GOATS: Goal Sampling Adaptation for Scooping with Curriculum Reinforcement Learning

Yaru Niu[1*], Shiyu Jin[2,†], Zeqing Zhang[2,3,†], Jiacheng Zhu[1], Ding Zhao[1], Liangjun Zhang[2]

I. Introduction

Scooping is an instinctive and straightforward skill for humans to acquire. We utilize spoons and shovels to scoop fluid and granular materials. This skill can be generalized to a variety of tasks, from ladling soup and collecting peas with a spoon on a dining table to excavating soils on a construction site. While a small amount of works have studied autonomous robotic scooping [1]–[4], it is still a challenging task for robots due to the high-dimensional interaction space involving the end-effector and the dynamic materials. What is more, because of the complex dynamics of the fluid materials (e.g., water), few prior works have investigated and formulated the problem of fluid or water scooping. Meanwhile, goal-conditioned water scooping can be very helpful in industry or daily life, as it can bring convenience for downstream tasks, such as water transportation [5], water pouring [6], [7], and caregiving [8], [9].

In this work, we formulate the problem of goal-conditioned water scooping, and propose a goal sampling adaptation method for curriculum reinforcement learning (RL) method to solve long-horizon goal-conditioned scooping tasks. As shown in Figure 1, our proposed method can successfully scoop a specific amount of water from a water tank with small errors, and then reach a desired goal position with different containers in both simulations and physical robot settings. This task presents three main challenges. Firstly, it is a long-horizon task with a multi-modal goal state space which incorporates the position and water amount goals, so the policy is required to learn different types of motions to reach both goals, i.e., the container needs to first move downwards to scoop a targeted amount of water, and then lift to reach the desired position goal. Secondly, the initial state of the task is randomly initialized over different water states, and a large space of position goals and water amount goals. Thus, the policy needs to accommodate a wide range of high-dimensional random situations and has good generalizability. Thirdly, the water dynamics is complex, controlling the water amount of scooping under various changing conditions is nontrivial. Our work is related to previous efforts in water manipulation [10]–[12], learning for goal-conditioned deformable object manipulation [13]–[16], and goal-conditioned curriculum RL [17]–[19], while no prior approaches have been proposed for the long-horizon robotic water scooping tasks with multi-modal goals. To this end, our work is developed to solve these challenges, and we summarize our contributions as follows.

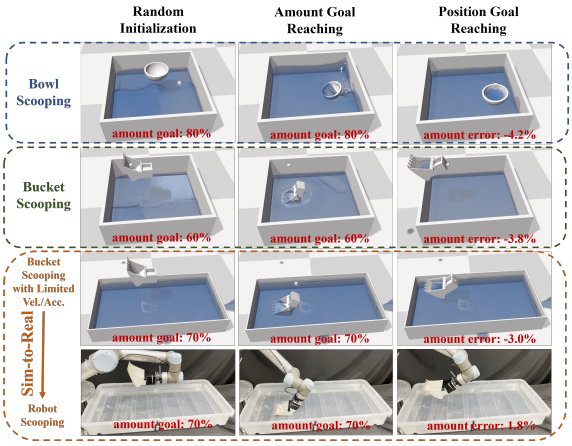


Fig. 1: This figure depicts our goal-conditioned water scooping tasks. The task is randomly initialized over different water states (i.e., waterlines and fluctuations in the tank), different targeted water amounts and targeted positions (shown as a small white box). Our method can scoop the water to the targeted place with a small amount error using different containers in simulation, and can generalize well to real-robot scooping under various configurations.

- To the best of our knowledge, we are the first to formulate and benchmark the tasks of goal-conditioned water scooping with RL.
- We propose a goal-factorized reward formulation and a novel goal sampling adaptation method, GOATS, for efficient curriculum RL on our water scooping tasks.
- Our proposed method achieves low amount errors in simulative scooping tasks with a large number of variations of initial water states and desired goal space, and demonstrate good generalizability in challenging real-robot scooping and out-of-distribution tasks. The videos of this work are available on our project page: https://sites.google.com/view/goatscooping.

