IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, August 2023.
Best Conference Paper Award.

Learning Generalizable Tool Use with Non-rigid Grasp-pose Registration

Malte Mosbach and Sven Behnke

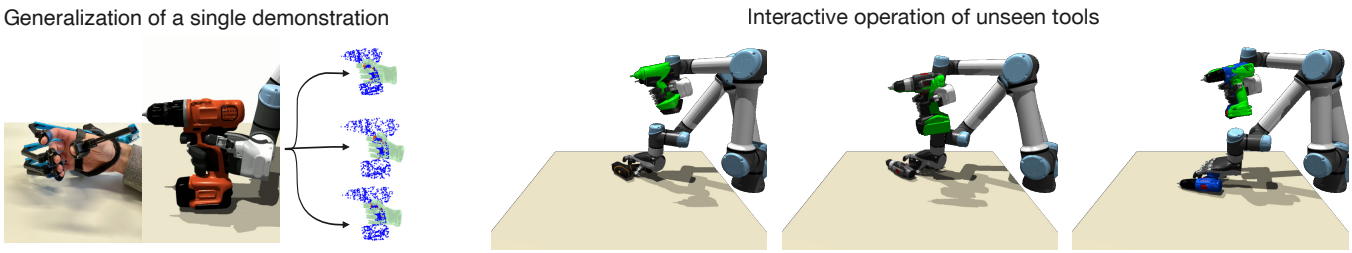


Fig. 1: A single grasping demonstration is transferred to other instances of a class, including instances not in the training set and only partially observed (left). These generalized demonstrations guide the learning of an interactive policy able to operate a variety of tools (right).

***Abstract*— Tool use, a hallmark feature of human intelligence, remains a challenging problem in robotics due the complex contacts and high-dimensional action space. In this work, we present a novel method to enable reinforcement learning of tool use behaviors. Our approach provides a scalable way to learn the operation of tools in a new category using only a single demonstration. To this end, we propose a new method for generalizing grasping configurations of multi-fingered robotic hands to novel objects. This is used to guide the policy search via favorable initializations and a shaped reward signal. The learned policies solve complex tool use tasks and generalize to unseen tools at test time. Visualizations and videos of the trained policies are available at https://maltemosbach.github.io/generalizable_tool_use.**

I. Introduction

The use of tools to achieve desired changes to the environment is a hallmark feature of human intelligence [1], [2]. This includes various behaviors, from using a vessel to carry water, to driving a nail with a hammer, to operating a power drill. While humans routinely use specialized tools for construction and assembly tasks, this behavior has been challenging to automate because of the high-dimensional action space of humanoids robots and the intra-class variability of the tools made for human hands.

Despite these challenges, tool use remains a central task in robot learning, due to its overwhelming practical utility. Classical approaches for tool use include affordance learning [3], [4] and dynamic motion primitives [5], [6]. These rely on predefined exploration primitives or trajectories and lack the interactive manipulation capabilities that humans so effortlessly exhibit. Reinforcement learning (RL) [7], [2] has recently been used to generate interactive control policies, but suffers from the high-dimensional action space of human-like robotic hands. This leads to excessive sample complexity, if convergences can be achieved at all. To enable RL to handle manipulation tasks in the intricacy of interactive tool use, auxiliary guidance such as demonstration datasets or precise reward engineering is needed [7].

Using demonstrations to communicate the desired behavior to a robot is a intuitive approach since humans can provide competent demonstrations for anthropomorphic end-effectors. However, existing methods make only limited use of demonstrations. Consider the task of grasping a hammer to drive a nail. To derive this general skill, regular imitation learning would necessitate a vast number of demonstrations spanning various tool instances. Instead, we want our robot to use tools as flexibly as humans do, relating demonstrated behaviors to different instances without the need for repeated demonstrations. While prior work considers intra-class variation of object instances [8], [9], the grasping of tools is framed as reaching a desired grasping position derived from an initial observation of the tool. This is in stark contrast to the way humans interact with tools and objects, where perception and action are continuously interleaved, making adaptive behaviors and operation in unstructured environments possible. Humans have the ability to effortlessly generalize prior knowledge and interactively adapt their behavior, enabling them to operate unfamiliar tools with ease. While generalization to new tools and their interactive operation have been demonstrated individually, to the best of our knowledge, no prior method realizes both.

