

Few-shot transfer of tool-use skills using human demonstrations with proximity and tactile sensing

Marina Y. Aoyama¹, Sethu Vijayakumar², and Tetsuya Narita³

Abstract—Tools extend the manipulation abilities of robots, much like they do for humans. Despite human expertise in tool manipulation, teaching robots these skills faces challenges. The complexity arises from the interplay of two simultaneous points of contact: one between the robot and the tool, and another between the tool and the environment. Tactile and proximity sensors play a crucial role in identifying these complex contacts. However, learning tool manipulation using these sensors remains challenging due to limited real-world data and the large sim-to-real gap. To address this, we propose a few-shot tool-use skill transfer framework using multimodal sensing. The framework involves pre-training the base policy to capture contact states common in tool-use skills in simulation and fine-tuning it with human demonstrations collected in the real-world target domain to bridge the domain gap. We validate that this framework enables teaching surface-following tasks using tools with diverse physical and geometric properties with a small number of demonstrations on the Franka Emika robot arm. Our analysis suggests that the robot acquires new tool-use skills by transferring the ability to recognise tool-environment contact relationships from pre-trained to fine-tuned policies. Additionally, combining proximity and tactile sensors enhances the identification of contact states and environmental geometry. See our videos at <https://sony.github.io/tool-use-few-shot-transfer/>.

I. INTRODUCTION

TOOLS extend the ability of robots to interact with the environment and manipulate objects [1]. For instance, mastering tool-use skills is crucial to perform complex tasks such as cutting vegetables with a knife, cleaning surfaces with a sponge, or picking up small screws with tweezers. This study focuses on manipulating grasped tools in a similar way to humans, rather than attaching tools to the end-effector as considered in most prior work [2]–[6]. The ability to manipulate grasped tools enables robots to use tools designed for humans and seamlessly swap tools for different tasks.

One of the challenges in acquiring tool-use skills is the complex contact relationships among the robot, the tool, and the environment. When a robot manipulates grasped tools, there are two locations of contact: the grasped point and the tool tip, as illustrated in Fig. 1. Intrinsic contact sensing

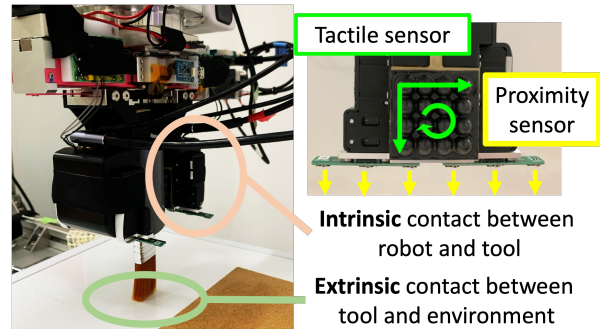


Fig. 1: Tool manipulation using tactile and proximity sensors, involving two points of contact.

involves direct contact between the tactile sensors and the tool. Extrinsic contact sensing [7] involves indirect sensing of the contact between the tool and the environment, with the tactile sensors being only in contact with the tool. The interplay between these two points of contact introduces complexity in sensing. Tactile sensing has proven useful in contact-rich tasks [8], [9], including tool-use scenarios [3], [10], [11], while proximity sensing provides information about local geometry and the pose of the robot’s fingertips relative to the environment [12], [13]. Despite these capabilities, there are currently no sensors capable of directly measuring extrinsic contact. Thus, the challenge lies in utilising multiple indirect sensor data sources to accurately identify these interactions across various tools and environments. To address this, we aim to combine tactile and proximity sensing to implicitly capture both intrinsic and extrinsic contacts, without requiring explicit disentanglement.

To identify intrinsic and extrinsic contacts and manipulate deformable tools where detailed models or simulations of complex environments are unavailable, Learning from Demonstration (LfD) provides a viable approach for robots to acquire skills or behaviours by imitating human demonstrations[14]. However, the number of demonstrations is often limited due to the significant time and effort required from humans. It is, therefore, impractical to learn tool-use skills solely from demonstrations, given the wide-ranging variations in the physical and geometric properties of tools and environments. Alternatively, while learning in simulation offers the advantage of collecting large dataset at a low cost of time and human effort, it suffers from the sim-to-real gap.

To enable robots to acquire various tool-use skills across different tools and task objectives, we propose a multimodal few-shot tool-use skill transfer framework. This approach involves two main phases: first, pre-training the base policy with primitive motions common in tool-use tasks in simu-

Manuscript received: January, 21, 2025; Revised April, 21, 2025; Accepted June, 6, 2025.

This paper was recommended for publication by Editor Ashis Banerjee upon evaluation of the Associate Editor and Reviewers’ comments.

¹School of Informatics, The University of Edinburgh, U.K. m.aoyama@sms.ed.ac.uk

²School of Informatics, The University of Edinburgh, U.K. sethu.vijayakumar@ed.ac.uk

³Sony Group Corporation, Japan. Tetsuya.A.Narita@sony.com

The first author conducted this research during an internship at Sony Group Corporation, Japan. The second author is partially funded by the Moonshot R&D Program (Grant No. JPMJMS2031). We gratefully acknowledge Takahisa Ueno for his support with the setup of simulation and hardware.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

lated environments; second, fine-tuning the policy with human demonstrations collected in the target domain. The key idea is to leverage inexpensive simulated data, which may differ from the target tool properties, environments, or tasks, while bridging this gap between the simulated pre-trained domain and the real target domain with human demonstrations. While the proposed framework is a general multimodal approach, we focus on the combined use of tactile and proximity sensors for tool-use skill acquisition.

We validate the proposed few-shot tool-use skill transfer framework on a surface-following task using tools with varying physical and geometric properties, such as brushes and sponges. This task serves as an exemplar of contact-rich tool manipulation tasks, where desired contact and motion depend on tool and task properties, and success requires fine control of contact. Our results demonstrate that fine-tuning the policy pre-trained in simulation with human demonstrations enables the robot to manipulate new tools, adapt to different environments, and perform new tasks, outperforming baseline approaches without pre-training or fine-tuning. Moreover, our analysis suggests the critical role of both proximity and tactile sensors in identifying contact force and the environment's geometry, particularly in the presence of sensor noise and the significant sim-to-real gap. Furthermore, a Principal Component Analysis (PCA) of the model's latent space supports our hypothesis that the pre-trained policy transfers its ability to recognise tool-environment contact relationships across various tools.