II. Methodology

A. Problem Formulation for Water Scooping

In this paper, we formulate the water scooping task as a goal-conditioned RL task. We aim to learn a policy parameterized by θ, π_θ, to control the container to scoop a specific amount of water in the tank and move to a targeted position above the tank. At the start of each episode, the initial water state in the tank, which encloses the water

[1]Carnegie Mellon University
[2]Robotics and Autonomous Driving Lab, Baidu Research, USA
[3]The University of Hong Kong
†Authors contributed equally to this work

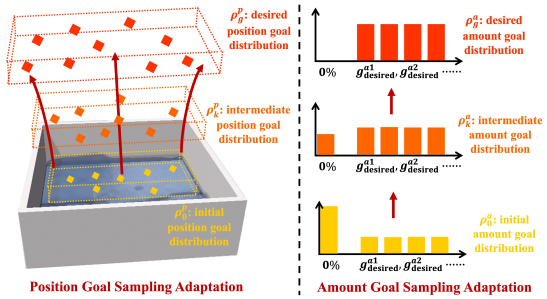


Fig. 2: This figure demonstrates the process of position goal sampling adaptation and the amount goal sampling adaptation. Here, diamonds on the left are samples from the desired, interpolation, or initial distributions.

amount, dynamics, and the initial position of the container, is sampled from the environment's initial state distribution ρ_0. Meanwhile, the desired goal state $g_{\text{desired}} = \{g^p_{\text{desired}}, g^a_{\text{desired}}\}$ is sampled from a goal distribution ρ_g. Here, $g^p_{\text{desired}} \in \mathbb{R}^3$ is the desired position goal of the container in the workspace, and $g^a_{\text{desired}} \in [0\%, 100\%]$ is the desired water amount goal in the container. At time step t, the scooping policy π_θ will take in an observation o_t from the environment, the desired goal state g_{desired} (fixed through an episode), and an achieved goal state g^t_{achieved}, and output a policy distribution $\pi_\theta(o_t, g_{\text{desired}}, g^t_{\text{achieved}})$. The achieved goal state $g^t_{\text{achieved}} = \{g^{p(t)}_{\text{achieved}}, g^{a(t)}_{\text{achieved}}\}$ is a mapping or a subspace vector from the current observation o_t. Here, it shares the same space as g_{desired}, and includes the real-time position of the container $g^{p(t)}_{\text{achieved}}$ and the water amount in the container $g^{a(t)}_{\text{achieved}}$. From the policy distribution $\pi_\theta(o_t, g_{\text{desired}}, g^t_{\text{achieved}})$, an action a_t can be sampled and executed. Then, the agent will obtain an observation for the next time step o_{t+1}, a newly achieved goal $g^{t+1}_{\text{achieved}}$, and a reward $r_t = r(g^{t+1}_{\text{achieved}}, g_{\text{desired}})$, where $r(\cdot)$ is the reward function of the water scooping task.

B. Goal-Factorized Reward Formulation

We represent $g^{t+1}_{\text{achieved}}$ and g_{desired} as the concatenated vectors of their position goal and amount goal vectors (e.g., $g^{p(t+1)}_{\text{achieved}} || g^{a(t+1)}_{\text{achieved}}$ and $g^p_{\text{desired}} || g^a_{\text{desired}}$), respectively. We propose to factorize the goal states and construct a hierarchical reward function combining sparse and dense reward formulations as follows:

$$\begin{aligned} &r(g_{\text{desired}}, g^{t+1}_{\text{achieved}}) \\ =&\mathbb{1}(\|g^{p(t+1)}_{\text{achieved}} - g^p_{\text{desired}}\| \leq \epsilon)\|g^{a(t+1)}_{\text{achieved}} - g^a_{\text{desired}}\| - 1 \end{aligned} \tag{1}$$

This reward function means that dense positive feedback will be produced when the container is close enough to the position goal. It takes advantage of both the binary sparse reward and the shaped (but simple) dense reward, and thus it can help to both position goal and amount goal reaching.

