LEGS: Learning Efficient Grasp Sets for Exploratory Grasping

Letian Fu¹, Michael Danielczuk¹, Ashwin Balakrishna¹, Daniel S. Brown¹, Jeffrey Ichnowski¹, Eugen Solowjow², Ken Goldberg¹

Abstract—Previous work defined Exploratory Grasping, where a robot iteratively grasps and drops an unknown complex polyhedral object to discover a set of robust grasps for each recognizably distinct stable pose of the object. Recent work used a multi-armed bandit model with a small set of candidate grasps per pose; however, for objects with few successful grasps, this set may not include the most robust grasp. We present Learned Efficient Grasp Sets (LEGS), an algorithm that can efficiently explore thousands of possible grasps by constructing small active sets of promising grasps and uses learned confidence bounds to determine when, with high-confidence, it can stop exploring the object. Experiments suggest that LEGS can identify a high-quality grasp more efficiently than prior algorithms which do not learn active sets. In simulation experiments, we measure the optimality gap between the success probability of the best grasp identified by LEGS and baselines and that of the true most robust grasp. After 5000 steps of exploration, LEGS achieves an average optimality gap 20% and 25% lower than baselines on the Dex-Net Adversarial and EGAD! datasets respectively. In physical experiments we find that across 3 objects, LEGS converges to high-performing grasps significantly faster than baselines. See https://sites.google.com/view/legs-exp-grasping for supplemental material and videos.

I. INTRODUCTION

Deep learning has enabled robust robot grasping for a wide variety of objects [19–21, 24, 25, 29]. But for all existing general-purpose grasping algorithms, adversarial objects can be created such that these grasping algorithms struggle to discover a robust grasp [23, 30]. Even non-adversarially generated objects that are outside the training distribution may cause these algorithms to fail. To address these situations, we propose to supplement general-purpose grasping algorithms with object-specific exploration.

One promising approach is to use bandit-style algorithms to extend general-purpose grasping policies by rapidly adapting these policies to specific objects [7, 15, 17, 18]. Recently, Danielczuk et al. [5] introduced Exploratory Grasping, where a robot learns to grasp novel objects through online exploration of grasps and stable poses. Their algorithm, Bandits for Online Rapid Grasp Exploration Strategy (BORGES), learns robust grasps for each stable pose by learning pose-specific grasping policies. However, to allow for convergence on a physical system, BORGES limits exploration to a fixed set of 100 grasps per stable pose, causing BORGES to possibly overlook high-quality grasps. We extend Danielczuk et al. [5] to settings with thousands of grasps per stable pose. Considering grasp sets of this scale increases the likelihood of converging to a robust grasp, but also makes efficient exploration challenging.

To address this challenge, we propose Learned Efficient Grasp Sets (LEGS), which adaptively curates an active set of promising grasps rather than restricting exploration to a small fixed subset. The key insight is to use a combination of priors from a general purpose grasping system and online trials to maintain confidence bounds on grasp success probabilities. LEGS uses these bounds to (1) update the grasps contained in its active set and (2) decide when to stop exploring.

We contribute: (1) Learned Efficient Grasp Sets (LEGS): a novel adaptive multi-armed bandits algorithm that curates a small set of high-performing grasps by actively removing and resampling grasps based on performance bounds; (2) a novel termination condition that enables a robot to predict (with high-confidence) when it reached a desired level of performance; (3) a self-supervised physical grasping system where a robot explores candidate grasps with minimal human intervention (interventions needed on average roughly every 100 grasp attempts), and (4) simulation and physical experiments suggesting that LEGS can identify higher quality grasps on challenging objects within a fixed time horizon than prior algorithms which do not learn an active set.

II. RELATED WORK

A. Universal Grasping Algorithms

Recent robotic grasping algorithms generalize to a wide range of objects [14]. These include open-loop algorithms which synthesize grasps and predict their quality based on the geometry of the object, and then plan and execute a motion
to attempt a high-quality grasp without feedback [16, 19–21, 25]. There has also been interest in closed-loop grasp planners which use vision-based gripper servoing [24, 29] and RL [11, 12]. LEGS is designed to leverage priors from these universal grasping algorithms to efficiently learn a robust grasp policy for a specific, difficult-to-grasp object [23, 30]. We learn priors from Dex-Net 4.0 [21], a general grasp planner that learns a grasp quality estimator from a large dataset of 3D object models in simulation and then uses this estimator to sample and evaluate the quality of grasps in physical trials.

