# UniGraspTransformer: Simplified Policy Distillation for Scalable Dexterous Robotic Grasping

# UniGraspTransformer:用于可扩展灵巧机器人抓取的简化策略蒸馏

Wenbo Wang Fangyun Wei Xiaohan Yi1 Yizhong Zhang Yaobo Liang1 Baining Guo

王文博 魏方云 易晓涵1 张一忠 梁耀波1 郭百宁

Microsoft Research Asia University of Sydney National University of Singapore

微软亚洲研究院 悉尼大学 新加坡国立大学

{rawe, xichen6, liluo, yixiaohan, yizzhan, yalia, yanlu, jiaoyan, bainguo}@microsoft.com

{rawe, xichen6, liluo, yixiaohan, yizzhan, yalia, yanlu, jiaoyan, bainguo}@microsoft.com

aaronwenbwang@gmail.com leizhou@u.nus.edu c.xu@sydney.edu.au

aaronwenbwang@gmail.com leizhou@u.nus.edu c.xu@sydney.edu.au

# Abstract

# 摘要

We introduce UniGraspTransformer, a universal Transformer-based network for dexterous robotic grasping that simplifies training while enhancing scalability and performance. Unlike prior methods such as UniDexGrasp++, which require complex, multi-step training pipelines, UniGraspTransformer follows a streamlined process: first, dedicated policy networks are trained for individual objects using reinforcement learning to generate successful grasp trajectories; then, these trajectories are distilled into a single, universal network. Our approach enables UniGraspTransformer to scale effectively, incorporating up to 12 self-attention blocks for handling thousands of objects with diverse poses. Additionally, it generalizes well to both idealized and real-world inputs, evaluated in state-based and vision-based settings. Notably, UniGraspTransformer generates a broader range of grasping poses for objects in various shapes and orientations, resulting in more diverse grasp strategies. Experimental results demonstrate significant improvements over state-of-the-art, UniDexGrasp++, across various object categories, achieving success rate gains of 3.5%, 7.7%, and 10.1% on seen objects, unseen objects within seen categories, and completely unseen objects, respectively, in the vision-based setting. Project page: https://dexhand.github.io/UniGraspTransformer/.

我们介绍了UniGraspTransformer，一种基于Transformer的通用网络，用于灵巧机器人抓取，简化了训练过程，同时提高了可扩展性和性能。与之前的方法(如UniDexGrasp++)不同，这些方法需要复杂的多步骤训练流程，而UniGraspTransformer遵循一个简化的过程:首先，使用强化学习为每个物体训练专用的策略网络，以生成成功的抓取轨迹；然后，将这些轨迹蒸馏到一个单一的通用网络中。我们的方法使UniGraspTransformer能够有效地扩展，包含多达12个自注意力块，以处理数千个具有不同姿态的物体。此外，它在理想化和现实世界的输入中都能很好地泛化，在基于状态和基于视觉的设置中进行了评估。值得注意的是，UniGraspTransformer为各种形状和方向的物体生成了更广泛的抓取姿态，从而产生了更多样化的抓取策略。实验结果表明，与最先进的UniDexGrasp++相比，在各种物体类别中都有显著改进，在基于视觉的设置中，对已见物体、已见类别中的未见物体和完全未见物体的成功率分别提高了3.5%、7.7%和10.1%。项目页面:https://dexhand.github.io/UniGraspTransformer/。

# 1. Introduction

# 1. 引言

Dexterous robotic grasping remains a formidable challenge in the field of robotics, especially when dealing with objects that exhibit a wide variety of shapes, sizes, and physical properties. Dexterous hands [12, 44], with their multiple degrees of freedom and complex control requirements, present unique difficulties in manipulation tasks. While methods such as UniDexGrasp++ [50] have made notable progress in this area, they encounter significant performance degradation when a single network is tasked with a large and diverse set of objects. Additionally, UniDexGrasp++ [50] employs a multifaceted training process, including policy learning, geometry-aware clustering, curriculum learning, and policy distillation, which complicates scaling and reduces efficiency.

灵巧机器人抓取 仍然是机器人领域中的一个巨大挑战，尤其是在处理具有各种形状、大小和物理特性的物体时。灵巧手 [12, 44] 由于其多个自由度和复杂的控制需求，在操作任务中呈现出独特的困难。尽管UniDexGrasp++ [50] 等方法在这一领域取得了显著进展，但当单个网络需要处理大量且多样化的物体时，它们会遇到显著的性能下降。此外，UniDexGrasp++ [50] 采用了多方面的训练过程，包括策略学习、几何感知聚类、课程学习和策略蒸馏，这使扩展变得复杂并降低了效率。



Figure 1. Performance comparison among UniDexGrasp [57], UniDexGrasp++ [50] and our UniGraspTransformer, across state-based and vision-based settings. For each setting, success rates are evaluated on seen objects, unseen objects within seen categories, and entirely unseen objects from unseen categories.

图1. UniDexGrasp [57]、UniDexGrasp++ [50] 和我们的UniGraspTransformer在基于状态和基于视觉的设置中的性能比较。对于每种设置，成功率在已见物体、已见类别中的未见物体和完全未见类别中的未见物体上进行了评估。

In this work, we simplify the training process of a universal network capable of handling thousands of objects while simultaneously improving both performance and generalizability. The workflow we propose is straightforward: 1) For each object in the training set, we begin by training a dedicated policy network using reinforcement learning, guided by carefully crafted reward functions that enable the robot to master object-specific grasping strategies; 2) Next, these well-trained policy networks are used to generate millions of successful grasp trajectories; 3) Finally, we train a universal Transformer-based network, namely UniGrasp-Transformer, in a supervised manner on this extensive trajectory set, allowing the network to generalize effectively to both the objects seen during training and new, unseen objects. Our architecture offers four key advantages:

在这项工作中，我们简化了能够处理数千个物体的通用网络的训练过程，同时提高了性能和泛化能力。我们提出的工作流程很简单:1)对于训练集中的每个物体，我们首先使用强化学习训练一个专用的策略网络，通过精心设计的奖励函数引导机器人掌握特定物体的抓取策略；2)接下来，这些训练有素的策略网络用于生成数百万条成功的抓取轨迹；3)最后，我们在这个广泛的轨迹集上以监督的方式训练一个基于Transformer的通用网络，即UniGraspTransformer，使网络能够有效地泛化到训练期间见过的物体和新的未见物体。我们的架构提供了四个关键优势:

[[1]](#footnote-29)

* Simplicity. We directly distill all dedicated reinforcement learning policies into a universal network in an offline style, without utilizing any extra techniques like network regularization or progressive distillation [9, 13, 50].
* 简单性。我们直接以离线方式将所有专用的强化学习策略蒸馏到一个通用网络中，而不使用任何额外的技术，如网络正则化或渐进蒸馏 [9, 13, 50]。
* Scalability. Larger grasping networks generally demonstrate the ability to handle a broader range of objects and exhibit greater robustness to variations in shape and size. Our approach, which leverages offline distillation, allows the final network, UniGraspTransformer, to be designed at a larger scale, accommodating up to 12 self-attention blocks [49]. This provides significant flexibility and capacity compared to traditional online distillation methods [50, 57], which often rely on smaller MLP networks to ensure convergence but limit scalability. Additionally, our dedicated policy networks are intentionally lightweight, as each only needs to handle a single object, ensuring efficiency without compromising performance.
* 可扩展性。较大的抓取网络通常表现出处理更广泛物体的能力，并对形状和大小的变化表现出更强的鲁棒性。我们的方法利用离线蒸馏，使最终的网络UniGraspTransformer能够设计得更大，容纳多达12个自注意力块 [49]。与传统的在线蒸馏方法 [50, 57] 相比，这提供了显著的灵活性和容量，后者通常依赖较小的MLP网络以确保收敛，但限制了可扩展性。此外，我们的专用策略网络特意设计得轻量级，因为每个网络只需要处理单个物体，确保效率而不影响性能。
* Flexibility. Each dedicated policy network is trained in a controlled, idealized environment where the full state of the system, including object representations (e.g., complete point clouds), dexterous hand states (e.g., finger-joint angles), and their interactions (e.g., hand-object distance), is fully observable and precisely accurate. Our architecture enables the distillation of knowledge from this ideal setting to more practical, real-world environments where some observations may be incomplete or unreliable [7, 11, 15, 37, 50, 57]. For instance, object point clouds might be noisy, or measurements of object poses may be imprecise. The primary role of these dedicated policy networks is to generate diverse, successful grasping trajectories across a wide range of objects. During the distillation process, these grasp trajectories serve as annotated data, enabling us to train our UniGraspTrans-former model using realistic inputs (e.g., noisy object point clouds and estimated object poses) to predict action sequences that closely mimic the successful grasp trajectories from the ideal setting.
* 灵活性。每个专用策略网络都在一个受控的理想化环境中进行训练，其中系统的完整状态，包括物体表示(例如完整的点云)、灵巧手状态(例如手指关节角度)以及它们的交互(例如手与物体的距离)，都是完全可观察且精确的。我们的架构使得能够从这种理想环境中提取知识，并将其应用于更实际的现实世界环境中，在这些环境中，某些观察可能不完整或不可靠[7, 11, 15, 37, 50, 57]。例如，物体的点云可能带有噪声，或者物体姿态的测量可能不精确。这些专用策略网络的主要作用是生成针对各种物体的多样化、成功的抓取轨迹。在蒸馏过程中，这些抓取轨迹作为标注数据，使我们能够使用真实的输入(例如带有噪声的物体点云和估计的物体姿态)来训练我们的UniGraspTransformer模型，以预测与理想环境中的成功抓取轨迹非常接近的动作序列。
* Diversity. In addition to being capable of grasping thousands of distinct objects, our larger universal network, coupled with the offline distillation strategy, demonstrates the ability to generate a broader range of grasping poses for objects presented in various orientations. This marks a significant improvement over prior methods, such as UniDexGrasp++ [50], which tend to produce repetitive, monotonous grasping poses across different objects.
* 多样性。除了能够抓取数千种不同的物体外，我们更大的通用网络结合离线蒸馏策略，展示了为以各种方向呈现的物体生成更广泛的抓取姿态的能力。这标志着相对于之前的方法(如UniDexGrasp++ [50])的显著改进，后者往往在不同物体上产生重复、单调的抓取姿态。

In our experiments, our proposed approach demonstrates substantial improvements over the previous state-of-the-art, UniDexGrasp++ [50], across a range of evaluation settings. Specifically, we evaluate our method in both the state-based setting-where object observations and dexterous hand states are perfectly accurate, as provided by a simulator-and the vision-based setting, where object point clouds are derived from multi-view reconstruction. Our approach consistently outperforms UniDexGrasp++ [50] across multiple object types, including seen objects, unseen objects within seen categories, and entirely unseen objects from unseen categories, as illustrated in Figure 1. For instance, our method achieves performance gains of , and 10.1% over UniDexGrasp++ [50] on seen objects, unseen objects within seen categories, and entirely unseen objects from unseen categories, under the vision-based setting.

在我们的实验中，我们提出的方法在一系列评估设置中展示了相对于之前最先进的UniDexGrasp++ [50]的显著改进。具体来说，我们在基于状态的设置(其中物体观察和灵巧手状态完全准确，由模拟器提供)和基于视觉的设置(其中物体点云来自多视图重建)中评估了我们的方法。我们的方法在多种物体类型上始终优于UniDexGrasp++ [50]，包括已见物体、已见类别中的未见物体以及完全未见类别中的未见物体，如图1所示。例如，在基于视觉的设置下，我们的方法在已见物体、已见类别中的未见物体以及完全未见类别中的未见物体上分别比UniDexGrasp++ [50]提高了 和10.1%的性能。

# 2. Related Works

# 2. 相关工作

Robotic Grasping. Robotic grasping [16, 18, 24, 58] has been a longstanding research in robotics and computer vision, aiming to enable robots to interact with objects reliably and adaptively. Although significant advances have been made in gripper-based robotic grasping [3, 6, 25, 26, 53, 62], the limited complexity of gripper structures restricts their adaptability to objects with intricate geometries.

机器人抓取。机器人抓取[16, 18, 24, 58]一直是机器人和计算机视觉领域的一个长期研究课题，旨在使机器人能够可靠且自适应地与物体交互。尽管在基于夹爪的机器人抓取方面取得了显著进展[3, 6, 25, 26, 53, 62]，但夹爪结构的有限复杂性限制了它们对具有复杂几何形状的物体的适应性。

Dexterous grasping [19, 22, 33, 34, 50, 52, 54, 55, 57] introduces advanced multi-fingered manipulation, enabling more versatile grasps for objects of diverse shapes. However, controlling the highly dexterous multi-fingered system poses significant challenges for traditional analytical techniques [1, 21, 47, 52]. Recent advances have utilized learning-based methods to enable effective dexterous manipulation. One approach decomposes the grasping process into two stages: generating a static grasp pose and then performing a dynamic grasping through trajectory planning or goal-conditioned reinforcement learning [2, 4, 14, 20, 45, 56, 57, 61]. Although promising diversity, the generated static grasp poses are often not validated in dynamic settings, which adversely affects the overall success. Alternatively, another line of approach directly learns the entire grasping process through expert demonstrations from humans or reinforcement learning agents . While effective, these approaches either require complex training pipelines or suffer significant performance degradation when a single policy is applied across a broad range of objects, due to the limited number of training objects and expert demonstrations, as well as the constrained capacity of their policy networks.

灵巧抓取[19, 22, 33, 34, 50, 52, 54, 55, 57]引入了先进的多指操作，使得能够对形状多样的物体进行更灵活的抓取。然而，控制高度灵巧的多指系统对传统的分析技术提出了重大挑战[1, 21, 47, 52]。最近的进展利用基于学习的方法实现了有效的灵巧操作。一种方法将抓取过程分解为两个阶段:生成静态抓取姿态，然后通过轨迹规划或目标条件强化学习执行动态抓取[2, 4, 14, 20, 45, 56, 57, 61]。尽管在多样性方面表现出色，但生成的静态抓取姿态通常未在动态环境中进行验证，这影响了整体成功率。另一种方法通过人类或强化学习代理的专家示范直接学习整个抓取过程 。虽然有效，但这些方法要么需要复杂的训练流程，要么在将单一策略应用于广泛物体时由于训练物体和专家示范的数量有限以及策略网络的容量受限而出现显著的性能下降。

To overcome these limitations, we extend the latter approach by proposing a novel pipeline that integrates online reinforcement learning with large-model offline distillation, simplifying training while improving scalability and grasping performance.

