ADEPT: Accelerating Dexterity via Pre-Training and Post-Training with Reinforcement Learning

Anonymous Author(s)

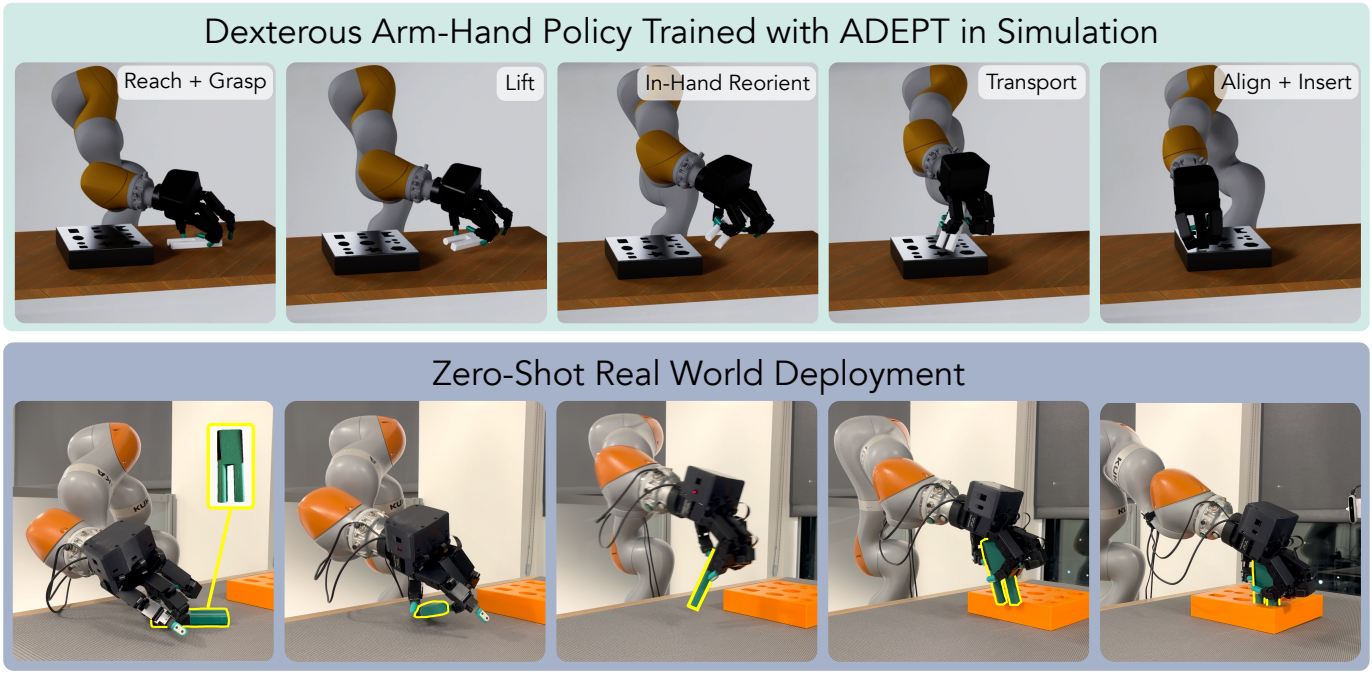

Fig. 1: ADEPT (Accelerating DExterity via Pre-Training) enables sim-to-real RL on the full joint configuration space of a 23-DoF dexterous arm–hand robot to reach, grasp, lift, reorient, transport, align, and insert directly from stereo RGB images, from diverse initial states, using a single policy.

***Abstract*—We introduce Accelerating DExterity via Pre-Training (ADEPT), a framework for learning high degree-of-freedom (DoF) arm–hand dexterity through large-scale reinforcement learning (RL). ADEPT pretrains a dexterous policy on a generic object reposing task, then post-trains downstream policies using this behavior as a prior for contact-rich tasks, enabling behaviors otherwise difficult to discover from scratch on multi-fingered robots. The pretrained policy generalizes and solves the reposing phase of downstream tasks zero-shot, but naive RL fine-tuning rapidly degrades this capability during transfer. We trace this to mismatched observation spaces, misaligned value estimates, and excessive policy drift, and address it with a post-training framework combining behavior-cloning distillation to adapt the actor, frozen-actor critic warm-up to recalibrate the value function, and conservative on-policy updates to bound policy drift. To safely exploit the full kinematic dexterity of the robot, we introduce a joint configuration-space Geometric Fabric controller between the RL policy and the robot. Finally, we distill the post-trained teacher into a stereo-RGB student that zero-shot sim-to-real transfers on a 23-DoF KUKA–Allegro arm–hand system, solving a contact-rich, long-horizon FMB peg insertion task requiring grasping, in-hand reorientation, and insertion across a wide range of initial states.**

