2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

REPLAB: A Reproducible Low-Cost Arm Benchmark for Robotic Learning

Brian Yang[1], Dinesh Jayaraman[2], Jesse Zhang[3], and Sergey Levine[4]

Abstract— **Standardized evaluation measures have aided in the progress of machine learning approaches in disciplines such as computer vision and machine translation. In this paper, we make the case that robotic learning would also benefit from benchmarking, and present a template for a vision-based manipulation benchmark. Our benchmark is built on "REPLAB," a reproducible and self-contained hardware stack (robot arm, camera, and workspace) that costs about 2000 USD and occupies a cuboid of size 70x40x60 cm. Each REPLAB cell may be assembled within a few hours. Through this low-cost, compact design, REPLAB aims to drive wide participation by lowering the barrier to entry into robotics and to enable easy scaling to many robots. We envision REPLAB as a framework for reproducible research across manipulation tasks, and as a step in this direction, we define a grasping benchmark consisting of a task definition, evaluation protocol, performance measures, and a dataset of over 50,000 grasp attempts. We implement, evaluate, and analyze several previously proposed grasping approaches to establish baselines for this benchmark. Project page with assembly instructions, additional details, and videos: `https://goo.gl/5F9dP4`.**

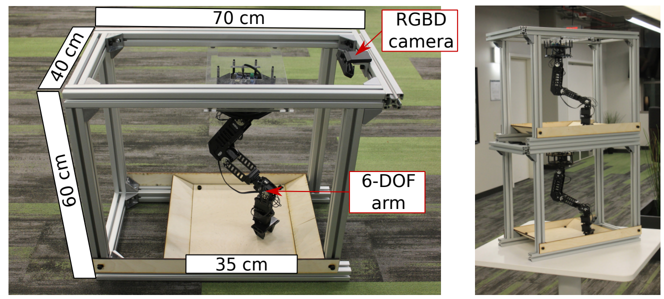

Fig. 1. **(Left)** One REPLAB cell with annotated dimensions **(Right)** Two REPLAB cells stacked on top of each other on a desk.

I. Introduction

Since the 90's, the study of artificial intelligence has been transformed by data-driven machine learning approaches. This has been accompanied and enabled by increased emphasis on reproducible performance measures in fields like computer vision and natural language processing. While benchmark-driven research has its pitfalls [1], [2], well-designed benchmarks and datasets [3], [4], [5] drive increased research focus on important problems, provide a way to chart the progress of a research community, and help to quickly identify, disseminate, and improve upon ideas that work well.

In robotic manipulation, establishing effective benchmarks has proven exceedingly challenging, especially for robotic learning, where the principal concern is with the generalization of learned models to new objects and situations, rather than raw proficiency on a single narrow task. An important reason for this is that progress in robotics comes not only through improvements in control algorithms, but also through improvements in hardware (such as sensing and actuation). Traditional approaches to robotic control are closely intertwined with the specifics of the robotic hardware—for instance, grasping with a parallel-jawed gripper, a five-fingered hand, and a suction cup would all be treated as different tasks, each requiring their own different control algorithms. In this view, the large space of hardware choices and tasks makes it futile to attempt to meaningfully measure progress through a few focused benchmarks.

However, in the light of relatively recent changes in the research landscape, we contend that it may now be time to reconsider the idea of manipulation benchmarks. First, research in machine learning-based manipulation [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] aims to develop data-driven approaches that are, at least to some degree, agnostic to the particular choice of hardware—although models trained on one platform are unlikely to work on another, the same learning algorithms can in principle be deployed on new platforms with minimal additional engineering. The performance of such an approach on one hardware platform is generally expected to be predictive of its performance on other platforms too. Given this, we might hope that progress in learning-based control may be treated as orthogonal to hardware improvements. Thus it may now be possible to meaningfully consolidate the space of task definitions and hardware configurations to a small representative set, which is a prerequisite for defining a benchmark.

Next, today's robotics hardware is already mature enough to permit the human-teleoperated performance of tasks that are substantially harder than those that can be done with automated control methods [17]. It is therefore reasonable to conclude that control, not hardware, is now the primary bottleneck for progress in robotics, and manipulation in particular. This means that a robotic learning benchmark is not merely *possible* as discussed above, it could potentially serve a very important purpose to the research community.