In this work, we present a system that learns a continuous control policy to operate a variety of tools under the guidance of only a single human demonstration. To this end, we introduce a procedure that utilizes non-rigid registration to generalize a canonical grasping demonstration to novel instances and use these demonstrations to guide policy search, without imposing rigid behaviors. This is achieved by initializing episodes in pre-grasp poses to enable efficient exploration and by inducing prior knowledge about how to grasp a tool through a shaped reward function. Teh effectiveness of our proposed approach is experimentally evaluated on three simulated tool-use tasks. Specifically, we

Both authors are with the Autonomous Intelligent Systems group, University of Bonn, Germany; mosbach@ais.uni-bonn.de

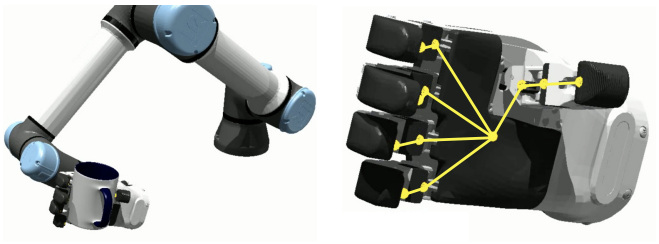


Fig. 2: Task space of human-like grasping. Grasping and manipulating instances happens by making contact with the elements of the fingers and palm (left). Hence, this naturally defines the task-space for multi-fingered robotic hands (right).

make the following contributions:

- We present a novel method that uses non-rigid registration to generalize grasp-configurations to unknown instances of a class.
- We examine how grasping demonstrations can be used to guide the learning of an RL policy.
- We demonstrate, in simulation, that interactive operation of different tools can be learned with model-free RL using only a single demonstration.

II. Generalizing Demonstrated Grasps

In grasping and tool use tasks, human-like robotic hands manipulate objects by inducing contact with the inside of the finger phalanges and the palm. The corresponding keypoints, also referred to as task-space vectors [10], which are shown in Fig. 2, define the features of a grasp to be preserved during generalization to novel instances. We found that such detailed multi-fingered grasping configurations can be accurately mapped between instances in a *two-step approach*. Specifically, we uniquely combine non-rigid registration and hand pose retargeting to construct a system for generalization of multi-fingered grasping configurations. First, in Sec. II-A, we leverage latent non-rigid registration to continuously deform the canonical object (and its demonstration keypoints) to match the observed object. This preserves characteristic category-level features of a grasp and works directly from partial point cloud observations. Second, in Sec. II-B, we optimize the end-effector pose and joint positions of the robot hand to find a kinematically feasible grasp that minimizes the distance in task space.

A. Category-level Grasp Pose Transfer

1) Coherent Point Drift: To explain the first step in our grasp pose generalization method, we briefly review the coherent point drift (CPD) algorithm [11]. Given a target point set $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and a source point set $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$, our goal is to find a transformation that maps $\boldsymbol{Y}$ to $\boldsymbol{X}$. CPD builds a Gaussian mixture model (GMM) from the moving point set, $\boldsymbol{Y}$, and treats the points in $\boldsymbol{X}$ as observations drawn from it. An expectation-maximization (EM) algorithm is used to optimize the GMM while obeying a smoothness constraint based on motion coherence theory [12]. The non-rigid transformation $\mathcal{T}$ mapping $\boldsymbol{Y}$ to $\boldsymbol{X}$ is given by:

$$\mathcal{T}(\boldsymbol{Y}, v) = \boldsymbol{Y} + v(\boldsymbol{Y}), \tag{1}$$

where the displacement function v is defined for any set of points $\boldsymbol{Z}$ as:

$$v(\boldsymbol{Z}) = G(\boldsymbol{Y}, \boldsymbol{Z})\boldsymbol{W}. \tag{2}$$

$G(\cdot, \cdot)$ denotes a Gaussian kernel matrix. CPD estimates the matrix of kernel weights, $\boldsymbol{W}$, which can be understood as a set of deformation vectors associated with the points in $\boldsymbol{Y}$. Thus, the transformation from a canonical model $\boldsymbol{C}$ to a training instance $\boldsymbol{T}_i$ is defined as:

$$\mathcal{T}_i(\boldsymbol{C}, \boldsymbol{W}_i) = \boldsymbol{C} + G(\boldsymbol{C}, \boldsymbol{C})\boldsymbol{W}_i. \tag{3}$$

