconstructing hand-object interactions and retargeting to dexterous robot hands Wang et al. (2024); Mu et al. (2026); Singh et al. (2025); Chen et al. (2026b); Paliwal et al. (2026); Lai et al. (2026). However, hand-object manipulation is inherently compositional and behaviorally multimodal: a single task may comprise heterogeneous motions such as grasping, lifting, placing, *etc*., each admitting multiple valid executions. Representing such a skill as a monolithic, full-trajectory supervision signal requires an imitation-learning policy to model long and highly variable motion sequences while simultaneously resolving human-robot embodiment differences, making learning difficult and limiting the reuse of shared behaviors across tasks. Recent approaches have begun to introduce temporal hierarchy, however, these methods generally transfer human behavior through complete trajectories Singh et al. (2025); Chen et al. (2025), coarse semantic subtasks Mu et al. (2026); Shi et al. (2025), fixed-horizon trajectory and action predictions Haldar & Pinto (2025); Zhou et al. (2026). Consequently, recurring local motion patterns can remain coupled to task-level ordering and global scene context, limiting their systematic recomposition and transfer across novel tasks and environments.

To this end, we propose to learn *Motion Primitives for Action Composition and Transfer* (M-PACT), a framework that learns localized *primitive motions* from human videos by factorizing transferable interaction structure from scene- and embodiment-specific realizations. This factorization allows individual primitives to be transferred across embodiments and compositionally reused across tasks and environments. At its core, a self-supervised motion segmentation module, SkillSeg, partitions 3D human hand trajectories using frame-level kinematic cues, and maps the resulting segments into an embodiment-agnostic vocabulary of primitive-motion tokens, by *Embodied Primitive Motion Tokenization* (EPMT) module. Learned from web-scale egocentric video corpora, EPMT transforms abundant, behaviorally diverse, and cost-effective human pose trajectories into reusable manipulation primitives. We then condition these primitive representations on contextual features from visual foundation models and perform lightweight fine-tuning using labeled video-motion-action data, enabling M-PACT to select, compose, and realize primitives for diverse manipulation tasks. By separating reusable motion structure from task-specific context and robot-specific control, our approach substantially reduces the need for costly robot demonstrations while improving generalization across tasks and scenes. In summary, our contributions are threefold:

- We propose to learn a motion tokenizer, EPMT that discovers localized, embodiment-agnostic primitive motions from human hand trajectories. Trained on low-cost, web-scale hand-pose data extracted from egocentric videos, EPMT learns primitive tokens that can be reused for skill transfer across embodiments.

- We introduce M-PACT, a multi-stage transfer framework that integrates EPMT with context-aware visual representations to select and compose primitive tokens and ground them into embodiment-specific, robot-executable actions. This design supports manipulation tasks of varying horizons with lightweight fine-tuning on labeled video-motion-action data.
- We evaluate motion reconstruction, cross-dataset representation reuse, humanoid decoder adaptation, and contextual primitive prediction across three datasets, complemented by qualitative real-world humanoid demonstrations.

2 Related Work

Learning Manipulation from Human Demonstrations. Human demonstrations offer a scalable source of manipulation experience, motivating approaches based on shared representations, geometric alignment, and motion retargeting. EgoMimic jointly trains policies from aligned egocentric human and robot data, while Point Policy predicts keypoint trajectories that are converted into robot actions through geometric constraints Kareer et al. (2025); Haldar & Pinto (2025). For dexterous manipulation, DexCap transfers captured hand motions through inverse kinematics and imitation learning, whereas DexImit reconstructs hand–object interactions and synthesizes robot demonstrations through subtask decomposition and trajectory generation Wang et al. (2024); Mu et al. (2026). In contrast, we first learn a compact vocabulary of local hand-motion primitives, then separately model contextual selection and embodiment-specific decoding, disentangling reusable motion structure from task context and robot execution.

Temporal Abstraction and Reusable Skill Discovery. Temporal abstraction organizes demonstrations into reusable behavioral units. CompILE jointly discovers segment boundaries and latent behaviors, while Play-LMP and SPiRL learn latent plans or skill priors from unstructured experience Kipf et al. (2019); Lynch et al. (2020); Pertsch et al. (2021). MimicPlay learns a planner through human hand-trajectory prediction and uses it to guide a robot policy trained from teleoperated demonstrations Wang et al. (2023a). Our approach instead discovers boundaries from hand kinematics and represents each segment through factorized translation, rotation, and aperture effects. These local primitives are not predefined semantic subtasks and can therefore be reused across demonstrated skills before contextual prediction is learned over the fixed vocabulary.

Discrete Action Representations and Motion Tokenization. Discrete action representations support categorical prediction and sequence modeling. VQ-BeT uses hierarchical vector quantization to model multimodal actions, while FAST tokenizes transformed robot action sequences for efficient autoregressive learning Lee et al. (2024); Pertsch et al. (2025). LAPA extends action tokenization to videos without robot labels by learning latent actions between frames and subsequently adapting with robot data Ye et al. (2025). Our framework instead learns from segmented 3D hand trajectories under a shared kinematic convention. Translation and aperture use independent discrete codebooks, rotation remains continuous, and the decoder reconstructs each segment's net effect conditioned on its initial state. This structure enables embodiment-specific adaptation while keeping the motion encoders and codebooks fixed.

3 Method

3.1 Framework Overview

Our central design separates a shared motion representation from embodiment-specific effect decoding. We learn the representation from human hand trajectories and retain it when fitting a compact decoder to robot demonstrations. This allows us to study which motion information can be reused across datasets and embodiments, and which mappings require adaptation.

The framework has three stages. SkillSeg first partitions trajectories into local motion segments using self-supervised reconstruction. *Embodied Primitive Motion Tokenization* (EPMT) then encodes each segment into discrete translation and aperture tokens and a continuous rotation latent, trained jointly with a state-conditioned effect decoder. Finally, target-embodiment adaptation fits a new decoder while freezing the segmentation model, motion encoders, and codebooks. Thus, embodiment adaptation changes the mapping from a shared representation to motion effects while preserving the representation itself. An illustration of the proposed pipeline is shown in Figure 2.

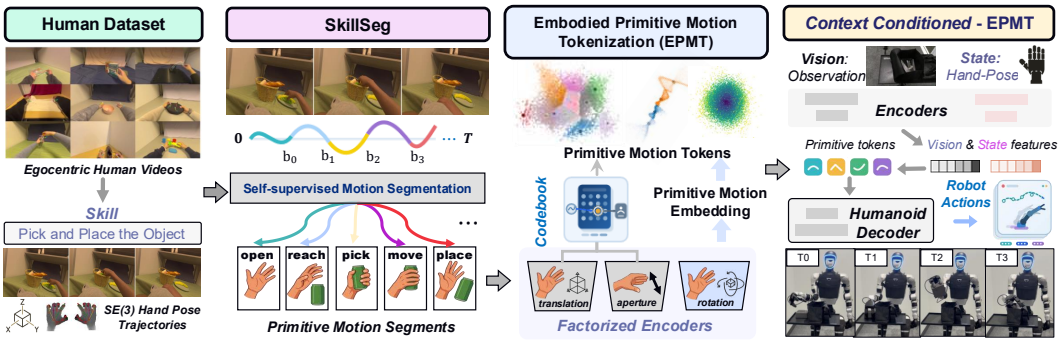

Figure 2: Overview of *Motion Primitives for Action Composition and Transfer* (M-PACT) pipeline.

We study two extensions of this core. Context-Conditioned EPMT (CC-EPMT) predicts the representation from information available at a segment's start. Human-to-humanoid grounding continues the robot decoder on human-derived representations to examine the resulting reconstruction trade-off. These components are trained in stages. Representation reconstruction, contextual prediction, and offline motion realization are evaluated separately, so their respective errors can be distinguished.

