Supplemental File for "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences"

Jeffrey Mahler
EECS Department
UC Berkeley
jmahler@berkeley.edu

Ken Goldberg
EECS and IEOR Department
UC Berkeley
goldberg@berkeley.edu

This document contains an errata for the PMLR version of the paper, additional details of our transfer learning experiments, additional qualitative performance analyses, and details on the parameters of our Partially Observable Markov Decision Process (POMDP) model to accompany the paper 'Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences" appearing at CoRL 2017.

1 Errata for CoRL PMLR Version

There are several minor errors in the CoRL PMLR version of this paper, available at http://proceedings.mlr.press/v78/:

Section 3.2, Page 3: The distributions are technically not independent, as claimed by the paper. For example, the Object Heap depends on the Object Count variable. The distribution is represented as the product of the described distributions.

Section 4.2, Page 5: The correct noise-injected rollout policy for $\epsilon \in [0, 1]$ is:

$$\pi(\mathbf{y}) = \begin{cases} \pi^*(\mathbf{y}) & \text{with prob. } 1-\epsilon \\ \text{Unif}(\mathcal{U}) & \text{with prob. } \epsilon \end{cases}$$

In other words, the rollout policy takes an action from the supervisor with probability $1-\epsilon$ and takes a random action with probability ϵ. The PMLR version erroneously states that the rollout policy took the supervisor's action with probability ϵ.

Section 5.1, Page 6: The dataset sizes were approximately: $\epsilon = 0.1$: 16.5k, $\epsilon = 0.5$: 34.8k, $\epsilon = 0.9$: 102.6k. The difference is because a smaller ϵ results in a larger percentage of actions being taken by the algorithmic supervisor, reducing the number of timesteps per rollout because the supervisor rapidly clears heaps. The PMLR version erroneously claims that all datasets contain approximately 10k datapoints.

2 Additional Experiments

We ran several additional experiments and analyses to evaluate the performance of the Dex-Net 2.1 bin picking policy.

2.1 Synthetic Training

To recap, we trained the following models with a 80-20 image-wise split on each of the $\epsilon = 0.1$, $\epsilon = 0.5$, and $\epsilon = 0.9$ datasets:

- **SVM.** A bagging classifier composed of 50 SVMs trained on the first 100 principal components of the GQ-CNN fc4 feature space [1].
- **Random Forest (RF).** A set of 50 trees of max depth 10 trained on the first 100 principal components of the GQ-CNN fc4 feature space [1].

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

- **Dex-Net 2.1 (Scratch).** A GQ-CNN [1] trained only on Dex-Net 2.1 for 25 epochs.
- **Dex-Net 2.1 (Fine-tuned).** A GQ-CNN [1] initialized with pretrained weights from Dex-Net 2.0 and fine-tuned on Dex-Net 2.1 for 10 epochs with fixed conv layers.

Fig. 1 shows the precision-recall curve for the methods. The GQ-CNN-based policy trained with transfer learning performs significantly better than the other methods with an Average Precision of 64% compared to 59% for the next best method. The SVM used the output of Conv2_2 layer of the Dex-Net 2.0 GQ-CNN as input and the Random Forest used the output of the fc4 layer of the Dex-Net 2.0 GQ-CNN as input.

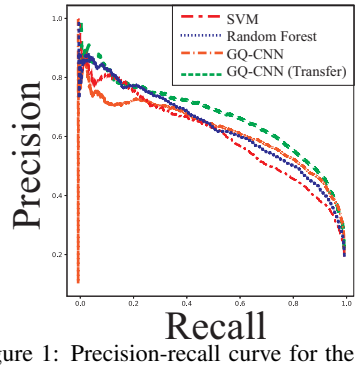

Figure 1: Precision-recall curve for the top four machine learning models on a fixed validation subset of the Dex-Net 2.1 $\epsilon = 0.9$ dataset containing approximately 20k datatpoints.