In summary, the contributions of this work are:

- We tackle the challenging problem of learning tool-use skills that require delicate contact control, involving both intrinsic contact between the robot's fingertips and the tool and extrinsic contact between the tool and the environment.
- We present a data-efficient framework that utilises proximity and tactile sensing for pre-training a base policy in simulation and fine-tuning it in the real world to adapt to tools with diverse geometric and physical properties.
- We demonstrate that the proposed few-shot tool-use skill transfer framework achieves motion and contact forces close to those demonstrated by a human, successfully performing tool-use tasks on a physical robotic setup without fixing the tool to the end-effector.

II. RELATED WORK

A. Tool manipulation

Tool-use skills are essential in many robot manipulation applications such as wiping [2], transferring [3], scraping [5], [15], and skewering [4]. However, most existing work simplifies these tasks by fixing the tool to the robot's end-effector, thereby avoiding the complex interactions between intrinsic and extrinsic contacts [2]–[5]. For robots to seamlessly swap and use various tools, they must be capable of grasping and manipulating tools.

Several recent works successfully estimate and control extrinsic contacts between a tool [10] or object [8], [11] and the environment. However, these approaches [8], [10], [11] are

limited to the manipulation of rigid objects making contact with a flat surface due to the need to model the mechanics of contact. In contrast, our approach builds on research that learns the contact states between the tool and the environment from interaction data [15], [16], enabling us to manipulate deformable tools and contacts with non-flat surfaces that are difficult to model. However, these prior works [15], [16] are limited to the manipulation of the tools used during training. No prior approaches have demonstrated the ability to perform tool manipulation with unattached, deformable tools across multiple tool types.

B. Learning from Demonstration

LfD has been successfully used for acquiring tool manipulation skills [2]–[6]. However, due to the time and human effort required for collecting demonstrations, the available demonstration data tends to be limited in both quantity and tool variation. One promising approach to enhance the robustness and generalisability of learnt skills, especially with limited demonstration data, is pre-training perceptual models [2], [9] or state prediction models [5], [17]. However, pre-training using real-world play data collected through human teleoperation [9] or unsupervised interactions based on random and heuristic motions [5], [17] demands extensive human effort or raises the possibility of unsafe contacts. On the other hand, pre-training in simulation suffers from the sim-to-real gap, which refers to the discrepancy between simulated and real-world environments [2]. While there is a framework enabling zero-shot sim-to-real transfer [18], it only predicts contact patches and does not allow for precise control of contact force between tools and the environment.

In our approach, we leverage learning the interaction model from inexpensive data collected safely in various simulated environments, with the capability to fine-tune this model using demonstration data from the real target domain. While there exists a framework for composing tool-use skills in multi-task and multi-domain learning contexts [19], it struggles in tasks requiring delicate contact control due to the domain gap between simulation or video and reality.

C. Tactile and proximity sensing

Tactile sensors determine contact forces, contact events, local geometries, and material properties when they are in contact with the object or the environment [20], while proximity sensors identify local geometries and positions relative to the environment without contact [12]. Most existing methods use proximity sensors for the pre-contact phase [13] and tactile sensors for the post-contact phase [21]. In contrast, we propose a configuration of proximity sensors that enables environmental observation while grasping and manipulating a tool, complemented by tactile sensors at the fingertips. We hypothesise that this configuration provides additional information about the geometry (*e.g.*, shape and distance) of the environment around the tool-environment contact area. While multimodal sensing, such as vision and tactile sensing [17], has shown improved manipulation performance, the simultaneous use of tactile and proximity sensing for tool manipulation involving intrinsic and extrinsic contacts remains underexplored.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

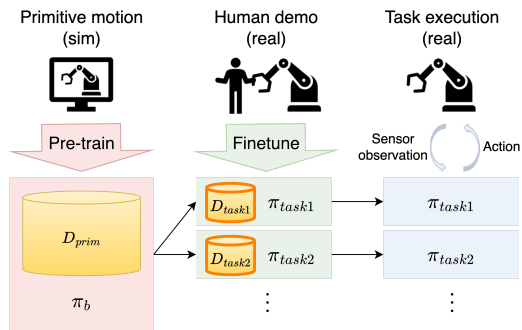


Fig. 2: Few-shot tool-use skill transfer framework.

III. PROBLEM FORMULATION

We address the problem of learning tool-use skills involving two locations of contact: one between the robot and the tool, and the other between the tool and the environment, as illustrated in Fig. 1. We consider a scenario where the physical and geometric properties of the tool and surface are unknown a priori. Therefore, the robot needs to identify them from multimodal sensory information on the fly and adapt the motions accordingly. Additionally, as the desirable contact force between the tool and the environment depends on the task, the robot needs to learn this force from demonstrations of the target task. More specifically, the goal is to enable robots to learn a policy π to predict the next desired actions $\hat{u}_{t:t+T_n}$ for the next T_n time steps, similar to a human demonstrator, given tactile sensor observations $x_{t-T_p:t-1}^{tac}$ and proximity sensor observations $x_{t-T_p:t-1}^{prox}$ from the past T_p time steps. In this study, we focus on surface-following tasks using grasped tools. These tasks require fine control of the contact force between the tool and the environment, considering the properties of the tool and the geometries of the surfaces. Examples include painting with a brush and sweeping with a broom.

IV. METHOD

To enable the learning of new tool-use skills from a limited number of demonstrations, we propose a few-shot tool-use transfer framework. This framework involves pre-training the base policy using primitive motions in simulation and then fine-tuning it with human demonstrations of downstream tasks using the target tool in the real world to bridge the gap.