2) Latent Deformation Field Manifold: Our goal is to extrapolate from a single demonstration to novel objects of the same category, utilizing understanding of common intra-class variability. Therefore, we use CPD to find the deformation $\mathcal{T}_i(\boldsymbol{C}, \boldsymbol{W}_i)$ from the canonical instance, $\boldsymbol{C}$, to all other training instances $\boldsymbol{T}_i$. The uniqueness of each deformation is captured entirely by $\boldsymbol{W}_i \in \mathbb{R}^{M\times 3}$. The corresponding row vectors $\boldsymbol{x}_i \in \mathbb{R}^{3M}$, which are the feature descriptors of the deformation fields, are assembled into a design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times 3M}$, where n is the number of training instances. Finally, we perform principal component analysis (PCA) on $\boldsymbol{X}$, to find a lower-dimensional manifold of characteristic deformations $\boldsymbol{L} \in \mathbb{R}^{q\times 3M}$, where $q \ll n$ is the number of eigenvectors to keep, i.e., the dimensionality of our ensuing latent space. Characteristic deformations of a category can now be modeled in the q-dimensional latent space.

When encountering a new instance that has been partially observed through a segmented point cloud $\boldsymbol{O}$, we fit the latent parameter vector $\boldsymbol{\ell}$ to match its shape. Specifically, we aim

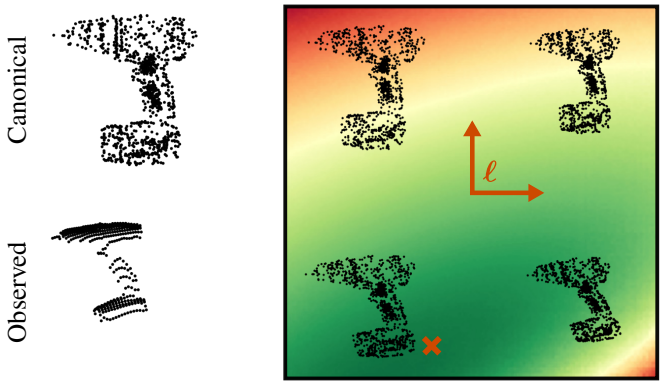


Fig. 3: Energy landscape induced by observed instance. A partially observed instance induces an energy function over the latent shape parameters $\boldsymbol{\ell}$. Optimizing this energy function yields latent space parameters that best map the canonical instance to the observation.

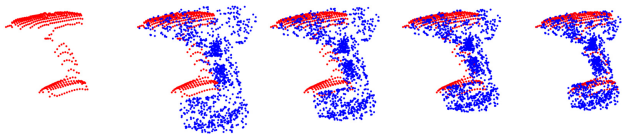


Fig. 4: Registration of observed instance. The latent space parameters are optimized to fit the canonical instance (blue) to the observation (red).

to minimize the energy function:

$$E(\boldsymbol{\ell}) = -\sum_{m=1}^{M} \log \sum_{n=1}^{N} e^{-\frac{1}{2\sigma^2}\|\boldsymbol{O}_n - \mathcal{T}(\boldsymbol{C}_m, W_m(\boldsymbol{\ell}))\|^2}, \quad (4)$$

illustrated in Fig. 3, via gradient descent. Fig. 4 shows how this optimization deforms the canonical object to match the observed instance. Since non-rigid registration in latent space permits only deformations actually observed in a class, we can register even partially observed objects without invoking undesired deformations of the canonical model. Once a minimum is found, we can use the resulting deformation field to transform the keypoints of the canonical demonstration into generalized keypoint poses.

B. Pose Regression in Task Space

So far, we have only shown how the deformation field between canonical and observed instances can be used to transform feature points of a grasp. While previous work [13], [9] uses this property to generalize trajectory control poses, this forgoes the inherent advantage of multi-fingered manipulators to grasp in diverse finger configurations. Our aim is to maintain the properties of a grasp when it is transferred to new objects, which entails determining a relationship between the deformation of an object and the joint angles of the resultant grasp. Thus, while the transferred keypoints represent the desired hand pose, they might not define a reachable position under the kinematic constraints of the used end-effector. To address this issue, we introduce a second optimization step, inspired by motion retargeting approaches [14], [10], to find an attainable grasp configuration. We optimize the wrist pose $\boldsymbol{p}$ and joint positions $\boldsymbol{q}$ of the robot hand to minimize the distance to the transferred keypoints $\boldsymbol{k}_i$:

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{i=0}^{N} \|\boldsymbol{k}_i - f_i(\boldsymbol{p}, \boldsymbol{q})\|^2, \quad (5)$$

where f_i represents the forward kinematics of the ith keypoint. Solving this optimization problem yields the hand pose and joint angles that best preserve task-space characteristics of the demonstrated grasp. Fig. 5 shows the process of optimizing for minimal task-space distance (Eq. 5). The robot hand converges to a feasible grasping position that maintains characteristic features of the original demonstration.

