Robotica (2025), **43**, pp. 2320–2332
doi:10.1017/S0263574725001444

RESEARCH ARTICLE

Manipulate as human: learning task-oriented manipulation skills by adversarial motion priors

Ziqi Ma[1], Changda Tian[2] and Yue Gao[3,4]

[1]ParisTech Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai, P.R. China
[2]Department of Automation, Shanghai Jiao Tong University, Shanghai, P.R. China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, P.R. China.
[4]Shanghai Innovation Institute, Shanghai, P.R. China
Corresponding author: Yue Gao; Email: yuegao@sjtu.edu.cn

Received: 8 May 2024; **Revised:** 8 January 2025; **Accepted:** 17 April 2025; **First published online:** 11 June 2025

Keywords: human-like manipulation; task-oriented; motion prior; discriminator; hammering

Abstract

In recent years, there has been growing interest in developing robots and autonomous systems that can interact with human in a more natural and intuitive way. One of the key challenges in achieving this goal is to enable these systems to manipulate objects and tools in a manner that is similar to that of humans. In this paper, we propose a novel approach for learning human-style manipulation skills by using adversarial motion priors, which we name HMAMP. The approach leverages adversarial networks to model the complex dynamics of tool and object manipulation and the aim of the manipulation task. The discriminator is trained using a combination of real-world data and simulation data executed by the agent, which is designed to train a policy that generates realistic motion trajectories that match the statistical properties of human motion. We evaluated HMAMP on one challenging manipulation task: hammering, and the results indicate that HMAMP is capable of learning human-style manipulation skills that outperform current baseline methods. Additionally, we demonstrate that HMAMP has potential for real-world applications by performing real robot arm hammering tasks. In general, HMAMP represents a significant step towards developing robots and autonomous systems that can interact with humans in a more natural and intuitive way, by learning to manipulate tools and objects in a manner similar to how humans do.

1. Introduction

Manipulating tools with robot arms is a long-standing area of study in the field of robot intelligence. To effectively manipulate an object with a tool and achieve a specific goal, robots must develop a comprehensive understanding of the environment based on sensor data and then perform intricate physical interactions with targets. Several prior efforts have focused on data-driven methods that aim at learning reliable tool representations for manipulation tasks. In particular, the application of end-to-end deep neural networks has gained popularity for acquiring such representations [1–3]. These methods allow for the generation of latent tool object representations through end-to-end neural networks and extensive datasets, eliminating the need for manually crafting stage pipelines and defining object features. However, these black-box methods often lack interpretability and compactness, and they may not fully account for subsequent manipulation processes, as they do not consider actions beyond the successful tool grasp.

Some works try to use keypoints to define the tool and environment in order to mathematically formulate the manipulation task. In the works of Qin *et al.* [4] and Manuelli *et al.* [5], they express the tool and the task in keypoints and devise optimization algorithms to manipulate actions. By acquiring tool keypoints through supervised or reinforcement learning and defining task keypoints in the environment, these approaches formulate Quadratic Programming problems to generate robot movement trajectories within the task context. Another recent study by Turpin *et al.* [6] also adopts keypoints to represent tool

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

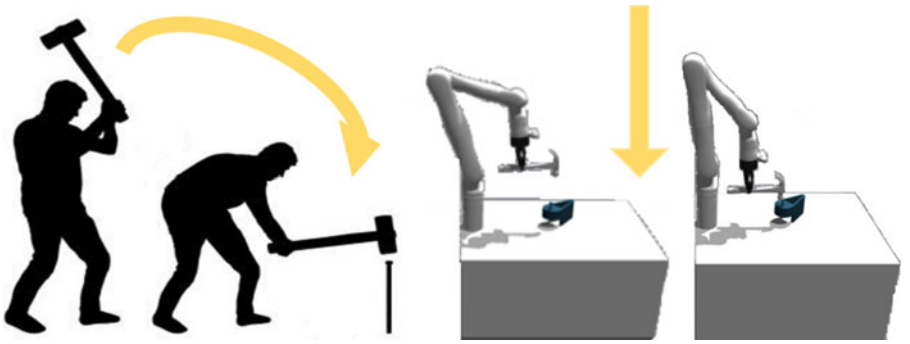


Figure 1. *Difference of hammering between humans and robots. When humans hammer the nail, they swing the hammer in the opposite direction of striking in order to stock energy, while robots only focus on the achievement of task and ignore this important action.*

objects. However, they employ reinforcement learning to predict tool affordances using a complexly designed reward function.

Nevertheless, previous studies focus on achieving task completion in complex manipulation scenarios as they often overlook whether the manipulation trajectory of robot is similar to that of human being. For instance, when faced with intricate tasks, such as hammering a nail shown in Figure 1, these methods typically provide a straightforward policy. After grasping a hammer, they position it directly on the nail and strike down until contact occurs. However, human hammering involves a distinct action: the buildup of kinetic energy by swinging the hammer in the opposite direction before striking above the nail's position. This natural human-style approach stores energy in the hammer during the swing and releases it upon impact with the nail. Such nuanced behavior is challenging to learn using conventional optimization methods because encoding the swing action with constraints in an objective equation proves difficult. The works in refs. [7–9] use various sensors to capture the effects that human conduct on the tool and on the oriented objects to generalize the use of tool from human to robot, these methods provide a preview by learning the human movement on the physical effects; however, the inputs that they use depends on tactile data, it is expensive to gain in the normal life and will limit the propagation of the method.

