DexPour: Effective and Efficient High-DoF Robotic Hand Liquid Pouring via Hierarchical Reward with Approximated Proxy Abstraction

Xinmin Fang[1], Lingfeng Tao[2], Zhengxiong Li[1]

***Abstract*— Pouring fluids is a routine task for humans but challenging for high-DoF robots, particularly given fluid simulation's computational demands while training policies. In this paper, we propose DexPour, a novel reinforcement learning method with hierarchical rewards and Approximated Proxy Abstraction (APA) method. APA efficiently approximates liquid behavior using a small set of spheres, reducing computational overhead. Meanwhile, our hierarchical reward framework breaks down the intricate pouring process into four distinct stages—approach, grasp, transport, and pour—providing fine-grained feedback and fostering stable policy learning. Extensive experiments demonstrate that DexPour achieves a 92% fluid transfer efficiency with a 70% cup fill and a 99% efficiency at 30% fill, highlighting its robust performance across varying liquid volumes. Ablation studies highlight the contribution of each component, confirming the necessity of detailed stage-wise guidance for complex dexterous manipulation. In addition, we compare DexPour with a full fluid simulation baseline, showing comparable pouring efficiency while reducing training time by 81.6%, demonstrating DexPour's efficiency and practical viability for fluid manipulation tasks.**

I. Introduction

Pouring liquids is an ubiquitous and essential operation in a variety of real-world scenarios, ranging from domestic tasks such as cooking and beverage preparation to industrial processes. Despite this, reliable robotic pouring remains a formidable challenge. The inherent fluid dynamics of pouring introduce substantial complexity, as even minute variations in grasp posture or pouring angles can lead to spillage or incomplete transfer of fluids. These challenges become more pronounced when high-degrees-of-freedom (DoF) dexterous hands are employed, where the added flexibility must be carefully orchestrated to maintain fluid flow control.

Learning-based methods have proven highly effective in a wide range of robotic manipulation tasks, demonstrating remarkable adaptability to unstructured environments and complex sensorimotor challenges [1]. Recent advances in deep reinforcement learning (DRL), in particular, have led to robust control policies for intricate operations such as in-hand object reorientation, bimanual coordination, and multi-step manipulation planning [2], [3]. However, applying these learning-based approaches to fluid manipulation tasks presents unique computational challenges due to the necessity of simulating accurate fluid dynamics. High-fidelity fluid simulations required for training such policies often demand massive computational resources and extensive training times, hindering experimental throughput and practical deployment.

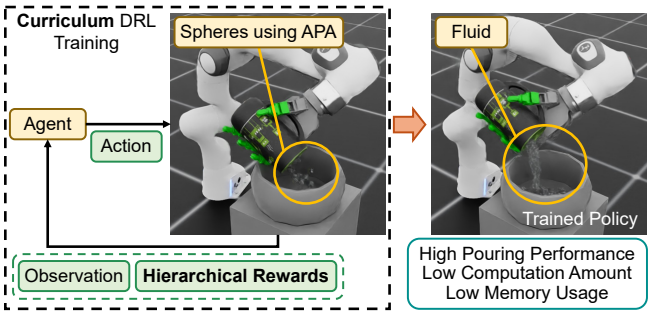

Fig. 1. DexPour pioneers high-DoF dexterous fluid manipulation through hierarchically decomposed rewards and APA, eliminating the need for explicit fluid dynamics simulation while maintaining task efficacy.

Curriculum learning has shown significant promise in enabling agents to tackle progressively more challenging tasks by structuring the training process in incremental stages [4], [2]. In robotic manipulation, this approach has been successfully applied to tasks like in-hand object reorientation and precise assembly operations, where control policies benefit from learning fundamental skills first before advancing to more complex subgoals [1]. Despite these advances, few studies have explored the synergy between curriculum-based training and fluid manipulation. Pouring tasks, in particular, present unique challenges, such as managing stable grasp and fine-grained motion control, factors that have yet to be investigated under a curriculum learning paradigm.