为了克服这些限制，我们通过提出一种将在线强化学习与大模型离线蒸馏相结合的新颖流程，扩展了后一种方法，简化了训练，同时提高了可扩展性和抓取性能。

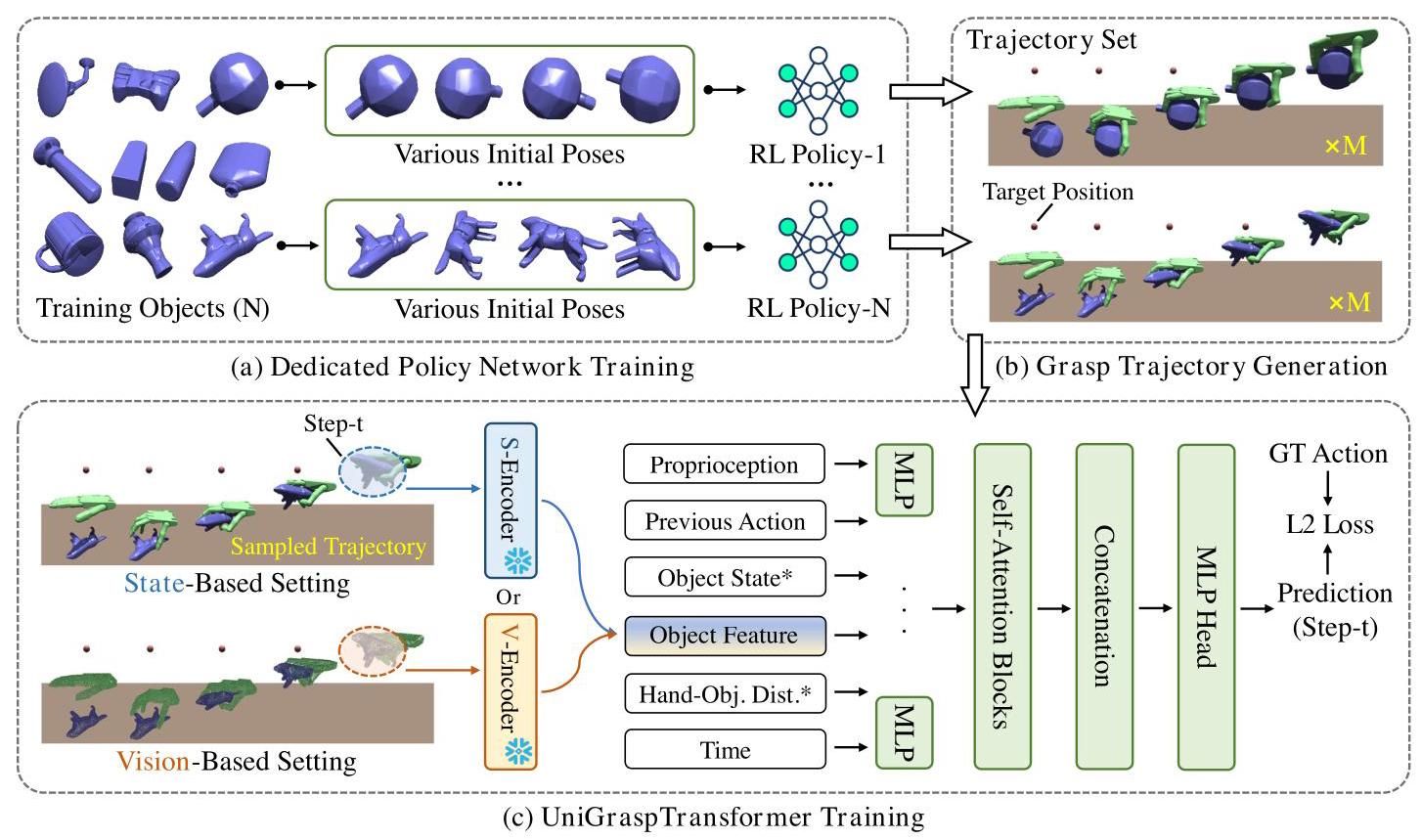


Figure 2. Overview of UniGraspTransformer. (a) Dedicated policy network training: each individual RL policy network is trained to grasp a specific object with various initial poses. (b) Grasp trajectory generation: each policy network generates successful grasp trajectories, forming a trajectory set . (c) UniGraspTransformer training: trajectories from are used to train UniGraspTransformer, a universal grasp network, in a supervised manner. We investigate two settings-state-based and vision-based-with the primary difference being in the input representation of object state and hand-object distance, as indicated by "\*" in the figure. The architecture of S-Encoder and V-Encoder can be found in Figure 3.

图2. UniGraspTransformer概述。(a) 专用策略网络训练:每个独立的RL策略网络被训练以抓取具有各种初始姿态的特定物体。(b) 抓取轨迹生成:每个策略网络生成 条成功的抓取轨迹，形成轨迹集 。(c) UniGraspTransformer训练:来自 的轨迹用于以监督方式训练UniGraspTransformer，这是一个通用抓取网络。我们研究了两种设置——基于状态的和基于视觉的——主要区别在于物体状态和手-物体距离的输入表示，如图中“\*”所示。S-Encoder和V-Encoder的架构可以在图3中找到。

Policy Distillation. Policy distillation [5, 10, 13, 35, 39, provides an effective approach for transferring knowledge from high-performance policies to a single universal policy, promoting both model compactness and generalization across diverse tasks.

策略蒸馏。策略蒸馏[5, 10, 13, 35, 39, 提供了一种有效的方法，将知识从高性能策略转移到单一通用策略，促进模型的紧凑性和跨多样化任务的泛化能力。

In robotics, recent works have focused on combining imitation learning and reinforcement learning [17, 29, 38, to enable student agents to learn from teacher demonstrations. This research generally follows two main approaches based on the source of demonstrations. The first approach directly trains the student policy on pre-collected human demonstrations, such as teleoperated human motions or recorded human videos [8, 17, 29, 38]. While effective, gathering extensive demonstrations can be costly, particularly for complex tasks like dexterous grasping with diverse objects in varied poses, limiting the student’s generalization capacity. The second approach generates demonstrations using pre-trained policies within the generalist-specialist learning framework [9, 13, 31, 46, 50, 51, 57]. Here, the task space is divided into sub-tasks, with reinforcement learning policies specialized and trained for each. These policies are then distilled into a single universal policy, enhancing the agent’s ability to generalize across the full task space. Despite the progress made, a single network handling a broad range of objects often experiences performance drops due to the limited teacher policies and the constrained capacity of student networks, which struggle to capture the complexity of the entire task space.

在机器人学中，最近的研究集中在结合模仿学习和强化学习[17, 29, 38, ，以使学生代理能够从教师示范中学习。这项研究通常遵循基于示范来源的两种主要方法。第一种方法直接在预收集的人类示范上训练学生策略，例如远程操作的人类动作或录制的人类视频[8, 17, 29, 38]。虽然有效，但收集大量示范可能成本高昂，特别是对于复杂任务，如抓取具有各种姿态的多样化物体的灵巧抓取，限制了学生的泛化能力。第二种方法在通用-专家学习框架内使用预训练策略生成示范[9, 13, 31, 46, 50, 51, 57]。在这里，任务空间被划分为子任务，强化学习策略专门化并针对每个子任务进行训练。然后，这些策略被蒸馏成单一通用策略，增强了代理在整个任务空间中的泛化能力。尽管取得了进展，但处理广泛范围物体的单一网络通常由于教师策略的有限性和学生网络容量的限制而经历性能下降，这些网络难以捕捉整个任务空间的复杂性。

Our approach addresses these challenges by initially training dedicated policies (i.e., teachers) for each object, resulting in a dataset of 3,200 objects and 3.2 million grasping trajectories. We then leverage UniGraspTransformer (i.e., student) to perform offline universal policy learning, utilizing up to 12 self-attention blocks to process diverse grasping trajectories and better preserve the diversity of teacher policies. This approach significantly enhances dexterous grasping performance across various settings.

我们的方法通过最初为每个物体训练专用策略(即教师)来解决这些挑战，生成了一个包含3,200个物体和320万条抓取轨迹的数据集。然后，我们利用UniGraspTransformer(即学生)进行离线通用策略学习，使用多达12个自注意力块来处理多样化的抓取轨迹，并更好地保留教师策略的多样性。这种方法显著提高了在各种设置下的灵巧抓取性能。

# 3. Methodology

# 3. 方法论

Problem Formulation. The objective is to train a robust universal network, UniGraspTransformer, that enables a dexterous, five-fingered robotic hand to grasp a variety of tabletop objects in diverse initial poses. Isaac Gym 3.0 [27] is utilized as our simulator.

问题表述。目标是训练一个鲁棒的通用网络UniGraspTransformer，使一个灵巧的五指机器人手能够抓取各种桌面物体，这些物体具有不同的初始姿态。Isaac Gym 3.0 [27]被用作我们的模拟器。

|  |  |
| --- | --- |
| Input Type | Elements (Dimension) |
| Proprioception (167) | Wrist position (3) and rotation (3); Finger-joint angle (22), angular velocity (22) and force (22); Finger-tip position , quaternion rotation , linear velocity , angular velocity , force and torque . |
| Previous Action (24) | Wrist force (3) and torque (3); Finger-joint angles (18). |
| Object State (16) | Object center (3), quaternion rotation (4), linear velocity (3), and angular velocity (3); Object-goal distance (3). |
| Hand-Object Distance (36) | Hand body points to object point cloud distances (36). |
| Time (29) | Current time (1); Sine-cosine time embedding (28). |

|  |  |
| --- | --- |
| 输入类型 | 元素(维度) |
| 本体感觉(167) | 手腕位置(3)和旋转(3)；手指关节角度(22)、角速度(22)和力(22)；指尖位置 ，四元数旋转 ，线速度 ，角速度 ，力 和扭矩 。 |
| 前一动作(24) | 手腕力(3)和扭矩(3)；手指关节角度(18)。 |
| 物体状态(16) | 物体中心(3)，四元数旋转(4)，线速度(3)和角速度(3)；物体-目标距离(3)。 |
| 手-物体距离(36) | 手部点到物体点云距离(36)。 |
| 时间(29) | 当前时间(1)；正弦-余弦时间嵌入(28)。 |

Table 1. Input types for our dedicated policy networks, organized into five groups. Definitions for each element within these groups are provided in the appendix. These inputs are also applicable to our UniGraspTransformer.

表1. 我们专用策略网络的输入类型，分为五组。这些组中每个元素的定义在附录中提供。这些输入也适用于我们的UniGraspTransformer。

Dexterous Hand. In our implementation, we use the Shadow Hand [44], which has 18 active degrees of freedom (DOFs) in the fingers-5 for the thumb, 4 for the little finger, and 3 each for the remaining fingers-along with an additional 6 DOFs at the wrist. This gives the dexterous hand a total of 24 active DOFs. The wrist’s active DOFs are controlled via force and torque, while the fingers’ active DOFs are managed through joint angles. In addition, each finger, except for the thumb, has a passive DOF that won’t be directly controlled.

灵巧手。在我们的实现中，我们使用了Shadow Hand [44]，它有18个主动自由度(DOFs)在手指上——拇指5个，小指4个，其余手指各3个——以及手腕上额外的6个DOFs。这使得灵巧手总共有24个主动DOFs。手腕的主动DOFs通过力和扭矩控制，而手指的主动DOFs通过关节角度管理。此外，除拇指外，每个手指都有一个被动DOF，不会被直接控制。

Overview. As shown in Figure 2, the UniGraspTransformer training process comprises three main stages: 1) Dedicated Policy Network Training (Section 3.1), where individual reinforcement learning (RL) policy networks are trained, each dedicated to a single object across various initial poses; 2) Grasp Trajectory Generation (Section 3.2), where each policy network generates successful grasping trajectories for downstream training. Each trajectory is a sequence of steps capturing the comprehensive knowledge of the environment, including robotic action (e.g., finger-joint angles) and object state (e.g., pose and point cloud); 3) UniGrasp-Transformer Training, where all successful grasping trajectories from various objects and initial poses are used to train UniGraspTransformer in both state-based (Section 3.3) and vision-based (Section 3.4) settings. This supervised training process allows UniGraspTransformer to generalize well to both seen and unseen objects.

概述。如图2所示，UniGraspTransformer的训练过程包括三个主要阶段:1)专用策略网络训练(第3.1节)，其中训练了单独的强化学习(RL)策略网络，每个网络专门用于单个对象在各种初始姿势下的抓取；2)抓取轨迹生成(第3.2节)，其中每个策略网络生成 个成功的抓取轨迹用于下游训练。每个轨迹是一个步骤序列，捕捉了环境的全面知识，包括机器人动作(例如，手指关节角度)和对象状态(例如，姿势和点云)；3)UniGraspTransformer训练，其中使用来自各种对象和初始姿势的所有成功抓取轨迹来训练UniGraspTransformer，包括基于状态(第3.3节)和基于视觉(第3.4节)的设置。这个监督训练过程使UniGraspTransformer能够很好地泛化到已见和未见对象。

# 3.1. Dedicated Policy Network Training

# 3.1. 专用策略网络训练

Our training set consists of 3,200 unique tabletop objects. For each object, we train a dedicated policy network across various initial poses, using PPO [43] as our reinforcement learning optimization algorithm. During training, each object is randomly rotated to enhance the initial pose diversity. Once trained, each policy network can successfully grasp its corresponding object across a range of poses.

我们的训练集由3,200个独特的桌面对象组成。对于每个对象，我们使用PPO [43]作为强化学习优化算法，在各种初始姿势下训练一个专用策略网络。在训练过程中，每个对象被随机旋转以增强初始姿势的多样性。一旦训练完成，每个策略网络可以在各种姿势下成功抓取其对应的对象。

Input. Table 1 summarizes the input types for our dedicated policy networks, organized into five groups: 167-d proprioception, 24-d previous action, 16-d object state, 36- d hand-object distance, and 29-d time. These groups are concatenated into a single 272-d input vector.

输入。表1总结了我们的专用策略网络的输入类型，分为五组:167维本体感觉、24维前一动作、16维对象状态、36维手-对象距离和29维时间。这些组被连接成一个单一的272维输入向量。

Network Architecture. Each policy network is a 4-layer MLP with hidden dimensions of , followed by an action prediction head implemented as a single fully connected layer. This head outputs a 24-d vector (18 DOFs for the fingers and 6 DOFs for the wrist) representing the action for the current time step. The value network shares the same architecture as the policy network, also comprising 4 MLP layers with identical hidden dimensions, but it outputs a single scalar value.

网络架构。每个策略网络是一个4层MLP，隐藏维度为 ，随后是一个作为单个全连接层实现的动作预测头。这个头输出一个24维向量(手指18个DOFs和手腕6个DOFs)，表示当前时间步的动作。价值网络与策略网络共享相同的架构，也由4个MLP层组成，具有相同的隐藏维度，但它输出一个标量值。

Reward Function. The reward function is defined as:

奖励函数。奖励函数 定义为:

where the grasp reward penalizes the distance between the dexterous hand and the object, encouraging the hand to maintain contact with the object surface for a secure grasp; the contact flag is set to 1 if the distance between the hand and the object is below a specified threshold; the reward encourages the hand to remain open until contact is made with the object; once contact is established (i.e., ), the lift reward encourages the hand to perform the lifting action; the goal reward penalizes the distance between the object and the target goal; and the success reward provides a bonus when the object successfully reaches the goal. The formal definitions of all rewards are available in the appendix. The reward function in Eq. 1 is applied to each grasp trajectory, consisting of steps. In our implementation, we set .

