

Grasp to Act: Dexterous Grasping for Tool Use in Dynamic Settings

Harsh Gupta¹ Mohammad Amin Mirzaee¹ Wenzhen Yuan¹
grasp2act.github.io

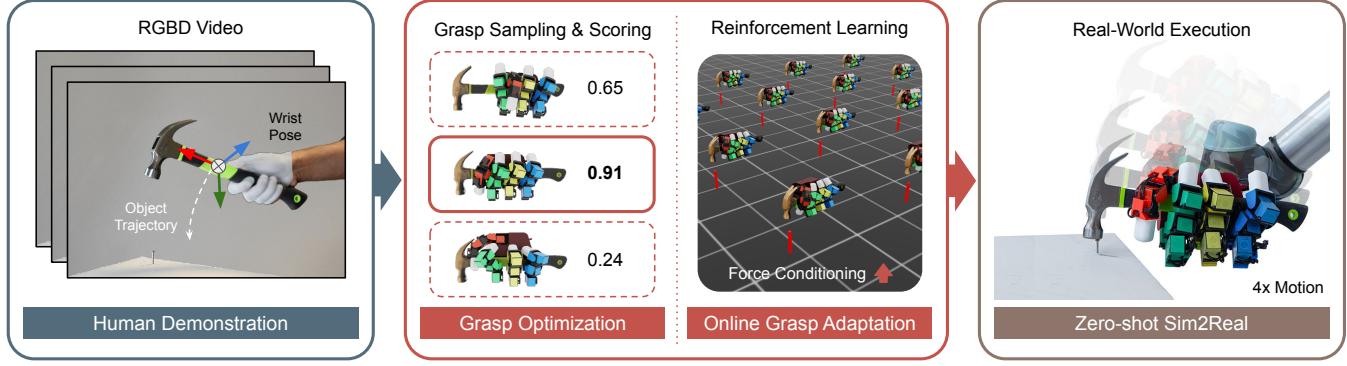


Fig. 1: *Overview of the Grasp-to-Act framework.* (Left) A human demonstration trajectory provides the a reference of the dynamic conditions during tool using, as well as the initial grasping locations. Accordingly, we sample a series of grasps and score them to identify stable grasp candidates (middle left). A reinforcement learning policy then performs online-grasp adaptation by adjusting finger joint positions under task-specific force conditioning (middle right). The resulting policy is zero-shot deployed in the real world (right).

Abstract— Achieving robust grasping with dexterous hands remains challenging, especially when manipulation involves dynamic forces such as impacts, torques, and continuous resistance—situations common in real-world tool use. Existing methods largely optimize grasps for static geometric stability and often fail once external forces arise during manipulation. We present *Grasp-to-Act*, a hybrid system that combines physics-based grasp optimization with reinforcement-learning-based grasp adaptation to maintain stable grasps throughout functional manipulation tasks. Our method synthesizes robust grasp configurations informed by human demonstrations and employs an adaptive controller that residually issues joint corrections to prevent in-hand slip while tracking the object trajectory. *Grasp-to-Act* enables robust zero-shot sim-to-real transfer across five dynamic tool-use tasks—hammering, sawing, cutting, stirring, and scooping—consistently outperforming baselines. Across simulation and real-world hardware trials with a 16-DoF dexterous hand, our method reduces translational and rotational in-hand slip and achieves the highest task completion rates, demonstrating stable functional grasps under dynamic, contact-rich conditions.

I. INTRODUCTION

Grasping is the first step toward manipulation and remains a long-standing challenge in robotics [1]. Most existing work focuses on identifying grasp locations based on object geometry, typically defining success as the ability to lift or hold an object stably in mid-air [2], [3], [4]. Such approaches often neglect the object’s weight and dynamic interactions. In real-world manipulation, however, grasping must account for diverse object properties and task requirements. Tool use exemplifies this challenge—tasks such as hammering a nail or cutting wood with a saw involve heavy objects and asymmetric grasping configurations that generate significant torque on the hand. Moreover, these tasks introduce substantial disturbance forces and torques, whether from object

dynamics (e.g., swinging a hammer) or contact interactions (e.g., impact during nail driving or resistance during sawing). These factors place much higher demands on grasp stability.

In such scenarios, dexterous hands are essential. While low degree of freedom (DoF) grippers—such as parallel-jaw or suction grippers—are widely used, their limited contact points restrict their ability to counteract large torques and maintain stable grasps under strong disturbances. Dexterous hands, in contrast, can form power grasps in which the palm provides a broad contact area to balance torque while the fingers wrap around the object to prevent slipping. Despite their potential, dexterous-hand grasping remains a difficult and relatively underexplored problem. The high dimensionality of these hands makes planning robust grasps challenging, and existing studies primarily emphasize geometric grasp location selection [5], [6], [7], with limited investigation into grasp stability under large external loads.

To tackle these challenges, grasp planning for dexterous hands must explicitly account for grasp stability under realistic conditions, incorporating the effects of object weight, grasp-induced torque, and disturbances from both object dynamics and environmental interactions.

We propose *Grasp-to-Act* (*G2A*), a hybrid framework designed to ensure stable grasps during dynamic, functional tasks. *G2A* introduces a stability evaluation protocol that tests candidate grasps under extreme forces and torques to identify those robust enough for contact-rich manipulation. The protocol is guided by human demonstrations—robots reproduce human motion trajectories during real-world tool-use tasks, while the resulting impedance forces and torques on the objects are modeled to assess grasp stability.

Our *G2A* pipeline consists of two stages for the grasp-to-act process: an initial grasp, where we sample a set of candidate grasp configurations in a physics-based simulation

¹University of Illinois Urbana-Champaign

platform and evaluate their stability under various disturbance forces; an online adaptation phase, when performing the human-demonstrated trajectory, where we develop an RL-based controller to counteract unexpected in-hand motion of the objects.

