UniDexGrasp++: Improving Dexterous Grasping Policy Learning via Geometry-aware Curriculum and Iterative Generalist-Specialist Learning

Weikang Wan [1*] Haoran Geng [1,3*] Yun Liu [2]
Zikang Shan [1] Yaodong Yang [1,3] Li Yi [2] He Wang [1†]
[1]Peking University [2]Tsinghua University
[3]Beijing Institute for General Artificial Intelligence

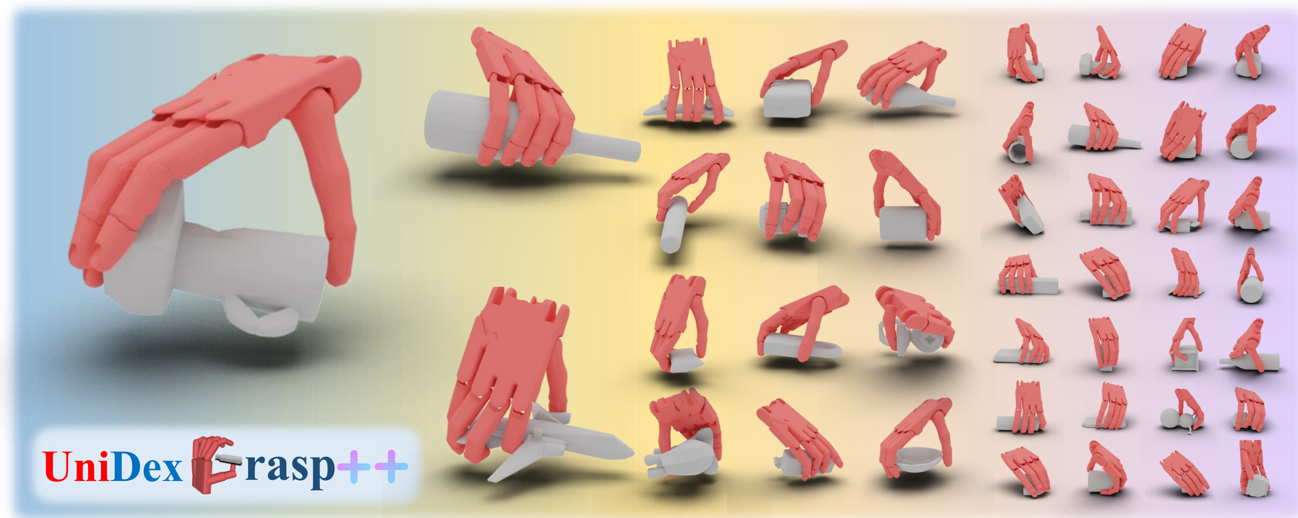

Figure 1: In this work, we present a novel dexterous grasping policy learning pipeline, UniDexGrasp++. Same to UniDexGrasp[69], UniDexGrasp++ is trained on 3000+ different object instances with random object poses under a table-top setting. It significantly outperforms the previous SOTA and achieves 85.4% and 78.2% success rates on the train and test set.

Abstract

*We propose a novel, object-agnostic method for learning a universal policy for dexterous object grasping from realistic point cloud observations and proprioceptive information under a table-top setting, namely UniDexGrasp++. To address the challenge of learning the vision-based policy across thousands of object instances, we propose Geometry-aware Curriculum Learning (**GeoCurriculum**) and Geometry-aware iterative Generalist-Specialist Learning (**GiGSL**) which leverage the geometry feature of the task and significantly improve the generalizability. With our proposed techniques, our final policy shows universal dexterous grasping on thousands of object instances with **85.4%** and **78.2%** success rate on the train set and test set which outperforms the state-of-the-art baseline UniDexGrasp by 11.7% and 11.3%, respectively.*

*Equal contribution.
†Corresponding author.

1. Introduction

Robotic grasping is a fundamental and extensively studied problem in robotics, and it has recently gained broader attention from the computer vision community. Recent works [61, 6, 17, 23, 65, 16, 12] have made significant progress in developing grasping algorithms for parallel grippers, using either reinforcement learning or motion planning. However, traditional parallel grippers have limited flexibility, which hinders their ability to assist humans in daily life.

Consequently, dexterous grasping is becoming more important, as it provides a more diverse range of grasping strategies and enables more advanced manipulation techniques. The high dimensionality of the action space (e.g., 24 to 30 degrees of freedom) of a dexterous hand is a key advantage that provides it with high versatility and, at the same time, the primary cause of the difficulty in executing a successful grasp. What's more, the complex hand articulation significantly degrades motion planning-based grasping methods, making RL the mainstream of dexterous grasping.

However, it is very challenging to directly train a vision-based universal dexterous grasping policy [36, 37, 39, 58]. First, vision-based policy learning is known to be difficult, since the policy gradients from RL are usually too noisy to update the vision backbone. Second, such policy learning is in nature a multi-task RL problem that carries huge variations (e.g., different geometry and poses) and is known to be hard [39, 28, 58]. Despite recent advancements in reinforcement learning (RL) [4, 2, 36, 8, 9, 48, 40, 26, 37, 57, 67]that have shown promising results in complex dexterous manipulation, the trained policy cannot easily generalize to a large number of objects and the unseen. At the same time, most works [4, 2, 67, 9, 57, 48, 26] assume the robot knows all oracle information such as object position and rotation, making them unrealistic in the real world.

A recent work, UniDexGrasp [68], shows promising results in vision-based dexterous grasping on their benchmark that covers more than 3000 object instances. Their policy only takes robot proprioceptive information and realistic point cloud observations as input. To ease policy learning, UniDexGrasp proposes object curriculum learning that starts RL with one object and gradually incorporates similar objects from the same categories or similar categories into training to get a state-based teacher policy. After getting this teacher policy, they distill this policy to a vision-based policy using DAgger [50]. It finally achieves 73.7% and 66.9% success rates on the train and test splits. One limitation of UniDexGrasp is that its state-based teacher policy can only reach 79.4% on the training set, which further constrains the performance of the vision-based student policy. Another limitation in the object curriculum is unawareness of object pose and reliance on category labels.