Table 1 compares the classifcation accuracy, AP, and percentage of objects cleared (successful grasps) from rolling out bin picking policies with various levels of niose injection in the simulator. We see that the policy with high noise ($\epsilon = 0.9$) has the highest classification accuracy and the policy with low noise ($\epsilon = 0.1$) level clears the most objects. This suggests that the policy trained with low noise levels may be overly optimistic while the policy with high noise tends to be pessimistic and fails to find successful grasps on many timesteps.

2.2 t-SNE Embedding

Noise (%)	Acc.(%)	AP	(%) Cleared
0.1	5.4	**0.8**	**62**
0.5	5.3	0.3	43
0.9	**13.8**	**0.8**	44

Table 1: Performance of bin picking policies on rollouts in the simulator as a function of the level of noise injection.

To gain insight into the feature space of the GQ-CNN finetuned on Dex-Net 2.0, we visualized a t-distributed Stochatic Neighbor (t-SNE) embedding [2] of the fc4 features. Specifically, we projected the validation subset of the Dex-Net 2.1 ($\epsilon = 0.9$) dataset on the 100 principal components of the fc4 features for the training subset of the data, using the fc4 features of both the Dex-Net 2.0 GQ-CNN and Dex-Net 2.1 ($\epsilon = 0.9$) GQ-CNN. Then we ran the Barnes-Hut version of t-SNE with tree-based acceleration [3] to compute a 2-dimensional embedding.

Fig. 2 compares the t-SNE embedding for the Dex-Net 2.0 GQ-CNN trained on singulated objects and the Dex-Net 2.1 ($\epsilon = 0.9$) GQ-CNN trained for bin picking. For each datapoint we plot the transformed grasp image colored by whether or not grasps are from the supervisor's action set (green) or from the set failed grasps from the 90% random actions (red). We see that in the original feature space some of the supervisor's grasps are mixed in with negative examples, and many of these examples have some level of clutter around the target object. This suggests that some robust grasps in clutter cannot be discriminated from failed grasps based on the GQ-CNN weights. There are also examples of false positives for the Dex-Net 2.0 GQ-CNN, such as the top-leftmost red image which is not collision-free for the gripper. However, after fine-tuning [4] we see that the supervisor's actions are more tightly clustered, with fewer negative examples mixed it.

2.3 Qualitative Differences

We also looked into the failure modes across all methods to gain insights into the performance gains for the Dex-Net 2.1 bin picking policy. Two examples are illustrated in Fig. 3.

The first demonstrates collision avoidance in piles. The Dex-Net 2.0 GQ-CNN predicts grasp success for a grasp that will put the hand in collision with the pile. We hypothesize that this is due to the GQ-CNN picking up on the edge between the objects as a graspable area without considering collisions, as it has only seen single objects in training. However, the Dex-Net 2.1 GQ-CNN plans a grasp that avoids the other objects in the pile, picking out a single one.