We validate our method across five representative functional manipulation tasks—hammering, sawing, cutting, stirring, and scooping—using a simulated and real-world robot platform equipped with a LEAP hand [8]. Our experiments show that combining grasp optimization with online-grasp adaptation significantly improves grasp stability and task success compared to baselines that rely solely on static optimization or RL. *G2A* represents a shift from grasping as a static objective to grasping as an enabling step for manipulation, moving toward robots that can perform purposeful, task-driven interaction with the physical world.

II. RELATED WORK

A. Dexterous Grasp Synthesis

Dexterous grasp synthesis aims to determine stable contact configurations for multi-fingered robotic hands given an object mesh, point cloud, or other perceptual input.

Classical approaches [5], [7], [9] optimize analytical metrics such as force closure [6], form closure [10], min-weight metric [11], or Ferrari–Canny metric [12] using sampling-based or gradient-based methods. However, these methods suffer from two key limitations. First, they rely on accurate object models and often make strong contact assumptions [1], such as perfect rigidity, point contacts, zero or uniform friction, and a quasi-static setting (i.e. no contact breakage). Second, they typically optimize only for fingertip contact points that result in precision grasps, overlooking the broader range of grasp taxonomies [13]. In practice, these methods are not applicable to heavy or irregularly shaped objects such as tools. To address these concerns, we build our grasp synthesis pipeline entirely within a physics-based simulator and employ stochastic sampling to generate a diverse set of candidate grasps across functional regions of the object.

Data-driven methods [2], [3], [14], [15], [16] for dexterous grasp synthesis typically rely on large-scale datasets [17], [18] of synthetic grasps paired with perceptual inputs, often generated using the grasp optimization techniques described earlier. Because these optimization frameworks frequently synthesize physically infeasible grasps, many datasets apply a post hoc filtering step using physics-based simulators to discard unstable grasps, typically through a simple 1D or 6D gravitational test. Recently, efforts have focused on retaining and ranking both successful grasps and high-quality failed examples [19], driven by the strong performance of discriminative models for parallel-jaw grasping [20]. Multi-GripperGrasp [21] addresses this by assigning a continuous score based on the object’s fall-off time under gravity. Get a Grip [3] simulates grasps under small perturbations and averages the outcomes to estimate the probability of success. However, both evaluation strategies are limited to testing gravitational stability. Drawing from these methods,

we design a set of rigorous wrench-space stability tests that evaluate our synthesized grasps under extreme forces and torques, enabling us to rank and select the most stable candidates for downstream tasks.

B. Reinforcement Learning for Dexterous Manipulation

Reinforcement learning (RL) has demonstrated strong potential in dexterous manipulation, achieving impressive results in tasks such as pick-and-place, in-hand reorientation, and object relocation [22], [23], [24]. However, these tasks primarily involve isolated object motion and do not account for the external forces, impacts, and torques that arise in real-world tool-use and contact-rich manipulation. As a result, existing RL policies are not designed to maintain grasp stability under dynamic interactions with the environment. Furthermore, they often rely on extensive training, carefully shaped task-specific rewards, and heavy domain randomization to achieve sim-to-real transfer—factors that limit their scalability to dynamic force-intensive tasks.

To improve sample efficiency and encourage physically meaningful behaviors, many works incorporate human priors by initializing RL policies from human motion data or pre-grasp configurations [25], [26], [27], [28], [29]. While such priors accelerate learning, they cannot guarantee grasp stability once dynamic forces act on the object. Other approaches reduce the action dimensionality using eigengrasp-based controllers [7], [30], [31], but the resulting low-rank control spaces limit the fine-grained torque regulation required for dynamic grasp adaptation.

In contrast, our method employs RL not to learn full manipulation behaviors but to adapt grasp stability online during tool-use trajectories. By initializing the policy from analytically stable grasps and training on residual joint corrections, our approach mitigates the impact of inaccurate contact dynamics [32], [33] and improves robustness under strong external forces.

III. METHOD

To enable robots to succeed in dexterous grasping tasks, grasps must not only be stable but also functionally appropriate (e.g., holding a hammer by the handle rather than by the head). Furthermore, for successful task execution, the robot hand must maintain this stable grasp while following the task-specific trajectories. Our framework integrates sampling-based grasp synthesis with online reinforcement learning-based adaptation, as illustrated in Fig. 2.

A. Human Demonstration

We begin by recording an RGBD video $\{I^t\}_{t=1}^T$ of a human demonstrating the process of grasping the target tool and performing the task. This demonstration data is used for two purposes: initializing the search space for the task-driven grasping, and providing a reference trajectory of moving the object in grasp.

Based on each frame in the video, we extract the object’s 6D pose trajectory using FoundationPose [34], conditioned



Fig. 2: Pipeline of sampling and evaluating grasps. (A) We initialize the range of grasp locations based on human demonstration. (B-C) Candidate grasps are generated by varying joint-level parameters. (D) Each grasp is evaluated through wrench-space stability tests along six force and torque axes. (E) Top-ranked grasps are used as the initial grasp for conducting the tool-using trajectory, followed by online RL-based adaptation.

on the object’s mesh, the full RGB-D video, and a segmentation mask of the object in the first frame. The initial mask is obtained using Grounded-SAM [35], prompted with the object name and the initial RGB image without occlusions. This process produces a per-frame object pose sequence $\{\mathbf{T}_{\text{obj}}^t\}_{t=1}^T$, where $\mathbf{T}_{\text{obj}}^t \in \text{SE}(3)$. The object trajectory is used to define a task-specific reward for trajectory tracking during policy learning.