What would a manipulation benchmark accomplish? Recent reinforcement learning (RL) benchmarks such as ALE [18] and Open AI Gym [19] are useful reference points

*This research was supported by Berkeley DeepDrive, NSF IIS-1651843, and the Office of Naval Research (ONR), Google, NVIDIA, and Amazon.

[1] brianhyang@berkeley.edu, UC Berkeley.
[2] dineshjayaraman@eecs.berkeley.edu, UC Berkeley.
[3] jessezhang@berkeley.edu, UC Berkeley.
[4] svlevine@eecs.berkeley.edu, UC Berkeley.

to answer this question. They serve three key functions for the RL community: enabling apples-to-apples comparison of RL algorithms by standardizing environments and tasks, enabling fast and easy replication and improvement of research ideas, and driving increased research by lowering the barrier to entry into RL.

In this paper, we propose a robotic learning benchmark by developing reproducible evaluation procedures with standardized hardware. In particular, we focus on arguably the most widely studied robotic manipulation task: grasping. We describe a reproducible "REPLAB" work cell design based on a low-cost commercially available robotic arm and an RGBD camera, and present a dataset that can be used together with this standardized hardware setup to evaluate learning algorithms for robotic grasping. We implement and evaluate prior grasping approaches on this cell platform to set up a grasping benchmark.

REPLAB's design is motivated by the following goals, in order of priority: (i) to provide a consistent and reproducible progress metric for robotic learning, (ii) to lower the barrier to entry into robotics for researchers in related disciplines like machine learning so that robotic learning research is not restricted only to a small number of well-established labs, (iii) to encourage and enable plug-and-play reproducible software implementations of robotic learning algorithms, by promoting a standardized and exhaustively specified platform, (iv) to allow easily scaling up and parallelizing robotic learning algorithms across multiple robots, potentially exploiting data from REPLAB cells across multiple research labs, and promote testing for generalization to new robots, and (v) to be able to afford to evolve through iterative community-driven improvement of the REPLAB platform itself, a luxury that would not be available with a more expensive design.

II. Related Work

Robotic approaches today are largely tested in custom settings: environments, hardware, task definitions, and performance measures may all vary from paper to paper. While the problem this raises for measuring the effectiveness of different approaches is widely acknowledged in the robotics community [20], [21], [22], [1], solutions have been elusive.

The majority of prior approaches to benchmarking in robotics have taken the form of a live competition between approaches, e.g., the DARPA Grand Challenge, Amazon Picking Challenge, and RoboCup. Each competing approach, consisting of specific hardware setups as well as control algorithms, is tested in the same physical location. This provides valuable performance measures of complete robotic systems, but it is logistically difficult to provide more than sparse point estimates of performance for each approach on a yearly basis.

Beyond such live competitions, for grasping in particular, there have been other previous efforts to standardize various aspects of the task. The YCB dataset [23] is a standardized diverse object set for evaluating all grasping approaches. The ACRV benchmark [24] goes one step further and proposes not only a standard object set, but also a standard test setting with a specified shelf design and specified object configurations within the shelf. The authors of DexNet [11] share a dataset of synthetic grasps to train grasp quality convolutional networks, and offer to perform on-robot evaluation of models with high accuracy on held-out grasps. OpenGRASP[25] proposes fully standardized task, hardware, and performance metrics are for grasping, but in a simulated environment. To our knowledge, ours is the first effort to propose a standardized *complete* stack for real-world grasping, including the full hardware configuration (such as robot, sensors, and work cell design) and performance measures.

REPLAB is also built with collective robotics in mind. Prior efforts in this direction include [7], [26], where data collection for grasping was parallelized across many physically collocated robots. Rather than a such a collocated group, the Million Object Challenge (MOC) [27] aims to crowdsource grasping data collection from 300 Baxter robots all over the world. REPLAB cells are designed to fit both these cases, since they are low-cost, low-volume, reproducible, and stackable: 20 REPLAB cells stacked to about 2m elevation occupy about the same floor space and cost less than two times as much as a single Baxter arm. The closest effort to this trains grasping policies for low-cost mobile manipulators [16] by collecting data from several such manipulators under varying lighting conditions.