I. Introduction

Dexterous manipulation with high degree-of-freedom (DoF) robotic systems, such as arms equipped with anthropomorphic multi-fingered hands, remains a major challenge for robot learning [1, 2, 3, 4]. These systems combine high-dimensional state and action spaces with contact-rich interactions, making useful behaviors hard to discover from sparse rewards. Even with massively parallel GPU simulation [5], RL policies trained from scratch for one task rarely transfer to another; each new task starts over, rediscovering the same low-level skills (reaching, grasping, lifting, reorientation) before any task-specific behavior emerges.

This motivates a different paradigm. Rather than training an RL specialist from scratch for each task, we first acquire a general-purpose dexterous foundation and then adapt it for specific downstream tasks [6, 7]. The policy's initialization in parameter space is as consequential as the downstream objective: a generic pretraining task that exercises the underlying motor primitives lands the network in a region from which task-specific behaviors build on, rather than overwrite, the pretrained skills [8].

We introduce Accelerating DExterity via Pre-Training (**ADEPT**), a framework that pretrains foundational arm–hand–object dexterity on a generic object reposing task [9] and post-trains it for downstream contact-rich tasks. This pretrained policy zero-shots the reposing segment of downstream tasks despite never seeing them during training, but adapting it to a contact-rich task is non-trivial: naive RL fine-tuning often collapses the pretrained behaviors before they can be specialized [10]. We trace this to mismatched observation spaces, misaligned value estimates, and excessive policy drift during early PPO [11] updates, and address it with a structured post-training procedure of actor distillation, critic warm-up, and conservative on-policy updates.

We finally distill the post-trained teacher into a stereo RGB student through a two-stage vision distillation curriculum and deploy it zero-shot on a 23-DoF KUKA iiwa7 + Allegro system, where it solves the Functional Manipulation Benchmark (FMB) [12] peg insertion task end-to-end as a single policy on two peg geometries spanning its difficulty range. FMB was set up for parallel-jaw grippers, which need external fixtures and multiple interaction stages to reorient; our system eschews both with a multi-fingered hand. Our contributions are:

- **ADEPT**, a framework for pretraining foundational arm–hand–object dexterity on a generic reposing task and post-training it for contact-rich downstream tasks, making full-DoF dexterous state-based RL tractable.
- A novel **full joint configuration space (Cspace) geometric fabric** [13] that gives the policy the full kinematic dexterity of a 23-DoF arm–hand system while enforcing hardware safety, unlike prior fabric-guided policies [3, 14] restricted to a low-dimensional PCA grasp subspace.
- A **structured post-training recipe** and **two-stage vision distillation curriculum** enabling zero-shot sim-to-real on a contact-rich, long-horizon insertion task—the *first demonstration* of pick-reorient-insert with arm–hand systems via sim-to-real RL.

II. Related Work

Sim-to-Real Dexterous Manipulation. Sim-to-real dexterous manipulation has converged on large-scale RL with domain randomization [15], parallel GPU physics [5], and Automatic Domain Randomization (ADR) [16, 1]. DeXtreme [2] scales this to agile in-hand cube reorientation, and follow-ups [17, 4, 18, 19] generalize to diverse in-hand rotation. DextrAH-G/RGB [3, 14] use geometric fabrics for sim-to-real grasping, and OmniReset [20] diversifies reset states to elicit emergent dexterity. Existing sim-to-real RL with hands either starts with an object in-hand or omits object–environment interaction; contact-rich, long-horizon arm–hand–object–environment tasks remain largely unexplored. ADEPT enables sim-to-real RL for multi-stage FMB peg insertion with behaviors beyond grasping, such as in-hand reposing and insertion.

RL Pretraining and Transfer. Reusing pretrained behavior to accelerate downstream RL is well studied: unsupervised RL benchmarks [6] formalize pretrain-then-finetune, and behavior priors are fine-tuned online with behavior cloning, KL regularization, or calibrated value learning [21, 22, 7] to counter the distribution shift that destabilizes naive fine-tuning. Shenfeld et al. [10] show forgetting is governed by the KL between fine-tuned and base policies, and EWC [8] constrains updates around the pretrained solution. ADEPT instead structurally preserves the prior for high-DoF dexterous policies through actor distillation, critic warm-up, and conservative PPO that keep the policy local to its pretrained initialization.