Currently, some robots have demonstrated the ability to pour liquids [5]. However, these robots are typically hard-coded to complete the task and lack learning capabilities. Some methods, including research using large language models (LLMs), have shown a certain level of competence in pouring tasks [6]. However, these approaches are usually limited to manipulating the cup, lack a deep understanding of the liquid dynamics, and perform poorly when handling different water volumes. Therefore, these types of methods are beyond the scope of discussion in this paper.

In this work, we propose DexPour, a novel approach to high-DoF dexterous fluid manipulation. As illustrated in Fig. 1, we address key challenges through a well-designed hierarchical reward mechanism and an Approximated Proxy Abstraction (APA), integrated with curriculum learning. The hierarchical rewards isolate each phase of pouring, from approach and grasp to transport and pour, while APA em-

*This work is partially supported by US NSF Awards 2426269 and 2426470, as well as the 2022 Meta Research Award.

[1]X. Fang, and Z. Li are with the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80217 USA (e-mail: xinmin.fang@ucdenver.edu, zhengxiong.li@ucdenver.edu)

[2]L. Tao are with the Oklahoma State University, 563 Engineering North, Stillwater, OK, 74078, USA (e-mail: lingfeng.tao@oksate.edu)

ploys multiple small spheres to replicate fluid-like behaviors without the prohibitive cost of full-scale fluid simulation. Finally, curriculum learning further refines DexPour's stability, ensuring reliable performance in complex pouring tasks.

Our work includes the following key contributions:

- DexPour is pioneering to demonstrate a robotic system with a high-DoF dexterous hand (23 DoF in total) that effectively performs pouring task (including approaching, grasping, transporting, and pouring stages), pushing the frontier of dexterous robotic manipulation tasks. *We will open-source it once it is accepted.*
- We propose a reward strategy that segregates the pouring process into distinct sub-tasks, ensuring that the policy learns critical skills at each stage (approach, grasp, transport, and pour). This methodology not only facilitates sample-efficient training but also provides interpretable insights into the robot's performance at each phase of the manipulation sequence.
- DexPour reduces 81.6% of computational time and maintains high pouring efficiency by substituting computationally heavy fluid dynamics APA method. Our RL-based approach radically reduces the required computational resources while maintaining robust performance in fluid transfer tasks.
- Our experiments demonstrate that DexPour achieves a fluid transfer efficiency of 92% when the cup is filled to 70% capacity, and this performance further improves to 99% at 30% fill. Through a series of ablation studies, we verify the contribution of each algorithmic component, illustrating how each reward term and training strategy underpins successful dexterous fluid manipulation.

II. Related Work

A. Dexterous Robotic Manipulation

High-DoF dexterous hands have achieved remarkable progress in manipulating rigid [7] or deformable solids [8], yet liquid handling remains underexplored due to the stochasticity and partial observability of fluid dynamics. Existing dexterous manipulation studies (e.g., multifingered grasping [9]) primarily target static objects, ignoring the dynamic coupling between hand motions and fluid behavior. Tesla has publicly demonstrated an embodied robot fetching water by turning on a faucet and using a cup [10], which differs from the objective of this work, grasping a liquid-filled container and performing controlled pouring. This gap highlights a critical need for solutions that jointly address high-dimensional control and fluid stochasticity. Our work tackles RL with hierarchical rewards, enabling a 23-DoF hand to learn pouring without high-fidelity fluid dynamics priors or dense sensing.

B. Reward Design in DRL

Designing rewards is crucial in DRL, especially for multi-stage tasks. Existing methods span potential-based shaping [11], intrinsic motivation [12], and even GPT-assisted reward generation [13]. Among these, hierarchical approaches subdivide tasks into subgoals to mitigate sparse and delayed rewards. Prior works leverage curriculum learning [9] or physics-inspired metrics [14] to guide agents through sequential subtasks. However, these methods still perform poorly when faced with challenging tasks, such as pouring liquid using a dexterous hand, leading to training failures and inability to complete the task. Our work draws inspiration from hierarchical reinforcement learning [15] but tailors the reward structure to fluid-specific physics and task-phase decomposition, enabling dexterous robotic fluid manipulation.