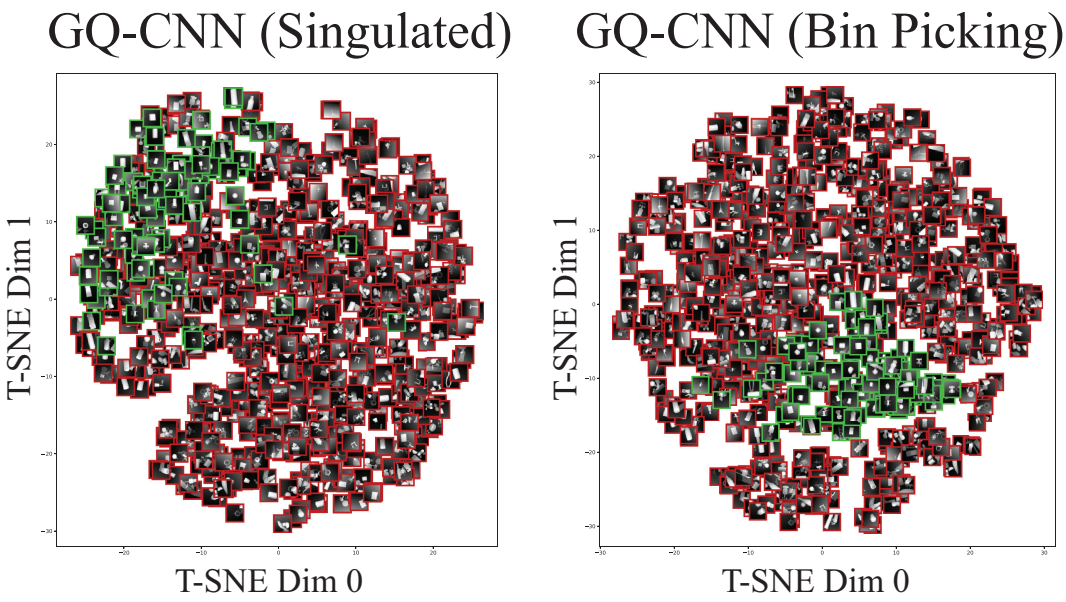

Figure 2: t-SNE embedding of the first 100 principal components of fc4 features for the validation subset of the Dex-Net 2.1 ($\epsilon = 0.9$) dataset (best viewed in color). (Left) The embedding for features from the original Dex-Net 2.0 GQ-CNN. (Right) The GQ-CNN finetuned on Dex-Net 2.1 ($\epsilon = 0.9$). Each datapoint shows the rotated and translated depth image that is input to the to GQ-CNN (see [1]). Images corresponding to actions taken by the algorithmic supervisor are outlined in green while images corresponding to failed random actions are outlined in red.

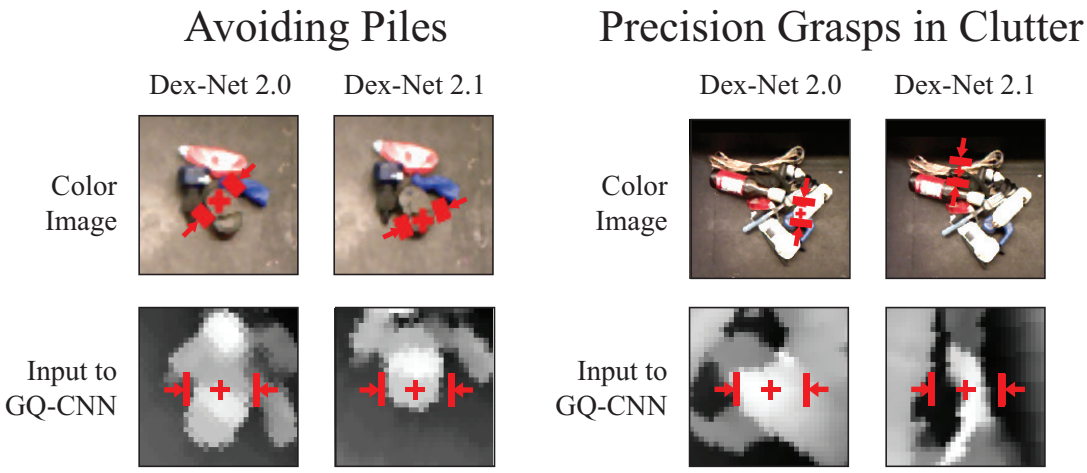

Figure 3: T-SNE embedding of fc4 features for the validation subset of the Dex-Net 2.1 ($\epsilon = 0.9$) dataset of rollouts from the algorithmic bin picking supervisor for the original Dex-Net 2.0 GQ-CNN (Left) and the GQ-CNN finetuned on Dex-Net 2.1 ($\epsilon = 0.9$). Images corresponding to actions taken by the supervisor are outlined in blue while images corresponding to failed random actions are outlined in red.

