DexJoCo-X: Benchmarking Action Representations for Multi-Hand Dexterous Manipulation

Xiangwei Jiang[1], Yao Mu[2], Lixin Duan[1], and Wen Li[1,*]

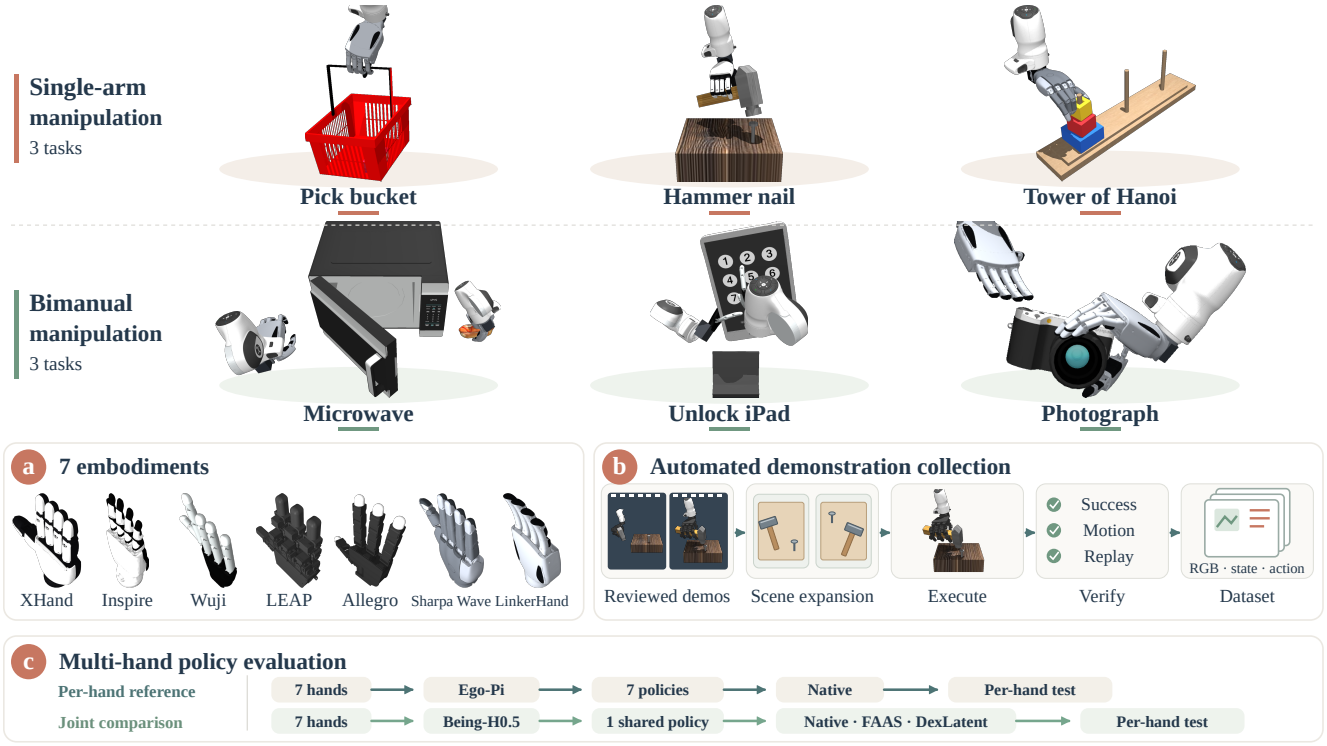


Fig. 1. **DexJoCo-X overview.** Three single-arm and three bimanual tasks provide a common setting for seven dexterous hands. (a) Embodiment models. (b) Reviewed demonstrations are adapted, executed, verified, and exported with native observations and actions. (c) The evaluation includes seven per-hand $\pi_{0.5}$ (Ego-Pi) multi-task policies as a Native-action reference and one joint Being-H0.5 policy trained with Native, FAAS, or DexLatent actions; every setting is tested on all applicable hand–task pairs.

***Abstract*— As dexterous hands proliferate, collecting data and training policies separately for every morphology becomes increasingly impractical. Scalable cross-embodiment learning therefore requires a unified representation that captures shared manipulation structure while preserving morphology-specific control. Differences in hands, tasks, datasets, and control interfaces prevent existing studies from isolating the effects of representation, pretraining, and architecture. We introduce DexJoCo-X, a benchmark and toolkit for controlled comparison across seven representative dexterous hands, six single-arm and bimanual tasks, and 2,100 balanced demonstrations. DexJoCo-X provides a matched multi-hand, multi-task protocol with common scenes, success criteria, and execution interfaces, redesigned glove-to-hand mappings, and an automated pipeline that expands reviewed demonstrations across randomized scenes. Using $\pi_{0.5}$, Ego-Pi, and Being-H0.5, we examine whether a shared action interface is sufficient for multi-hand learning. Expanding $\pi_{0.5}$ to an 80-dimensional bimanual output yields near-zero success. Ego-Pi preserves the pretrained action head through interleaved prediction and supports per-hand multi-task learning, but remains ineffective for seven-hand joint training. By contrast, Being-H0.5 combines cross-embodiment pretraining, a unified action space, and embodiment-aware experts, enabling one policy to control all seven hands. Within this architecture, function-aligned action slots achieve 47.7% mean success, compared with 47.0% for native coordinates and 33.1% for DexLatent. These results show that cross-embodiment representation depends on the entire learning system: action coordinates, pretraining, and architecture must jointly separate shared manipulation structure from embodiment-specific control.**

[1]University of Electronic Science and Technology of China, China.
[2]Shanghai Jiao Tong University, China.
[*]Corresponding author: Wen Li.