C. Robotic Manipulation of Fluids

Prior work in robotic fluid manipulation primarily relies on physics-based models or specialized sensing (e.g., audio-vibration fusion [16], haptic-auditory feedback [17]) to estimate liquid states, yet such methods demand precise instrumentation and lack generalizability. Learning-based approaches, while promising, often depend on high-fidelity fluid simulators (e.g., SPH [18]) for training, incurring prohibitive computational costs. Recent vision-centric methods [19] reduce hardware dependencies but struggle with dynamic liquid behaviors (e.g., splashing, viscosity changes). Notably, most studies focus on low-DoF grippers [20] or simplified dynamics, overlooking the challenges of multi-finger coordination in dexterous pouring. These limitations, such as sensor dependency, simulation overhead, and limited dexterity, motivate our method that abstracts fluid dynamics via the liquid approximation, allowing efficient RL training for complex dexterous pouring.

III. Methodology

DexPour addresses the challenge of training a multi-fingered dexterous hand mounted on a robotic arm to grasp a liquid-filled cup, transport it, and pour the liquid into a target container. We assume that state observations, including container pose estimation, centroid displacement of the liquid, and contact interaction characteristics between the manipulator and the container, are accessible. We evaluate grasping stability and pouring dynamics.

A. Hierarchical Reward Mechanism with Physics Metrics

DexPour decomposes the fluid manipulation task into four sequential sub-objectives governed by physics-based reward signals: Approaching, Grasping, Transporting, and Pouring. The entire structure of the hierarchical physics-guided reward mechanism is described in Fig. 2.

Stage 1 Approaching: In this stage, we introduce a set of penalty terms that guide the dexterous hand to move closer to the cup in a stable, grasp-ready configuration. Specifically, we penalize the hand-cup distance ($p_{hand\_cup\_dist}$) and the height discrepancy between the hand (including its fingers) and the center of the cup (p_{height_dist}) to ensure appropriate approach behavior. Additionally, we impose penalties on the relative distance between the thumb and the other fingers, prompting the hand to open and prepare for a secure grip (p_{inter_finger}). The penalties are formulated as follows: $p_{appriaching} = p_{hand\_cup\_dist} + p_{height\_dist} + p_{inter\_finger}$. Notably, these penalties are only applied until the hand reaches a defined pre-grasp distance.

Stage 2 Grasping: We reward the agent based on the distance between the cup and the dexterous hand ($r_{hand\_cup\_dist}$) and the distances between the thumb and the cup's near side, as well as between the other fingers and the cup's far side ($r_{finger\_cup\_dist}$). This incentivizes the dexterous hand to adopt a grip that firmly encloses the cup. To further assess grasp quality, we count the number of finger–cup contact points, each contact yields a small reward $r_{contact}$ to encourage additional engagement. Once all four fingers are in contact with the cup, the agent receives a substantial grasping reward r_{grasp}. The grasping rewards are formulated as follows: $r_{grasping} = r_{hand\_cup\_dist} + r_{finger\_cup\_dist} + r_{contact} + r_{grasp}$.

Stage 3 Transporting: We give a linear reward r_{lift} based on the cup's height to encourage a controlled lift. It is a high multiple reward since lifting is a high-failure action that requires stronger incentivization. Once the cup reaches a certain height threshold, the lift reward ceases to accumulate, preventing the agent from continuously exploiting this incentive. Subsequently, to guide the agent in moving to the container, we provide a reward based on the distance between the opening of the cup and the target container (r_{cup_dist}):

$$r_{cup\_dist} = e^{-2\times dist_{cup\_target}}, \tag{1}$$

where $dist_{cup\_target}$ is the distance between the opening of the cup and the target container. Finally, we impose a tilt penalty p_{tilt} to maintain stability during transport, minimizing the risk of fluid spillage. The transporting rewards are formulated as follows: $r_{transporting} = r_{lift} + r_{cup\_dist} + p_{tilt}$.

