2024 24th International Conference on Control, Automation and Systems (ICCAS)
Jeju Shinhwa World, Jeju, Korea. Oct 29 - Nov 1, 2024

Effective Exploration via Tsallis Actor-Critic on 6D Robot Grasping

Jaeyeon Jeong[1,2], Jonghyun Park[3], and Songhwai Oh[1,2*]

[1]Department of Electrical and Computer Engineering, Seoul National University,
Seoul, 08826, Korea
[2] ASRI, Seoul National University,
Seoul, 08826, Korea
[3] Department of Naval Architecture and Ocean Engineering, Seoul National University,
Seoul, 08826, Korea
(jaeyeon.jeong@rllab.snu.ac.kr, whdgus1054@snu.ac.kr, songhwai@snu.ac.kr) * Corresponding author

Abstract: Grasping arbitrary objects is challenging due to the need to understand unseen object shapes, generate plausible grasp poses, and plan trajectories simultaneously. In this paper, we propose Goal-Auxiliary Tsallis Actor-Critic (GA-TAC) method, which enhances grasping policy learning by integrating goal prediction auxiliary tasks with Tsallis Actor-Critic [1] methods. Building on the Goal-Auxiliary Deep Deterministic Policy Gradient (GA-DDPG) [2], GA-TAC addresses the limitations of suboptimal grasp policies by dynamically adjusting the exploration-exploitation trade-off using the Tsallis Actor-Critic method. This hybrid approach leverages the strengths of both learning paradigms and adaptive exploration, resulting in more robust and optimal grasping policies. We demonstrate that GA-TAC significantly increases the success rate of grasping tasks by ensuring effective exploration near critical grasping points in Dex-YCB dataset [3] with different q-indices. We also show that GA-TAC outperforms other baselines in the Handover-Sim [4] environment and our method is robust to unseen environments.

Keywords: Robotic Grasping, Robot Application, Human Robot Interaction

1. INTRODUCTION

Grasping arbitrary objects has been a challenging task since it requires the understanding of the shape of unseen objects, generation of plausible grasp poses, and trajectory planning for the grasp poses at the same time. Recent studies have focused on the top-down grasping of arbitrary objects, either by generating grasp poses [5–8], or by learning end-to-end policies via reinforcement learning [9, 10]. However, these settings are not realistic in cases where the robot should generate a trajectory that reaches an object and then grasps the object.

Other previous work focus on generating 6D poses from the object. The actual robot trajectory that reaches the object is then generated via a motion planner [11–15]. This could be problematic in real-world settings where open-loop planning methods could be insufficient due to the dynamic and unpredictable nature of real-world environments, requiring more robust, adaptive strategies to handle unforeseen changes and obstacles effectively.

Goal-Auxiliary Deep Deterministic Policy Gradient (GA-DDPG) [2] has tackled the issue by combining imitation learning (IL) and reinforcement learning (RL), and introducing the goal prediction auxiliary task to generate a plausible grasping pose of the robot. By using Deep Deterministic Policy Gradient (DDPG) algorithm [16], the robot can successfully utilize off-policy data and improve the policy. However, since this method uses DDPG for policy learning, it does not guarantee active exploration near the grasping point and could generate suboptimal grasp policies.

Tsallis Actor-Critic [1, 17] introduces a unified framework for the RL problem and maximum entropy RL problem with various types of entropy. By alternating through various entropic indices, either Shannon-Gibbs entropy-based policy with more exploration and numerical stability, or sparse Tsallis entropy-based policy with more greedy behaviors could be learned, and by balancing between active exploration and greedy policy, the agent can learn the effective exploration behavior that maximizes the reward.

We build upon the two papers, GA-DDPG [2] and Tsallis Actor-Critic [1], to generate a grasping policy, Goal-Auxiliary Tsallis Actor-Critic (GA-TAC), that effectively explores near the grasping point. By integrating the goal prediction auxiliary task from GA-DDPG, which provides a plausible grasping pose, with the Tsallis Actor-Critic method, we aim to balance exploration and exploitation. Specifically, we incorporate the Tsallis entropy framework to adjust the exploration-exploitation trade-off dynamically, fostering more effective exploration near the critical grasping points. This hybrid approach allows the robot to leverage the strengths of imitation learning, reinforcement learning, and adaptive exploration strategies, thereby generating more robust and optimal grasping policies. Our proposed method not only ensures improved grasp success rates but also enhances the overall efficiency and stability of the learning process.