The second example demonstrates that the Dex-Net 2.1 policy can also find narrow openings to place the fingers in a dense heap in order to achieve a robust grasp. The Dex-Net 2.0 GQ-CNN falsely predicts that a grasp of the tip of the glue gun will be successful, perhaps because it has seen no similar example in training. On the other hand, the Dex-Net 2.1 policy finds the frame of the goggles on the top left of the pile, where it can safely fit a finger between the goggles and bottle of sauce.

3 Details of POMDP Parameters

For replicability, this section lists the numeric values of parameters used in our Partially Observable Markov Decision Process (POMDP) model of bin picking. The maximum time horizon used was 50 timesteps. If the heap was cleared before this time, then the rollout ended immediately.

3.1 Initial State Distribution (ρ)

The initial state distribution is a product of the following distributions:

1. *Object Count (m):* Poisson distribution with mean $\lambda = 5$.
2. *Object Heap ($\mathcal{O}$):* Discrete uniform distribution over the $1,500$ 3D triangular meshes $\{\mathcal{M}_0, ...\mathcal{M}_{m-1}\}$ from the KIT [5] and 3DNet [6] datasets, as well as the pose from which each mesh is dropped into the heap. The pose distribution consists of a fixed drop height of $0.2m$, a random planar translation above the table in $[-0.1, 0.1] \times [-0.1, 0.1]$ (units of meters), and a random rotation sampled by first uniformly sampling spherical coordinates, then sampling an angle in $[0, 2pi)$ radians for the orientation in the table plane.
3. *Depth Camera ($\mathcal{C}$):* Uniform distribution over the camera pose using spherical coordinates $r, \theta, \varphi \sim \mathcal{U}([0.65, 0.75] \times [0, 2\pi) \times [1°, 10°])$, where the camera optical axis always intersects the center of the table. The camera focal length was sampled uniformly from $[520, 530]$ pixels.
4. *Coulomb Friction (α):* Truncated Gaussian with mean 0.5 and variance 0.1, constrained to $[0, 1]$.

The maximum distance that an object could translate or roll from the world center before removal was $W = 0.2$.

3.2 Next State Distribution (p)

The next state distribution used the parameters of [1] for the robust epsilon metric, which consists of

1. **Gripper Pose:** An isotropic Gaussian with mean on the target gripper pose, a translational variance of $1.0mm$, and a rotational variance of 0.001 in Lie Algebra coordinates.
2. **Object Pose:** An isotropic Gaussian with mean on the true object pose, a translational variance of $5.0mm$, and a rotational variance of 0.01 in Lie Algebra coordinates.

We used the wrench space norm of $\|\mathbf{w}| = \sqrt{\|\mathbf{f}\|_2^2 + \nu\|\tau\|_2^2}$ [7] with $\nu = 1000$. If an object translated or rolled from the world center a distance greater than $W = 0.2$, then it was re-dropped in the pile using the initial object pose distribution of ρ.

3.3 Observation Distribution (q)

Our observation distribution follows [1]. We model images as $\mathbf{y} = \alpha * \hat{\mathbf{y}} + \epsilon$ where $\hat{\mathbf{y}}$ is a rendered depth image created using OSMesa offscreen rendering. We model α as a Gamma random variable with shape$= 1000.0$ and scale=0.001. We model ϵ as Gaussian Process noise drawn with measurement noise $\sigma = 0.005$ and kernel bandwidth $\ell = \sqrt{2}px$.

References

[1] 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 *Proc. Robotics: Science and Systems (RSS)*, 2017.

[2] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[3] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[4] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[5] 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. *Int. Journal of Robotics Research (IJRR)*, 31(8):927–934, 2012.

[6] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze. 3dnet: Large-scale object class recognition from cad models. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 5384–5391. IEEE, 2012.

[7] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2290–2295, 1992.