Improving Data Efficiency of
Self-supervised Learning for Robotic Grasping

Lars Berscheid[1], Thomas Rühr[2] and Torsten Kröger[1]

Abstract— **Given the task of learning robotic grasping solely based on a depth camera input and gripper force feedback, we derive a learning algorithm from an applied point of view to significantly reduce the amount of required training data. Major improvements in time and data efficiency are achieved by: Firstly, we exploit the geometric consistency between the undistorted depth images and the task space. Using a relative small, fully-convolutional neural network, we predict grasp and gripper parameters with great advantages in training as well as inference performance. Secondly, motivated by the small random grasp success rate of around 3%, the grasp space was explored in a systematic manner. The final system was learned with $23\,000$ grasp attempts in around $60\,\mathrm{h}$, improving current solutions by an order of magnitude. For typical bin picking scenarios, we measured a grasp success rate of $(96.6 \pm 1.0)\%$. Further experiments showed that the system is able to generalize and transfer knowledge to novel objects and environments.**

I. INTRODUCTION

Grasping is an essential task in today's robotics, but even more so in future applications like large-scale warehouse automation or service robotics. In particular, bin picking is the task of grasping objects out of unsystematic environments like a randomly filled bin. In comparison to general robotic grasping, bin picking emphasizes some difficulties as partially hidden objects and an obstacle-rich environment. The state-of-the-art industrial solution requires the 3D model of the grasped object. It is localized in a segmented point cloud of the bin, its pose is estimated and grasped at taught coordinates [1]. However, this method does neither work with unknown objects, nor does the manual effort of creating 3D models and hand-teaching grasping coordinates scale to many object types.

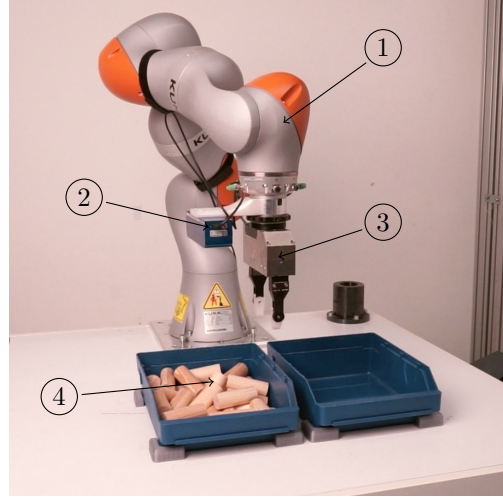Self-learning methods could reduce this manual input to a minimum. As those data-driven methods have the

Fig. 1: Our setup consists of a KUKA LBR iiwa robot (1) with a flange-mounted stereo camera (2), a force-feedback gripper (3), and two industrial storage bins (4).

intrinsic capability to generalize, a universal grasping controller for many objects and scenarios appears possible. Despite recent innovations in robot learning, the progress of self-learning methods is largely based on the consumption of ever-growing datasets. To reverse this costly trend, we emphasize an applied point of view for learning robotic grasping. We therefore consider the time and data efficiency as the major challenge. We restrict ourselves to a simple setup (Fig. 1) with a single robot, typical industrial storage bins and a time frame of a weekend ($60\,\mathrm{h}$). Given a depth camera and a force-feedback gripper, the system must be able to learn without human supervision.

Our key contribution is a data-efficient approach for learning robotic grasping. In the following, we will derive our algorithm and simplify where necessary based on the framework of reinforcement learning (RL). The robot trains a neural network (NN) while interacting with the objects using active learning. Therefore, we define a set of rules for exploring the grasp space efficiently. We propose the use of weighted retraining, a procedure for reducing the impact of measurement errors. Then, we present the overall training procedure and implementation. Finally, we conduct real robot experiments, a quantitative grasp rate evaluation and qualitative tests for generalization to unknown objects.