A. Pre-training base policy

In the pre-training phase, we train the base policy π_b by leveraging a large amount of data $D_{prim} = \{\tau_1, \dots, \tau_N\}$, where $\tau = \{(x_0, u_0), (x_1, u_1), \dots, (x_T, u_T)\}$ collected through primitive motions in simulation. Here, N denotes the number of trajectories, and T represents the number of timesteps in each trajectory. By performing this set of primitive motions repeatedly in simple simulated environments with varying conditions, we expect the base policy π_b to recognise contact states between the tool and the environment from multi-modal sensor observations and to capture motions common in tool-use skills, in a manner transferable across different tasks and tools. For instance, in surface-following tasks, we hypothesise that contact states such as contact angle and tooltip contact with a wall, and motions such as tracking a desired contact

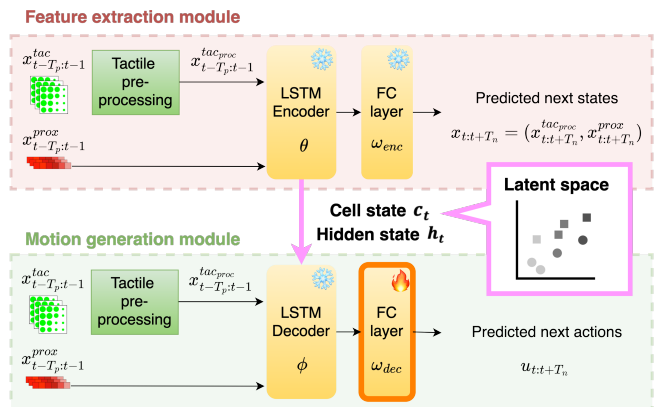


Fig. 3: Seq2Seq architecture of the tool-use skill policy. The feature extraction module captures the contact relationship between the robot, tool, and environment by predicting next states. The motion generation module predicts the next desired actions to perform the task.

force or lifting up when the tooltip collides with a wall, are common to all surface-following tasks regardless of the tool properties and task goals. It is important to highlight that the proposed framework requires primitive data collection and pre-training only once, as illustrated in Fig. 2. Every new task with a new tool, environment, or task goal shares this single base policy. For the surface-following task, we use pre-defined primitive motions to collect data in simulation, as explained in Section V-C1. However, we emphasise that our framework can utilise other low-cost data sources, such as interaction data gathered through teleoperation or autonomous exploration.

For this, we adopt the Sequence-to-Sequence (Seq2Seq) model [22], a widely used time-series prediction model in natural language processing [23] and robot control [24]. The model employs an encoder-decoder structure using Long Short-Term Memory (LSTM), as illustrated in Fig. 3. The encoder serves as a feature extraction module. This module encodes multi-modal sensor observations x from the past T_p timesteps into latent representations consisting of cell and hidden states by predicting the next T_n states $\hat{x}_{t:t+T_n}$. In our case, with tactile and proximity sensors, the sensor observations x include $x_{t-T_p:t-1}^{tac}$ and $x_{t-T_p:t-1}^{prox}$. The decoder functions as the motion generation module, predicting the desired actions $u_{t:t+T_n}^{prim}$ in specified in primitive motions for the next T_n timesteps. We condition this motion generation on the encoded contact states, which are passed from the encoder's cell and hidden states to the decoder. The encoder-decoder structure offers key advantages for our framework. The encoder extracts latent representations of contact dynamics, decoupled from motion generation, enabling the reuse of the feature extraction module across tasks. Meanwhile, we can fine-tune the decoder for task-specific motion generation, allowing data-efficient adaptation to new tasks while retaining the generalisable features from the encoder. To achieve this, we train the model on primitive data by minimising the pre-training loss

$$\begin{aligned} \hat{\theta}, \hat{\phi}, \hat{\omega}_{enc}, \hat{\omega}_{dec} &= \arg \min_{\theta, \phi, \omega_{enc}, \omega_{dec}} L(x_{t:t+T_n}, u_{t:t+T_n}) \\ &= E_{MSE}(\hat{x}_{t:t+T_n}, x_{t:t+T_n}^{obs}) + \beta E_{MSE}(\hat{u}_{t:t+T_n}, u_{t:t+T_n}^{prim}) \end{aligned} \quad (1)$$

where θ and ω_{enc} denote the parameters of the LSTM layer and fully connected layer in the encoder, respectively, and ϕ and ω_{dec} represent the parameters of the LSTM layer and fully connected layer in the decoder of the seq2seq model, respectively, as shown in Fig. 3. The first term of the loss function is the state prediction loss, which computes the Mean Squared Error E_{MSE} between the observed states $x_{t:t+T_n}^{obs}$ and the predicted states $\hat{x}_{t:t+T_n}$ for the next T_n time steps to train the feature extraction module. The second term of the loss function is the action prediction loss, which computes the Mean Squared Error E_{MSE} between the actions $u_{t:t+T_n}^{prim}$ observed during the execution of primitive motions and the predicted actions $\hat{u}_{t:t+T_n}$ for the next T_n time steps to train the motion generation module. β is the weight coefficient that balances the two loss terms. Although we chose Seq2Seq for its data efficiency and fast execution, our framework potentially extend to other encoder-decoder models, such as transformer-based approaches [25], [26].

B. Few-shot fine-tuning using demonstration

Subsequently, we fine-tune the pre-trained base policy π_b to obtain the fine-tuned policy π_{task} using a small amount of demonstration data D_{task} specific to each new downstream task. We acknowledge the differences in tool properties, environments, percepts, and task goals between the pre-trained domain and the target downstream task domain. We expect that fine-tuning using demonstration data collected in the target domain (*i.e.*, using the target tool to achieve the downstream task in the target environment) bridges this domain gap and adapt the policy to manipulate new tools, adapt to new environments, adapt from simulated to real environments, and achieve new task goals.