其中抓取奖励 惩罚灵巧手与对象之间的距离，鼓励手保持与对象表面的接触以实现安全抓取；接触标志 在手与对象之间的距离低于指定阈值时设置为1；奖励 鼓励手在接触对象之前保持打开；一旦建立接触(即 )，提升奖励 鼓励手执行提升动作；目标奖励 惩罚对象与目标之间的距离；成功奖励 在对象成功到达目标时提供奖励。所有奖励的正式定义在附录中提供。公式1中的奖励函数应用于每个抓取轨迹，由 个步骤组成。在我们的实现中，我们设置 。

# 3.2. Grasp Trajectory Generation

# 3.2. 抓取轨迹生成

Each dedicated policy network is now able to grasp its assigned object across various initial poses, achieving an average success rate of across the 3,200 training objects. For each object, we randomly rotate it and use its corresponding policy network to generate a successful grasp trajectory. This process is repeated times per object, resulting in a dataset of successful grasp trajectories. Each trajectory, , is a sequence of steps, where represents robotic action (18 active DOFs for the fingers and 6 active DOFs for the wrist) at timestep- , and represents the observation of proprioception, previous action, object state, hand-object distance, time, and object point cloud, as defined in Table 1. The dataset is then used to train our UniGraspTransformer in a supervised manner, as described in Section 3.3.

每个专用策略网络现在能够在其分配的对象上跨越各种初始姿态进行抓取，在3,200个训练对象上实现了平均成功率为 。对于每个对象，我们随机旋转它并使用其对应的策略网络生成一个成功的抓取轨迹。这个过程每个对象重复 次，生成了一个包含 个 成功抓取轨迹的数据集 。每个轨迹 是一个步骤序列，其中 表示在时间步 的机器人动作(手指的18个主动自由度(DOF)和手腕的6个主动自由度)， 表示本体感觉、前一个动作、对象状态、手-对象距离、时间和对象点云的观察，如表1所定义。数据集 随后用于以监督方式训练我们的UniGraspTransformer，如第3.3节所述。

# 3.3. UniGraspTransformer Training

# 3.3. UniGraspTransformer训练

The objective is to use the generated trajectory dataset to train a universal grasp network, UniGraspTransformer, capable of grasping a variety of tabletop objects in diverse initial poses. UniGraspTransformer is designed to generalize to both seen objects from the training set and previously unseen objects within either seen or unseen categories.

目标是使用生成的轨迹数据集 训练一个通用的抓取网络UniGraspTransformer，能够抓取各种桌面对象在不同初始姿态下的情况。UniGraspTransformer被设计为能够泛化到训练集中的已见对象以及之前未见过的类别中的对象。

Settings. We train UniGraspTransformer under two settings: (1) a state-based setting, where object point clouds are perfectly accurate, with direct access to object’s positions and rotations, and (2) a vision-based setting, where object point clouds are estimated and reconstructed using five cameras mounted at the top and borders of the table, with object’s positions and rotations estimated rather than directly obtained. The primary distinction between these two settings is the method of acquiring object point clouds and the availability of oracle-level object states.

设置。我们在两种设置下训练UniGraspTransformer:(1)基于状态的设置，其中对象点云完全准确，可以直接访问对象的位置和旋转，以及(2)基于视觉的设置，其中对象点云通过安装在桌子顶部和边缘的五台摄像头进行估计和重建，对象的位置和旋转是估计的而不是直接获取的。这两种设置之间的主要区别在于获取对象点云的方法以及oracle级别对象状态的可用性。

We use the state-based setting to illustrate the key components of UniGraspTransformer, and describe its adaptation to the vision-based setting in Section 3.4.

我们使用基于状态的设置来说明UniGraspTransformer的关键组件，并在第3.4节中描述其适应基于视觉的设置。

Input Process. As detailed in Section 3.2, each trajectory consists of time steps. At each step , we train UniGraspTrans-former with -which includes information on proprioception, previous action, object state, hand-object distance, time, and object point cloud- as input to predict the corresponding action , a 24-d vector. Table 1 provides dimensions for each component: proprioception (167-d), previous action (24-d), object state (16-d), hand-object distance (36- d), and time (29-d). For encoding the object point cloud, an object encoder named S-Encoder, which has a similar structure as the PointNet [36], is trained specifically for the state-based setting, encoding the point cloud into an 128-d feature (see Figure 3). Thus, the model has six input vectors: 167-d proprioception, 24-d previous action, 16-d object state, 26- d hand-object distance, 29-d time and 128-d object feature. As illustrated in Figure 2(c), each input vector is mapped to a 256-d token via an individual MLP network. These six tokens serve as the input to UniGraspTransformer.

输入处理。如第3.2节所述，每个轨迹 由 个时间步组成。在每个步骤 ，我们训练UniGraspTransformer，以 作为输入来预测相应的动作 ，一个24维向量。 包括本体感觉、前一个动作、对象状态、手-对象距离、时间和对象点云的信息。表1提供了每个组件的维度:本体感觉(167维)、前一个动作(24维)、对象状态(16维)、手-对象距离(36维)和时间(29维)。为了编码对象点云，一个名为S-Encoder的对象编码器被训练用于基于状态的设置，将点云编码为128维特征(见图3)。因此，模型有六个输入向量:167维本体感觉、24维前一个动作、16维对象状态、26维手-对象距离、29维时间和128维对象特征。如图2(c)所示，每个输入向量通过一个单独的MLP网络映射到256维的token。这六个token作为UniGraspTransformer的输入。

Network Architecture and Loss Function. As illustrated in Figure 2(c), UniGraspTransformer consists primarily of several self-attention blocks, followed by a 4-layer MLP head that outputs a 24-d action prediction. By default, we use 12 self-attention blocks. For a given data pair at time step from trajectory , we first convert into six tokens as previously described. These tokens are then fed into UniGraspTransformer to produce the prediction . The model is optimized using L2 loss, defined as

网络架构与损失函数。如图2(c)所示，UniGraspTransformer主要由多个自注意力块组成，随后是一个4层MLP头，输出24维动作预测。默认情况下，我们使用12个自注意力块。对于来自轨迹 的时间步 的给定数据对 ，我们首先将 转换为六个令牌，如前所述。然后将这些令牌输入UniGraspTransformer以生成预测 。模型使用L2损失进行优化，定义为 。

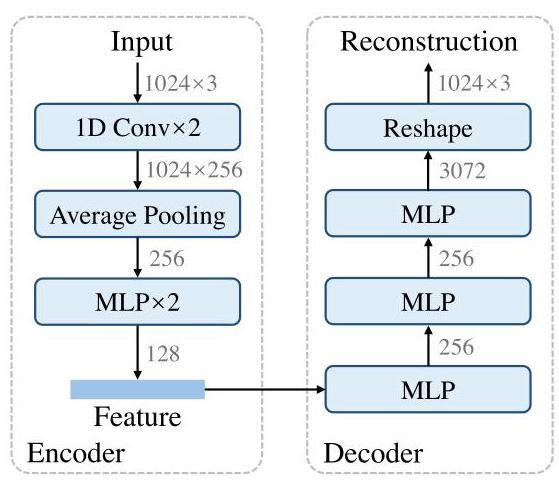


Figure 3. Illustration of the network architecture of the object point cloud encoder, S-Encoder, in the state-based setting. The process begins with sampling 1,024 points from the object point cloud, producing an input with a dimension of . This input is passed through the encoder, producing a 128-dimensional object feature, which the decoder then uses to reconstruct the 1,024 sampled points, with the Chamfer Distance serving as the loss function. During inference, only the encoder is used to convert an object point cloud into a 128-dimensional object feature.

图3. 在基于状态的设置中，对象点云编码器S-Encoder的网络架构示意图。该过程从对象点云中采样1,024个点开始，生成一个维度为 的输入。该输入通过编码器，生成一个128维的对象特征，解码器随后使用该特征重建1,024个采样点，以Chamfer Distance作为损失函数。在推理过程中，仅使用编码器将对象点云转换为128维的对象特征。

# 3.4. Adaptation to the Vision-Based Setting

# 3.4. 适应基于视觉的设置

Input Adaptation. In the vision-based setting, we use five cameras mounted at the table’s top and borders to estimate the object point clouds. The estimated point clouds consist of two components: 1) the partial object point cloud and 2) the hand point cloud, which is segmented and removed in our implementation. In the state-based setting, the object point clouds are uniformly sampled from the object mesh, which are complete and accurate. This difference affects the input to UniGraspTransformer as follows: 1) for object state representation, we use the center of the partial object point cloud as the object position and apply PCA on this partial cloud to represent the object rotation; 2) we use the partial object point cloud to calculate hand-object distance; and 3) we re-train a dedicated object encoder, termed V-Encoder, to extract features from the partial object points. Other configurations, such as network architecture, loss function, and supervision signals, remain unchanged.

输入适应。在基于视觉的设置中，我们使用安装在桌面和边界的五个摄像头来估计对象点云。估计的点云由两部分组成:1)部分对象点云和2)手部点云，在我们的实现中，手部点云被分割并移除。在基于状态的设置中，对象点云是从对象网格中均匀采样的，这些点云是完整且准确的。这种差异对UniGraspTransformer的输入影响如下:1)对于对象状态表示，我们使用部分对象点云的中心作为对象位置，并对该部分点云应用PCA来表示对象旋转；2)我们使用部分对象点云来计算手与对象的距离；3)我们重新训练一个专门的对象编码器，称为V-Encoder，以从部分对象点中提取特征。其他配置，如网络架构、损失函数和监督信号，保持不变。

V-Encoder. In Section 3.3, for the state-based setting, we train an S-Encoder (see Figure 3) that encodes complete object point clouds into object features. In contrast, in the vision-based setting, we only have access to partial object point clouds. To extract their features, we re-train a V-Encoder, maintaining the same network architecture as the S-Encoder. The key modifications are as follows: 1) the input consists of 1,024 sampled points from the partial object point cloud; 2) a distillation loss is applied to regularize the V-Encoder’s latent features, with supervision provided by the latent features of the corresponding complete object point cloud extracted by the S-Encoder. After training, the V-Encoder can extract a 128-d object feature from partial object point clouds.

V-Encoder。在第3.3节中，对于基于状态的设置，我们训练了一个S-Encoder(见图3)，它将完整的对象点云编码为对象特征。相比之下，在基于视觉的设置中，我们只能访问部分对象点云。为了提取它们的特征，我们重新训练了一个V-Encoder，保持与S-Encoder相同的网络架构。关键修改如下:1)输入由部分对象点云中的1,024个采样点组成；2)应用蒸馏损失来正则化V-Encoder的潜在特征，监督由S-Encoder提取的相应完整对象点云的潜在特征提供。训练后，V-Encoder可以从部分对象点云中提取128维的对象特征。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | State-Based Setting (%) | | | Vision-Based Setting (%) | | |
| Seen Obj. | Unseen Obj. Seen Cat. | Unseen Obj. Unseen Cat. | Seen Obj. | Unseen Obj. Seen Cat. | Unseen Obj. Unseen Cat. |
| [43] | 24.3 | 20.9 | 17.2 | 20.6 | 17.2 | 15.0 |
| [40] | 20.8 | 15.3 | 11.1 | 17.9 | 15.2 | 13.9 |
|  | 31.9 | 26.4 | 23.1 | 27.6 | 23.2 | 20.0 |
| GSL† [13] | 57.3 | 54.1 | 50.9 | 54.1 | 50.2 | 44.8 |
| UniDexGrasp [57] | 79.4 | 74.3 | 70.8 | 73.7 | 68.6 | 65.1 |
| UniDexGrasp++ [50] | 87.9 | 84.3 | 83.1 | 85.4 | 79.6 | 76.7 |
| Ours | 91.2 | 89.2 | 88.3 | 88.9 | 87.3 | 86.8 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | 基于状态的设置 (%) | | | 基于视觉的设置 (%) | | |
| 可见物体 | 未见物体 可见类别 | 未见物体 未见类别 | 可见物体 | 未见物体 可见类别 | 未见物体 未见类别 |
| [43] | 24.3 | 20.9 | 17.2 | 20.6 | 17.2 | 15.0 |
| [40] | 20.8 | 15.3 | 11.1 | 17.9 | 15.2 | 13.9 |
|  | 31.9 | 26.4 | 23.1 | 27.6 | 23.2 | 20.0 |
| GSL† [13] | 57.3 | 54.1 | 50.9 | 54.1 | 50.2 | 44.8 |
| UniDexGrasp [57] | 79.4 | 74.3 | 70.8 | 73.7 | 68.6 | 65.1 |
| UniDexGrasp++ [50] | 87.9 | 84.3 | 83.1 | 85.4 | 79.6 | 76.7 |
| 我们的方法 | 91.2 | 89.2 | 88.3 | 88.9 | 87.3 | 86.8 |

Table 2. Comparison with state-of-the-art methods using a universal model for dexterous robotic grasping across both state-based and vision-based settings, evaluated by success rate. Evaluation on unseen objects from either seen or unseen categories assesses the models’ generalization capability. indicates results reported in UniDexGrasp++ [50]. "Obj.": Objects. "Cat.": categories.

表2. 使用通用模型在基于状态和基于视觉的设置下进行灵巧机器人抓取的与最先进方法的比较，通过成功率进行评估。对来自已见或未见类别的未见对象的评估评估了模型的泛化能力。 表示在UniDexGrasp++ [50]中报告的结果。"Obj.":对象。"Cat.":类别。

# 4. Experiments

# 4. 实验

Datasets. We utilize the UniDexGrasp++ [50] dexterous grasping dataset, comprising 3,200 objects across 133 categories for training. Evaluation is conducted on these 3,200 seen objects, as well as on 140 unseen objects from seen categories and 100 unseen objects from unseen categories. For seen objects, we generate test initial poses separately from those used during training, applying this protocol for both dedicated policy network evaluation and UniGrasp-Transformer evaluation.

数据集。我们使用UniDexGrasp++ [50]灵巧抓取数据集，包含133个类别的3,200个对象用于训练。评估在这些3,200个已见对象上进行，以及在来自已见类别的140个未见对象和来自未见类别的100个未见对象上进行。对于已见对象，我们生成与训练期间使用的初始姿势不同的测试初始姿势，将此协议应用于专用策略网络评估和UniGrasp-Transformer评估。

Evaluation Protocols. Following UniDexGrasp++ [50], each object is randomly rotated and dropped onto the table to enhance the diversity of its initial poses. This process is repeated 1,000 times for robust evaluation. A grasp is considered successful if the object reaches the target goal within steps. We report the average success rate across all objects and grasp attempts. The evaluation is performed in two configurations: a state-based setting and a vision-based setting. The state-based setting represents an ideal scenario where the object point cloud is entirely accessible and flawlessly accurate. Conversely, in the vision-based setting (see Section 3.4), the object point cloud is reconstructed using depth images captured from five different viewpoints by five cameras. Additionally, the dexterous hand may partially occlude the object, resulting in only a portion of the object’s point cloud being accessible.