3.2 Shared Cross-Embodiment Motion Representation

We represent a demonstration by a hand-state sequence $\tau = \{(p_t, R_t, \alpha_t)\}_{t=0}^{T-1}$, where $p_t \in \mathbb{R}^3$ is wrist position, $R_t \in \mathrm{SO}(3)$ is wrist orientation, and $\alpha_t \in \mathbb{R}$ is hand aperture. For human demonstrations, aperture is the thumb-index fingertip distance and poses are in the head-camera convention. Robot trajectories are recovered through forward kinematics and mapped into the same convention using a coordinate transformation, arm-length scaling, wrist-orientation alignment, and an affine aperture calibration. We denote this embodiment-specific mapping by $\mathcal{A}_e$ and use (p_t, R_t, α_t) for the aligned quantities throughout the method. See Appendix 1.1 for concrete humanoid mapping.

The shared convention makes motion inputs comparable across embodiments; it does not require identical kinematics or guarantee that an aligned human motion is robot-reachable. Statistical normalization is applied separately: the tokenizer uses its source-training statistics, while each embodiment-specific decoder carries its own hand-state and effect normalizers. We write $\pi_6(R) \in \mathbb{R}^6$ for the concatenated first two columns of a rotation matrix and $\mathrm{Log}_v(R) \in \mathbb{R}^3$ for its axis–angle rotation vector.

3.3 SkillSeg: Self-Supervised Motion Segmentation

Local motion phases provide temporal units for learning reusable primitives. SkillSeg discovers their boundaries without annotated segment labels by learning a differentiable temporal decomposition that supports motion reconstruction for cross-embodiment generalization. A temporal network receives smoothed kinematic features f_t containing translational velocity, speed, aperture, aperture rate, and rotation-representation derivatives. It predicts an increment $q_t \in (0, 1)$. A binary motion gate m_t suppresses accumulation during inactive frames, yielding a soft segment index $c_t = \sum_{s=0}^{t} m_s q_s$. For a segment slot k, soft membership is defined by a triangular kernel:

$$w_{t,k} = \max\left(0, 1 - \frac{|c_t - (k + 1/2)|}{\rho}\right), \qquad \widetilde{w}_{t,k} = \frac{w_{t,k}}{\sum_s w_{s,k} + \varepsilon}, \tag{1}$$

where $\rho = 0.6$. A frame encoder is pooled into a segment embedding $a_k = \sum_t \widetilde{w}_{t,k} F(f_t)$, which is quantized through an auxiliary vector-quantization codebook. With v_t denoting translational velocity, an auxiliary decoder reconstructs the motion target $d_k = \sum_t w_{t,k} v_t$. This membership-weighted velocity sum is proportional to a soft displacement integral at the fixed frame rate. Both the segment embeddings and targets depend on the boundary predictions, allowing reconstruction gradients to shape the temporal decomposition.

The segmentation objective combines reconstruction, vector-quantization, rate, and entropy terms:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{motion}} + \lambda_c \mathcal{L}_{\text{VQ}} + \lambda_r \frac{(c_{T-1} - N^\star)^2}{\max((N^\star)^2, 1)} - \lambda_e H(q). \tag{2}$$

Here, $\mathcal{L}_{\text{motion}}$ is mean squared error between the auxiliary decoder's predictions and d_k, averaged over active slots; $\mathcal{L}_{\text{VQ}}$ is the auxiliary quantization loss. A slot is active when its membership mass exceeds 10^{-3}. The duration-dependent target $N^\star$ controls the segment count, and $H(q)$ is mean Bernoulli entropy. The rate term controls temporal granularity, while the entropy term discourages saturated increments. The segmentation decoder receives no initial hand-state information. Its auxiliary encoder and codebook serve segmentation training and are distinct from the factor-specific EPMT modules below.

At inference, we average the increments of $M = 7$ independently trained networks, apply the common motion gate, and extract boundaries at successive integer crossings of the cumulative score. Crossings within $K = 8$ frames of the previous boundary are skipped. Including the trajectory endpoints gives $0 = b_0 < \cdots < b_M = T$, with segment i containing frames $[b_{i-1}, b_i)$. Training details and gate definitions appear in Appendix 1.2.

3.4 EPMT: Factorized Primitive Motion Tokenization

EPMT learns recurring motion components across demonstrated skills without receiving skill labels. Separate encoders expose translation, rotation, and aperture as individually addressable channels, while a common decoder reconstructs their combined motion effect. This design provides a structured interface for factor prediction and decoder adaptation; it does not impose statistical independence between factors.

Segment features and factorized encoding. Let $a_i = b_{i-1}$ and $e_i = b_i - 1$ be the first and last included frames of segment i. We linearly resample each channel at $L = 8$ uniformly spaced points:

$$(x_i^{\text{tr}}, x_i^{\text{rot}}, x_i^{\text{ap}}) = \text{resample}_L\big(\{(p_t - p_{a_i}, \pi_6(R_{a_i}^\top R_t), \alpha_t)\}_{t=a_i}^{e_i}\big) \in \mathbb{R}^{3L} \times \mathbb{R}^{6L} \times \mathbb{R}^L. \tag{3}$$

Translation is expressed as displacement from the segment start in the shared coordinate axes; rotation is relative to its initial wrist orientation. Aperture retains its absolute aligned value. Let $\bar{x}_i^f = \mathcal{N}_x^f(x_i^f)$ denote component-wise standardization using fixed source-training statistics for factor f. Translation and aperture are quantized, whereas rotation remains continuous:

$$\begin{aligned}(z_i^{\text{tr}}, z_i^{\text{rot}}, z_i^{\text{ap}}) &= \big(\text{VQ}_{16}(E_{\text{tr}}(\bar{x}_i^{\text{tr}})), E_{\text{rot}}(\bar{x}_i^{\text{rot}}), \text{VQ}_4(E_{\text{ap}}(\bar{x}_i^{\text{ap}}))\big) \\ &\in \{1, \ldots, 16\} \times \mathbb{R}^3 \times \{1, \ldots, 4\}.\end{aligned} \tag{4}$$

The operator VQ_K returns the nearest codebook entry's index. The codebook sizes are $K_{\text{tr}} = 16$ and $K_{\text{ap}} = 4$, with 32-dimensional embeddings and lookups $e^f(k) = c_k^f$ for $f \in \{\text{tr}, \text{ap}\}$. We distinguish the symbolic representation from its embedded decoder input:

$$Z_i = (z_i^{\text{tr}}, z_i^{\text{rot}}, z_i^{\text{ap}}), \qquad \widetilde{Z}_i = [e^{\text{tr}}(z_i^{\text{tr}}), z_i^{\text{rot}}, e^{\text{ap}}(z_i^{\text{ap}})] \in \mathbb{R}^{67}. \tag{5}$$

This hybrid representation makes translation and aperture independently enumerable while retaining continuous rotational variation. The two discrete indices require six bits in total; the three-dimensional rotation latent has an additional precision-dependent storage cost. Factorization is a structural choice rather than a requirement for minimum reconstruction error, and we compare it with joint encoding in the experiments.

State-conditioned effect reconstruction. The initial state and net motion effect are

$$h_i = [p_{a_i}, \pi_6(R_{a_i}), \alpha_{a_i}] \in \mathbb{R}^{10}, \tag{6}$$

$$y_i = [p_{e_i} - p_{a_i}, \text{Log}_v(R_{a_i}^\top R_{e_i}), \alpha_{e_i} - \alpha_{a_i}] \in \mathbb{R}^7. \tag{7}$$

Let $\mathcal{N}_h$ and $\mathcal{N}_y$ denote component-wise standardization of states and effects, with $\bar{h}_i = \mathcal{N}_h(h_i)$ and $\bar{y}_i = \mathcal{N}_y(y_i)$. The three effect channels are denoted by superscripts tr, rot, and ap. The decoder predicts a normalized effect: $\widehat{\bar{y}}_i = D_\theta(\mathcal{N}_h(h_i), \widetilde{Z}_i)$, $\hat{y}_i = \mathcal{N}_y^{-1}(\widehat{\bar{y}}_i)$. Conditioning on h_i permits a shared primitive representation to produce different effects from different initial configurations. We balance the three factors by averaging errors within each channel before averaging across channels:

$$\ell_{\text{bal}}(\hat{\bar{y}}, \bar{y}) = \frac{1}{3}\left(\frac{\|\hat{\bar{y}}^{\text{tr}} - \bar{y}^{\text{tr}}\|_2^2}{3} + \frac{\|\hat{\bar{y}}^{\text{rot}} - \bar{y}^{\text{rot}}\|_2^2}{3} + |\hat{\bar{y}}^{\text{ap}} - \bar{y}^{\text{ap}}|^2\right). \tag{8}$$

The encoders and decoder minimize:

$$\mathcal{L}_{\text{EPMT}} = \mathbb{E}_i \left[\ell_{\text{bal}}(\widehat{y}_i, \mathcal{N}_y(y_i)) + \beta \sum_{f \in \{\text{tr},\text{ap}\}} \mathcal{L}^f_{\text{commit},i} \right]. \tag{9}$$

The commitment losses penalize encoder distance from stop-gradient codebook embeddings. Straight-through estimation passes reconstruction gradients to the discrete-factor encoders, and exponential moving averages update the codebooks. Although the encoders observe within-segment trajectories, supervision reconstructs only their net effects. Thus, the representation summarizes motion outcomes without explicitly requiring preservation of the full trajectory.