I. Introduction

Learning manipulation skills that can be shared across dexterous hands is essential for deploying general-purpose robots on diverse hardware. Diverse hand kinematics and actuation fragment demonstrations and policies, making separate training costly and limiting experience reuse. Multi-embodiment datasets and generalist policies, including Open

X-Embodiment/RT-X, Octo, OpenVLA, HPT, and the π series [1]–[6], establish a foundation for sharing robot experience. RT-2 [7] transfers vision–language knowledge to control. RDT-1B [8] uses unified action coordinates, and X-VLA [9] addresses embodiment heterogeneity through soft prompts, while Being-H0.5 [10] and Ego-Pi [11] connect human demonstrations with robot learning. Complementary approaches include diffusion policies with shared action latents [12], [13] and cross-hand reinforcement learning [14]. These efforts motivate action representations that link shared skills to embodiment-specific execution. Recent dexterous foundation-model frameworks, including UniDex [15] and XL-VLA [16], report promising multi-hand learning capabilities through different representations. Yet results obtained with different hands, datasets, and task protocols do not establish which approach best supports cross-hand manipulation under a common evaluation standard.

Existing benchmarks provide complementary capabilities: DexArt [17] studies articulated objects, Bi-DexHands [18] emphasizes bimanual control, DexJoCo [19] supplies functional tasks and demonstration tools, and DexVerse [20] broadens task and embodiment coverage. These benchmarks expand task coverage, but leave open which action representations best support cross-hand learning under matched conditions. A fair comparison requires the same hands, tasks, demonstrations, policy backbone, and success criteria. A shared control interface is also needed to evaluate new representations without rebuilding the hand-control pipeline. These needs motivate a dedicated benchmark for fair and accessible cross-hand evaluation.

To address these needs, we introduce **DexJoCo-X**, a unified benchmark and toolkit designed for fair comparison and straightforward integration of cross-hand policies (Fig. 1). Built on DexJoCo, it combines seven hand interfaces, six tasks, and 2,100 demonstrations with a reusable collection pipeline. We fine-tune $\pi_{0.5}$ [6] with the Ego-Pi protocol [11] as a per-hand multi-task reference, training one Native-action policy for each hand. Our central cross-hand comparison instead trains one Being-H0.5 policy [10] jointly on all seven hands under Native, FAAS, or DexLatent, and evaluates every hand on independent scenes. Our contributions are:

1) **A unified cross-hand benchmark.** Seven hands and six tasks share evaluation scenes, success criteria, and control interfaces for fair comparison of action representations under matched training conditions.
2) **A reusable demonstration collection toolkit.** Redesigned glove-to-hand mappings for all seven hands support human teleoperation, while automated scene expansion and trajectory verification scale demonstration collection.
3) **A balanced multi-hand demonstration dataset.** We collect 2,100 demonstrations spanning six single-arm and bimanual tasks, with 50 trajectories for each of the 42 hand–task pairs.

II. Related Work

A. Manipulation Benchmarks and Dexterous Tasks

RLBench [21], CALVIN [22], and LIBERO [23] benchmark arm–gripper manipulation; Robosuite [24] and ManiSkill2 [25] provide broader simulation infrastructure. Arm–gripper tasks do not test independent finger coordination.

Multi-finger hand benchmarks address bimanual coordination [18], articulated objects [17], deformable objects [26], and trajectory tracking across hands [27]. Grasp-focused resources evaluate pose generation [28], [29] and functional or language-guided grasping [30], [31]. DexJoCo [19] evaluates complete functional tasks, while DexVerse [20] expands task and embodiment coverage. Building on DexJoCo, we enable fair comparison of cross-hand action representations through balanced seven-hand, six-task demonstrations, shared evaluation scenes and success criteria, and reusable control interfaces.

B. Human Demonstrations and Automated Expansion

Human-to-robot hand mapping spans joint, Cartesian, task-oriented, and reduced-dimensional formulations [32]. DexPilot [33], Robotic Telekinesis [34], AnyTeleop [35], and DIME [36] use human motion to teach dexterous control. ALOHA [37] and UMI [38] broaden accessible demonstration capture, while LEAP Hand [39] provides an affordable dexterous embodiment.

DexMV [40] and ViViDex [41] learn from human video demonstrations. CyberDemo [42] augments simulated human demonstrations for real-world manipulation. MimicGen [43] expands demonstrations across contexts, and DexMimicGen [44] extends this approach to bimanual dexterity. Our toolkit supplies native demonstrations through seven glove mappings and verified scene expansion.

C. Cross-Hand Learning and Action Representations

Recent advances span action-chunking and diffusion policies [37], [45]–[48], vision–language–action models [3], [5]–[7], [9]–[11], and world-model-based control [49]–[51]. Cross-hand grasping shares contact points or maps [52]–[54], interaction geometry [55], [56], and morphology-conditioned eigengrasps [57]. For closed-loop cross-hand control, shared keypoint commands [58], retargeted human-hand actions [14], morphology-aligned graphs [59], and canonical hand descriptions [60] connect common skills to different kinematics. Particle-based world models [51] additionally share interaction dynamics.

Shared action spaces also enable learning from heterogeneous demonstrations. LAD [12] and OPFA [13] learn aligned action latents; AdvDex [61] aligns human and robot joints. UniDex [15] uses functional slots, while XL-VLA [16] learns cross-hand codecs. DexJoCo-X compares these representation principles under matched data, backbones, scenes, and control interfaces across seven hands.