评估协议。遵循UniDexGrasp++ [50]，每个对象被随机旋转并放置在桌子上，以增强其初始姿势的多样性。此过程重复1,000次以进行稳健评估。如果对象在 步内达到目标，则认为抓取成功。我们报告所有对象和抓取尝试的平均成功率。评估在两种配置下进行:基于状态的设置和基于视觉的设置。基于状态的设置代表了一个理想场景，其中对象点云完全可访问且完美准确。相反，在基于视觉的设置中(见第3.4节)，对象点云使用从五个不同视角由五个摄像头捕获的深度图像重建。此外，灵巧手可能会部分遮挡对象，导致只有部分对象的点云可访问。

Implementation Details. For a single dedicated policy network, we create 1,000 simulation environments, and update the policy network every 16 steps over iterations, with a learning rate of . Training is conducted parallelly on 16 NVIDIA V100 GPUs, requiring 80 hours for 3,200 dedicated policies. The UniGraspTransformer model is trained with a batch size of 800 trajectories over 100 epochs, using a fixed learning rate of . Training is performed on 8 NVIDIA A100 GPUs and takes around 70 hours to complete. The object encoders, both state-based and vision-based, are trained on point cloud data with a batch size of 100. Training runs for iterations with a learning rate of 5e-4 on an NVIDIA A100 GPU, requiring 40 hours.

实现细节。对于单个专用策略网络，我们创建了1,000个模拟环境，并在 次迭代中每16步更新一次策略网络，学习率为 。训练在16个NVIDIA V100 GPU上并行进行，需要80小时完成3,200个专用策略的训练。UniGraspTransformer模型以800个轨迹的批量大小训练100个周期，使用固定学习率 。训练在8个NVIDIA A100 GPU上进行，大约需要70小时完成。基于状态和基于视觉的对象编码器在点云数据上训练，批量大小为100。训练在NVIDIA A100 GPU上进行 次迭代，学习率为5e-4，需要40小时。

# 4.1. Main Results

# 4.1. 主要结果

Dedicated Policy Networks. As detailed in Section 3.1, we train an individual policy network for each of the 3,200 training objects. During evaluation, each policy is used exclusively with its corresponding object, achieving an average success rate of . However, these dedicated policies cannot be evaluated on unseen objects, as they lack generalization capability.

专用策略网络。如第3.1节所述，我们为3,200个训练对象中的每一个训练一个单独的策略网络。在评估期间，每个策略仅用于其对应的对象，平均成功率为 。然而，这些专用策略无法在未见对象上进行评估，因为它们缺乏泛化能力。

UniGraspTransformer. Table 2 compares our UniGrasp-Transformer with state-of-the-art methods using a universal model for dexterous robotic grasping across both state-based and vision-based settings. Our model outperforms UniDexGrasp++ [50] by 3.3% and 3.5% on seen objects in the state-based and vision-based settings, achieving success rates of 91.2% and 88.9%, respectively. Furthermore, our UniGraspTransformer demonstrates strong generalization capabilities, achieving high success rates on unseen objects and unseen categories in both settings. It surpasses UniDexGrasp++ [50] by 4.9% (7.7%) and 5.2% (10.1%) on unseen objects from seen categories and unseen objects from unseen categories in the state-based (vision-based) setting. The transition from seen to unseen categories further verifies the generalization capability of our approach, with

UniGraspTransformer。表2将我们的UniGrasp-Transformer与使用通用模型在基于状态和基于视觉的设置下进行灵巧机器人抓取的最先进方法进行了比较。我们的模型在基于状态和基于视觉的设置下，在已见对象上分别比UniDexGrasp++ [50]高出3.3%和3.5%，成功率分别为91.2%和88.9%。此外，我们的UniGraspTransformer展示了强大的泛化能力，在两种设置下对未见对象和未见类别都取得了高成功率。在基于状态(基于视觉)的设置下，对来自已见类别的未见对象和来自未见类别的未见对象，它分别比UniDexGrasp++ [50]高出4.9%(7.7%)和5.2%(10.1%)。从已见类别到未见类别的过渡进一步验证了我们方法的泛化能力，

|  |  |  |  |
| --- | --- | --- | --- |
| Trajectory Number(M) | 0.2K | 0.5K | 1K |
| Success Rate (%) | 87.2 | 89.3 | 91.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 轨迹编号(M) | 0.2K | 0.5K | 1K |
| 成功率 (%) | 87.2 | 89.3 | 91.2 |

Table 3. For each of the 3,200 dedicated policy networks, we generate successful grasping trajectories, which are then distilled into UniGraspTransformer. We analyze the impact of varying .

表3. 对于每个3,200个专用策略网络，我们生成 条成功的抓取轨迹，这些轨迹随后被提炼到UniGraspTransformer中。我们分析了不同 的影响。

|  |  |  |  |
| --- | --- | --- | --- |
| Self-Attention Blocks(K) | - | 6 | 12 |
| Success Rate (%) | 85.5 | 89.7 | 91.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 自注意力模块(K) | - | 6 | 12 |
| 成功率 (%) | 85.5 | 89.7 | 91.2 |

Table 4. Analysis of the impact of different numbers of self-attention blocks in UniGraspTransformer. The model consists of self-attention blocks followed by a 4-layer MLP head. "-" indicates a configuration that uses only the MLP head without any self-attention blocks.

表4. 分析UniGraspTransformer中不同数量自注意力块的影响。该模型由 个自注意力块组成，后接一个4层MLP头。"-"表示仅使用MLP头而不使用任何自注意力块的配置。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object Number | 400 | 800 | 1,600 | 3,200 (All) |
| Success Rate (%) | 92.5 | 91.8 | 91.3 | 91.2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对象编号 | 400 | 800 | 1,600 | 3,200(全部) |
| 成功率(%) | 92.5 | 91.8 | 91.3 | 91.2 |

Table 5. Analysis of distilling varying numbers of dedicated policy networks (one policy per object) into UniGraspTransformer. The evaluation is conducted on the corresponding seen object set. only a minimal drop in success rate observed in both the state-based (91.2% to 88.3%) and vision-based (88.9% to 86.8%) settings.

表5. 将不同数量的专用策略网络(每个对象一个策略)蒸馏到UniGraspTransformer中的分析。评估在相应的已见对象集上进行。在基于状态(从91.2%降至88.3%)和基于视觉(从88.9%降至86.8%)的设置中，仅观察到成功率的轻微下降。

|  |  |  |
| --- | --- | --- |
| Method | DAgger | UniGraspTransformer |
| Success Rate (%) | 88.2 | 92.5 |

|  |  |  |
| --- | --- | --- |
| 方法 | DAgger | UniGraspTransformer |
| 成功率 (%) | 88.2 | 92.5 |

Table 6. Analysis of distilling 400 dedicated policies into one universal policy, via online or offline methods. The evaluation is conducted on the corresponding seen object set.

表6. 通过在线或离线方法将400个专用策略蒸馏为一个通用策略的分析。评估在相应的已见对象集上进行。

# 4.2. Ablation Studies

# 4.2. 消融研究

Unless otherwise specified, the ablation studies are conducted on the 3,200 seen objects.

除非另有说明，消融研究均在3,200个已见对象上进行。

Scalability of UniGraspTransformer. Our approach employs a two-step process: initially, dedicated policy networks are individually trained for each object using reinforcement learning to generate a set of successful grasp trajectories. These trajectories are then distilled into a single universal network, UniGraspTransformer. This design allows us to investigate the scalability of UniGraspTrans-former in terms of trajectory set size, network capacity, and the number of objects it can manage. The following studies are conducted under the state-based setting.

UniGraspTransformer的可扩展性。我们的方法采用两步过程:首先，使用强化学习为每个对象单独训练专用策略网络，以生成一组成功的抓取轨迹。然后，这些轨迹被蒸馏到一个单一的通用网络UniGraspTransformer中。这种设计使我们能够研究UniGraspTransformer在轨迹集大小、网络容量和可管理对象数量方面的可扩展性。以下研究在基于状态的设置下进行。

* Trajectory Set Size. As shown in Table 3, UniGrasp-Transformer demonstrates strong scalability in handling an increasing number of trajectories. The success rate improves as the number of grasping trajectories per object used for training grows, indicating the model’s ability to effectively learn and generalize from diverse trajectories across various objects.
* 轨迹集大小。如表3所示，UniGraspTransformer在处理越来越多的轨迹时表现出强大的可扩展性。随着用于训练的每个对象的抓取轨迹数量的增加，成功率提高，表明模型能够有效地从各种对象的不同轨迹中学习和泛化。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proprio- ception | Prev. Action | Obj. State | Obj. Feat. | Hand- Obj. Dist. | Time | SR (%) |
| ✓ | ✓ | ✓ |  |  |  | 78.4 |
| ✓ | ✓ | ✓ | ✓ |  |  | 86.6 |
| ✓ | ✓ | ✓ | ✓ | ✓ |  | 89.9 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 91.2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 本体感觉 | 前一个动作 | 物体状态 | 物体特征 | 手与物体距离 | 时间 | 成功率(%) |
| ✓ | ✓ | ✓ |  |  |  | 78.4 |
| ✓ | ✓ | ✓ | ✓ |  |  | 86.6 |
| ✓ | ✓ | ✓ | ✓ | ✓ |  | 89.9 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 91.2 |

Table 7. Effects of different input components on UniGraspTrans-former Training. "Prev.": previous. "Obj.": Object. "Feat.": feature. "Dist.": distance. "SR": success rate.

表7. 不同输入组件对UniGraspTransformer训练的影响。"Prev.": 前一个。"Obj.": 对象。"Feat.": 特征。"Dist.": 距离。"SR": 成功率。

|  |  |  |
| --- | --- | --- |
| Center of Partial Object | PCA of Partial Object | Success Rate (%) |
|  |  | 83.2 |
| ✓ |  | 86.4 |
| ✓ | ✓ | 88.9 |

|  |  |  |
| --- | --- | --- |
| 部分对象中心 | 部分对象的主成分分析(PCA) | 成功率(%) |
|  |  | 83.2 |
| ✓ |  | 86.4 |
| ✓ | ✓ | 88.9 |

Table 8. Utilizing approximate estimations-specifically, the center and PCA of partial object point clouds-enhances performance compared to the baseline without these estimations.

表8. 利用近似估计——特别是部分物体点云的中心和主成分分析(PCA)——相比没有这些估计的基线，提升了性能。

* Network Capacity. As illustrated in Figure 2(c), our UniGraspTransformer consists of input MLP layers, self-attention blocks, and a 4-layer MLP head. Table 4 presents an analysis of the impact of network capacity on performance. The results show that increasing the number of self-attention blocks enhances the success rate, indicating that UniGraspTransformer scales effectively with additional self-attention layers. Specifically, the success rate improves from 85.5% (without self-attention blocks) to 89.7% with 6 blocks and finally reaches 91.2% with 12 blocks.
* 网络容量。如图2(c)所示，我们的UniGraspTransformer由输入MLP层、 个自注意力块和一个4层MLP头组成。表4展示了网络容量对性能影响的分析。结果表明，增加自注意力块的数量提高了成功率，表明UniGraspTransformer在增加自注意力层时能够有效扩展。具体来说，成功率从85.5%(没有自注意力块)提高到89.7%(6个块)，最终达到91.2%(12个块)。
* Object Number. Table 5 shows the success rate of our UniGraspTransformer when distilling varying numbers of object-specific policy networks, evaluated across different object sets with400,800,1,600, and 3,200 objects. The results indicate a high and stable success rate, with a slight decline as the number of objects increases.
* 物体数量。表5展示了我们的UniGraspTransformer在蒸馏不同数量的物体特定策略网络时的成功率，评估了包含400、800、1,600和3,200个物体的不同物体集。结果表明，成功率较高且稳定，随着物体数量的增加略有下降。
* Online vs. Offline. Table 6 compares the online distillation method: DAgger with MLPs, to our offline distillation method: UniGraspTransformer. The results highlight the advantages of offline distillation with large policy networks when handling a diverse set of teacher policies.
* 在线与离线。表6比较了在线蒸馏方法:使用MLPs的DAgger，与我们的离线蒸馏方法:UniGraspTransformer。结果突出了在处理多样化教师策略时，使用大型策略网络的离线蒸馏方法的优势。

Input of UniGraspTransformer. Table 7 presents an analysis of different input components and their effects on performance. The basic input includes proprioception, previous actions, and object state. We progressively enhance the input by incorporating features from the state-based object encoder, hand-object distances, and temporal information. The performance improves consistently, indicating that Un-iGraspTransformer effectively utilizes diverse information sources to enhance robotic grasping capabilities.

UniGraspTransformer的输入。表7展示了不同输入组件及其对性能影响的分析。基本输入包括本体感知、先前动作和物体状态。我们通过逐步加入基于状态的物体编码器特征、手-物体距离和时间信息来增强输入。性能持续提升，表明UniGraspTransformer有效利用多样化信息源来增强机器人抓取能力。

UniGraspTransformer in the Vision-Based Setting. As described in Section 3.4, the vision-based setting involves estimating object point clouds from five cameras, resulting in incomplete point clouds. Unlike the state-based setting-where complete object states, including rotation, are accessible-object rotation information is unavailable in this configuration. A baseline solution in the vision-based setting is to omit the point cloud center and object rotation inputs. Alternatively, the center of partial object point clouds can substitute the center of full point clouds, and principal component analysis (PCA) can approximate object rotations. Table 8 compares these alternatives with the baseline, demonstrating that these estimations improve performance over the baseline lacking any estimation.

基于视觉的UniGraspTransformer。如第3.4节所述，基于视觉的设置涉及从五个摄像头估计物体点云，导致点云不完整。与基于状态的设置不同——在基于状态的设置中可以访问包括旋转在内的完整物体状态——在这种配置中无法获取物体旋转信息。基于视觉设置的一个基线解决方案是省略点云中心和物体旋转输入。或者，部分物体点云的中心可以替代完整点云的中心，主成分分析(PCA)可以近似物体旋转。表8将这些替代方案与基线进行了比较，表明这些估计相比没有任何估计的基线提升了性能。

|  |  |
| --- | --- |
| Distillation | Success Rate (%) |
|  | 86.7 |
| ✓ | 88.9 |

|  |  |
| --- | --- |
| 蒸馏 | 成功率 (%) |
|  | 86.7 |
| ✓ | 88.9 |

Table 9. Impact of using a vision-based object encoder with and without distillation loss on UniGraspTransformer’s performance.

