No-Regret Shannon Entropy Regularized Neural Contextual Bandit Online Learning for Robotic Grasping

Kyungjae Lee, Jaegu Choy, Yunho Choi, Hogun Kee, and Songhwai Oh

Abstract— **In this paper, we propose a novel contextual bandit algorithm that employs a neural network as a reward estimator and utilizes Shannon entropy regularization to encourage exploration, which is called Shannon entropy regularized neural contextual bandits (SERN). In many learning-based algorithms for robotic grasping, the lack of the real-world data hampers the generalization performance of a model and makes it difficult to apply a trained model to real-world problems. To handle this issue, the proposed method utilizes the benefit of an online learning. The proposed method trains a neural network to predict the success probability of a given grasp pose based on a depth image, which is called a grasp quality. We theoretically show that the SERN has a no regret property. We empirically demonstrate that the SERN outperforms ϵ-greedy in terms of sample efficiency.**

I. Introduction

Recent advances of deep learning have enabled to develop many robotics applications that employ high dimensional observations such as point clouds [1], or depth image [2]. In particular, a convolutional neural network (CNN) has shown powerful performances in many image-based data-driven methods [1]–[13]. The benefit of a CNN has been widely employed in a robotic grasping problem [1], [2], [4]. For example, a CNN is utilized to predict a grasp success probability of given depth images of an object and corresponding grasp poses [2] or to generate high-quality grasp poses from an image of an object [1].

However, while a CNN has a large capacity to learn high dimensional data, it often suffers from an over-fitting problem when the number of training data is small, which leads to poor prediction results for unseen situations. Hence, most existing data-driven robotic grasping methods employing CNNs have focused on generating or collecting enough training data [2]–[4]. In [3], training data are generated by 3D mesh data and by using dynamic simulators where a depth image of objects is synthesized and corresponding grasp poses are generated geometrically from the mesh data. Using these data set, Mahler et al. [2] successfully trained a neural network to predict a grasp pose given a depth image and empirically showed that the trained network can be applied to the the real-world grasping. In [4], Mahler et al. extends [2] to a the real-world bin picking problem, which is a sequential grasping problem, by augmenting the real-world grasping data. However, using simulated data to train a neural network has the limitation since there exists a discrepancy between simulations and the real-world environment as mentioned in [5]–[9]. In particular, a synthesized depth image has a different visual property from that of the the real-world. Furthermore, when it comes to dynamics, contact simulations may be inaccurate and not similar to the real-world phenomena. To handle this discrepancy, [5]–[9] have incorporated domain adaptation and domain randomization techniques which diversify the parameters of simulations to cover various types of dynamic environments when training data are collected.

While diversifying simulation environments can alleviate the lack of data, covering all possible situations using simulated data requires heavy computational loads and expensive costs. Hence, we augment efficiently the real-world data using an online learning instead of utilizing simulated data. In general, online learning methods have been widely used for a robot to adapt to unexpected situations by autonomously exploring an environment [14], [15]. Since collecting data using a real robot is a time-consuming, efficient exploration method plays a crucial role in practice. In [13], [16], ϵ-greedy method is employed where a grasp is sampled from a uniform distribution for exploration with probability ϵ. However, the uniformly random exploration is inefficient since it does not employ the grasp quality estimated by a neural network. Hence, we apply a softmax distribution whose probability of a grasp pose is exponentially weighted by its quality estimation.

In this paper, we propose a no regret Shannon entropy regularized neural contextual bandit algorithm for robotic grasping where the Shannon entropy is employed to encourage explorations. Since the Shannon entropy induces a softmax distribution of the grasp quality, The probability of exploring a grasp pose is exponentially weighted by the estimated success probability. Hence, the proposed method randomly explores various grasp poses, but the promising grasp poses which have highly estimated qualities are explored more frequently. We also prove that the proposed method converges into optimal policy efficiently, which is called no regret property. For practice, to prevent an over-fitting issue of a CNN, we augment a core set of synthetic datasets. In simulations, we apply the SERN to learn to grasp unknown 3D meshes, and empirically show that the SERN improves a grasp success rate by at most 212%. In the real-world experiment, the SERN improves a grasp success rate of three unseen objects from 0% to 80%.