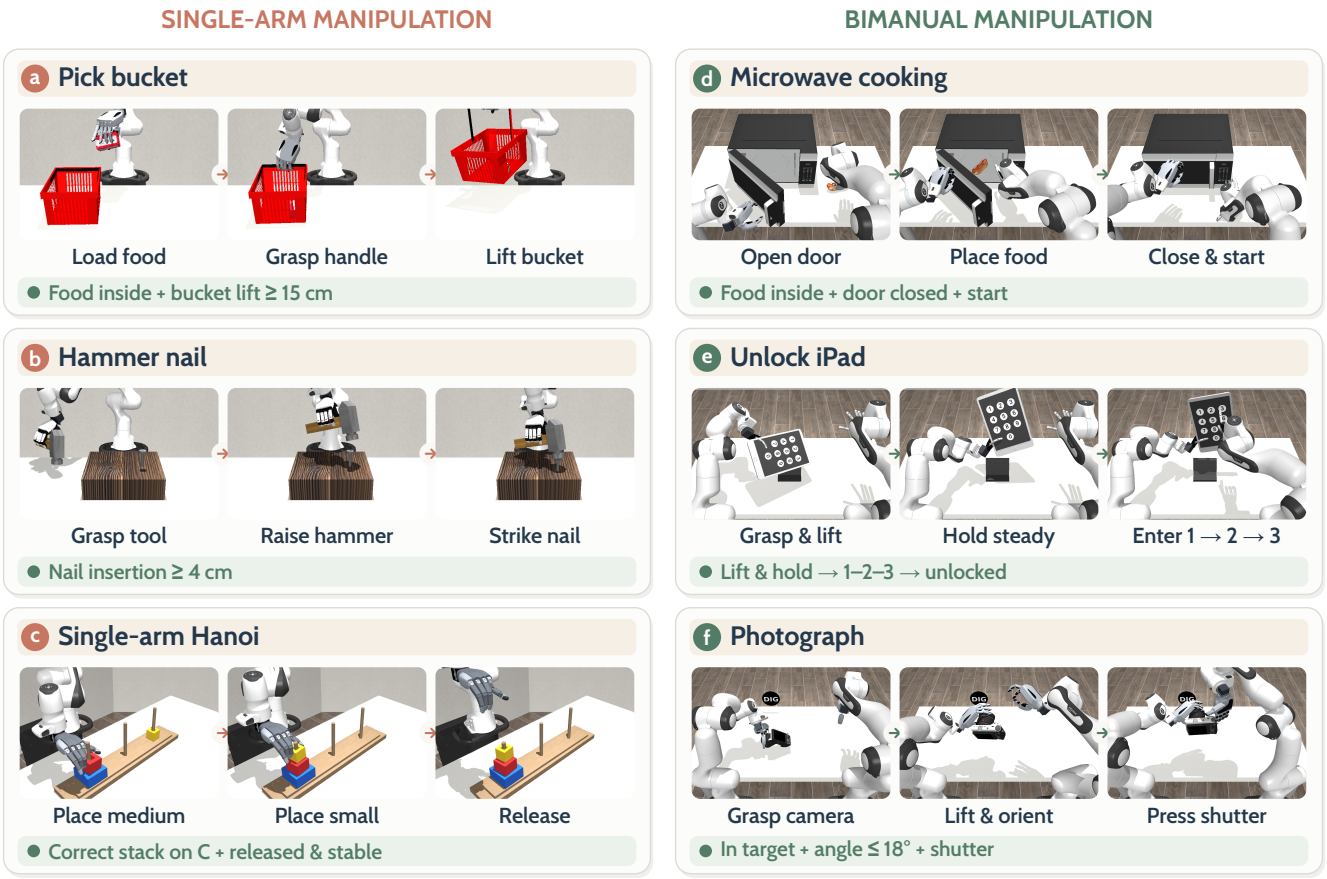


Fig. 2. Task stages and success conditions: (a–c) single-arm tasks; (d–f) bimanual tasks. Each sequence shows three representative stages; green labels summarize success criteria.

III. DexJoCo-X Benchmark

A. Embodiments and Task Scope

The seven hands are XHand, Inspire, Wuji, LEAP, Sharpa Wave, LinkerHand, and Allegro. Each retains its native kinematics, actuation, coupling, and joint limits; state and actuator dimensions are recorded separately.

Figure 2 shows three single-arm tasks—bucket lifting, nail hammering, and Hanoi—and three bimanual tasks—microwave cooking, iPad unlocking, and photography. Together they require stable grasping, directed tool use, sequential placement, and coordination between support and operation. Task success is determined by task-specific geometric, contact, and temporal criteria.

Tasks share initial scenes, objects, and success rules across hands, enabling paired comparisons of morphology and representation. Records retain successful and failed attempts.

B. Common Learning Interface and Native Control

To compare action representations without changing the underlying control problem, DexJoCo-X separates a common learning interface from each hand's native controller. At 20 Hz, every episode synchronizes RGB observations, task instructions, proprioceptive states, and robot commands. Arm control follows a shared convention based on world-frame translation and rotation-vector increments, whereas hand states and commands retain each embodiment's native joint order, limits, actuation, and coupling. This design preserves the control structure of each hand instead of imposing an artificial joint correspondence.

Native state and execution-command layouts are 2×31 and 2×28, with masks for valid coordinates and active arms. Encoded policy outputs have representation-specific layouts and are decoded to native commands for execution. Up to five 256×256 RGB streams provide front, side/ego, and wrist views, and each backbone comparison uses the same fixed view subset. Under this interface, Native, FAAS, and DexLatent share observations, arm commands, timing, and execution conditions, differing only in how hand actions are represented. Their performance can therefore be attributed more directly to the action representation and its interaction with the learning architecture.

C. Training Data and Coverage

The dataset contains 2,100 accepted demonstrations across seven hands and six tasks, with 50 trajectories per hand–task pair. All 2,100 demonstrations are used for policy training: each per-hand Ego-Pi policy receives 300 trajectories, while each joint Being-H0.5 policy receives the complete 2,100-trajectory set. Normalization statistics are computed from the

corresponding training demonstrations. Policy performance is measured through separate closed-loop evaluation rollouts.

IV. Demonstration Collection

A. Seven-Hand Teleoperation

The capture setup follows DexJoCo [19]: Rokoko gloves measure finger pose, and Vive trackers measure hand motion for arm control (Fig. 3). Each tracker attaches to the dorsal glove module through a printed connector. The operator observes the simulated task while the system records native robot states and commands.

We redesign the glove-to-hand mappings for all seven embodiments, including Allegro. Each mapping respects digit correspondence, coordinate order, motion direction, actuator limits, and joint coupling. Four-finger and five-finger hands use their respective digit configurations; dependent joints follow the native coupling model. The capture interface is

$$u_k^{\text{hand}} = R_h(g_k), \qquad u_k^{\text{arm}} = C_h(T_k), \tag{1}$$

where h indexes the hand embodiment and k the control step; g_k is the glove measurement, $T_k \in \mathrm{SE}(3)$ is the tracked wrist pose, and R_h and C_h are the calibrated hand and arm mappings. The outputs u_k^{hand} and u_k^{arm} are the native hand and arm commands. These mappings generate native demonstrations. Policy-action encoding is applied afterward, so all representations share the recorded trajectories.