Recent advances have seen a surge in deep reinforcement learning algorithms for manipulation tasks [10–12] with easy data gain, highlighting their potential for teaching robots tool-based tasks. In imitation learning, there has been progress, with supervised training using human-teleoperated demonstrations [13] or hand-manipulated trajectories [10]. The research in ref. [12] introduces a robot tool manipulation strategy using human manipulation videos. They create a simulation environment aligned with the guidance video and calculate robot states with guided policy samples and trajectory optimization. Although effective, this approach involves solving optimization problems for each aligned environment, which consumes considerable time and computing resources.

In order to teach a robot to learn human manipulation skills in a more natural and intuitive way, we combine adversarial motion priors (AMP) within a reinforcement learning problem. AMP [14] is a cutting-edge approach first appearing in computer graphics, it uses an adversarial network to learn a "style" with a reference motion dataset. The reward function involves a style reward that encourages the agent to replicate similar trajectories to those in the dataset and a task reward that assesses whether the agent achieves the task while mimicking the motion style. We also adapt style and task rewards to teach a robot arm human-like tool manipulation skills using demonstration video clips. Our approach involves competitive training between a policy network and an adversarial network. The policy network is trained using both task-specific and adversarial rewards to generate a policy that accomplishes the task with human style. The adversarial network acts as a discriminator, determining the origin of state transitions and providing a reward that effectively motivates the training of an agent. Due to the use of

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

AMP in guiding the robot to learn human-like manipulation skills, we name the method HMAMP. The contribution of our work is that:

- We introduce an implementation of task-oriented reinforcement learning combined with style in manipulation domain and evaluate its performance on the task hammering.
- We provide an idea where the training data is easily acquired to learn a robot tool manipulating policy in human style.
- We construct an environment of tool manipulation in simulation and verify that the HMAMP is also useful in the real world.

2. Related work

2.1. Manipulation of tools

Tools use has been a fundamental issue in cognitive science studies that seeks to comprehend the nature of intelligence [15–17]. To enable robots to perform complex tasks that require the use of tools, many advanced studies have focused on recognizing affordance-specific features on tool objects [18,19], which describes the potential physical interaction between object and manipulator and associates the regional part to planned sequential actions. These approaches have been successful in equipping robots with the capability to understand how objects may serve different purposes. In addition to recognizing the features of tool objects, learning, and planning are essential components in the manipulation of tools, as demonstrated in various studies [4,6,20]. Researchers have also explored methods of incorporating real-time feedback and environmental factors to improve the accuracy and precision of tool grasping [21,22]. Some studies aim to identify suitable tools for a certain task, they learn an embedding knowledge by DNN model between grasping tool, desired action, and target goal [23,24].

2.2. Learning from human videos

Plenty of recent research has explored utilizing human videos to improve the efficiency of RL in robots. Some works use data from the egocentric view to enable robots learning human skills [25,26]. However because of the variety of data source and the slight difference of view, it becomes difficult to use the pre-trained representation in a specific manipulation task. In order to mend the domain gap, another type of work utilizes the in-domain human demonstration, where the sequence of human pose is recorded by motion capture [27] or by a view from the third person [28]. Data of this kind have a narrower disparity between the human and robot domains, which makes it possible to construct efficient reward function for training imitation learning algorithms. Instead of extracting and re-targeting the whole human pose from the video, we focus on the motion of parts of important joints and the motion of the tool, which allows a flexible transfer from human morphology to robot morphology. By cooperating with the task-oriented approach with an AMP, we enable our system to learn the movement of robot arm using unstructured motion data.

2.3. Generative adversarial imitation learning

Generative adversarial imitation learning (GAIL) [29] is inspired by the idea developed for generative adversarial networks (GAN) [30]. It aims to train generators that learn policies matching the trajectory distribution of the dataset, meanwhile, the discriminators serve as the reward functions to judge whether the generated behaviors look like the demonstrations. By using a small number of demonstration data from experts, GAIL learns both the unknown environment's policy and reward function. Although

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

these methods have demonstrated success in low-dimensional domains [29], their performance in high-dimensional tasks is not outstanding. Recently, Peng *et al.* [14] have introduced AMP, which integrates task goals with generative adversarial imitation learning. This allows simulated agents to perform high-level tasks by learning to imitate behaviors from extensive motion dataset. Escontrela *et al.* [31] also apply this adversarial technique with a limited number of reference motion clips to learn locomotion skills for legged robots. In manipulation areas, we apply this technique to guide robots to interpret and perform behavior shown by demonstration and we show that the agents have more flexibility to perform more natural and feasible behaviors.

3. Learning tool manipulation policy

We model the problem of learning human-like tool-manipulating skills as a Markov Decision Process. The goal of the reinforcement learning is to find the parameter θ that optimize policy π_θ in which the expected discounted return is maximal $J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right]$. We then propose the mathematical abstraction of tool manipulation task and introduce the design of reward function.

3.1. Tool keypoint definition

For each manipulation task with tool, we make the assumption that the tool objects interact with the environment containing one or more target objects. Inspired by ref. [4], in the framework of HMAMP, we define some keypoints in each task manually, which consists of a set of keypoints on the tool $\boldsymbol{K_o}$ and a set of keypoints in the environment $\boldsymbol{K_e}$. Specifically, we consider $\boldsymbol{K_o} = \left[x_g, x_f, x_m\right]$, where $\boldsymbol{x_g}$ characterizes the grasping position on the tools and function point, $\boldsymbol{x_f}$ characterizes the functional part that make contact with the target object and $\boldsymbol{x_m}$ represents an auxiliary point that aides to determine the direction of tool. Environment keypoints are represented as $\boldsymbol{K_e} = [\boldsymbol{x_c}]$, where $\boldsymbol{x_c}$ denotes the position where the target interacts with the tool. The keypoints of tools and environment are mostly used to determine the goal reward which needs to be optimized in a reinforcement learning problem.