II. Related Work

Many existing robotic grasp methods have been developed based on data-driven approaches [1]–[4], [7], [10]–[13], [17]. In general, these methods train two types of networks: grasp quality network and grasp proposal network. The grasp quality network predicts the success probability of given grasp pose and information about an object to be grasped where object information is generally given by a depth

K. Lee, J. Choy, Y. Choi, H. Kee, and S. Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea (e-mail: {kyungjae.lee, jaegu.choy, yunho.choi}@rllab.snu.ac.kr, {hogunkee, songhwai}@snu.ac.kr)

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning).

image, RGB image, or both. The grasp proposal network generates a grasp pose of end-effector based on given inputs such as RGBD image. Most existing methods focused on how to generate training data for a deep neural network and how to generate simulated data for robust transfer and generalization in the real-world. While [1]–[4], [7], [10]–[13], [17] have been shown powerful results, however, learning-based methods have the limitation in that a grasp performance can degenerate for unseen objects which are not included in training data. To handle this issue, online learning approaches for robotic grasping have been investigated where a robot is trained with sequentially generated data during the test phase to adapt to unseen objects.

A. Robotic Grasping with Deep Learning

Learning-based robotic grasping approaches have been developed [1]–[4], [18] where grasp poses are often predicted by a deep neural network instead of computing grasp poses from geometric information. These methods often utilize a convolutional neural network (CNN) which shows high performance for image data [2]–[4]. To train a CNN for predicting the success probability of a grasp pose, a large number of training examples are required. In [2], training data are collected in simulation using given 3D mesh data. On the contrary, image based methods [1], [18] predict grasp poses from point cloud data. In [1], point cloud data and corresponding grasp poses are learned with conditional variational auto-encoder where, in testing time, grasp candidates are generated using the decoder.

While existing learning-based methods [1]–[4], [18] have demonstrated that a deep neural network trained with synthetic data can generate grasp poses of unseen objects, there still exists an issue about applying such model to the real-world grasping. Due to a discrepancy between simulation and the real-world, the network trained with synthetic data shows poor generalization performance.

B. Online Learning in Robotic Grasping

Even if existing learning-based grasping methods including sim-to-real methods handle the discrepancy, the lack of training data for unseen objects which have never been simulated is still an issue. In particular, since the dynamic contact simulation has an error compared to the real-world contacts, the training data collected by simulations may be imperfect in that they do not reflect the real-world contacts.

To handle this issue, the real-world data augmentation is essential to reduce the gap between simulation and the real-world dynamics. In [16], Dmitry et al. proposed a deep reinforcement learning (RL) approach, called QT-Opt, which learns to grasp with multiple robots using deep Q learning. Dmitry et al. employed ϵ-greedy method for exploration. In [13], Berscheid et al. also employed ϵ-greedy method or normalized probability method whose probability is obtained by normalizing the grasp quality.

III. Background and Problem Formulation

In this section, we introduce a contextual bandit problem and formulate a depth image based robotic grasping problem in a contextual bandit framework.

A. Contextual Bandit Problem

A contextual bandit problem is defined by a tuple with three elements: $\{\mathcal{S}, \mathcal{A}, \mathbf{R}\}$, where $\mathcal{S}$ is a context space, $\mathcal{A}$ is an action space, $\mathbf{R}$ is a reward which is a random variable indicating goodness of an action given a context. We assume that $\mathbf{R} \in [0, 1]$. Then, the expected reward of pulling $a \in \mathcal{A}$ given $s \in \mathcal{S}$ is defined as a conditional expectation of the reward, $r_a(s) := \mathbb{E}[\mathbf{R}|s, a]$. The goal of a contextual bandit problem is to find the best arm whose expected reward is the maximum by consecutively pulling arms and obtaining contexts and rewards every round.

A robot plays T rounds of grasping. At the t round, an arbitrary context s_t is given, then, a contextual bandit algorithm proposes a policy π_t based on s_t and sample an action a_t from π_t. The feedback of a_t is given as a reward $\mathbf{R}_t$. Since an expected reward $r_a(s_t)$ of each arm is unknown, rewards of each arm given s_t should be estimated. To estimate the expected rewards, $\hat{r}_a(s; \theta)$ is maintained where θ is the parameter of an estimator. $\hat{r}_a(s; \theta)$ is trained from the collected context and reward pairs. Generally, as the number of data increases, the error of reward estimations decreases. After estimators become accurate, the best arm can be selected based on $\hat{r}_a(s)$. Collecting more data to train $\hat{r}$ more accurately is called exploration and choosing the estimated best arm based on $\hat{r}$ is called exploitation. The main hurdle of a bandit problem is the trade-off between exploration and exploration.