To overcome these limitations, we propose UniDexGrasp++, a novel pipeline that significantly improves the performance of UniDexGrasp. First, to improve the performance of the state-based teacher policy, we first propose Geometry-aware Task Curriculum Learning ***(GeoCurriculum)*** that measures the task similarity based on the geometry feature of the scene point cloud. To further improve the generalizability of the policy, we adopt the idea of *generalist-specialist learning* [62, 38, 22, 28] where a group of specialists is trained on the subset of the task space then distill to one generalist. We further propose Geometry-aware iterative Generalist-Specialist Learning ***GiGSL*** where we use the geometry feature to decide which specialist handles which task and iteratively do distillation and fine-tuning. Our method yields the best-performing state-based policy, which achieves **87.9%** and **83.7%** success rate on the train set and test set. Then we distill the best-performing specialists to a vision-based generalist and do GiGSL again on vision-based policies until it reaches performance saturation. With our full pipeline, our final vision-based policy shows universal dexterous grasping on 3000+ object instances with **85.4%** and **78.2%** success rate on the train set and test set that remarkably outperforms the state-of-the-art baseline UniDexGrasp by 11.7% and 11.3%, respectively. The additional experiment on Meta-World [71] further demonstrates the effectiveness of our method which outperforms the previous SOTA multi-task RL methods.

2. Related Work

2.1. Dexterous Grasping

Dexterous hand has received extensive attention for its potential for human-like manipulation in robotics [52, 51, 42, 13, 3, 11, 32, 31, 40, 35, 44, 34]. It is of high potential yet very challenging due to its high dexterity. Dexterous grasping is a topic of much interest in this field. Some works [5, 14, 3] have leveraged analytical methods to model the kinematics and dynamics of both hands and objects, but they typically require simplifications, such as using simple finger and object geometries, to ensure the feasibility of the planning process. Recent success has been shown in using reinforcement learning and imitation learning methods [8, 9, 48, 40, 26, 4, 57, 67]. While these works have shown encouraging results, they all suppose that the robot can get all the oracle states (e.g., object position, velocity) during training and testing. However, this state information can not be easily and accurately captured in the real world. To mitigate this issue, some works [37, 36, 45, 68] consider a more realistic setting with the robot proprioception and RGB image or 3D scene point cloud as the input of the policy which can be captured more easily in the real world. Our work is more related to the recently proposed work UniDexGrasp [68] which learns a vision-based policy over 3000+ different objects. In this paper, we propose a novel pipeline that significantly improves the performance and generalization of UniDexGrasp, namely UniDexGrasp++.

2.2. Vision-based Policy Learning

Extensive research has been conducted to explore the learning of policies from visual inputs [72, 29, 59, 70, 60, 21, 20, 19]. To ease the optimization and training process, some works have utilized a pre-trained vision model and frozen the backbone, as shown in works such as [56, 47, 55]. Others, such as [67, 66], have employed multi-stage training. Our work is more related to [8, 7, 68], who firstly train a state-based policy and then distill to a vision-based policy. Also, our work makes good use of the pre-training of vision-backbone in the loop of imitation (supervised) learning and reinforcement learning which enables us to train a generalizable policy under the vision-based setting.

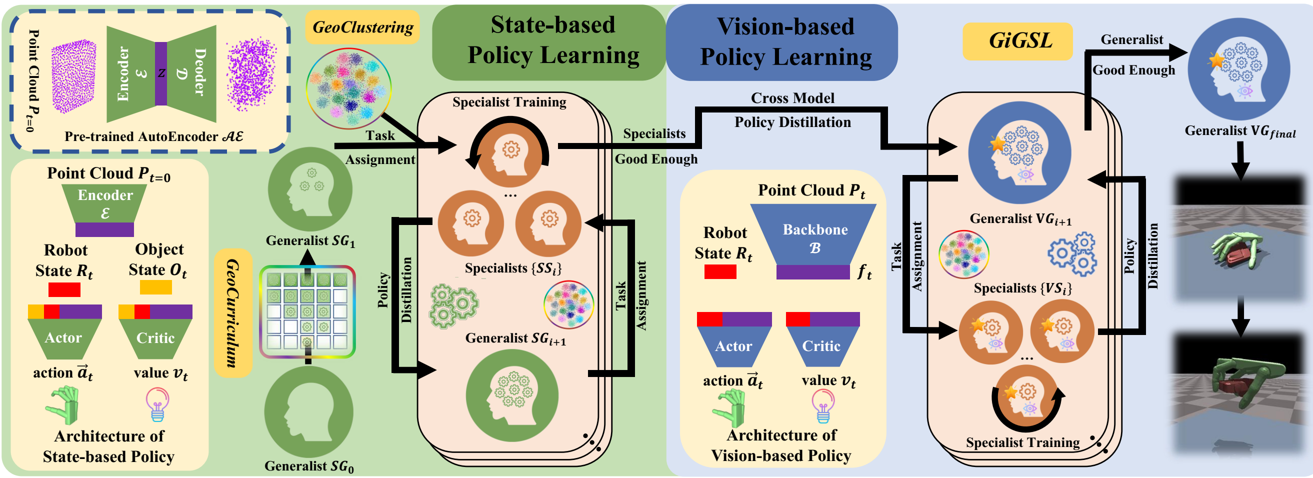

Figure 2: **Method Overview.** We propose to first adopt a state-based policy learning stage followed by a vision-based policy learning stage. The state-based policy takes input robot state R_t, object state S_t, and the geometric feature z of the scene point cloud of the first frame. We leverage a geometry-aware task curriculum (***GeoCurriculum***) to learn the first state-based generalist policy. After that, this generalist policy is further improved via iteratively performing specialist fine-tuning and distilling back to the generalist in our proposed geometry-aware iterative generalist-specialist learning (***GiGSL***), where the task assignment to which specialist is decided by our geometry-aware clustering (***GeoClustering***). For vision-based policy learning, we first distill the final state-based specialists to an initial vision-based generalist and then do *GiGSL* for the vision generalist, until we obtain the final vision-based generalist with the highest performance.