B. Multi-Armed Bandits

Multi-Armed bandits [27] is a formalism for adaptively exploring actions to maximize a reward function. There is significant prior work studying the many-armed bandit problem, in which the number of actions is large compared to the number of timesteps allocated for exploration [2, 4, 9, 10, 28, 31, 32]. One popular algorithmic framework for this setting is called best arm identification, where the goal is to adaptively reject a set of arms from consideration when there is high confidence that they are suboptimal [1, 3, 13]. LEGS builds on these ideas, but provides a new algorithm which adaptively filters actions from an active set by maintaining confidence bounds on the reward corresponding to each action. This filtering mechanism makes it possible to efficiently perform best arm identification across multiple many-armed bandits problems, where each bandit problem represents a distinct stable pose of an object. Our experiments suggest that LEGS can consider thousands of grasp actions per stable pose, converging quickly to high-quality grasps.

C. Exploratory Grasping

Prior work demonstrates that universal grasping algorithms can struggle with certain objects and present algorithms to systematically synthesize adversarial examples for grasp planning systems [23, 30]. Danielczuk et al. [5] show that grasping algorithms such as Dex-Net [21] are difficult to fine-tune online on such objects, and propose exploratory grasping, a problem formulation where the objective is to perform rapid online adaptation to grasp specific, unknown objects. To achieve this, prior works apply multi-armed bandit algorithms and adapt them to online grasping [7, 15, 17, 18]. These methods treat the grasps sampled on a given objects as “arms”, and use multi-armed bandit algorithms to search for the best grasp. Danielczuk et al. [5] propose Bandits for Online Rapid Grasp Exploration Strategy (BORGES), which optimizes grasps on each stable pose of the object separately using Thompson sampling and a learned Dex-Net prior [17]. However, the small set of initial grasps used by BORGES may not include the most robust grasp. To address this issue, we propose a new algorithm, LEGS, which adaptively curates sets of promising grasps by adding and removing grasp candidates during exploration. By doing this, LEGS is able to converge to better long-term performance than BORGES (which uses a small fixed set of grasps), while also learning to robustly grasp an object faster than baselines which seek to explore the entire, much larger, set of grasp candidates.

III. PROBLEM STATEMENT

Overview: Consider a difficult-to-grasp polyhedral object of unknown geometry that rests on a planar surface and is viewed by an overhead RGBD camera. Our objective is to efficiently learn to successfully grasp the object in all of its stable poses.

Problem Setup: Given a polyhedral object $o$, let $N$ be its number of stable poses. Each stable pose $s \in \{1, 2, \ldots, N\}$ is associated with a landing probability $\lambda_s$, which indicates the probability of the object landing in pose $s$ when released from sufficient height in a randomized orientation [8, 22]. Following Danielczuk et al. [5], we model our problem as a finite-horizon Markov Decision Process $M = (S, A, T, R, H)$. We let $S$ be the set of equivalence classes of distinguishable stable poses of the object and $A$ be the set of all possible grasps on the object. Thus, $A = \bigcup_{s \in S} A_s$, where $A_s$ are the grasps available at a stable pose $s$. Given a grasp action $a$ in stable pose $s$, the transition function $T : S \times A \times S \to [0, 1]$ determines the probability distribution over next stable poses. The reward function $R : S \times A \to \{0, 1\}$ is binary: a grasp is successful and $R(s, a) = 1$ if the grasped object does not fall from the gripper after it is lifted, and $R(s, a) = 0$ otherwise. Let $p_s = \mathbb{E}[R(s, a)]$ be the expected success probability of grasp $a$ on stable pose $s$. We define a grasping policy as: $\pi : S \times A \to [0, 1]$, where $\pi(a|s)$ denotes the probability of selecting grasp $a$ in pose $s$. We denote the finite horizon of the MDP as $H$. If a grasp is successful, the robot randomizes the orientation of the object in the gripper and drops the object so that the new stable pose $s'$ is determined by the landing probabilities $\{\lambda_s\}_{s=1}^N$.