[1]Intelligent Process Automation and Robotics Lab (IPR), Karlsruhe Institute of Technology (KIT) {lars.berscheid, torsten}@kit.edu

[2]KUKA Deutschland GmbH, thomas.ruehr@kuka.com

II. RELATED WORK

In recent years, research of robotic grasping in unsystematic environments gained a lot of traction. Bohg et al. [2] divided possible approaches into the grasping of known, familiar or unknown objects. For known objects, grasping is mostly reduced to object recognition and pose estimation. In a more flexible manner, this process can be extended by matching the similarity of known objects to new, familiar objects. In the case of unknown objects, the approaches can be split into analytical and empirical methods [3]. Analytical grasp synthesis relies on the definition of a grasp quality measure, such as the force closure or other geometrically motivated measures. However, these approaches usually need high-quality and flawless 3D maps of the scene. Data-driven methods intrinsically develop generalization and error robustness, but suffer from the demand of large datasets. First implementations sampled grasp candidates in a simulator and evaluated them based on an analytical measure [4]. In 2008, Saxena et al. [5] applied a probabilistic model to a mostly simulated dataset to identify grasping points in image patches. More recently, Mahler et al. [6] built a large synthetic grasping database with analytic metrics, used supervised learning to train a NN and applied it to bin picking [7].

Alternatively to simulation, the system can be automated and trained in a large-scale self-supervised manner. Our work is closely related to the research of Pinto and Gupta [8]. They were able to retrain a NN within 700 h and achieved a grasping rate for seen objects of 73 %. Levine et al. [9] scaled up the data collection to 800 000 grasp attempts on 14 robots. They trained in an end-to-end manner, including the camera calibration, spatial relationships and time-dependency.

Our work is based on the theoretical framework of RL in combination with convolutional NNs for function approximation [10]. While deep RL showed impressive results in learning from visual input in simple simulated environments [11], [12], direct applications of RL for robot learning have proven to be more difficult. On real robots, continuous policy-based methods have been applied to visuomotor learning, either in an end-to-end fashion [13] or by using spatial autoencoders [14]. Quillen et al. [15] showed in a recent comparison of RL algorithms for simulated grasping that simple value-based approaches performed best.

III. LEARNING FOR GRASPING

Reinforcement learning (RL) is a powerful framework to formulate robot learning problems in an abstract way. We define a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, T, r, p_0)$ with the state space $\mathcal{S}$, the action space $\mathcal{A}$, the transition distribution T, the reward function r and the initial configuration p_0. The robot chooses an action a_t at every discrete time step t, maximizing the expected collected reward $\mathbb{E}\left[\sum_t r_t\right]$ over all following time steps. A solution of a RL problem is a stochastic

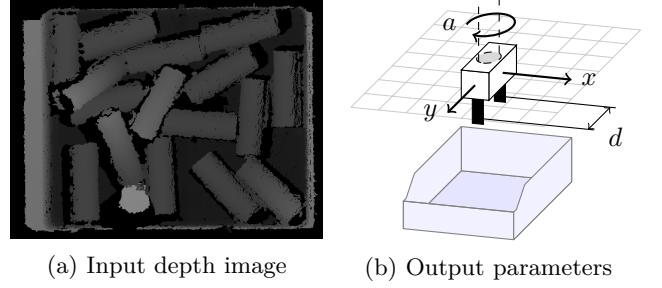(a) Input depth image   (b) Output parameters

Fig. 2: Our algorithm maps an undistorted depth image (a) to four parameters (x, y, a, d) in the gripper-task space (b). The image ranges from near (white) to far (black), missing depth information are displayed black. The image plane is aligned with the x-y-plane of the task space, which is a crucial requirement for our algorithm.

policy $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$, mapping a state s to an action probability distribution $\hat{a}$. A major challenge of RL is the possible time-delay between a crucial action a_t and the reward $r_{t+\Delta t}$. To reduce this complexity, we describe a grasp as a single action furthermore. A complete grasp attempt is therefore formulated as sensing a state s, planning an action a and receiving a reward r in an open-loop fashion.

A. Single-Shot Grasps

The grasping pose is defined as the position (x, y, z) and Euler angles (a, b, c) (with a around the z-axis) where the gripper is closed. Given a calculated grasping pose, the natural process of a single-shot grasp attempt is:

1) The gripper jaw distance is set to d.
2) The gripper approaches the grasping pose with a movement parallel to its fingers. If the joint torque sensors detect a collision, the gripper retracts a few mm and continues the grasping process.
3) The gripper closes and tries to clamp an object with a predefined force f.
4) The robot lifts the object and moves to a filing position. The reward r is set to 1 for a successful grasp, otherwise to 0.

We limit ourselves to planar grasps by setting $b = c = 0$. The height z is calculated trivially by reading the corresponding value at (x, y) from the depth image. Eventually, the policy needs to calculate four parameters (x, y, a, d) from the depth image (Fig. 2).