III. Interactive Tool Use

Thus far, we have introduced our method for intra-class generalization of functional grasps. However, we only concerned ourselves with static grasp poses. In the following, we describe how the obtained demonstrations can be used to guide the learning process of interactive RL policies. Sec. III-A presents pre-grasp poses as efficient exploration primitives. In Sec. III-B, we propose a shaped reward function to direct the policy based on the generalized grasp poses. Lastly, Sec. III-C discusses how grasp poses and tool-use policies can generalize to instances beyond the training set.

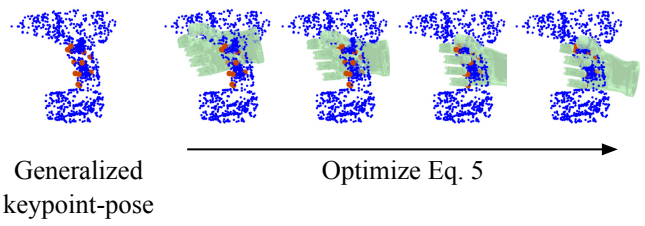


Fig. 5: Pose regression in task-space. Given a generalized keypoint-pose (left), an attainable grasp pose is found by minimizing the keypoint distance (right).

A. Pre-grasp Poses for Efficient Exploration

The process of grasping tools can be described by an initial *reaching* and a subsequent *dexterous manipulation* phase [15]. The first phase, where the robot reaches for the tool but does not yet make contact, can be solved very efficiently by conventional feedback control methods. Only the subsequent high-contact manipulation requires an interactive RL policy. In this context, our generalized grasp poses can be utilized to favorably position the robotic hand at the beginning of each episode by moving to a pose offset from our final target. Specifically, we have the robot approach a pose that is removed from the target grasp in the direction normal to the palm, and interpolate the joint angles between the open and target configurations. On the right of Fig. 1, we overlay both the pre-grasp configuration, which represents the beginning of the episode, and a later configuration just before the task is completed. Pre-grasp poses serve as a critical precursor to efficient exploration and successful learning of the task at hand.

B. Grasp-pose Guided RL

The operation of various tools can be mastered efficiently once robust grasps have been learned. Hence, the generalized grasp poses are used to inject prior knowledge on how a tool ought to be grasped by parametrizing a shaped reward function. The reward terms encouraging the agent to reach the demonstrated grasp-pose are detailed at the bottom of Tab. I. Specifically, an incentive is provided to minimize the distance to the demonstrated keypoints. A second reward term trains the agent to pick up the tool from the table, which requires learning a stable grasp and simple maneuvering of the tool. We have found that defining the object height in terms of the object's root coordinate system can lead to undesirable local optima. For example, the coordinate root of the drill is at the tip of the tool, which causes the agent to learn unhelpful solutions, such as tilting the drill up without ever actually lifting it. To avoid these problems, we

TABLE I: Grasp-pose guided reward. The reward function combines **task-specific rewards** encouraging goal-directed behavior and **tool grasping rewards** that encourage the agent to reach the demonstrated grasping pose.

Term	Equation	Weight
○ *Place mug*		
r_{pose}: target pose matching	$e^{-\alpha \|x_t^{(p)} - \overline{x}_t^{(p)}\| - \beta \angle(x_t^{(o)}, \overline{x}_t^{(o)})}$	25.0
r_{success}: target pose reached	$\mathbb{1}(\text{pose\_reached})$	100.0
○ *Position drill*		
r_{pose}: target pose matching	$e^{-\alpha \|x_t^{(p)} - \overline{x}_t^{(p)}\| - \beta \angle(x_t^{(o)}, \overline{x}_t^{(o)})}$	25.0
r_{success}: target pose reached	$\mathbb{1}(\text{pose\_reached})$	100.0
○ *Drive nail*		
r_{dist}: move hammer to nail	$(\epsilon + \Delta x)^{-1}$	0.25
r_{depth}: nail depth	Δd_{nail}	100.0
r_{kp}: keypoint matching	$(\epsilon + \Delta k)^{-1}$	0.001
r_{lift}: tool lifting	$(\epsilon + \Delta h)^{-1}$	0.05

Task-specific

Tool grasping

We use $\alpha = 10$, $\beta = 1$, and $\epsilon = 0.025$.