2.3. Generalization in Imitation Learning and Policy Distillation

To generalize to large environment variations (e.g., object geometry, task semantics) in policy learning, previous works have used imitation learning including behavior cloning [63, 30], augmenting demonstrations to Reinforcement Learning [48, 67, 46, 58, 15] and Inverse Reinforcement Learning [41, 1, 25, 18, 33] to utilize the expert demonstrations or policies. Some works [62, 38, 22, 28] have adopted the *Generalist-Speciliast Learning* idea in which a group of specialists (teacher) is trained on a subset of the task space, and then distill to a single generalist (student) in the whole task space using the above imitation learning and policy distillation methods. While these works have made great progress on several benchmarks [39, 71, 27, 64, 10], they either do not realize the importance of how to divide the tasks or environment variations for specialists or focus on a different setting to our method. In this work, we leverage the geometry feature of the task in the specialists' division and curriculum learning which greatly improves the generalizability.

3. Problem Formulation

In this work, we focus on learning a universal policy for dexterous object grasping from realistic point cloud observations and proprioceptive information under a table-top setting, similar to [68, 45].

We learn such a universal policy from a diverse set of grasping tasks. A grasping task is defined as $\tau = (o, R)$, where $o \in \mathbb{O}$ is an object instance from the object dataset $\mathbb{O}$, and $R \in \mathbf{SO(3)}$ is the initial 3D rotation of the object. To construct the environment, we randomly sample an object o, let it fall from a height, which randomly decides an initial pose, and then move the object center to the center of the table. We always initialize the dexterous hand at a fixed pose that is above the table center. The task is successful if the position difference between the object and the target is smaller than a threshold value. This is a multi-task policy learning setting and we require our learned policy to generalize well across diverse grasping tasks, *e.g.*, across random initial poses and thousands of objects including the unseen.

4. Method

This section presents a comprehensive description of our proposed method for solving complex tasks. In Sec. 4.1, we provide an overview of our approach along with the training pipeline. Our proposed method leverages DAgger-based distillation and iterative Generalist-Specialist Learning (iGSL) strategy, which is explained in detail in Sec. 4.2. Moreover, we introduce Geometry-aware Clustering to decide which specialist handles which task, achieving Geometry-aware iterative Generalist-Specialist Learning (GiGSL), which is presented in Sec. 4.3. In Sec. 4.4, we present a Geometry-aware Task Curriculum

Learning approach for training the first state-based generalist policy.

4.1. Method Overview

Following [68, 8, 7], we can divide our policy learning into two stages: 1) the state-based policy learning stage; 2) the vision-based policy learning stage. It is known that directly learning a vision-based policy is very challenging, we thus first learn a state-based policy that can access oracle information and let this policy help and ease the vision-based policy learning. The full pipeline is shown in Figure 2.

State-based policy learning stage. The goal of this stage is to obtain a universal policy, or we call it a *generalist*, that takes inputs from robot state R_t, object state O_t, and the scene point cloud $P_{t=0}$ at the first frame. Here the object point cloud is fused from multiple depth point clouds captured by multi-view depth cameras. And we include $P_{t=0}$ in the input to retain the scene geometry information and we use the encoder of a pre-trained point cloud autoencoder to extract its geometry feature. Note that at this point cloud encoder is frozen to make it as simple as possible, so it doesn't interfere with policy learning. We leave the visual processing of P_t to the vision-based policy.

Although learning a state-based policy through reinforcement learning is more manageable than learning a vision-based policy, it is still very challenging to achieve a high success rate under such a diverse multi-task setting. We thus propose a geometry-aware curriculum learning ***(GeoCurriculum)*** to ease the multi-task RL and improve the success rate.

After this ***GeoCurriculum***, we obtain the first state-based generalist SG_1 that can handle all tasks. We then propose a geometry-aware iterative Generalist-Specialist Learning strategy, dubbed as ***GiGSL***, to further improve the performance of the generalist. This process involves iterations between learning several state-based specialists $\{\mathrm{SS}_i\}$ that specialize in a specific range of tasks and distilling the specialists to a generalist SG_{i+1}, where i denotes the iteration index. The overall performance kept improving through this iterative learning until saturation.

Vision-based policy learning. For vision-based policy, we only allow it to access information available in the real world, including robot state R_t and the scene point clouds P_t. In this stage, we need to jointly learn a vision backbone $\mathcal{B}$ that extracts f_t from P_t along with our policy (see the blue part of Fig.2). Here we adopt PointNet+Transformer [39] as $\mathcal{B}$, which we find has a larger capacity and thus outperforms PointNet [43]. We randomly initialize the network weight of our first vision generalist VG_1. We start with performing a cross-modal distillation that distills the latest state-based specialists $\{\mathrm{SS}_n\}$ to VG_1. We can then start the *GiGSL* cycles for vision-based policies that iterate between finetuning $\{\mathrm{VS}_i\}$ and distilling to VG_{i+1} until the performance of the vision-based generalist saturates. The final vision-based generalist $\mathrm{VG}_{\text{final}}$ is our learned universal grasping policy that yields the highest performance. Please refer to supplementary material for the pseudo-code of the whole pipeline.

4.2. *iGSL*: iterative Generalist-Specialist Learning

Recap Generalist-Specialist Learning (*GSL*). The idea of Generalist-Specialist Learning comes from a series of works [62, 38, 22, 28] that deal with multi-task policy learning. The most recent paper [28] proposes *GSL*, a method that splits the whole task space into multiple subspaces and lets one specialist take charge of one subspace. Since each subspace has fewer task variations and thus is easier to learn, each specialist can be trained well and perform well on their task distributions. Finally, all the specialists will be distilled into one generalist.

Note that [28] only has one cycle of specialist learning and generalist learning. Straightforwardly, more cycles may be helpful. In *GSL*, the distillation is implemented using GAIL [25] or DAPG [48] but we find their performance mediocre. In this work, we propose a better policy distillation method based on DAgger, iteratively enabling Generalist-Specialist Learning.

Dagger-based policy distillation. DAgger [50] is an on-policy imitation learning algorithm. Different from GAIL or DAPG, which only require expert demonstrations, DAgger [50] requires an expert policy, which is called a teacher, and the student that takes charge of interacting with the environment. When the student takes action, the teacher policy will use its action to serve as supervision to improve the student. Given that the student always uses its policy to interact with the environment, such imitation is on-policy and thus doesn't suffer from the covariate shift problem usually seen in the behavior cloning algorithm. Previous works, such as [68] for dexterous grasping and [8, 7] for in-hand manipulation, have used DAgger for policy distillation from a state-based teacher to a vision-based student and it is shown in UniDexGrasp [68] that DAgger outperforms GAIL and DAPG for policy distillation.