We represent the actions, $A_s$, at each stable pose $s$ as candidate grasps sampled on the object. We use the same method as Mahler et al. [21] to sample antipodal grasps on each stable pose. We do not make any assumptions on the grasping modality, so in practice these grasps can be sampled from various different grasp planners, including parallel-jaw or suction grasp planners. We denote the number of possible grasps for pose $s$ as $K_s = |A_s|$ and the total number of grasps over all states as $K = \sum_{s \in S} K_s$.

An important difference between our problem setting and prior work [5] is that we consider settings in which $K$ is large ($> 1000$) and thus is of the same order of magnitude as the exploration horizon, $H$. This significantly exacerbates exploration challenges, since there is not enough time to fully explore each grasp, motivating the key innovations in LEGS.

Assumptions: In this work, we assume access to the following: (1) a grasp sampler which accepts as input a depth map and outputs a set of candidate grasp configurations on the surface of the depth map with associated robustness values; (2) a robot/gripper that can either execute any of these grasps or detect that they are in collision; (3) sufficient information in the camera image to detect whether the object stable pose changes; (4) an evaluation function to detect whether a grasp is successful. We note that these assumptions are satisfied by the system we build to instantiate LEGS in practice. In addition, we make the following assumptions about object’s
interaction with the environment: (5) if a grasp is unsuccessful, the object either remains in the same stable pose or topples into another stable pose; and (6) there exists a grasp with non-zero success probability on each stable pose. These last two assumptions are consistent with [5].

**Metrics:** We define the *optimality gap*, $\Delta_\pi$ as

$$
\Delta_\pi = \mathbb{E}_{s \in S} \left[ p_s^* - p_{\pi(s)} \right] = \sum_{s \in S} \lambda_s \cdot \left( p_s^* - p_{\pi(s)} \right),
$$

(III.1)

where $p_{\pi(s)} = \max_{a \in A_s} \mathbb{E}[R(s, a)]$ and $p_{\pi(s)} = \mathbb{E}[R(s, \pi(s))]$. Intuitively, the optimality gap $\Delta_\pi$ measures the expected difference, across all stable poses, between the optimal policy, which selects the best available grasp, and the policy $\pi$.

The objective is to find a policy that minimizes the optimality gap for a given object within $H$ grasp attempts. Denoting a policy learned after $H$ grasp attempts by $\pi_H$, the objective is to identify $\pi_H^*$ such that:

$$
\pi_H^* = \arg \min_{\pi_H} \Delta_{\pi_H}.
$$

(III.2)

### IV. LEARNED EFFICIENT GRASP SETS

We propose Learned Efficient Grasp Sets (LEGS), a multi-armed bandits algorithm that uses confidence bounds on grasp success probability to maintain a small active set of candidate grasps. LEGS builds off BORGES [5], an algorithm that learns to grasp an object by exploring grasps across each of the object stable poses. However, while BORGES starts with a small, fixed set of grasps sampled from each of the object’s stable poses and restricts grasp exploration within this set, LEGS iteratively refines an active set of grasps for exploration by pruning poorly-performing grasps from consideration and replacing them with newly sampled grasps.

LEGS starts with an estimate of the prior success probabilities for all grasps in a large reservoir of possible grasps, and updates their grasp-success probabilities based on online grasp trials using Thompson sampling as in Danielczuk et al. [5]. However, unlike BORGES, LEGS uses the priors and online grasp trials to construct confidence bounds on the grasp-success probabilities for each grasp (Section IV-A).