To localize the human’s grasp, we annotate the timestep τ at which the object begins task motion. At this frame, we estimate the 6D wrist pose $\mathbf{T}_{\text{wrist}}^\tau \in \text{SE}(3)$ in the camera frame using the pipeline introduced by Lum et al [26]. Specifically, we use an off-the-shelf hand pose estimator, HaMeR [36], to obtain a detailed representation of the hand [37], then refine the result using ICP alignment between the predicted hand mesh and the observed depth image I^τ . The resulting wrist pose $\mathbf{T}_{\text{wrist}}^\tau$ is used to guide downstream grasp synthesis.

B. Grasp Synthesis and Evaluation

To generate an initial grasp, we sample a set of feasible candidate grasp configurations and evaluate their stability. Based on a stability score, we select the optimal grasp configuration to execute the trajectory. This process is performed in simulation (Isaac Lab [38]).

1) Grasp Generation: We define a grasp configuration as $G = \{q_o, \mathbf{T}_{\text{grasp}}\}$, consists of the robot’s finger joint angles $q_o \in \mathbb{R}^n$ and the wrist pose $\mathbf{T}_{\text{grasp}} \in \text{SE}(3)$ in the object frame. During the grasp formation process, the object is suspended in mid-air. The grasp generation procedure involves two primary steps:

a) Functional Wrist Region Sampling: We define the grasp region based on the human wrist pose $\mathbf{T}_{\text{wrist}}^\tau$. Candidate wrist poses $\mathbf{T}_{\text{grasp}}$ are uniformly sampled within a predefined region centered on the human wrist pose $\mathbf{T}_{\text{wrist}}^\tau$. Formally, $\mathbf{T}_{\text{grasp}} = \mathbf{T}_{\text{wrist}}^\tau \cdot \mathbf{T}_{\text{perturb}}$, where $\mathbf{T}_{\text{perturb}}$ includes translations $\Delta\{x, y, z\}$ and rotations $\Delta\{\theta_x, \theta_y, \theta_z\}$, each sampled uniformly within specified limits, represented by the box in Fig. 2A.

b) Finger Configuration Sampling: To synthesize diverse grasps that can robustly conform to varied object geometries, we sample several parameters controlling the initial finger configurations (shown in Fig. 2B) and dynamics of finger closing:

- *Inter-finger Angles (θ_f):* Angles between adjacent fingers f are uniformly sampled to directly control how spread apart these fingers are at grasp initiation. The thumb angle θ_t is sampled separately based on its initial configuration. Varying these angles enables the grasp to accommodate objects of different widths and shapes.
- *Joint Group Closing Rates (\dot{q}_g):* Each finger is divided into hierarchical joint groups (proximal, intermediate, distal segments), and each group g is assigned its own randomized closing speed. By independently varying these closing rates, the fingers can form distinctly different grasp shapes. For example, faster distal joints with slower proximal joints lead to claw-like grasps, while synchronized rates among all joints produce enveloping, power-like grasps.
- *Joint Torque Threshold (τ_q^*):* Finger joints stop closing when their exerted joint torque exceeds a uniformly sampled threshold τ_q^* . Randomizing this threshold leads to grasps with different levels of tightness, which helps the hand adapt to objects with varying shapes and surface features. This variation makes it easier to generate grasps that can handle surface irregularities or misalignments during grasp formation.

After the grasping is complete, the object is unfrozen, allowing it to settle naturally under contact and gravity (shown in Fig. 2C).

2) Grasp Stability Scoring: Each generated grasp configuration G is evaluated for stability through a set of disturbance wrench tests. Here, the wrench-space comprises all Cartesian force and torque axes in both positive and negative directions ($\pm x, \pm y, \pm z$), totaling 12 independent test dimensions, Fig. 2D. In each wrench-space test dimension i , the magnitude of the external wrench is increased linearly from zero up to a predefined maximum force F_{max} or torque τ_{max} for a fixed duration t_{max} . In-hand slip is defined as object deviation beyond position or orientation thresholds (δ_p, δ_θ). When a slip occurs, the test is terminated, and we record the elapsed stable time $t_{\text{stable}}^{(i)}$ for that direction. The grasp stability score S_G is then computed by averaging the stable time fractions across all wrench-space dimensions: $S_G = \frac{1}{12} \sum_{i=1}^{12} \frac{t_{\text{stable}}^{(i)}}{t_{\text{max}}}$.

Before each test, the grasp configuration G is reinstated to ensure consistent initial conditions. The grasps are ranked

according to their stability score S_G , and the highest-ranked grasps G^* are selected and stored for online grasp adaptation. The grasp synthesis and evaluation pipeline is parallelized and can identify hundreds of usable grasps and multiple robust grasps within 2 minutes.

C. Online Grasp Adaptation

After performing the initial grasp, we implement a reinforcement learning (RL) policy π_θ focused on fine-tuning finger joint movements to maintain grasp stability throughout dynamic task execution. We formulate this as a grasp refinement problem from optimized grasps G^* , where the joint angles are initialized to q_o , and the grasp pose $\mathbf{T}_{\text{grasp}}$ is kept fixed in the object’s frame. Formally, $\mathbf{T}_{\text{hand}}^t = \mathbf{T}_{\text{obj}}^t \cdot \mathbf{T}_{\text{grasp}}$, giving us the location of the robot wrist at timestep t .

1) Simulation Environment: For efficient training and scalability across diverse tasks, we develop a simulation environment that abstracts complex physical interactions into simplified force-based models. Tasks that involve interactions with fluids or deformable materials, such as stirring pancake batter or cutting cucumbers, are computationally expensive to simulate directly. To overcome this limitation, we approximate these interactions using strategically applied forces at relevant object contact points. We categorize these forces into two types:

- *Resistive Forces (F_{res}):* These forces act in the opposite direction of the intended tool motion, simulating frictional and resistive interactions. Formally, $F_{\text{res}} = k_{\text{res}} \cdot v_{\text{eff}}$, where k_{res} is the resistive coefficient and v_{eff} is the local velocity vector. For example, a resistive force is applied to the end effector of the spoon while stirring pancake batter.
- *Application Forces (F_{app}):* These are modeled as directional forces applied onto the object (e.g., payload, normal). Formally, $F_{\text{app}} = k_{\text{app}} \cdot \hat{d}$, where k_{app} is the application coefficient and \hat{d} is the task-specific unit direction. For instance, a continuous downward force simulates the weight of the scooped item, turning off upon dumping.