Stage 4 Pouring: In this stage, we introduce a rotation-based reward r_{tilt} to encourage the agent to tilt the cup for pouring. The reward peaks when the angle between the cup rim and the horizontal plane is approximately 45°, preventing excessive tilt. To further guide the agent in aligning the cup with the target container, we define an alignment reward:

$$r_{align} = \frac{1+\cos(\theta)}{2}, \tag{2}$$

where θ is the angle between two normalized vectors: the cup's orientation vector and the displacement vector from the cup's center to the target container's center , thereby measuring how closely they point in the same direction. Although this alignment metric is not perfectly accurate from a strict geometric perspective, it provides a sufficiently robust signal to help the agent achieve the correct pouring motion. The pouring rewards are as follows: $r_{pouring} = r_{tilt} + r_{align}$.

The multi-stage reward mechanism employs binary triggers (λ, μ, ν, ρ) to sequentially activate stage-specific rewards upon satisfying phase-transition criteria (Fig. 3). Proximity detection (λ) initiates approach rewards when the hand enters the cup's near threshold. Subsequent stages activate upon: secure grip establishment (μ), successful cup elevation (ν), and target alignment verification (ρ). Crucially, each stage's activation inherently validates completion of prior phases, ensuring temporal coherence in the manipulation sequence. hey are defined as:

$$\lambda = \begin{cases} 0 & \text{if } dist_{hand\_cup} \geq d_{approach} \\ 1 & \text{if } dist_{hand\_cup} < d_{approach} \end{cases} \tag{3}$$

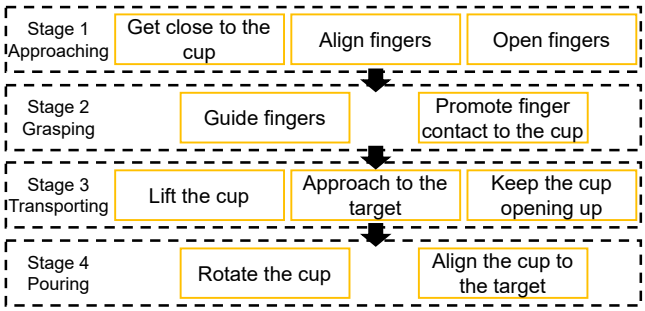

Fig. 2. The dexterous hand fluid manipulation task is decomposed into four stages: approaching, grasping, transporting, and pouring. The hierarchical reward mechanism is implemented at all stages.

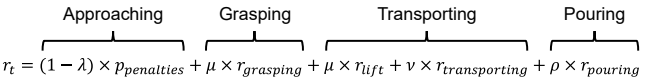

Fig. 3. The total hierarchical reward function including all four stages.

$$\mu = \begin{cases} 0 & \text{if } c_{contact} \neq c_{finger} \\ \lambda \times 1 & \text{if } c_{contact} = c_{finger} \end{cases} \tag{4}$$

$$\nu = \begin{cases} 0 & \text{if } height_{cup} < h_{lift} \\ \lambda \times \mu \times 1 & \text{if } height_{cup} \geq h_{lift} \end{cases} \tag{5}$$

$$\rho = \begin{cases} 0 & \text{if } dist_{cup\_target} \geq d_{pour} \\ \lambda \times \mu \times \nu \times 1 & \text{if } dist_{cup\_target} < d_{pour} \end{cases} \tag{6}$$

where $d_{approach}$ is set to 0.1m, c_{finger} is set to four, since the dexterous hand we use has four fingers, h_{lift} is determined as 0.15m, d_{pour} is determined as 0.17m. These parameter values are chosen with consideration of both the cup's dimensions and the target container's size.

B. Approximated Proxy Abstraction (APA)

As shown in Fig. 4, DexPour introduces a novel abstraction paradigm that replaces computationally intensive fluid dynamics with task-oriented proxy modeling. Central to our approach is the use of rigid-body spheres to emulate liquid behavior, grounded in two key hypotheses: (1) Macro-scale kinematic similarity between sphere motion and realistic fluid simulation during pouring, and (2) Insensitivity of policy learning to micro-scale fluid dynamics discrepancies. This abstraction eliminates reliance on high-fidelity fluid simulations while preserving essential manipulation cues. By focusing on task-relevant physical features rather than exhaustive fluid modeling, DexPour achieves an 81.6% reduction in computational amount compared to conventional fluid simulation, as validated in Section V-C.