To prevent over-fitting when fine-tuning the policy with a very small amount of demonstration data in the target domain, we minimize the number of layers to fine-tune. Specifically, we finetune only the fully connected (FC) layer of the decoder while freezing the weights of all other layers in both the encoder and decoder. Therefore, while the primary role of the decoder during pre-training is motion generation, we expect the updated weights of ω_{dec} to adapt not only to differences in task goals and motions but also to domain differences, such as tool properties and other gaps between the pre-trained and target task domains. We achieve this adaptation by incorporating both state prediction and action prediction losses, as shown in LfD loss

$$\begin{aligned} \omega_{dec}^{\hat{}} &= \arg \min_{\omega_{dec}} L(x_{t:t+T_n}, u_{t:t+T_n}) \\ &= E_{MSE}(\hat{x}_{t:t+T_n}, x_{t:t+T_n}^*) + \beta E_{MSE}(\hat{u}_{t:t+T_n}, u_{t:t+T_n}^*). \end{aligned} \quad (2)$$

C. Task execution

At test time, the robot predicts the next desired actions $\hat{u}_{t:t+T_n}$ based on past sensor observations $x_{t-T_p:t-1}$ using the fine-tuned policy π_{task} to perform the downstream task. At each timestep, the robot predicts actions for the next T_n time steps to facilitate smoother trajectories and capture the temporal dependencies of its actions, while executing only the action predicted for the immediate next timestep t .

V. EXPERIMENTAL SETUP

A. Tool-use tasks

We conducted our experimental evaluation on surface-following tasks using tools with varying geometric (*e.g.*, tool length, contact area) and physical (*e.g.*, stiffness, friction) properties. In simulation, we designed a brush with a 3cm rigid handle and a soft tooltip (2×3×3 capsules, 2.5cm spacing, stiffness 30 using MuJoCo’s composite API) for pre-training. We then validated our proposed framework in real environments using four different tools: a small brush, a wide brush, a sponge, and a broom on a physical setup as depicted in Fig. 4. Each tool had a soft material around the handle to ensure a stable grasp. For the surfaces, we prepared two basic environments: “Inclined,” which is a flat surface inclined at $\psi = [-0.3, 0.3]$ radians, and “Step” which has a step with a height of $h = [0.5, 5.0]$ cm in simulation. We tested in the ‘Inclined’ and ‘Step’ environments in the real world. Additionally, we tested more complex environments, including ‘Stairs’ with multiple consecutive steps in simulation and ‘Deforming Surface’ in the real world.

B. System design and robot setup

Fig. 4 depicts the experimental setup and sensor configurations. We utilise a 7 Degrees of Freedom (DoF) Franka Emika Panda robotic arm with a position controlled 2-finger parallel gripper attached to its end-effector for both simulation and real robot experiments. For the simulation setup, we utilise the Mujoco physics engine [27], [28], configured to replicate the physical robot setup. At the start of each trial, the robot grasps the tool handle near its centre, with a position shift sampled from $\mathcal{N}(0, 1)$ cm, by closing the gripper until the normal force exceeds 5N, and then maintains a fixed gripper width throughout the trial. The base and fine-tuned policies output end-effector velocities in the x and z directions, as shown in Eq. (3), which we convert to end-effector positions and then map to joint configurations using inverse kinematics to control the robot. We provide demonstrations by kinaesthetically moving the robot arm in gravity compensation mode.

Inside each fingertip, we attach tactile sensors [29] with 4 by 4 elastic hemispheres placed on 12 by 12 pressure sensor nodes. The tactile sensors estimate the 3-axis force at each hemisphere by measuring its distortion. In simulation, we use 3-axis force sensors arranged in a corresponding 4 by 4 configuration as an approximate match. For both simulated and real tactile observations, we pre-process 3-axis forces from each fingertip, denoted as x^{tac} , into shear forces $x^{shear} \in \mathbb{R}^2$ and translational and rotational slip $x^{slip} \in \mathbb{R}^3$, following the theory of translational and rotational slip [29]. This tactile representation, $x^{tac_{proc}} = (x^{shear}, x^{slip})$, provides a compact and meaningful low-dimensional tactile feature that reduces the sim-to-real gap and enhances robustness against variations in raw sensor values caused by slight changes in grasp location or sensor noise. Additionally, we place six Time Of Flight (ToF) proximity sensors on the right fingertip, oriented to look down at the environment. Each sensor measures the distance between itself and the closest object as a scalar value.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

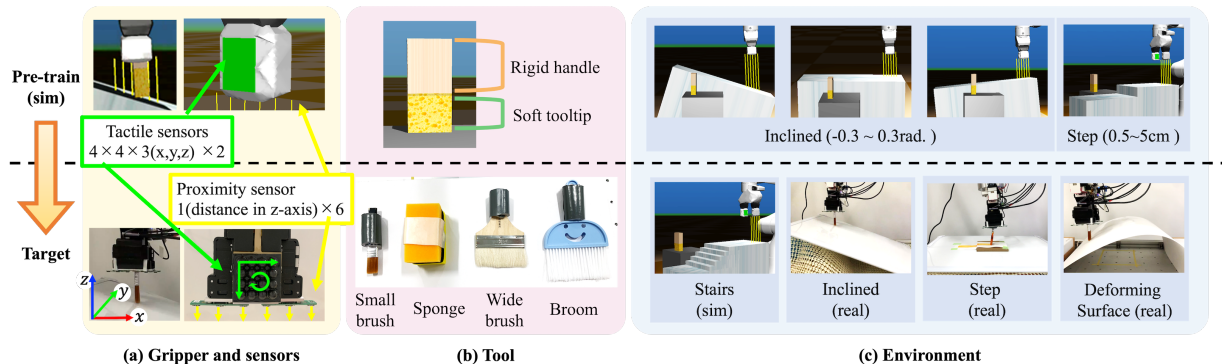


Fig. 4: Experimental setup.