LEGS is summarized in Algorithm 1. LEGS first initializes an active set of candidate grasps, $A_s$, along with the parameters of a Beta distribution associated with each grasp in the active set (lines 2-3). For each stable pose, we rank the grasps in the reservoir by their estimated grasp success probabilities under the Grasp Quality Convolutional Neural Network (GQ-CNN) from Dex-Net 4.0 [21] and select the $k = 100$ grasps with the highest values. As in [5], we also ensure that there exists at least one grasp with non-zero success probability in the initial active set. In each iteration, LEGS executes the grasp with the highest sampled value from the posterior (lines 5-6), observes the outcome (line 8), and updates the posterior distribution (lines 9-12). In conjunction, LEGS also constructs confidence bounds on each of the success probabilities of each grasp (Section IV-A). Every $n$ iterations, it uses these confidence bounds to identify and remove the grasps with low robustness (Section IV-B) (line 14), and replaces them with newly sampled grasps where grasps are ranked by their estimated grasp success probabilities under GQ-CNN (lines 15-16).

#### A. Constructing Confidence Bounds on Robustness

To determine which grasps to remove from the active set, LEGS constructs upper and lower confidence bounds on grasp robustness. We model the success probability of grasp $i$ via $X_i \sim \text{Beta}(\alpha_i, \beta_i)$, and empirically select a confidence threshold $\delta$. Then the percent-point function $\text{PPF}(X_i, \delta)$, the inverse of the cumulative distribution function $F_{X_i}(x)$, returns the value $x$ such that $F_{X_i}(x) = \delta$. The $(1-\delta)$-lower and -upper confidence bounds for $X_i$ are $X_i^{\ell} = \text{PPF}(X_i, \delta)$ and $X_i^{u} = \text{PPF}(X_i, 1-\delta)$, respectively. As a grasp is sampled more often, the interval $[X_i^{\ell}, X_i^{u}]$ tightens to reflect increased certainty in the robustness of the grasp.

#### B. Posterior Dependent Grasp Removal

LEGS avoids over-exploring less robust grasps by identifying and removing grasps from the active set that are highly likely to be either (1) inferior to another grasp in the active set (locally suboptimal) or (2) below a desired global grasp success probability threshold (globally suboptimal). Let the highest lower confidence bound across all active grasps be:

$$
X_i^{\ell} = \max_{i \in A_s} X_i^{\ell}.
$$

(IV.1)

We define the set of locally suboptimal grasps as the set of grasps for which their $(1-\delta)$-confidence upper bound is worse than the $(1-\delta)$-confidence lower bound for the best grasp in the active set:

$$
B_{\ell} = \{i : X_i^{u} < X_i^{\ast}\}.
$$

(IV.1)

Thus, $B_{\ell}$ represents the set of grasps that are likely to be inferior to the best known grasp in the active set.

However, in the early stages of exploration, we may not yet have sampled a high-performing grasp and $B_{\ell}$ may be empty. For example, if all the grasps initially sampled are very poor grasps, then even though no grasp may satisfy the condition for inclusion in $B_{\ell}$, we may still remove and resample all the grasps that we believe, with high-confidence, are clearly low performing. Thus, given a minimum performance threshold $\gamma \in [0, 1]$, we define a set of globally suboptimal grasps in the active set which have been sampled, but are likely to have success probability less than $\gamma$ (denoted $B_{\gamma}$). Define the set of attempted grasps in the active set as $P$. Then, we define $B_{\gamma}$ as follows:

$$
B_{\gamma} = \{i : X_i^{u} < \gamma, i \in P, i \neq i\}
$$

(IV.2)

In summary, the full set of grasps removed by LEGS is constructed by taking the union of the above sets: $B = B_{\ell} \cup B_{\gamma}$. This allows LEGS to simultaneously remove grasps which are are unlikely to outperform the best known grasp in the current active set, as well as filter out grasps which are highly likely to be low-performing in general.

#### C. Early Stopping

Rather than setting the exploration horizon $H$ to a fixed number of iterations, we can set a performance threshold and let LEGS stop exploring grasps once it has high confidence that it has achieved the desired performance threshold. This
early stopping condition allows LEGS to efficiently allocate exploration time by only continuing to spend time exploring objects that it cannot yet robustly grasp.