3.2. Rewards design

As mentioned in ref. [14], the total reward r_t is determined as a goal reward r_t^g, which is specific to the task, helps to describe the completion of task and a style reward r_t^s that evaluates whether the behaviors produced by the agent is similar to the behaviors produced via the distribution of reference motion set. The more similar the behaviors are, the higher value the style reward is. The proportion between style reward and goal reward is adjusted manually before training.

$$r_t = \alpha^g r_t^g + \beta^s r_t^s \tag{1}$$

3.2.1. Goal reward about task

The goal reward is related to the task and must be designed specifically. For the tool manipulation tasks where there is a contact with environment, we design the goal reward r_t^g composed by two terms at one instance t.

$$\begin{aligned} r_t^g &= \omega^f r_t^f + \omega^d r_t^d \\ r_t^f &= \begin{cases} \|F_t^s(x_f, x_c)\|/F^d & F_t^s \le F^d \\ 1 & F_t^s > F^d \end{cases} \\ r_t^d &= 1 - \tanh\left(\left\|x_f - x_c\right\|_t\right) \end{aligned} \tag{2}$$

The first term r_t^f stimulates task completion, which is defined with the contact force detected on the target object that is exerted by the tool, the force is captured by force sensor in the simulation. The

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

detected force is encouraged to converge towards the force F_d that we desire. This reward has values only when contact occurs and after the detection of contact force, the policy terminates. The second term r_t^d is defined between tool function point x_f and environment target point x_c to guide to exploit a policy that minimizes the distance between tool and target. The utilization of tanh function aims to bind the reward to [0, 1]. These terms are weighted with manually specified coefficients ω^f and ω^d, however, the magnitude of second term is much lower than the first.

3.2.2. Style reward with motion prior

As the idea mentioned in ref. [14], we define a discriminator D_ϕ as a neural network with parameter ϕ, the discriminator is trained to distinguish whether a transition (s, s') is a fake one produced by an agent or a true one sampled from a real motion distribution $d^{\mathcal{M}}$.

We update the objective of discriminator as:

$$\begin{aligned}\underset{\phi}{\operatorname{argmin}} \quad & \mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\left(D_\phi(s,s')-1\right)^2\right] \\ & +\mathbb{E}_{d^{\pi}(s,s')}\left[\left(D_\phi\left(s,s'\right)+1\right)^2\right] \\ & +\frac{w^{gp}}{2}\mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\|\nabla_\phi D_\phi\left(s,s'\right)\|^2\right]\end{aligned} \tag{3}$$

The former two terms serve to motivate the discriminator to differentiate between the input state derived from a policy and the input state derived from the reference motion data. They are proposed in LSGAN [32] to solve the challenge of vanishing gradients caused by the standard GAN objective function where a sigmoid cross-entropy loss function is usually used. LSGAN prefer to optimize χ^2 divergence between the reference distribution and the policy distribution, which may alleviate the mode collapse problem and lead to stable performance during the training process [33]. The last term in (3) is a gradient penalty term to penalize nonzero gradients on samples, which may avoid oscillations and improve training stability. The w^{gp} in the formula is a coefficient adjusted manually.

The style reward is then defined by:

$$r_t^s = \max\left[0, 1-\gamma^d(D_\phi(s,s')-1)^2\right] \tag{4}$$

with the additional offset and scale, the style reward is bounded between [0, 1].

The training process of policy and discriminator is shown in Figure 2. The agent steps to interact with environment and produces a state transition (s, s'), the observation in the environment is used to calculate the goal reward r_t^g. The discriminator takes the state transition from a simulated environment and from a reference motion clips to calculate the style reward r_t^s. In the end, the combined reward is used to optimize competitively the policy and the discriminator. The training details are demonstrated in Algorithm 1.

4. Training

4.1. Data preprocess

The raw data used for HMAMP consist of video clips capturing manipulation skills from a third-person perspective. This type of data offers several advantages: it minimizes the domain gap between simulation and reality, and it is relatively easy to acquire, making it a practical choice for training purposes. In this work, we create the dataset which records human skill: hammering. The duration of the human motion in each video clips is less than one second, and we collect five pieces of video clips that perform the hammering movement by two persons. Then, we use the most popular keypoint detection algorithm BlazePose [34] to detect human joint including Hip, Elbow, Wrist, Hand, and we use a CV algorithm to extract time-series of tool keypoints by manually signing them on the tool.