Real-world payloads and contact forces are measured during task execution. We match our measured values to the simulation by tuning our force coefficients ($k_{\text{res}}, k_{\text{app}}$) on a per-task basis. Further, we apply domain randomization to our coefficients to improve performance under real-world conditions.

2) Reinforcement Learning Formulation: We define a reinforcement learning policy π_θ parameterized by θ as follows: $\pi_\theta : s_t \mapsto \Delta q_t$, where the observation state s_t is given by the concatenation of the robot’s proprioceptive state s_{robot} , the goal object state s_{goal} , and a normalized task timestep t/T . Here, the robot’s proprioceptive state includes joint positions $q_t \in \mathbb{R}^n$, target joint positions $q_t^{\text{target}} \in \mathbb{R}^n$, and joint torques $\tau_t \in \mathbb{R}^n$. The output residual joint adjustments $\Delta q_t \in \mathbb{R}^n$ represent incremental corrections to the current joint configuration. Formally, $q_{t+1} = q_t + \Delta q_t$, where q_{t+1} are the executed joint positions.

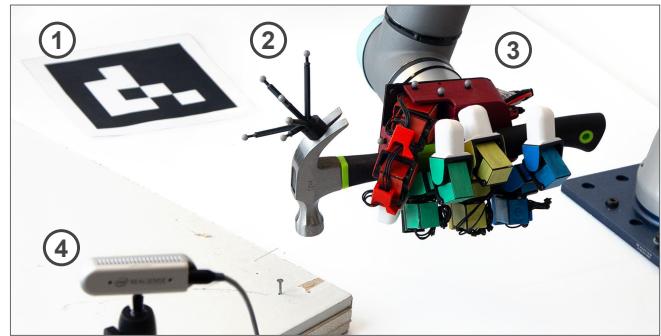


Fig. 3: *Real-world experiment setup.* The workspace includes an (1) AprilTag, (2) tool tracking markers, (3) hand tracking markers, and an (4) RGBD camera.

The reward function combines separate terms for position alignment, orientation alignment, and penalties for large deviations. We define the position reward as using a mean square error:

$$r_p^t = \max(1 - \alpha_p \cdot \|p_{\text{obj}}^t - p_{\text{goal}}^t\|_2, 0),$$

where $p_{\text{obj}}^t, p_{\text{goal}}^t \in \mathbb{R}^3$ are the current and target object centroid positions, and α_p is a scaling coefficient. We define the orientation reward using a quaternion dot product:

$$r_q^t = \max(1 - \alpha_q \cdot (1 - |q_{\text{obj}}^t \cdot q_{\text{goal}}^t|), 0)$$

where $q_{\text{obj}}^t, q_{\text{goal}}^t \in \mathbb{S}^3$ are the current and target object quaternions, and α_q is a scaling coefficient. A large negative penalty r_{pen}^t is applied if the position or orientation exceeds predefined thresholds δ_p for position or δ_θ for orientation. Episodes are reset after the penalty is applied. The final reward at timestep t is given by: $r^t = r_p^t + r_q^t + r_{\text{pen}}^t$.

We implement our policy π_θ using proximal policy optimization (PPO) [39]. Our network consists of an input layer that receives the state s_t , followed by a 512-unit LSTM [40] layer, an MLP with hidden sizes 512 and 256, and an output layer that predicts the residual joint adjustments Δq_t .

Given optimized grasps G^* , our simple reward formulation is sufficient to guide the policy to converge in ~ 15 minutes on a single NVIDIA RTX 4070 Ti Super.

IV. EXPERIMENTS

Our experiments seek to validate whether *Grasp-to-Act* enables dexterous robot hands to maintain stable, functional grasps during dynamic and contact-rich manipulation tasks. The proposed tasks pose diverse challenges, requiring grasps to maintain low positional and angular errors, while facing repetitive impacts or continuous external forces.

A. Hardware Setup

We deploy our pipeline using a 16-DoF LEAP hand [8] mounted on a 6-DoF UR5 robotic arm, and a stationary Intel RealSense D435 RGB-D camera. The execution procedure in the real world involves the following steps:

We identify the object’s initial pose relative to the camera frame $\mathbf{T}_{\text{cam} \leftarrow \text{obj}} \in \text{SE}(3)$ using FoundationPose [34]. We use an AprilTag [41] fixed in the workspace to determine the camera’s pose relative to the global coordinate frame

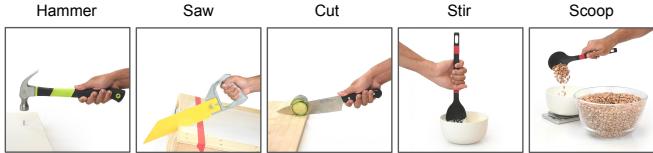


Fig. 4: Human demonstrations of the five functional tasks we test with in this paper: hammering, sawing, cutting, stirring, and scooping.

$\mathbf{T}_{\text{world} \leftarrow \text{cam}} \in \text{SE}(3)$. Given our optimized grasp’s wrist pose relative to the object’s frame $\mathbf{T}_{\text{grasp}}$, we can compute the grasp location in the world frame $\mathbf{T}_{\text{world} \leftarrow \text{grasp}}$ by chaining the transformations: $\mathbf{T}_{\text{world} \leftarrow \text{grasp}} = \mathbf{T}_{\text{world} \leftarrow \text{cam}} \cdot \mathbf{T}_{\text{cam} \leftarrow \text{obj}} \cdot \mathbf{T}_{\text{grasp}}$

The UR5 arm moves the robot wrist to $\mathbf{T}_{\text{world} \leftarrow \text{grasp}}$ via inverse kinematics (IK), after which the LEAP hand executes the optimized finger joint configuration q_o . We assume that the grasping region is unobstructed at execution time.