The efficiency of a bandit algorithm is often measured by the expected cumulative regret defined as $\mathcal{R}_T := \sum_{t=1}^{T} \max_{a'} \mathbb{E}_{s_{1:t}, a_{1:t-1}} [r_{a'}(s_t) - r_{a_t}(s_t)]$, where $s_{1:t}$ indicates contexts given during t rounds and $a_{1:t-1}$ indicates actions selected during $t-1$ rounds. If the algorithm focuses on exploring arbitrary arms, $\mathcal{R}_T$ linearly increases. On the contrary, if the exploitation is focused, the estimation error of rewards is hardly reduced and $\mathcal{R}_T$ also linearly increases. When $\mathcal{R}_T$ increases sub-linearly, such algorithms are called no regret and the error converges to zero as the number of rounds increases, i.e., $\lim_{T \to \infty} \frac{\mathcal{R}_T}{T} = 0$.

B. Problem Formulation

We formulate the problem of finding a grasp pose using a depth sensor as a contextual bandit problem. For a grasping problem, $\mathcal{S}$ is depth images that contain information about objects to grasp and $\mathcal{A}$ is grasp poses of a gripper of a manipulator and we assume a parallel jaw gripper. Hence, a grasp pose is defined as a four dimensional vector that combines a grasp point x, y, z and a rotation angle θ with respect to the z-axis. All continuous variables are discretized. The reward $\mathbf{R}$ is defined as a binary random variable that indicates the success of grasping. $\mathbf{R} = 1$, if grasp is succeeded, $\mathbf{R} = 0$, otherwise. Then, a deep neural network estimates the expected rewards $r_a(s)$ that indicates the success probability of grasping, similarly to prior work [2]–[4].

IV. Shannon Entropy Regularized Neural Contextual Bandit Algorithm

In this section, we propose a Shannon entropy regularized neural contextual bandit algorithm (SERN) by utilizing an artificial neural network as a reward estimator and exploring various actions due to entropy maximization. The main difference of the proposed method from existing regularized bandit algorithms is that we do not assume unbiased estimation such as a linear model or Gaussian process regression.

Furthermore, we analyze the upper bound of the cumulative regret of SERN and show that it is no regret. Thus, the proposed method enables to use a neural network which has the large capacity and shows a powerful performance for high dimensional data while maintaining no regret property.

A. Shannon Entropy Regularization for Exploration

For each round, the SERN estimates rewards $\hat{r}_a(s_t;\theta)$ for given depth image s_t where θ is a parameter of a neural network, and computes a policy π_t which is computed as follows:

$$\pi_t(s_t) := \arg\max_{\pi\in\Pi} \left\{\mathbb{E}_{a\sim\pi}\left[\hat{r}_a(s_t;\theta_{t-1})\right] + \alpha S(\pi)\right\}, \quad (1)$$

where $S(\pi) := \mathbb{E}_{a\sim\pi}[-\ln([\pi]_a))]$ is the Shannon entropy, and α is a regularization coefficient. It is well known fact that the solution of (1) is a softmax distribution. Hence, $\pi_t(s_t) = \exp(\hat{r}_{a'}(s_t;\theta_{t-1})/\alpha)/\sum_{a'}\exp(\hat{r}_{a'}(s_t;\theta_{t-1})/\alpha)$.

We sample a grasp pose $a_t \sim \pi_t$ and try the sampled grasp by controlling a robot. We collect a set of context, action and reward $(s_t, a_t, \mathbf{R}_t)$ and update the parameter θ_{t-1} to θ_t based on collected data using a stochastic gradient descent to minimize the estimation error.

In practice, since it is not possible to run a the real-world robot infinitely often, it is important to improve sample efficiency. Furthermore, a neural network has the limitation in that, if the number of training data is not enough, the network is easily over-fitted which causes a poor generalization performance. To handle this issue, we utilize a pretrained model from [2] and augment a subset of pretraining data to newly collected data.

B. Pretrained Model and Core Set Selection

We initialize the grasp quality network using the pretrained model. By doing so, the sample efficiency can be improved by rejecting trivially infeasible grasp poses. In this paper, the pretrained model of [2] is utilized but we would like to note that the other model also can be used. After collecting several trials, we fine-tune the pretrained model to adapt to unseen objects. We initialize a parameter of grasp quality network with θ_0 of the pretrained model.

Let us denote the training data set used for pretraining as $\mathcal{D}_0$, and the newly collected data from online learning as $\mathcal{D}$. When we fine-tune the pretrained model, catastrophic forgetting [19] occurs where the model forgets the information of $\mathcal{D}_0$ after fine-tuning the network over newly collected dataset $\mathcal{D}$. Since the number of data $\mathcal{D}$ can be limited, fine-tuning the grasp quality network can be easily over-fitted to $\mathcal{D}$ and can easily forget the pretrained information $\mathcal{D}_0$.