Geometric Fabrics. Geometric fabrics [13] provide a smooth, second-order action prior with stability guarantees and built-in collision and joint-limit avoidance [23, 24, 25]. DextrAH-G/RGB [3, 14] use fabrics for sim-to-real grasping but restrict hand control to a 5D PCA subspace plus a 6D palm pose target, limiting finger coordination. ADEPT instead drives the fabric in the full 23-DoF joint configuration space, exposing the full kinematic dexterity of the arm and hand to the policy.

III. Methodology

ADEPT consists of three stages (Fig. 2): (1) pretraining a dexterous policy in simulation on a generic object reposing task; (2) transferring the pretrained behavior through distillation and adapting it via massively parallel on-policy RL post-training while preventing catastrophic forgetting; and (3) teacher–student distillation of the task specialist into a stereo RGB student deployable zero-shot in the real world. We distinguish the phases by the nature of interaction: pretraining focuses on arm–hand–object manipulation, while downstream adaptation introduces arm–hand–object–environment interaction.

A. Problem Formulation

We cast dexterous manipulation as a discrete-time MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$ with action space $\mathcal{A} = [-1, 1]^{n_q}$ ($n_q = 23$), and learn a policy $\pi_\theta(a_t \mid o_t)$ maximizing the expected discounted return via PPO [11] with an asymmetric actor–critic. ADEPT operates over two MDPs, $\mathcal{M}_{\text{pre}}$ and $\mathcal{M}_{\text{post}}$, that share $\mathcal{A}$ but differ in observation space, dynamics, and reward; in particular, o^{post} extends o^{pre} with task-specific signals, the receptacle pose $\mathbf{p}_{\text{rec}}$ and object–receptacle contact forces $\mathbf{f}_{\text{or}}$. We first solve $\mathcal{M}_{\text{pre}}$ to obtain $(\pi_{\text{pre}}, V_{\text{pre}})$, then bootstrap from it to learn $(\pi_{\text{post}}, V_{\text{post}})$ on $\mathcal{M}_{\text{post}}$.

B. Pre-Training Foundational Dexterity

We pretrain a dexterous teacher actor–critic with PPO on a generic object reposing task. Each episode spawns one of 16 primitive shapes (cylinders, cuboids, spheres, cones) at a randomized scale on the table, and the policy must reach, grasp, lift, reorient in-hand, transport, and repose it to a sampled goal pose. Following [1, 2], we use ADR as an online curriculum that advances the goal and environmental complexity as success improves, Population-Based Training [26, 9] to search over PPO hyperparameters, and ADR-anneal gravity from 0 to $-9.81\,\text{m/s}^2$. This encourages reusable dexterous skills (precise grasps, coordinated finger motion, in-hand reorientation) that solve the reposing segment of downstream tasks zero-shot.

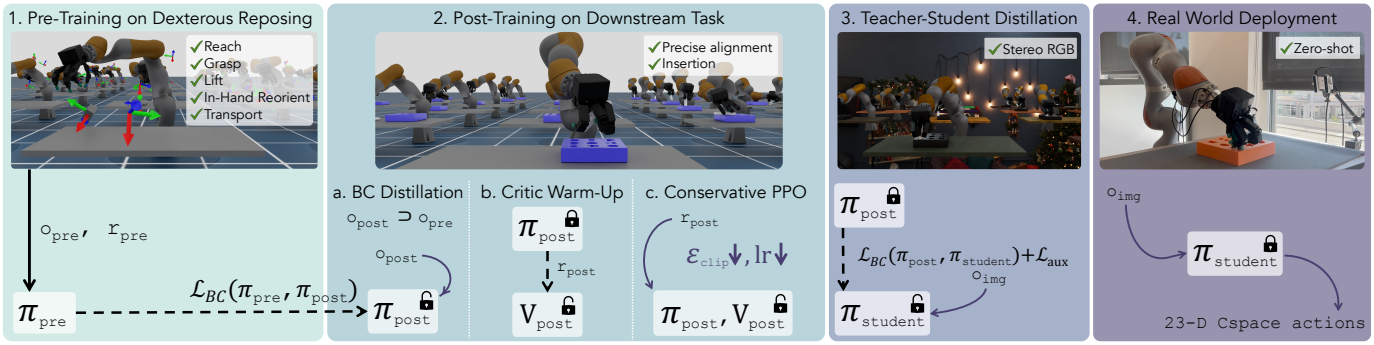

Fig. 2: **ADEPT Overview.** (1) Pre-train $(\pi_{\text{pre}}, V_{\text{pre}})$ via PPO on a generic reposing task. (2) Post-train into $(\pi_{\text{post}}, V_{\text{post}})$ for the downstream contact-rich task via BC distillation, frozen-actor critic warm-up, and conservative PPO. (3) Distill π_{post} into a stereo RGB student π_{student}. (4) Deploy π_{student} zero-shot on the real robot.