Overall, our contributions are as follows:

1. We propose a Goal-Auxiliary Tsallis Actor-Critic (GA-TAC) method, which integrates goal prediction auxiliary tasks with Tsallis entropy-based exploration strategies to enhance grasping policy learning.

2. We show that the proposed method increases the success rate of grasping objects in the Dex-YCB dataset [3] by ensuring the robot explores critical areas more ef-

fectively and learns more optimal grasping strategies.

3. We further compare the methods on Handover-Sim [4] environment and show that GA-TAC outperforms the baselines.

2. RELATED WORK

2.1 Robot Grasping

Previous studies on robot grasping either focused on grasp generation or end-to-end policy methods. Methods with grasp synthesis [7,11,19,20] first generate robot grasp for the objects and then integrate additional motion planners for robot action generation. On the other hand, end-to-end learning methods [9,21,22] train vision-based grasping policies from a large-scale dataset.

Observation for vision-based grasping policies include depth and segmentation masks [7], keypoints [23], and point clouds [24]. In this paper, we use point cloud observations to effectively process the shape of the objects.

3. BACKGROUND

3.1 Goal-Auxiliary DDPG

Our method builds upon a powerful grasping policy algorithm, GA-DDPG [2]. GA-DDPG is a grasping policy, $\pi : s_t \rightarrow a_t$, that maps state s_t at timestep t to a 6D action (3D translation, 3D rotation) a_t of an end effector. States are 3D point cloud inputs, and PointNet++ [18] is used for feature extraction. From the expert dataset collected by OMG-Planner [25], the BC loss L_{BC} is defined as a point matching loss [26],

$$L_{BC}(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{|X_g|}\|\mathcal{T}_1(x) - \mathcal{T}_2(x)\|_1, \tag{1}$$

where $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{SE}(3)$, and X_g is a predefined grasps set on the gripper.

The final actor loss and critic loss for the GA-DDPG agent is then defined as

$$\begin{aligned} L_\phi &= \frac{1}{2}(Q_\phi(s,a) - y)^2 + L_{AUX}(g, g_\phi) \\ L_\theta &= \lambda L_{BC}(a^*, a_\theta) + (1-\lambda)L_{DDPG}(s, a_\theta) \\ &\quad + L_{AUX}(g, g_\theta) \\ y &= r + \gamma Q_{\phi'}(s', \pi_{\theta'}(s') + \epsilon), \end{aligned} \tag{2}$$

where Q'_ϕ and π'_θ are the target networks, ϵ is a predefined clipped noise, a_θ and g_θ are action and goal from the actor, and λ is a hyperparameter that weighs L_{BC} and $L_{DDPG} = -Q_\phi(s, a_\theta)$.

3.2 Tsallis Entropy

The q-exponential and q-logarithm are used to define the Tsallis entropy. The equation for q-exponential and q-logarithm are as follows [27]:

$$\begin{aligned} \exp_q(x) &\triangleq \begin{cases} \exp(x) & \text{if } q = 1 \\ [1 + (q-1)x]_+^{\frac{1}{q-1}} & \text{if } q \neq 1, \end{cases} \\ \ln_q(x) &\triangleq \begin{cases} \log(x) & \text{if } q = 1 \text{ and } x > 0 \\ \frac{x^{q-1}-1}{q-1} & \text{if } q \neq 1 \text{ and } x > 0 \end{cases} \end{aligned} \tag{3}$$

Then, the Tsallis entropy of a random variable is defined as follows [27]:

$$S_q(P) \triangleq \mathop{\mathbb{E}}_{X\sim P}[-\ln_q(P(X))] \tag{4}$$

, where q is an entropic-index.

By changing the entropic-index, we can represent various types of entropy with Tsallis entropy. For example, when $q \rightarrow 1$, $S_q(P)$ becomes the Shannon-Gibbs entropy, and when $q = 2$, $S_q(P)$ becomes the sparse Tsallis entropy.