To prevent a catastrophic forgetting, we select a subset of pretraining data $\mathcal{C} \subset \mathcal{D}_0$ and augment them to $\mathcal{D}_0$ during the online learning. While using entire set $\mathcal{D}_0$ may improve the entire performance and more effectively prevent catastrophic forgetting, it has disadvantages in that a large number of data set requires more computational resources and makes the whole learning processes slow. Thus, we select an important subset from $\mathcal{D}_0$ by employing a determinantal point process method (DPP) [20] which can sample k points covering $\mathcal{D}_0$ uniformly. We first transfer all training data to a feature space using the pretrained CNN and choose 172 subset of $\mathcal{D}_\prime$ using the DPP on features of data points. The entire algorithm is summarized in Algorithm 1.

Algorithm 1 Shannon Entropy Regularized Neural Contextual Bandit Algorithm (SERN) for Grasping

Input: $p \in (0, 1/3)$, T, θ_0, $\mathcal{D}_0$, and k
Initialize $\alpha = 1/\ln(T^p)$
$\mathcal{D} = \text{DPP}(\mathcal{D}_0, k)$ [20]
for $t = 1, \cdots, T$ **do**
  A context s_t is given and agent chooses $a_t \sim \pi_t$ where $\pi_t := \arg\max_\pi \left\{\mathbb{E}_{a\sim\pi}\left[\hat{r}_a(s_t;\theta_{t-1})\right] + \alpha S(\pi)\right\}$
  Agent gets a reward $\mathbf{R}_t$ and stores $(s_t, a_t, \mathbf{R}_t)$ into $\mathcal{D}$
  $\theta_t = \arg\max_\theta \sum_{(s_t,a_t,\mathbf{R}_t)\in\mathcal{D}} |\mathbf{R}_t - \hat{r}_{a_t}(s_t;\theta)|$
end for

V. Theoretical Analysis

In this section, we provide a theoretical analysis of our algorithm. We prove that a regret of the proposed method grows sub-linearly and, thus, it has no regret property. By deriving the no regret property of the SERN, it is guaranteed that using a neural network with the softmax policy can find an optimal policy of the contextual bandit problem. Before starting the analysis, we introduce some assumptions for a reward function, a neural network, and its error bounds.

A. Assumption

Assumption 1 (Separable Reward Structure)**.** *Define the reward gap as* $\Delta_a(s) = \max_{a'} r_{a'}(s) - r_a(s)$ *for given* s*. Note that* $\min_a \Delta_a(s) = 0$ *at the best arm* $a^\star = \arg\max_{a'} r_{a'}(s)$*. Let the second minimum reward gap be* $\Delta_2(s) = \min_{a\neq a^\star} \Delta_a(s)$*. Then, we assume that* $\Delta_2(s) > 0$ *for all* s *and define* $\Delta_2 := \min_s \Delta_2(s)$

Assumption 2 (Rewards Estimation)**.** *For each arm* a*, we have the reward estimator* $\hat{r}_a(s;\theta_{n_a})$ *where* n_a *is the number of training data for* $\hat{r}_a$ *collected by pulling* a *and* θ *is the parameter of* $\hat{r}_a$*. We assume that the parameter has the least error, i.e.,* $\theta_{n_a} = \arg\min_\theta \mathbb{E}_{s_{1:n_a}}\left[\sum_{i=1}^{n_a} |r_a(s_i) - \hat{r}_a(s_i;\theta)|\right]$.

Assumption 3 (Error Bound or Sample Complexity)**.** *We assume the upper bound of the error of a reward estimation as follows:* $\forall s \in \mathcal{S} \;\; \mathbb{E}\left[|r_a(s) - \hat{r}_a(s;\theta_{n_a})|\right] < \beta\sqrt{1/(n_a+1)}$ *where* β *is a positive constant depending on an estimation model and a learning algorithm. When* n_a *data are given, the expected error decreases proportionally to the square root of the number of data.*

Assumption 1 indicates a given reward function is not trivial. If $\Delta_2 = 0$, then, it means there is no second optimal arm and rewards for all arms are the same. Assumption 2 and 3 generally hold for a deep neural network. For Assumption 2, we believe that the best parameter for given training data can be achieved by using general optimization techniques for a deep neural network. For Assumption 3, in [21], Barron showed that the regression error bound of a neural network follows $O(1/\sqrt{n})$ and, in [22], Suzuki showed that it is bounded by $O(\log_+(\sqrt{n})/n) \leq O(1/\sqrt{n})$. Thus, our assumptions generally hold.