We use object point clouds as the object observation, which generalizes better to downstream tasks.

C. Post-Training Dexterous RL Specialists

Challenges in transfer. Directly fine-tuning the pretrained policy on a downstream task with standard RL rapidly degrades its behavior: it loses the ability to zero-shot the reposing segment within a few updates (Fig. 3, inset), even when zero-shot performance was initially strong, indicating the failure arises during training rather than from insufficient pretraining. We attribute this to mismatches under transfer: (1) a changed reward function, (2) extra downstream-only observations ($\mathbf{p}_{\text{rec}}$, $\mathbf{f}_{\text{or}}$), (3) poor value/advantage estimates, and consequently (4) large policy updates.

Structured RL adaptation. We frame downstream learning as layering new behaviors on top of the pretrained prior rather than relearning from scratch, transferring $(\pi_{\text{pre}}, V_{\text{pre}})$ in three steps. **(1) BC actor distillation:** distill π_{pre} into a new downstream actor π_{post} with the downstream observation space via supervised imitation. **(2) Critic warm-up:** freeze π_{post} and train a fresh critic V_{post} on rollouts from the frozen actor under the downstream reward, aligning value estimates before any policy update. **(3) Conservative PPO:** unfreeze and jointly update $(\pi_{\text{post}}, V_{\text{post}})$ conservatively (actor LR 1e-3 $\rightarrow$ 1e-5 with linear decay, clip ϵ: 0.2 $\rightarrow$ 0.05, critic LR 5e-5). All three steps run against the ADR-annealed goal path, pinned at ADR 20 for (1)–(2) and advancing toward the insertion target during (3). We find every stage critical for stable transfer.

D. Joint Configuration Space Geometric Fabrics

We place a geometric fabric [13] between the policy and the robot to enforce hardware constraints (joint limits, self- and environmental collision), provide a smooth second-order action prior that shapes RL exploration, and guarantee the same low-level controller in simulation and on the real robot. A fabric is an autonomous second-order system on configuration space, $\mathbf{M}_f(\mathbf{q}_f, \dot{\mathbf{q}}_f)\,\ddot{\mathbf{q}}_f + \mathbf{f}_f(\mathbf{q}_f, \dot{\mathbf{q}}_f) + \mathbf{f}_\pi(\mathbf{a}) = \mathbf{0}$, where $\mathbf{f}_f$ collects autonomous geometric and dissipative terms and $\mathbf{f}_\pi(\mathbf{a})$ is the policy-driven forcing term. We drive the fabric in the full joint configuration space: the policy outputs $\mathbf{a}_t \in [-1, 1]^{n_q}$ ($n_q = 23$), interpreted as per-joint relative deltas mapped to a Cspace target consumed by $\mathbf{f}_\pi$. This exposes the full 23-DoF kinematic dexterity to the policy while retaining the fabric's safety and stability guarantees, and minimizes the controller gap since the same fabric runs in simulation and on hardware.

E. Teacher–Student Distillation

We distill the state-based teacher into a stereo RGB student that consumes only proprioception, fabric state, and two RGB images, deployable zero-shot on the real robot. The student is trained with DAgger [27] using a ResNet backbone with a DextrAH-RGB-style [14] unfreeze schedule. Since the dominant insertion failure is misperceiving the peg's orientation, we add an 8-keypoint pose head supervised against the simulator pose, giving the objective $\mathcal{L} = \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{aux}}$. Mirroring our RL pipeline, a **two-stage curriculum** isolates perception from policy learning: (1) *student pretraining* re-tasks the reposing teacher to a perception-heavy surrogate (lift and reorient the peg upright) so $\mathcal{L}_{\text{aux}}$ shapes the encoder; (2) *vision distillation* from the downstream teacher, initialized from Stage 1, lets $\mathcal{L}_{\text{BC}}$ refine contact-rich policy on an already-competent encoder. Both stages use aggressive randomization of physics, object wrenches, visuals, and sensor noise.

IV. Experiments

We evaluate whether pretraining yields a reusable prior, whether structured post-training makes downstream insertion RL trainable, and whether the resulting teacher distills into a vision policy that transfers zero-shot to the real robot.

A. Simulation Experiments

Does the pretrained teacher generalize to unseen objects? Evaluated on the reposing task with no further training, the teacher matches or slightly exceeds its in-distribution success rate (0.73 over 16 primitives) on both out-of-distribution sets, reaching 0.76 on the FMB pegs and 0.77 on 152 VisDex [4] objects, despite training only on primitives.