However, one limitation of DAgger is that it only cares about the policy network and discards the value networks that popular actor-critic RL like PPO [54] and SAC [24] usually have. In this case, when a teacher comes with both an actor and a critic distills to a student, the student will only have an actor without a critic and thus can't be further fine-tuned using actor-critic RL. This limits *GSL* to simply one cycle and hinders it from further improving the generalist.

To mitigate this issue, we propose a new distillation method that jointly learns a critic function while learning the actor using DAgger. Our DAgger-based distillation learns both a policy and a critic function during the supervised policy distillation process, where the policy loss is the mean squared error (MSE) between the actions from the teacher policy $\pi_{teacher}$ and the student policy π_θ (same in DAgger), and the critic loss is the MSE between the predicted value function V_ϕ and the estimated returns $\hat{R}_t$ using Generalized Advantage Estimation (GAE) [53].

$$\mathcal{L} = \frac{1}{|D_{\pi_\theta}|}\sum_{\tau\in D_{\pi_\theta}}(\pi_{teacher}(s_t)-\pi_\theta(s_t))^2 + \frac{1}{|D_{\pi_\theta}|T}\sum_{\tau\in D_{\pi_\theta}}\sum_{t=0}^{T}(V_\phi(s_t)-\hat{R}_t)^2 \tag{1}$$

This DAgger-based distillation method allows us to retain both the actor and critic while achieving very high performance. Compared to ILAD [67] that only pre-trains the actor and directly finetunes the actor-critic RL (the critic network is trained from scratch), our method enables actor-critic RL to fine-tune on both trained actor and critic networks, enhancing the stability and effectiveness of RL training.

Iteration between specialist fine-tuning and generalist distillation. With our proposed DAgger-based distillation method, we can do the following: 1) start with our first generalist learned through *GeoCurriculum*; 2) clone the generalist to several specialists, finetune each specialist on their own task distribution; 3) using DAgger-based distillation method to distill all specialists to one generalist; we can iterate between 2) and 3) until the performance saturates.

4.3. *GiGSL*: Geometry-aware iterative Generalist-Specialist Learning

One important question left for *iGSL* is how to partition the task space. In [28], they are dealing with a limited amount of tasks and it is possible for them to assign one specialist to one task or randomly. However, in our work, we are dealing with an infinite number of tasks considering the initial object pose can change continuously. We can only afford a finite number of specialists and need to find a way to assign a sampled task to a specialist. We argue that similar tasks need to be assigned to the same specialist since one specialist will improve effectively via reinforcement learning only if its task variation is small. To this end, we propose ***GeoClustering***, a strategy for geometry-aware clustering in the task space.

***GeoClustering* strategy**. We split the task space $\mathbb{T} = \mathbb{O}\times \mathbf{SO(3)}$ into N_{clu} clusters, with tasks in each cluster C_j being handled by a designated specialist S_j during specialist fine-tuning. We begin by sampling a large number of tasks $\{\tau^{(k)}\}_{k=1}^{N_{\text{sample}}}$ from $\mathbb{T}$ ($N_{\text{sample}} \approx 270,000$ in our implementation) and clustering their visual features using K-Means. The clustering of the large-scale task samples provides an approximation of the clustering of the whole continuous task space.

We first train a point cloud 3D autoencoder using the point cloud $\{P_{t=0}^{(k)}\}_{k=1}^{N\text{sample}}$ of the initialized objects in the sample tasks $\{\tau^{(k)}\}_{k=1}^{N\text{sample}}$. The autoencoder follows an encoder-decoder structure. The encoder $\mathcal{E}$ encodes $P_{t=0}^{(k)}$ and outputs the encoding latent feature $z^{(k)} = \mathcal{E}(P_{t=0}^{(k)})$. The decoder $\mathcal{D}$ takes $z^{(k)}$ as input and generates the point cloud $\hat{P}_{t=0}^{(k)}$. The model is trained using the reconstruction loss $\mathcal{L}_{\text{AE}}$, which is the Chamfer Distance between $P_{t=0}^{(k)}$ and $\hat{P}_{t=0}^{(k)}$. See Supplementary Materials for more details.

During clustering for the state-based specialists, we use the pre-trained encoder $\mathcal{E}$ to encode the object point cloud $P_{t=0}^{(k)}$ for a task $\tau^{(k)}$ and obtain the latent code $z^{(k)}$. We use this geometry and pose encoded latent code $z^{(k)}$ as the feature for clustering. We then use K-Means to cluster the features of these sampled tasks $\{z^{(k)}\}_{k=1}^{N_{\text{sample}}}$ and generate N_{clu} clusters and corresponding cluster centers $\{c_j\}_{j=1}^{N_{\text{clu}}}$:

And for vision-based specialists, thanks to the trained vision backbone, we directly use it to generate feature $f^{(k)}$ to replace the corresponding encoding feature $z^{(k)}$ in the state-based setting. Finally, the clustering for specialists can be formulated as:

During the specialists fine-tuning, we assign a given task $\tau^{(k)}$ to the specialist in an online fashion to handle the infinite task space. During fine-tuning, we assign $\tau^{(k)}$ to SS_j or VS_j if the Specialist have the nearest center c_j to the feature $z^{(k)}$. or $f^{(k)}$. Then each Specialist only needs to train on the assigned task set and distill their learned specific knowledge to the Generalist.

Summary and Discussion. ***GeoClustering*** strategy resolves the problem of task space partition, allows one specialist to focus on concentrated task distribution, and thus facilitates the performance gain for each specialist. Please refer to Algorithm 1 for the pseudo-code of *GeoClustering*.

As a way to partition task space, our geometry-aware clustering is much more reasonable and effective than category label-based partition, based on the following reasons: 1) not every object instance has a category label; 2) considering the large intra-category geometry variations, not necessarily objects that belong to the same category would be taken care by the same specialist; 3) object pose can also affect grasping, which is completely ignored in category label based partition but is well captured by our method.