Following grasp execution, the robot moves the grasped object to align with the initial position of the demonstrated trajectory. Specifically, at each timestep t , the target wrist pose $\mathbf{T}_{\text{hand}}^t$ is calculated from the human-demonstrated object trajectory $\{\mathbf{T}_{\text{obj}}^t\}_{t=1}^T$ as described in Sec. III-C. IK is used to move the robot wrist to each sequential pose $\mathbf{T}_{\text{hand}}^t$. Concurrently, the RL policy π_θ issues joint adjustments Δq_t at each timestep. Both the wrist trajectory and finger joint angles are updated synchronously at a control frequency of 30 Hz.

We use an OptiTrack motion-capture system with a rigid four-marker mount attached to each object to accurately record its 6D pose during real-world trials (see Fig. 3).

B. Tasks and Objects

We evaluate our approach on five manipulation tasks that require different stable grasps and dynamic control (see Fig. 4). These tasks involve different object geometries, force applications, and motion patterns. For each task, we also define a task completion metric \mathcal{T} to quantitatively assess performance in the real world:

- (a) *Hammer*: Drive a nail 2 cm above a drywall stack using a steel claw hammer (30 cm length, rubber handle, 520 g) with 4 strikes. Performance is the fraction of the nail embedded.
- (b) *Saw*: Cut an 8 cm groove through drywall using a handheld saw (26 cm serrated blade, metal handle, 365 g) with 12 forward–return motions. Performance is the sawed length relative to the target.
- (c) *Cut*: Slice a 5 cm cucumber segment using a stainless-steel kitchen knife (8-inch blade, plastic handle, 155 g) with 16 forward–return motions. Performance is the cut depth normalized by the cucumber diameter.
- (d) *Stir*: Stir pancake batter with a plastic ladle (32 cm length, plastic handle, 60 g) for 26 clockwise rotations. Performance is the fraction of total stirring time the tool is held securely.
- (e) *Scoop*: Transfer 35 g of pinto beans using the same ladle in one motion. Performance is the transferred mass ratio to the target weight.

C. Baselines

We compare our method against several baselines representative of analytical optimization and RL-based approaches commonly used in dexterous manipulation tasks:

(1) *Analytical* [6], [2], [3]: Grasps are synthesized using a geometric differentiable optimization method. To make this baseline suitable for our functional tasks, we initialize the optimization from human-demonstrated wrist poses near functional regions of the tool, and then rank generated grasps using our evaluation protocol.

(2-5) *RL-based methods*: These baselines are given a 5 second initialization phase, where the object remains fixed in space, allowing the robot hand to change its initial wrist and finger configurations to make a grasp. The initial wrist pose is placed at the human-demonstrated wrist location. To encourage wrist alignment during this initialization phase, we incorporate a proximity reward r_w that penalizes the distance between the wrist and the object. Formally written as: $r_w = \max(1 - \alpha_w \|p_{\text{wrist}} - p_{\text{obj}}\|_2, 0)$. (2) *RL base*: This policy initialized from a default joint-angle configuration with no modifications. (3) *RL w/ Contact Rewards* [42]: This policy integrates explicit finger-object proximity rewards to encourage grasping with all fingers. (4) *RL w/ Pre-Grasp Pose* [26]: The policy initialized from a retargeted human-demonstrated finger joint configuration, enabling the policy to refine grasp poses that closely match human grasps. (5) *RL w/ Eigengrasp* [30], [31], [26]: The policy actions are constrained within a low-dimensional (5-dimensional) eigengrasp space derived from principal component analysis (PCA) of human grasps. This reduces action-space dimensionality and guides the exploration toward human-like grasps. We directly use Lum et al’s controller [26].

(6) *G2A w/o Adaptation*: This baseline refers to the grasps synthesized by our grasp optimization pipeline without the RL-based online adaptation. (7) *G2A*: This is our full method.

D. Evaluation Metrics

We evaluate performance using three metrics shared across simulation and real-world experiments: in-hand translational slip distance E_t , in-hand rotational slip distance E_θ , and either grasp success \mathcal{S} (used in simulation) or task completion \mathcal{T} (used in real-world experiments).

1) *In-hand translational slip distance E_t (cm)*: For episodes without object dropping, we measure the average positional deviation of the object centroid from the target trajectory when no slip occurs: $E_t = \frac{1}{T} \sum_{t=1}^T \|p_{\text{obj}}^t - p_{\text{goal}}^t\|_2$

2) *In-hand slip rotation distance E_θ (°)*: Similarly, for successful episodes, we calculate the average angular deviation between the object’s orientation and the target orientation when no slip occurs: $E_\theta = \frac{1}{T} \sum_{t=1}^T 2 \cdot \arccos(|q_{\text{obj}}^t \cdot q_{\text{goal}}^t|)$

3) *Task performance*: As the tasks are not explicitly modeled in simulation, we instead report a grasp success \mathcal{S} (%) indicating whether the object remains held throughout the episode without dropping. In real-world experiments, we