C. Curriculum Learning via Factorized Goal Sampling Adaptation

We propose Goal Sampling Adaptation for Scooping (GOATS), which performs factorized goal sampling adaptation by generating intermediate position goal distributions and amount goal distributions. We represent the desired goal distribution as $\rho_g = \{\rho^p_g, \rho^a_g\}$ which decomposes ρ_g into a desired position goal distribution ρ^p_g and a desired amount goal distribution ρ^a_g. First, an initial position goal distribution, ρ^p_0, and an initial amount goal distribution, ρ^a_0, are selected to represent the initial state or initial achieved goal distribution of the task. Then the intermediate distributions, ρ^p_k and ρ^a_k, are generated by interpolating between the initial goal distributions (e.g., ρ^p_0 and ρ^a_0) and the desired goal distribution (e.g., ρ^p_g and ρ^a_g), respectively. Then the adaptive desired position goals $g^{p(k)}_{\text{desired}}$ and position goals $g^{a(k)}_{\text{desired}}$ that represent simpler tasks can be sampled from ρ^p_k and ρ^a_k. A principled approach to measure the task distribution similarity is the 2-Wasserstein distance [20]. Thus, the interpolations are the Wasserstein barycenters on a geodesic,

$$\rho^p_k := \arg\min_{\rho'} (1-k)W(\rho^p_0, \rho') + kW(\rho', \rho^p_g), \tag{2}$$

$$\rho^a_k := \arg\min_{\rho'} (1-k)W(\rho^a_0, \rho') + kW(\rho', \rho^a_g), \tag{3}$$

where $W(\cdot,\cdot)$ denotes the Wasserstein distance between two distributions, $k \in [0,1]$ is a temporal factor to indicate the procedure of the curriculum learning.

Algorithm 1 Goal Sampling Adaptation for Scooping (**GOATS**) with Curriculum RL

Input: Desired position and amount goal distributions $\{\rho^p_g, \rho^a_g\}$, initial position and amount goal distributions $\{\rho^p_0, \rho^a_0\}$, reward function $r(\cdot)$, initialized policy θ, replay buffer R
Output: Learned policy π_θ
for `each episode` **do**
  Update temporal factor k for curriculum learning
  Update adaptive desired goal distributions ρ^p_k and ρ^a_k with k, ρ^p_g, ρ^p_0, ρ^a_g and ρ^a_0 by solving Eq. (2, 3)
  Sample adaptive desired goals $g^{p(k)}_{\text{desired}} \sim \rho^p_k$, $g^{a(k)}_{\text{desired}} \sim \rho^a_k$
  for `each step` t **do**
    Sample action: $a_t \sim \pi_\theta(o_t, g^k_{\text{desired}}, g^t_{\text{achieved}})$
    Step environment: $o_{t+1} \sim p(o_{t+1}|o_t, a_t)$
    Get $g^{t+1}_{\text{achieved}}$ from o_{t+1}
    Compute reward: $r_t = r(g^k_{\text{desired}}, g^{t+1}_{\text{achieved}})$
    Update replay buffer R
    Update policy θ via SAC [21], HER [17]
  end for
end for

As shown in Figure 2, in our scooping task, ρ^p_0 is a uniform distribution over a cuboid region near the bottom of the water tank, ρ^p_g is a uniform distribution over positions on a cuboid region that encloses the targeted region, ρ^a_0 is a distribution over the 0% water amount, and equally distributed desired (discrete) amount goals, and ρ^a_g is a uniform distribution over only the desired (discrete) amount goals. We provide an algorithm for GOATS in Algorithm 1.