4.4. *GeoCurriculum*: Geometry-aware Task Curriculum Learning

Problems of *GiGSL* from Scratch For state-based policy learning, we in theory can start *GiGSL* from scratch. One straightforward way is to directly learn a generalist from scratch on the whole task space and then improve it following G-S-G-S-... steps. However, learning this first generalist directly on the whole task space using reinforcement learning would be very challenging, usually yielding a generalist

Algorithm 1 ***GeoClustering***

Require: Task Space $\mathbb{T}$, Encoder $\mathcal{E}$ from the pre-trained AutoEncoder or backbone $\mathcal{B}$ from the Vision Policy. Number of target clusters N_{clu}

Sample N_{sample} tasks $\{\tau^{(k)}\}_{j=1}^{N_{\text{sample}}}$ from $\mathbb{T}$

Get features:

  state-based: $\{z^{(k)}\}_{k=1}^{N_{\text{sample}}} \leftarrow \{\mathcal{E}(P_{t=0}^{(k)})\}_{k=1}^{N_{\text{sample}}}$

  vision-based: $\{f^{(k)}\}_{k=1}^{N_{\text{sample}}} \leftarrow \{\mathcal{B}(P_{t=0}^{(k)})\}_{k=1}^{N_{\text{sample}}}$

Get cluster centers using K-Means:

  state-based: $\{c_j\}_{j=1}^{N_{\text{clu}}} \leftarrow$ K-Means$(\{z^{(k)}\})$

  vision-based: $\{c_j\}_{j=1}^{N_{\text{clu}}} \leftarrow$ K-Means$(\{f^{(k)}\})$

return Cluster centers $\{c_j\}_{j=1}^{N_{\text{clu}}}$

Algorithm 2 ***GeoCurriculum***

Require: Task Space $\mathbb{T}$, N_{train} tasks for training $\{\tau^{(k)}\}_{k=1}^{N_{\text{train}}} \subset \mathbb{T}$, N_{level} hierarchical levels of curriculum learning and N_{sub} sub-clusters for each level, Encoder $\mathcal{E}$ from the pre-trained AutoEncoder

Get features from the encoder: $\{z^{(k)}\}_{k=1}^{N_{\text{train}}}$

Level 0: Find the center of the feature space $z_c \leftarrow$ ***GeoClustering***$(N_{\text{clu}} = 1)$ and the task τ_c with features nearest to z_c, train $\mathcal{C}_0 = \{\tau_c\}$ (where $\|\mathcal{C}_0\| = 1$).

for Level l in $1, \ldots, N_{\text{level}} - 1$, **do**

  Level l: Split each cluster of the Level $l-1$ into N_{sub} sub-clusters. Find the N_{sub} tasks with features nearest to each sub-cluster feature center and add these tasks to $\mathcal{C}_l$, train $\mathcal{C}_l$ (where $\|\mathcal{C}_l\| = N_{\text{sub}}^l$).

end for

Level N_{level}: train $\mathcal{C}_{N_{\text{level}}} = \{\tau^{(k)}\}_{k=1}^{N_{\text{train}}}$ (where $\|\mathcal{C}_{N_{\text{level}}}\| = N_{\text{train}}$)

return $\{\mathcal{C}_l\}_{l=0}^{N_{\text{level}}}$

with an unsatisfactory success rate.

An alternative would be to first learn N_{clu} specialist, distill to a generalist, and then follow S-G-S-G-... steps. However, this is still very suboptimal. Given the huge variations in our task space, we need $N_{\text{clu}} >> 1$ so that the task variations in each specialist are small enough to allow them effectively learn. This large number of specialists would be very costly for training. Furthermore, because each specialist is trained separately from scratch, their policy can be substantially different from each other, which may lead to new problems. Considering two tasks that are similar but assigned to different specialists (they are just around the boundary of the task subspace). Then, since the two specialists are trained independently, there is no guarantee that the specialists will do similar things to these two similar tasks, which means the policy is discontinuous around the subspace boundary. During policy distillation, a generalist may get significantly different action supervision from different specialists for those "boundary tasks". As a result, this discontinuity in policy may lead to difficulty in convergence and hurt the policy generalization toward unseen tasks.

Recap Object Curriculum in UniDexGrasp Following UniDexGrasp [68], we consider leveraging curriculum learning to make the first generalist learning easier. [68] introduced an object curriculum: they start with training a policy using RL to grasp one object instance (this object may be in the different initial poses); once this policy is well trained, they increase the number of objects by incorporating several similar objects from the same category and then finetuning the policy using RL on the new collection of objects; then, they increase the number of objects again by taking all objects from the category and finetune the policy; finally, they expand the object range to all different kinds of categories in the whole training objects and finish the final fine-tuning. [68] shows that object curriculum is crucial to their performance, improving the success rate of their state-based policy from 31% to 74% on training set.

GeoCurriculum. One fundamental limitation in the object curriculum used in [68] is unawareness of object pose and reliance on category labels. Similar to our argument in the discussion of Sec.4.3, we propose to leverage geometric features to measure the similarity between tasks, rather than object identity and category label. We thus introduce *GeoCurriculum*, a geometry-aware task curriculum that leverages hierarchical task space partition.

In detail, we design a N_{level} task curriculum that assigns tasks with increasing level of variations to policy learning and facilitate a step by step learning. As shown in Algorithm 2, we first find a task $\tau^{(k_c)}$ with the feature nearest to the feature center of all sampled tasks and train the policy (Level 0). Then iteratively, for level l, we split each cluster in the previous level $l-1$ into N_{sub} sub-clusters (30 in our implementation) based on geometry feature $z^{(k)}$ and find N_{sub} corresponding centers. We then add tasks that have features nearest to these sub-centers to the currently assigned tasks $\mathcal{C}_{i-1}$. Finally, we get the hierarchical task groups for the curriculum, that is:

$$\{\mathcal{C}_l\}_{l=0}^{N_{\text{level}}} = \boldsymbol{\mathit{GeoCurriculum}}(\mathbb{T}) \tag{2}$$

During training, we iteratively train the policy under each assigned task set. From tackling only one task in $\mathcal{C}_0$ to all the training tasks in $\mathcal{C}_{N_{\text{level}}}$, the policy grows up step by step and have better performance than directly training it under all tasks.