3.5 Cross-Embodiment Decoder Adaptation and Grounding

Target-embodiment adaptation. To transfer the learned representation to a target embodiment, we align robot demonstrations through A_e and process them using the frozen SkillSeg and EPMT modules. The resulting segments provide tuples $(h_i^e, \widetilde{Z}_i^e, y_i^e)$, from which we train a new embodiment-specific decoder:

$$\theta_e = \arg\min_\theta \mathbb{E}_i \left[\ell_{\text{bal}} \left(D_\theta \left(\mathcal{N}_h^e(h_i^e), \widetilde{Z}_i^e \right), \mathcal{N}_y^e(y_i^e) \right) \right]. \tag{10}$$

Each decoder uses the same 77-dimensional input, 7-dimensional output, and 27,399-parameter architecture. Its state and effect normalizers are fitted on the target training data, while the motion-feature normalization, segmentation networks, encoders, and codebooks remain fixed. Source and target demonstrations need not be paired: each target segment supplies its own initial state, encoded representation, and observed effect. Thus, transferring EPMT to a new embodiment requires only learning a compact realization model rather than a new motion vocabulary.

Human-to-humanoid grounding. The adaptation above learns to realize shared primitive representations extracted from robot demonstrations. When primitives instead originate from human demonstrations, the joint distribution of initial states and primitive representations can shift. We therefore optionally continue the robot decoder θ_{rob} on aligned human segments while keeping the EPMT representation and robot-fitted normalizers fixed:

$$\mathcal{L}_{\text{gnd}}(\theta) = \mathbb{E}_{i \sim \mathcal{D}_{\text{hum}}} \left[\text{MSE} \left(D_\theta \left(\mathcal{N}_h^{\text{rob}}(h_i^{\text{hum}}), \widetilde{Z}_i^{\text{hum}} \right), \mathcal{N}_y^{\text{rob}}(y_i^{\text{hum}}) \right) \right]. \tag{11}$$

All continuation inputs and targets come from human segments; no paired human–robot trajectories or robot-effect labels are required. Only the embodiment-specific decoder is updated, preserving the learned primitive vocabulary. We evaluate the resulting trade-off between reconstruction of human-derived motions and preservation of native robot fidelity. Detailed sequential replay, inverse-kinematics evaluation, and execution procedures are provided in the Appendix.

3.6 CC-EPMT: Context-Conditioned Primitive Prediction

EPMT encodes a segment after observing its motion. CC-EPMT instead predicts its representation from context available at the segment's start.

Context variants. The model accepts state-only context or additional visual and language features. With $\mathcal{N}_h^{\text{ctx}}$ denoting the predictor's state standardization, representative inputs are

$$u_i^{\text{state}} = \mathcal{N}_h^{\text{ctx}}(h_i), \qquad u_i^{\text{full}} = [\bar{f}_{\text{img}}(I_i), \bar{f}_{\text{text}}(\ell_i), \mathcal{N}_h^{\text{ctx}}(h_i)]. \tag{12}$$

Here, I_i is the initial image, ℓ_i is the task name, and the image and text features come from frozen CLIP encoders. Subsets of these blocks define the text-only and text–state variants. Each context block is standardized using its training statistics. The decoder input separately uses the state normalizer associated with its fixed checkpoint.

Prediction objective. A shared trunk predicts categorical distributions $P_\phi^{\text{tr}}(\cdot \mid u_i)$ and $P_\phi^{\text{ap}}(\cdot \mid u_i)$ and a continuous rotation latent $\hat{z}_i^{\text{rot}} = g_\phi(u_i)$. The frozen EPMT encoder supplies the training labels. In addition to classification and latent regression, we supervise the effect implied by the

predicted factors. For each discrete factor, define the expected embedding $\mu_i^f = \sum_{k=1}^{K_f} P_\phi^f(k \mid u_i) c_k^f$, $f \in \{\mathrm{tr}, \mathrm{ap}\}$. The frozen decoder combines the expected embeddings and predicted rotation latent with the initial hand state, $\widehat{\tilde{y}}_i^{\mathrm{soft}} = D_\theta\left(\mathcal{N}_h(h_i), [\mu_i^{\mathrm{tr}}, \hat{z}_i^{\mathrm{rot}}, \mu_i^{\mathrm{ap}}]\right)$. For vectors in $\mathbb{R}^d$, define $\mathrm{MSE}(a, b) = d^{-1}\|a - b\|_2^2$. The contextual objective is

$$\mathcal{L}_{\mathrm{CC}} = \mathbb{E}_i\big[-\log P_\phi^{\mathrm{tr}}(z_i^{\mathrm{tr}} \mid u_i) - \log P_\phi^{\mathrm{ap}}(z_i^{\mathrm{ap}} \mid u_i) + \lambda_{\mathrm{rot}} \mathrm{MSE}(\hat{z}_i^{\mathrm{rot}}, z_i^{\mathrm{rot}}) + \lambda_{\mathrm{pose}} \mathrm{MSE}(\widehat{\tilde{y}}_i^{\mathrm{soft}}, \mathcal{N}_y(y_i))\big]. \tag{13}$$

Expected embeddings make the decoded-effect term differentiable w.r.t. the categorical predictions. The representation and decoder remain frozen; only the contextual predictor is optimized. The term couples factor prediction through the frozen decoder during training. At inference, discrete predictions use codebook entries rather than expected embeddings. We evaluate the contribution of decoded-effect supervision by setting $\lambda_{\mathrm{pose}} = 0$ with the architecture and training budget fixed.

To isolate robot translation prediction, the experiments additionally fit independent translation classifiers with cross-entropy using state, text, or both. These classifiers use the frozen translation vocabulary but have no rotation/aperture heads or decoded-effect objective. They are distinguished from the shared-trunk CC-EPMT model and from the multi-network controller implementation in Appendix 3.2.

3.7 Primitive Composition and Closed-Loop Skill Realization

The components above define a common interface from human motion to embodiment-specific execution. A manipulation skill is represented as a sequence of local primitive decisions rather than as a single retargeted trajectory. At each decision step, CC-EPMT uses the current observation, task context, and hand state to predict the next factorized primitive representation $\hat{Z}_t = (\hat{z}_t^{\mathrm{tr}}, \hat{z}_t^{\mathrm{rot}}, \hat{z}_t^{\mathrm{ap}})$. The embodiment-specific decoder then realizes this shared representation from the current robot state:

$$\hat{y}_t^e = (\mathcal{N}_y^e)^{-1}\left[D_{\theta_e}\left(\mathcal{N}_h^e(h_t^e), \widetilde{\hat{Z}}_t\right)\right]. \tag{14}$$

The decoded effect specifies the next embodiment-specific motion target, which is passed to robot execution stack. After execution, the robot is re-observed and the updated state conditions next primitive decision. Repeating this predict–realize–observe cycle composes local primitives into multi-step manipulation skills while allowing execution to adapt to the state reached by previous actions.

This interface also defines cross-embodiment skill transfer in M-PACT: human demonstrations provide the shared primitive vocabulary and, when available, primitive sequences, while embodiment adaptation changes only the mapping from these primitives to executable robot motion. Consequently, the same primitive representation can support recorded human-to-humanoid transfer as well as autonomous closed-loop primitive prediction. Sec. 4.3 evaluates these modes on the real humanoid, while detailed motion realization, sequential replay, inverse-kinematics evaluation, and execution procedures are provided in Appendix 3.2.