Method	Hammer			Saw			Cut			Stir			Scoop		
	$S \uparrow$	$E_t \downarrow$	$E_\theta \downarrow$	$S \uparrow$	$E_t \downarrow$	$E_\theta \downarrow$	$S \uparrow$	$E_t \downarrow$	$E_\theta \downarrow$	$S \uparrow$	$E_t \downarrow$	$E_\theta \downarrow$	$S \uparrow$	$E_t \downarrow$	$E_\theta \downarrow$
Analytical [6], [2], [3]	0.0	–	–	0.0	–	–	0.0	–	–	0.0	–	–	0.0	–	–
RL base	100.0	0.93	4.07	54.7	6.63	13.53	81.2	0.85	6.25	98.5	0.51	5.34	100.0	0.52	2.57
+ contact reward [42]	100.0	0.86	3.99	49.3	6.92	14.16	74.6	0.94	7.43	100.0	0.47	4.54	100.0	0.46	1.94
+ pre-grasp pose [26]	100.0	0.88	9.48	100.0	2.24	6.74	20.4	2.20	17.04	100.0	0.45	3.47	96.9	1.23	6.32
+ eigengrasp [30], [31], [26]	67.2	1.70	14.36	100.0	1.95	5.84	0.0	–	–	0.0	–	–	0.0	–	–
G2A w/o Adaptation	100.0	0.82	7.33	100.0	1.29	3.90	100.0	0.54	3.53	100.0	0.92	4.73	100.0	0.69	3.69
G2A (ours)	100.0	0.69	1.35	100.0	1.06	3.13	100.0	0.48	2.23	100.0	0.24	2.08	100.0	0.17	1.25

TABLE I: *Simulation results across five tasks.* We report the grasp success S (%), in-hand translational slip distance E_t (cm) and in-hand slip rotation distance E_θ ($^\circ$). All error metrics (E_t, E_θ) are averaged over successful episodes.

Method	S (%) \uparrow	E_t (cm) \downarrow	E_θ ($^\circ$) \downarrow	Hammer
RL base	100.0	0.93	4.07	
G2A w/o Adaptation	100.0	0.82	7.33	
G2A (ours)	100.0	0.69	1.35	
RL base	46.6	2.04	17.12	
G2A w/o Adaptation	92.3	1.02	11.43	
G2A (ours)	100.0	0.94	2.69	
RL base	83.7	1.91	5.98	
G2A w/o Adaptation	100.0	0.91	9.75	
G2A (ours)	100.0	0.78	1.57	

TABLE II: *Generalization across hammer types.* \star indicates that the method was trained for that tool instance only. We report the grasp success S (%), in-hand translational slip distance E_t (cm) and in-hand slip rotation distance E_θ ($^\circ$).

measure task completion \mathcal{T} defined in Section IV-B, which directly reflects functional task success (e.g., nail inserted, material cut).

V. RESULTS

A. Experiments in Simulation

We evaluate our method and all baseline methods across all tasks in simulation over 3 random seeds, reporting average performance in Table I. For the optimization-based baselines, we select the top three highest-scoring grasp candidates. The RL-based policies are trained for 100K environment steps per task.

1) *Performance Across Tasks:* We first evaluate all baseline and proposed methods across the five tasks (Sec. IV-B) in simulation.

Analytical baseline method fails to achieve stable grasps across all tasks. The synthesized grasps are mostly precision grasps, which suffice for picking up lighter objects such as the knife or ladle, but fail when subjected to any task-specific forces.

RL base and *RL w/ Contact Reward* perform similarly in simulation. Both methods often converge to suboptimal local minima, characterized by limited finger utilization and visually unstable grasp configurations that nonetheless work under simulation dynamics. Further, the lack of a human grasp prior is evident in the sawing task, where the policies grip around the handle rather than inserting fingers into it, making it difficult to maintain a secure hold. *RL w/ Pre-Grasp Pose* baseline provides strong performance in tasks where the human strategy is compatible with the robot embodiment.

For instance, initializing with fingers already within the saw handle helps maintain stability during execution. However, this benefit does not generalize across all tasks. In the knife task, the human grasp relies on the thenar eminence (the lump at the base of the thumb), which the LEAP hand lacks. As a result, direct retargeting can lead to poor initialization. *RL w/ Eigengrasp* is particularly interesting. The approach yields human-like grasping strategies, as observed in prior work. The initial training converges rapidly compared to RL-only methods, reflecting the lower-dimensional, structured action space. However, this same action space reduction limits dexterity for objects with smaller or more intricate grasp regions. As the handle size decreases, the inability to execute fine-grained finger adjustments prevents the policy from establishing sufficient contact on objects.

G2A w/o Adaptation baseline produces grasps that remain stable throughout the entire trajectory and do not drop the object during execution. However, under repeated application of task-specific forces, particularly in hammering, cumulative pose errors accrue over time. *G2A* consistently outperforms all baselines. By leveraging analytically optimized grasps as initialization, the RL policy starts with a stable grasp configuration, allowing the policy to focus on minor adjustments to prevent or fix slip or rotation under dynamic task conditions. This narrows the exploration space and substantially accelerates policy convergence.

2) *Generalization across Tool Variants:* To assess robustness to unseen tool geometries, we evaluate across three hammer variants that differ in mass distribution and handle shape (Table II). All policies were trained and optimized only on the first (marked with \star) hammer instance and directly tested on the other two without retraining.

RL base policy, which learns a task-specific grasping strategy, performs well only on the original hammer but fails to generalize—minor shifts in handle thickness or head geometry lead to loss of contact and frequent object drops. *G2A w/o Adaptation* baseline maintains secure grasps initially but cannot compensate for variations in handle diameter, resulting in partial slip for the thinner-handled hammer. In contrast, our full *G2A* method consistently preserves grasp stability across all hammer variants, achieving 100% success with minimal translational and rotational slip errors (E_t, E_θ). These results highlight that the online adaptation policy effectively compensates for geometric and inertial changes.

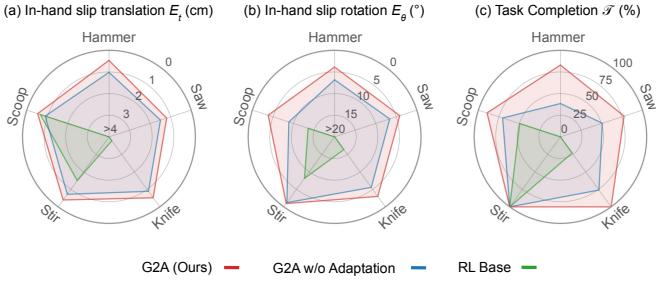


Fig. 5: *Real-world results across five functional tasks.* Comparison of (a) In-hand translational slip distance E_t (cm), (b) In-hand slip rotation distance E_θ ($^\circ$), and (c) task completion \mathcal{T} . Task-specific definitions of \mathcal{T} are provided in Section IV-B.