B. Geometric Consistency

A key contribution of this work is to utilize the geometric consistency between the x and y parameter in the task space and the aligned, undistorted depth image. This is made possible by the invariance of the depth image under planar translations of the robotic camera system. However, this demands that the largest parts of the policy computation is translational invariant itself. Fortunately, in RL, the policy is often split into two

functions $\pi = \sigma \circ \psi$. In our case, ψ, called the implicit policy, is a standard value function, such as the state or action value. Then, σ selects the final action distribution $\hat{a}$. For greedy policies in the exploitation phase, σ is usually the max-function. For exploration, a random term ε is often added and decreased with experience. A data-efficient exploration is discussed in section III-D.

In our work, the value function ψ predicts the grasp success probability at the center of a sliding window. In RL terms, this corresponds to the state-value of a substate $s' \subset s$. Although the given reward r is binary, predicting the grasping probability is not a classification but a regression task. The grasping success is intrinsically probabilistic, mostly because the MDP is in reality only partially observable. Due to the sliding window, the grasp space $\mathcal{A}$ is discretized.

C. Neural Network

The state-value function $\psi(s')$, predicting the grasping probability from a depth image subwindow, is approximated by a NN. We implement this sliding-window approach as a fully-convolutional NN, enabling a variable input and corresponding output size, as proposed by Long et al. [16]. This increases the inference performance by an order of magnitude. The gripper distance d is integrated as a depth dimension. The NN consists of five layers with around $450\,000$ parameters (Fig. 3a). If not otherwise stated, all kernels have single stride and $L_2 = 0.3$. ReLU is used as the activation function. Batch normalization (BN) and dropout with a minimum rate of 0.4 are applied to most layers. In comparison to image classification, grasping requires exact spatial information, which would get lost in max-pooling layers. Furthermore, the subwindow images are generated from the overview image using the affine transformation $\mathbf{R}_a \mathbf{T}_{xy}$ applying the translation (x, y) and rotation (a) and subsequently cropping around the center. This approach is made possible by the precise intrinsic calibration of the camera as well as the position accuracy of the LBR iiwa, both below a millimeter. Let $\psi(s, a)$ denote the combination of the affine transformation and the state-value function $\psi(s')$. For training, the loss is defined as the cross entropy of the reward at the measured gripper distance.

During inference, the NN scales with the size of the overview depth image (Fig. 3b). The angle a is calculated by rotating the input image first, otherwise the kernel sizes would need to be increased accordingly. The grasp space has $|\mathcal{A}| = 96\,000$ discrete elements, corresponding to 20 rotations a, 40 x- and y positions and 3 gripper distances d. This leads to a resolution of approximately $4\,\text{mm}$ in the x-y-plane, $20\,\text{mm}$ for the gripper distance d and less than $9°$ for the rotation a. Calculating a grasp takes around $12\,\text{ms}$ on a single NVIDIA GeForce GTX 1070, making embedded solutions feasible.
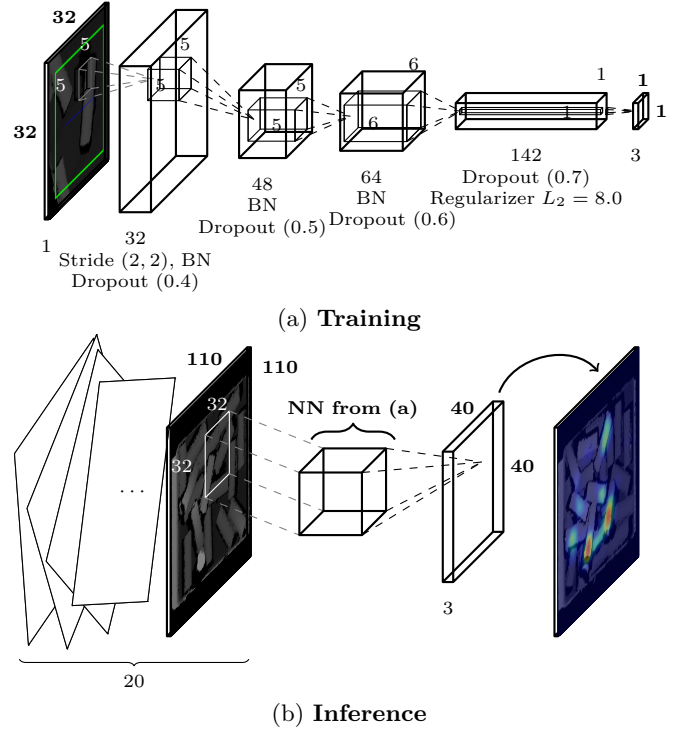

(a) **Training**

(b) **Inference**

Fig. 3: (a) Our neural network (NN) architecture. The training output has a size of $(1 \times 1 \times 3)$, corresponding to three gripper distances d. (b) During inference, the NN calculates from the overview image an output of size $(40 \times 40 \times 3)$, corresponding to the (x, y, d) parameters. For the rotation a, the input image is pre-transformed and the NN is recalculated for 20 angles. A single output can be interpreted as a grasp probability heatmap.