B. Reviewed Sources and Scene Expansion

Figure 4 details the expansion workflow. A reviewed source pool provides candidate motions for each task. Small development trials identify failures in grasp acquisition, stage timing, tool alignment, and bimanual coordination. The resulting task-specific recipe is frozen before batch generation.

Each recipe combines scene-relative motions, stage transitions, and hand-specific parameters. Hammering varies hammer placement and nail position on the board; bimanual recipes coordinate support and operation. Attempts retain task, hand, scene seed, source, and recipe identifiers. Retries use approved sources under a fixed budget without changing the scene or success predicate.

C. Verification and Export

A candidate trajectory τ is accepted through

$$A(\tau) = S(\tau) \wedge Q(\tau) \wedge R(\tau) \wedge D(\tau), \tag{2}$$

where $A(\tau) \in \{0,1\}$ is the acceptance decision for candidate trajectory τ. The binary predicates S, Q, R, and D respectively check physical task success, motion quality, action replay, and data validity; $\wedge$ requires all four checks to pass. Motion checks encode task-specific requirements such as upright tool use during hammering. Replay verifies that saved commands reproduce the recorded trajectory. Export checks validate image availability, masks, timestamps, and metadata. Failed attempts and rejected exports remain in the audit record, and approved releases retain their source and configuration identifiers.

V. Hand-Action Representations

A. A Common Native Execution Interface

Let $u_h \in \mathbb{R}^{n_h}$ denote the valid native hand-command vector for hand h, where n_h is its dimension. A representation supplies an encoder e_h and execution decoder d_h: the policy predicts z, and the environment receives $\hat{u}_h = d_h(z)$. Arm commands retain the same interface across all three alternatives. Figure 5 summarizes both the coordinate meanings and the two policy interfaces. The $\pi_{0.5}$ (Ego-Pi) reference uses Native actions and interleaves left and right commands across consecutive 32-D tokens; Being-H0.5 predicts Native, FAAS, or DexLatent actions with one joint seven-hand policy.

B. Native Coordinates

Native preserves the original coordinate order of each hand. An insertion operator P_h pads the valid commands to a shared dimensionality, giving $z = P_h u_h$; the decoder selects the corresponding valid coordinates. A binary mask excludes unused entries. Coordinates preserve their native meanings; the representation shares tensor dimensions across hands.

C. Function-Aligned Slots

FAAS [15] organizes commands by functional role. We encode hand commands in its 32-slot layout, separately from arm commands. These slots belong to the policy representation; decoding restores the native hand-command vector for execution. For a valid native coordinate i, the adapter takes the form

$$z_{\sigma_h(i)} = s_{h,i} u_{h,i} + b_{h,i}, \tag{3}$$

where σ_h is the functional-slot assignment and $s_{h,i}$ and $b_{h,i}$ specify direction and offset. Execution uses the corresponding inverse for active coordinates; any dependent-joint expansion must follow the hand model. The adapter aligns functional roles while retaining hand-specific command values.

D. Learned Cross-Hand Latents

DexLatent follows the hand-specific codec formulation of XL-VLA [16]. An encoder E_h maps native hand coordinates to a shared latent, and a decoder D_h converts predicted latents to commands for hand h. Its fitting objective combines

$$\mathscr{L}_{\text{codec}} = \lambda_r \mathscr{L}_{\text{rec}} + \lambda_g \mathscr{L}_{\text{geom}} + \lambda_p \mathscr{L}_{\text{prior}}. \tag{4}$$

Here, $\mathscr{L}_{\text{rec}}$, $\mathscr{L}_{\text{geom}}$, and $\mathscr{L}_{\text{prior}}$ respectively measure native-command reconstruction, cross-hand fingertip-geometry alignment through differentiable forward kinematics, and latent-distribution regularization; λ_r, λ_g, and λ_p are their nonnegative weights. Geometry is evaluated on corresponding available digits. Codec checkpoints, fitting data, and calibration settings accompany each evaluated policy.



Fig. 3. Collection toolkit. (A) Glove-to-hand mappings enable teleoperation of seven hands using the DexJoCo capture setup [19]. (B) Reviewed demonstrations are expanded into randomized scenes, verified, and exported as RGB, state, and action records.

TABLE I

SEVEN-HAND MULTI-TASK SUCCESS RATES (%), AVERAGED OVER THREE INDEPENDENT 50-ROLLOUT EVALUATIONS PER HAND–TASK PAIR. TASK ENTRIES ARE ROUNDED TO INTEGERS; AVG. IS COMPUTED FROM THE UNROUNDED VALUES OVER 21 PAIRS PER PANEL.

(A) Single-arm tasks Order: Bucket / Hammer / Hanoi (%)

Policy (training)	Representation	XHand	Inspire	Wuji	LEAP	Sharpa Wave	LinkerHand	Allegro	Avg.
$\pi_{0.5}$ + Ego-Pi (per hand)	Native	62/69/20	29/12/31	53/47/3	20/25/21	37/31/11	66/29/4	66/16/8	31.5
Being-H0.5 (joint)	Native	87/46/1	99/74/72	77/85/22	50/39/56	76/63/6	86/58/62	67/35/41	57.2
	FAAS	93/47/7	97/74/69	89/85/22	65/36/17	70/37/0	93/63/38	53/23/64	54.3
	DexLatent	97/14/1	93/41/59	15/46/0	61/30/0	87/49/0	94/62/17	64/13/9	40.7

(B) Bimanual tasks Order: Microwave / Unlock iPad / Photo (%)

Policy (training)	Representation	XHand	Inspire	Wuji	LEAP	Sharpa Wave	LinkerHand	Allegro	Avg.
$\pi_{0.5}$ + Ego-Pi (per hand)	Native	19/38/36	45/9/14	40/1/11	2/1/22	49/1/32	82/36/9	26/0/14	23.2
Being-H0.5 (joint)	Native	1/31/52	6/29/26	100/10/30	0/21/88	59/18/71	53/49/66	33/0/27	36.7
	FAAS	42/64/63	1/44/35	99/1/21	8/29/79	31/8/80	47/79/64	9/1/55	41.0
	DexLatent	0/5/57	1/1/7	95/5/1	3/17/79	82/13/67	11/16/69	8/0/0	25.6

Overall Being-H0.5 average (seven hands, six tasks): Native 47.0%, FAAS 47.7%, DexLatent 33.1%.