C. Data collection

1) Primitive data collection

First, we collect primitive data D_{prim} for the surface-following task in two basic environments, "Inclined" and "Step", in simulation. For the surface-following task, we define primitive motions along the x -axis as $u_x = u_x^{const}$, and along the z -axis as

$$u_z = \begin{cases} u_z^{up} & \text{if } F_x > F_x^{threshold} \text{ (hitting wall)} \\ u_z^{down} & \text{elif } F_z < 0.0 \text{ (out of contact)} \\ \alpha(F_z^{target} - F_z) & \text{otherwise (in contact)} \end{cases} \quad (3)$$

where we define the axis coordinates in the task frame, as illustrated in Fig. 4. The robot moves its end-effector at a constant velocity u_x^{const} in the horizontal direction. When it detects a collision with a wall, it moves the end-effector up by u_z^{up} ; if out of contact, it moves the end-effector down by u_z^{down} to reestablish contact. Otherwise, it adjusts the velocity of the end-effector in the normal direction u_z to achieve the specified target force F_z^{target} , without tool slippage, given the current normal force F_z and the scaling factor α . In our experiment, we set parameters to $u_x^{const} = 0.3$ cm/s, $u_z^{up} = 0.5$ cm/s, $u_z^{down} = -0.5$ cm/s, $F_x^{threshold} = 0.5$ N, $F_z^{target} = 0.3$ N, and $\alpha = 0.1$. We collected 11 trajectories in the "Inclined" environment and 10 trajectories in the "Step" environment. For each trial, we record the end-effector positions, tactile data ($4 \times 4 \times 3$ axes for 2 fingertips), and proximity data (6 dim.) at a frequency of 20 Hz over a duration of 10 seconds.

2) Demonstration data collection

We collect demonstration data D_{task} to fine-tune the base policy π_b for learning the target task motion. For the simulated experiments, we collect 3 demonstration trajectories each for (1) the new target force where $F_z^{target} = 0.5$ N and (2) the stairs environment. For the physical robot experiments, we collect 3 demonstration trajectories for each downstream task, using different tools. We record the end-effector positions, tactile data ($4 \times 4 \times 3$ axes for 2 fingertips), and proximity data (6 dim.) at a frequency of 20 Hz for 10 seconds.

D. Model training

The seq2seq model consists of 1 LSTM layer and 1 fully connected layer in the encoder, and 1 LSTM layer and 1 fully connected layer in the decoder. We apply dropout [30] to the fully connected layers following the LSTM layers in both the encoder and decoder to prevent over-fitting during pre-training,

with a dropout rate of 0.2. We set the dimensions of the cell state and hidden state to $c \in \mathbb{R}^{100}$ and $h \in \mathbb{R}^{100}$, respectively. We employ stochastic gradient descent [31] to update the weights of the neural network. We normalise all end-effector position data, shear force and slip data, and proximity data to the range $[-0.9, 0.9]$.

First, we pre-train the seq2seq model using primitive data as detailed in Section IV-A. We train the model for 2000 epochs at a learning rate of 0.001, utilising a loss function defined in Eq. (1) with $\beta = 0.1$. It is important to note that the collection of primitive data and the pre-training of the base policy are only required once. This single pre-trained model serves as the base policy for all downstream tasks. After pre-training the base policy, we freeze the weights of all layers in the encoder and the LSTM layer in the decoder. For each task, we fine-tune the fully connected layer in the decoder on demonstration data for 300 epochs at a learning rate of 0.005 using the loss function Eq. (2).

VI. RESULTS AND DISCUSSION

We evaluate the ability of the proposed few-shot tool-use skill transfer framework to adapt the robot's motion for new tools and target tasks with only a few demonstrations.

We compared our proposed approach, denoted as 'Fine-tuned,' with a behaviour cloning baseline, "Demo only" [32], which uses a policy trained solely on demonstration data without any pre-training in simulation, and an ablated baseline, "No FT," which uses the pre-trained policy without fine-tuning in the target domain. We also included two alternative methods used in settings with limited supervised data: Gaussian Processes (GP) with LSTM [33], to incorporate temporal information for fair comparison, trained on demonstration data without pre-training; and Dataset Aggregation (DAGger) [34], where we trained on pre-training data in the first iteration, and on aggregated data—including additional fine-tuning demonstrations as corrections—in the second. In all approaches, we collected three demonstrations and empirically tuned hyperparameters to minimise the training loss.

For the "Inclined" environment in Fig. 4, we computed: 1) the Root Mean Squared Error (RMSE) between the ground truth and the followed inclination, and 2) the RMSE between the desired contact force and the applied force to evaluate how effectively the tool followed the surface geometry while applying the desired force. We collected demonstration data

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Env.	Evaluation Criteria	Demo only	No FT	FT (diff. tool)	GP	Dagger	Finetuned
Inclined	RMSE of slope (rad) ↓	0.073 ± 0.002	0.041 ± 0.006	0.038 ± 0.003	0.095 ± 0.007	0.072 ± 0.005	0.040 ± 0.005
	RMSE of force (N) ↓	0.68 ± 0.06	0.22 ± 0.01	0.17 ± 0.02	0.61 ± 0.05	0.62 ± 0.07	0.11 ± 0.02
Stairs	Wiped area (%) ↑	45.0 ± 3.50	49.7 ± 2.31	53.1 ± 2.98	43.3 ± 4.01	51.1 ± 3.80	59.2 ± 4.31

TABLE I: Comparison of RMSE when transferring the pre-trained tool-use skill to a new task (*i.e.*, with a new target force) and to a more complex environment (*i.e.*, stairs).

for test cases in addition to the training data to obtain the desired contact force for evaluation. For surfaces with steps (*i.e.*, "Step" and "Stairs" in Fig. 4), the RMSE metrics were unsuitable due to intentional tool lifting when hitting a step. Thus, we used wiped area (%)—the contact area relative to the total surface area—as the evaluation metric. Since demonstrations aim to maximise the wiped area by quickly reestablishing contact after lifting the tool, this metric reflects both imitation accuracy and task completeness.

A. Does the tool-use skill policy, fine-tuned with demonstrations, improve the performance of downstream tasks?

First, we evaluated our approach in two test cases using the simulated environments "Inclined" and "Stairs". For the inclined surface, the task involved wiping an inclined surface with a desired force of 0.5N, which differed from the 0.3N used during pre-training. In the "Stairs" environment, the task was to wipe a surface with stairs instead of a single step seen in the pre-training phase.