To retarget human motion to robot motion, it is necessary to construct an effective transfer function that maps the human world space to the robot world space. Numerous studies [35–37] have explored

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

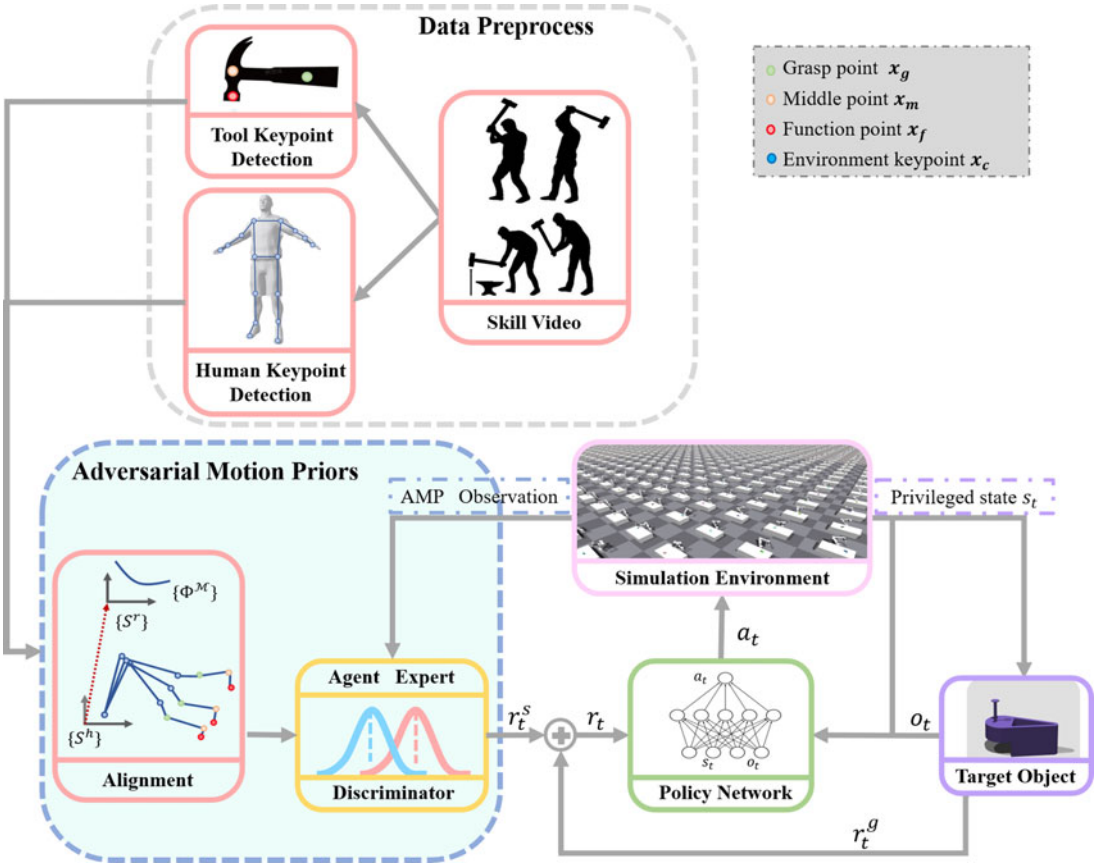


Figure 2. *Framework of HMAMP. With human manipulation video clips, we extract the keypoints of human arm and manipulation tools. Then we do keypoints alignment between robot arm in simulation and real-world human motion clips. The AMP Discriminator is to discriminate whether an action sequence is a real human expert motion or generated by the policy network. The AMP reward and task reward for manipulation task is added to be the total reward for RL training.*

motion retargeting between these two domains, and any established retargeting method can be applied in this process. In our work, we adopt a simple and straightforward approach: direct mapping. While there are significant differences in topology between the human arm and the robot arm, they share corresponding joints, such as the human elbow, wrist, and hand, which align with certain robot joints and the end-effector. By mapping key human joints to their robotic counterparts, we achieve a rough but effective transfer of motion from the human domain to the robot domain.

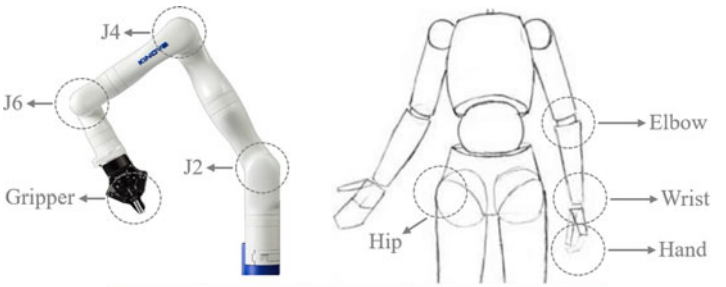
4.2. Model details

The policy that we used in HMAMP is PPO, the hidden parameters of policy network are of size [512, 256, 128] with exponential linear unit activation layers. The policy outputs the distribution from which the target joint positions are sampled with the representation of mean and standard deviation. Then, the target joint positions are fed to our customized PD controllers to compute the motor torques. The policy is trained on an observation o_t derived from the state, which contains environment information such as hammer position and nail position and robot information such as joint angles, joint velocities, end-effector orientation, and previous actions. The discriminator is an MLP with hidden layers of size [1024, 512] and exponential linear unit activation layers. It takes the orientation of robot joints

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

Algorithm 1. HMAMP: Learning Human-like Manipulation Skills by Adversarial Motion Prior