4 Experiments

We separately evaluate representation transfer, design choices, contextual prediction, and humanoid realization to isolate representation, prediction, and execution errors. Experimental setups are:

Datasets. EgoDex Hoque et al. (2025) provides representation training (27,411 episodes; 2.92M retained segments), HoloAssist Wang et al. (2023b) frozen transfer (35,087; 0.80M), and HumanoidEveryday Zhao et al. (2025) G1 adaptation (4,064; 96,241). We remove segments outside 5–90 frames or with negligible change in all factors. EgoDex uses a 90/10 episode split (292,806 test segments); robot decoder evaluation uses 85/15 (14,754 test segments), and robot prediction uses 70/30 (28,871 test segments). Candidate sets average three episode-level calibration/test partitions. Task generalization holds out 37/124 task names under three seeds against matched episode splits. Statistics are training-only and the source tokenizer stays frozen.

Evaluation Metrics. We report median endpoint translation, rotation, and aperture error; robot translation is scaled to human-equivalent units by $\sigma = 0.561$. Parentheses denote error divided by

median motion magnitude (below one beats zero change). We also report code use, top-1 accuracy, and 90%-coverage translation-set size. Replay uses interpolated-path and geodesic orientation errors; multi-run values average runs unless stated otherwise.

4.1 EPMT Reconstruction, Transfer, and Ablations

Transfer and adaptation. Table 1 evaluates whether the EPMT vocabulary learned on EgoDex remains useful across datasets and embodiments. With the source decoder, EPMT reconstructs EgoDex motions with normalized errors well below the zero-change baseline. When transferred to HoloAssist without decoder adaptation, all codes remain active and normalized errors remain below one, although the increased translation and rotation errors and reduced effective vocabulary indicate a distribution shift. On HumanoidEveryday, refitting only the target decoder yields low normalized translation and rotation errors while preserving full code utilization, supporting reuse of the learned vocabulary for a different embodiment. Aperture remains challenging: despite a small absolute error of 0.229 cm, its normalized error of 1.388 does not outperform zero change because aperture variations in this dataset are limited.

Table 1: Reconstruction with a shared EPMT representation. HoloAssist uses the frozen source decoder; HumanoidEveryday fits only a target decoder. Parentheses give error relative to median motion magnitude. All discrete codes remain active across datasets.

Metric	EgoDex	HoloAssist	HumanoidEveryday
Decoder	Source	Frozen	Refitted
Translation (cm)	2.658 (0.532)	3.123 (0.811)	3.442 (0.454)
Rotation (°)	0.931 (0.041)	2.732 (0.152)	0.901 (0.062)
Aperture (cm)	0.439 (0.493)	0.454 (0.525)	0.229 (1.388)
Translation codes used	16/16	16/16	16/16
Aperture codes used	4/4	4/4	4/4
Effective translation vocabulary	10.23	8.63	13.19

Ablations. To examine how temporal segmentation affects primitive-motion learning, we compare fixed-window, heuristic, and learned frontends through their downstream reconstruction performance. With the same tokenizer, learned segmentation gives the lowest translation (3.022 cm) and aperture (4.514 mm) errors (Table 2). Because frontends define different targets, this tests the full segmentation–tokenization pipeline; the learned row is a single-seed precursor to final SkillSeg. Additional EPMT representation ablations are provided in the supplementary.

Table 2: Segmentation and downstream reconstruction on each frontend's own segments.

Segmentation	Rate (s^{-1})	Segments	Trans. (cm)	Ap. (mm)
Fixed windows, 19 frames	1.616	51,337	3.451	4.655
Previous heuristic	1.742	55,353	3.188	5.220
Learned candidate	1.752	55,669	3.022	4.514

4.2 CC-EPMT Prediction

We evaluate shared-trunk CC-EPMT on human motion and independent target-domain translation classifiers on robot motion.

Human-motion prediction. To evaluate contextual primitive selection while separating prediction error from reconstruction error, we compare motions decoded from predicted factors with those reconstructed from recorded EPMT factors. Both use the same frozen decoder and are evaluated on 292,806 held-out EgoDex segments. All prediction variants share the same architecture and training budget; results are reported for one seed. Results from Table 3a show context improves translation accuracy from 36.37% to 40.42% and error from 5.307 to 4.952 cm, but remains far from recorded factors, especially in rotation (20.28° versus 0.93°; Table 3a). Prediction is therefore the main downstream error source. Removing decoded-effect supervision changes errors by at most 0.03° (0.13 accuracy points); one seed cannot establish an auxiliary-loss benefit.

Table 3: Human and robot contextual primitive prediction. Robot calibrated-set statistics average three partitions.

(a) CC-EPMT on held-out EgoDex; accuracy is percent.

Representation	Trans. (cm)	Rot. (°)	Ap. (mm)	Trans. top-1	Ap. top-1
Recorded EPMT	2.658	0.93	4.39	—	—
State-only prediction	5.307	21.65	9.19	36.37	57.54
Context prediction	4.952	20.28	8.46	40.42	59.07
Context, no effect loss	4.960	20.25	8.47	40.29	59.08
Training-majority label	—	—	—	35.01	54.67

(b) Robot translation prediction (T: text; S: state).

Predictor	Top-1 (%)	Size	Cov. (%)
Frozen human (T+S)	7.48	13.25	89.6
Robot (S)	30.64	9.12	89.9
Robot (T)	22.83	11.36	90.0
Robot (T+S)	33.98	8.35	89.9
Majority	22.56	—	—

Robot prediction and candidate sets. To evaluate whether human-derived motion primitives can be selected from robot-domain context, we train translation classifiers conditioned on state, language, or both. These models predict primitive codes directly, without images or decoder execution. Top-1 accuracy is evaluated on 28,871 test segments using one seed, while candidate-set results are averaged over three episode-level calibration/test partitions.

Combining language and state achieves 33.98% accuracy, compared with 30.64% for state alone, 22.56% for the majority baseline, and 7.48% for frozen human transfer (Table 3b). At approximately 90% coverage, target-domain adaptation reduces the mean candidate-set size from 13.25 to 8.35 out of 16 codes. However, language reduces held-out-task accuracy by 0.8 percentage points, providing limited evidence of improved task generalization. Coverage measures inclusion of the recorded code rather than feasibility or task success; candidate-set construction is detailed in the supplement.

4.3 Real World Experiments

We evaluate our framework on a real humanoid through four increasingly challenging settings: a) EPMT – Motion Reconstruction. We compress a previously executed robot manipulation episode into discrete primitive tokens and decode them back into robot motion, testing whether real manipulation can be faithfully represented through a compact token space, b) Human2Robot – Cross-Embodiment Transfer. Primitive decisions are extracted from egocentric human demonstrations and executed by the humanoid, with only the decoder adapted, testing whether human-derived motion primitives can transfer across embodiments, c) CC-EPMT (State-Only) – Closed-Loop Token Prediction. Rather than providing the token sequence, the robot predicts the next primitive from its current arm state, executes it, and re-observes, enabling autonomous closed-loop primitive composition without visual or task context, d) CC-EPMT (Full) – Context-Conditioned Manipulation. We further condition primitive prediction on the robot's camera observation and task instruction, testing whether visual and task context improves the selection and execution of appropriate motion primitives.

Together, these four settings progressively validate EPMT from motion reconstruction and cross-embodiment transfer to autonomous closed-loop prediction and context-conditioned manipulation. The results demonstrate how the same primitive token representation can bridge human motion and real-world humanoid execution across different levels of autonomy and context. Qualitative results can be found in `https://tinyurl.com/ykxkbu3z` and supplementary.

5 Conclusion

We presented M-PACT, a framework that bridges human demonstrations and humanoid manipulation through a shared primitive motion representation. By separating reusable motion structure from embodiment-specific realization, EPMT enables primitive reuse across datasets and adaptation to humanoid motion through a lightweight decoder, while CC-EPMT extends these primitives to context-conditioned prediction and closed-loop execution. Experiments across human and humanoid datasets, together with real-world humanoid demonstrations, validate the framework from motion reconstruction and cross-embodiment transfer to autonomous manipulation. Overall, our results demonstrate the potential of factorized primitive motion tokens as a reusable interface between scalable human motion data and embodiment-specific robot control.