The proof strategy of no regret property consists of two parts. We first show that our algorithm explores every arm infinitely often. Then, we prove that infinite explorations eventually reduce the estimation error small enough and, after that, the best arm can be verified. While the proposed method explores every arm infinitely, the ratio of choosing each arm is exponentially proportional to its estimated rewards. Hence, we can achieve the sub-linearly growth of $\mathcal{R}_T$,

which is no regret. Note that the detail proofs are omitted here and can be found in the supplementary [23].

B. Exploration Ratio

In this section, we analyze the ratio of the time sub-optimal action is selected. Furthermore, this ratio controls the number of data for each reward estimation. Let $N_a(t)$ be a random variable indicating how many times an arm a is selected during t rounds.

Theorem 1. *For any arm a, the expected count has the following lower bound,* $\mathbb{E}\left[N_a(t)\right] \geq ct$ *where* $c = \frac{1}{K}\exp(-\frac{1}{\alpha})$.

Theorem 1 tells us that the lower bounds of $N_a(i)$ linearly grows. Since $\lim_{t\to\infty} ct = \infty$, the expectation of $N_a(i)$ goes to infinity. Thus, the proposed method explores every arm infinitely many. From this fact, it can be observed that every arm is explored infinitely many times. Using Theorem 1, we can derive the upper bound of the tail probability of $N_a(t)$.

Theorem 2. *For any arm a, let* $N'_t := N_a(t) - ct$. *Then,* N'_t *is submartingale and, from this fact, the following inequality holds, for any* $\delta > 0$, $\mathbb{P}(N_a(t) < ct - \delta) \leq \exp\left(-\delta^2/(8t)\right)$.

The proof can be found in the supplementary [23]. This theorem tells us that the probability that random variable $N_a(t)$ is below the expected lower bound has an exponential upper bound with respect to its deviation δ. Using this upper bound, we can control the error term $\beta\sqrt{1/(n+1)}$ of a neural network.

C. Upper Bounds for Expected Cumulative Regret

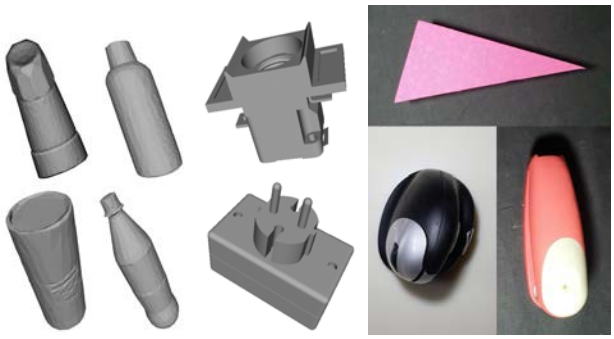
Now, we prove the no regret property of SERN. We first derive the general upper bound of the cumulative regret and derive more specific bounds by controlling α. Then, finally we show that the proposed method is no regret.

Theorem 3. *For* $\alpha > 0$ *and* $1 > q > 0$, *the expected cumulative regret of SERN is bounded by*

$$\beta\sum_{t=1}^{T}\mathbb{E}\left[(N_{a^\star}(t-1)+1)^{-\frac{1}{2}}\right] + \beta\sum_{t=1}^{T}\mathbb{E}\left[(N_{a_t}(t-1)+1)^{-\frac{1}{2}}\right] + \sum_{t=1}^{T}\mathbb{P}(a^\star \neq \hat{a}^\star_{t-1}) + \alpha\ln(K)T,$$

where $K = |\mathcal{A}|$, $a^\star = \arg\max_a \mathbb{E}_s\left[r_a(s)\right]$, *and* $\hat{a}^\star_t = \arg\max_a \mathbb{E}_s\left[\hat{r}_a(s;\theta_t)\right]$.

The first and third term comes from the estimation error of a neural network, the second term comes from the failure probability of a neural network for discriminating the best arm, and the last term indicates the regret induced by Shannon entropy regularization. Before deriving detail upper bounds, we would like to give some intuition of proof strategies for each term. The first and third term will be bounded by using Theorem 1 and 2. Furthermore, we prove that the second term has a constant bound using Assumption 1. Note that we assume that there always exists a positive gap Δ between the optimal and sub-optimal arms. Thus, if our estimation error becomes below Δ, then, after that point, we can discriminate the best arm from other sub-optimal arms. Finally, the last term will be bounded by controlling α. The entire bound can be derived as follows.



