Emergent Dexterity via Diverse Resets and Large-Scale Reinforcement Learning

Patrick Yin[1,*], Tyler Westenbroek[1,*], Octi Zhang[2], Joshua Tran[1], Ignacio Dagnino[1], Eeshani Shilamkar[1], Numfor Mbiziwo-Tiapo[1], Simran Bagaria[3], Xinlei Liu[1], Galen Mullins[3], Andrey Kolobov[3], Abhishek Gupta[1]

***Abstract*—Reinforcement learning (RL) in massively parallel simulation has driven major progress in sim-to-real robotics, but current approaches remain brittle and task-specific, relying on per-task reward shaping, curricula, and demonstrations. We introduce `OmniReset`, a simple and scalable framework that enables on-policy RL to robustly solve a broad class of dexterous manipulation tasks using fixed hyperparameters, no curricula, minimal reward engineering, and no human demonstrations. Our key insight is that long-horizon exploration can be dramatically simplified by using simulator resets to systematically expose the RL algorithm to the diverse robot–object interactions underlying dexterous manipulation. `OmniReset` programmatically generates such resets with minimal user input, converting compute directly into broader behavioral coverage and continued performance gains. We show that `OmniReset` scales to long-horizon dexterous manipulation tasks beyond the capabilities of existing approaches, and that the resulting policies can be distilled into RGB visuomotor policies that transfer zero-shot to real hardware with robust retrying behavior. Project website: https://omnireset.github.io.**

I. Introduction

RL in massively parallel simulators [1], [2] has fueled recent sim-to-real successes [3], [4], but standard exploration [5], [6] saturates as compute is scaled, repeatedly sampling a narrow state distribution and getting stuck in local minima [7]. Practitioners typically inject task-specific structure through hand-designed rewards [8], [9], curricula [10], or demonstrations [11]–[13], or scaffold the problem with motion planning [14], [15]. These approaches are bounded by the human effort they require.

We argue that with the right system design this scaffolding is unnecessary. Although the space of dexterous behaviors is vast, successful policies reuse a small set of recurring interaction modes: approaching, contacting, grasping, and goal-directed contact-rich motion. By *systematically resetting* the simulator to states that densely cover these modes, sparse rewards propagate smoothly through the state space and large-scale RL can stitch together multi-stage strategies on its own.

We introduce `OmniReset`, which automatically generates diverse, manipulation-centric reset distributions from minimal user input (a target object, goal set, and workspace). Combined with large-batch on-policy PPO [5], `OmniReset` solves long-horizon contact-rich tasks (table-leg assembly, drawer insertion, peg insertion, cube stacking, non-prehensile reorientation, cupcake placement) without task-specific reward shaping, curricula, or demonstrations. We further distill the resulting state-based policies into RGB visuomotor policies that transfer zero-shot to a UR7e arm with robust retrying behavior.

Fig. 1. `OmniReset` uses diverse simulator resets and large-scale RL to solve contact-rich, long-horizon tasks. Policies are distilled to RGB inputs and transferred to the real world zero-shot.

II. Generating Diverse Resets for RL

Setup and User Inputs. `OmniReset` formulates each task as an MDP with a single user-specified target object $s^{\text{tar}} \subset s$, a goal set $\mathcal{G} \subset \mathcal{S}$, and a robot workspace $\mathcal{W} \subset \mathcal{S}$. From these, it constructs a diverse initial-state distribution ρ that lets a single task-agnostic reward solve all considered tasks.

Reset Distributions. We pre-sample 1,000 feasible grasp points on s^{tar} using the IsaacLab grasp sampler [1], and precompute near-goal offsets by spawning s^{tar} at $\mathcal{G}$ and applying small perturbations [16]. We then form four reset regions (Fig. 2):

- **Reaching (S^R):** s^{tar} on the table, gripper at random poses in $\mathcal{W}$.

[1]University of Washington. [2]NVIDIA. [3]Microsoft Research. *Equal contribution. Correspondence to patyin@cs.washington.edu.