Input: Task-specific environment $\mathcal{E}$, Dataset of human reference motions $\mathcal{M}$
Initialize AMP discriminator D
Initialize policy π
Initialize reply buffer $\mathcal{B}$
for each training episode **do**
  Reset environment $\mathcal{E}$ to initial state
  for trajectory $i = 1, \ldots, m$ **do**
    collect trajectory τ_i with π: $\tau_i \leftarrow \{(s_t, a_t, r_t^g)_{t=0}^{T-1}, s_T^g\}$
    for time step $t = 0, \ldots, T-1$ **do**
      $d_t \leftarrow D\phi(s_t, s_t')$
      calculate r_t^s by Eq. (4).
      calculate r_t by Eq. (1).
      record r_t in τ_i.
    end for
    store τ_i in $\mathcal{B}$.
  end for
  for update step $= 1, \ldots, n$ **do**
    $b^{\mathcal{M}} \leftarrow$ sample batch of K transitions from $\mathcal{M}$.
    $b^{\pi} \leftarrow$ sample batch of K transitions from $\mathcal{B}$.
    update D according to Eq. (3) using $b^{\mathcal{M}}$ and b^{π}.
  end for
  update π using data from trajectories $\{\tau_i\}_{i=1}^m$
end for



Robot	J2	J4	J6	Gripper
Human	Hip	Elbow	Wrist	Hand

Figure 3. *Direct mapping between human and robot arm. Some joints and the gripper of a Kinova Gen3 are mapped to act as human hip, elbow, wrist, and hand.*

and the orientation of tools as input. The value of all manually determined parameters is: $\alpha^g = 0.6$, $\beta^s = 0.4$, $\gamma^d = 0.25$, $\omega^f = 10^5$, $\omega^d = 1$, $\omega^{gp} = 1$, $F^d = 100$.

4.3. Simulation

The robot that we choose to train the policy is Kinova Gen3 with the gripper 2f85, the correspondence joint mapping result between Gen3 and human is shown in Figure 3.

We selected Isaac Gym [38] as our simulation platform due to its ability to accelerate the RL training process using GPU resources. The policy was trained in parallel across 2048 agents, utilizing a single

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

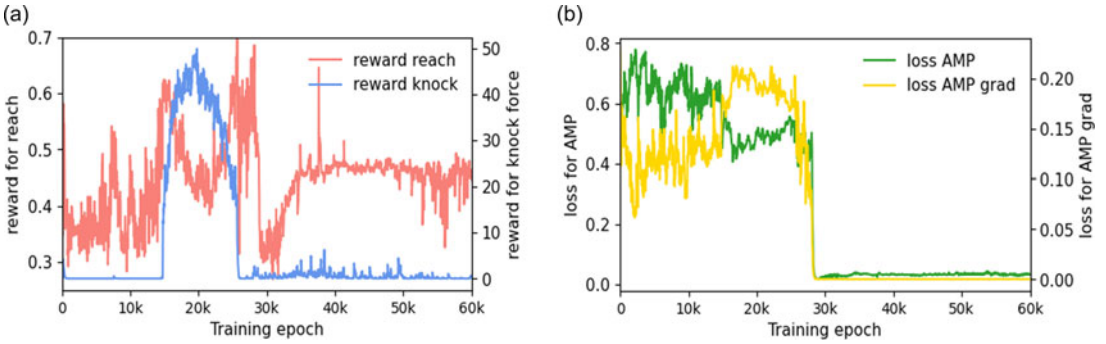


Figure 4. *Training curves of HMAMP. Left figure shows the evolution of reach reward and knock force reward, while right figure indicates the discriminator loss and gradient in the training process. The two figures show the confrontation and balance between style reward and goal reward. In the early stage goal reward has a strong guiding effect while in the late stage amp discriminator converges quickly, giving the trajectory of robot a human style.*

NVIDIA RTX 3090 GPU. The entire training process required 11 h of wall-clock time and spanned 60,000 training epochs. Each RL episode lasted a maximum of 152 steps, corresponding to 3 s of simulated time, and terminated early if the termination criteria were met. The policy operated at a control frequency of 50 Hz during the simulation.

4.4. Termination

An episode terminates and the next one starts when the robot satisfy the termination criteria. It contains task finish signal when the hammer knocks the nail and the collision signal when a force is detected between robot components or between the robot and the table or the hammer and the table.

4.5. Domain randomization and training process

In order to improve the robustness of HMAMP policy and facilitate the transfer of learned policy from simulation to the real world, we apply domain randomization in the training process. In detail, we randomize the coefficient of friction applied to hammer and nail to [0.5, 1.25] and the joint-level PD gains to [0.9, 1.1]. In addition, the same observation noise as in ref. [12] is added during the training phase, where the Cartesian position observation noise is ± 0.01 m and joint position observation noise is ± 0.02 rad.

From Figure 4, the training curve shows the confrontation and balance between style reward and goal reward. The early stage goal reward has a strong guiding effect, while in the late stage, amp discriminator quickly converges, giving the movements of robot a human style.

5. Experiment

In this section, we present the experimental setup and results to demonstrate the effectiveness of HMAMP in learning task-oriented, human-like manipulation skills. To evaluate the performance of our method, we conducted a series of comparative experiments against two baseline approaches: a direct path-planning control policy and a reinforcement learning (RL) approach without AMP. The experimental results were recorded and analyzed quantitatively to highlight the contributions of HMAMP. The evaluation focused on three key aspects: the quality of the learned manipulation skills, task completion efficiency, and the similarity of the robot's movements to human behavior.

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

5.1. Comparative experiment in simulation

5.1.1. Task definition

The chosen manipulation task for the experiments involves knocking a nail with a hammer. The experiments are conducted on the Isaac Gym simulation platform, using the model of a 7-dof Kinova Gen3 Arm with 2f-85 gripper to complete the task. At the start of the task, the hammer is securely grasped by the robotic arm, which begins in its initial home position, with the gripper oriented perpendicular to the platform (see Figure 6). The nail is placed arbitrarily on the manipulation platform. The objective of the task is to successfully hammer the nail while replicating a human style of movement.