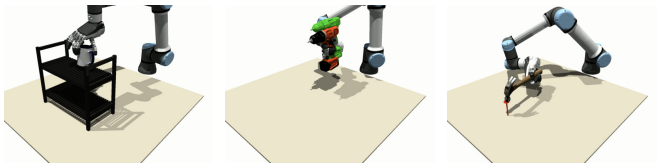
decided to sample a synthetic point cloud on the tool's mesh. The height of the tool is then defined by the height of its lowest surface point, which is an approximation of the actual clearance between the tool and the table. We use proximal policy optimization [16] to train our policies to maximize this reward.

C. Transfer to Unseen Tools

Ultimately, we want the learned policies to be able to operate unseen tools. Thus, our goal is to generalize the grasp pose to a novel instance without access to its object mesh. We therefore add a depth camera sensor to the environment, as shown in Fig. 7. Unlike the synthetic point clouds, the sensor data suffers from occlusions. Simply applying CPD would now deform the canonical instance in unhelpful ways. However, the learned category-level shape space can circumvent this problem. Since the low-dimensional deformation field manifold only allows for deformations that are characteristic of the variance in a class, the canonical model can even be fitted to a partially observed instance.

IV. Experimental Setup

Our experiments aim to answer how effectively the proposed method can solve the challenging task of robotic tool use based on a single demonstration. Specifically, we evaluate (1) Whether a canonical demonstration can be generalized to new instances; (2) how effectively model-free RL can solve the posed tasks based on the generalized demonstrations; (3) whether our policies can generalize to novel, partially observed tools in a zero-shot manner.



Place mug Position drill Drive nail

Fig. 6: Tool use tasks. The environments represent familiar tool use tasks a robot might be asked to solve.

A. Problem Statement

Our goal is to learn a policy π to utilize a tool $(T_i | i = 1, \ldots, N)$ in order to achieve some goal-directed behavior, e.g. using a hammer to drive a nail. Moreover, the policy should be able to operate a variety of tool instances and generalize to unseen tools at test time. In a repeated interaction, the policy observes the current state of the environment $\boldsymbol{s}_t \in \mathcal{S}$, performs an action $\boldsymbol{a}_t \in \mathcal{A}$, and receives a reward signal r_t. We define the observations of the policy to include proprioceptive observations of the robot state (wrist pose and keypoints of the hand), as well as a low-dimensional observation of the tool represented by its generalized demonstration and latent shape parameters. Additionally, the policy receives information about task-specific objectives, such as the desired pose of the drill. The action space $\mathcal{A}$ comprises the desired change to the end-effector pose and joint positions of the robot hand. The agent chooses actions at a frequency of 30 Hz. The reward function is the sum of the terms detailed in Tab. I.

B. Tool Use Tasks

We evaluated our method on three tool categories: *Drills*, *Hammers*, and *Mugs*, each with one canonical, 10 training, and 3 test instances. The models were obtained from the online databases GrabCAD[1], 3DWarehouse[2], and Sketchfab[3]. The simulated robot combines a UR5e arm controlled by its end effector pose with a Schunk SIH hand that has 11 degrees of freedom (DoFs), 5 of which are fully actuated. We use NVIDIA Isaac Gym [17] to simulate the tool use tasks shown in Fig. 6. In each run, 16,384 parallel agents are trained for a total of 134 million simulated steps, which corresponds to approximately 52 real-time days. This requires just under 3 hours of wall-clock time on a single NVIDIA A6000 GPU. At test time, we required an average of 3 seconds to match the canonical model to an observed instance.

C. Demonstrations

Our method draws on human grasping knowledge to accelerate the learning process. To demonstrate grasping

[1] https://grabcad.com/library
[2] https://3dwarehouse.sketchup.com/
[3] https://sketchfab.com

TABLE II: Task-space distance.

	Mugs	Drills	Hammers
Ours	$\mathbf{0.68 \pm 0.26}$	$\mathbf{0.72 \pm 0.30}$	$\mathbf{0.78 \pm 0.34}$
WP	2.27 ± 1.12	2.76 ± 1.74	2.64 ± 1.12
CG	2.00 ± 0.97	2.89 ± 1.45	3.88 ± 2.22

Mean distance in cm of the grasps proposed by our method and ablations to the keypoints of the generalized demonstration.

postures in an intuitive way, we introduce a virtual reality (VR) interface to Isaac Gym. The operator's movements are tracked by a SenseGlove DK1, which captures finger angles, and an HTC Vive tracker, which records the hand pose. This device, worn by the operator, can be seen on the left in Figure 1. An HTC Vive headset is integrated with Isaac Gym's camera sensors to provide a stereoscopic visualization of the scene. The operator interacts with the tasks in a natural way, indicating at the push of a button that the current pose should serve as the canonical demonstration.