Finally, previous efforts have also provided standardized and easily accessible full hardware stacks such as Duckietown for navigation [28] and Robotarium for swarm robotics [29]. We share their motivation of democratizing robotics and driving increased participation, and our focus is on manipulation tasks.

III. REPLAB Design Overview

We now describe various key aspects of the design of the REPLAB platform and grasping benchmark. Exhaustive documentation for constructing a complete REPLAB cell are hosted at: `https://goo.gl/5F9dP4`.

A. Cell Design

A REPLAB cell, shown in Fig 1, is a portable, self-contained complete hardware stack (arm, sensor, workspace, and cage) for manipulation tasks. It occupies a cuboid of size 70x40x60 cm (length, width, height). The outer cage is constructed with easily composable lightweight aluminum rods manufactured to our specifications. A low-cost WidowX arm from Interbotix Labs is suspended upside down and its base is mounted to the ceiling of the cell to maximize its reachable effective workspace. The arm has six degrees of freedom: a 1-DOF rotating base, three 1-DOF joints, and a 1-DOF rotating wrist, and a 1-DOF parallel-jawed gripper with minimum width 1 cm and maximum width 3 cm. A Creative Blasterx Senz3D SR 300 RGB-Depth camera is mounted to a specified standard position on the ceiling near the front of the cell so that the entire workspace is comfortably within its optimal field of view and operating distance. Mounts for the robot, the camera, and a 35x40 cm workspace are manufactured through laser cutting. A full list of parts, laser cutting templates, and all design parameters are exhaustively recorded shared on the project page for reproducibility. We verified that an undergraduate student with little prior robotics experience was able to build a REPLAB cell using our instructions within three hours, (given a pre-assembled arm and all other required components).

The physical cell dimensions are designed to allow stacking of multiple cells on top of one another, as shown in Fig 1. With our current design, we expect that it will be feasible for up to 20 arm cells, stacked to 2.2 metres in height (four cell heights), to occupy the same floor space as a typical setup for a single Baxter arm, for instance.

A single REPLAB cell costs about 2000 USD, and can be assembled in a few hours. Together with one spare servo for each of six servos on the arm, the cost is under 3000 USD. This is comparable to the cost of a single workstation. During experiments for this paper, REPLAB cells proved to be quite robust. With software constraints in place to avoid arm collisions with the boundaries of the REPLAB cell, we encountered no major breakages over more than 100,000 grasp attempts. No servos needed to be replaced. Repair maintenance work was largely limited to occasional tightening of screws and replacing frayed cables.

B. Camera-Arm Calibration

We perform camera-robot calibration on a single REPLAB cell by using a checkerboard and registering robot coordinate points to 3D image pixels from the camera. Since the cell design is exhaustively specified, our construction protocol ensures that the same calibration matrix may be reused for other cells.

In particular, for each cell after the first, we propose to finely adjust the camera position so that its view of its workspace and robot are aligned to those from the first cell camera. Fig 3 shows this calibration protocol in action. While calibration from scratch is frequently time-consuming, this protocol enables simple calibration for all cells, and also helps ensure that all cells are near-identical in construction. We have applied this protocol in constructing our second REPLAB cell, and verified that it works in practice. Sec IV-B presents quantitative evidence for this.

C. Control Noise Compensation

For all arm motions, we use ROS for inverse kinematics and planning with the *MoveIt* package and execution through PID position control. While calibration noise is minimal, control noise is more difficult to avoid given that we use low-cost arms [15], [16].

In our setting, we found that control noise is primarily along the horizontal coordinates (x, y). We tackle this using a simple approach. Since most desired grasping targets are near the cell floor, we first command the arm to move its end effector over a 5x5 grid on the floor and record the actual achieved positions of the end effector using the planner and controller described above. Comparing the target positions p_i and achieved positions q_i, we fit a linear model $q = \alpha p + \beta$, where parameters α and β are learned for each cell separately.