VI. EXPERIMENTS

A. Training and Evaluation

We evaluate two multi-task training protocols. The $\pi_{0.5}$ [6] (Ego-Pi [11]) setting trains seven independent Native-action policies, one per hand, with each policy using the 300 demonstrations for that hand's six tasks. Being-H0.5 [10] trains one shared policy on all 2,100 demonstrations across seven hands and six tasks; separate runs use Native, FAAS, or DexLatent actions. Table II summarizes both settings. The three Being-H0.5 runs share demonstrations, image views, arm commands, sampling, and optimization budgets. Normalization uses valid training coordinates, and the DexLatent codec remains frozen during policy training.

The $\pi_{0.5}$ (Ego-Pi) setting therefore yields seven per-hand policies, each covering six tasks, whereas each Being-H0.5 representation yields one policy covering all 42 hand–task

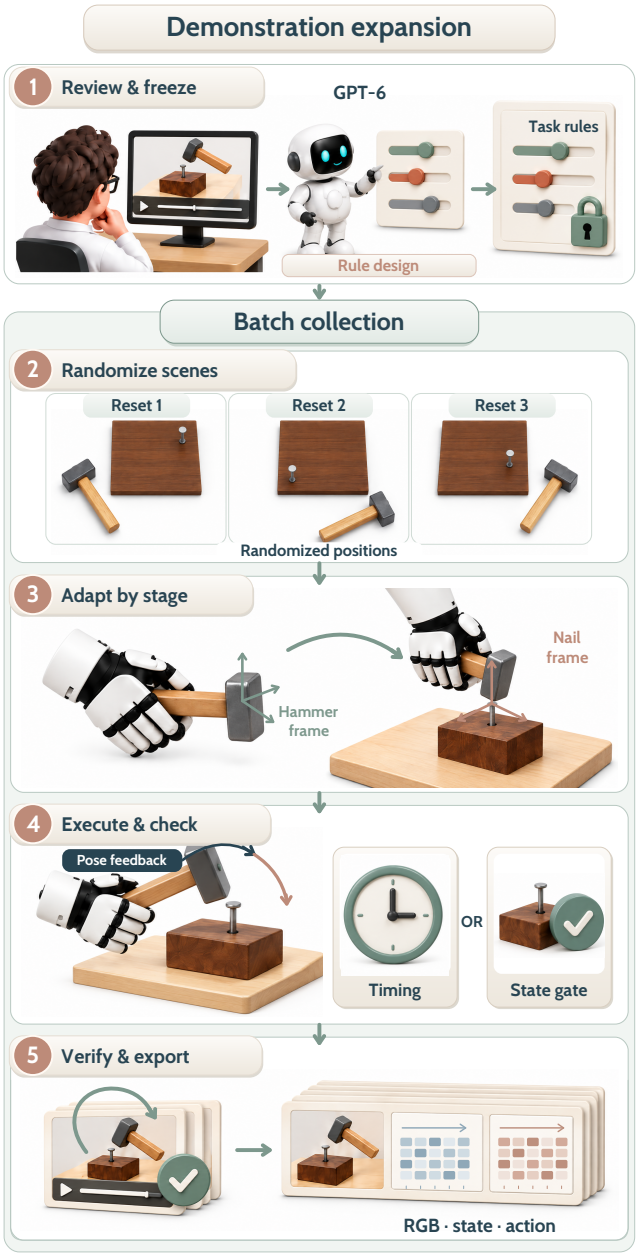


Fig. 4. Demonstration expansion illustrated with hammering. GPT-6 assists task-rule development; frozen rules drive batch collection. Randomized scenes are handled through stage-dependent reference frames, pose feedback, and timing- or state-based transitions. Verified trajectories are exported as RGB, state, and action records.

pairs. For a given hand, both protocols use the same 50 demonstrations per task. Training samples are drawn uniformly over the applicable frames; longer trajectories therefore contribute more frames despite balanced demonstration counts.

Each hand–task cell reports the mean of three independent 50-reset evaluations, giving 150 rollouts per cell and 6,300 per seven-hand configuration. Methods share reset sets disjoint from demonstration-generation seeds. Ego-Pi outputs 50 interleaved tokens for 25 control steps, whereas Being-H0.5 outputs 30-step chunks. Decoded native commands

TABLE II
MULTI-TASK TRAINING SETTINGS.

Setting	$\pi_{0.5}$ + Ego-Pi	Being-H0.5
Training scope	1 hand, 6 tasks	7 hands, 6 tasks
Policies trained	7	1
Adaptation	Full fine-tuning	Full fine-tuning
Updates	5,000	120,000
Learning rate	5×10^{-5}	10^{-4}
Warmup updates	5	2,000
LR schedule	Cosine	Cosine
Global batch	128	8
Prediction horizon	25 steps (50 tokens)	30 steps
Demonstrations	300 per policy	2,100

execute at 20 Hz with timestamped asynchronous replanning.

B. Success Metrics

For method m, hand h, task t, evaluation run r, and reset i, let $s_{m,h,t,r,i} \in \{0,1\}$ denote task success. We compute the hand–task success rate $\hat{p}_{m,h,t}$ and full-grid macro-average $\bar{p}_m$ as

$$\hat{p}_{m,h,t} = \frac{1}{150} \sum_{r=1}^{3} \sum_{i=1}^{50} s_{m,h,t,r,i}, \qquad \bar{p}_m = \frac{1}{42} \sum_{h=1}^{7} \sum_{t=1}^{6} \hat{p}_{m,h,t}. \quad (5)$$

Every hand–task pair is weighted equally. Success requires complete task execution, and each single-arm or bimanual score averages 21 pairs.