(a) KIT [24] (b) T-Less [25] (c) Real Objects

Fig. 1: Objects

Theorem 4. *Let* $\alpha = \frac{\alpha_0}{\ln(T^p)}$ *for* $\alpha_0 > 0$. *Then, the expected cumulative regret of SERN is bounded by*

$$\frac{C_0}{c_0^{3/2}}T^{\frac{3p+1}{2}} + C_1\left(1-\exp\left(-c_0^2 d_1 T^{-2p}\right)\right)^{-1} + C_2\left(1-\exp\left(-c_0^2 d_2 T^{-2p}\right)\right)^{-1} + \alpha_0\ln(K)T\left(\ln(T^p)\right)^{-1},$$

where $c_0 = \exp(-1/\alpha_0)$, $C_0 = 2^{\frac{7}{2}}K^{\frac{3}{2}}\beta$, $C_1 = 2\beta K$, $C_2 = 2(K-1)\exp((\beta/\Delta_2)^2 - 1/4)$, $d_1 = 1/(32K^2)$, *and* $d_2 = 1/(8K^2)$.

The first and second terms, the expectations of the estimation errors, are bounded by $O(T^{\frac{3p+1}{2}}) + O((1-\exp(-d_1T^{-2p}))^{-1})$. The third term $\mathbb{P}(a^\star \neq a^\star_{t-1})$ is bounded by $O((1-\exp(-d_2T^{-2p}))^{-1})$. The last term is bound by $O(T(\ln(T^p))^{-1})$. By using Theorem 4, we can show no regret property as follows.

Theorem 5. *For* $1/3 > p > 0$, *if the number of rounds, T, goes to infinity, then, time-averaged regret converges to zero:* $\lim_{T\to\infty}\mathcal{R}_T/T = 0$.

Theorem 5 tells us the proposed method eventually find the best arm for given context. Entire proof can be found in the supplementary [23]. Here, we provide a proof sketch. From Theorem 4, we have the upper bound of $\mathcal{R}_T$ which consists of four parts. First, since the first term and forth term follow $O\left(T^{\frac{3p+1}{2}}\right)$ and $O\left(T(\ln(T^p))^{-1}\right)$, respectively. Hence, $\lim_{T\to\infty} O\left(T^{\frac{3p+1}{2}}\right)/T = \lim_{T\to\infty} O(T^{\frac{3p-1}{2}}) = 0$. Furthermore, since $3p-1<0$, $\lim_{T\to\infty} O\left(T(\ln(T^p))^{-1}\right)/T = 0$. Finally, the limit of the second and third terms can be proven by showing that $\lim_{x\to\infty}(x(1-\exp(-ax^{-b})))^{-1} = 0$. The entire proof can be found in the supplementary [23].

VI. Experimental Results

To verify our theorems and effectiveness of the proposed exploration method, we conduct both dynamic simulation and the real-world experiments.

A. Setup

In the dynamic simulation, we compare three exploration methods including ours. First, a greedy method is compared as a baseline model which simply tries the best grasp whose estimated grasp quality is the maximum. Second, we compare a ϵ-greedy method which selects the best grasp pose with probability $1-\epsilon$ and chooses a uniformly random grasp with probability ϵ. The ϵ-greedy method has been widely used in many existing methods. For ϵ-greedy method, we set

Obj.	Alg.	Round 1.	Round 2.	Round 3.	Round 4.	Round 5.	Max.	Imprv.
Marjoram	SERN	52% (±4.90)	72% (±10.20)	88% (±6.32)	75% (±11.66)	**75%** (±6.32)	**88%**	44%
	ϵ-Greedy	72% (±4.90)	55% (±4.90)	61% (±9.80)	76% (±10.20)	70% (±6.32)	76%	−3%
	Greedy	48% (±12.00)	60% (±6.32)	64% (±7.48)	68% (±10.20)	68% (±8.00)	68%	42%
SaltCylinderSmall	SERN	68% (±10.20)	60% (±10.95)	56% (±9.80)	56% (±11.66)	**76%** (±11.66)	**76%**	12%
	ϵ-Greedy	64% (±11.66)	52% (±12.00)	60% (±8.94)	60% (±8.94)	56% (±4.00)	64%	−13%
	Greedy	64% (±11.66)	52% (±10.20)	48% (±4.90)	40% (±6.32)	48% (±10.20)	64%	−25%
BathDetergent	SERN	36% (±13.27)	48% (±13.56)	48% (±8.00)	60% (±6.32)	**64%** (±7.48)	**64%**	78%
	ϵ-Greedy	60% (±16.73)	52% (±13.56)	48% (±12.00)	60% (±18.97)	52% (±16.25)	60%	−13%
	Greedy	36% (±11.66)	40% (±6.32)	36% (±9.80)	60% (±6.32)	20% (±6.32)	60%	−44%
T-Less 10	SERN	56% (±7.48)	74% (±10.20)	52% (±14.14)	71% (±12.00)	**81%** (±7.48)	**81%**	45%
	ϵ-Greedy	52% (±4.90)	50% (±9.80)	61% (±12.00)	71% (±8.00)	73% (±13.56)	73%	40%
	Greedy	36% (±7.48)	50% (±7.48)	49% (±7.48)	55% (±9.80)	69% (±16.00)	69%	92%
T-Less 20	SERN	72% (±8.00)	60% (±10.95)	60% (±6.32)	72% (±10.20)	**84%** (±4.00)	**84%**	17%
	ϵ-Greedy	72% (±8.00)	60% (±6.32)	44% (±4.00)	68% (±4.90)	68% (±4.90)	72%	−6%
	Greedy	68% (±10.20)	76% (±4.00)	60% (±6.32)	64% (±9.80)	68% (±13.56)	76%	0%