Having calibrated the control noise, we can compensate for it by setting the target position to $p' = (p - \beta)/\alpha$. For our two cells, we set β to 0 and α to 0.87 and 0.95. We find that this simple approach works well to eliminate most control noise. Combining camera calibration noise and residual control noise after compensation, the end-effector positions are within 2 cm of the target over the 35 x 40 cm workspace floor, and usually smaller near the center. Qualitatively, this error falls within the tolerance that the grasping task (defined below) naturally permits.

D. Grasping Task Definition

REPLAB is intended to serve as a common platform for benchmarking robotic manipulation tasks. In our first benchmark, we focus on arguably the longest studied manipulation task: grasping. Multiple objects are randomly scattered over the cell floor. Each grasp attempt may target any object in the workspace. All algorithms have access to a standard input-output interface. The RGBD image and raw point cloud from the camera are available as inputs. The RGB image, depth image, and point cloud are shown for a sample grasp attempt in Fig 4. The algorithm output is required to be a fixed target grasp configuration.

We restrict the gripper to be oriented vertically. This restriction is used in a number of prior works [7], [8], [13], [12] and significantly simplifies inverse kinematics and planning, since the arm is unlikely to collide with clutter during motion towards a grasp point. A grasp is specified fully by a robot coordinate 3D point (x, y, z), and a wrist angle θ for the parallel-jawed gripper. The arm is moved first to a position directly above the intended grasp, and then lowered to the correct grasp position. Once the target coordinates are achieved, the parallel-jaw gripper is closed, and the arm is moved into a preset standard configuration, with the gripper facing the camera, and held for two seconds. A successful grasp requires the object to stay in the gripper throughout this time. This protocol is common to all evaluated approaches. Grasp success detection is discussed in Sec III-F.

E. Objects

As pointed out in Sec II, standard object sets for grasping have previously been proposed in [23], [24]. However, since these object sets were designed for much larger arms, we design new object sets for REPLAB — a training set with over 100 objects of varying shapes and sizes, a "seen object" test set of 20 toys among the training objects, and an "unseen object" test set of 20 toys. Our objects are of varying shapes and sizes, with about 50% hard plastic toys and 50% soft toys. We will specify shopping lists for reproducibility. Toys are picked so that there is at least one feasible stable grasp with the gripper. Some sample objects are shown in Fig 2.

F. Dataset and Data Collection

We have collected a dataset of over 50,000 randomly sampled grasps together with grasp success labels collected

Fig. 2. We train learning-based grasping approaches on over 50k grasp attempts with over 100 objects, and evaluate them on two sets of objects: 20 seen objects (sampled from the training objects), and 20 unseen objects from a different distribution with more complicated shapes. Here, we show a subset of seen (top) and unseen (bottom) test objects used in our benchmark evaluations.

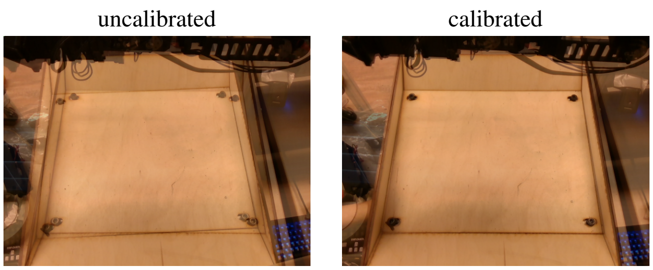

Fig. 3. To calibrate REPLAB cells, we propose a protocol of finely adjusting the camera position until its camera image aligns nearly perfectly with that from the first REPLAB cell. Here, images from our two REPLAB cells are shown overlaid on top of each other before (left) and after (right) alignment.

using two REPLAB cells in parallel, at the rate of about 2,500 grasps per day per cell. For each grasp attempt, the 3D point cloud returned by the camera is clustered using DBSCAN [30] to find objects, and a random cluster is selected. A grasp is sampled as follows: grasp coordinates (x, y, z) are sampled with a small random perturbation from the center of the selected cluster. The grasp angle θ is sampled uniformly at random.