5. Experiment

5.1. Experiment Setting

We evaluate the effectiveness of our method in the challenging dexterous grasping benchmark UniDexGrasp [68] which is a recently proposed benchmark suite designated for learning generalizable dexterous grasping.

UniDexGarsp contains 3165 different object instances spanning 133 categories. Since the ground-truth grasp

Model	Train(%)	Test(%)	
		Uns. Obj. Seen Cat.	Uns. Cat.
PPO[54]	24.3±2.7	20.9±2.0	17.2±1.8
DAPG[48]	20.8±3.2	15.3±1.9	11.1±1.3
ILAD[67]	31.9±2.6	26.4±2.0	23.1±1.4
GSL[28]	57.3±3.9	54.1±3.4	50.9± 4.1
UniDexGrasp[68]	79.4±5.2	74.3±4.0	70.8±4.1
Ours (state-based)	**87.9±4.9**	**84.3±4.6**	**83.1±4.4**
PPO[54]+DAgger[50]	20.6±2.6	17.2±3.0	15.0±2.3
DAPG[48]+DAgger	17.9±2.8	15.2±3.2	13.9±2.4
ILAD[67]+DAgger	27.6±4.0	23.2±3.1	20.0±2.9
GSL[28]+DAgger	54.1±3.6	50.2±4.7	44.8±4.1
UniDexGrasp[68]	73.7±5.0	68.6±4.1	65.1±3.8
Ours (state)+DAgger	77.4±4.9	72.6±5.1	68.8±4.4
Ours (vision-based)	**85.4±5.6**	**79.67±4.4**	**76.7±4.9**

Table 1: **The Average Success Rate of the Evaluated Objects on Both Training and Test Set**. For better clarity, we use green for the state-based policy and blue for the vision-based policy.

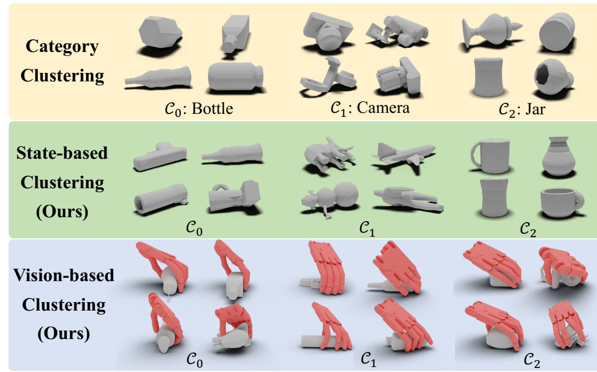

Figure 3: **Comparison between Category-label-based Clustering and our Geometry-aware Clustering.** Our state-based clustering is based on the features of the first-frame point clouds from the pre-trained encoder, while the vision-based policy utilizes its vision backbone to extract features for clustering. Due to the vision-based clustering being task-aware, we also show the grasping poses of the dexterous hands in the third row.

pose generation for pretraining and point cloud rendering processes are very expensive for UniDexGrasp environments, we only consider the non-goal conditioned setting in UniDexGarsp which does not specify the grasping hand pose. Each environment is randomly initialized with one object and its initial pose, and the environment consists of a panoramic 3D point cloud P_t captured from the fixed cameras for vision-based policy learning.

For the network architecture, we use MLP with 4 hidden layers (1024,1024,512,512) for the policy network and value network in the state-based setting, and an additional PointNet+Transformer [39] to encode the 3D scene point cloud input in the vision-based setting. We freeze the vision backbone during the vision-based specialist training. We use $K = N_{clu} = 20$ in our experiments. Other detailed hyperparameters are shown in supplementary materials.

5.2. Main Results

We first train our method in the state-based policy learning setting and compare it with several baselines (green part in Tab.1). We use PPO [54] for the specialist RL in our pipeline. For these baselines: PPO [54] is a popular RL method, DAPG [49], and ILAD [67] are imitation learning methods that further leverage expert demonstrations with RL; GSL [28] adopts the idea of generalist-specialist learning which use PPO for specialist learning and integrates demonstrations for generalist learning using DAPG, but with a random division for each specialist and only performs policy distillation once. UniDexGrasp [68] uses PPO and category-based object curriculum learning. To compare our method to these baselines, we distill our final state-based specialists $\{\text{SS}_n\}$ to a state-based generalist SG_{n+1} (although we won't use the latter later). With our proposed techniques, our method achieves a success rate of 88% and 84% on the train and test set, which is **9%** and **11%** improvement over the UniDexGrasp in the state-based setting.

We then compare our method in the vision-based policy learning setting with the baseline methods(blue part in Tab.1). For PPO [54], DAPG [49], ILAD [67] and GSL [28], we distill the state-based policy to the vision-based policy using DAgger [50] since they don't consider the observation space change (state to vision) and directly training these methods under a vision input leads to completely fail. For our method, we compare our proposed whole pipeline "Ours (vision-based)" with the variant of directly distilling our state-based policy to vision-based policy using DAgger, namely "Ours (state)+DAgger". Our final results in the vision-based setting reach 85% and 78% on the train set and test set which outperforms the SOTA baseline UniDexGrasp for **12%** and **11%**, respectively.