B. Real-World Experiments

We evaluate over 10 real-world trials per task against two baselines: *G2A w/o Adaptation*, the strongest non-adaptive policy, and *RL Base*, a consistent RL variant that performed reliably across most tasks. Results are shown in Fig. 5. Some examples of the experiment episodes and the rotational slip deviation is shown in Fig. 6.

RL Base policy performs reasonably in simulation but transfers poorly to hardware. In practice, objects such as the hammer and saw slip once large external forces are applied. Only lightweight tools can be manipulated successfully, and even then, significant orientation drift occurs.

G2A w/o Adaptation baseline demonstrates markedly stronger sim-to-real transfer. The synthesized grasps remain stable throughout most trials, reflected in lower translation and orientation errors than *RL Base*. However, as seen before, while these grasps remain secure, residual pose drift accumulates during longer or more dynamic motions, leading to moderate E_t and E_θ values. Our full *G2A* method achieves the lowest translation and orientation errors across all tasks, along with the highest task completion \mathcal{T} . As shown in Fig. 6, per-task orientation traces highlight the stability of *G2A* over time. Orientation errors remain bounded even under repeated impacts or resistive forces. The adaptive policy adjusts finger configurations in response to contact, maintaining grasp stability throughout the task.

VI. DISCUSSION

In this paper, we study grasp configurations and adaptive control strategies to maintain stability during real-world manipulation tasks involving significant dynamic forces. The current implementation is based on two simplifying assumptions: (1) the initial graspable region is unobstructed, and (2) the object geometries are relatively simple.

In practice, any objects rest on tables or shelves, where contact with the surface can obstruct part of the graspable area. In this case, our framework cannot execute the desired grasp. To address this, our method could be combined with adaptive grasping strategies to maneuver objects into target grasp configurations [14].

For objects with more complex geometries—though such cases are less common in dynamic, force-intensive manipulation—our sampling-based grasp synthesis could be further improved by integrating geometric grasp-planning

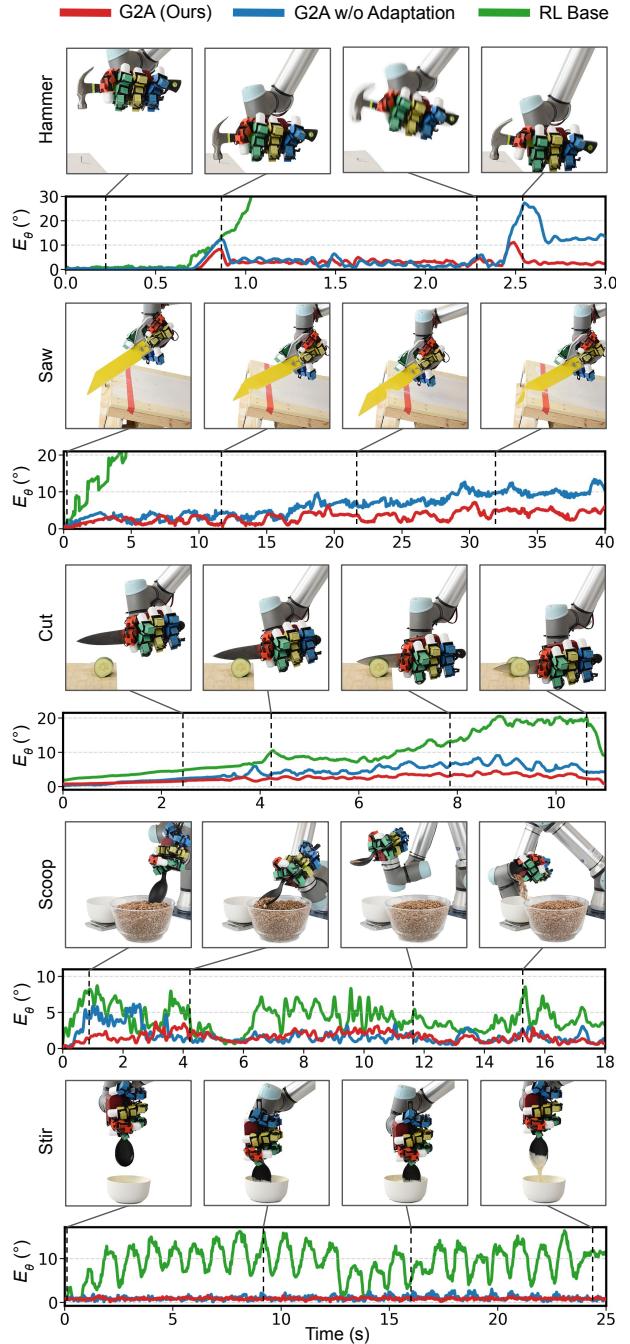


Fig. 6: *Real-world rollouts.* In-hand slip rotation distance E_θ for one representative rollout per baseline across the tasks. Task snapshots show the G2A execution.

techniques, which may enable more efficient generation and evaluation of grasp candidates.

VII. CONCLUSION

We presented *Grasp-to-Act*, a framework that enables dexterous robotic hands to achieve stable and functional grasps across diverse real-world manipulation tasks. By combining grasp optimization with reinforcement learning-based adaptation, our approach maintains robust control under dynamic, contact-rich conditions. In real-world experiments, *G2A* consistently outperforms baselines, achieving the highest task

completion rates across a range of tasks such as hammering, sawing, cutting, stirring, and scooping. These results look to bridge the gap between simulation and practical deployment for functional dexterity.