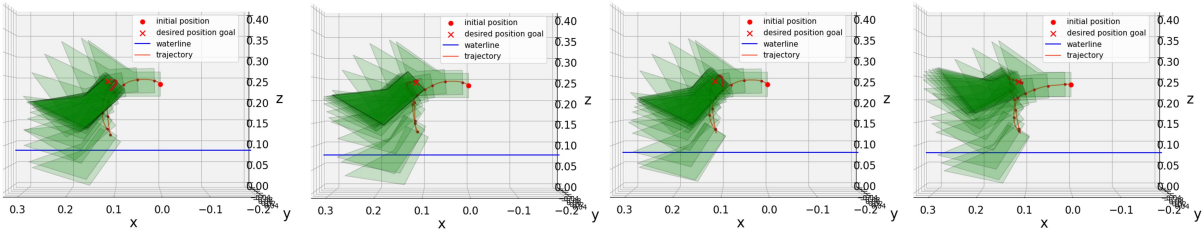
III. Experiments

A. Experiment Setup

We design the task and build our simulated scenarios based on SoftGym [5], a 3D simulator for deformable object manipulation by RL. We test all the methods on two types of containers, including a bowl and a bucket, which have different shapes, sizes, and volumes, as shown in Figure 1. At the start of each episode, the initial water state is sampled from more than 500 variations that include very shallow waterlines. The baselines in simulation are composed of one or several method elements including **SAC**, **HER**, **Universal Goal Sampling (GS)**, and **Partially Adaptive GS**, where

TABLE I: In this table, we display the performance in simulation with a single amount goal (70%) and multiple amount goals ($60\%, 65\%, 70\%, 75\%, 80\%$). The results are averaged over the best evaluation rewards in 3 seeds during training. The corresponding model for each seed is then evaluated on 100 episodes, with randomly sampled initial waterlines, position goals, and water amount goals, to obtain the absolute amount error.

Method	Bowl Scooping				Bucket Scooping			
	Single Amount Goal		Multi. Amount Goals		Single Amount Goal		Multi. Amount Goals	
	Reward ↑	Amount Error↓	Reward↑	Amount Error↓	Reward↑	Amount Error↓	Reward↑	Amount Error↓
SAC	-69.41 ± 0.78	$69.60\% \pm 0.33\%$	-61.21 ± 2.00	$71.02\% \pm 0.34\%$	-71.20 ± 1.12	$69.99\% \pm 0.01\%$	-69.47 ± 0.77	$71.00\% \pm 0.35\%$
SAC+HER	-72.72 ± 0.32	$67.28\% \pm 1.66\%$	-69.59 ± 2.32	$63.36\% \pm 5.91\%$	-73.40 ± 0.47	$52.35\% \pm 13.79\%$	-72.15 ± 1.05	$55.76\% \pm 0.35\%$
SAC+Universal GS	-71.7 ± 0.69	$69.51\% \pm 0.40\%$	-72.05 ± 0.41	$71.02\% \pm 0.34\%$	-72.96 ± 0.65	$70.00\% \pm 0.00\%$	-71.48 ± 1.28	$70.83\% \pm 0.43\%$
SAC+Partially Adaptive GS	-72.89 ± 0.59	$70.00\% \pm 0.00\%$	-71.87 ± 0.18	$67.51\% \pm 2.23\%$	-73.73 ± 0.24	$69.81\% \pm 0.15\%$	-73.14 ± 1.01	$70.98\% \pm 0.35\%$
SAC+HER+Universal GS	-36.45 ± 4.41	$26.18\% \pm 14.33\%$	-37.88 ± 2.48	$11.24\% \pm 2.51\%$	-42.48 ± 1.04	$12.76\% \pm 2.60\%$	-37.32 ± 1.24	$13.39\% \pm 0.69\%$
SAC+HER+Partially Adaptive GS	-28.80 ± 0.41	$8.54\% \pm 1.11\%$	-28.98 ± 0.43	$7.43\% \pm 1.41\%$	-35.22 ± 0.35	$\mathbf{9.61\% \pm 2.68\%}$	-33.12 ± 0.60	$14.16\% \pm 3.16\%$
GOATS (Ours)	$\mathbf{-25.67 \pm 0.32}$	$\mathbf{5.93\% \pm 1.20\%}$	$\mathbf{-25.77 \pm 0.60}$	$\mathbf{4.99\% \pm 0.37\%}$	$\mathbf{-33.36 \pm 0.69}$	$9.97\% \pm 2.09\%$	$\mathbf{-32.51 \pm 0.61}$	$\mathbf{7.45\% \pm 1.65\%}$