How far can the teacher zero-shot the downstream task? We evaluate the teacher on FMB peg insertion across ADR

TABLE I: Zero-shot success rate (%) of the pretrained reposing teacher on FMB peg reposing across ADR levels, over 1024 episodes.

ADR	0	5	10	15	20	25	30	35	40	45	50
SR	98.6	94.1	84.7	78.6	71.9	67.9	58.7	52.5	39.6	18.0	0.0

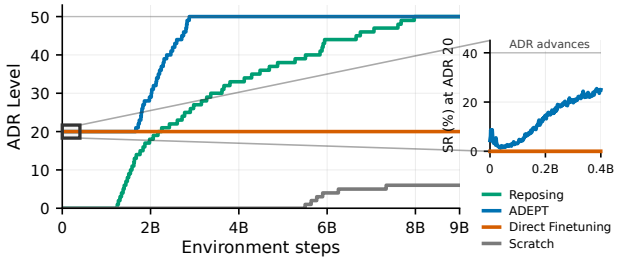

Fig. 3: Training curves on FMB peg insertion comparing **(1)** training from scratch, **(2)** direct PPO fine-tuning of the pretrained reposing teacher, and **(3)** ADEPT post-training. Blue inset: success-rate collapse for (2) during transfer.

levels with no post-training (Tab. I). ADR advances the goal along an L-shaped path: a horizontal lift-and-transport segment (ADR 0–25), then a vertical insertion segment descending into the hole (ADR 25–50). Zero-shot success stays above 50% through ADR 35 and reaches 0% at the insertion goal (ADR 50), where it begins contacting the receptacle, which is out-of-distribution. We therefore begin post-training from ADR 20, where the policy reliably zero-shots the reposing portion and only the contact-rich insertion remains.

How does ADEPT compare to training from scratch? Fig. 3 plots ADR level over environment steps for (1) from scratch, (2) direct PPO fine-tuning of the pretrained teacher, and (3) ADEPT. ADEPT trains a downstream teacher in 3B steps on top of the 8B-step reposing pretraining; from scratch is highly seed-sensitive, with most seeds plateauing below ADR 6. Crucially, the 8B pretraining cost is amortized across all downstream tasks, so the marginal cost per task reduces to the 3B post-training.

Why does naive RL fine-tuning fail? The inset of Fig. 3 zooms in on ADR 20, where (2) and (3) start from the same teacher: naive PPO drives success to zero within a few updates while ADEPT keeps improving. The pretrained critic V_{pre} is calibrated to the reposing reward, so under the insertion reward its advantages are unreliable; the resulting gradients push the actor off its pretrained distribution faster than the critic can recalibrate, destroying the behavior within a few iterations. ADEPT's three stages address exactly these failure modes.

Emergent natural grasps. Since the insertion reward does not constrain how the peg is grasped, from-scratch PPO converges to unnatural, seed-dependent grasps, whereas reposing pretraining initializes the policy where natural grasps emerge; post-training then refines rather than replaces them, filtering out task-misaligned grasps.

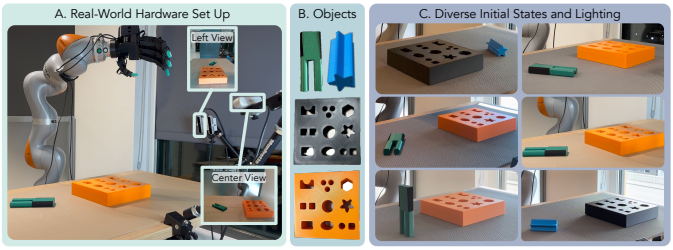

Fig. 4: **(a)** KUKA–Allegro real-world setup with the FMB pegs (star and asymmetric square/round) and insertion boards. **(b)** Stereo views from the left and center RGB cameras. **(c)** Sample diverse initial states used in real-world testing.

TABLE II: Per-stage success rates of two-stage vision students on real-world FMB peg insertion. Each stage is cumulative.

Peg	Reach	Grasp	Lift	Reorient	Align	Insert (SR)
Star	10/10	9/10	8/10	8/10	7/10	**5/10**
Square/Round	10/10	8/10	6/10	4/10	3/10	**3/10**

B. Real-World Experiments

Setup. Our platform is a 7-DoF KUKA iiwa7 arm with a 16-DoF Allegro hand on the flange (Fig. 4) and two calibrated Intel RealSense RGB cameras; the student zero-shot transfers by sharing the same geometric-fabric interface as in simulation. We evaluate two FMB [12] pegs spanning the benchmark's difficulty: a symmetric star peg (multiple valid insertion orientations) and the asymmetric square-and-round peg, the hardest FMB geometry (a single valid orientation). Pegs start flat on the table over a $30 \times 25\,\text{cm}$ spawn range and $[-\pi, \pi]$ rad in yaw; a trial succeeds when the peg is fully inserted.