3.3 Tsallis Actor-Critic

Tsallis MDPs is an approach to maximum entropy RL that generalizes maximum entropy reinforcement learning with various types of entropy [1]. By controlling the entropic index, various types of entropy, from soft MDPs to sparse MDPs [17], can be generated. Here, the Tsallis entropy of a policy distribution π is defined as $S_q^\infty(\pi) \triangleq \mathbb{E}_{\tau\sim P,\pi}[\sum_{t=0}^\infty \gamma^t S_q(\pi(\cdot|s_t))]$.

With this Tsallis entropy, the objective of Tsallis Actor-Critic (TAC) is defined as

$$\underset{\pi}{\text{maximize}} \mathop{\mathbb{E}}_{\rho,\pi,\mathcal{P}}[\sum_{t=0}^{\infty}\gamma^t(R(s_t,a_t,s_{t+1}) + \beta S_q(\pi(\cdot|s_t)], \tag{5}$$

where β is an entropy weight.

The value function and Q-function for TAC are redefined as follows:

$$\begin{aligned} V_q^\pi(s) &:= \mathop{\mathbb{E}}_{\pi,\mathcal{P}}[\sum_{t=0}^{\infty}\gamma^t(r_t + \beta S_q(\pi(\cdot|s_t)))|s_0 = s] \\ Q_q^\pi(s,a) &:= \mathbb{E}[R(s,a,s') + \gamma V_q^\pi(s')|s' \sim \mathcal{P}(\cdot|s,a)] \end{aligned} \tag{6}$$

4. METHOD

4.1 Task Setting

We focus on the task of grasping an arbitrary object in a closed-loop setting. Our goal is to learn a 6D grasping policy $\pi : s_t \rightarrow a_t$, where s_t is a 3D point cloud state at timestep t, and a_t is a 6D action of an end effector of the robot.

4.2 Goal-Auxiliary Tsallis Actor-Critic

We propose a Goal-Auxiliary Tsallis Actor-Critic (GA-TAC) agent for 6D grasping of arbitrary objects. By introducing an extra Tsallis Entropy term to the original DDPG objective, we can balance between the exploration