D. Evaluation Procedure

For the *Place mug* and *Position drill* tasks, the success criterion is based on the distance of the tool pose and target pose. We consider an episode as completed successfully if $d < \bar{d}$ and $\theta < \bar{\theta}$, where d and θ are the positional and angular distance to the target pose. For both environments we use $\bar{d} = 0.03\text{m}$ and $\bar{\theta} = 0.2\text{rad}$. The *Drive nail* environment considers runs successful, where the nail has been driven by a depth of greater than 0.075m.

V. Results

A. Analysis of Generated Grasps

First, we investigate the kind of grasp poses that the proposed framework generates. Here we compare with two

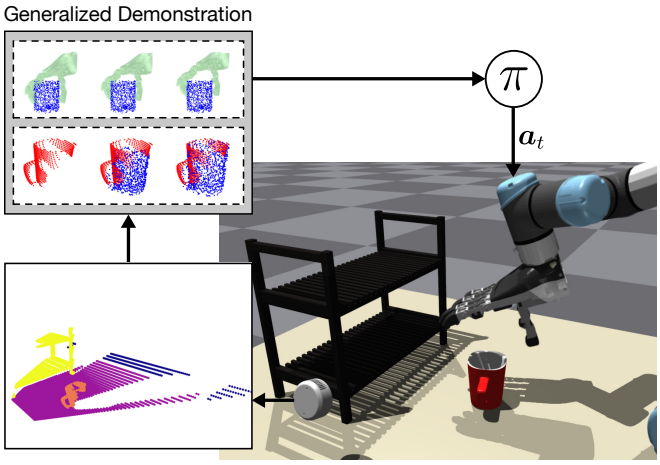


Fig. 7: Grasp generalization from vision. We add a sensor to the simulation that outputs segmented point clouds. We fit the canonical model of the respective object category to the measurements belonging to the tool (Eq. 4). We then find the joint configuration that minimizes the task-space distance (Eq. 5) and pass this generalized demonstration to our policy.

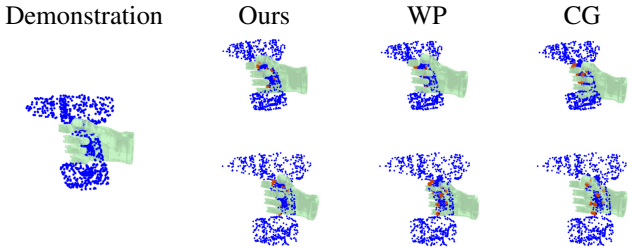


Fig. 8: Generated grasp-poses. The proposed approach generates grasp poses that aim to be equivalent in task-space. Generalizing trajectory control poses, such as the wrist pose to be reached before closing the fingers (WP) does not have this desired property.

ablations: Retention of the canonical grasp (CG) and transformation of the wrist pose while keeping grasping behavior constant (WP). To assess the quality of a grasp, we measure the distance to the transformed keypoints over all training instances in a class. Quantitative results are shown in Tab. II. The proposed method outperforms both baselines by a large margin, and the results are consistent across tool categories. Examples of the generalized grasp-poses shown in Fig. 8 confirm that our approach finds feasible grasps for varying object shapes.

B. Grasp-pose Guided RL

Next, we evaluate the ability of generalized grasp pose demonstrations to guide policy search on challenging tool use tasks. The results in Tab. III show that the proposed method consistently finds the intended grasps across the tasks and tools studied. Furthermore, the Place Mug and Position Drill tasks are solved with high reliability. Driving a nail proved to be the most challenging task to complete, as the agent must maintain its grasp on the hammer while making forceful contact with the environment. Now, we compare the performance of our proposed method to multiple ablations. First, we examine how performance changes when we disable our grasp generalization (w/o GG) and instead apply canonical demonstration to all objects. As can be seen, this still leads to viable training performance for objects with lower variance, such as cups, while performance deteriorates more severely for objects that vary greatly in their extent and grasping position, such as drills and hammers. Not navigating to a pre-grasp pose at the beginning of the episode (w/o PG) causes the training to fail. The agent is not able to find the correct grasp posture, but frequently gets stuck in local optima. Lastly, we compare to a baseline where the task is approached without demonstration guidance (w/o demo). Here, the agent receives only task-specific rewards and is initialized in a default neutral position above the table. Again, learning the full manipulation tasks is unsuccessful, as discovering useful behaviors that make progress on the proposed task is extremely difficult in this situation.

The results show that knowledge about how to grasp an object, which can be incorporated via shaped rewards or pre-grasp poses, is a valuable addition to RL training. Moreover,

TABLE III: Training performance. Success rates of the proposed method and studied ablations to grasp the tool and solve the full task.