5.3. Analysis of the Training Process

Geometry-aware Clustering Helps the Policy Learning. We visualize some qualitative result in Fig.3. The first row shows a simple way of clustering, which is based on the object category. But as we analyzed above, this clustering method has no object geometry information and thus has limited help in grasping learning. The second row shows our stated-based clustering strategy, which is based on the features from the point cloud encoder $\mathcal{E}$ and can cluster objects with similar shapes. And furthermore, in the third row,

		Techniques						Success Rate (%)		
		Geo-Aware Curri.	Iterative Fine-tuning$_{\text{S}}$	Geo-Aware Clustering	Iterative Fine-tuning$_{\text{V}}$	End2End Distillation	Transformer Backbone	Training	Test Uns. Obj.	Test Uns. Cat.
State -based	1							79.4	74.3	70.8
	2	✓						82.7	76.8	74.2
	3	✓	✓					84.0	77.9	74.8
	4	✓	✓	✓				**87.9**	**84.3**	**83.1**
Vision -based	5	✓	✓					77.4	72.6	68.8
	6	✓	✓			✓		78.0	72.1	69.1
	7	✓	✓		✓	✓		78.9	74.7	70.2
	8	✓	✓	✓	✓	✓		82.1	77.1	71.9
	9	✓	✓	✓	✓		✓	82.7	76.2	73.4
	10	✓	✓	✓		✓	✓	82.5	76.1	72.0
	11	✓	✓		✓	✓	✓	78.6	73.7	72.3
	12	✓	✓	✓	✓	✓	✓	**85.4**	**79.6**	**76.7**

Table 2: **Ablation Study.** For state-based policy (green) and vision-based policy learning (blue), we compare our techniques with various ablations.

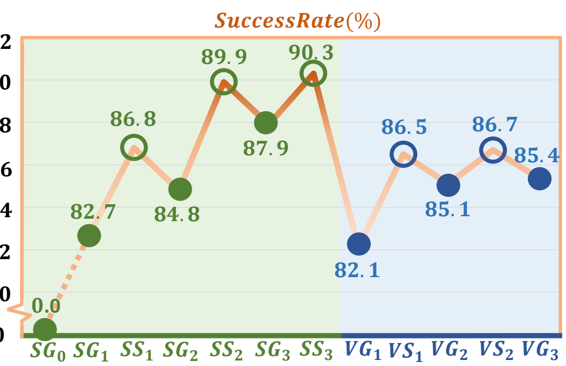

Figure 4: **Success Rate during our *GiGSL* Training.** We plot the success rate of each training step, where green represents the state-based policy, blue represents the vision-based policy, hollow points represent the specialist policy, and solid points represent the generalist policy.

our vision-based clustering strategy utilizes the vision backbone which has more task-relative information, and thus the clustered objects have similar shapes as well as similar grasping poses.

Quatitative Performance Improvement of our *GiGSL*. We visualize the success rate of each learning or fine-tuning step in Fig.4. No matter whether for state-based or vision-based policy, the improvement of Generalist-Specialist fine-tuning and distillation shows the effectiveness of our Geometry-aware iterative Generalist-Specialist Learning ***GiGSL*** strategy design and boosts the final performance of Universal Dexterous Grasping.

5.4. Ablation Study

The ablation studies are shown in Tab.2. For **state-based policy learning stage** (green part), we analyze the ablation results as follows.

1) (Row 1,2) Effective of *GeoCurriculum*. Using our proposed *GeoCurriculum* (Row 2) performs better than using object-curriculum-learning in [68] (Row 1).

2) (Row 2,3) Effective of iterative fine-tuning$_{\text{S}}$. The policy can benefit from the iterative fine-tuning process and reach a higher success rate on both training and test set (Row 3) than a single cycle (Row 2). Also see Figure 4.

3) (Row 3,4) Effective of *GeoClustering* in the state-based setting. With the pre-trained visual feature, the tasks assigned to one specialist are around the same feature clusters and thus are similar to each other. This significantly reduces the difficulty of policy learning and, in return, improves performance (Row 4), compared to randomly assigning tasks to the specialists (Row 3).

For the ablation studies of **vision-based policy learning stage** (blue part), we use *GeoClustering* in the state-based policy training by default, and the checkmark of *GeoClustering* in this part indicates whether we use it in the vision-based policy learning. We use PointNet [43] if there's no checkmark in "Transformer Backbone".

4) (Row 5,6 & 9,12) Effective of end-to-end distillation. We find directly distilling the final state-based specialists $\{\text{SS}_n\}$ to the vision-based generalist VG_1 (Row 6, 12) performs better than first distilling the state-based specialists $\{\text{SS}_n\}$ to the state-based generalist SG_{n+1}, then distilling this generalist to vision-based generalist VG_1 (Row 5, 9).

5) (Row 6,7 & 10,12) Effective of iterative fine-tuning$_{\text{V}}$. The policy can benefit from the iterative fine-tuning process and reach a higher success rate on both training and test set

than the single stage. Also see Figure 4.

6) (Row 7,8 & 11,12) Effective of *GeoClustering* in the vision-based setting. By dividing the specialists using the learned visual feature from the vision backbone of the generalist, the final performance can be significantly improved than randomly dividing the specialists (**7%** and **5%** on training and test set, comparing Row 11 and 12).

7) (Row 8,12) Effective of the Transformer backbone. The results show that the PointNet+Transformer backbone [39] (Row 13) has a better expressive capacity which can improve the performance of DAgger-based distillation than using the PointNet [43] backbone (Row 8).

5.5. Failure Cases and Limitation.

Within training object distribution, the two major types of failure cases are dropping during lifting (**(a)** in Fig.5) and moving during grasping **(b)**. These objects require the robotic hand to delicately and precisely establish contact with the objects and thus places higher demands on the effectiveness and robustness of our algorithm. The limitation of our method (also for UniDexGrasp [68]) is mainly in grasping thin objects, such as iPads and cards **(c)**, and large-scale objects **(d)**. It is noteworthy that even training a state-based RL policy on some of these objects individually would fail. These challenging cases may require pushing and then grasping which is beyond the scope of this paper.

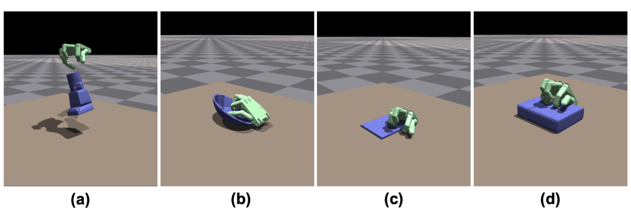

Figure 5: **Failure cases analysis.**

5.6. Real-world experiments.

Due to the limitation of our resources, we can only leverage Allegro Hand for real-world grasping. We first exactly follow our training pipeline to train a grasping policy using Allegro in the simulator. Given the observation, we use the learned policy to generate a trajectory in the simulator and then use the controller to follow this trajectory in the real world. We ascertained that our policy can be successfully transitioned from simulation to reality, and it effectively facilitates the grasping of the target object. The successful execution of our grasping operation is depicted in Fig. 6.