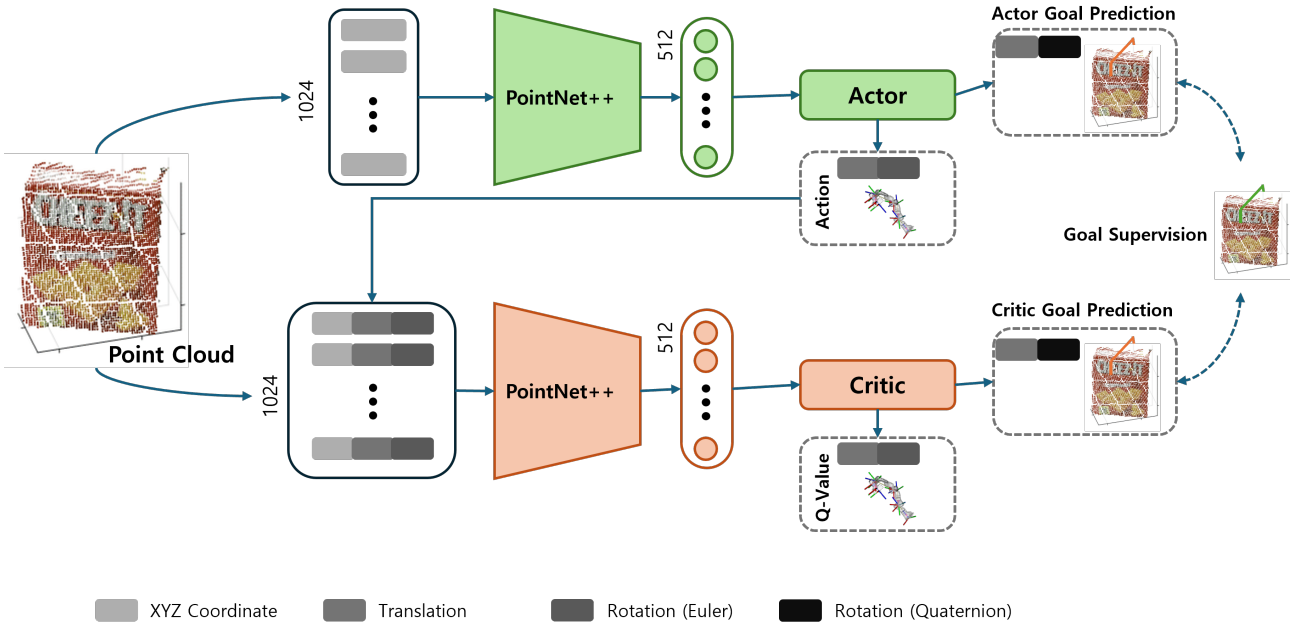

Fig. 1. Overall network architecture used for GA-DDPG [2] and GA-TAC. PointNet++ [18] is used for feature extraction and the auxiliary goal prediction losses are used for both the actor network and the critic network.

Table 1. The result of baselines and GA-TAC with various q-indices on the Dex-YCB dataset [3]. The results with * were reproduced using the code available on the official code repository. For DDPG, the agent was re-trained without pretraining phase to compare the effectiveness of exploration. For GA-DDPG and GA-TAC, the results were averaged over 3 seeds and for each seed, the result was tested for 30 episodes for each object.

item	Success Rate (%) ↑						
	BC*	DDPG*	TAC				
			q=1.0	q=1.2	q=1.5	q=1.8	q=2.0
cracker box	53.3	64.4 (± 15.4)	61.1 (± 15.8)	61.1 (± 13.5)	**84.4 (± 10)**	80 (± 10)	70 (± 10)
sugar box	**83.3**	75.6 (± 8.4)	74.4 (± 8.4)	74.4 (± 7.7)	**83.3 (± 6.9)**	78.9 (± 6.9)	80 (± 8.8)
tomato soup can	73.3	67.8 (± 8.4)	76.7 (± 10.7)	76.7 (± 17.6)	74.4 (± 5.1)	**82.2 (± 5.1)**	67.8 (± 7.7)
mustard bottle	**96.7**	87.8 (± 13.5)	91.1 (± 18.6)	91.1 (± 19.2)	82.2 (± 3.3)	**96.7 (± 3.3)**	88.9 (± 3.8)
potted meat can	36.7	43.3 (± 12.0)	46.7 (± 23.3)	46.7 (± 5.8)	**63.3 (± 3.8)**	57.8 (± 3.8)	57.8 (± 13.5)
bleach cleanser	60	84.4 (± 10.7)	81.1 (± 11.7)	81.1 (± 19.2)	**91.1 (± 8.8)**	90 (± 8.8)	**91.1 (± 15.4)**
bowl	86.7	85.6 (± 22.2)	**97.8 (± 5.1)**	**97.8 (± 3.8)**	**97.8 (± 5.8)**	96.7 (± 5.8)	91.1 (± 15.4)
mug	**90**	56.7 (± 14.5)	71.1 (± 13.5)	71.1 (± 7.7)	74.4 (± 6.9)	64.4 (± 6.9)	58.9 (± 13.5)
foam brick	73.3	**80 (± 5.7)**	64.4 (± 12.6)	64.4 (± 22.7)	75.6 (± 11.5)	63.3 (± 11.5)	67.8 (± 10.2)
average	72.6	71.7 (± 8.0)	73.6 (± 9.4)	73.8 (± 5.2)	**80.7 (± 2.3)**	78.9 (± 2.4)	74.2 (± 4.1)

and exploitation of the agent, thus effectively exploring through the environment with high returns.