We have implemented an automatic data collection procedure that is able to collect over 2500 grasp attempts on a single cell in 24 hours, with minimal manual intervention (on average, fewer than two interventions per cell per day). Roughly 21% of grasp attempts during random data collection result in successful grasps. Success/failure label for grasps is semi-automatic: we first manually labeled 5k grasps by looking at the image of the held-up object, then trained a classifier to predict grasp success given the RGBD image and the gripper width as inputs. This model yields 99.6% classification accuracy on held-out data from both cells, and is used to aid in labeling the rest of the data without exhaustive manual labeling.

G. Evaluation and Performance Metric

Evaluation is done on an episodic bin-clearing task. At the start of an episode, 20 objects are scattered over the workspace floor using a fixed protocol: a box is filled with the objects, shaken, and inverted over the center of floor, similar to [31]. Each episode consists of 60 grasp attempts. For each grasp attempt, 500 grasp candidates are evaluated from the neighborhood of each cluster returned by DBSCAN. In particular, we sample (x, y, z) from points in the cluster and sample θ uniformly at random. Each successfully grasped object is automatically discarded to a clearing area outside the workspace, and one of the remaining objects must be picked at the next attempt. In rare cases when either clustering fails (number of objects is too low or too high), or there has been no successful grasp in 10 attempts, we sweep the arm over the workspace floor to perturb objects. We report cumulative success rate (CSR) plots of the number of successfully grasped objects vs. the number of grasp attempts. See Fig 5 for an example.

H. Plug-and-Play Software

We aim to lower the barrier to entry into manipulation research not just by keeping REPLAB costs low, but also by emphasizing ease of use and reproducible algorithm implementations. In particular, we will release all code in a Docker image that runs out of the box on Ubuntu machines, for quick reproducibility. Our image contains scripts for automatic data collection, grasp success annotation through the trained classifier, camera calibration, noisy control compensation, and benchmark evaluation. Further, it includes REPLAB-specific implementations of several baselines for grasping, described in the next section. With this image, setting up an Ubuntu laptop to start collecting data on a REPLAB cell takes about five minutes.

The importance of such plug-and-play implementations in accelerating research progress cannot be overstated, and we believe that this is one of the key advantages of a fully reproducible and standardized hardware platform. Going forward, we will invite and encourage authors of the leading approaches on the REPLAB benchmark to contribute implementations to include in future iterations of the Docker image. Any researcher with access to a REPLAB cell would be able to run the best-performing grasping approaches out of the box, and modify and build upon them.

IV. Experiments

We now present experiments that aim to answer the following questions: (i) Is manipulation feasible on the low-cost REPLAB platform despite noisy control? (ii) How suitable are REPLAB cells to serve as foundations for a

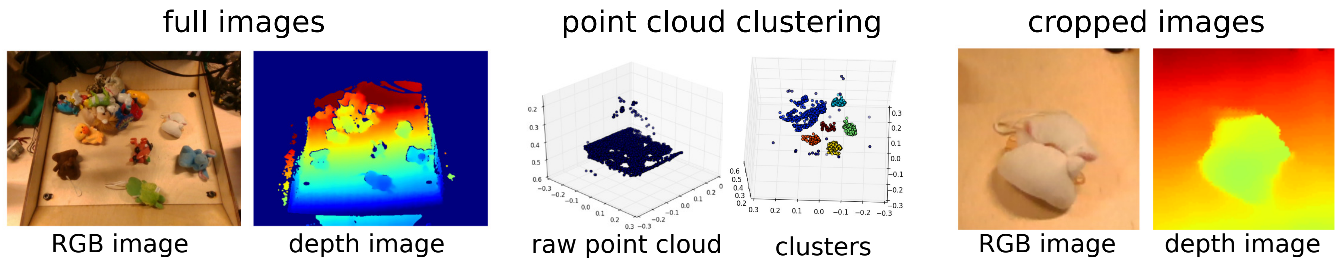

Fig. 4. Raw and preprocessed RGB images, depth maps (blue is close, red is far), and pointclouds used in various grasping approaches. `full-image` operates directly on the raw RGB and depth images. `random-xyzθ`, `random-θ`, and `principal-axis` rely only on discovering clusters in the point cloud. Point cloud clustering is shown in the middle, where background points are removed from the point cloud before running DBSCAN. In this example, DBSCAN successfully gets the isolated objects and fails on the objects clumped together in the top left, detecting them as a single cluster. `pinto2016` uses input images cropped to the neighborhood of the candidate grasp (shown here to the right).