D. Grasp Space Exploration

Random grasps result in a success rate of only 3%. This motivates our systematic approach for a data-efficient self-supervised exploration. Therefore, learned information is used to improve learning itself. As this RL problem is simplified towards a supervised learning problem, the task of exploration resembles active learning. First of all, images of successful grasps contain more valuable information, as they are harder to collect. Secondary, a grasp attempt is valuable for learning if the predicted grasp probability differs from the measured reward, leading to grasps a given by $\arg\max_a |r - \psi(s, a)|$. To fulfill both conditions, exploration is implemented by choosing one of the following selection functions σ:

a) Random: A uniformly random selection function can be given as $\sigma = \hat{a} \sim P(a) = |\mathcal{A}|^{-1}$. Only this method is guaranteed to fully span the grasp space $\mathcal{A}$. Random grasp rates were measured using this method.

b) Maximum(N): A more general form of the greedy action $\sigma(\psi) = \arg\max_a \psi(s, a)$ is to choose one of the N highest-ranked poses randomly. In application, this method with $N = 1$ is first applied. If the grasp fails, a grasp attempt with $N = 5$ is repeated. This method is
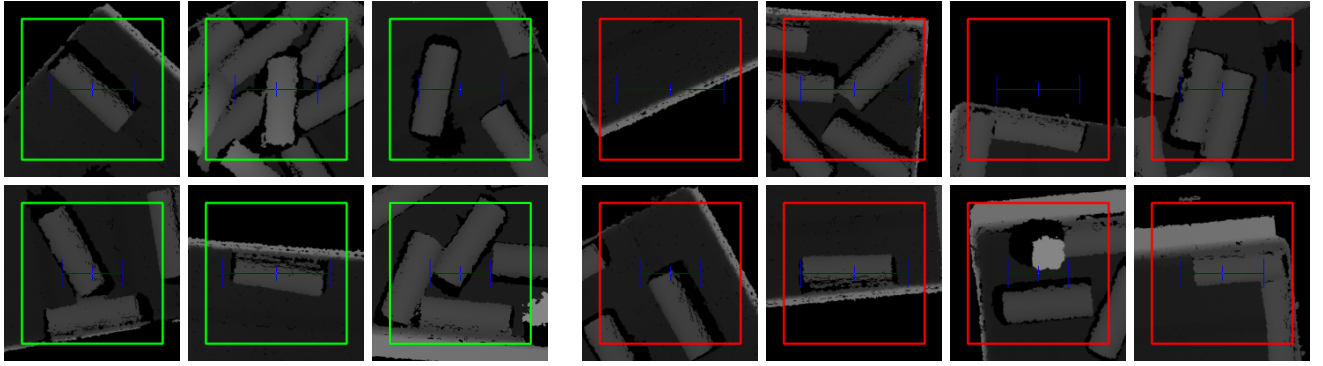
Fig. 4: Examples of cropped window depth images. Successful (left, green) and failed (right, red) grasps as well as the gripper position (blue) are marked.

used for exploitation.

    c) Probabilistic: By choosing a grasp according to the given and normalized probability $\sigma(\psi) = \hat{a} \sim \psi$, the overall measure of probability improves. This method increases the grasping rate but keeps exploration active and is used largely during training.

    d) Uncertain: The selection function for uncertain grasps with the nearest probability to 0.5 is given by $\sigma(\psi) = \arg\min_a |0.5 - \psi(s,a)|$. Those grasps contain valuable information, because the difference between the grasp probability and the measured reward is near 0.5.