C. Action-Space Adaptation and Multi-Hand Results

Before adopting interleaved prediction, we tested a direct bimanual adaptation of $\pi_{0.5}$ that reserved 40 action dimensions per side and replaced its pretrained 32-D action projection with an 80-D output head. Fine-tuning this expanded interface produced near-zero task success. We therefore follow Ego-Pi [11]: each native arm–hand command, containing at most 28 valid values in our benchmark, is embedded in one 32-D token, and left and right commands alternate across the output sequence. This formulation preserves the pretrained action projection and supports 25 bimanual control steps with 50 tokens.

We also trained a single Ego-Pi policy by mixing demonstrations from all seven hands, but it performed substantially below the per-hand models. A common tensor layout alone therefore did not align the heterogeneous hand-control distributions. Being-H0.5 addresses this problem earlier in learning: its pretraining maps human MANO motion and data from 30 robot embodiments into semantically aligned action slots, while its Mixture-of-Flow architecture combines shared dynamics with embodiment-aware experts [10]. This pretraining and routing provide a stronger basis for joint multi-hand optimization than downstream data mixing alone, so we use seven per-hand Ego-Pi policies as the $\pi_{0.5}$ reference and a single Being-H0.5 policy for the joint representation comparison.

Table I shows comparable overall success for Native and FAAS under joint seven-hand training. FAAS performs better

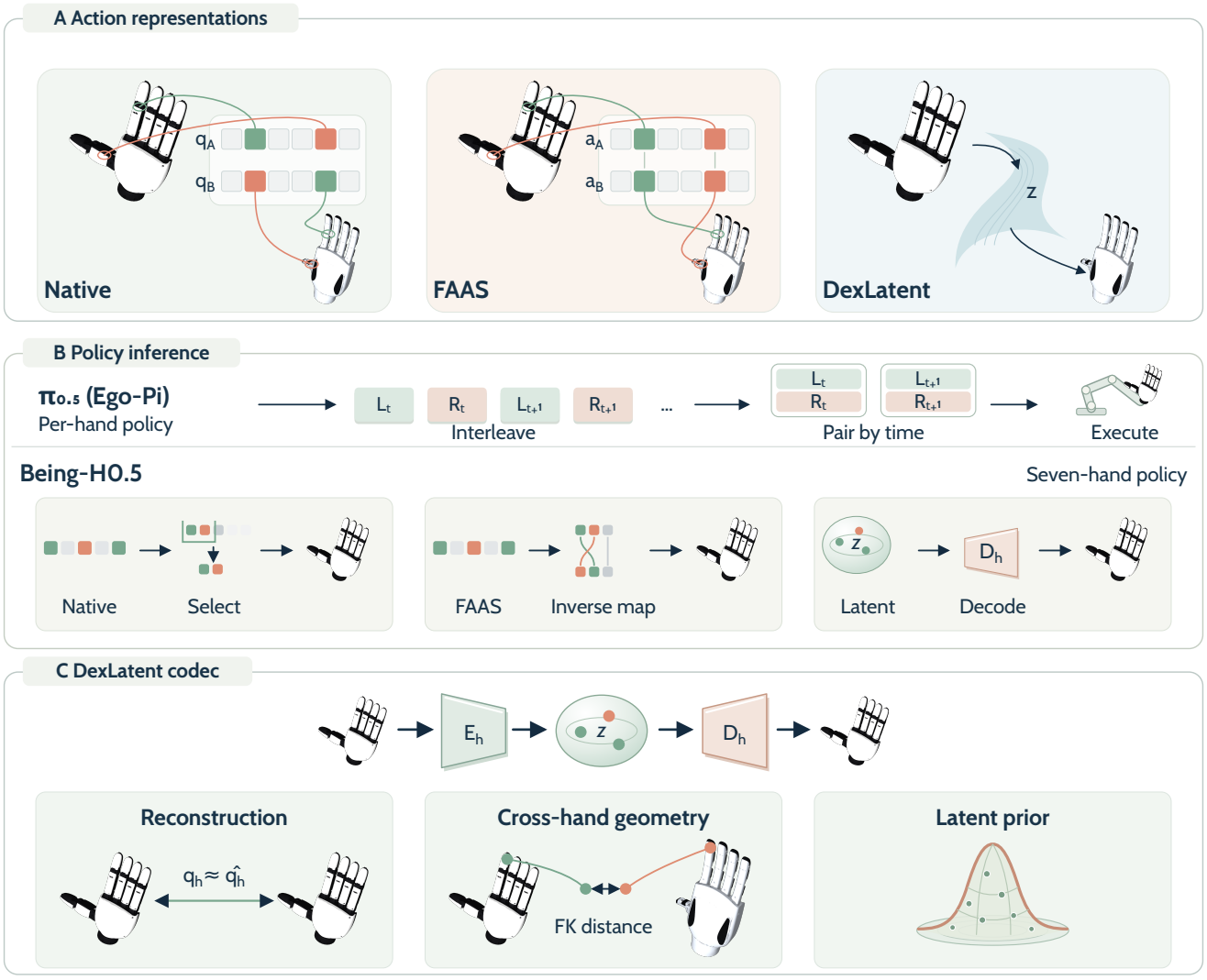


Fig. 5. **Hand-action representations and policy interfaces.** Top: Native coordinates, function-aligned slots (FAAS) [15], and DexLatent [16]. Middle (bimanual token example): $\pi_{0.5}$ (Ego-Pi) interleaves left/right Native actions across 32-D tokens, while joint Being-H0.5 predicts and decodes the selected representation. Bottom: DexLatent fitting uses reconstruction, cross-hand geometry, and latent-prior regularization.

on bimanual tasks, whereas Native performs better on single-arm tasks: functional alignment provides task-dependent benefits. DexLatent achieves lower success in both groups. All three representations perform worse on bimanual tasks, highlighting coordination as a common challenge.