	Place mug		Position drill		Drive nail	
	grasp	full	grasp	full	grasp	full
Ours	**0.97**	**0.96**	**0.94**	**0.76**	**0.8**	**0.65**
w/o GG	**0.97**	0.95	0.81	0.66	0.74	0.61
w/o PG	0.01	0.0	0.41	0.0	0.4	0.0
w/o demo	0.0	0.0	0.0	0.0	0.0	0.0

having a method that generalizes such demonstrations to new objects in a class removes the high overhead of collecting a large number of demonstrations and allows training to scale more easily.

C. Zero-shot Transfer to Unseen Tools

Finally, we investigate whether the policy is able to transfer to unseen tools in a zero-shot manner. Here, we do not assume access to the object mesh, instead perceiving the scene via a segmented point-cloud, as shown in Fig. 7. The canonical demonstration is then adjusted to fit the observed instance and given to the policy. It can be seen in Tab. IV, that the policies can grasp and operate even some of the unseen tools without finetuning. Extending the training set may help to close the performance gap between the training and test instances in the future.

VI. Related Work

1) Robotic grasping: Despite decades of active research efforts, robotic grasping remains an unsolved problem [18]. Grasping has traditionally been framed as the open-loop procedure of grasp-pose prediction (grasp synthesis). Several prior works estimate grasp-poses through analytical [19], [20], [21] or learned [22], [23] methods. In recent years, RL has become popular for robotic grasping and manipulation due to its ability to generate interactive policies in a model-free manner. Kalashnikov et al. [24] train a vision-based grasping policy to control a parallel gripper. Shahid et al. [25] demonstrate that RL can be used to continuously control a Franka Emika Panda manipulator to lift objects off a table. However, RL has struggled with the high-dimensional action space of anthropomorphic end-effectors. One group of work has aimed to scale up experience collection via parallelized GPU-accelerated physics simulation [17], [26], [27]. Alternatively, human demonstrations have been used by themselves [14], [10] or in combination with RL [28], [7] to solve grasping and manipulation tasks.

2) Robotic tool use: Prior works studying robotic tool use span classical [29], [30] and learning-based [31], [2], [32] approaches. Xie et al. [32] learn to predict the visual outcome of actions based on human demonstration and autonomous interaction data. Planning with the learned model can solve improvised tool use tasks with a parallel gripper. Wenke et al. [2] study reasoning and generalization in RL through the lens of tool use. They train RL agents to solve grid-world versions of the classical trap-tube experiment. Notably, Dasari et al. [15] demonstrate that pre-grasp poses can be used to improve dexterous manipulation learning. While their objective of using grasp-poses to accelerate RL is aligned with the goal of our work, they do not consider generalization of grasp-poses or policies between different tools. To the best of our knowledge, the amalgamation of transferring demonstrations between instances and interactive RL training is novel to our work.

TABLE IV: Test performance. Success rate of the proposed method when operating unseen tools. The generalized demonstration is estimated from a partial point-cloud of the tool.

Place mug	Position drill	Drive nail
0.67 ± 0.11	0.62 ± 0.15	0.55 ± 0.1

3) Grasp-pose transfer.: Multiple lines of work aim to generalize demonstrated behaviors to novel instances in a class. Object-meshes segmented via shape and volumetric information are used by Vahrenkamp et al. [33] to transfer grasps from a template set to familiar objects. Stückler et al. [13] transfer poses and trajectories defining grasping motion via the dense deformation field from the known object model to an observed instance. Rodriguez et al. [9] extend this work by modelling deformations not only between a known and observed instance, but within a category. This makes it possible to register partially observed objects. In [34] and [35], contact points are warped from a known object to an observed object. However, both assume that the objects are fully observed. Simeonovdu et al. [36] present neural descriptor fields which represent an object by a mapping from each 3D point $\boldsymbol{x}$ to a latent descriptor z encoding relations to salient object features. This description is used to establish correspondences of semantically meaningful object features, and thereby generalize demonstrations to new instances. Our work builds on [9], but generalizes characteristic features of a multi-fingered grasp through optimization in task space. Further, we demonstrate how the generalized demonstrations can be used as a basis for learning interactive tool use policies, rather than as parameters for open loop grasping behavior.