In both environments, "Finetuned" achieved the highest task performance compared to the baselines ("Demo only" and "No FT") as well as the alternative methods "GP" and "Dagger," as depicted in TABLE I. Fig. 5a illustrates that "Demo only" failed to achieve the desired force due to insufficient demonstrations. In contrast, "Finetuned" successfully achieved the new desired contact force, different from the desired force used in pre-training. Fig. 5b shows the end-effector position in the z-direction as well as the contact force in the normal direction while wiping stairs using the fine-tuned policy. The red shading indicates the contact of the tool tip with the wall, and the blue indicates the flat region (*i.e.*, no contact of the tool tip with the wall). We observed that the fine-tuned policy detects the contact of the tool tip with the wall, leading to successful lifting of the tool, and returns to the desired contact force (*i.e.*, 0.3N in this case) when reaching the flat region. We also evaluated the generalisability of the policy fine-tuned on a different tool instance with a longer handle (5cm vs. 3cm) and softer tip (stiffness 10 vs. 30) (FT (diff. tool) in TABLE I); while this policy outperformed the baselines, the one fine-tuned on the target tool achieved the lowest error, highlighting the importance of task-specific fine-tuning.

Second, we evaluated our approach in two real-world environments: "Inclined" and "Step", using four different tools. In both environments, the robot needs to adapt its motions to real tools with various properties that are inherently different from those used in simulation, while also accounting for the sim2real gap in sensing. Additionally, the robot needs to achieve a new target contact force required for each task, as

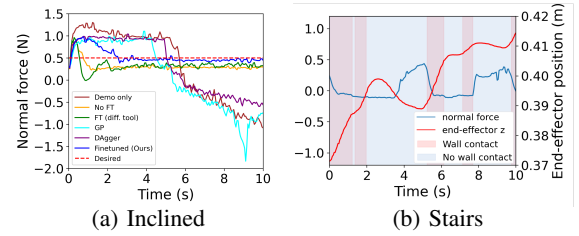


Fig. 5: End-effector motion and contact force profile.

Tool	Demo only		No FT		Fintuned	
	x	σ	x	σ	x	σ
Small brush	0.41	0.16	0.25	0.07	0.17	0.02
Sponge	0.30	0.12	0.29	0.01	0.17	0.03
Wide brush	0.18	0.04	0.37	0.12	0.10	0.02
Broom	0.84	0.33	0.66	0.37	0.12	0.05

(a) RMSE of the inclination followed by the end-effector (↓).

Tool	Demo only		No FT		Fintuned	
	x	σ	x	σ	x	σ
Small brush	0.041	0.006	0.031	0.011	0.014	0.002
Sponge	0.065	0.034	0.051	0.015	0.017	0.004
Wide brush	0.017	0.008	0.038	0.007	0.009	0.004
Broom	0.034	0.022	0.057	0.047	0.020	0.009

(b) RMSE of the normal contact force (↓).

Env.	Demo only		No FT		Fintuned	
	x	σ	x	σ	x	σ
Step	72.74	16.45	87.07	5.61	93.54	4.01

(c) Wiped area (↑).

TABLE II: Comparison of task performance using different tools on "Inclined" and "Steps" surfaces on a physical setup.

demonstrated by a human.

TABLE IIa and TABLE IIb display the RMSE of the inclination and the contact force, respectively, for four downstream tasks demonstrated by a human using different tools shown in Fig. 4. In all cases, "Finetuned" achieved a significant reduction in RMSE compared to the baselines. In real-world task performance, in the "small brush" painting task, "Finetuned" painted $93.5 \pm 1.68\%$ of the surface, outperforming "No FT" ($87.1 \pm 1.61\%$) and "Demo Only" ($72.7 \pm 4.75\%$). In the "broom" task, it swept 8.0 ± 1.0 dust pieces out of 10, compared to 2.7 ± 1.53 for "No FT" and 6.3 ± 0.58 for "Demo Only". Furthermore, for the task of painting a surface with a step, "Finetuned" notably increased the wiped area, as evidenced by the results in TABLE IIc.

These results, where the proposed approach outperforms the "Demo only" and "No FT" baselines, demonstrate the ability of our few-shot tool-use skill transfer framework to manipulate various tools by: (1) leveraging the base policy pre-trained in simulation, and (2) addressing the sim-to-real gap and differences in tool properties with human demonstrations collected in the target domain.

B. Does the pre-trained feature extraction module capture transferable contact states across tools?

Fig. 6 illustrates a PCA applied to a snapshot of the latent space with cell states $\mathbf{c}_t \in \mathbb{R}^{100}$, where $t = 5s$, when following a surface with unseen inclinations of $-0.25, 0.0$, and 0.25 radians using the simulated brush used for pre-training

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

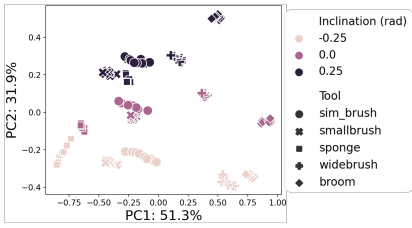


Fig. 6: PCA applied to a snapshot of the latent space while following a surface with unseen inclinations.

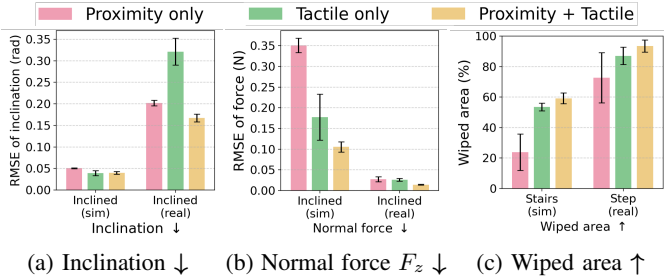


Fig. 7: Contribution of each modality.

and four tools used in the real-world environment. While the first principal component (PC1) delineates the different tools as indicated by the shape of the markers, the second principal component (PC2) demonstrates the separation according to the inclination of the surface, as indicated by the colour of the markers. The self-organisation of the latent space based on surface inclination, even across different tools, validates our hypothesis that the pre-trained model captures the contact dynamics between the end-effector, tools, and the environment in a transferable manner across different tools. This transferability enables successful execution of downstream tasks with minimal demonstrations when adapting to new tools. Despite pre-training with a single tool, the latent space captures both environments and tools, which we hypothesise emerged from variations in grasp location and contact forces experienced during pre-training.