Given a user-specified, minimum performance threshold $\rho_{\text{min}} \in [0, 1]$, we want to detect when, with high likelihood, the true performance of LEGS is above this threshold. More formally, given a confidence parameter $\delta_{\text{stop}} \in [0, 1]$, we want to calculate a $(1 - \delta_{\text{stop}})$-confidence lower bound, denoted by $p_t$, on the true expected performance of the grasping policy $\pi$, i.e., we want to find $p_t$ such that:

$$\Pr\left(p_t \leq \mathbb{E}_{s \in S}[p_{\pi(s)}]\right) \geq 1 - \delta_{\text{stop}}. \quad (IV.3)$$

Given the high-confidence lower bound $p_t$ described above, the robot can stop exploring when $p_t \geq \rho_{\text{min}}$.

We now discuss how to obtain an empirical estimate, $\hat{p}_t$, of the above high-confidence lower bound, $p_t$. Note that we cannot directly compute $\mathbb{E}_{s \in S}[p_{\pi(s)}]$ since we do not know the true stable pose distribution $S$. Thus, we take a Bayesian approach where we approximate $p_t$ by sampling likely values from the posterior distribution over the observed data, and then taking the $\delta_{\text{stop}}$-percentile of these samples. First, for each observed stable pose, $s$, we estimate the expected performance of the best grasp as $p_s^\ast = \max_{i \in A_s} \frac{\alpha_i}{\alpha_i + \beta_i}$, where $\alpha_i$ and $\beta_i$ are the parameters of the Beta posterior distribution over the success probability of grasp $i$. To reason about the performance of LEGS, we must also account for uncertainty over the stable pose distribution, parametrized by the drop probabilities $\lambda_1, \ldots, \lambda_N$. However, $N$ is unknown. Thus, we model our belief over drop probabilities using a Dirichlet posterior distribution over $N + 1$ drop probabilities, where $N$ is the number of stable poses visited so far by LEGS and the plus one allocates probability mass to unobserved stable poses.

Assuming a uniform Dirichlet prior, we take the empirical drop counts $c_1, \ldots, c_N$ for $N$ seen stable poses, and sample from the posterior distribution over stable pose drop probabilities, $Pr((\lambda_s)_{s=1}^{N+1} \mid c_1, \ldots, c_N, 0)$. Due to conjugacy [6], the desired posterior distribution is also a Dirichlet distribution with parameters $(\alpha_1 = c_1 + 1, \ldots, \alpha_N = c_N + 1, \alpha_{N+1} = 1)$. Given a sample, $(\lambda_s)_{s=1}^{N+1}$, from the above Dirichlet posterior, we transform it into a sample from the posterior over expected grasp robustness, $p_s^\ast = \frac{\sum_{s=1}^{N} \hat{p}_s^\ast \lambda_s}{\sum_{s=1}^{N} \lambda_s}$, where we conservatively assume that the robot will fail to grasp the object in any unseen poses. We calculate a $(1 - \delta_{\text{stop}})$-confidence lower bound on the overall grasp robustness for an object by finding the $\delta_{\text{stop}}$-percentile: $\hat{p}_t = \text{PPF}(p_{\pi^\ast, \delta_{\text{stop}}})$. We estimate this empirically using $M$ samples of $p_{\pi}$.\n
V. SIMULATION EXPERIMENTS

A. Experimental Setup

We first evaluate LEGS in exploratory grasping with a variety of adversarial objects in simulation. As in Danielczuk et al. [5], we consider 14 Dex-Net 2.0 adversarial objects [20] and all 39 EGAD! Adversarial evaluation objects [23]. We use Dex-Net 4.0 [21] to sample a large reservoir of $K = 2000$ grasps for each stable pose. We also use GQ-CNN to set the beta prior for LEGS following the method from [5, 17].

B. Baselines

We compare LEGS against four baseline algorithms: DexNet, BORGES ($K_s = 100$), BORGES ($K_s = 2000$), and LEGS (-AS). Dex-Net greedily chooses the best grasp evaluated by Dex-Net 4.0 [21] for each stable pose and does not do any online exploration. BORGES ($K_s = 100$) leverages a prior calculated by GQ-CNN to seed grasp success probability estimates, and then performs Thompson Sampling for each encountered stable pose to explore an initial active set of $k = 100$ grasps sampled on each of the poses. While BORGES ($K_s = 100$) is provided with the same initial active set as LEGS, unlike LEGS, BORGES ($K_s = 100$) does not update this initial active set over time. BORGES ($K_s = 2000$) is identical to BORGES ($K_s = 100$), but instead directly explores the full reservoir of $K_s = 2000$ sampled grasps. Finally, LEGS (-AS) is not provided with an initial active set, but instead operates on the full reservoir of $K_s = 2000$ grasps and uses the posterior dependent removal procedure in Section IV-B to remove grasps from the reservoir.