REFERENCES

Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. RT-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chen, Jerry Bai, Jiacheng Wei, Boyu Chen, Xiaoji Zheng, Fan Wu, Minghao Yang, Tianrun Chen, Ruibo Li, Xiaoyu Yue, et al. Anyworld: Factorized egocentric world models for cross-embodiment generalization. *arXiv preprint arXiv:2608.29242*, 2026a.

Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *CVPR*, 2025.

Hongyi Chen, Tony Dong, Tiancheng Wu, Liquan Wang, Yash Jangir, Yaru Niu, Yufei Ye, Homanga Bharadhwaj, Zackory Erickson, and Jeffrey Ichnowski. Dexterous manipulation policies from rgb human videos via 3d hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026b.

Kristen Grauman, Andrew Westbury, Eugene Byrne, et al. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. In *CoRL*, 2025.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video, 2025. URL https://arxiv.org/abs/2505.11709.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

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 *ICRA*, 2025.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Yuanchuan Lai, Qing Gao, Ziyan Liang, Junjie Hu, and Zhaojie Ju. ObjRetarget: An object-aware motion retargeting framework with anthropomorphic arm constraints and polyhedral hand modeling. *IROS*, 2026.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. In *CoRL*, 2025.

Guangrun Li, Yaoxu Lyu, Zhuoyang Liu, Chengkai Hou, Jieyu Zhang, and Shanghang Zhang. H2r: A human-to-robot data augmentation for robot pre-training from videos. *arXiv preprint arXiv:2505.11920*, 2025.

Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *CoRL*, 2024.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *ICLR*, 2025.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. Pmlr, 2020.

Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, Linning Xu, Boyi Li, Huazhe Xu, and Jiangmiao Pang. Deximit: Learning bimanual dexterous manipulation from monocular human videos. In *Robotics: Science and Systems (RSS)*, 2026.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration^0. In *ICRA*, 2024.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos. *arXiv preprint arXiv:2606.19333*, 2026.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16939–16947. IEEE, 2025.

Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *ICRA*. IEEE, 2025.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023a.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20213–20224. IEEE, 2023b.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, et al. Ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation. *arXiv preprint arXiv:2510.08807*, 2025.

Han Zhou, Jinjin Cao, Liyuan Ma, Xueji Fang, and Guo-Jun Qi. Traj2Action: A co-denoising framework for trajectory-guided human-to-robot skill transfer. *arXiv preprint arXiv:2510.00491*, 2026.

Supplementary Material

1 Additional Implementation Details

Translation prediction sets. To measure how selectively context identifies translation primitives, we construct split-conformal sets for the translation factor only. For held-out calibration examples $\{(u_j, z_j^{\mathrm{tr}})\}_{j=1}^{n}$, let $s_j = 1 - P_\phi^{\mathrm{tr}}(z_j^{\mathrm{tr}} \mid u_j)$. At target miscoverage δ, let $\hat{q}$ be the $\lceil (n+1)(1-\delta) \rceil$-th smallest score, with $\hat{q} = +\infty$ when this index exceeds n. Then

$$\mathcal{S}_\delta(u) = \{k \in \{1, \dots, 16\} : 1 - P_\phi^{\mathrm{tr}}(k \mid u) \le \hat{q}\}. \tag{15}$$

Under exchangeability of calibration and test examples, this construction provides marginal inclusion probability of at least $1 - \delta$ for the encoded translation label. Our experimental implementation uses a conservative empirical-quantile variant and adds the argmax label if absent. We report empirical coverage together with set size, including after target-domain recalibration. These sets characterize translation-label uncertainty; they do not certify decoded-motion feasibility and are not used to rerank actions in the implemented controllers.

1.1 Humanoid Alignment and Calibration

We map humanoid trajectories into the shared hand-state convention of Sec. 3.2 using forward kinematics of the logged joint states. The mapping consists of coordinate alignment, arm-length normalization, wrist-orientation alignment, and aperture calibration.

Position alignment. Let x_{wrist} and x_{torso} denote wrist and torso positions obtained through forward kinematics, and let R_{torso} denote the torso orientation. The wrist position in the shared head-camera convention is

$$p^{\mathrm{cam}} = \frac{1}{\sigma} Q R_{\mathrm{torso}}^{\top} \left(x_{\mathrm{wrist}} - x_{\mathrm{torso}} \right), \qquad Q = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}, \tag{16}$$

where Q maps the robot torso frame (x forward, y left, z up) to the human head-camera convention (x right, y down, z forward). We use the measured arm-length ratio $\sigma = 0.561$ to normalize robot displacements to the scale of the human representation.

Orientation alignment. The corresponding wrist orientation is

$$R^{\mathrm{cam}} = Q \left(R_{\mathrm{torso}}^{\top} R_{\mathrm{wrist}} \right) A, \tag{17}$$

where A is a fixed hand-frame alignment derived from the robot geometry. Specifically, fingertip positions in the robot wrist frame define an orthonormal hand triad, which is aligned with the corresponding human-hand convention. The rotation representation used by EPMT is then obtained from the first two columns of R^{cam}.

Aperture calibration. Because the humanoid hand does not directly provide the thumb–index aperture used by the human representation, we compute the Euclidean separation between the thumb and index fingertips through forward kinematics and apply an affine calibration:

$$\alpha = \alpha_{\mathrm{lo}}^{\mathrm{hum}} + g \left(\| x_{\mathrm{thumb}} - x_{\mathrm{index}} \|_2 - d_{\mathrm{lo}}^{\mathrm{rob}} \right), \tag{18}$$

with $g = 0.7107$. This maps the measured humanoid fingertip range $[15.7, 157.7]$ mm to the human aperture range $[7.06, 107.98]$ mm.

For execution, the inverse aperture mapping is implemented through a one-dimensional finger-closure synergy. A scalar closure variable controls the index, middle, and thumb joints, and the resulting fingertip separation is tabulated and inverted by lookup. Separate synergies are constructed for the left and right hands to account for their mirrored joint conventions.

If an episode does not identify the acting arm, we select the arm with the larger summed absolute joint displacement over the episode.

1.2 SkillSeg Implementation Details

Input features. SkillSeg operates on 10-dimensional frame-level kinematic features. After Gaussian temporal smoothing with $\sigma = 2$ frames at 30 Hz, we construct

$$f_t = \left[v_t,\ \|v_t\|_2,\ \alpha_t,\ \dot{\alpha}_t,\ \|\dot{r}_t^6\|_2,\ \dot{r}_t^6[1{:}3]\right] \in \mathbb{R}^{10}, \tag{19}$$

where v_t is translational velocity and $r_t^6 = \pi_6(R_t)$. Derivatives are computed using central differences. The features contain motion but not absolute position and are standardized using statistics from the training episodes.

Motion gate. To prevent stationary portions of a trajectory from accumulating segment index, the predicted increment q_t is multiplied by a binary motion gate m_t before integration:

$$m_t = \mathbf{1}\left[\|v_t\|_2 > \theta_{\text{sp}} \ \vee\ |\dot{\alpha}_t| > \theta_{\text{ap}} \ \vee\ \|\dot{r}_t^6\|_2 > \theta_{\text{rot}}\right]. \tag{20}$$

We use $\theta_{\text{sp}} = 1.76$ cm/s, $\theta_{\text{ap}} = 8.8$ mm/s, and $\theta_{\text{rot}} = 31.8°$/s. These values are obtained by converting the static-motion thresholds used by the preprocessing pipeline to rates using the median segment duration. For episodes containing fewer than five moving frames, we use an all-ones gate.

Training objective. No boundary annotations are used. The cumulative quantity

$$c_t = \sum_{s=0}^{t} m_s q_s \tag{21}$$

acts as a differentiable segment index, and the triangular memberships in Sec. 3.3 produce soft segment embeddings and displacement targets. The target segment count in the rate loss is

$$N^\star = 1.742\,\frac{F}{\text{FPS}}, \tag{22}$$

where F is the trajectory length and $\text{FPS} = 30$. We use $\lambda_c = 0.25$, $\lambda_r = 1.0$, and $\lambda_e = 0.02$ in Eq. 2. The auxiliary VQ module contains 16 32-dimensional codes and is used only for learning the segmentation.