C. How do proximity and tactile sensing contribute to tool manipulation?

We investigated the contribution of each sensor modality to surface-following tasks in both simulated and real-world environments, as summarised in Fig. 7.

In simulation, all modalities perform similarly in following the desired inclination. However, proximity sensing alone struggles to achieve the desired force on inclined surfaces and to detect tool-tip contact on stairs, where tactile information is essential. Additionally, the sensor configurations capturing only intrinsic contact, as shown in Fig. 8(a)(1)-(3) performed poorly across tasks, as indicated in Fig. 8(b)-(d). Combining an array of proximity sensors and shear force from tactile sensors which allows to capture both intrinsic and extrinsic contact (Fig. 8(a)(4)) resulted in the lowest RMSE of contact force on inclined surfaces and the largest wiped area on stairs. These results underscore the importance of our sensor configurations and learning framework combining both sensing to implicitly capture both intrinsic and extrinsic contact for achieving the desired tool-environment contact.

In real-world environments, combining proximity and tac-

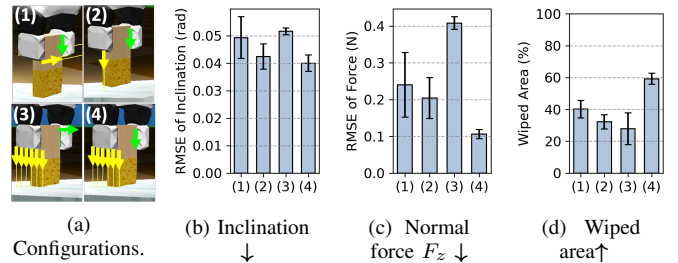


Fig. 8: Comparison of different sensor configurations. : (1) proximity sensors inside the finger, (2) a single proximity sensor, (3) normal force only, (4) proposed shear and slip force with an array of proximity sensors.

Evaluation Metric	Simulation			Physical setup		
	Demo only	No FT	Finetuned	Demo only	No FT	Finetuned
RMSE of slope (rad) ↓	0.082 ± 0.007	0.052 ± 0.005	0.042 ± 0.006	0.76 ± 0.21	0.30 ± 0.10	0.22 ± 0.05
RMSE of force (N) ↓	0.72 ± 0.05	0.29 ± 0.05	0.14 ± 0.03	0.052 ± 0.006	0.035 ± 0.002	0.017 ± 0.004

TABLE III: Task performance on online changing inclination from -0.25 rad. (down) to 0.0 rad. (flat) and to 0.25 rad. (up).

tile sensing significantly improved task performance in both inclined and step environments compared to using either sensor alone. While tactile sensing outperformed proximity sensing in simulation, their performances were closer in real-world scenarios, suggesting a larger sim-to-real gap and higher sensor noise for tactile sensors in physical environments.

D. How adaptive and robust are the learnt tool-use skills?

Finally, we qualitatively evaluated the online adaptability of the learnt tool-use policy to changing environments. We apply t-SNE to the LSTM encoder’s cell state $c_t \in \mathbb{R}^{100}$ at each timestep t for both training and testing data, with the fixed hyperparameters: perplexity = 30, learning rate = 200, number of iterations = 1000, and using Euclidean distance as the metric. Fig. 9(1) and Fig. 9(3) show the t-SNE-transformed cell states, with grey dashed lines illustrating the transition of cell states in response to changing surface inclination from -0.25 radians (down) to 0.0 radians (flat) and to 0.25 radians (up) in simulation and real settings, respectively. Similarly, the model updated cell states when detecting tool-tip contact with walls in the stairs environment in simulation (see Fig. 9(2)) and in the step environment in physical setups (see Fig. 9(4)). This analysis, which shows that our policy can quickly update the contact state encoded in the model’s cell states based on sensing feedback, supports our quantitative results on successful adaptation when hitting the wall in the “Stairs” and “Step” environments in TABLE I and TABLE II, as well as online adaptation to changing inclination of the surface shown in TABLE III. Furthermore, when the surface deformed as the robot wiped the paper, as shown in the rightmost target environment in Fig. 4, the robot successfully adapted its motion to the deforming surface.

VII. CONCLUSIONS

Our research addresses the challenge of few-shot tool-use skill transfer, enabling robots to manipulate various tools in real environments. Our proposed framework leverages

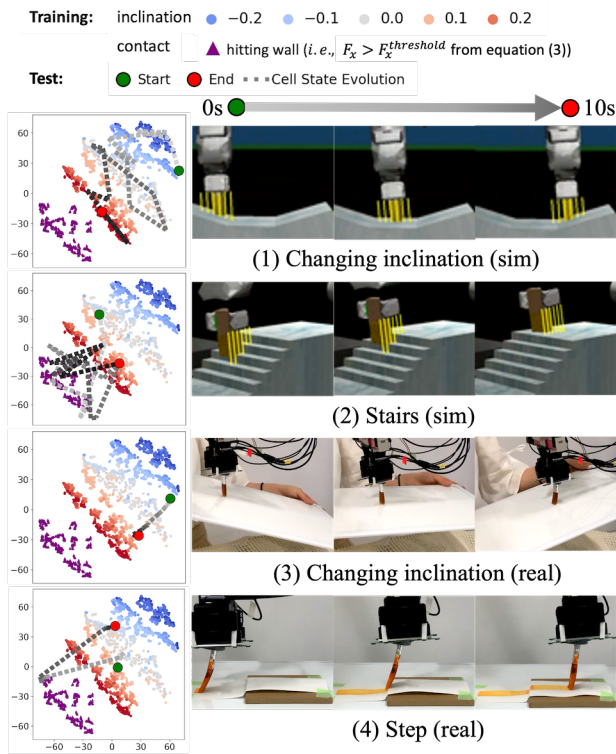


Fig. 9: The t-SNE visualisation of cell state evolution in the feature extraction module over time and the adaptation of robot motion on the fly in simulated and physical setups.

inexpensive simulation data to pre-train a base policy, followed by fine-tuning with human demonstrations to bridge the domain gap. Experimental results demonstrate that our approach enables robots to manipulate new tools with varying physical properties and adapt to environments with different geometries, with limited demonstrations in surface-following tasks. Additionally, our analysis indicates that the pre-trained policy’s ability to recognise tool-environment contact relationships using proximity and tactile sensing transfers to various downstream tool-use skills.