E. Weighted Retraining

    Weighted retraining aims to minimize the impact of two error types. Firstly errors from defective measurements, e.g. caused by moving objects, and secondly the sampling error known in RL. In the context of a MDP, the grasping success is always probabilistic. This is in contrast to the binary value of r and reinforces the sampling error. Let w_i be the weight of a grasp attempt during training and N be the total dataset size. The NN is at first trained with $w_i = 1$. Then, the NN predicts the reward ψ and adapts the weights according to

$$w_i = \frac{N \left(1 - |r_i - \psi(s_i, a_i)|\right)}{\sum_j \left(1 - |r_j - \psi(s_j, a_j)|\right)}.$$

The normalized difference between measured and predicted reward is used as a quality measure of the input data, and the NN is retrained with a weighted training set.

F. Training Procedure

    For training the NN, two effects need to be considered: Firstly, an average reward of 0.5 for the overall dataset is preferred, otherwise the NN could overfit to the specific distribution. Secondly, both grasping error types are asymmetric. A false negative error is a missed chance for grasping, while there are typically a multitude of positive grasping poses. However, false positive errors correspond to real grasping errors, which should be minimized with higher priority. Both effects are implemented as two

successive, reward-depending weights in the calculation of the cross entropy loss.

    To increase the quality of grasps, the clamping force f was reduced during training in comparison to application. This should minimize grasping torques and draw grasps near the center of mass. See [17] for comparison.

    The recorded grasp attempts are split into a training and test set. Even as new measurements are added, elements of the two sets must not be exchanged. This is implemented by using a hash function to map the grasp timestamp to a number $\in [0, 1]$. If the value is less than 0.2, the grasp is assigned to the test set. The exact set sizes are therefore not predetermined.

    Additionally, our system of two GPUs can train the NN as well as infer new grasps for exploration in parallel. Therefore grasp predictions are always up-to-date, improving the data efficiency of the training procedure. The system is able to train completely without human input and supervision.

IV. EXPERIMENTAL RESULTS

    For experiments, a KUKA LBR iiwa 7 kg robot arm combined with a Weiss WSG 50 two-finger gripper was used (Fig. 1). The integrated force-feedback sensor detected if an object was grasped successfully. The flange-mounted Ensenso N10 stereo camera provided 3D data of the scene. If not otherwise stated, wooden cylinders with a diameter of 1.5 cm and a height of 6 cm were used as test objects. A supplementary video of our experiments can be found at `https://youtu.be/0P3FJAo4ZOA`.

A. Data Recording

    The grasped objects are filed into a second bin. This allows grasp attempts from bins with a varying number of objects and continuous operation by switching grasping and filing bin. Typical durations for grasp attempts are around 10 s. The further evaluated NN was trained with a dataset of 23 000 grasp attempts, recorded in 60 h. Fig. 4 shows examples of the depth image subwindow. Complete random grasps were conducted for the first 1000 grasp attempts. During exploration, the random
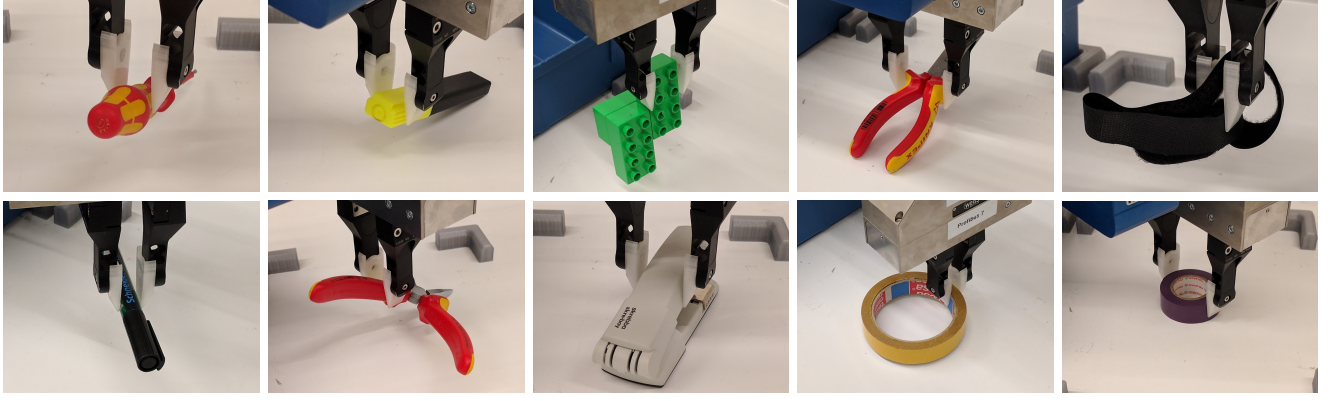
Fig. 5: Our robot generalizes to unseen object types. Although it was trained just with cylinders, grasps of similar (left) to more differing (to the right) objects work reliably.
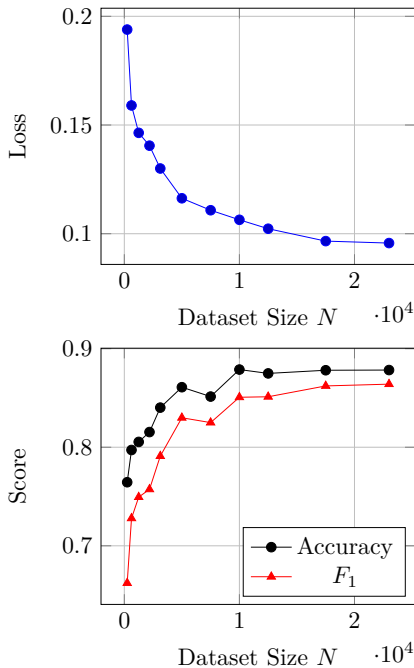