The output-layer bias is initialized to $\text{logit}(1.742/30)$ so that the initial expected accumulation approximately matches the target segmentation rate.

Architecture and inference. Each SkillSeg model is a dilated 1-D CNN with 64 hidden channels, kernel size 5, dilations 1, 2, and 4, GELU activations, and a 1×1 output layer. Models are optimized with Adam at learning rate 10^{-3} for 25 epochs using batches of eight episodes and gradient-norm clipping at 5.0.

At inference, we ensemble $M = 7$ independently trained models by averaging their frame-level increments before accumulation:

$$\bar{q}_t = \frac{1}{7}\sum_{j=1}^{7} q_t^{(j)}, \qquad c_t = \sum_{s=0}^{t} m_s \bar{q}_s. \tag{23}$$

Hard boundaries are extracted when c_t crosses successive integers. Crossings occurring within eight frames of the preceding boundary are discarded, giving a minimum segment duration of approximately 267 ms at 30 Hz.

2 Qualitative Results

Please refer to the anonymous link: `https://tinyurl.com/ykxkbu3z` for qualitative results. Each trial is presented using external- and head-camera views. The EPMT examples visualize recorded robot episodes across tasks involving closing, pressing, pouring, placing, and stacking. Human2Robot illustrates robot execution of primitives extracted from egocentric human demonstrations, while CC-EPMT presents closed-loop primitive prediction from robot state alone or jointly with visual observations and task instructions. These videos illustrate the progression from motion reconstruction to cross-embodiment realization and contextual primitive selection.

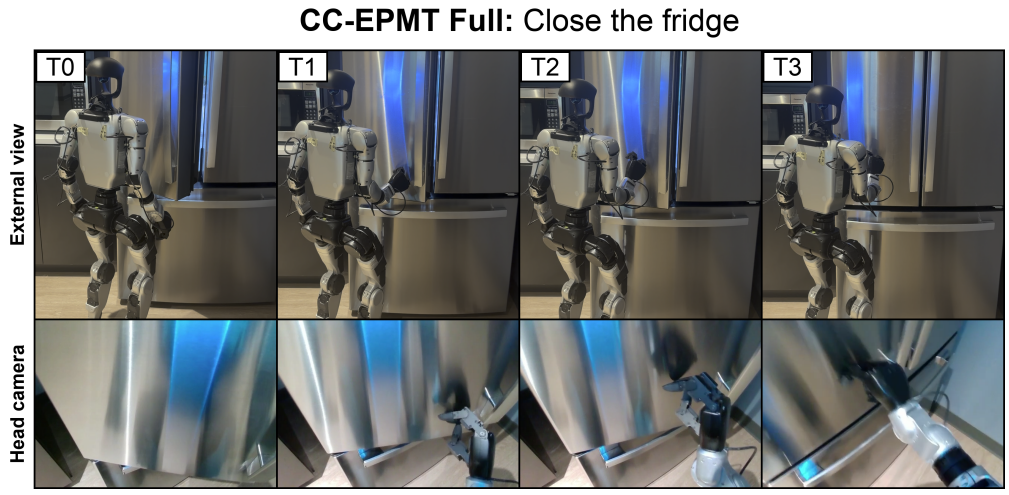

Figure 3: Close the fridge using full CC-EPMT.

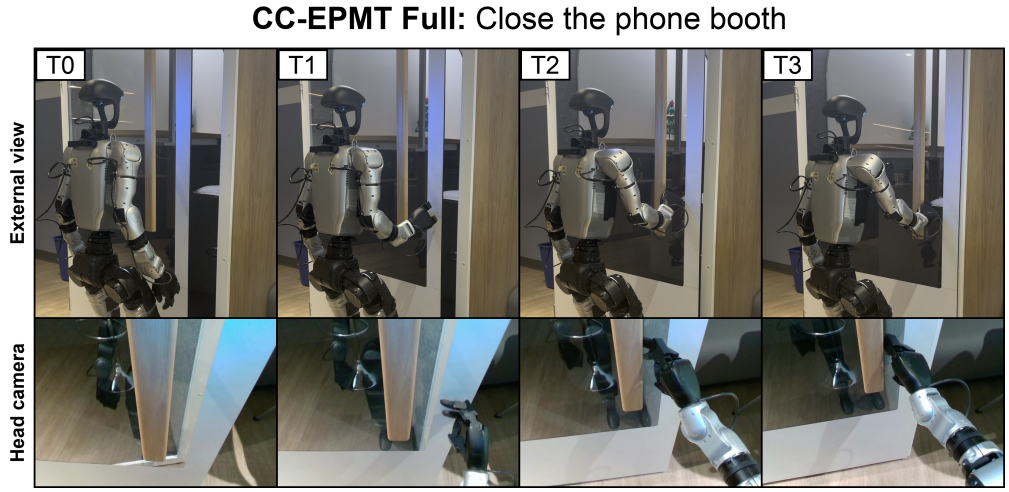

Figure 4: Close the phone booth using full CC-EPMT.

3 Additional Experiments

3.1 Ablations

3.1.1 EPMT

Translation representation. To assess whether the learned primitive tokens provide an efficient discrete representation of translational motion, we compare EPMT with alternative representations in terms of reconstruction error and bit rate. All translation-only models use the same 14,754 robot test segments, frozen segmentation, decoder, budget, and three seeds; they are not directly comparable to the full decoder. At four translation bits, EPMT yields 3.720 cm error, 13.0% below k-means and below FAST's interpolated 3.917 cm; FAST reaches 3.609 cm at 4.74 bits (Table S1). Rates exclude EPMT's aperture index and continuous rotation.

Representation ablations. To examine the roles of factorization, state conditioning, and continuous rotation modeling, we evaluate controlled variants of the EPMT representation. Variants use one reduced-budget seed. The joint model replaces separate codebooks with one 64-entry codebook, matching index bits but not all dimensions or parameters.

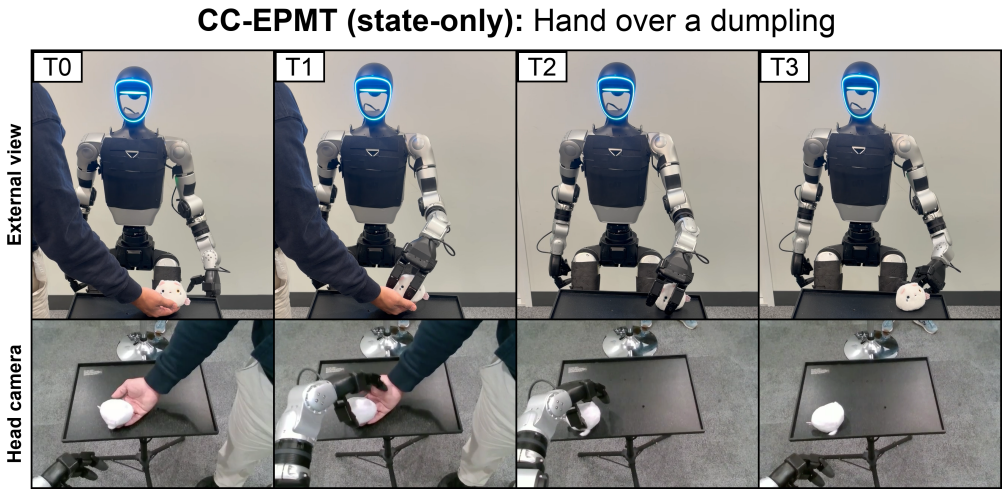

Figure 5: Handover the toy dumpling using state-only CC-EPMT.