In this work, we used only one tool during pre-training; future work will explore multiple tools for more thorough validation of tool generalisation. One area of interest is improving robustness to tool slippage by incorporating grip force control or integrating with other grasp mechanisms [29]. Additionally, integrating vision could provide a global view of the environment, complementing local proximity and tactile sensing. Finally, applying our method to more complex tasks, such as drawing and spreading butter, offers a promising direction for demonstrating its broader adaptability.

REFERENCES

- [1] M. Qin *et al.*, “Robot tool use: A survey,” *Front. Robot. AI*, vol. 9, Jan. 2023.
- [2] M. Y. Aoyama *et al.*, “Few-shot learning of force-based motions from demonstration through pre-training of haptic representation,” in *ICRA*¹, 2024.
- [3] N. Saito *et al.*, “How to select and use tools? : Active perception of target objects using multimodal deep learning,” *RA-L*², vol. 6, no. 2, pp. 2517–2524, 2021.

¹IEEE International Conference on Robotics and Automation.

²IEEE Robotics and Automation Letters.

- [4] P. Sundaresan *et al.*, “Learning visuo-haptic skewering strategies for robot-assisted feeding,” in *6th CoRL*³, 2022.
- [5] Y. Wi *et al.*, “VIRDO++: Real-world, visuo-tactile dynamics and perception of deformable objects,” in *6th CoRL*³, 2022.
- [6] K. Yamane *et al.*, “Soft and rigid object grasping with cross-structure hand using bilateral control-based imitation learning,” *RA-L*², 2023.
- [7] D. Ma *et al.*, “Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements,” *ICRA*¹, 2021.
- [8] S. Kim *et al.*, “Simultaneous tactile estimation and control of extrinsic contact,” in *ICRA*¹, 2023.
- [9] I. Guzey *et al.*, “Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play,” in *7th CoRL*³, 2023.
- [10] Y. Shirai *et al.*, “Tactile tool manipulation,” in *ICRA*¹, 2023.
- [11] A. Bronars *et al.*, “Texterity: Tactile extrinsic dexterity,” in *ICRA*¹, 2024.
- [12] S. E. Navarro *et al.*, “Proximity perception in human-centered robotics: A survey on sensing systems and applications,” *T-RO*⁵, vol. 38, no. 3, pp. 1599–1620, 2022.
- [13] K. Sasaki *et al.*, “Robotic grasping using proximity sensors for detecting both target object and support surface,” in *IROS*⁴, 2018.
- [14] A. Billard *et al.*, “Robot programming by demonstration,” *Springer Handbook of Robotics*, B. Siciliano *et al.*, Eds., pp. 1371–1394, 2008.
- [15] M. V. der Merwe *et al.*, “Learning the dynamics of compliant tool-environment interaction for visuo-tactile contact servoing,” in *6th CoRL*³, 2022.
- [16] H. Zhang *et al.*, “Interaction control for tool manipulation on deformable objects using tactile feedback,” *RA-L*², vol. 8, no. 5, pp. 2700–2707, 2023.
- [17] M. A. Lee *et al.*, “Making sense of vision and touch: Learning multimodal representations for contact-rich tasks,” *T-RO*⁵, vol. 36, no. 3, pp. 582–596, 2020.
- [18] Y. Wi *et al.*, “CALAMARI: Contact-aware and language conditioned spatial action MAPPING for contact-Rich manipulation,” in *7th CoRL*³, 2023.
- [19] L. Wang *et al.*, *Poco: Policy composition from and for heterogeneous robot learning*, 2024.
- [20] Q. Li *et al.*, “A review of tactile information: Perception and action through touch,” *T-RO*⁵, vol. 36, no. 6, pp. 1619–1634, 2020.
- [21] H. Li *et al.*, “See, hear, and feel: Smart sensory fusion for robotic manipulation,” in *6th CoRL*³, 2022.
- [22] I. Sutskever *et al.*, “Sequence to sequence learning with neural networks,” *NeurIPS*⁶, vol. 27, 2014.
- [23] K. Cho *et al.*, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing*, 2014.
- [24] K. Kutsuzawa *et al.*, “Sequence-to-sequence model for trajectory planning of nonprehensile manipulation including contact model,” *RA-L*², vol. 3, no. 4, pp. 3606–3613, 2018.
- [25] T. Zhao *et al.*, “Learning fine-grained bimanual manipulation with low-cost hardware,” in *Robot.: Sci. Syst. XIX*, 2023.
- [26] C. Chi *et al.*, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Robot.: Sci. Syst. XIX*, 2023.
- [27] E. Todorov *et al.*, “Mujoco: A physics engine for model-based control,” in *IROS*⁴, 2012.
- [28] K. Zakka *et al.*, *MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo*, 2022.
- [29] T. Narita *et al.*, “Theoretical derivation and realization of adaptive grasping based on rotational incipient slip detection,” in *ICRA*¹, 2020.
- [30] N. Srivastava *et al.*, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [31] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [32] D. A. Pomerleau, “Alvin: An autonomous land vehicle in a neural network,” *NeurIPS*⁶, 1988.
- [33] M. Al-Shedivat *et al.*, “Learning scalable deep kernels with recurrent structure,” *J. Mach. Learn. Res.*, vol. 18, no. 82, pp. 1–37, 2017.
- [34] S. Ross *et al.*, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *Proc. of the 14th Int. Conf. Artif. Intell. Stat.*, 2011.

³Annual Conference on Robot Learning.

⁴IEEE/RSJ International Conference on Intelligent Robots and Systems.

⁵IEEE Transactions on Robotics.

⁶Advances in neural information processing systems.