表9. 使用基于视觉的对象编码器(vision-based object encoder)在有和无蒸馏损失(distillation loss)情况下对UniGraspTransformer性能的影响。

|  |  |  |  |
| --- | --- | --- | --- |
|  | w/ Point Cloud | w/ Center | SR (%) |
|  |  | ✓ | 90.3 |
|  | ✓ |  | 92.9 |
| ✓ | ✓ |  | 94.1 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 带点云 | 带中心 | SR (%) |
|  |  | ✓ | 90.3 |
|  | ✓ |  | 92.9 |
| ✓ | ✓ |  | 94.1 |

Table 10. Impact of incorporating reward and two variants of reward on the performance of dedicated policy networks.

表10. 引入奖励 和两种奖励 变体对专用策略网络性能的影响。

In Section 3.4, we introduce a distillation loss for training the vision-based object encoder, enabling it to extract a 128-dimensional feature from partial object point clouds. This distillation process transfers knowledge from the state-based object encoder, which encodes complete object point clouds into a single feature, to the vision-based encoder. Table 9 studies the impact of using a vision-based object encoder with and without distillation loss on UniGraspTrans-former’s performance, showing a 2.2% improvement in success rate, underscoring the value of distillation in training the vision-based encoder.

在第3.4节中，我们引入了一种蒸馏损失用于训练基于视觉的目标编码器，使其能够从部分目标点云中提取128维特征。这一蒸馏过程将基于状态的目标编码器的知识转移到基于视觉的编码器，前者将完整的目标点云编码为单一特征。表9研究了使用和不使用蒸馏损失的基于视觉的目标编码器对UniGraspTransformer性能的影响，显示成功率提高了2.2%，突显了蒸馏在训练基于视觉的编码器中的价值。

Reward Functions for Dedicated Policy Network. As outlined in Section 3.1, the dedicated policies are trained with reward functions defined in Eq.1. We explore two reward functions: (1) the grasp reward , which penalizes the distance between the dexterous hand and the object, and (2) the reward , which encourages the hand to stay open until it contacts the object. In our default setup, is computed by measuring the distances between 36 hand points and the object point clouds. In Table 10, we examine an alternative version where is based on distances between the 36 hand points and the object center. Additionally, Table 10 assesses the impact of including or excluding the reward . A well-designed reward function enhances the performance of all dedicated policy networks, achieving an average success rate of 94.1% across 3,200 training objects. Please refer to our supplementary material for more details.

专用策略网络的奖励函数。如第3.1节所述，专用策略通过公式1中定义的奖励函数进行训练。我们探索了两种奖励函数:(1)抓取奖励 ，惩罚灵巧手与目标之间的距离；(2)奖励 ，鼓励手在接触目标之前保持张开。在我们的默认设置中， 通过测量36个手点与目标点云之间的距离来计算。在表10中，我们考察了另一种版本，其中 基于36个手点与目标中心之间的距离。此外，表10评估了包含或排除奖励 的影响。设计良好的奖励函数提升了所有专用策略网络的性能，在3,200个训练目标上实现了94.1%的平均成功率。更多详情请参阅我们的补充材料。

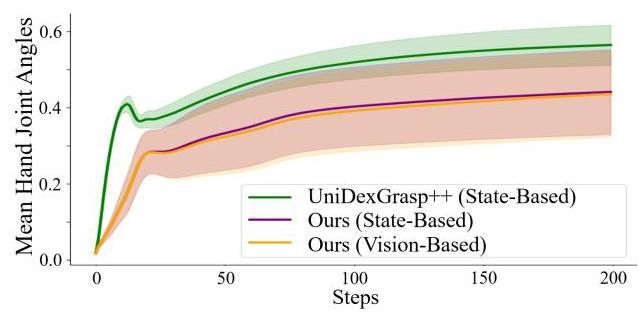


Figure 4. Quantitative analysis of grasp pose diversity.

图4. 抓取姿势多样性的定量分析。

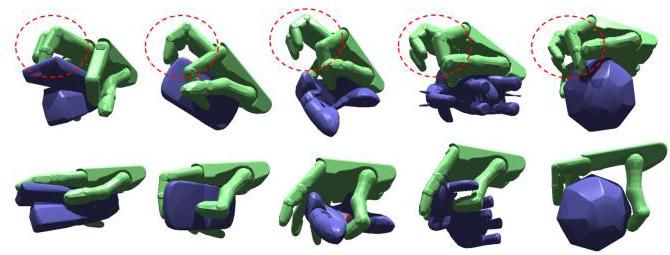


Figure 5. Comparison of grasp poses generated by the state-based universal model from the UniDexGrasp++ [50] (top row) and our UniGraspTransformer (bottom row). Each column displays two distinct grasp poses for the same object with the same initial pose.

图5. 由UniDexGrasp++ [50]的基于状态的通用模型(上行)和我们的UniGraspTransformer(下行)生成的抓取姿势对比。每列显示同一目标在相同初始姿势下的两种不同抓取姿势。

Diversity Analysis on Grasp Pose. Figure 4 provides a quantitative comparison of grasp pose diversity between the state-based UniDexGrasp++ [50] and both the state-based and vision-based versions of UniGraspTransformer. During inference, each model generates 10 successful 200- step trajectories for each of the 3,200 training objects, with mean hand joint angles (normalized) used to represent the hand state at each step. The plotted range in Figure 4 demonstrates that UniGraspTransformer exhibits a broader range, indicating its capability to produce diverse grasp poses across a variety of objects. Figure 5 presents visual examples that highlight the greater diversity of grasp poses generated by our model compared to the previous method.

抓取姿势的多样性分析。图4提供了基于状态的UniDexGrasp++ [50]与基于状态和基于视觉的UniGraspTransformer在抓取姿势多样性上的定量比较。在推理过程中，每个模型为3,200个训练目标生成10条成功的200步轨迹，使用平均手关节角度(归一化)表示每一步的手状态。图4中的绘制范围显示，UniGraspTransformer表现出更广泛的范围，表明其能够在各种目标上生成多样化的抓取姿势。图5展示了视觉示例，突出了我们的模型相比之前方法生成的抓取姿势的更大多样性。

# 5. Conclusion

# 5. 结论

In this work, we introduce UniGraspTransformer, a universal Transformer-based network that streamlines the training process for dexterous robotic grasping while enhancing scalability, flexibility, and diversity in grasping strategies. Our approach simplifies traditional complex pipelines by employing dedicated reinforcement learning-based policy networks for individual objects, followed by an efficient offline distillation process that consolidates successful grasping trajectories into a single, scalable model. Our UniGraspTransformer is capable of handling thousands of objects in varied poses, exhibiting robustness and adaptability across both state-based and vision-based settings. Notably, our model significantly improves grasp success rates on seen, unseen within-category, and fully novel objects, outperforming the current state-of-the-art with substantial gains in success rates across various settings.

在本工作中，我们介绍了UniGraspTransformer，一种基于Transformer的通用网络，简化了灵巧机器人抓取的训练过程，同时增强了抓取策略的可扩展性、灵活性和多样性。我们的方法通过为单个目标使用专用的基于强化学习的策略网络，简化了传统的复杂流程，随后通过高效的离线蒸馏过程将成功的抓取轨迹整合到一个可扩展的单一模型中。我们的UniGraspTransformer能够处理数千个不同姿势的目标，在基于状态和基于视觉的设置中表现出鲁棒性和适应性。值得注意的是，我们的模型在已见、未见类别内和全新目标上的抓取成功率显著提高，在各种设置中均优于当前的最先进技术，取得了显著的提升。

References

参考文献

[1] 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

Yunfei Bai 和 C. Karen Liu. 使用手掌和手指进行灵巧操作。在2014年IEEE国际机器人与自动化会议(ICRA)上，第1560-1565页，2014年。2

[2] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact, 2019. 2

Samarth Brahmbhatt, Ankur Handa, James Hays, 和 Dieter Fox. Contactgrasp:从接触中合成功能性多指抓取，2019年。2

[3] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter, 2021. 2

Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, 和 Juan Nieto. 体积抓取网络:在杂乱环境中实时检测6自由度抓取，2021年。2

[4] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions, 2022. 2

Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, 和 Otmar Hilliges. D-grasp:手-目标交互中物理上合理的动态抓取合成，2022年。2

[5] Wojciech Marian Czarnecki, Razvan Pascanu, Simon Osin-dero, Siddhant M. Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation, 2019. 3

Wojciech Marian Czarnecki, Razvan Pascanu, Simon Osin-dero, Siddhant M. Jayakumar, Grzegorz Swirszcz, 和 Max Jaderberg. 蒸馏策略蒸馏，2019年。3

[6] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11441- 11450, 2020. 2

方浩舒, 王晨曦, 苟明浩, 和陆策武. Graspnet-1billion: 一个用于通用物体抓取的大规模基准. 在2020年IEEE/CVF计算机视觉与模式识别会议(CVPR)上, 页码11441-11450, 2020. 2

[7] 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

耿浩然, 李子明, 耿亦然, 陈佳怡, 董浩, 和王鹤. Partmanip: 从点云观察中学习跨类别可泛化的部件操作策略, 2023. 2

[8] Abraham George, Alison Bartsch, and Amir Barati Farimani. Openvr: Teleoperation for manipulation, 2023. 3

亚伯拉罕·乔治, 艾莉森·巴特施, 和阿米尔·巴拉蒂·法里马尼. Openvr: 用于操作的远程操作, 2023. 3

[9] Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning, 2018. 2, 3

迪比亚·戈什, 阿维·辛格, 阿拉温德·拉杰斯瓦兰, 维卡什·库马尔, 和谢尔盖·莱文. 分而治之的强化学习, 2018. 2, 3

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3

杰弗里·辛顿, 奥里奥尔·维尼亚尔斯, 和杰夫·迪恩. 蒸馏神经网络中的知识, 2015. 3

[11] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation, 2024. 2

黄文龙, 王晨, 李云竹, 张若涵, 和李飞飞. Rekep: 用于机器人操作的关系关键点约束的时空推理, 2024. 2

[12] InspireRobots. https://inspire-robots.store/ collections/the-dexterous-hands,2016.1,14

InspireRobots. https://inspire-robots.store/ collections/the-dexterous-hands,2016.1,14

[13] Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning, 2022. 2, 3, 6, 14

贾志伟, 李轩林, 凌展, 刘爽, 吴亦然, 和苏浩. 通过通才-专家学习改进策略优化, 2022. 2, 3, 6, 14

[14] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation, 2021. 2

姜汉文, 刘少伟, 王佳顺, 和王晓龙. 用于人类抓取生成的手-物体接触一致性推理, 2021. 2

[15] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation, 2018. 2

德米特里·卡拉什尼科夫, 亚历克斯·伊尔潘, 彼得·帕斯特, 朱利安·伊巴尔兹, 亚历山大·赫尔佐格, 埃里克·张, 迪尔德丽·奎伦, 伊桑·霍利, 姆里纳尔·卡拉克里希南, 文森特·范霍克, 和谢尔盖·莱文. Qt-opt: 可扩展的基于视觉的机器人操作深度强化学习, 2018. 2

[16] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 4304-4311, 2015. 2

丹尼尔·卡普勒, 珍妮特·博格, 和斯特凡·沙尔. 利用大数据进行抓取规划. 在2015年IEEE国际机器人与自动化会议(ICRA)上, 页码4304-4311, 2015. 2

[17] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. Hg-dagger:

迈克尔·凯利, 切尔西·西德兰, 凯瑟琳·德里格斯-坎贝尔, 和迈克尔·J·科琴德弗. Hg-dagger:

Interactive imitation learning with human experts, 2019. 2, 3

与人类专家进行交互式模仿学习, 2019. 2, 3

[18] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms, 2020. 2

奥利弗·克雷默, 斯科特·尼库姆, 和乔治·科尼达里斯. 机器人操作学习的回顾: 挑战、表示和算法, 2020. 2

[19] Vikash Kumar, Abhishek Gupta, Emanuel Todorov, and Sergey Levine. Learning dexterous manipulation policies from experience and imitation, 2016. 1, 2

维卡什·库马尔, 阿布舍克·古普塔, 伊曼纽尔·托多罗夫, 和谢尔盖·莱文. 从经验和模仿中学习灵巧操作策略, 2016. 1, 2

[20] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers, 2020. 2

刘敏, 潘哲荣, 徐凯, 甘古利·卡尼什卡, 和迪内什·马诺查. 用于高自由度夹持器的深度可微分抓取规划器, 2020. 2

[21] Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. IEEE Robotics and Automation Letters, 7(1): 470-477, 2022. 2

刘腾宇, 刘泽宇, 焦子源, 朱艺昕, 朱松纯. 使用可微分力闭合估计器合成具有任意手部结构的多样且物理稳定的抓取. IEEE机器人与自动化快报, 7(1): 470-477, 2022. 2

[22] Qingkai Lu, Kautilya Chenna, Balakumar Sundaralingam, and Tucker Hermans. Planning multi-fingered grasps as probabilistic inference in a learned deep network, 2018. 2

卢庆凯, Kautilya Chenna, Balakumar Sundaralingam, Tucker Hermans. 在学习的深度网络中规划多指抓取作为概率推理, 2018. 2

[23] Tyler Ga Wei Lum, Martin Matak, Viktor Makoviychuk, Ankur Handa, Arthur Allshire, Tucker Hermans, Nathan D. Ratliff, and Karl Van Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics, 2024. 2

Tyler Ga Wei Lum, Martin Matak, Viktor Makoviychuk, Ankur Handa, Arthur Allshire, Tucker Hermans, Nathan D. Ratliff, Karl Van Wyk. Dextrah-g: 使用几何织物的像素到动作灵巧手臂抓取, 2024. 2

[24] Jeffrey Mahler and Ken Goldberg. Learning deep policies for robot bin picking by simulating robust grasping sequences. In Proceedings of the 1st Annual Conference on Robot Learning, pages 515-524. PMLR, 2017. 2

Jeffrey Mahler, Ken Goldberg. 通过模拟稳健抓取序列学习机器人箱拣选的深度策略. 第一届机器人学习年会论文集, 515-524页. PMLR, 2017. 2

[25] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, 2017. 2

Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, Ken Goldberg. Dex-net 2.0: 使用合成点云和分析抓取指标规划稳健抓取的深度学习, 2017. 2

[26] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. Science Robotics, 4(26):eaau4984, 2019. 2

Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, Ken Goldberg. 学习双手灵巧机器人抓取策略. 科学机器人, 4(26):eaau4984, 2019. 2

[27] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. 3, 11

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, Gavriel State. Isaac gym: 基于GPU的高性能物理模拟用于机器人学习, 2021. 3, 11

[28] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances, 2021. 2

Priyanka Mandikal, Kristen Grauman. 学习以对象为中心的视觉可供性进行灵巧抓取, 2021. 2

[29] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video, 2022. 3

Priyanka Mandikal, Kristen Grauman. Dexvip: 从视频中学习基于人手姿势先验的灵巧抓取, 2022. 3

[30] Malte Mosbach and Sven Behnke. Grasp anything: Combining teacher-augmented policy gradient learning with instance segmentation to grasp arbitrary objects, 2024. 2