VII. Conclusion

DexJoCo-X standardizes six-task evaluation across seven heterogeneous hands. Native and FAAS achieve comparable overall success, with advantages on single-arm and bimanual tasks, respectively. This tradeoff highlights task-dependent representation benefits, while bimanual coordination remains a shared challenge. Future work will extend evaluation to physical robots, broaden task coverage, and introduce held-out-hand zero-shot and one-shot protocols to assess transfer to unseen hands and adaptation from minimal demonstrations.

Acknowledgments

OpenAI image generation tools assisted the schematic illustrations.

References

[1] Open X-Embodiment Collaboration, A. O'Neill, A. Rehman, *et al.*, "Open X-Embodiment: Robotic Learning Datasets and RT-X Models," in *Proc. IEEE ICRA*, 2024.
[2] D. Ghosh, H. R. Walke, K. Pertsch, *et al.*, "Octo: An Open-Source Generalist Robot Policy," in *Proc. RSS*, 2024.
[3] M. J. Kim, K. Pertsch, S. Karamcheti, *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," in *Proc. CoRL*, 2025.
[4] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling Proprioceptive-Visual Learning with Heterogeneous Pre-trained Transformers," in *Adv. Neural Inf. Process. Syst.*, 2024.
[5] K. Black, N. Brown, D. Driess, *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024.
[6] Physical Intelligence, K. Black, N. Brown, *et al.*, "$\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[7] B. Zitkovich, T. Yu, S. Xu, *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *Proc. CoRL*, 2023.

[8] S. Liu, L. Wu, B. Li, *et al.*, "RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation," *arXiv preprint arXiv:2410.07864*, 2024.

[9] J. Zheng, J. Li, Z. Wang, *et al.*, "X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model," in *Proc. ICLR*, 2026.

[10] H. Luo, Y. Wang, W. Zhang, *et al.*, "Being-H0.5: Scaling Human-Centric Robot Learning for Cross-Embodiment Generalization," *arXiv preprint arXiv:2601.12993*, 2026.

[11] J. W. Kim, K. Wang, Z. Fu, S. Chen, C. Zhao, J. Lai, and C. Finn, "Ego-Pi: VLA Fine-Tuning for Ego-Centric Human and Robot Data," *arXiv preprint arXiv:2606.08107*, 2026.

[12] E. Bauer, E. Nava, and R. K. Katzschmann, "Latent Action Diffusion for Cross-Embodiment Manipulation," in *Proc. IEEE ICRA*, 2026.

[13] J. Mu, S. Yang, H. Bae, *et al.*, "One-Policy-Fits-All: Geometry-Aware Action Latents for Cross-Embodiment Manipulation," *arXiv preprint arXiv:2603.14522*, 2026.

[14] H. Yuan, B. Zhou, Y. Fu, and Z. Lu, "Cross-Embodiment Dexterous Grasping with Reinforcement Learning," in *Proc. ICLR*, 2025.

[15] G. Zhang, Q. Xu, H. Zhang, *et al.*, "UniDex: A Robot Foundation Suite for Universal Dexterous Hand Control from Egocentric Human Videos," in *Proc. IEEE/CVF CVPR*, 2026.

[16] G. Jiang, Y. Liang, J. Ye, *et al.*, "Cross-Hand Latent Representation for Vision-Language-Action Models," *arXiv preprint arXiv:2603.10158*, 2026.

[17] C. Bao, H. Xu, Y. Qin, and X. Wang, "DexArt: Benchmarking Generalizable Dexterous Manipulation With Articulated Objects," in *Proc. IEEE/CVF CVPR*, 2023, pp. 21 190–21 200.

[18] Y. Chen, T. Wu, S. Wang, *et al.*, "Towards Human-Level Bimanual Dexterous Manipulation with Reinforcement Learning," *arXiv preprint arXiv:2206.08686*, 2022.

[19] H. Wang, W. Zhao, X. Wang, *et al.*, "DexJoCo: A Benchmark and Toolkit for Task-Oriented Dexterous Manipulation on MuJoCo," *arXiv preprint arXiv:2605.16257*, 2026.

[20] Y. Yao, Z. Xu, T. Zhang, *et al.*, "DexVerse: A Modular Benchmark for Multi-Task, Multi-Embodiment Dexterous Manipulation," *arXiv preprint arXiv:2607.08751*, 2026.

[21] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The Robot Learning Benchmark & Learning Environment," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3019–3026, 2020.

[22] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7327–7334, 2022.

[23] B. Liu, Y. Zhu, C. Gao, *et al.*, "LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning," in *Adv. Neural Inf. Process. Syst.*, 2023.

[24] Y. Zhu, J. Wong, A. Mandlekar, *et al.*, "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning," *arXiv preprint arXiv:2009.12293*, 2020.

[25] J. Gu, F. Xiang, X. Li, *et al.*, "ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills," in *Proc. ICLR*, 2023.

[26] S. Li, Z. Huang, T. Chen, *et al.*, "DexDeform: Dexterous Deformable Object Manipulation with Human Demonstrations and Differentiable Physics," in *Proc. ICLR*, 2023.

[27] S. Dasari, A. Gupta, and V. Kumar, "Learning Dexterous Manipulation from Exemplar Object Trajectories and Pre-Grasps," in *Proc. IEEE ICRA*, 2023, pp. 3889–3896.

[28] R. Wang, J. Zhang, J. Chen, *et al.*, "DexGraspNet: A Large-Scale Robotic Dexterous Grasp Dataset for General Objects Based on Simulation," in *Proc. IEEE ICRA*, 2023.

[29] T. Zhong and C. Allen-Blanchette, "GAGrasp: Geometric Algebra Diffusion for Dexterous Grasping," in *Proc. IEEE ICRA*, 2025.

[30] J. Hang, X. Lin, T. Zhu, *et al.*, "DexFuncGrasp: A Robotic Dexterous Functional Grasp Dataset Constructed from a Cost-Effective Real-Simulation Annotation System," in *Proc. AAAI*, 2024, pp. 10 306–10 313.