C. Curriculum-based Training Process

In DexPour, we couple APA with a three-stage curriculum to balance feasibility and stability. In the first stage (16k steps), the cup contains only one proxy sphere and penalties on linear acceleration, velocity, and angular velocity remain low, letting the agent learn basic approach, grasp, transport, and pour actions. In the second stage (32k steps), we raise the

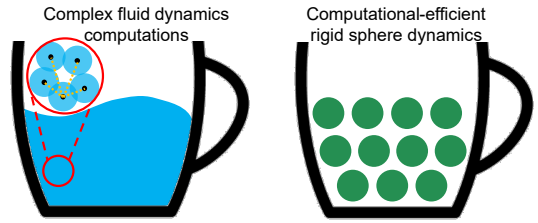

Fig. 4. Comparison of (a) Position-based-dynamics fluid simulation that need complex and time-consuming fluid simulation and (b) APA that utilize computational-efficient rigid simulation to simulate fluids.

penalty weights to encourage smoother, lower-acceleration motions and reduce spillage. Finally, in the third stage (64k steps), we significantly increase these penalties and add more proxy spheres (up to 32), compelling the agent to refine its pouring technique for precise fluid handling. By gradually scaling complexity and constraints, our curriculum fosters efficient, stable learning for the entire pouring process.

IV. Experiments

A. Task Design

DexPour is evaluated in a high-fidelity simulated environment to facilitate efficient training and rigorous testing. The robotic platform comprises a 7-DoF Franka Emika Panda arm integrated with a 16-DoF Allegro Hand, forming a 23-DoF dexterous system equipped with tactile sensors on all four fingertips. The experiments are conducted in NVIDIA Isaac Lab [21], a widely recognized physics simulation platform renowned for its computational accuracy and scalability in robotic learning scenarios. We train a unified policy to control all joints of the robotic system through reinforcement learning. The observation space encompasses: joint positions/velocities, fingertip positions, cup pose (position, quaternion, linear/angular velocities), target position, centroid positions of proxy spheres, and previous actions. The policy is trained using PPO with actor-critic networks comprising fully connected layers (512, 512, 256, 128) and ELU activations. Input and output dimensions correspond to the observation and action spaces.

To emphasize the effectiveness of our approach, we design a challenging evaluation scenario: the robot must retrieve a standard cylindrical mug (8.4 cm diameter $\times$ 18 cm height) positioned at ground level. This configuration demands the policy to master complex maneuvers, lowering the dexterous hand without ground collisions while achieving precise grasp alignment, particularly challenging given that the mug's height closely matches the hand's operational width ($\approx$12 cm). After grasping, the policy must lift the mug to at least 0.5 m and pour its contents into a wide-rimmed bowl (20 cm diameter, 10 cm height) at 0.4 m elevation, demanding careful dynamic control to avoid spilling. All hyperparameters are detailed in Table I. The training environment uses 2048 parallel instances, with the simulation running at a 0.008 s timestep for high dynamic fidelity on a workstation equipped with an AMD Ryzen 7 7700X, an NVIDIA RTX 4060Ti GPU, and 64GB of RAM.

TABLE I
Proximal Policy Optimization Hyperparameters

Hyperparameter	Value
Value Function Loss Coefficient	1.0
Clipped Value Loss	Enabled
Policy Clipping Parameter (ε)	0.2
Entropy Regularization Coefficient	0.005
Optimization Epochs per Update	5
Mini-batch Count	4
Learning Rate	5.0×10^{-4}
Learning Rate Schedule	Adaptive
Discount Factor (γ)	0.99
GAE Parameter (λ)	0.95
Target KL Divergence	0.016
Gradient Clipping Threshold	1.0
Environment Steps per Iteration	16