Results. The ADEPT-trained state teacher achieves 90% and 89% success on the star and square/round pegs in simulation. Our two-stage distillation transfers these to a vision-only student with $5/10$ and $3/10$ zero-shot real-world success (Tab. II). The star peg's ridges let fingers wrap between them for stable grasps; the asymmetric peg's round side yields unstable point contacts, producing larger drops at lifting and reorienting. The single-stage baseline fails to transfer in both cases, confirming Stage-1 perception pretraining is critical. Our policy executes the entire sequence as a single continuous behavior in under 10 s—versus the 20–40 s, multi-stage, fixture-dependent parallel-jaw demonstrations in FMB—and lets recovery behaviors like drop-and-regrasp emerge naturally.

V. Conclusion and Limitations

ADEPT offers a more efficient path toward real-world dexterity by offloading arm–hand coordination to a generic pretraining objective, then specializing it for contact-rich tasks via structured post-training that preserves the pretrained prior. The remaining failures show perception is the primary bottleneck for vision-only distillation, the main avenue for future work.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5977–5984. IEEE, 2023.

[3] Tyler Ga Wei Lum, Martin Matak, Viktor Makoviychuk, Ankur Handa, Arthur Allshire, Tucker Hermans, Nathan D. Ratliff, and Karl Van Wyk. DextrAH-G: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. In *Conference on Robot Learning (CoRL)*, 2024.

[4] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84), 2023.

[5] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU-based physics simulation for robot learning. In *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[6] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. URLB: Unsupervised reinforcement learning benchmark. In *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[7] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline rl pretraining for efficient online fine-tuning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[9] Aleksei Petrenko, Arthur Allshire, Gavriel State, Ankur Handa, and Viktor Makoviychuk. Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training, 2023. URL https://arxiv.org/abs/2305.12127.

[10] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[12] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: a functional manipulation benchmark for generalizable robotic learning. *International Journal of Robotics Research (IJRR)*, 2024.

[13] Karl Van Wyk, Ankur Handa, Viktor Makoviychuk, Yijie Guo, Arthur Allshire, and Nathan D. Ratliff. Geometric fabrics: a safe guiding medium for policy learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[14] Ritvik Singh, Arthur Allshire, Ankur Handa, Nathan Ratliff, and Karl Van Wyk. DextrAH-RGB: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024.

[15] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

[16] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *International Journal of Robotics Research (IJRR)*, 39(1):3–20, 2020.

[17] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning (CoRL)*, 2022.

[18] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning (CoRL)*, 2022.

[19] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning (CoRL)*, 2023.

[20] Patrick Yin, Tyler Westenbroek, Zhengyu Zhang, Joshua Tran, Ignacio Dagnino, Eeshani Shilamkar, Numfor Mbiziwo-Tiapo, Simran Bagaria, Xinlei Liu, Galen Mullins, Andrey Kolobov, and Abhishek Gupta. Emergent dexterity via diverse resets and large-scale reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.15789.

[21] Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. *International Conference on Learning Representations (ICLR)*, 2025.

[22] Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning.

In *European Symposium on Artificial Neural Networks (ESANN)*, 2022.

[23] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.

[24] Ching-An Cheng, Mustafa Mukadam, Jan Issac, Stan Birchfield, Dieter Fox, Byron Boots, and Nathan Ratliff. RMPflow: A computational graph for automatic motion policy generation. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

[25] Nathan D. Ratliff, Karl Van Wyk, Mandy Xie, Anqi Li, and Muhammad Asif Rana. Optimization fabrics. *arXiv preprint arXiv:2008.02399*, 2020.

[26] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks, 2017. URL https://arxiv.org/abs/1711.09846.

[27] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

APPENDIX

Embodiment and control. The arm–hand system is a 7-DoF KUKA iiwa7 + 16-DoF Allegro hand ($n_q = 23$). The policy commands the joint configuration-space geometric fabric at 60 Hz, interpreting its $[-1, 1]^{23}$ action as per-joint relative deltas applied to the previous fabric target and clamped to URDF joint limits. The same fabric instance, with identical parameters, runs in simulation during training and on the real robot during deployment, where it is hosted in its own process and receives Cspace targets over shared memory; this removes the control-stack reality gap so domain randomization can focus on physics and perception.

Pre-training. The reposing teacher is trained with PPO [11] and an asymmetric actor–critic using Population-Based Training [26, 9] over 16 primitive shapes at randomized scale, with ADR-annealed gravity and goal curriculum, for ∼8B environment steps.