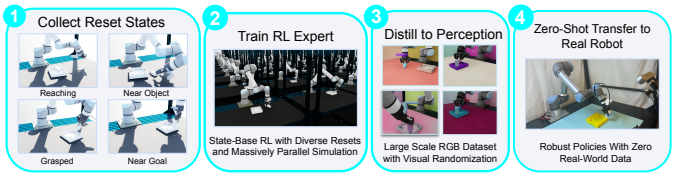

Fig. 2. **Sim-to-real pipeline.** (1) Generate partial assemblies and grasps; (2) collect Reaching, Near-Object, Grasp, and Near-Goal resets; (3) train a state-based RL policy from this reset distribution; (4) student–teacher distillation to an RGB policy; (5) deploy zero-shot.

- **Near-Object** (S^{NO})**:** s^{tar} on the table, gripper at a sampled grasp point with small offset; gripper open or closed.
- **Stable Grasp** (S^{G})**:** s^{tar} in the air, gripper at a feasible grasp.
- **Near-Goal** (S^{NG})**:** s^{tar} at a near-goal offset, gripper in contact.

We reject infeasible samples offline (collision checking + simulator stabilization) to obtain validated datasets $D^{R}, D^{NO}, D^{G}, D^{NG}$. During training we sample uniformly from their union, cached on-GPU. Crucially, the reset states are not ordered or graph-connected; the trajectories that link them are entirely emergent from RL.

Algorithmic Choices. We use a single shared reward across tasks:

$$r = r_{\text{success}} + r_{\text{dist}} + r_{\text{reach}} + r_{\text{smooth}} + r_{\text{term}},$$

with weights fixed across all experiments. Three additional choices matter for scaling: (i) very large parallel-environment counts to prevent forgetting and maintain value propagation; (ii) an asymmetric actor–critic [17] that exposes privileged state only to the critic; and (iii) gSDE exploration noise [18] for temporally correlated, state-dependent exploration in heterogeneous multi-stage tasks.

III. Simulation Experiments

We evaluate on six contact-rich tasks (`Leg Twisting`, `Drawer Insertion`, `Peg Insertion`, `Cube Stacking`, `Wall Slide`, `Cupcake Placement`), each with `Easy` (narrow initial conditions, fixed goal) and `Hard` (wide initial conditions, randomized goals) variants.

Baselines. We compare against three demonstration-based baselines, each given 100 successful demonstrations: BC-PPO [19], [20], DeepMimic-style reward augmentation [12], and a success-weighted Demo Curriculum in the spirit of DemoStart [11].

Results. As shown in Fig. 3, `OmniReset` consistently obtains high success rates across tasks and substantially outperforms baselines, particularly on the `Hard` variants where prior methods fail to scale to wider initial-condition distributions. The baselines often make partial progress on the `Easy` variants but collapse as the initial-condition distribution widens.

Emergent reverse curriculum. Although no curriculum is specified, `OmniReset` naturally solves tasks "backwards":

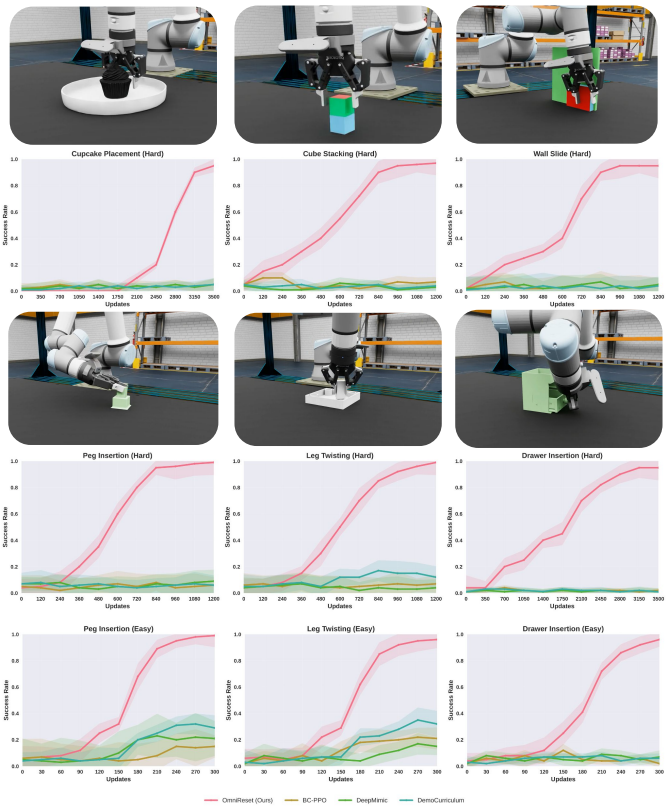

Fig. 3. **Success rates during training.** `OmniReset` scales to tasks where demonstration-based baselines struggle to make meaningful progress, especially on `Hard` variants with wide initial-condition distributions.