TABLE I: Grasp success rate in simulation. The number in the parenthesis indicates a standard deviation. Obj. indicates a name of 3D mesh in KIT and T-Less dataset. Max. is the maximum success rate achieved during five trials. Imprv. is a performance improvement after training compared to the first performance, which is computed as $(r_5 - r_1)/r_1$ where r_i is the ith success rate. The best performances are marked in bold.

ϵ to be 0.1. Finally, we compare the SERN with $\alpha = 0.05$. ϵ and α are selected by the brute force search.

Furthermore, we employ the grasp quality network and training dataset of Mahler et al. [2] as a pretrained model θ_0 and pretrained data $\mathcal{D}_0$, respectively. We samples $k = 172$ grasping examples from data in $\mathcal{D}_0$. For each round, we generate 64 grasp candidates and corresponding qualities from a given depth image using the pretrained model. We sample one grasp among 64 grasps by applying three sampling methods: greedy, ϵ-greedy, and SERN. By doing so, we can fairly verify effects of the sampling methods since all methods share the pretrained network and only differences are the exploration method. Each round consists of a exploration and evaluation phase. In exploration phase, we collect 20 grasp, image, and result pairs and update the grasp quality network with gathered data. In evaluation, we run the updated network 5 times by selecting the best grasp to verify the actual performance without an effect of exploration.

For the dynamic simulation, a GAZEBO simulator [26] is used with an open dynamics engine [27]. We utilize 3D mesh dataset from KIT [24] and T-Less [25]. As shown in Fig. 1(a) and 1(b), we select four mesh models from [24] and two mesh models from [25], respectively, which are hardly grasped by the pretrained model [2] in simulations. All algorithms run with five random seeds.

In the the real-world experiment, we compare two methods: ϵ-greedy and the proposed method. We also conduct both exploration and evaluation steps separately. In exploration, we gather 5 grasp examples and, in evaluation, we measure 5 grasp tests. We select three objects: triangle, round stapler, and vertical mouse, which are hardly grasped by a parallel jaw gripper due to its rounded surface and nonparallel shape as shown in Fig. 1(c). We use a Baxter robot which has a 7 DoF manipulator to grasp the objects and a RealSense D435 depth camera.

B. Simulation Results

We measure the improvement of grasp success rate between the first and last performance, and the maximum grasp success rate among success rates of five rounds. The results are shown in Table I.

The greedy policy without exploration shows the worst performance in terms of the maximum performance and final performance. From this observation, it is shown that exploration is essential for learning to grasp. Since the greedy policy tries similar grasps when gathering 20 exploratory grasp examples, it cannot gather diverse training data, which causes the over-fitting issue. On the contrary, the SERN and ϵ-greedy method show better performance than the greedy method. In particular, the proposed method, SERN, outperforms other methods in terms of both maximum grasp success rate and final grasp success rate. In all cases, after training for five rounds, the grasp success rates of SERN are improved compared to initial performances. In particular, the success rate for the CokePlastic increase by 212%.

Obj.	Alg.	Rnd 1.	Rnd 2.	Rnd 3.	Rnd 4.	Rnd 5.
Triangle	SERN	0%	20%	20%	60%	**80%**
	ϵ-Greedy	0%	20%	0%	40%	40%
Stapler	SERN	0%	0%	40%	80%	**80%**
	ϵ-Greedy	0%	0%	40%	0%	40%
Mouse	SERN	60%	60%	80%	80%	**80%**
	ϵ-Greedy	0%	20%	20%	20%	20%

TABLE II: Grasp success rate in the real-world experiments. Rnd i indicates the ith round. The best performances are marked in bold.