6. Conclusions and Discussions

In this paper, we propose a novel pipeline, UniDexGrasp++, that significantly improves the performance and

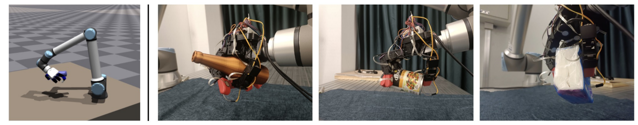

Figure 6: **Real-world experiments with Allegro Hand Left**: grasping in the simulator, **Right**: grasping in the real world.

generalization of UniDexGrasp. We believe such generalizability is also essential for Sim2Real transfer for real robot dexterous grasping. The limitation is that we only tackle the dexterous grasping task in simulation and we will conduct the real-robot extension in our future work.

Acknowledgments This work is partly sponsored by Beijing Municipal Science & Technology Commission (Project ID: Z221100003422004).

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004. 3

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 2

[3] Sheldon Andrews and Paul G Kry. Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics*, 37(7):830–839, 2013. 2

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2

[5] Yunfei Bai and C. Karen Liu. Dexterous manipulation using both palm and fingers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1560–1565, 2014. 2

[6] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. *arXiv preprint arXiv:2101.01132*, 2021. 1

[7] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand dexterous manipulation from depth. *arXiv preprint arXiv:2211.11744*, 2022. 2, 4

[8] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. *Conference on Robot Learning*, 2021. 2, 4

[9] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object in-

teractions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[10] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020. 3

[11] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014. 2

[12] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. *arXiv preprint arXiv:2210.06575*, 2022. 1

[13] Mehmet R Dogar and Siddhartha S Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130. IEEE, 2010. 2

[14] Mehmet R. Dogar and Siddhartha S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130, 2010. 2

[15] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016. 3

[16] Hongjie Fang, Hao-Shu Fang, Sheng Xu, and Cewu Lu. Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline. *IEEE Robotics and Automation Letters*, pages 1–8, 2022. 1

[17] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020. 1

[18] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017. 3

[19] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations, 2023. 2

[20] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. *arXiv preprint arXiv:2211.05272*, 2022. 2

[21] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. *arXiv preprint arXiv:2209.12941*, 2022. 2

[22] Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning. *arXiv preprint arXiv:1711.09874*, 2017. 2, 3, 4

[23] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, 2021. 1

[24] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018. 4

[25] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016. 3, 4

[26] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021. 2

[27] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 3

[28] Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning. In *International Conference on Machine Learning*, pages 10104–10119. PMLR, 2022. 2, 3, 4, 5, 7

[29] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018. 2

[30] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019. 3

[31] Vikash Kumar, Abhishek Gupta, Emanuel Todorov, and Sergey Levine. Learning dexterous manipulation policies from experience and imitation. *arXiv preprint arXiv:1611.05095*, 2016. 2

[32] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016. 2

[33] Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. *arXiv preprint arXiv:1911.10947*, 2019. 3

[34] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2

[35] Qingkai Lu, Kautilya Chenna, Balakumar Sundaralingam, and Tucker Hermans. Planning multi-fingered grasps as probabilistic inference in a learned deep network. In

Robotics Research: The 18th International Symposium ISRR, pages 455–472. Springer, 2020. 2

[36] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning (CoRL)*, 2021. 2

[37] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 2

[38] Tongzhou Mu, Jiayuan Gu, Zhiwei Jia, Hao Tang, and Hao Su. Refactoring policy for compositional generalizability using self-supervised object proposals. *Advances in Neural Information Processing Systems*, 33:8883–8894, 2020. 2, 3, 4

[39] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 2, 3, 4, 7, 9

[40] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020. 2

[41] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000. 3

[42] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 255–262. IEEE, 2000. 2

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 4, 8, 9

[44] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. *arXiv preprint arXiv:2210.04887*, 2022. 2

[45] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2211.09423*, 2022. 2, 3

[46] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7865–7871. IEEE, 2021. 3

[47] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022. 2

[48] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2, 3, 4, 7

[49] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 7

[50] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 2, 4, 7

[51] Daniela Rus. In-hand dexterous manipulation of piecewise-smooth 3-d objects. *The International Journal of Robotics Research*, 18(4):355–381, 1999. 2

[52] J Kenneth Salisbury and John J Craig. Articulated hands: Force control and kinematic issues. *The International journal of Robotics research*, 1(1):4–17, 1982. 2

[53] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 5

[54] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4, 7

[55] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. *arXiv preprint arXiv:2206.14244*, 2022. 2

[56] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR, 2022. 2

[57] Qijin She, Ruizhen Hu, Juzhan Xu, Min Liu, Kai Xu, and Hui Huang. Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. *arXiv preprint arXiv:2204.13998*, 2022. 2

[58] Hao Shen, Weikang Wan, and He Wang. Learning category-level generalizable object manipulation policy via generative adversarial self-imitation learning from demonstrations. *arXiv preprint arXiv:2203.02107*, 2022. 2, 3

[59] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020. 2

[60] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021. 2

[61] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. 1

[62] Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

[63] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018. 3

[64] Yusuke Urakami, Alec Hodgkinson, Casey Carlin, Randall Leu, Luca Rigazio, and Pieter Abbeel. Doorgym: A scalable door opening environment and baseline agent. *arXiv preprint arXiv:1908.01887*, 2019. 3

[65] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15964–15973, October 2021. 1

[66] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects. In *International Conference on Learning Representations*, 2022. 2

[67] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance. *arXiv preprint arXiv:2204.02320*, 2022. 2, 3, 5, 7

[68] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy, 2023. 2, 3, 4, 6, 7, 8, 9

[69] Zhenjia Xu, Zhanpeng He, and Shuran Song. Universal manipulation policy network for articulated objects. *IEEE Robotics and Automation Letters*, 7(2):2447–2454, 2022. 1

[70] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021. 2

[71] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 2, 3

[72] Fangyi Zhang, Jürgen Leitner, Michael Milford, Ben Upcroft, and Peter Corke. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv preprint arXiv:1511.03791*, 2015. 2