We first start with the DDPG loss function from Equation (7). From Equation 7, the loss function for the actor and the critic of the agent is as follows:

$$
\begin{aligned}
L_\phi &= \frac{1}{2}(Q_\phi(s,a) - y)^2 + L_{AUX}(g, g_\phi) \\
L_\theta &= \lambda L_{BC}(a^*, a_\theta) + (1-\lambda) L_{DDPG}(s, a_\theta) \\
&\quad + L_{AUX}(g, g_\theta),
\end{aligned}
$$

We introduce a Tsallis Entropy term (4) for additional regularization for exploration of the agent. That is, the loss for the actor and the critic of the agent for GA-TAC is as follows:

$$
\begin{aligned}
L_\phi &= \frac{1}{2}(Q_q^\phi(s,a) - y)^2 + L_{AUX}(g, g_\phi) \\
L_\theta &= \lambda L_{BC}(a^*, a_\theta) + (1-\lambda) L_{TAC}(s, a_\theta) \\
&\quad + L_{AUX}(g, g_\theta) \\
y &= r + \gamma Q_q^{\phi'}(s', \pi_{\theta'}(s') + \epsilon),
\end{aligned}
\tag{7}
$$

where q is an entropic index of Tsallis entropy, and $L_{TAC} = -Q_q^\phi(s, a_\theta)$.

5. EXPERIMENTS

5.1 Training

The overall network architecture for both GA-DDPG [2] and GA-TAC are as in Figure 1. We experiment with

Table 2. The result of baselines and GA-TAC with various q-indices on the Handover-Sim [4] benchmark. The results with * were reproduced using the code available on the official code repository. For DDPG, the agent was re-trained without pretraining phase to compare the effectiveness of exploration. For GA-DDPG and GA-TAC, the results were averaged over 3 seeds and for each seed, the result was tested for 144 episodes for each object.

	DDPG*	TAC				
		q=1.0	q=1.2	q=1.5	q=1.8	q=2.0
Success Rate (%) ↑	15.7 (± 2.9)	9.3 (± 3.4)	13.7 (± 6.3)	**19.4 (± 8.9)**	2.5 (± 2.6)	19.2 (± 6.8)
Mean Acc. Time (s) ↓	7.3 (± 0.2)	7.3 (± 0.2)	7.3 (± 0.2)	7.4 (± 0.0)	**7.2 (± 0.6)**	7.3 (± 0.1)

the Franka Emika Panda arm, which has a 7-DoF arm with a parallel 2-finger gripper. We use ShapeNet [28] for the objects for grasping in the training phase, as in GA-DDPG [2] setting. A task scene is generated with objects in random poses, placed on a tabletop in a PyBullet Simulator [29]. The maximum horizon for the policy is set to 30 timesteps and an episode terminates when either a maximum horizon is reached or an agent successfully grasps the object. The observation is an RGB-D image with size 112×112. Note that we do not pre-train the agent using behavioral cloning to compare the exploration with different q values. The hyperparameters for each example were set to be the same as in GA-DDPG [2], except for the entropic indices in GA-TAC methods. The training was conducted over three random seeds, and for each seed, the training took about 15 hours with 4 Nvidia RTX 3090 GPUs.

5.2 Baselines

We compare the grasping performance on the Dex-YCB [3] benchmark dataset. The compared baseline algorithms are as follows.

- **OMG-Planner [25] + BC** We train a behavioral cloning policy with expert demonstrations collected from Optimization-based Motion and Grasp Planner (OMG Planner) [25]. OMG-Planner plans a robot trajectory to grasp the robot, given a planning scene and a set of pre-defined grasps from the target object [11, 12]. With planned trajectories as expert demonstrations, the behavioral cloning agent utilizes the point-matching loss function [26] to learn the policy.
- **GA-DDPG** [2] We also re-train a GA-DDPG agent from scratch and compare the result with Tsallis Actor-Critic based policies.

5.3 Grasping Performance

We evaluate the effectiveness of adapting Tsallis Entropy to the grasping policy on 9 Dex-YCB [3] objects. To effectively compare the result of exploration on the policy, we train the agents without pre-training, unlike [2]. For GA-DDPG and GA-TAC, the results were averaged over 3 seeds and for each seed, the result was tested for 30 episodes for each object. The results for the grasping performance on YCB objects are as in Table 1. As shown in the table, GA-TAC performed the best in all of the objects except for the mug and the foam brick objects. Also, GA-TAC with $q = 1.5$ showed the highest average performance among all methods, which states that effectively balancing between exploration and exploitation leads to the best performance.