C. Experimental Results

We first study aggregated results of LEGS and baselines over objects in the Dex-Net Adversarial and EGAD! evaluation datasets in Table I. On all of these objects we find that Dex-Net, which is not updated online, achieves relatively poor performance (high optimality gap) across all objects considered, motivating online grasp exploration. Results

<table>
<thead>
<tr>
<th>Algorithm 1: Learned Efficient Grasp Sets (LEGS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> object $o$, pose $s$, grasp sampler $f_\theta$; resample interval $n$, number of active grasps $k$</td>
</tr>
<tr>
<td><strong>Use</strong> $f_\theta$ to sample $k$ grasps as the active set $A_s$</td>
</tr>
<tr>
<td><strong>For each grasp $i \in A_s$, set $\alpha_i$ and $\beta_i$ based on prior</strong></td>
</tr>
<tr>
<td><strong>for</strong> $t = 1, 2, \ldots$ <strong>do</strong></td>
</tr>
<tr>
<td><strong>foreach</strong> grasp $i \in A_s$ <strong>do</strong></td>
</tr>
<tr>
<td><strong>sample</strong> $\phi_i \sim \text{Beta}(\alpha_i + 1, \beta_i + 1)$</td>
</tr>
<tr>
<td><strong>Execute grasp</strong> $i = \arg\max_j \phi_j$</td>
</tr>
<tr>
<td><strong>Observe reward</strong> $r = R(s, i)$</td>
</tr>
<tr>
<td><strong>if</strong> $r = 1$ <strong>then</strong></td>
</tr>
<tr>
<td>$\alpha_i \leftarrow \alpha_i + 1$</td>
</tr>
<tr>
<td><strong>else</strong></td>
</tr>
<tr>
<td>$\beta_i \leftarrow \beta_i + 1$</td>
</tr>
<tr>
<td><strong>if</strong> $t \equiv 0$ <strong>(mod</strong> $n$) <strong>then</strong></td>
</tr>
<tr>
<td><strong>Remove the grasps in</strong> $B = B_i \cup B_\gamma$ <strong>from</strong> $A_s$ (equations (IV.1) and (IV.2))</td>
</tr>
<tr>
<td><strong>Sample</strong> $</td>
</tr>
<tr>
<td><strong>For each new grasp</strong> $j = 1, \ldots</td>
</tr>
</tbody>
</table>

Using the method outlined in Section IV, we update the active grasp set after every $n = 100$ timesteps and use $\delta = 0.01$ for constructing grasp confidence intervals with upper confidence threshold $\gamma = 0.1$. All experiments are run over a time horizon of $H = 5000$. Additionally, we run 10 trials of each algorithm with 10 rollouts per trial, where each trial involves sampling a different reservoir of grasps, and each rollout for a trial involves running a grasp exploration algorithm.
suggested that LEGS is able to converge to a lower optimality gap than baselines. We find that LEGS achieves an average optimality gap 20% and 25% lower than the best baseline on the Dex-Net Adversarial and EGAD! datasets respectively. The improvement in performance between LEGS over LEGS (-AS) and BORGES ($K_s = 2000$) indicates the increased efficiency of restricting exploration within a compact active set, while the gap between LEGS and BORGES ($K_s = 100$) indicates the importance of updating this active set over time to rapidly prune out poor performing grasps while discovering new, high-quality grasps outside of the initial active set. Notably, BORGES ($K_s = 100$) cannot outperform the success rate of the best grasp in its initial set ($K_s = 100$ upper bound). By contrast, LEGS, retains the efficiency of only exploring a small set of grasps while also being able to adapt this set over time to reach a lower optimality gap.