Malte Mosbach, Sven Behnke. 抓取任何物体: 结合教师增强策略梯度学习与实例分割以抓取任意物体, 2024. 2

[31] Tongzhou Mu, Jiayuan Gu, Zhiwei Jia, Hao Tang, and Hao Su. Refactoring policy for compositional generalizability using self-supervised object proposals, 2020. 3

穆同舟, 顾佳源, 贾志伟, 唐浩, 苏浩. 使用自监督对象提议重构策略以实现组合泛化性, 2020. 3

[32] Kieran Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-pdf: Nonparametric representation of probability distributions on the rotation manifold, 2022. 14

Kieran Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, Ameesh Makadia. Implicit-pdf: 旋转流形上概率分布的非参数表示, 2022. 14

[33] Anusha Nagabandi, Kurt Konoglie, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation, 2019. 2

Anusha Nagabandi, Kurt Konoglie, Sergey Levine, Vikash Kumar. 用于学习灵巧操作的深度动力学模型, 2019. 2

[34] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2019. 2

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, Wojciech Zaremba. 学习灵巧的手内操作, 2019. 2

[35] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdi-nov. Actor-mimic: Deep multitask and transfer reinforcement learning, 2016. 3

Emilio Parisotto, Jimmy Lei Ba, Ruslan Salakhutdi-nov. Actor-mimic: 深度多任务和迁移强化学习, 2016. 3

[36] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for classification and segmentation, 2017. 5, 12

Charles R. Qi, 苏浩, 莫开春, Leonidas J. Guibas. Pointnet: 点集上的深度学习用于 分类和分割, 2017. 5, 12

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

秦宇哲, 黄炳豪, 尹兆恒, 苏浩, 王晓龙. Dexpoint: 可泛化的点云强化学习用于仿真到现实的灵巧操作, 2022. 2

[38] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Rui-han Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2022.2 ,3

秦宇哲, 吴越华, 刘绍伟, 江汉文, 杨瑞涵, 付阳, 王晓龙. Dexmv: 从人类视频中进行灵巧操作的模仿学习, 2022.2 ,3

[39] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation, 2021. 3

伊利亚·拉多萨维奇, 王晓龙, 勒雷尔·平托, 吉滕德拉·马利克. 仅状态模仿学习用于灵巧操作, 2021. 3

[40] 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, 2018. 2, 6, 14

阿拉温德·拉杰斯瓦兰, 维卡什·库马尔, 阿布舍克·古普塔, 朱利亚·韦扎尼, 约翰·舒尔曼, 伊曼纽尔·托多罗夫, 谢尔盖·莱文. 通过深度强化学习和示范学习复杂的灵巧操作, 2018. 2, 6, 14

[41] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. 3, 14

斯蒂芬·罗斯, 杰弗里·J·戈登, J·安德鲁·巴格内尔. 将模仿学习和结构化预测简化为无遗憾在线学习, 2011. 3, 14

[42] Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gul-cehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016. 3

安德烈·A·鲁苏, 塞尔吉奥·戈麦斯·科尔梅纳雷霍, 查拉尔·古尔切雷, 纪尧姆·德雅尔丹, 詹姆斯·柯克帕特里克, 拉兹万·帕斯坎, 弗拉基米尔·姆尼赫, 科拉伊·卡武克乔格鲁, 拉亚·哈德塞尔. 策略蒸馏, 2016. 3

[43] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 4, 6, 11

约翰·舒尔曼, 菲利普·沃尔斯基, 普拉富拉·达里瓦尔, 亚历克·拉德福德, 奥列格·克利莫夫. 近端策略优化算法, 2017. 4, 6, 11

[44] ShadowRobot. https://www.shadowrobot.com/ dexterous-hand-series,2005. 1,4,11

ShadowRobot. https://www.shadowrobot.com/ 灵巧手系列,2005. 1,4,11

[45] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes, 2021. 2

马丁·桑德迈尔, 阿尔萨兰·穆萨维安, 鲁道夫·特里贝尔, 迪特尔·福克斯. Contact-graspnet: 在杂乱场景中高效生成6自由度抓取, 2021. 2

[46] Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning, 2017. 3

叶伟, 维克多·巴普斯特, 沃伊切赫·马里安·恰尔内茨基, 约翰·权, 詹姆斯·柯克帕特里克, 拉亚·哈德塞尔, 尼古拉斯·希斯, 拉兹万·帕斯坎. Distral: 鲁棒的多任务强化学习, 2017. 3

[47] Dongying Tian, Xiangbo Lin, and Yi Sun. Adaptive motion planning for multi-fingered functional grasp via force feedback, 2024. 2

田东英, 林祥波, 孙毅. 通过力反馈进行多指功能抓取的自适应运动规划, 2024. 2

[48] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation, 2018. 3

法拉兹·托拉比, 加勒特·沃内尔, 彼得·斯通. 从观察中进行行为克隆, 2018. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2, 13

阿什什·瓦斯瓦尼, 诺姆·沙泽尔, 尼基·帕尔马, 雅各布·乌斯科雷特, 利昂·琼斯, 艾丹·N·戈麦斯, 卢卡什·凯撒, 伊利亚·波洛苏金. 注意力就是一切, 2023. 2, 13

[50] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning, 2023.1,2,3,6,8,11,14

万伟康, 耿浩然, 刘云, 单子康, 杨耀东, 李毅, 王赫. Unidexgrasp++: 通过几何感知课程和迭代通用-专家学习改进灵巧抓取策略学习, 2023.1,2,3,6,8,11,14

[51] Jun Wang, Ying Yuan, Haichuan Che, Haozhi Qi, Yi Ma, Jitendra Malik, and Xiaolong Wang. Lessons from learning to spin "pens", 2024. 2, 3

王俊, 袁颖, 车海川, 齐浩志, 马毅, 吉滕德拉·马利克, 王晓龙. 从学习转“笔”中获得的教训, 2024. 2, 3

[52] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation, 2023. 1, 2, 12

王瑞成, 张家亮, 陈佳怡, 徐寅珍, 李浦浩, 刘腾宇, 王赫. Dexgraspnet: 基于仿真的通用物体大规模机器人灵巧抓取数据集, 2023. 1, 2, 12

[53] Wenbo Wang, Gen Li, Miguel Zamora, and Stelian Coros. Trtm: Template-based reconstruction and target-oriented manipulation of crumpled cloths, 2024. 2

王文博, 李根, Miguel Zamora, 和 Stelian Coros. TRTM: 基于模板的褶皱布料重建与目标导向操作, 2024. 2

[54] Abraham Itzhak Weinberg, Alon Shirizly, Osher Azulay, and Avishai Sintov. Survey of learning-based approaches for robotic in-hand manipulation, 2024. 1, 2

Abraham Itzhak Weinberg, Alon Shirizly, Osher Azulay, 和 Avishai Sintov. 基于学习的机器人手内操作方法综述, 2024. 1, 2

[55] Bohan Wu, Iretiayo Akinola, Jacob Varley, and Peter Allen. Mat: Multi-fingered adaptive tactile grasping via deep reinforcement learning, 2019. 2

吴博涵, Iretiayo Akinola, Jacob Varley, 和 Peter Allen. MAT: 通过深度强化学习实现多指自适应触觉抓取, 2019. 2

[56] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance, 2022. 2, 6, 14

吴岳华, 王嘉顺, 和汪小龙. 从人类抓取能力中学习通用灵巧操作, 2022. 2, 6, 14

[57] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yi-jia 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. 1, 2, 3, 6, 14

徐寅臻, 万伟康, 张嘉良, 刘浩然, 山子康, 沈浩, 王瑞成, 耿浩然, 翁一佳, 陈佳怡, 刘腾宇, 李毅, 和王贺. UniDexGrasp: 通过多样化提案生成和目标条件策略学习实现通用机器人灵巧抓取, 2023. 1, 2, 3, 6, 14

[58] Hanbo Zhang, Jian Tang, Shiguang Sun, and Xuguang Lan. Robotic grasping from classical to modern: A survey, 2022. 2

张汉波, 唐健, 孙世光, 和兰旭光. 机器人抓取从经典到现代: 综述, 2022. 2

[59] Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. Artigrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation. In 2024 International Conference on Vision (3DV), pages 235-246, 2024. 2

张辉, Sammy Christen, 范子聪, 郑罗成, Jemin Hwangbo, 宋杰, 和 Otmar Hilliges. ArtiGrasp: 物理上可信的双手机器人灵巧抓取与关节合成. 在2024年国际 视觉会议(3DV)上, 第235-246页, 2024. 2

[60] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. In Computer Vision - ECCV 2024, pages 386-403, Cham, 2025. Springer Nature Switzerland. 2

张辉, Sammy Christen, 范子聪, Otmar Hilliges, 和宋杰. GraspXL: 大规模生成多样化物体的抓取动作. 在计算机视觉 - ECCV 2024上, 第386-403页, Cham, 2025. Springer Nature Switzerland. 2

[61] Zhengshen Zhang, Lei Zhou, Chenchen Liu, Zhiyang Liu, Chengran Yuan, Sheng Guo, Ruiteng Zhao, Marcelo H. Ang Jr. au2, and Francis EH Tay. Dexgrasp-diffusion: Diffusion-based unified functional grasp synthesis method for multi-dexterous robotic hands, 2024. 2

张正申, 周磊, 刘晨晨, 刘志阳, 袁成然, 郭升, 赵瑞腾, Marcelo H. Ang Jr. au2, 和 Francis EH Tay. DexGrasp-Diffusion: 基于扩散的多灵巧机器人手统一功能抓取合成方法, 2024. 2

[62] Lei Zhou, Haozhe Wang, Zhengshen Zhang, Zhiyang Liu, Francis E.H. Tay, and Marcelo H. Ang. You only scan once: A dynamic scene reconstruction pipeline for 6-dof robotic grasping of novel objects. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 13891- 13897, 2024. 2

周磊, 王浩哲, 张正申, 刘志阳, Francis E.H. Tay, 和 Marcelo H. Ang. 你只需扫描一次: 用于6自由度机器人抓取新物体的动态场景重建管道. 在2024年IEEE国际机器人与自动化会议(ICRA)上, 第13891-13897页, 2024. 2

# UniGraspTransformer: Simplified Policy Distillation for Scalable Dexterous Robotic Grasping

# UniGraspTransformer: 简化策略蒸馏以实现可扩展的灵巧机器人抓取

Supplementary Material

补充材料

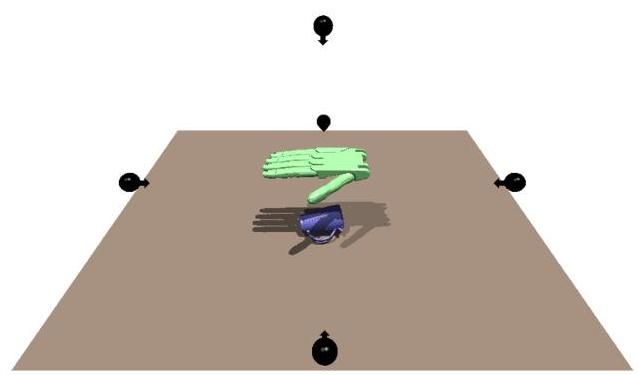


Figure 6. Illustration of the simulation environment.

图6. 仿真环境示意图。

# A. Implementation Details

# A. 实现细节

# A.1. Environment Setup

# A.1. 环境设置

Initialization. We use Isaac Gym 3.0 [27] to build simulation environments, each containing a table (brown), an object placed on top (blue), a controllable Shadow Hand (green) [44], and five surrounding cameras (black), as illustrated in Figure 6. The system’s origin is defined at the center of the table, where all objects are initially placed. The Shadow Hand is positioned 0.2 meters above the table center, with the goal located 0.3 meters above the table center.

初始化。我们使用Isaac Gym 3.0 [27]构建仿真环境，每个环境包含一张桌子(棕色)、一个放置在桌子上的物体(蓝色)、一个可控的Shadow Hand(绿色)[44]和五个周围的摄像头(黑色)，如图6所示。系统的原点定义在桌子的中心，所有物体最初都放置在此处。Shadow Hand位于桌子中心上方0.2米处，目标位于桌子中心上方0.3米处。

For each object utilized in our project, we randomly drop it onto the table with arbitrary rotations to generate a dataset comprising static tabletop poses. This dataset is divided into three subsets for specific purposes: poses for dedicated policy training, poses for offline trajectory generation, and poses for evaluation.

对于我们项目中使用的每个物体，我们随机将其以任意旋转方式放置在桌面上，生成包含 个静态桌面姿态的数据集。该数据集被分为三个子集用于特定目的: 个姿态用于专用策略训练， 个姿态用于离线轨迹生成，以及 个姿态用于评估。

Task Definition. The objective is to develop a robust universal policy capable of controlling the Shadow Hand [44] to grasp and transport a diverse range of tabletop objects to a designated midair goal position. Each grasping consists of 200 execution steps and is deemed successful if the positional difference between the object and the goal remains within a predefined threshold by the end of the sequence.

任务定义。目标是开发一个强大的通用策略，能够控制Shadow Hand [44]抓取并运输各种桌面物体到指定的空中目标位置。每次抓取由200个执行步骤组成，如果序列结束时物体与目标之间的位置差异保持在预定义的阈值内，则视为成功。

Observation Space. At each simulation step, the observation space of state-based UniGraspTransformer includes a 167-d proprioceptive state of the hand, a 24-d representation of the hand’s previous action, a 16-d object state, a 128-d object visual feature, a 36-d hand-to-object distance, and a 29-d time embedding, as detailed in Table 1 of the main paper. During dedicated RL policy training, the 128- d object visual feature is excluded to enhance training efficiency. For vision-based UniGraspTransformer training, the original 16-d object state is replaced by the center position of the partial object point cloud (3-d) and its three principal component axes (3 3-d). Additionally, we compute 36 distances between 36 selected points on the Shadow Hand and the partial object point cloud, as illustrated in Figure 7(c).

观察空间。在每个模拟步骤中，基于状态的UniGraspTransformer的观察空间包括手的167维本体感觉状态、手的24维先前动作表示、16维物体状态、128维物体视觉特征、36维手到物体距离以及29维时间嵌入，如主论文表1所述。在专用RL策略训练期间，128维物体视觉特征被排除以提高训练效率。对于基于视觉的UniGraspTransformer训练，原始的16维物体状态被部分物体点云的中心位置(3维)及其三个主成分轴(3 3维)取代。此外，我们计算了Shadow Hand上36个选定点与部分物体点云之间的36个距离，如图7(c)所示。

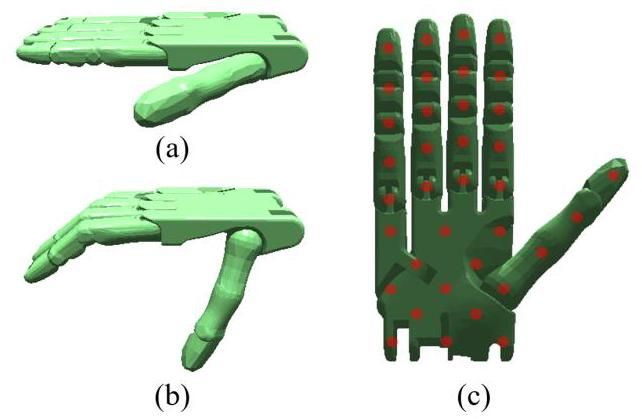


Figure 7. Shadow Hand poses. (a) Initial pose at the first frame. (b) Pre-contact opening pose used in dedicated policy training. (c) 36 selected hand points for computing hand to object distance.