manipulation benchmark? In particular, does our grasping benchmark evaluation protocol produce consistent, reproducible evaluations across multiple REPLAB cells? (iii) What are the best-performing baseline approaches on the REPLAB evaluation protocol, and what can we learn by analyzing their performance?

A. Grasping Approaches

We implement and evaluate five grasping approaches in all. The first three are based on sampling near clusters detected in the point cloud: (i) `random-xyzθ`: grasp angle θ is sampled uniformly at random, and grasp coordinates (x, y, z) are perturbed with random uniform noise in a 4x4x2 cm region from each cluster center, (ii) `random-θ`: Only θ is random, where (x, y, z) is set to a cluster center, and (iii) `principal-axis`: we find the major axis of a cluster by computing the largest eigenvector of the correlation matrix of (x, y) coordinates of points in the cluster. A grasp is attempted along the perpendicular bisector of this axis. z is fixed to the cluster center.

We evaluate two approaches based on training convolutional neural networks to predict the quality of a grasp in a given scene: (i) `full-image` [7], [12] takes two inputs: the full image of the workspace and the full (x, y, z, θ) parameterization of a candidate grasp, (ii) `pinto2016` [8] instead crops the input image around the (x, y, z) position of the candidate grasp and predicts success or failure for each of 18 quantized θ bins. The inputs are schematically shown in Fig 4. These are both trained on the same set of 50k random grasps described in Sec III-F.

For testing on the robot, each baseline approach is provided with grasp candidates from which it picks one to execute. `full-image` and `pinto2016` evaluate the grasp quality of 512 grasp candidates per detected cluster, each parameterized by (x, y, z, θ) as described in Sec III-G, before executing the highest quality grasp. `random-θ` and `random-xyzθ` select one cluster at random before selecting grasps near that cluster center. For `principal-axis`, we assign a confidence score to each cluster based on the ratio of the largest eigenvalue to the smallest, and select the cluster with the highest confidence.

In practice, selecting only the highest confidence grasp tends to lead to the arm getting stuck in a loop attempting the same unsuccessful grasp over and over. To prevent this, we sample from top-5 grasp candidates for the learning approaches and the top 5 clusters for `principal-axis`.

B. Reproducibility of Evaluation

We have taken care in the design of REPLAB cells to make it possible to construct near-identical replicas. A reproducible hardware platform is key to establishing reproducible evaluation procedures, which is the primary aim of REPLAB. Evaluations of control algorithms should produce similar results on all REPLAB cells.

With this in mind, we have proposed a calibration protocol in Sec III-B so that two REPLAB cells should in theory share the exact same calibration matrix C mapping camera coordinates to robot coordinates. We have constructed two REPLAB cells using this procedure—the second cell inherits the calibration matrix computed for the first cell. To evaluate whether the cells are indeed constructed near-identically, we collect a small dataset of 25 corresponding camera and robot coordinate points p_{cam} and p_{arm} in each cell using a checkerboard (similar to correspondences used in calibration). We then measure the average calibration errors $\|Cp_{cam} - p_{arm}\|_2$ for each arm—if the cells are indeed identical, calibration errors should be similar for both cells. For the first and second cell respectively, the errors are 0.87 cm and 1.52 cm. Given a gripper of width 3 cm, this difference is tolerable and in practice leads to the same grasping behavior, as we show below.

To measure reproducibility in the context of grasping evaluation, we evaluate the `principal-axis` grasping baseline over three runs on each cell separately. Fig 5 shows cumulative success rate (CSR) plots. As seen here, the population of CSR curves is evenly matched across the two cells.