5.4 Human-to-Robot Handover Performance

We further compare the result on Handover-Sim benchmark [4]. As in the original paper [4], we do not further train the agents in the benchmark and use the same agent trained for the ShapeNet [28] objects. In this benchmark, the agent must adeptly receive the object being handed over by the human, ensuring that there is no collision between the agent and the human hand and that the object is not dropped. As in Table 2., we find out that effective exploration finds optimal policies that are robust to unseen environments, even when a new constraint is introduced.

6. CONCLUSION

In this study, we addressed the challenge of grasping arbitrary objects by proposing the Goal-Auxiliary Tsallis Actor-Critic (GA-TAC) method. By integrating goal prediction auxiliary tasks with Tsallis entropy-based exploration strategies, GA-TAC enhances grasping policy learning and addresses the limitations of previous approaches. By adjusting the exploration-exploitation trade-off, the agent learns more effective exploration near critical grasping points and shows the highest performance in the Dex-YCB dataset [3]. We further show that our method is robust to unseen settings, and also shows the best performance compared to the baselines.

Our method could be further improved with more introduction of additional constraints to the problem. Especially in Human-to-Robot handover settings, one could additionally integrate Safe RL method, as in SafeTAC [30], to integrate effective exploration with Safe reinforcement learning that does not violate the collision constraint.

ACKNOWLEDGEMENT

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).

REFERENCES

[1] K. Lee, S. Kim, S. Lim, S. Choi, M. Hong, J. I. Kim, Y.-L. Park, and S. Oh, "Generalized tsallis entropy reinforcement learning and its application to soft mobile robots.," in *Robotics: Science and Systems*, vol. 16, pp. 1–10, 2020.

[2] L. Wang, Y. Xiang, W. Yang, A. Mousavian, and D. Fox, "Goal-auxiliary actor-critic for 6d robotic grasping with point clouds," in *Conference on Robot Learning*, pp. 70–80, PMLR, 2022.

[3] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox, "DexYCB: A benchmark for capturing hand grasping of objects," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] Y.-W. Chao, C. Paxton, Y. Xiang, W. Yang, B. Sundaralingam, T. Chen, A. Murali, M. Cakmak, and D. Fox, "HandoverSim: A simulation framework and benchmark for human-to-robot object handovers," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[5] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1316–1322, IEEE, 2015.

[6] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 3406–3413, IEEE, 2016.

[7] 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," *arXiv preprint arXiv:1703.09312*, 2017.

[8] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[9] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*, pp. 651–673, PMLR, 2018.

[10] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6284–6291, IEEE, 2018.

[11] 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.

[12] C. Eppner, A. Mousavian, and D. Fox, "A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set," in *The International Symposium of Robotics Research*, pp. 890–905, Springer, 2019.

[13] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[14] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3766–3773, IEEE, 2018.

[15] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2901–2910, 2019.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[17] K. Lee, S. Choi, and S. Oh, "Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1466–1473, 2018.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[19] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[20] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6232–6238, IEEE, 2020.

[21] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[22] S. Iqbal, J. Tremblay, A. Campbell, K. Leung, T. To, J. Cheng, E. Leitch, D. McKay, and S. Birchfield, "Toward sim-to-real directional semantic grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7247–7253, IEEE, 2020.

[23] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*, pp. 132–157, Springer,

2019.

[24] X. Yan, M. Khansari, J. Hsu, Y. Gong, Y. Bai, S. Pirk, and H. Lee, "Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks," *arXiv preprint arXiv:1906.08989*, 2019.

[25] L. Wang, Y. Xiang, and D. Fox, "Manipulation trajectory optimization with online grasp synthesis and selection," in *Robotics: Science and Systems (RSS)*, 2020.

[26] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 683–698, 2018.

[27] S.-i. Amari and A. Ohara, "Geometry of q-exponential family of probability distributions," *Entropy*, vol. 13, no. 6, pp. 1170–1185, 2011.

[28] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[29] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[30] D. Kim, J. Heo, and S. Oh, "Safetac: Safe tsallis actor-critic reinforcement learning for safer exploration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4070–4075, IEEE, 2022.