图7. Shadow Hand姿态。(a) 第一帧的初始姿态。(b) 专用策略训练中使用的预接触张开姿态。(c) 用于计算手到物体距离的36个选定手点。

Action Space. The action space comprises motor commands for 24 actuators on the Shadow Hand. The first 6 actuators manage the wrist’s position and orientation through applied forces and torques, while the remaining 18 actuators control the positions of the finger joints. The action values are normalized to a range of -1 to 1 according to the specifications of the actuators.

动作空间。动作空间包括Shadow Hand上24个执行器的电机命令。前6个执行器通过施加的力和扭矩管理手腕的位置和方向，而剩余的18个执行器控制手指关节的位置。动作值根据执行器的规格归一化到-1到1的范围。

Camera Setup. Following a similar approach to UniDex-Grasp++ [50], five RGBD cameras are mounted around the table, as illustrated in Figure 6. The cameras are positioned relative to the table center at coordinates(0.0,0.0,0.55), , and , , with their focal points aligned at(0,0,0.15). In the vision-based setting, the depth images captured by these cameras are fused to generate a scene point cloud, from which the partial point cloud of the object is segmented.

相机设置。采用与UniDex-Grasp++ [50]类似的方法，在桌子周围安装了五个RGBD相机，如图6所示。相机相对于桌子中心的位置坐标为(0.0,0.0,0.55)， ，以及 ， ，其焦点对齐在(0,0,0.15)。在基于视觉的设置中，这些相机捕获的深度图像被融合以生成场景点云，从中分割出物体的部分点云。

# A.2. Dedicated Policy Training

# A.2. 专用策略训练

PPO. Proximal Policy Optimization [43] is a widely used model-free, on-policy reinforcement learning algorithm that simultaneously learns a policy and estimates a value function. We utilize PPO to train dedicated RL policies for each of the 3,200 objects. Both the policy and value networks are implemented as 4-layer MLPs with hidden dimensions of . At each simulation step, the policy network takes the current observation as input and outputs a 24-d action, while the value network predicts a 1-d value. The simulation environment then executes the action, updates the observation, and calculates the corresponding reward. The policy and value networks are updated every 16 simulation steps using the collected observations, actions, values, and rewards. Each dedicated RL policy is trained on an NVIDIA V100 GPU for a total of 10,000 update iterations, taking approximately 3 hours to complete.

PPO。近端策略优化[43]是一种广泛使用的无模型、在策略强化学习算法，它同时学习策略并估计价值函数。我们使用PPO为3,200个物体中的每一个训练专用RL策略。策略和价值网络均实现为4层MLP，隐藏维度为 。在每个模拟步骤中，策略网络将当前观察作为输入并输出24维动作，而价值网络预测1维值。模拟环境然后执行动作，更新观察，并计算相应的奖励。策略和价值网络每16个模拟步骤使用收集的观察、动作、价值和奖励进行更新。每个专用RL策略在NVIDIA V100 GPU上训练总共10,000次更新迭代，大约需要3小时完成。

Reward Function. The reward function described in Eq.(1) of the main paper comprises five components: , , and . These reward components are governed by a contact flag , which indicates whether the hand is in contact with the object.

奖励函数。主论文中公式(1)描述的奖励函数包括五个组成部分: ， ，以及 。这些奖励组成部分由接触标志 控制，该标志指示手是否与物体接触。

The distance reward penalizes the average Chamfer Distance between the hand points and the object point cloud , promoting contact and encouraging the hand to maintain a secure grasp on the object’s surface. Specifically, 36 points are selected on the Shadow Hand for this calculation, as illustrated in Figure 7(c).

距离奖励 对手点 和物体点云 之间的平均倒角距离进行惩罚，促进接触并鼓励手部保持对物体表面的稳固抓握。具体来说，在Shadow Hand上选择了36个点 进行计算，如图7(c)所示。

where the reward weight is set to 1.0 .

其中奖励权重 设置为1.0。

The contact flag is set to 1 if the average Chamfer Distance between the hand points and the object point cloud falls below a predefined threshold . This is determined as follows:

如果手点和物体点云之间的平均倒角距离低于预定义阈值 ，则接触标志 设置为1。具体确定如下:

(3)

where denotes the indicator function.

其中 表示指示函数。

Inspired by DexGraspNet [52], before the contact is established, the opening reward penalizes deviations of the current hand pose from a predefined opening pose , as depicted in Figure 7(b). This encourages the hand to remain open until it makes contact with the object. The reward is calculated as:

受DexGraspNet [52] 启发，在接触建立之前，张开奖励 对当前手部姿态 与预定义张开姿态 的偏差进行惩罚，如图7(b)所示。这鼓励手部在接触物体之前保持张开状态。奖励计算如下:

where the reward weight is set to 0.1 .

其中奖励权重 设置为0.1。

Once contact is established, the rewards , and are introduced to guide the grasping process:

一旦接触建立，奖励 和 被引入以引导抓取过程:

* Lift reward : This reward encourages the hand to perform a lifting action along the axis:
* 提升奖励 :该奖励鼓励手部沿 轴执行提升动作 :

where is set to 0.1 .

其中 设置为0.1。

* Goal reward : This reward penalizes the Euclidean distance between the object center position and the target goal position :
* 目标奖励 :该奖励惩罚物体中心位置 与目标位置 之间的欧几里得距离:

where is set to 2.0 .

其中 设置为 2.0。

* Success reward : This reward provides a bonus when the object successfully reaches the goal position, defined by a threshold :
* 成功奖励 :当物体成功到达目标位置时，该奖励提供额外奖励，由阈值 定义:

where is set to 1.0 .

其中 设置为 1.0。

# A.3. Grasp Trajectory Generation

# A.3. 抓取轨迹生成

Our 3,200 dedicated RL policies achieve an average success rate of across all 3,200 training objects. For each object, we randomly initialize it in diverse poses and apply its corresponding RL policy to generate successful trajectories, which are used for offline training of the UniGraspTransformer model. Each trajectory, , consists of a sequence of steps. Here, represents robotic action at timestep- , and captures the environment state, including proprioception (167-d), previous action (24-d), object state (16-d), hand-object distance (36-d), and time embedding (29-d), as detailed in Table 1 of the main paper. Additionally, we save the complete object point cloud - d) and the partial object point cloud , which are used to generate object features to train the state-based and vision-based versions of the UniGraspTransformer model.

我们的 3,200 个专用 RL 策略在所有 3,200 个训练对象上实现了平均成功率 。对于每个对象，我们随机初始化其在不同姿态中，并应用其对应的 RL 策略生成 条成功轨迹，这些轨迹用于 UniGraspTransformer 模型的离线训练。每条轨迹 由一系列步骤组成。其中， 表示时间步 的机器人动作， 捕捉环境状态，包括本体感知(167 维)、前一动作(24 维)、物体状态(16 维)、手-物体距离(36 维)和时间嵌入(29 维)，如主论文表 1 所述。此外，我们保存完整的物体点云 和部分物体点云 ，这些点云用于生成物体特征，以训练基于状态和基于视觉的 UniGraspTransformer 模型。

# A.4. Point Cloud Encoder Training

# A.4. 点云编码器训练

S-Encoder. To train our S-Encoder, we use a dataset consisting of 3,200 point clouds of seen objects, denoted as , where each represents the canonical point cloud of a specific object. During each training iteration, a batch of 100 object point clouds is randomly sampled from this dataset. For each point cloud in the batch, indexed by , the centroid is subtracted to center the point cloud, followed by the application of a random rotation matrix . The resulting transformed point cloud is expressed as , which serves as input to the S-Encoder.

S-编码器。为了训练我们的 S-编码器，我们使用了一个包含 3,200 个已见物体点云的数据集，记为 ，其中每个 表示特定物体的规范点云。在每次训练迭代中，从该数据集中随机抽取 100 个物体点云作为批次。对于批次中的每个点云，索引为 ，减去质心 以将点云居中，然后应用随机旋转矩阵 。生成的变换点云表示为 ，作为 S-编码器的输入。

The S-Encoder, as part of an encoder-decoder framework [36], processes to produce a latent feature . This latent feature is then passed to the decoder, which reconstructs the point cloud, yielding . The model is trained by minimizing the reconstruction loss , defined as the Chamfer Distance between the original transformed point cloud and its reconstruction :

S-编码器作为编码器-解码器框架 [36] 的一部分，处理 以生成潜在特征 。该潜在特征随后传递给解码器，解码器重建点云，生成 。模型通过最小化重建损失 进行训练，重建损失定义为原始变换点云 与其重建 之间的 Chamfer 距离:

The S-Encoder is trained for iterations on an NVIDIA A100 GPU. After training, the state-based object features are generated by encoding the complete object point clouds using the trained S-Encoder.

S-编码器在 NVIDIA A100 GPU 上训练 次迭代。训练完成后，通过使用训练好的 S-编码器对完整物体点云进行编码，生成基于状态的物体特征。

V-Encoder. The V-Encoder is trained using a knowledge distillation approach, leveraging the pre-trained S-Encoder and the grasp trajectories generated by the dedicated RL policies, as illustrated in Figure 8. In each training iteration, a batch of 100 steps is randomly sampled from the generated trajectories. Both the complete object point cloud and the partial point cloud are centered by subtracting their mean positions. The centered complete point cloud is passed through the pre-trained S-Encoder (with frozen weights), producing a latent feature . Simultaneously, the centered partial point cloud is fed to the V-Encoder, which outputs both a latent feature and a reconstructed point cloud .

V-编码器。V-编码器通过知识蒸馏方法进行训练，利用预训练的S-编码器和由专用强化学习策略生成的抓取轨迹 ，如图8所示。在每次训练迭代中，从生成的轨迹中随机抽取100步的批次。完整的物体点云 和部分点云 通过减去它们的平均位置进行中心化处理。中心化后的完整点云 通过预训练的S-编码器(权重冻结)进行处理，生成潜在特征 。同时，中心化的部分点云 被输入到V-编码器中，输出潜在特征 和重建的点云 。

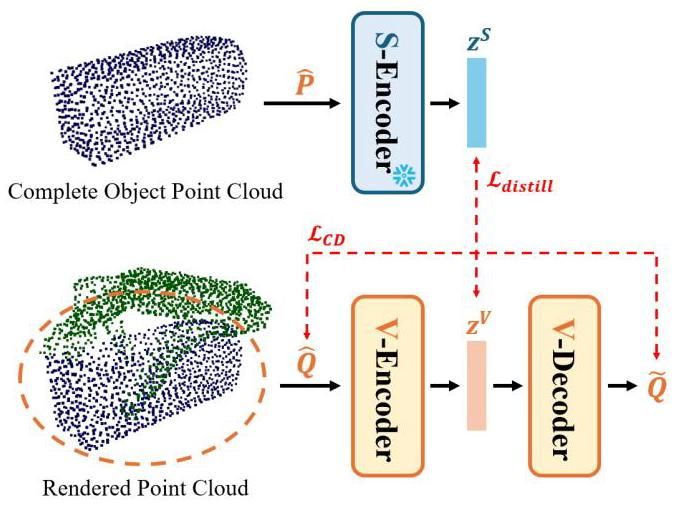


Figure 8. V-Encoder training with distillation.

图8. 使用蒸馏方法训练V-编码器。

The V-Encoder is optimized using two loss functions:

V-编码器通过两个损失函数进行优化:

* Feature Distillation loss ( ): This L2 loss measures the difference between the latent features produced by the S-Encoder and the V-Encoder:
* 特征蒸馏损失( ):此L2损失衡量S-编码器和V-编码器生成的潜在特征之间的差异:
* Reconstruction loss : This is the Chamfer Distance between the centered partial point cloud and its reconstruction :
* 重建损失 :这是中心化部分点云 与其重建 之间的Chamfer距离:

The total loss for training the V-Encoder is defined as:

训练V-编码器的总损失定义为:

where the weights are set to and 0.1. The V-Encoder is trained on an NVIDIA A100 GPU for iterations. After training, the vision-based object features are generated by encoding the partial object point clouds using the trained V-Encoder.

其中权重设置为 和 0.1。V-编码器在NVIDIA A100 GPU上训练 次迭代。训练完成后，通过使用训练好的V-编码器对部分物体点云进行编码，生成基于视觉的物体特征。

# A.5. UniGraspTransformer Training

# A.5. UniGraspTransformer训练

Input Types. The UniGraspTransformer is trained using the generated grasp trajectories and encoded object features under two configurations, as outlined in Table 11:

输入类型。UniGraspTransformer在两种配置下使用生成的抓取轨迹和编码的物体特征进行训练，如表11所示:

Input of UniGraspTransformer

UniGraspTransformer的输入

|  |  |
| --- | --- |
| State-Based | Vision-Based |
| Proprioception (167) | Proprioception (167) |
| Previous Action (24) | Previous Action (24) |
| Object State (16) | Object State\* (12) |
| Object Feature (128) | Object Feature\* (128) |
| Hand-Obj. Dist. (36) | Hand-Obj. Dist.\* (36) |
| Time (29) | Time (29) |

|  |  |
| --- | --- |
| 基于状态 | 基于视觉 |
| 本体感觉 (167) | 本体感觉 (167) |
| 先前动作 (24) | 先前动作 (24) |
| 物体状态 (16) | 物体状态\* (12) |
| 物体特征 (128) | 物体特征\* (128) |
| 手-物距离 (36) | 手-物距离\* (36) |
| 时间 (29) | 时间 (29) |

Table 11. Input types for state-based and vision-based UniGrasp-Transformer, organized into six groups.