5.1.2. Baseline methods

We compare HMAMP against the following baseline methods:

- Direct Path-Planning Control Policy (DPPCP): This baseline approach determines manipulation actions using predefined path-planning strategies. In this method, proportional-derivative (PD) control is employed to generate a planned trajectory guiding the hammer from its initial position to the nail.
- Reinforcement Learning without AMP (RL-noAMP): This baseline approach uses a standard reinforcement learning method for the agent to acquire manipulation skills. The configuration of this approach is identical to that of HMAMP, except that the AMP component is removed. The training process follows the same procedure as in HMAMP.

5.1.3. Evaluation metrics

To quantitatively evaluate the performance of each approach, we define the following criteria:

- **Knock Impulse**: A measure of the knock effect received by the nail. It is calculated by the formula : $I = \int F_{nail}(t)\, dt$. Large Impulse means the nail receives large force at one instance.
- **Energy Efficiency**: The energy used by the arm is $E = \sum_{i=1}^{n} \int \tau_i(t) \cdot \omega_i(t)\, dt$, where n is the joint number of the arm. The energy efficiency can be represented as the ratio of knock impulse received by the nail and the energy cost by the arm: $\eta = I/E$
- **Vertical Force Ratio**: The vertical force ratio reflects how efficiently the force is applied in the vertical direction, which is critical for tasks involving hammering. Higher ratios indicate more effective and efficient nail hammering: Vertical Force Ratio $= \frac{F_{\text{vertical, nail}}}{F_{\text{total, nail}}}$
- **Frechet Distance**: The quantitative analysis of motion similarity between human and robot arm manipulation [39]. The smaller the Frechet distance between two trajectories is, the more similar their shapes are.

5.1.4. Results in simulation

We implemented the two baseline control strategies, DPPCP and RL-noAMP, as described in Section 5.1.2, within the simulation environment. Comparative experiments were conducted to evaluate their performance against our proposed method. Each method was tested across 10 trials, and the average values for each evaluation criterion were calculated to ensure stable and reliable performance measurements.

Table I presents a comprehensive comparison of the various approaches, highlighting key performance metrics. The HMAMP consistently outperforms both DPPCP and RL-noAMP in all evaluated aspects. Notably, it has an overwhelming advantage in the impulse received by the nail, which is the most critical task of hammering nails. From the table, we can also obtain that the HMAMP is more

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

Table I. *Comparison results of HMAMP with baselines.*

Method	HMAMP	DPPCP	RL-noAMP
Knock impulse	4238 kg m/s	2351 kg m/s	2507 kg m/s
Energy efficiency	1.56 s/m	1.34 s/m	1.26 s/m
Vertical force ratio	0.99	0.99	0.91
Frechet distance	0.29	0.88	1.36

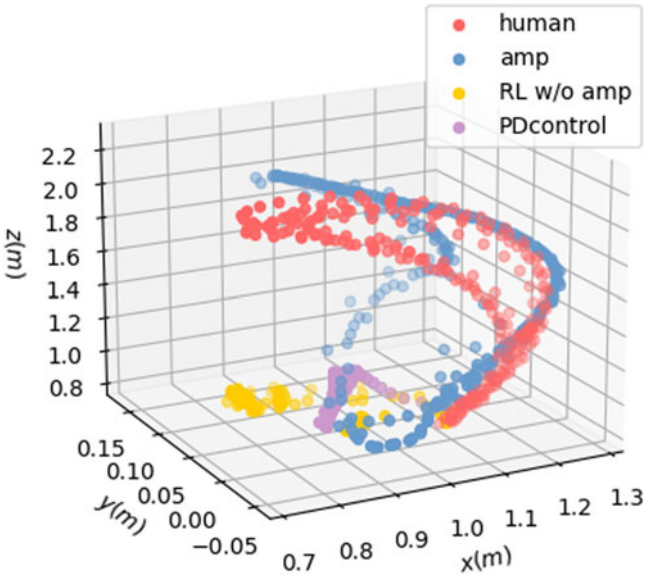


Figure 5. *Movement trajectory of the end of the robotic arm in Cartesian space. The end-of-arm motion trajectory obtained by HMAMP is the most similar to the human expert trajectory.*

efficient in hammering, since both the energy efficiency and vertical force ratio excel the other methods. In addition, HMAMP has the closest Frechet Distance to human manipulation trajectories, which reflects the effectiveness of our method in learning human motion styles. We can experience this more intuitively from Figure 5.

The experimental results clearly demonstrate the effectiveness of the proposed HMAMP framework in learning task-oriented, human-like manipulation skills through the use of AMP. The integration of AMP significantly improves task completion efficiency, knock impulse, and energy efficiency, resulting in superior performance compared to both the direct path-planning approach and traditional reinforcement learning (RL) methods.

5.2. Real robot arm experiment

We employed the same arm and the same task as the simulation for real-world experiments. The environment setup closely resembled real-world scenarios to ensure the applicability of our approach in practical scenarios.

Our proposed approach, which incorporates AMP into reinforcement learning (RL), was integrated into the control system of the robotic arm. The parameters and settings were optimized based on the training results obtained in the simulation environment. The robotic arm was tasked with performing the manipulation task using the learned skills.