In Figure 2, we study LEGS and baselines on specific objects to better understand what object properties affect the difficulty of grasp exploration. We selected two objects (Climbing Hold and B6) where LEGS outperforms prior algorithms and selected two objects (F6 and Turbine Housing) where LEGS does not outperform all baselines. Somewhat surprisingly, we find that the distribution over ground truth grasp success probabilities does not seem to have a significant effect on the relative performance of LEGS and baselines. However, we do find that LEGS tends to perform more poorly on objects where the GQ-CNN prior is less accurate. This makes intuitive sense, since LEGS relies on the prior when sampling new grasps to add to the active set.

### TABLE I: Grasping in Simulation Aggregated Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dex-Net</th>
<th>BORGES ($K_s = 100$)</th>
<th>$K_s = 100$ Upper Bound</th>
<th>BORGES ($K_s = 2000$)</th>
<th>LEGS (-AS)</th>
<th>LEGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dex-Net Adversarial</td>
<td>0.62 ± 0.02</td>
<td>0.10 ± 0.02</td>
<td>0.09 ± 0.01</td>
<td>0.06 ± 0.02</td>
<td>0.12 ± 0.02</td>
<td>0.05 ± 0.01</td>
</tr>
<tr>
<td>EGAD!</td>
<td>0.58 ± 0.02</td>
<td>0.10 ± 0.01</td>
<td>0.09 ± 0.01</td>
<td>0.15 ± 0.02</td>
<td>0.17 ± 0.02</td>
<td>0.08 ± 0.01</td>
</tr>
</tbody>
</table>

Fig. 2: Simulated Grasping Experiments Case Study: We report the performance of LEGS and baselines on four specific objects to investigate how object properties affect performance. For each object, we include a 3D rendering of the object and the number $N$ of stable poses (left), a histogram of the ground truth grasp success probabilities over 2000 sampled grasps (middle), and learning curves for LEGS and baselines (right). We also include a measure of the mismatch ($\tau$) between the GQ-CNN prior and the ground truth grasp success probabilities as defined in [17] (0 indicates low mismatch, 1 indicates high mismatch). These results suggest that LEGS performs worse when there is a higher prior mismatch ($\tau = 0.5$), as LEGS uses the prior when sampling new grasps for inclusion in the active set.

### D. Early Stopping Results

Next, we study the accuracy and effectiveness of the early stopping criterion (Section V-D). We tested the proposed high-confidence performance bound across all objects in the Dex-Net Adversarial object set (individual results per object are reported in the supplement). We check whether LEGS has reached the stopping condition every 100 grasps for a horizon of $H = 5000$ grasp attempts and use $\delta_{\text{stop}} = 0.05$, resulting in a 95%-confidence lower bound $\hat{\rho}_{\text{stop}}$. We first tested how often the predicted lower bound actually lower bounds the true performance. On average, across all Dex-Net Adversarial objects, our empirical lower bound is a 95.8%-accurate lower bound on the true performance over the true stable pose distribution. Thus, $\hat{\rho}_{\text{stop}}$ forms an empirically valid $(1 - \delta_{\text{stop}})$-confidence lower bound. We next tested the tightness of our lower bound. On average, the difference between the true performance of LEGS and our empirical lower bound is only 2.97%. These results suggest that our lower bound is highly accurate and tight enough to provide a practical signal for when the robot can safely stop exploring.

We next explored early stopping. As described in Section V-D, given a user-specified, minimum performance threshold, $\rho_{\text{min}}$, the robot stops exploring when the lower confidence bound $\hat{\rho}_{\text{stop}}$ is greater than $\rho_{\text{min}}$. When the robot chooses to stop exploring the object, we evaluate the ground truth performance of the learned policy using the true grasp success probabilities and true stable pose distribution and evaluate whether the true performance is also above the threshold $\rho_{\text{min}}$. We perform this evaluation over a wide range of thresholds and plot the results in Figure 3. Results suggest that we can...
We plot the accuracy averaged over all objects and find that using our
where the object rebound height is always lower than the
\( \rho \) when using a range of stopping thresholds,
\( \rho_{\text{min}} \). All results use a
95%-confidence lower bound on expected grasp robustness. **Left:**
We plot the accuracy averaged over all objects and find that using our
empirical lower bound described in Section V-D is highly accurate
across all stopping thresholds, \( \rho_{\text{min}} \). **Right:** We plot the number
of steps before stopping, averaged across all objects. Intuitively,
the required exploration time increases with higher performance
thresholds. Importantly, the average number of steps before stopping
is much lower on average than the maximum 5000 step horizon,
even for high stopping thresholds.