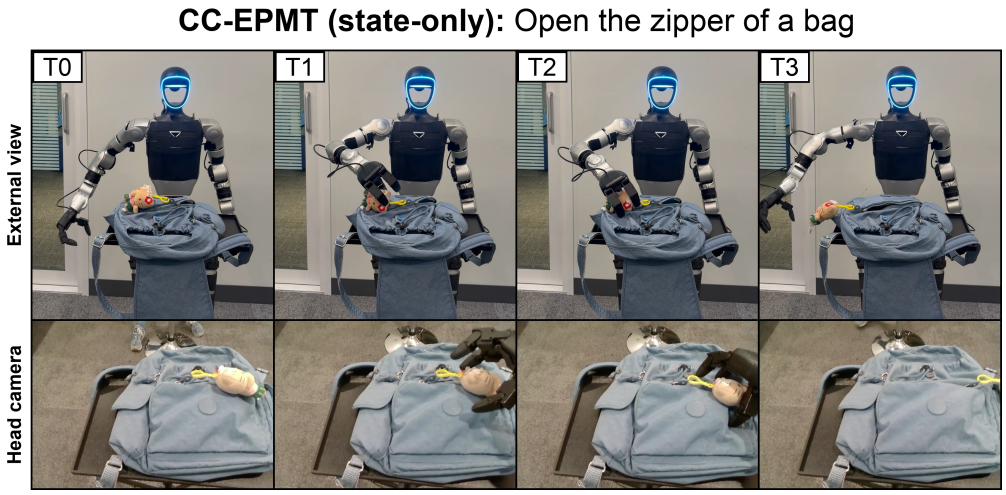

Figure 6: Unzip the bag using state-only CC-EPMT.

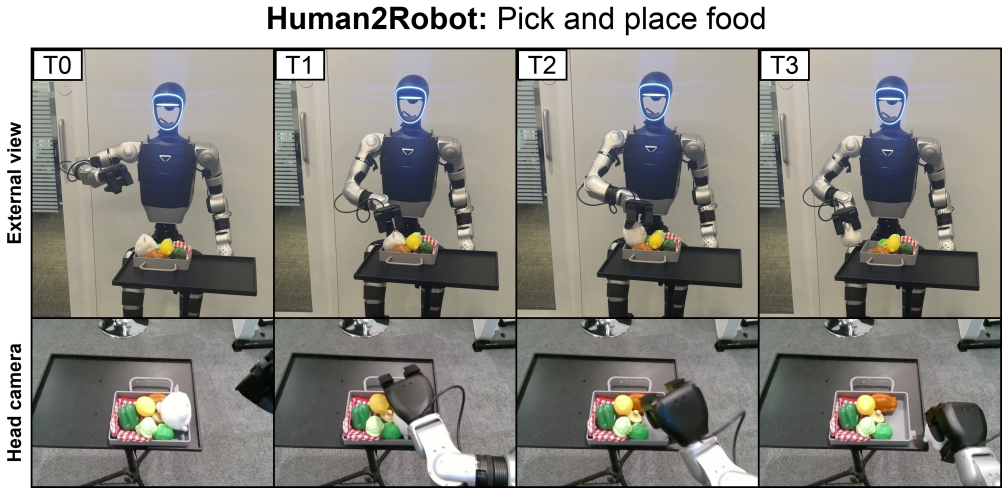

Figure 7: Pick food through human to humanoid skill transfer by utilizing EPMT.

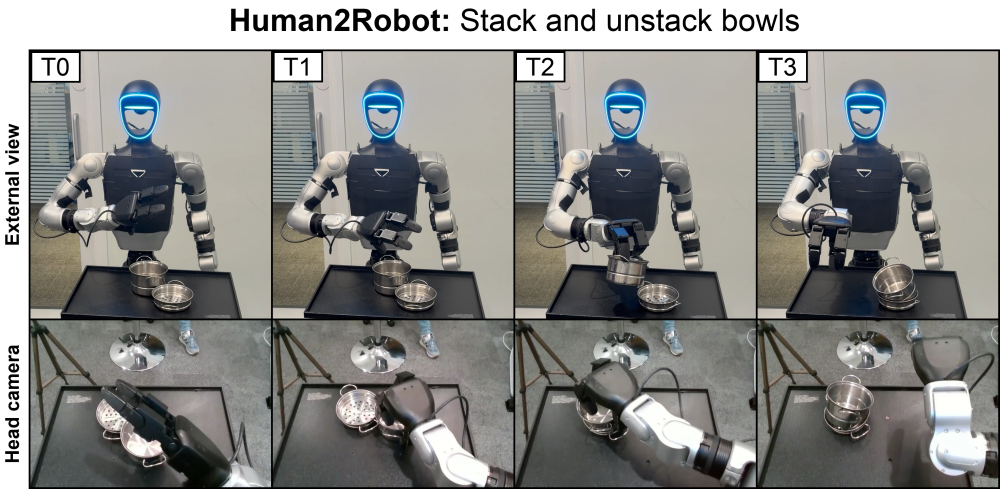

Figure 8: Stack bowls through human to humanoid skill transfer by utilizing EPMT.

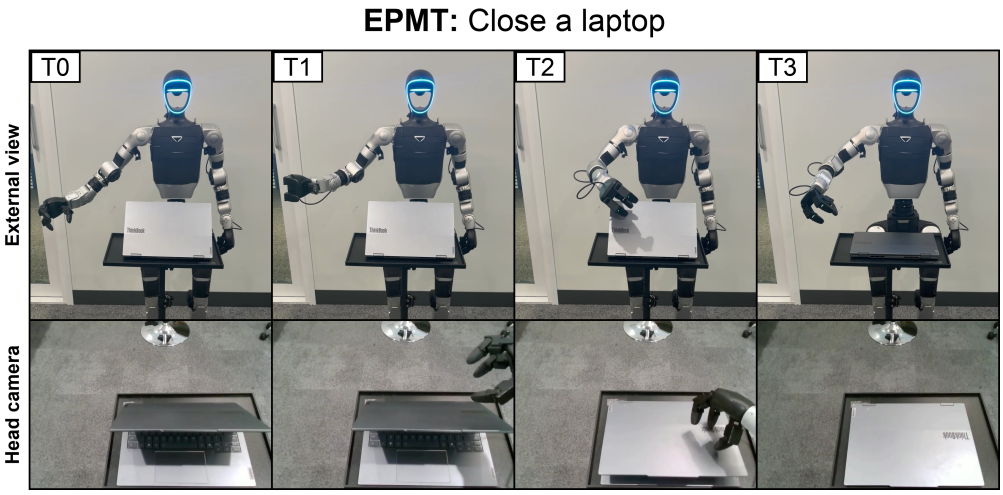

Figure 9: Close the laptop using EPMT encoding and decoding.

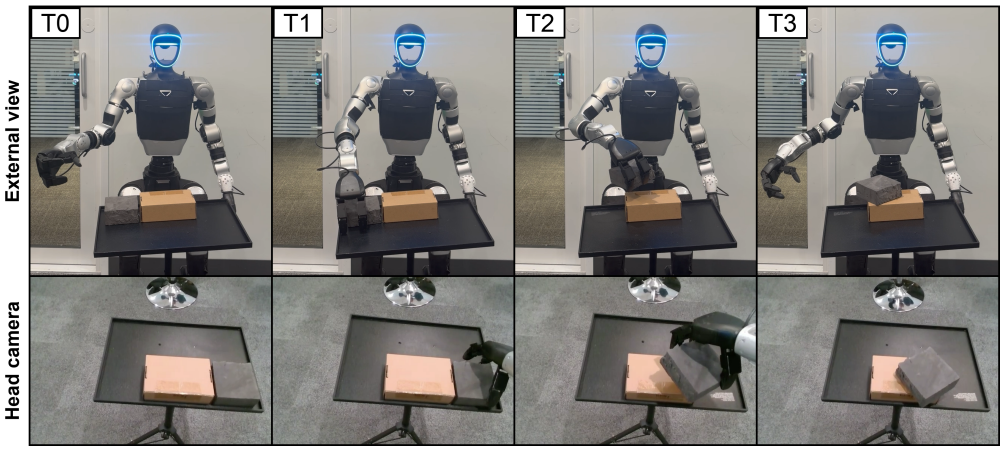

Figure 10: Stack cubes using EPMT encoding and decoding.

Table S1: Translation-only comparison with initial-state conditioning. Errors are means of per-run median errors over three decoder seeds. Bits count the translation representation only; FAST at four bits is interpolated.