Post-training. BC actor distillation runs for 40k iterations; critic warm-up runs ∼20 PPO iterations (∼1M env steps/GPU at 4096 envs) with the actor frozen; conservative PPO then jointly updates the actor and critic (actor LR 1e-3 → 1e-5 with linear decay, clip ϵ : 0.2 → 0.05, critic LR 5e-5) for ∼3B steps.

Distillation. The stereo RGB student uses a ResNet backbone with a DextrAH-RGB-style [14] unfreeze schedule, trained with DAgger [27] under $\mathcal{L} = \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{aux}}$, where $\mathcal{L}_{\text{aux}}$ supervises an 8-keypoint object pose head against the simulator pose. Stage 1 pretrains the encoder via a perception-heavy reposing surrogate; Stage 2 distills the downstream insertion teacher from the Stage-1 checkpoint. Both stages use aggressive randomization of physics, object wrenches, lighting, background, camera intrinsics/pose, and proprioception/contact sensor noise.

Pretraining: object reposing. Fig. 5 shows the generic reposing task used for pretraining. Each episode spawns one of 16 primitive shapes at a randomized scale and pose; the policy must reach, grasp, lift, reorient in-hand, transport, and repose the object to a sampled SE(3) goal pose (red/green workspace bounds), exercising the motor primitives that transfer to downstream tasks.

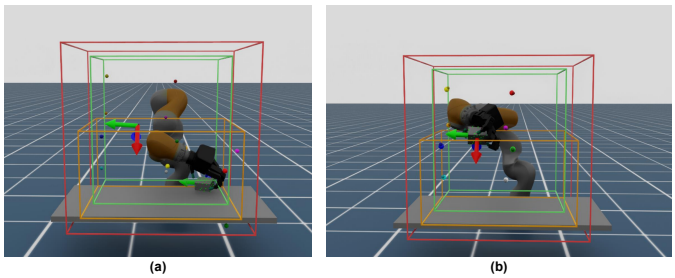

Fig. 5: **Reposing pretraining task. (a, b)** A primitive object must be reposed to a sampled goal pose within the workspace bounds, driving the emergence of reusable grasping, lifting, and in-hand reorientation skills.

Downstream: FMB peg insertion. Fig. 6 shows the downstream FMB insertion task and the ADR-annealed goal path (grey spheres). ADR advances the goal along an L-shaped trajectory: a horizontal lift-and-transport segment from above the peg spawn to above the board hole (ADR 0–25), then a vertical insertion segment descending into the hole (ADR 25–50). Post-training begins from ADR 20.

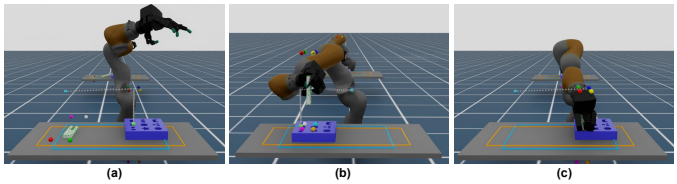

Fig. 6: **FMB peg insertion task. (a)** Peg spawned on the table with the insertion board; **(b)** mid-task grasp and transport along the ADR goal path; **(c)** alignment above the hole. Grey spheres mark the ADR-annealed goal trajectory.

Fig. 7 reports the same comparison as the main-text training curves but against wall-clock time, and Fig. 8 shows the seed sensitivity of training from scratch. Most from-scratch seeds plateau at or below ADR 6 even after 9B environment steps, whereas ADEPT reliably reaches the full ADR 50 by reusing the amortized reposing prior.

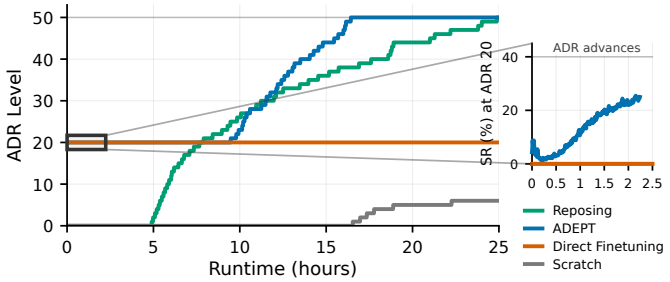

Fig. 7: **Training curves vs. wall-clock time.** ADR level over runtime for reposing pretraining, ADEPT post-training, direct fine-tuning, and from-scratch RL. Inset: success rate at ADR 20 as ADR advances.