achieve highly accurate early stopping across all stopping
thresholds. We find that the early stopping condition allows
the robot to accurately terminate exploration well before the
full horizon of 5000 steps, allowing the robot to save time
by terminating exploration early once it has high-confidence
that it has reached an acceptable level of performance.

**VI. PHYSICAL EXPERIMENTS**

In the following sections, we discuss our experimental
setup for physical experiments (Section VI-A), the methods
we used to enable intervention-free grasp exploration on
a physical robot (Section VI-B) and results evaluating the
performance of LEGS and BORGES (\( K_s = 2000 \)) across 3
physical objects (Section VI-C).

**A. Experimental Setup**

To deploy exploratory grasping algorithms on a physical
robot, we adopt the perception system introduced by Danielczuk
et al. [5] to sample grasps and identify changes in the
object stable pose. When the object lands in a new stable
pose, we use Dex-Net 4.0 [21] to sample, evaluate, and cache
antipodal grasps in the grasp reservoir. This makes it possible
to explore grasps on objects with unknown geometries and
unknown numbers of stable poses. However, Danielczuk et al.
[5] found that re-dropping the object during experiments often
causd it to fall out of the workspace, requiring extensive
human interventions to reset the object in practice [5]. In
the next section, we address this challenge by introducing
new methods that enable self-supervised grasp exploration in
physical trials (Section VI-B).

**B. Self-Supervised Exploratory Grasping with LEGS**

To enable the robot to collect grasp data on a variety
of different stable poses without human intervention, we
introduce strategies to prevent the object from toppling out
of the workspace while still maintaining access to a wide
variety of grasps. To prevent the object from falling out of
the workspace, we drop the object within a bowl (Fig.1),
where the object rebound height is always lower than the
rim of the bowl. This allows the object to always stay in the
visible range of the overhead camera. One challenge with this
strategy is that introducing a vertical physical boundary can
remove access to a number of grasps if the object falls near
the boundary, as grasps that are collision-free in a boundary-
free environment may now collide with the boundary. We
introduce two “reset” behaviors to address this. First, upon
successful grasps, we center the object above the center of
the bowl before dropping the object, reducing the likelihood
that the object settles into a position near the boundary of the
bowl. Second, when the object does topple near the boundary,
we execute a push from the rim of the bowl towards the center.
Due to the time-consuming nature of physical experiments,
we run 3 trials with 1 rollout per trial for each object.

**C. Experimental Results**

In Figure 4 we present learning curves from physical
experiments comparing LEGS with BORGES (\( K_s = 2000 \)) on three challenging objects from the Dex-Net Adversarial
Dataset [20]. We find that on 2 out of the 3 objects, LEGS is
able to identify high-performing grasps within a few hundred
timesteps of online exploration while BORGES (\( K_s = 2000 \))
makes very little progress in this time.

**VII. DISCUSSION**

We present Learned Efficient Grasp Sets, an algorithm
which enables efficient exploration of large sets of possible
grasps by adaptively constructing a small active set of
promising grasps and rapidly pruning out grasps from
consideration that are clearly suboptimal. Experiments suggest
that LEGS identifies high-performing grasps more efficiently
than baseline algorithms on adversarial Dex-Net and EGDAD!
objects in simulation experiments and on three challenging
objects in physical trials. We also proposed a novel early
stopping condition by computing a high-confidence lower
bound on the expected grasp performance. Simulation results
show that this high-confidence lower bound is highly accurate
and tight and enables accurate early stopping. In future
work, we hope to analyze LEGS to determine how the
quality of the Dex-Net prior and the distribution over grasp
success probabilities affects its convergence rate and explore
applications outside of grasping.
REFERENCES