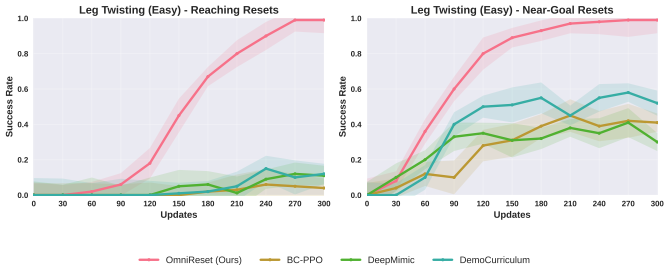

Fig. 4. **Success by task stage on `Leg Twisting`.** Baselines achieve moderate success near the goal but cannot stitch together the long-horizon reaching task. `OmniReset` solves both, illustrating the emergent backward propagation of value induced by diverse resets.

it first succeeds from Near-Goal resets, then Grasp, Near-Object, and finally Reaching, purely as a side-effect of broad reset coverage and large batch sizes. Fig. 4 visualizes this on `Leg Twisting`: the demonstration-based baselines plateau at moderate success when starting near the goal and fail almost entirely from reaching states, while `OmniReset` propagates value backwards through the state space and ultimately succeeds from the entire workspace.

Robustness. We perturb sampled initial conditions with random forces of varying magnitude and report the resulting drop in success (Fig. 5). Baselines degrade sharply under even small perturbations, while `OmniReset` is nearly invariant

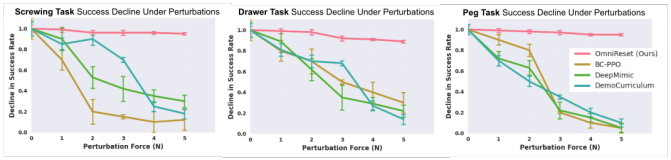

Fig. 5. **Robustness to initial-state perturbations.** Relative success rate vs. perturbation magnitude. `OmniReset` is largely unaffected; baselines collapse.

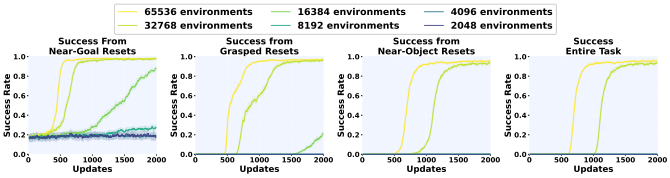

Fig. 6. **Parallel-environment ablation on `Leg Twisting Hard`.** Success rates from each reset region across training. Small environment counts solve only the Near-Goal portion; large counts are required for the full long-horizon task.

across the perturbation range, a direct consequence of training over a much broader initial-state distribution. Scatter plots over the workspace (omitted for space) show `OmniReset` succeeding across the full x–y region, whereas the strongest baseline (Demo Curriculum) succeeds only in narrow patches near the demonstrated initializations.

Ablations. We ablate (i) the number of parallel environments and (ii) the breadth of grasp sampling on `Leg Twisting Hard` (Fig. 6). With few environments, PPO can solve the task only from Near-Goal resets and never propagates value back to Reaching states; large parallelism is essential for the full multi-stage task. Likewise, narrowing the grasp-sampling range degrades both sample efficiency and final success. Together, these confirm that dense coverage of robot–object interaction modes (not algorithmic novelty) is the primary driver of `OmniReset`'s performance.