REFERENCES

- [1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- [2] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, “Dexgrasnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 359–11 366.
- [3] T. G. W. Lum, A. H. Li, P. Culbertson, K. Srinivasan, A. D. Ames, M. Schwager, and J. Bohg, “Get a grip: Multi-finger grasp evaluation at scale enables robust sim-to-real transfer,” *arXiv preprint arXiv:2410.23701*, 2024.
- [4] H. Chen, Y. Yao, Y. Ye, Z. Xu, H. Bharadhwaj, J. Wang, S. Tulsiani, Z. Erickson, and J. Ichniowski, “Web2grasp: Learning functional grasps from web images of hand-object interactions,” *arXiv preprint arXiv:2505.05517*, 2025.
- [5] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [6] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, “Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.
- [7] M. Ciocarlie, C. Goldfeder, and P. Allen, “Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem,” in *Robotics: Science and systems manipulation workshop—sensing and adapting to the real world*, 2007.
- [8] K. Shaw, A. Agarwal, and D. Pathak, “Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning,” *Robotics: Science and Systems (RSS)*, 2023.
- [9] D. Berenson and S. S. Srinivasa, “Grasp synthesis in cluttered environments for dexterous hands,” in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 189–196.
- [10] A. Wu, M. Guo, and C. K. Liu, “Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization,” *arXiv preprint arXiv:2207.00195*, 2022.
- [11] A. H. Li, P. Culbertson, J. W. Burdick, and A. D. Ames, “Frogger: Fast robust grasp generation via the min-weight metric,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6809–6816.
- [12] C. Ferrari, J. Canny *et al.*, “Planning optimal grasps,” in *Proceedings., 1992 IEEE International Conference on Robotics and Automation, 1992.*, vol. 3. IEEE, 1992, pp. 2290–2295.
- [13] T. Feix, J. Romero, H.-B. Schmidmayer, A. M. Dollar, and D. Kragic, “The grasp taxonomy of human grasp types,” *IEEE Transactions on human-machine systems*, vol. 46, no. 1, pp. 66–77, 2015.
- [14] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, T. Liu, L. Yi, and H. Wang, “Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 4737–4746.
- [15] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, “D(r,o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping,” *arXiv preprint arXiv:2410.01702*, 2024.
- [16] V. Mayer, Q. Feng, J. Deng, Y. Shi, Z. Chen, and A. Knoll, “Ffhnet: Generating multi-fingered robotic grasps for unknown objects in real-time,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [17] P. Li, T. Liu, Y. Li, Y. Zhu, Y. Yang, and S. Huang, “Gendexgrasp: Generalizable dexterous grasping,” *arXiv preprint arXiv:2210.00722*, 2022.
- [18] D. Turpin, T. Zhong, S. Zhang, G. Zhu, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, “Fast-grasp’d: Dexterous multi-finger grasp generation through differentiable simulation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [19] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, “Contrastive learning with hard negative samples,” in *International Conference on Learning Representations*, 2021.
- [20] A. Mousavian, C. Eppner, and D. Fox, “6-dof graspernet: Variational grasp generation for object manipulation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [21] L. F. Casas, N. Khargonkar, B. Prabhakaran, and Y. Xiang, “Multi-grippergrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2978–2984.
- [22] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, “General in-hand object rotation with vision and touch,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2549–2564.
- [23] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. Van Wyk, “Dextrahrgb: Visuomotor policies to grasp anything with dexterous hands,” *arXiv preprint arXiv:2412.01791*, 2024.
- [24] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, “Visual dexterity: In-hand reorientation of novel and complex object shapes,” *Science Robotics*, vol. 8, no. 84, p. eade9244, 2023.
- [25] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations,” *arXiv preprint arXiv:1709.10087*, 2017.
- [26] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg, “Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration,” *arXiv preprint arXiv:2504.12609*, 2025.
- [27] X. Liu, J. Adalibieke, Q. Han, Y. Qin, and L. Yi, “Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references,” *arXiv preprint arXiv:2502.09614*, 2025.
- [28] L. Huang, H. Zhang, Z. Wu, S. Christen, and J. Song, “Fungrasp: Functional grasping for diverse dexterous hands,” *IEEE Robotics and Automation Letters*, 2025.
- [29] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song, “Dexmachina: Functional retargeting for bimanual dexterous manipulation,” *arXiv preprint arXiv:2505.24853*, 2025.
- [30] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak, “Dexterous functional grasping,” *arXiv preprint arXiv:2312.02975*, 2023.
- [31] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. Van Wyk, “Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics,” *arXiv preprint arXiv:2407.02274*, 2024.
- [32] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, “Lessons from learning to spin” pens,” *arXiv preprint arXiv:2407.18902*, 2024.
- [33] E. Salvato, G. Fenú, E. Medvet, and F. A. Pellegrino, “Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning,” *IEEE Access*, vol. 9, pp. 153 171–153 187, 2021.
- [34] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “Foundationpose: Unified 6d pose estimation and tracking of novel objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.
- [35] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, “Grounded sam: Assembling open-world models for diverse visual tasks,” *arXiv preprint arXiv:2401.14159*, 2024.
- [36] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, “Reconstructing hands in 3d with transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9826–9836.
- [37] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *arXiv preprint arXiv:2201.02610*, 2022.
- [38] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, “Orbit: A unified simulation framework for interactive robot learning environments,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.
- [39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [40] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [41] E. Olson, “Apriltag: A robust and flexible visual fiducial system,”

- in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.
- [42] L. Xu, Z. Liu, Z. Gui, J. Guo, Z. Jiang, Z. Xu, C. Gao, and L. Shao, “Dexsingrasp: Learning a unified policy for dexterous object singulation and grasping in cluttered environments,” *arXiv preprint arXiv:2504.04516*, 2025.