(a) AG 60%, PG 2, WL 8.5 cm (b) AG 60%, PG 2, WL 7.5 cm (c) AG 60%, PG 2, WL 8 cm (d) AG 70%, PG 2, WL 8 cm

Fig. 3: This figure depicts trajectories under different scooping conditions for the UR5 robot. Here, all initial positions are at 23 cm above the ground. AG means the amount goal, PG means the position goal (all cases have the same PG), and WL means the initial waterline. The bucket dives deeper when the initial waterline is lower and the targeted amount is larger.

TABLE II: In this table, we display the absolute amount errors in both the sim-to-real simulation and the real-robot environments using **a single trained model** by GOATS. Each value is an average over three tested position goals.

Initial Position	Waterline (cm)	60% Amount Goal		65% Amount Goal		70% Amount Goal	
		Sim.	Robot	Sim.	Robot	Sim.	Robot
Height 23cm	7.5	$3.33\% \pm 0.57\%$	$1.66\% \pm 0.75\%$	$3.80\% \pm 0.70\%$	$6.58\% \pm 3.06\%$	$4.91\% \pm 0.91\%$	$11.27\% \pm 1.75\%$
	8.0	$6.15\% \pm 1.12\%$	$4.39\% \pm 2.94\%$	$3.47\% \pm 0.60\%$	$2.67\% \pm 0.98\%$	$4.93\% \pm 0.48\%$	$4.95\% \pm 2.62\%$
	8.5	$6.52\% \pm 1.13\%$	$5.64\% \pm 0.83\%$	$4.67\% \pm 0.37\%$	$3.74\% \pm 1.51\%$	$5.13\% \pm 0.29\%$	$5.31\% \pm 2.79\%$
Height 30cm	7.5	$3.66\% \pm 0.23\%$	$1.76\% \pm 0.18\%$	$5.79\% \pm 1.20\%$	$7.83\% \pm 1.95\%$	$5.08\% \pm 0.92\%$	$11.43\% \pm 1.73\%$
	8.0	$3.98\% \pm 0.30\%$	$4.78\% \pm 1.81\%$	$3.64\% \pm 1.53\%$	$1.83\% \pm 1.19\%$	$3.28\% \pm 0.52\%$	$5.07\% \pm 2.14\%$
(Unseen in training)	8.5	$6.53\% \pm 0.91\%$	$6.90\% \pm 0.99\%$	$5.17\% \pm 1.25\%$	$2.93\% \pm 1.14\%$	$4.25\% \pm 0.75\%$	$1.78\% \pm 0.73\%$
Height 40cm	7.5	$4.31\% \pm 2.31\%$	$6.96\% \pm 0.44\%$	$5.21\% \pm 0.99\%$	$6.31\% \pm 0.89\%$	$3.92\% \pm 0.50\%$	$11.47\% \pm 1.99\%$
	8.0	$3.94\% \pm 0.56\%$	$2.29\% \pm 0.73\%$	$3.06\% \pm 1.34\%$	$3.81\% \pm 1.64\%$	$4.13\% \pm 1.01\%$	$6.95\% \pm 0.53\%$
(Unseen in training)	8.5	$6.34\% \pm 1.88\%$	$7.90\% \pm 1.97\%$	$5.63\% \pm 2.06\%$	$3.85\% \pm 0.52\%$	$5.40\% \pm 0.44\%$	$5.11\% \pm 3.04\%$

Universal GS maintains universal stable distributions for both position and amount goals, and Partially Adaptive GS only maintains a universal stable distribution for amount goals. Furthermore, we evaluate our proposed method in real-robot scooping using a UR5 robot (Figure 1) by limiting the bucket velocity and acceleration in simulation and zero-shot transfer.