Figure 6 showcases the manipulation effect of HMAMP on the real robot arm. The sequence of images illustrates the robotic arm successfully completing the manipulation task with precision and human-like

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

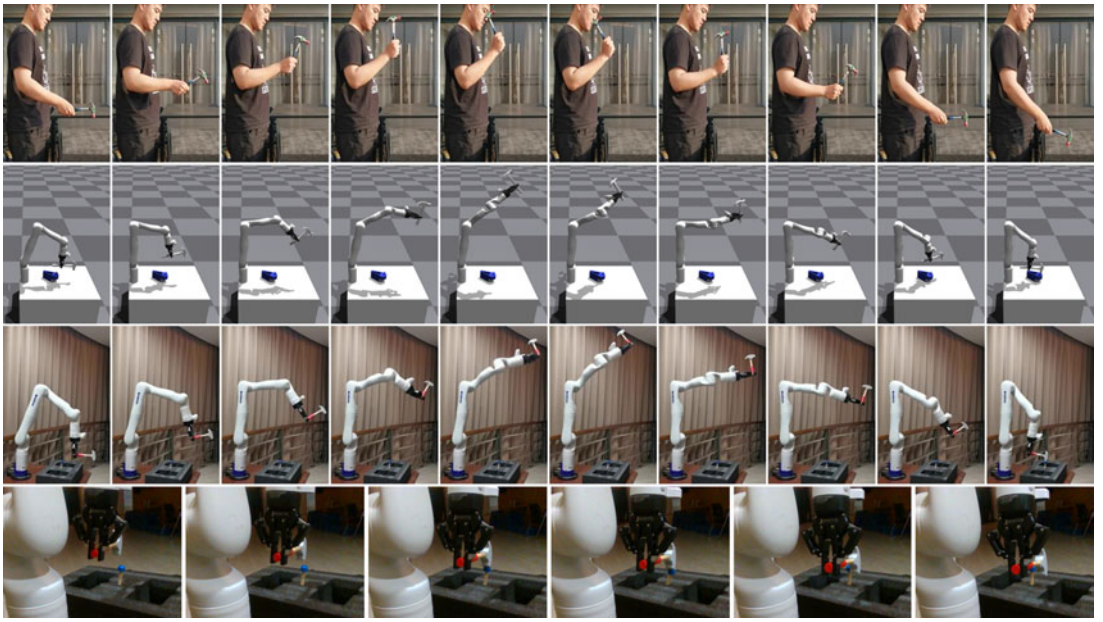


Figure 6. *Experiment in simulation and real world. The first row shows human knocking motion clips that we used as motion priors. The second row shows the policy HMAMP in simulation, the hammer can successfully complete the task with the manipulation trajectory that we desired. The third row shows the HMAMP implemented in real world on Kinova Gen3, and the fourth row is the details about hammering a nail in real world.*

motion. The trajectory followed by the arm demonstrates smoothness, accuracy, energy efficiency, and human manipulation skills, confirming the benefits of incorporating AMPs.

6. Conclusion

In this paper, we presented a novel approach named HMAMP to enable robotic arms to perform tool manipulation with human-like skills. Our method integrates AMP with deep reinforcement learning to capture complex manipulation dynamics. By leveraging both real-world motion data and synthetic motion data generated through simulation, we demonstrated the ability of our approach to surpass existing techniques in learning human-style manipulation behaviors. The evaluation of the challenging hammering task highlighted the effectiveness of our method and its potential for real-world applications. This research bridges the gap between robotic and human capabilities, paving the way for more intuitive and natural human-robot interactions. The proposed framework serves as a foundation for future research aimed at developing robots with advanced manipulation skills, envisioning a future where machines seamlessly mimic human manipulation.

Supplementary material. The supplementary material for this article can be found at http://dx.doi.org/10.1017/S0263574725001444.

Author contribution. Ziqi Ma, Changda Tian, and Yue Gao designed the study. Ziqi Ma and Changda Tian wrote the code. Ziqi Ma conducted the experiments and data gathering. Ziqi Ma and Changda Tian performed statistical analyses. Ziqi Ma and wrote the article.

Financial support. This work was supported by the National Natural Science Foundation of China (Grant No. 92248303 and No. 62373242) and the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102).