B. Ablation Experiments

To thoroughly evaluate the effectiveness of our proposed hierarchical reward strategy and curriculum design in DexPour, we conduct a series of ablation experiments that systematically remove different reward components. The goal is to isolate the impact of each reward term and the curriculum process on overall performance and learning efficiency. Specifically, we examine seven configurations: (1) *Full rewards + Curriculum*: Uses the complete hierarchical reward structure in conjunction with curriculum training, which is the proposed DexPour method. (2) *Full rewards*: Retains the full set of reward terms but omits the curriculum phase. (3) *Reward Stages 1, 2, 3 + Curriculum*: Incorporates reward signals for initial alignment, grasp, and transport, excluding pouring rewards. (4) *Reward Stages 1, 2, 4 + Curriculum*: Focuses on transport rewards, intentionally removing transport-specific terms. (5) *Reward Stages 1, 3, 4 + Curriculum*: Focus on grasp rewards, offering insights for the necessity of grasp rewards. (6) *Reward Stages 2, 3, 4 + Curriculum*: Focus on aligning rewards, offering insights into the necessity of explicit alignment. (7) *Goal reward only*: Provides a baseline scenario where only the final task completion reward is present.

We select four main metrics for the evaluation: (1) **Fluid Transfer Efficiency** (η_{ft}): During training, DexPour employs APA in which a small number of spheres are used in place of a full fluid simulation. For testing, we evaluate the policy using a realistic liquid simulation. The metric was calculated by the ratio of liquid successfully poured into the target cup to the total volume of liquid. (2) **Alignment Success Rate** (P_{align}): Measures how often the end-effector is correctly positioned and oriented before the grasp. (3) **Grasp Success Rate** (P_{grasp}): Evaluates the agent's ability to establish a stable grip on the cup through contact points. (4) $RMSE_{cup\_a}$: Quantifies how steadily the cup is transported by calculating its Root Mean Square Error (RMSE) acceleration. Let a_i represent the instantaneous acceleration of the cup at time step i over N time steps. The cup stability is then defined as: $RMSE_{cup\_a} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} a_i^2}$

C. Comparison with Full Fluid Simulation

To further validate the efficiency of DexPour, we compare our APA approach with a baseline that employs a full fluid

TABLE II

PERFORMANCE OF ALL ABLATION CONFIGURATIONS MEASURES OVER 200 TRIALS (THE BEST PERFORMANCE IS BOLDED)

	η_{ft}	P_{align}	P_{grasp}	$RMSE_{cup\_a}$
Full Reward + Curriculum (DexPour)	**92%**	**100%**	**100%**	**24.1**
Full Reward	0%	0%	0%	Failed
Stage 1, 2, 3 + Curriculum	91%	100%	100%	29.1
Stage 1, 2, 4 + Curriculum	0%	96%	0%	Failed
Stage 1, 3, 4 + Curriculum	0%	0%	0%	Failed
Stage 2, 3, 4 + Curriculum	0%	0%	0%	Failed
Goal Reward Only	0%	0%	0%	Failed

η_{ft}: Fluid Transfer Efficiency, P_{align}: Align Success Rate, P_{grasp}: Grasp Success Rate

dynamics simulation using the same amount of particles. The fluid simulation is implemented by NVIDIA Isaac Sim, which is a position-based-dynamics (PBD) particle simulation [22]. While full-fledged fluid simulation can capture the intricate behavior of liquids more accurately, it often requires substantial computational resources and extended training durations. To characterize and quantify these differences, we select the following key metrics: (1) **Fluid Transfer Efficiency**: The same as mentioned in Section IV-B. (2) **Steps**: The number of steps taken to complete a single episode reflects the agent's policy efficiency in completing the task. (3) **Memory Usage**: Memory usage reflects the resources consumed during policy training; higher memory usage translates into greater resource consumption and higher costs. (4) **Training Time per Iteration**: This metric provides a clear view of the computational cost associated with each training iteration. (5) **Sample Efficiency**: It was calculated by the percentage of fluid transfer efficiency divided by number of samples. High sample efficiency implies that the training method can reach competent performance with fewer interactions, reducing both computation time and cost.

V. RESULTS AND DISCUSSION

A. Ablation Study Results

Table II presents the performance of DexPour and its ablated configurations under a 70% cup-filling condition in realistic fluid simulation. As the results indicate, **Config. 1** (Full Reward + Curriculum, i.e., DexPour) achieves the highest performance across all metrics. Among the remaining configurations, only **Config. 3** succeeds in completing the pouring task, albeit with a 1% lower fluid transfer efficiency compared to **Config. 1**. We attribute this gap to the absence of an explicit pouring reward, the policy occasionally misaligns the cup during the pour, causing liquid spillage.