Fig. 6: Loss, accuracy and F_1-score of the test set depending on the dataset size N.

method was used 2440, the probabilistic method 8830, the uncertain method 1300 and the maximum(N=5) method 9430 times. Dependencies of different evaluation metrics of the NN on the dataset size are shown in Fig. 6. Note that this development is influenced heavily by the current exploration strategy, since the underlying data distribution changes on purpose. The training was finished as the loss saturated above 20 000 grasp attempts.

B. Grasp Rate

The grasp rate is measured by counting the attempts of grasping n objects out of a bin with a total of m objects without replacement. Note that this is more influenced by outliers as well as more important for applications than just counting grasped objects at n attempts. For typical bin picking scenarios, a grasp rate of $(96.6 \pm 1.0)\%$ was measured (Table I). Human performance is around 600 picks per hour (PPH) [7], exceeding our robot system by 40%. As expected, the grasp rate improves with decreasing clutter and increasing grasp choices.

TABLE I: The measured grasp rate for grasping n objects without replacement out of a bin initially containing m objects. The random grasp rate is $\approx 3\%$.

n out of m	Grasp Rate	PPH	Number Samples
1 out of 1	$(99.8 \pm 0.2)\%$	367 ± 2	460
5 out of 5	$(99.3 \pm 0.7)\%$	418 ± 8	146
5 out of 10	$(99.5 \pm 0.5)\%$	428 ± 3	181
10 out of 20	$(96.6 \pm 1.0)\%$	425 ± 9	207

Weighted retraining of the NN improves the grasp rate (10 out of 20) by around 1% from $(95.2 \pm 2.8)\%$ to $(96.6 \pm 1.0)\%$. The main effect of weighted retraining is the reinforcement of reliable grasps. As a side effect, this decreases the probability of making the same grasp mistake again, leading to a reduced uncertainty.

C. Generalization

While the robot was only trained with cylinder objects, it is able to generalize to novel object types (Fig. 5). The grasp rate for the shown isolated objects is above 90%. Even objects with holes or a flexible geometry work reliably. For cluttered situations, the grasp rate decreases due to stacked and partially hidden objects. Problems arise with objects larger than the trained cylinders. As larger objects may look like multiple smaller objects, the system attempts a grasp to separate possible objects.

The grasping success depends on the similarity with the trained object type. A typical grasp mistake is a collision with sharp edges of objects, which is probably due to the rounded edges of the trained cylinders. Furthermore, the ability to generalize was evaluated with sharp cubes as the novel object type. Fig. 7 shows the robot's capability of transfer learning between object types. Only by combining the small dataset of cubes with

the previous dataset of cylinders, a comparable grasp rate to Table I is achieved. Therefore, the training of novel object types in applications should rely on existing datasets.

The robotic system is also able to generalize to new environments. Simple height differences have no measurable effect, as well as removing the bin and grasping from table surfaces. Tilted backgrounds have only small effects, as the sliding window only operates on local areas of the overall depth image. Even horizontal shelf picking works reliably as well, although the system was trained while gravity pointed in another direction.
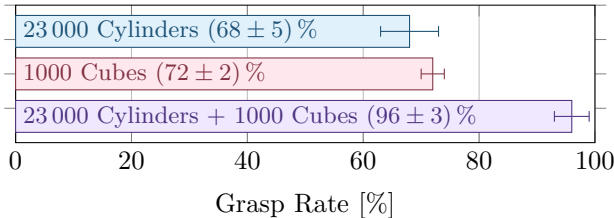

Fig. 7: The grasp rate of cubes (10 out of 15 objects), depending on the trained object type (label) and their corresponding dataset sizes N.