[31] J. He, D. Li, X. Yu, *et al.*, "DexVLG: Dexterous Vision-Language-Grasp Model at Scale," in *Proc. IEEE/CVF ICCV*, 2025, pp. 14 248–14 258.

[32] R. Meattini, R. Suarez, G. Palli, and C. Melchiorri, "Human to Robot Hand Motion Mapping Methods: Review and Classification," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 842–861, 2023.

[33] A. Handa, K. Van Wyk, W. Yang, *et al.*, "DexPilot: Vision-Based Teleoperation of Dexterous Robotic Hand-Arm System," in *Proc. IEEE ICRA*, 2020.

[34] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube," *arXiv preprint arXiv:2202.10448*, 2022.

[35] Y. Qin, W. Yang, B. Huang, *et al.*, "AnyTeleop: A General Vision-Based Dexterous Robot Arm-Hand Teleoperation System," in *Proc. RSS*, 2023.

[36] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto, "Dexterous Imitation Made Easy: A Learning-Based Framework for Efficient Dexterous Manipulation," in *Proc. IEEE ICRA*, 2023.

[37] T. Z. Zhao, V. Kumar, S. Levine, *et al.*, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Proc. RSS*, 2023.

[38] C. Chi, Z. Xu, C. Pan, *et al.*, "Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots," in *Proc. RSS*, 2024.

[39] K. Shaw, A. Agarwal, and D. Pathak, "LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning," in *Proc. RSS*, 2023.

[40] Y. Qin, Y.-H. Wu, S. Liu, *et al.*, "DexMV: Imitation Learning for Dexterous Manipulation from Human Videos," in *Proc. ECCV*, 2022.

[41] Z. Chen, S. Chen, E. Arlaud, *et al.*, "ViViDex: Learning Vision-Based Dexterous Manipulation from Human Videos," in *Proc. IEEE ICRA*, 2025.

[42] J. Wang, Y. Qin, K. Kuang, *et al.*, "CyberDemo: Augmenting Simulated Human Demonstration for Real-World Dexterous Manipulation," in *Proc. IEEE/CVF CVPR*, 2024, pp. 17 952–17 963.

[43] A. Mandlekar, S. Nasiriany, B. Wen, *et al.*, "MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations," in *Proc. CoRL*, 2023.

[44] Z. Jiang, Y. Xie, K. Lin, *et al.*, "DexMimicGen: Automated Data Generation for Bimanual Dexterous Manipulation via Imitation Learning," in *Proc. IEEE ICRA*, 2025.

[45] C. Chi, Z. Xu, S. Feng, *et al.*, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," *Int. J. Robot. Res.*, vol. 44, no. 10–11, 2025.

[46] Z. Liu, Y. Wang, K. Wang, *et al.*, "Spatial-Temporal Aware Visuomotor Diffusion Policy Learning," in *Proc. IEEE/CVF ICCV*, 2025, pp. 7122–7131.

[47] Y. Ze, G. Zhang, K. Zhang, *et al.*, "3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations," in *Proc. RSS*, 2024.

[48] Y. Zhang, W. Dong, Y. Shi, *et al.*, "R3DP: Real-Time 3D-Aware Policy for Embodied Manipulation," *arXiv preprint arXiv:2603.14498*, 2026.

[49] Y. Du, S. Yang, B. Dai, *et al.*, "Learning Universal Policies via Text-Guided Video Generation," in *Adv. Neural Inf. Process. Syst.*, 2023.

[50] S. Zhou, Y. Du, J. Chen, *et al.*, "RoboDreamer: Learning Compositional World Models for Robot Imagination," in *Proc. ICML*, 2024.

[51] Z. He, B. Ai, T. Mu, *et al.*, "Scaling Cross-Embodiment World Models for Dexterous Manipulation," in *Proc. IEEE/RSJ IROS*, 2026.

[52] L. Shao, F. Ferreira, M. Jorda, *et al.*, "UniGrasp: Learning a Unified Model to Grasp with Multifingered Robotic Hands," *IEEE Robotics and Automation Letters*, 2020.

[53] P. Li, T. Liu, Y. Li, *et al.*, "GenDexGrasp: Generalizable Dexterous Grasping," in *Proc. IEEE ICRA*, 2023.

[54] Z. Wu, R. A. Potamias, X. Zhang, *et al.*, "CEDex: Cross-Embodiment Dexterous Grasp Generation at Scale from Human-like Contact Representations," *arXiv preprint arXiv:2509.24661*, 2025.

[55] Z. Wei, Z. Xu, J. Guo, *et al.*, "D(R,O) Grasp: A Unified Representation of Robot and Object Interaction for Cross-Embodiment Dexterous Grasping," in *Proc. IEEE ICRA*, 2025.

[56] H.-S. Fang, H. Yan, Z. Tang, *et al.*, "AnyDexGrasp: General Dexterous Grasping for Different Hands with Human-level Learning Efficiency," *arXiv preprint arXiv:2502.16420*, 2025.

[57] H. Zhang, K. Y. Ma, M. Z. Shou, W. Lin, and Y. Wu, "MachaGrasp: Morphology-Aware Cross-Embodiment Dexterous Hand Articulation Generation for Grasping," *arXiv preprint arXiv:2510.06068*, 2026.

[58] Q. She, S. Zhang, Y. Ye, R. Hu, and K. Xu, "Learning Cross-hand Policies of High-DOF Reaching and Grasping," in *Proc. ECCV*, 2024.

[59] Y. Wu, Y. Lin, W. Lao, *et al.*, "DexGrasp-Zero: A Morphology-Aligned Policy for Zero-Shot Cross-Embodiment Dexterous Grasping," *arXiv preprint arXiv:2603.16806*, 2026.

[60] Z. Wei, Y. Yao, and M. Ding, "One Hand to Rule Them All: Canonical Representations for Unified Dexterous Manipulation," in *Proc. RSS*, 2026.

[61] Z. Zhao, J. Wu, H. Liu, *et al.*, "AdvDex: Learning Dexterous Manipulation from Human Demonstrations via Joint-Aligned Actions and Adversarial Learning," *arXiv preprint arXiv:2608.14028*, 2026.