Other configurations fail for various reasons. **Config. 2** converges prematurely, avoiding cup movement to minimize penalties, demonstrating curriculum learning's necessity for progressive training. In **Config. 4**, after removing transport-related rewards, the policy can only receive new rewards if it happens to lift the cup by chance. Lacking guidance, the agent never discovers a stable lifting motion, underscoring the significance of transport-stage incentives. Similarly, **Config. 5** removes grasp-oriented rewards and forces the policy to learn high-dimensional finger coordination without any direct reinforcement, leading to near-impossible conditions for stable gripping. **Config. 6** cannot position the hand near the cup, entirely blocking subsequent reward signals. The sparse-reward baseline (**Config. 7**) failed, confirming hierarchical rewards are indispensable for multi-stage manipulation.

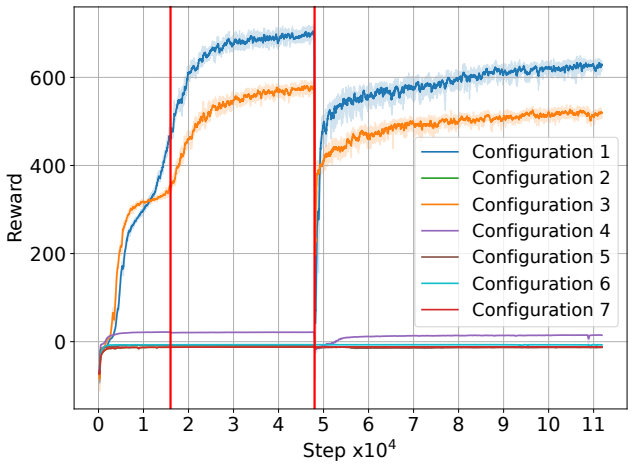

Fig. 5. The reward curve of all ablation configurations. The red lines are the dividing points of different curriculum stages.

Fig. 5 plots the reward curves during training for each configuration. Compared to **Config. 1**, **Config. 3** only learns to grasp at around 5,000 training steps and experiences a prolonged plateau before finally mastering pouring near 16,000 steps. In contrast, the pouring reward in **Config. 1** swiftly guides the policy to complete the pouring task at near 11,000 steps, as evidenced by a rapid increase in reward. Policies in the other ablated configurations remain in negative-reward territory throughout training, proving that complex dexterous fluid manipulation with a high-DoF robotic hand demands a carefully designed hierarchical reward mechanism.

B. Performance Under Varying Liquid Volumes

To evaluate the robustness of DexPour across different task conditions, we examine its fluid transfer efficiency under three cup-fill levels: 30%, 50%, and 70%. Each condition is tested over a minimum of 200 trials. The results are that the fluid transfer efficiency was 99% at 30% fill, 96% at 50% fill, and 92% at 70%. Despite the increased fluid volume, the policy maintains consistently high success rates, revealing that DexPour exhibits a high degree of stability even under more demanding pouring scenarios.

TABLE III

PERFORMANCE OF APA AND POLICY TRAINED WITH FLUID SIMULATION.

Metrics	APA	Fluid Simulation
Fluid Transfer Efficiency	92%	92%
Steps per Epoch	56.1	56.5
Memory Usage (MB)	5,429	4,763
Training time per Iteration (s)	**9.6**	52.2
Sample Efficiency	$4.x10^{-7}$%	$4.0x10^{-7}$%

C. Comparison with Fluid Simulation Training

To further assess the effectiveness of our proposed APA approach, we also train a policy in a full fluid simulation environment using the same methodology (*Config. 1*) and test it with a 70% fill level. The number of the environment is 1024, and particle count is 32. As shown in Table III, the policy trained via APA and the one trained in the fluid simulation environment exhibit negligible differences in fluid transfer efficiency, number of steps, and sample efficiency. This result demonstrates that APA can serve as a viable substitute for high-fidelity fluid simulation without compromising policy performance. Moreover, compared to fluid simulation, DexPour achieves an 81.6% reduction in per-iteration training time, with only a marginal 14.0% increase in memory utilization. This improvement underscores the significant computational savings conferred by our approach, making it both scalable and resource-efficient for complex dexterous manipulation tasks.