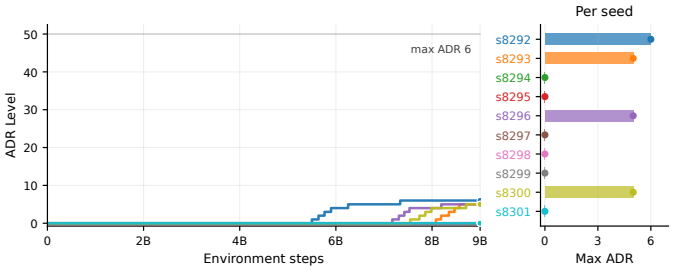

Fig. 8: **From-scratch seed sensitivity.** ADR level over environment steps across 10 seeds of training FMB insertion from scratch (left), with per-seed maximum ADR (right). Most seeds stall at or below ADR 6.

Fig. 9 contrasts grasps from the zero-shot reposing teacher, a from-scratch downstream teacher, and ADEPT. Reposing

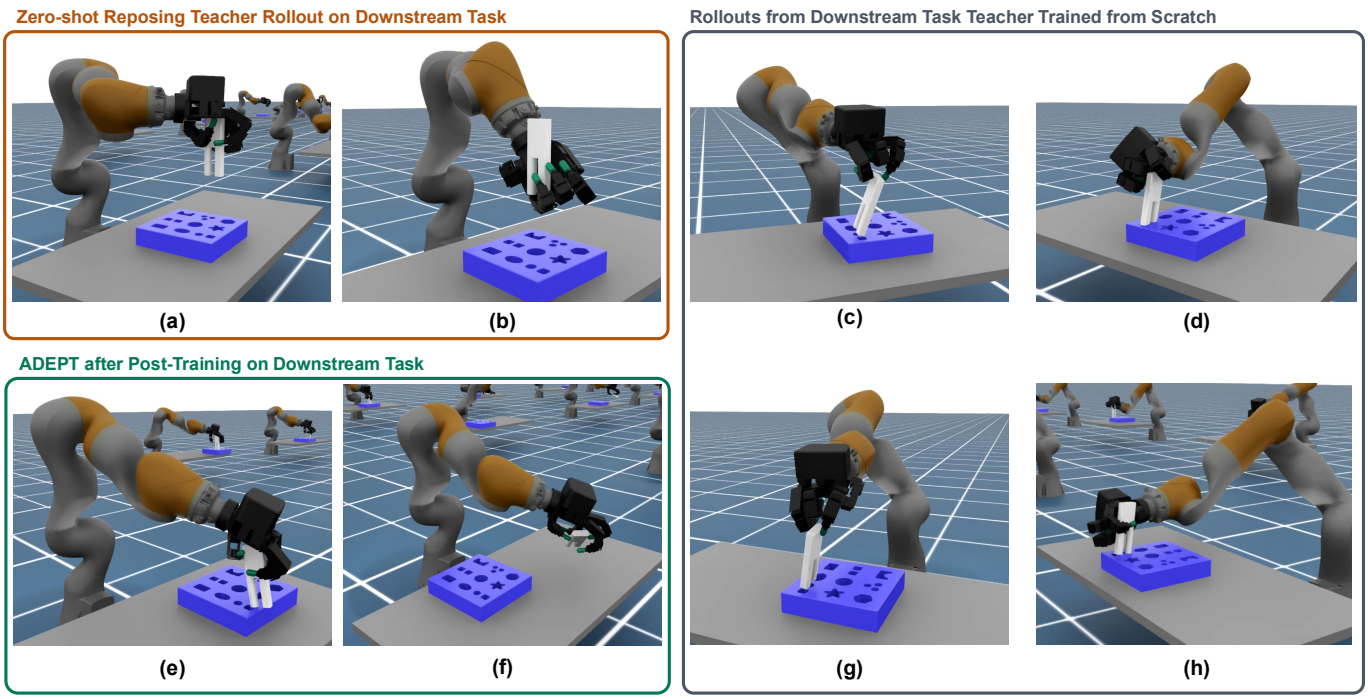

Fig. 9: **Qualitative grasp comparison. (a, b)** Zero-shot reposing-teacher rollouts on the downstream task; **(c, d)** downstream teacher trained from scratch; **(e, f)** ADEPT after post-training; **(g, h)** additional from-scratch rollouts. Reposing pretraining produces natural fingers-wrapped grasps, which post-training refines rather than replaces.

pretraining yields natural grasps in which the fingers wrap around the peg; from-scratch RL converges to unnatural, seed-dependent grasps; ADEPT refines the pretrained grasps for the downstream task while filtering out task-misaligned ones (e.g., grasping the peg from the bottom).