C. Benchmark Baseline Performance

We now evaluate all five baselines on our grasping benchmark. First, our learning-based baselines `pinto2016` and `full-image`, trained on our dataset of random grasps, yield 57.7% and 58.9% accuracy respectively on a balanced, held-out validation dataset of sampled grasps.

Moving to on-robot evaluation, Fig 5 (middle) shows the cumulative success rate (CSR) curve on seen objects.

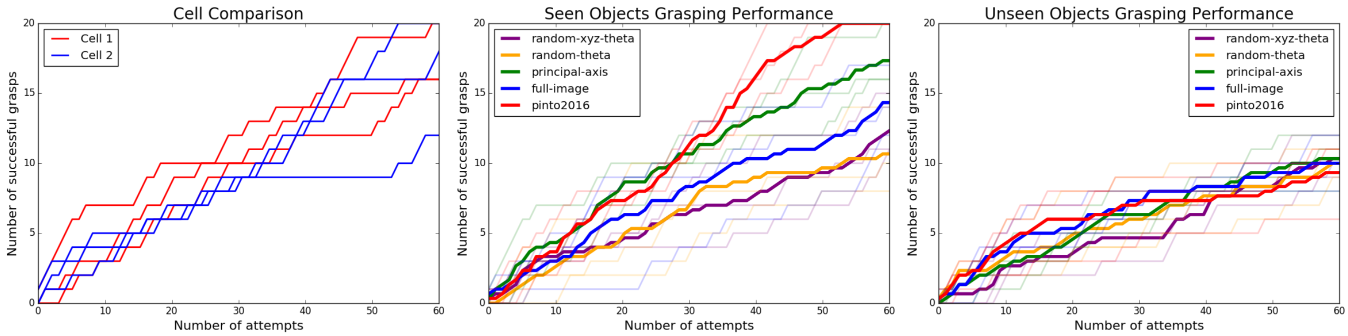

Fig. 5. **(Left)** Cumulative Success Rate (CSR) plots from three runs on each REPLAB cell for the `principal-axis` baseline. Quantitatively and qualitatively, we observe very similar behavior on both cells with the same model, verifying the reproducibility of the REPLAB platform. **(Middle)** Cumulative Success Rate (CSR) plots for all baselines on seen objects. The mean over three runs is plotted in the thick curves, while the faded lines show individual runs for each method. **(Right)** Cumulative Success Rate (CSR) plots for all baselines on unseen objects. The mean over three runs is plotted in thick curves, while the faded lines show individual runs for each method.

`pinto2016` clears all 20 test objects each time, emerging as the leading approach, followed by `principal-axis`, and `full-image`. We believe that the advantage of `pinto2016` over `full-image` comes from preprocessing the image inputs to focus on the region of interest. In contrast, `full-image` must *learn* this association between grasp parameters and image locations with only grasp success/failure as supervision. `full-image` also requires a larger network to process its larger inputs, making it more prone to overfitting. We expect that this gap in performance will fall as the size of the grasping dataset increases. Both learning approaches `pinto2016` and `full-image` also produce low-variance CSR curves compared to `principal-axis`, which is the third best approach. `principal-axis` relies heavily on discovering objects through clustering, and has high variance early in evaluation runs when objects clump together.

Fig 5 (right) shows the CSR curves for unseen objects that were not encountered during training. By design, our unseen objects are significantly more complex shapes than were seen at training time, as shown in Fig 2. Unsurprisingly, all methods perform worse on this set. `principal-axis` explicitly relies on objects having simple ellipsoidal geometries, and struggles to handle these more complex shapes. The learning-based approaches `pinto2016` and `full-image` fare marginally better during early grasp attempts, but their ability to generalize to these objects is limited by the fact that training data was collected using only objects with relatively simple geometries—we found that a training set of simple objects was important to ensure sufficient success rate (about 20%) at data collection time so that there were enough successful grasps in training data. We expect that learning-based methods will benefit from a curriculum-based approach for collecting a larger dataset, where the current best policy may be deployed (with some exploration) to collect data on increasingly more difficult objects.

Note that over all approaches and all trials reported in Fig 5, the fastest clearance still takes over 40 attempts to clear 20 objects. Together with the performance on the unseen object set, this is a good sign for REPLAB as a benchmark; reasonable baselines still have plenty of room for improvement.