VII. Discussion and Conclusion

We have shown that the challenging domain of robotic tool use becomes approachable for model-free RL with the use of only a single human demonstration. The proposed generalization scheme can transfer grasp poses even to partially observed instances while retaining characteristic features of the demonstrated functional grasp. The RL experiments underscore the benefits of extending grasp pose generalization to the domain of interactive control, as the policies are for example able to continuously manipulate drills lying on the table until a desired grasp is achieved. Although we only present results in simulation, we have shown how the latent shape parameters and grasping configuration of a novel object can be estimated from its partial point-cloud observation.

Still, there are several limitations and opportunities for future work. Transferring the obtained results to the real robot system is the most evident task. Developing a way to track tools during the grasping process and obtain well separated point-clouds of a scene are key challenges to be overcome. In addition, developing an approach that can generate class-independent grasping or pre-grasp poses would be valuable.

ACKNOWLEDGEMENT

This work has been funded by the German Ministry of Education and Research (BMBF), grant no. 01IS21080, project "Learn2Grasp: Learning Human-like Interactive Grasping based on Visual and Haptic Feedback".

REFERENCES

[1] L. Jamone, "Modelling human tool use in robots," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 907–908, 2022.

[2] S. Wenke, D. Saunders, M. Qiu, and J. Fleming, "Reasoning and generalization in RL: A tool use perspective," *arXiv preprint arXiv:1907.02050*, 2019.

[3] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 3060–3065.

[4] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Modeling affordances using bayesian networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 4102–4107.

[5] J. Kober, B. Mohler, and J. Peters, "Learning perceptual coupling for motor primitives," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 834–839.

[6] K. Muelling, J. Kober, and J. Peters, "Learning table tennis with a mixture of motor primitives," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 411–416.

[7] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[8] D. Rodriguez, F. Huber, and S. Behnke, "Category-level 3D non-rigid registration from single-view rgb images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 617–10 624.

[9] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4229–4236.

[10] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "DexMV: Imitation learning for dexterous manipulation from human videos," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 570–587.

[11] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 12, pp. 2262–2275, 2010.

[12] A. L. Yuille and N. M. Grzywacz, "The motion coherence theory," in *International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 1988, pp. 344–345.

[13] J. Stückler and S. Behnke, "Efficient deformable registration of multi-resolution surfel maps for object manipulation skill transfer," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 994–1001.

[14] Y. Qin, H. Su, and X. Wang, "From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 10 873–10 881, 2022.

[15] S. Dasari, A. Gupta, and V. Kumar, "Learning dexterous manipulation from exemplar object trajectories and pre-grasps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

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

[17] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac Gym: High performance GPU-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[18] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research (IJRR)*, vol. 41, no. 7, pp. 690–705, 2022.

[19] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[20] J. Ponce, S. Sullivan, J.-D. Boissonnat, and J.-P. Merlet, "On characterizing and computing three-and four-finger force-closure grasps of polyhedral objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1993, pp. 821–827.

[21] D. Ding, Y.-H. Liu, and S. Wang, "Computing 3-d optimal form-closure grasps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 3573–3578.

[22] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics (T-RO)*, vol. 30, no. 2, pp. 289–309, 2013.

[23] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, pp. 239–249, 2020.

[24] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning (CoRL)*. PMLR, 2018, pp. 651–673.

[25] A. A. Shahid, L. Roveda, D. Piga, and F. Braghin, "Learning continuous control actions for robotic grasping with reinforcement learning," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 4066–4072.

[26] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning (CoRL)*. PMLR, 2022, pp. 297–307.

[27] M. Mosbach and S. Behnke, "Efficient representations of object geometry for reinforcement learning of interactive grasping policies," in *IEEE International Conference on Robotic Computing (IRC)*, 2022.

[28] M. Mosbach, K. Moraw, and S. Behnke, "Accelerating interactive human-like manipulation learning with gpu-based simulation and high-quality demonstrations," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 435–441.

[29] S. Brown and C. Sammut, "A relational approach to tool-use learning in robots," in *International Conference on Inductive Logic Programming (ILP)*, 2012, pp. 1–15.

[30] M. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 6231–6235.

[31] 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," *The International Journal of Robotics Research (IJRR)*, vol. 39, no. 2-3, pp. 202–216, 2020.

[32] A. Xie, F. Ebert, S. Levine, and C. Finn, "Improvisation through physical understanding: Using novel objects as tools with visual foresight," *arXiv preprint arXiv:1904.05538*, 2019.

[33] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 919–925.

[34] T. Stouraitis, U. Hillenbrand, and M. A. Roa, "Functional power grasps transferred through warping and replanning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4933–4940.

[35] H. B. Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2043–2050.

[36] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: SE(3)-equivariant object representations for manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6394–6400.