Representation	Translation bits	Translation error (cm)
EPMT, 16 codes	4.00	3.720
k-means, projected centroid	4.00	4.278
k-means, direct 24D centroid	4.00	4.279
State only	0	7.757
FAST, measured	3.07	4.304
FAST, interpolated	4.00	3.917
FAST, measured	4.74	3.609

Joint encoding reconstructs best, so factorization offers addressable vocabularies rather than lower error here (Table S2). Removing state slightly worsens translation/aperture; the tested rotation quantizer raises error from 0.941° to 9.524°, supporting the continuous channel only against this quantizer.

Table S2: Representation ablations at one reduced budget and seed. Index bits exclude the continuous channel. Quantized rotation replaces that channel with 64 codes; it is not total-rate matched.

Variant	Index bits	Trans. (cm)	Rot. (°)	Ap. (mm)
Factorized EPMT	6	2.935	0.941	4.251
Without decoder state	6	2.993	0.764	5.034
Joint translation/aperture	6	2.815	0.953	2.639
Quantized rotation	12	2.957	9.524	4.375

3.1.2 CC-EPMT

Data requirements. M-PACT separates primitive selection—predicting which motion token to execute—from embodiment-specific realization, which decodes the selected token into robot motion. We therefore examine their target-domain data requirements separately to identify which stage presents the greater adaptation bottleneck. Across three seeds, selection accuracy remains near the 22.25% majority baseline through 250 training episodes before increasing to 24.16%, 30.22%, and 33.70% with 1,000, 2,000, and 2,844 episodes, respectively; the mean candidate-set size decreases from 12.28 to 8.27. In comparison, decoder adaptation recovers most of its reconstruction improvement with approximately 100 demonstrations. These trends suggest that primitive selection may require more target-domain data than embodiment-specific realization, although the two sweeps are not directly controlled.

3.2 Humanoid Motion Grounding and Realization

We provide additional details and component-level evaluations of human-to-humanoid motion realization. The decoder-grounding objective is defined in Sec. 3.5. Here, we evaluate the effect of decoder continuation, sequential realization of recorded human primitives, and local kinematic reachability. These experiments isolate realization from contextual primitive prediction and are not end-to-end task benchmarks.

Motion realization and sequential replay. Let $i(t)$ denote the recorded human segment used at replay step t. Given its frozen EPMT representation $\widetilde{Z}^{\mathrm{hum}}_{i(t)}$ and the current replay state h^{rob}_t, a decoder checkpoint θ_d predicts

$$\hat{y}^{\mathrm{rob}}_t = (\mathcal{N}^{\mathrm{rob}}_y)^{-1}\left[D_{\theta_d}\left(\mathcal{N}^{\mathrm{rob}}_h(h^{\mathrm{rob}}_t), \widetilde{Z}^{\mathrm{hum}}_{i(t)}\right)\right]. \tag{24}$$

Table S3: One-seed decoder grounding. Both evaluation sets informed checkpoint selection.

	Human			Humanoid		
Steps	Tr. (cm)	Rot. (°)	Ap. (mm)	Tr. (cm)	Rot. (°)	Ap. (mm)
0	5.312	2.19	5.05	3.406	0.90	2.19
25	3.995	2.33	4.75	4.294	2.07	3.76
100	3.413	2.02	4.72	4.250	2.04	4.61
400	3.084	1.43	4.58	4.852	2.09	3.85
1600	2.968	1.01	4.48	6.411	1.71	3.98
6400	2.862	0.76	4.41	8.552	1.47	4.18
Human decoder	2.663	0.94	4.353	8.069	3.58	5.198

Writing $\hat{y}_t^{\mathrm{rob}} = (\widehat{\Delta p}_t, \hat{\omega}_t, \widehat{\Delta\alpha}_t)$, the corresponding endpoint in the shared convention is

$$p_t^{\mathrm{end}} = p_t^{\mathrm{rob}} + \widehat{\Delta p}_t, \tag{25}$$

$$R_t^{\mathrm{end}} = R_t^{\mathrm{rob}} \operatorname{Exp}([\hat{\omega}_t]_\times), \tag{26}$$

$$\alpha_t^{\mathrm{end}} = \alpha_t^{\mathrm{rob}} + \widehat{\Delta\alpha}_t, \tag{27}$$

where $[\cdot]_\times$ forms the skew-symmetric matrix of a rotation vector. In sequential replay, the predicted endpoint becomes the state for the next decoder step, with aperture clipped to $[0, 0.20]$ m. IK does not feed back into this state update. Unless otherwise specified, replay uses recorded human representations rather than context-predicted factors, thereby isolating motion realization from primitive-selection error.

Decoded endpoints are mapped into the robot convention for inverse kinematics. Position and orientation residuals determine waypoint reachability while the solver enforces joint limits. The selected-pool analysis uses 5 mm position and 5° orientation residual thresholds. This evaluates local kinematic reachability of the supplied poses, without certifying collision avoidance, dynamics, or feasibility between waypoints.

Decoder grounding. We continue the robot decoder on 120,000 aligned human chunks using the grounding objective in Sec. 3.5, while keeping the EPMT representation and robot-fitted normalizers fixed. One run is evaluated on separate 40,000-human and 30,000-robot sets.

At the selected 400-step checkpoint, human translation error falls by 41.9%, from 5.312 to 3.084 cm, while humanoid translation error increases from 3.406 to 4.852 cm (Table S3). With longer continuation, human reconstruction continues to improve, but native humanoid fidelity generally degrades: at 6,400 steps, humanoid translation error reaches 8.552 cm. The other factors vary non-monotonically. These results show that adapting the decoder toward human-derived representations introduces a trade-off between human-motion grounding and preservation of the original humanoid mapping.

Sequential replay. To measure accumulated realization error independently of grounding and contextual prediction, we use recorded human factors with the robot-only decoder θ_{rob}. We replay eight-decision spans across 63 tasks without feedback and apply IK only afterward. Decoded and reference paths are interpolated at 30 Hz in the torso frame using linear position interpolation and orientation slerp.

The median across per-task median span maxima is 13.26 cm in position and 34.7° in orientation, with a path-length ratio of 0.84. Moreover, 88.9% of tasks exceed the 25° orientation gate. Thus, although individual primitives can be reconstructed accurately, errors accumulate substantially when recorded human primitives are replayed open loop over multiple decisions.

Kinematic reachability. We separately evaluate whether decoded targets are locally reachable by the humanoid. IK is applied to 56 quality-selected spans from five pick-and-place families while enforcing joint limits and the 5 mm/5° residual thresholds described above.

Overall, 285/312 decoded waypoints (91.3%) and 3,887/4,272 reference waypoints (91.0%) satisfy both residual thresholds, with 50/56 complete spans (89.3%) passing at every waypoint (Table S4). Thus, the selected decoded targets are usually locally reachable despite the accumulated open-loop

Table S4: IK on quality-selected pools. A span passes when every reference and decoded waypoint meets both thresholds.

Pool	Phase	Total	Position	Orientation	Both
1	Reference	1,870	1,801	1,801	1,801
1	Decoded	138	133	133	133
2	Reference	2,402	2,100	2,109	2,086
2	Decoded	174	152	153	152
Combined	Reference	4,272	3,901	3,910	3,887
Combined	Decoded	312	285	286	285

replay error. This analysis does not test collisions, dynamics, between-waypoint feasibility, or closed-loop task success.

Closed-loop execution interface. The real-world controller replaces recorded human representations with predicted primitive factors and conditions each decision on the latest measured hand state. At each step, the controller selects the argmax translation and aperture labels, predicts the continuous rotation latent, and passes these factors together with the current state through the robot-specific decoder. The resulting endpoint is executed through an interpolating trajectory generator, after which the robot is re-observed before the next decision.

The deployed controller bundles use independently fitted prediction networks and robot-specific decoders; they do not use the shared-trunk CC-EPMT objective or the human-grounded decoder evaluated above. Decoder grounding, open-loop replay, kinematic reachability, and closed-loop execution therefore probe complementary components of M-PACT rather than constituting a single end-to-end evaluation.

AI USE STATEMENT

AI tools were used solely to assist with polishing the writing and improving the clarity and readability of this paper. The technical content, methodology, experiments, analysis, and conclusions were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

Implementation details, including the model architecture, training procedure, experimental setup, and hyperparameter configurations, are provided in the main paper and supplementary material. The codebase will be released upon acceptance of this paper to facilitate reproduction of the reported results.