V. DISCUSSION AND OUTLOOK

We developed a self-learning robotic grasping system while keeping more realistic requirements for industrial applications in mind. In this regard, we used a depth camera in contrast to [8], [9], [18]. Our implementation works on a simple experimental setup and improves time and data efficiency significantly [8], [9]. An average grasp lasts less than 10 s, leading to a fivefold reduction compared to [8] with a similar setting. Our algorithm runs at 50 frames per second and outperforms [19], despite their focus on real-time capability. Although our random grasp rate of around 3 % is an order of magnitude lower, the final dataset is only half in size in comparison to [8]. In contrast, we used only one object type in training and testing. For seen objects, [8] measured a grasp rate of 73 %. [9] achieved a grasp rate of 90 % for unseen objects in a otherwise similar setting. However, both setups had significantly higher random grasp rates of up to 33 %. Because we used typical storage bins used in intra-logistic applications, our gripper collided more often during exploration. Consistently, our NN additionally needed to learn a collision model between the gripper and the bin. In opposition to [9], our approach generalizes to unseen environments and can therefore be applied not only to bin picking, but to general planar robotic grasping. [17] suggested shaking the grasped object to improve the stability of learned grasps. As an alternative, we simply reduced the gripper force f during training. This leads to similar results, but faster training and less robotic wear and tear.

Due to the planar simplification, the robot can physically not grasp in the corners of the bin. In future, this can be resolved by adding the two remaining angles b and c. This increases the dimension of the grasp space, so that a trivial extension of this work would suffer from the curse of dimensionality. We suggest to combine our value-based approach for x, y and a with a continuous policy-based method for b and c. As more parameters are introduced, the random grasp rate decreases even further. Two main approaches could solve this problem: Firstly, the grasp space exploration should be investigated in greater detail. Secondly, the difficulty of the robotic grasping tasks can be increased in incremental steps. At a more general level, we believe that the robot is able to transfer knowledge from an easier to a more difficult task. In practice, full grasps could be trained by planar grasps while slowly deviating the angles b and c. Then, a key challenge in robot learning lies in an appropriate task design.

ACKNOWLEDGMENT

REFERENCES

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, ser. Springer Handbooks. Springer International Publishing, 2016, ch. 54. Industrial Robots.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis - A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, Apr. 2014. [Online]. Available: http://arxiv.org/abs/1309.2660

[3] A. Sahbani, S. El-Khoury, and P. Bidaud, "An Overview of 3d Object Grasp Synthesis Algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, Mar. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0921889011001485

[4] 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.

[5] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects Using Vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[6] 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:1703.09312 [cs]*, Mar. 2017. [Online]. Available: http://arxiv.org/abs/1703.09312

[7] J. Mahler and K. Goldberg, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," in *Conference on Robot Learning*, 2017, pp. 515–524.

[8] L. Pinto and A. Gupta, "Supersizing Self-supervision: Learning to Grasp from 50k Tries and 700 Robot Hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3406–3413.

[9] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 173–184.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, nachdr. ed. Cambridge, Mass.: MIT Press, 1998.

[11] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," 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. 262–270. [Online]. Available: http://proceedings.mlr.press/v78/rusu17a.html

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv:1312.5602 [cs]*, Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[13] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," *Journal of Machine Learning Research*, p. 40, 2016.

[14] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep Spatial Autoencoders for Visuomotor Learning," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 512–519.

[15] 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," *arXiv:1802.10264 [cs, stat]*, Feb. 2018, arXiv: 1802.10264. [Online]. Available: http://arxiv.org/abs/1802.10264

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[17] L. Pinto, J. Davidson, and A. Gupta, "Supervision via Competition: Robot Adversaries for Learning Tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1601–1608.

[18] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-End Learning of Semantic Grasping," *arXiv:1707.01932 [cs, stat]*, Jul. 2017. [Online]. Available: http://arxiv.org/abs/1707.01932

[19] J. Redmon and A. Angelova, "Real-time Grasp Detection using Convolutional Neural Networks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1316–1322.