D. Discussion and Key Takeaways

DexPour effectively leverages a proxy-based abstraction (APA), employing rigid spheres rather than high-fidelity fluid models, to train a high-DoF dexterous hand for pouring tasks. Our hierarchical reward design, split into four distinct stages, not only accelerates skill acquisition but also ensures stability and accuracy throughout the grasp, transport, and pouring phases. Ablation results further confirm that each component of the reward structure is critical, while comparisons with a full fluid simulation underscore our method's comparable efficiency at a fraction of the computational cost. This high-fidelity performance indicates a strong potential for successful sim-to-real transfer, facilitating the deployment of DexPour on physical systems.

VI. Conclusion

In this paper, we presented DexPour, a computationally efficient learning method for high-DoF dexterous hand fluid manipulation. It eliminates costly fluid simulations by using our APA and hierarchical reward structure, the policy can lift a cup from the floor, stabilize it during transport, and accurately pour liquid into a target container. In particular, when filling the cup to 70% capacity, the learned policy achieves an 92% fluid transfer efficiency, comparable to results trained with full fluid simulation. Our ablation studies underscore the crucial role of carefully designed rewards for alignment, grasp, transport, and pouring. Moreover, APA reduces training time by 81.6% while maintaining comparable performance, emphasizing its value for real-world robotic applications such as household assistance, pharmaceutical mixing, and laboratory automation.

References

[1] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[2] 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.

[3] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges," *Representations, and Algorithms*, p. 82, 2019.

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[5] C. Schenck and D. Fox, "Visual closed-loop control for pouring liquids," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2629–2636.

[6] N. Yoshikawa, M. Skreta, K. Darvish, S. Arellano-Rubach, Z. Ji, L. Bjørn Kristensen, A. Z. Li, Y. Zhao, H. Xu, A. Kuramshin, *et al.*, "Large language models for chemistry robotics," *Autonomous Robots*, vol. 47, no. 8, pp. 1057–1086, 2023.

[7] 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.

[8] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv preprint arXiv:1802.09464*, 2018.

[9] H. Liang, L. Cong, N. Hendrich, S. Li, F. Sun, and J. Zhang, "Multifingered grasping based on multimodal reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1174–1181, 2021.

[10] B. Standard, "How tesla's robots delivered drinks and smiles, but were "not machines"," Oct 2024. [Online]. Available: https://www.business-standard.com/world-news/how-tesla-s-robots-delivered-drinks-and-smiles-but-were-not-machines-124101400457_1.html

[11] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99. Citeseer, 1999, pp. 278–287.

[12] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.

[13] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[14] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on robot learning*. PMLR, 2017, pp. 262–270.

[15] Y. Luo, T. Ji, F. Sun, H. Liu, J. Zhang, M. Jing, and W. Huang, "Goal-conditioned hierarchical reinforcement learning with high-level model approximation," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[16] H. Liang, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, and J. Zhang, "Making sense of audio vibration for liquid height estimation in robotic pouring," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5333–5339.

[17] H. Liang, C. Zhou, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, M. Stoffel, and J. Zhang, "Robust robotic pouring using audition and haptics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 880–10 887.

[18] C. Do, C. Gordillo, and W. Burgard, "Learning to pour using deep deterministic policy gradients," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3074–3079.

[19] G. Narasimhan, K. Zhang, B. Eisner, X. Lin, and D. Held, "Self-supervised transparent liquid segmentation for robotic pouring," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4555–4561.

[20] D. Zhang, Q. Li, Y. Zheng, L. Wei, D. Zhang, and Z. Zhang, "Explainable hierarchical imitation learning for robotic drink pouring," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3871–3887, 2021.

[21] 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.

[22] Feb 2025. [Online]. Available: https://docs.omniverse.nvidia.com/extensions/latest/ext$_physics/physics-particles.html$