表11. 基于状态和基于视觉的UniGrasp-Transformer的输入类型，分为六组。

* State-Based Setting: The complete object point clouds are assumed to be perfectly accurate and are encoded using the S-Encoder. Object states, including positions, rotations, and velocities, are directly accessible, as detailed in Table 1 of the main paper.
* 基于状态的设置:假设完整的物体点云完全准确，并使用S-Encoder进行编码。物体的状态，包括位置、旋转和速度，可以直接访问，如主论文表1所述。
* Vision-Based Setting: Partial object point clouds are reconstructed and segmented from depth data captured by five cameras mounted above and around the table. These object features are encoded using the V-Encoder, and object states are estimated rather than directly accessed.
* 基于视觉的设置:从安装在桌子上方和周围的五个摄像头捕获的深度数据中重建和分割部分物体点云。这些物体特征使用V-Encoder进行编码，物体的状态是估计的，而不是直接访问的。

The key differences between the inputs for the state-based and vision-based UniGraspTransformer are:

基于状态和基于视觉的UniGraspTransformer输入的主要区别是:

* For the object state representation, the vision-based setting uses the center of the partial object point cloud (3-d) as the object position and three principal component axes (9-d) to represent object orientation.
* 对于物体状态表示，基于视觉的设置使用部分物体点云的中心(3维)作为物体位置，并使用三个主成分轴(9维)来表示物体方向。
* The object feature is derived from the partial object point cloud and encoded using the V-Encoder in the vision-based setting.
* 物体特征是从部分物体点云中提取的，并在基于视觉的设置中使用V-Encoder进行编码。
* The hand-object distance is computed using the partial object point cloud in the vision-based setting.
* 在基于视觉的设置中，使用部分物体点云计算手与物体的距离。

Training Process. Each trajectory step consists of six observation groups, as detailed in Table 11, paired with a ground truth action . The UniGraspTransformer processes these observations as follows: (1) The six observation groups are converted into six 256-dimensional tokens using individual single-layer MLPs; (2) These tokens are passed through 12 self-attention layers [49], producing six refined 256-dimensional features; (3) The six features are concatenated into a single 1536-dimensional representation, which is then processed by a 4-layer MLP to predict the final 24-d action . The model is optimized using a single L2 loss, defined as: .

训练过程。每个轨迹步骤由六个观察组组成，如表11所述，与一个真实动作 配对。UniGraspTransformer按以下方式处理这些观察:(1) 使用单个单层MLP将六个观察组转换为六个256维的token；(2) 这些token通过12个自注意力层[49]，生成六个精炼的256维特征；(3) 将六个特征连接成一个1536维的表示，然后通过一个4层MLP处理，以预测最终的24维动作 。模型使用单个L2损失进行优化，定义为: 。

Training is conducted on a dataset of 3,200 objects with 3.2 million trajectories, using a batch size of 800 trajectories (each with 200 steps) over 100 epochs. The process is carried out on 8 NVIDIA A100 GPUs and takes approximately 70 hours to complete. The average L2 loss at convergence is around 0.015 .

训练在包含3200个物体和320万条轨迹的数据集上进行，使用800条轨迹(每条轨迹有200步)的批量大小，训练100个epoch。该过程在8个NVIDIA A100 GPU上进行，大约需要70小时完成。收敛时的平均L2损失约为0.015。

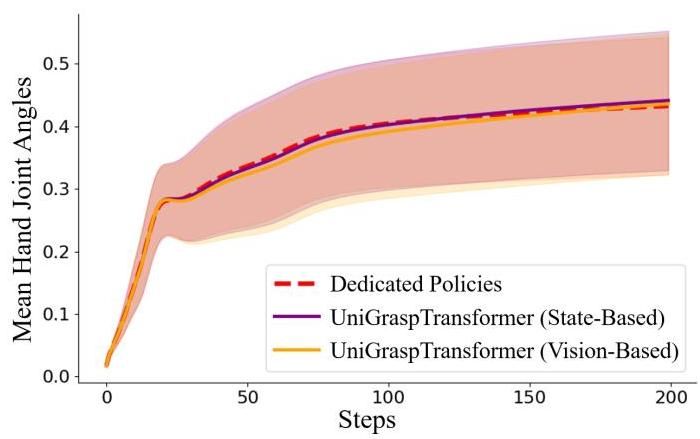


Figure 9. Quantitative analysis of grasp pose diversity.

图9. 抓取姿势多样性的定量分析。

# B. Experiment Details

# B. 实验细节

# B.1. Baseline Methods

# B.1. 基线方法

The implementation of baseline methods listed in Table 2 of the main paper is outlined below. Additional details can be found in UniDexGrasp++[50].

主论文表2中列出的基线方法的实现概述如下。更多细节可以在UniDexGrasp++[50]中找到。

PPO. This reinforcement learning baseline directly trains a state-based universal model using PPO with all training objects. The vision-based universal policy is derived from the state-based policy through distillation using DAgger [41].

PPO。这个强化学习基线直接使用PPO与所有训练对象训练一个基于状态的通用模型。基于视觉的通用策略通过使用DAgger[41]从基于状态的策略中蒸馏得到。

DAPG. Demo Augmented Policy Gradient (DAPG) [40] is a widely used imitation learning method that leverages expert demonstrations to reduce RL sampling complexity. In this baseline, grasp trajectories generated via motion planning serve as demonstrations to train a state-based deep RL policy. The vision-based universal policy is then distilled from the state-based policy using DAgger [41].

DAPG。演示增强策略梯度(DAPG)[40]是一种广泛使用的模仿学习方法，利用专家演示来减少RL采样复杂性。在这个基线中，通过运动规划生成的抓取轨迹作为演示来训练一个基于状态的深度RL策略。然后使用DAgger[41]从基于状态的策略中蒸馏出基于视觉的通用策略。

ILAD. ILAD [56] enhances the generalization capabilities of DAPG [40] by introducing an imitation learning objective focused on the object’s geometric representation. In this baseline, a pipeline similar to DAPG [40] is implemented.

ILAD。ILAD[56]通过引入一个专注于物体几何表示的模仿学习目标，增强了DAPG[40]的泛化能力。在这个基线中，实现了一个类似于DAPG[40]的管道。

GSL. Generalist-Specialist Learning (GSL) [13] begins by training a generalist policy using PPO over the entire task space. Specialists are then fine-tuned to master each subset of the task space. The final generalist is trained using DAPG [40], leveraging demonstrations generated by the trained specialists.

GSL。通用-专家学习(GSL)[13]首先使用PPO在整个任务空间上训练通用策略。然后对专家进行微调，以掌握任务空间的每个子集。最终的通用策略使用DAPG [40]进行训练，利用由训练好的专家生成的演示。

UniDexGrasp. UniDexGrasp [57] decomposes the grasping task into two stages: static grasp pose generation followed by dynamic grasp execution via goal-conditioned reinforcement learning. First, an IPDF-based [32] grasp pose generator is trained using all training objects. An Object Curriculum Learning protocol is then applied, starting reinforcement learning with a single object and gradually incorporating similar objects to train a state-based universal policy. Finally, DAgger [41] is used to distill the state-based universal policy into a vision-based universal policy.

UniDexGrasp。UniDexGrasp [57]将抓取任务分解为两个阶段:静态抓取姿态生成，然后通过目标条件强化学习进行动态抓取执行。首先，使用所有训练对象训练基于IPDF [32]的抓取姿态生成器。然后应用对象课程学习协议，从单个对象开始进行强化学习，并逐渐引入相似对象来训练基于状态的通用策略。最后，使用DAgger [41]将基于状态的通用策略提炼为基于视觉的通用策略。

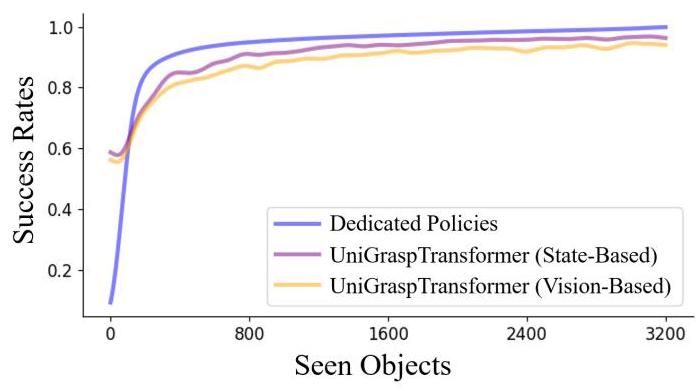


Figure 10. Success rates across seen objects.

图10。已见对象的成功率。

UniDexGrasp++. UniDexGrasp++ [50] builds on the Generalist-Specialist Learning framework by integrating geometry-based clustering during specialist training, where each specialist focuses on a group of geometrically similar objects. Additionally, it introduces a generalist-specialist iterative process in which specialists are repeatedly trained from the generalist, followed by generalist distillation.

UniDexGrasp++。UniDexGrasp++ [50]在通用-专家学习框架的基础上，通过在专家训练期间集成基于几何的聚类，每个专家专注于一组几何相似的对象。此外，它引入了通用-专家迭代过程，其中专家从通用策略中反复训练，然后进行通用策略提炼。

# C. More Analysis

# C. 更多分析

From Dedicated to Universal. Our 3,200 dedicated RL policies achieve an average success rate of 94.1% across all 3,200 training objects. In comparison, the UniGraspTrans-former achieves success rates of on 3,200 seen objects, 89.2% (87.3%) on 140 unseen objects from seen categories, and 88.3% (86.8%) on 100 unseen objects from unseen categories under the state-based (vision-based) settings, respectively.

从专用到通用。我们的3,200个专用RL策略在所有3,200个训练对象上实现了94.1%的平均成功率。相比之下，UniGraspTransformer在3,200个已见对象上实现了 的成功率，在来自已见类别的140个未见对象上实现了89.2%(87.3%)的成功率，在来自未见类别的100个未见对象上实现了88.3%(86.8%)的成功率，分别在基于状态(基于视觉)的设置下。

As depicted in Figure 9, the UniGraspTransformer effectively replicates the grasping trajectories generated by the dedicated RL policies through offline distillation. While there is a minor performance drop from 94.1% to 91.2% (88.9%) in the state-based (vision-based) setting, as illustrated in Figure 10, the model demonstrates robust generalization and efficiency.

如图9所示，UniGraspTransformer通过离线蒸馏有效地复制了由专用RL策略生成的抓取轨迹。虽然在基于状态(基于视觉)的设置下，性能从94.1%略微下降到91.2%(88.9%)，如图10所示，该模型展示了强大的泛化能力和效率。

Qualitative Results. The progressive online distillation approach [13] employed in UniDexGrasp++[50] results in a universal policy that tends to grasp different objects using similar poses. In contrast, our UniGraspTransformer, utilizing a larger model and an offline distillation framework, demonstrates the ability to grasp objects of various shapes with a wide range of diverse poses. This increased diversity in grasping strategies is further highlighted in Figure 11.

定性结果。UniDexGrasp++ [50]中采用的渐进式在线蒸馏方法[13]产生了一种通用策略，倾向于使用相似的姿态抓取不同的对象。相比之下，我们的UniGraspTransformer利用更大的模型和离线蒸馏框架，展示了能够以多种不同的姿态抓取各种形状对象的能力。这种抓取策略的多样性在图11中得到了进一步强调。

Real-World Deployment. We extend the deployment of our vision-based UniGraspTransformer to a real-world environment using the Inspire Hand [12], which features six active DoFs for its fingers. The training process remains identical to that used for the Shadow Hand. Demonstration videos showcasing grasping across 12 distinct objects are provided in the supplementary materials.

实际部署。我们使用Inspire Hand [12]将基于视觉的UniGraspTransformer扩展到实际环境中，该手具有六个主动自由度。训练过程与用于Shadow Hand的过程相同。展示抓取12个不同对象的演示视频在补充材料中提供。

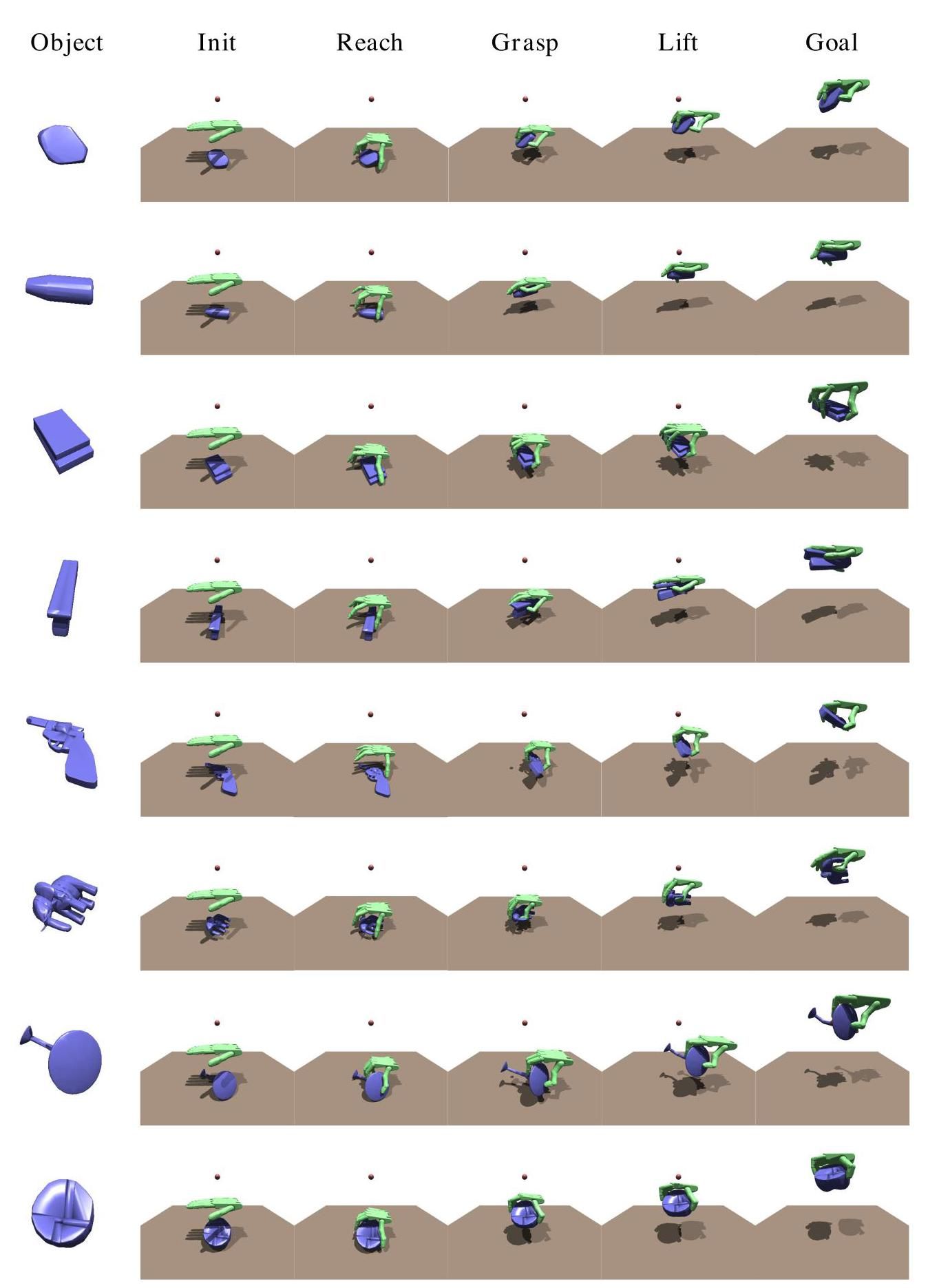


Figure 11. Qualitative analysis of the grasp pose diversity achieved by UniGraspTransformer.

图11。UniGraspTransformer实现的抓取姿态多样性的定性分析。

1. \*Corresponding author.

   \*通讯作者。 [↑](#footnote-ref-29)