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

1. S. Ainetter and F. Fraundorfer, "End-to-End Trainable Deep Neural Network for Robotic Grasp Detection and Semantic Segmentation from RGB," *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021) pp. 13452–13458.
2. K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *Int. J. Robot. Res.* **39**(2-3), 202–216 (2020).
3. D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke and S. Levine, Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv preprint arXiv: 1806.10293 (2018).
4. Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei and S. Savarese, "Keto: Learning Keypoint Representations for Tool Manipulation," *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020) pp. 7278–7285.
5. L. Manuelli, W. Gao, P. Florence and R. Tedrake, "kpam: Keypoint Affordances for Category-Level Robotic Manipulation," *Robotics Research: The 19th International Symposium ISRR* (Springer, 2022) pp. 132–157.
6. D. Turpin, L. Wang, S. Tsogkas, S. Dickinson and A. Garg, Gift: Generalizable interaction-aware functional tool affordances without labels. arXiv preprint arXiv: 2106.14973 (2021).
7. M. Edmonds, F. Gao, H. Liu, X. Xie, S. Qi, B. Rothrock, Y. Zhu, Y. N. Wu, H. Lu and S.-C. Zhu, "A tale of two explanations: Enhancing human trust by explaining robot behavior," *Sci. Robot.* **4**(37), eaay4663 (2019).
8. H. Liu, C. Zhang, Y. Zhu, C. Jiang and S.-C. Zhu, "Mirroring Without Overimitation: Learning Functionally Equivalent Manipulation Actions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. **33** (2019) pp. 8025–8033.
9. Z. Zhang, Z. Jiao, W. Wang, Y. Zhu, S.-C. Zhu and H. Liu, "Understanding physical effects for effective tool-use," *IEEE Robot. Autom. Lett.* **7**(4), 9469–9476 (2022).
10. E. Johns, "Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration," *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021) pp. 4613–4619.
11. C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu and A. Anandkumar, Mimicplay: Long-horizon imitation learning by watching human play (2023).
12. K. Zorina, J. Carpentier, J. Sivic and V. Petrík, Learning to manipulate tools by aligning simulation to video demonstration (2021).
13. T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018) pp. 5628–5635.
14. X. B. Peng, Z. Ma, P. Abbeel, S. Levine and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graphics (TOG)* **40**(4), 1–20 (2021).
15. C. Sanz, J. Call and C. Boesch, *Tool Use in Animals: Cognition and Ecology* (Cambridge University Press, 2013).
16. R. St Amant and T. E. Horton, "Revisiting the definition of animal tool use," *Anim. Behav.* **75**(4), 1199–1208 (2008).
17. J. Van Lawick-Goodall, "Tool-Using in Primates and Other Vertebrates," **In:** *Advances in the Study of Behavior*, vol. **3** (Academic Press, 1971) pp. 195–249.
18. W. Chen, H. Liang, Z. Chen, F. Sun and J. Zhang, Learning 6-dof task-oriented grasp detection via implicit estimation and visual affordance (2022).
19. R. Xu, F.-J. Chu, C. Tang, W. Liu and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robot. Autom. Lett.* **6**(2), 2870–2877 (2021).
20. A. Murali, W. Liu, K. Marino, S. Chernova and A. Gupta, "Same object, different grasps: Data and semantic knowledge for task-oriented grasping. CoRR abs/2011.06431 (2020).
21. A. Al-Shanoon and H. Lang, "Robotic manipulation based on 3-d visual servoing and deep neural networks," *Robot. Auton. Syst.* **152**(C), 104041 (2022).
22. E. G. Ribeiro, R. de Queiroz Mendes and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robot. Auton. Syst.* **139**(C), 103757 (2021).
23. N. Saito, T. Ogata, S. Funabashi, H. Mori and S. Sugano, "How to select and use tools? : Active perception of target objects using multimodal deep learning. CoRR abs/2106.02445 (2021).
24. M. Sun and Y. Gao, "Gater: Learning grasp-action-target embeddings and relations for task-specific grasping," *IEEE Robot. Autom. Lett.* **7**(1), 618–625 (2022).
25. S. Nair, A. Rajeswaran, V. Kumar, C. Finn and A. Gupta, "R3m: A Universal Visual Representation for Robot Manipulation," *6th Annual Conference on Robot Learning* (2022).
26. H. Xiong, H. Fu, J. Zhang, C. Bao, Q. Zhang, Y. Huang, W. Xu, A. Garg and C. Lu, "Robotube: Learning Household Manipulation from Human Videos with Simulated Twin Environments," *6th Annual Conference on Robot Learning* (2022).
27. O. Taheri, N. Ghorbani, M. J. Black and D. Tzionas, GRAB: A dataset of whole-body human grasping of objects. CoRR abs/2008.11200 (2020).
28. H. Xiong, Q. Li, Y. Chen, H. Bharadhwaj, S. Sinha and A. Garg, Learning by watching: Physical imitation of manipulation skills from human videos. CoRR abs/2101.07241 (2021).

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press

29. J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," *Advances in Neural Information Processing Systems 29* (2016).
30. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," *Commun. ACM* **63**(11), 139–144 (2020).
31. A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg and P. Abbeel, "Adversarial Motion Priors make Good Substitutes for Complex Reward Functions," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022) pp. 25–32.
32. X. Mao, Q. Li, H. Xie, R. Y. K. Lau and Z. Wang, Multi-class generative adversarial networks with the L2 loss function. CoRR abs/1611.04076 (2016).
33. X. Mao, Q. Li, H. Xie, R. Lau, W. Zhen and S. Smolley, "On the effectiveness of least squares generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(12), 2947–2960 (2018).
34. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang and M. Grundmann, Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv: 2006.10204 (2020).
35. T. Geng, M. Lee and M. Hülse, "Transferring human grasping synergies to a robot," *Mechatronics* **21**(1), 272–284 (2011).
36. G. Gioioso, G. Salvietti, M. Malvezzi and D. Prattichizzo, "Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain," *IEEE Trans. Robot.* **29**(4), 825–837 (2013).
37. R. Suárez, J. Rosell and N. García, "Using Synergies in Dual-Arm Manipulation Tasks," *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 5655–5661.
38. N. Physics Simulation Environment for Reinforcement Learning Research, Isaac gym - preview release (2023). https://developer.nvidia.com/isaac-gym.
39. B. Aronov, S. Har-Peled, C. Knauer, Y. Wang and C. Wenk, "Fréchet Distance for Curves, Revisited," *Algorithms–ESA 2006: 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006. Proceedings 14* (Springer, 2006) pp. 52–63.

Cite this article: Z. Ma, C. Tian and Y. Gao (2025). "Manipulate as human: learning task-oriented manipulation skills by adversarial motion priors", Robotica **43**, 2320–2332. https://doi.org/10.1017/S0263574725001444

https://doi.org/10.1017/S0263574725001444 Published online by Cambridge University Press