B. Results in Simulation

We demonstrate our results from the simulation in Table I. GOATS can achieve higher rewards and lower success water amount error than other methods, showing that our method can successfully finish both the position goal and amount goal reaching tasks. Comparing the amount errors between using a single amount goal and multiple amount goals on the same task, we can tell that training on multiple goals can be beneficial and improve the amount goal-reaching performance. SAC+HER+Partially Adaptive GS is always better than SAC+HER+Universal GS, from which we can conclude that performing position goal sampling adaptation is very helpful to our scooping tasks. Meanwhile, the performance discrepancy between GOATS and SAC+HER+Partially Adaptive GS shows the effectiveness of the amount of goal sampling adaption. The results here indicate that GOATS can accommodate complex water dynamics, different position and amount goals, and different types of containers in the water scooping task.

C. Results in Real-Robot Scooping

We demonstrate trajectories from real-robot scooping in Fig. 3. Comparing Fig. 3 (a) and (b), the trained policy can adaptively adjust the trajectory from the identical start point and PG to meet the same AG, and the bucket dives deeper when the waterline is lower, resulting in amount errors 5.92% and 3.07%, respectively. Furthermore, with the same WL, Fig. 3 (c) and (d) show that the bucket can dive deeper to get more water when the targeted amount is larger. The corresponding amount errors on the robot are 2.76% and -3.31%, respectively. This shows that our trained policy can adapt to different water states in the tank and adjust scooping schemes to reach different amount goals on the physical robot. In Table II, We display the average absolute amount errors in both robot and (sim-to-real) simulation settings. We can find that the amount errors of robot scooping are under 8% in most cases, and the sim-to-real gap is small. We directly apply the trained policy to unseen initial bucket positions in training without fine-tuning. Compared to the performance on the in-distribution tasks (23cm), our method shows no evident performance drop on two more difficult out-of-distribution tasks (30cm, 40cm), and are surprisingly better at various amount goal and waterline settings. This shows that GOATS has good generalizability and are empowered with the potential to achieve more complicated tasks when only training on simpler ones.

REFERENCES

[1] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Conference on Robot Learning*. PMLR, 2017, pp. 239–248.

[2] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Learning compositional models of robot skills for task and motion planning," *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 866–894, 2021.

[3] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg, "Rethinking optimization with differentiable simulation from a global perspective," *arXiv preprint arXiv:2207.00167*, 2022.

[4] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, "Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds," *arXiv preprint arXiv:2211.09006*, 2022.

[5] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, 2020.

[6] T. Tsuji and Y. Noda, "High-precision pouring control using online model parameters identification in automatic pouring robot with cylindrical ladle," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 2563–2568.

[7] T. Chen, Y. Huang, and Y. Sun, "Accurate pouring using model predictive control enabled by recurrent neural network," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7688–7694.

[8] Z. Erickson, Y. Gu, and C. C. Kemp, "Assistive vr gym: Interactions with real people to improve virtual assistive robots," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 299–306.

[9] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp, "Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned," *Robotics and Autonomous Systems*, vol. 124, p. 103344, 2020.

[10] A. LaGrassa and O. Kroemer, "Planning with learned model preconditions for water manipulation," 2022.

[11] T. L. Guevara, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Subr, "Adaptable pouring: Teaching robots not to spill using fast but approximate fluid simulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, vol. 2, 2017.

[12] L. Rozo, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *9th International Workshop on Robot Motion and Control*. IEEE, 2013, pp. 227–232.

[13] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[14] R. Jangir, G. Alenya, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4630–4636.

[15] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation," in *International Conference of Robotics and Automation (ICRA)*, 2022.

[16] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.

[17] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[18] D. Warde-Farley, T. Van de Wiele, T. Kulkarni, C. Ionescu, S. Hansen, and V. Mnih, "Unsupervised control through non-parametric discriminative rewards," *arXiv preprint arXiv:1811.11359*, 2018.

[19] S. Pitis, H. Chan, S. Zhao, B. Stadie, and J. Ba, "Maximum entropy gain exploration for long horizon multi-goal reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7750–7761.

[20] C. Villani *et al.*, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[21] 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.