While ϵ-greedy method outperforms the greedy method in three objects as shown in Table I, it shows poorer performance than the SERN. Since the SERN samples a grasp based on a softmax distribution of estimated grasp qualities, potentially feasible grasp poses are first searched. The ϵ-greedy method, however, explores all grasp poses randomly and it causes inefficiency of exploration in practice. Since we employ the pretrained network, sampling a grasp pose based on the grasp quality estimation from the pretrained network shows better performance while ϵ-greedy method . Thus, the SERN shows the best performance compared to greedy and ϵ-greedy method.

C. the real-world Results

The results are shown in Table II. In the real-world experiments, the SERN outperforms the ϵ-greedy method. In particular, for the Mouse object, the SERN finds success grasps much faster than ϵ-greedy so that it achieves 60% success rate at the first round and outperforms 80%. These results support the fact that using softmax distribution has benefits over the ϵ-greedy method since it frequently explores

the grasp poses that has the high chance of success. Furthermore, the softmax distribution is more suitable than the ϵ-greedy method to employ the pretrained model. From this reason, the SERN generally shows the better performance than the ϵ-greedy method.

The main benefit of SERN compared to ϵ-greedy is the exploration tendency. In ϵ-greedy, the exploration is conducted by an uniform distribution. Thus, ϵ-greedy tries random grasps with ϵ ratio. On the contrast, SERN combine both exploitation and exploration since the greedy action has the largest probability mass and the other actions have the probability mass proportional to its grasp quality.

However, SERN tries more promising grasps which have the potential to successfully grasp the object. In SERN, $\mathbb{P}(a_t = a)$ is proportional to $\exp\left(\hat{Q}_a\right)$ where $\hat{Q}_a$ is a grasp quality of the grasp a. Thus, we can conclude that exploration with soft max distribution has the benefit in practice.

VII. Conclusion

In this paper, we have proposed a novel Shannon entropy regularized neural contextual bandit online learning (SERN) and have applied SERN to learn to grasp unknown objects. We also proved that SERN has no regret properties and its error converges to zero. In both simulation and the real-world experiments, we empirically show that SERN outperforms a ϵ-greedy method and improves the grasp performance efficiently.

References

[1] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," *CoRR*, vol. abs/1905.10520, 2019. [Online]. Available: http://arxiv.org/abs/1905.10520

[2] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[3] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1957–1964.

[4] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 515–524.

[5] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, 2017, pp. 334–343.

[6] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel, "Domain randomization and generative models for robotic grasping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*, 2018, pp. 3482–3489.

[7] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauzá, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. A. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, 2018, pp. 1–8.

[8] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, 2018, pp. 4243–4250.

[9] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[10] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 2018, pp. 9112–9122.

[11] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, 2018, pp. 1–8.

[12] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, 2018, pp. 651–673.

[13] L. Berscheid, T. Rühr, and T. Kröger, "Improving data efficiency of self-supervised learning for robotic grasping," in *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, 2019, pp. 2125–2131.

[14] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *57th IEEE Conference on Decision and Control, CDC 2018, Miami, FL, USA, December 17-19, 2018*, 2018, pp. 6059–6066.

[15] C. Zimmer, M. Meister, and D. Nguyen-Tuong, "Safe active learning for time-series modeling with gaussian processes," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 2018, pp. 2735–2744.

[16] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *CoRR*, vol. abs/1806.10293, 2018. [Online]. Available: http://arxiv.org/abs/1806.10293

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 77–85.

[18] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, 2019, pp. 3629–3635.

[19] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[20] A. Kulesza, B. Taskar, *et al.*, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012.

[21] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," in *Proceedings of the Fourth Annual Workshop on Computational Learning Theory, COLT 1991, Santa Cruz, California, USA, August 5-7, 1991*, 1991, pp. 243–249.

[22] T. Suzuki, "Fast generalization error bound of deep learning from a kernel perspective," in *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 2018, pp. 1397–1406.

[23] K. Lee, J. Choy, Y. Choi, H. Kee, and S. Oh. No-regret shannon entropy regularized neural contextual bandit online learning for robotic grasping: Supplementary material. [Online]. Available: http://rllab.snu.ac.kr/publications/papers/2020_icra_sern_supp.pdf

[24] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, 2012.

[25] T. Hodan, P. Haluza, S. Obdrzálek, J. Matas, M. I. A. Lourakis, and X. Zabulis, "T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects," in *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, 2017, pp. 880–888.

[26] N. Koenig and J. Hsu, "The many faces of simulation: Use cases for a general purpose simulator," in *International Conference on Robotics and Automation, ICRA 2013*, vol. 13, 2013, pp. 10–11.

[27] R. Smith *et al.*, *Open dynamics engine*, 2005.