Finally, we perform a data ablation study on `pinto2016`, evaluating only held-out validation accuracy on the seen object grasps. With 5k, 10k, 15k, 25k, and 45k grasps, accuracies are 55.1%, 55.2%, 58.3%, 59%, and 57.7% respectively. The diminishing returns suggest that our dataset is large enough to train this model for the seen objects. However, we expect that more data would benefit larger models, as well as the ability to generalize to unseen objects. Our entire dataset will be available on the project webpage.

V. Future Work

We have proposed a fully standardized hardware stack on which to develop reproducible evaluation procedures for manipulation tasks. To illustrate the use of such a platform, we have described a grasping benchmark. One immediate shortcoming with the current platform in terms of its widespread adoptability is the reliance on a specific robot arm supplier. We plan to address this in future versions of the REPLAB platform through a 3D-printable arm design. We also plan to build upon this foundation by (i) inviting participation through an open challenge, where leading methods on held-out data validation accuracy would be evaluated on our REPLAB cells, (ii) releasing full simulators for REPLAB cells, and (iii) implementing more grasping approaches on REPLAB.

We also plan to develop a larger challenge dataset for grasping, and release open-source code for robotic control approaches such as visual servoing, video prediction-based model predictive control, and reinforcement learning on the REPLAB platform. We invite other dataset and software contributions from the robotics research community.

Acknowledgments

We thank Dian Chen and Fred Shentu for their help in early stages of the project, Jeff Mahler, David Gealy, Roberto Calandra, and Jitendra Malik for helpful discussions, Mohammad Kamil Shams and Alan Downing for participating in verifying the reproducibility of REPLAB and providing useful feedback, and Trossen Robotics for their assistance with the WidowX arm.

References

[1] V. C. Müller, "Measuring progress in robotics: Benchmarking and the measure-target confusion," in *Metrics of sensory motor coordination and integration in robots and animals*, 2018.

[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1521–1528.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[4] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communication*, 1990.

[5] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *JMLR*, 2004.

[6] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, 2015.

[7] 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," *IJRR*, 2018.

[8] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *ICRA*, 2016.

[9] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *IROS*, 2016.

[10] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *CORL*, 2017.

[11] 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," 2017.

[12] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?" *CORL*, 2017.

[13] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *arXiv preprint arXiv:1710.01330*, 2017.

[14] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *RAL*, 2018.

[15] M. P. Deisenroth, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," *RSS*, 2012.

[16] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *NIPS*, 2018.

[17] J. Vertut, *Teleoperation and robotics: applications and technology*. Springer Science & Business Media, 2013, vol. 3.

[18] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, 2013.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[20] F. Bonsignorio and A. P. Del Pobil, "Toward replicable and measurable robotics research [from the guest editors]," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 32–35, 2015.

[21] A. P. del Pobil, R. Madhavan, and E. Messina, "Benchmarks in robotics research," in *Workshop IROS*, 2006.

[22] R. Dillmann, "Benchmarks for robotics research," *European Robotics Network (EURON), IST-2000-26048*, 2004.

[23] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *IEEE Robotics and Automation Magazine*, 2015.

[24] J. Leitner, A. W. Tow, N. Sünderhauf, J. E. Dean, J. W. Durham, M. Cooper, M. Eich, C. Lehnert, R. Mangels, C. McCool, *et al.*, "The ACRV picking benchmark: A robotic shelf picking benchmark to foster reproducible research," in *ICRA*, 2017.

[25] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann, "The OpenGRASP benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation," in *IROS*, 2011.

[26] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *CORL*, 2018.

[27] S. Tellex. Million object challenge. [Online]. Available: http://h2r.cs.brown.edu/million-object-challenge/

[28] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, *et al.*, "Duckietown: an open, inexpensive and flexible platform for autonomy education and research," in *ICRA*, 2017.

[29] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1699–1706.

[30] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *AAAI*, 1996.

[31] A. t. Pas and R. Platt, "Using geometry to detect grasps in 3d point clouds," *International Symposium on Robotics Research*, 2015.