IV. Distillation and Real-World Transfer

We distill state-based `OmniReset` policies into RGB visuomotor policies for zero-shot deployment on a UR7e with a Robotiq 2F-85 gripper. Using IsaacLab's photorealistic rendering [1], we collect 80k expert rollouts and train a ResNet-18 + Gaussian MLP student [21] on three RealSense views (D455 front, D435 side, D415 wrist) at 10 Hz. We apply visual domain randomization following DextrAH-G [22] (lighting, backgrounds, textures, camera pose/FOV jitter) and dynamics randomization with system-identified actuator parameters following PACE [23], and use a shared task-space operational-space controller [24] in sim and real.

Table I reports zero-shot real-world success rates of 85.4% (Peg), 56.4% (Leg), and 15.4% (Drawer), more than an order of magnitude above the BC baseline. Qualitatively, the policies exhibit robust retrying behavior: they recover from initial grasp failures and re-engage the object until task completion (see videos at https://omnireset.github.io/#evaluations).

TABLE I
Sim-to-real performance. Distilled `OmniReset` policies transfer zero-shot, far outperforming a Diffusion Policy [25] BC baseline trained on 100 real demos.

Metric	Peg	Leg	Drawer
State RL (Sim)	94.97	93.94	89.60
Distilled Image (Sim)	55.45	43.69	84.00
Distilled Image (Real)	**85.37**	**56.36**	**15.38**
First-Try Success (Real)	21.95	43.64	5.77
Throughput (succ/min)	1.49	0.63	0.31
Real-Only BC, 100 demos	2.44	1.82	1.92

V. Conclusion

`OmniReset` shows that a diverse, minimally structured reset distribution paired with large-batch on-policy RL is sufficient to produce surprisingly dexterous behavior on long-horizon contact-rich tasks, without curricula, demonstrations, or task-specific reward shaping, and that the resulting policies transfer zero-shot to hardware. Limitations include reliance on the quality of pre-computed grasps (challenging for highly non-convex objects and bimanual or multi-finger settings), modest dynamics randomization compared to large sim-to-real pipelines, and a remaining sim-to-real gap for the distilled RGB policies that we expect can be narrowed by combining DAgger with on-policy RL from images and by scaling the distillation dataset further.

References

[1] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[2] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[3] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[4] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[7] J. Singla, A. Agarwal, and D. Pathak, "Sapg: split and aggregate policy gradients," *arXiv preprint arXiv:2407.20230*, 2024.

[8] T. Westenbroek, F. Castaneda, A. Agrawal, S. Sastry, and K. Sreenath, "Lyapunov design for robust and efficient robotic reinforcement learning," *arXiv preprint arXiv:2208.06721*, 2022.

[9] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5977–5984.

[10] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[11] M. Bauza, J. E. Chen, V. Dalibard, N. Gileadi, R. Hafner, M. F. Martins, J. Moore, R. Pevceviciute, A. Laurens, D. Rao *et al.*, "Demostart: Demonstration-led auto-curriculum applied to sim-to-real with multi-fingered robots," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 6756–6763.

[12] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[13] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[14] M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, and D. Fox, "Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 7505–7512.

[15] B. Tang, I. Akinola, J. Xu, B. Wen, A. Handa, K. Van Wyk, D. Fox, G. S. Sukhatme, F. Ramos, and Y. Narang, "Automate: Specialist and generalist assembly policies over diverse geometries," *arXiv preprint arXiv:2407.08028*, vol. 1, no. 2, 2024.

[16] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. Narang, "Industreal: Transferring contact-rich assembly tasks from simulation to reality," *arXiv preprint arXiv:2305.17110*, 2023.

[17] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.

[18] A. Raffin, J. Kober, and F. Stulp, "Smooth exploration for robotic reinforcement learning," in *Conference on robot learning*. PMLR, 2022, pp. 1634–1644.

[19] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[20] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[21] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning, 8-11 November 2021, London, UK*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 2021, pp. 297–307. [Online]. Available: https://proceedings.mlr.press/v164/chen22a.html

[22] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. V. Wyk, "Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands," 2025. [Online]. Available: https://arxiv.org/abs/2412.01791

[23] F. Bjelonic, F. Tischhauser, and M. Hutter, "Towards bridging the gap: Systematic sim-to-real transfer for diverse legged robots," *arXiv preprint arXiv:2509.06342*, 2025.

[24] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[25] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.