A Large-Scale Benchmark Study Investigating the Impact of User Experience, Task Complexity, and Start Configuration on Robot Skill Learning

Asif Rana[1], **Daphne Chen**[1], **Reza Ahmadzadeh**[2], **Jake Williams**[1],
Vivian Chu[1], **and Sonia Chernova**[1]
[1]Institute for Robotics & Intelligent Machines, Georgia Institute of Technology
[2]Department of Computer Science, University of Massachusetts Lowell

Abstract

In this work, we contribute a large-scale study benchmarking the performance of existing skill learning from demonstration approaches. By way of this study, our aim is to bring to attention some of the challenges associated with skill learning in real-world scenarios and the limitations of existing methods in overcoming these challenges. We provide our conclusions based on ratings provided by Amazon Mechanical Turk (AMT) users for the task executions.

1 Introduction

Given the number and diversity of existing techniques in the area of trajectory-based skill learning from demonstration [2], it is critical that comprehensive empirical studies be performed comparing the relative strengths of these learning techniques. Prior work by Lemme et al. [13] has contributed a valuable benchmarking framework to evaluate the performance of reaching motion generation approaches on a 2D handwriting dataset. However, to date, no large-scale benchmarking study has been conducted on real robot data under a wide range of execution conditions.

Based on the choice of model representation, the majority of existing techniques in this field can be broadly categorized into one of four categories: statistical approaches [6, 4], dynamical systems [11, 12, 9, 19], geometric techniques [16, 14, 1], or probabilistic inference [18, 17, 8, 20]. In this work, we compare the performance of four algorithmic techniques, one from each category, i.e., TpGMM [4], CLFDM [12], TLGC [1], and ProMP [17] respectively. Our experimental design was aimed at simulating the challenges associated with real-world settings and hence identifying the limitations of existing approaches in overcoming them. Specifically, we evaluate how i) complexity of the task, ii) the expertise level of the human demonstrator, and iii) the starting configuration of the robot affect performance of each technique. To perform this evaluation, we collected data from 9 participants, across 4 different manipulation tasks with varying starting conditions, on a Rethink Sawyer robot. The resulting demonstrations were used to train 180 task models. Each of the resulting models was then tested and video recorded for new starting configurations, resulting in 720 videos of robot task reproductions. Finally, we compared the methods based on 3,600 Amazon Mechanical Turk queries which rated the performance of the robot in each video [1].

The scenario we evaluate in this study is that of a potential deployment of a robot as a consumer product. To this end, we pre-tune the parameters of each of the above algorithms to a default configuration that works well for the given class of problems, but we do not tune the parameters for each individual scenario following training, since this would be impractical in a real-world setting.

2 Experimental Design

Robot Tasks: We selected four tasks for evaluation (Fig. 1), each with differing goals: (i) *Reaching* - Move toward and touch the circle on the gray block; (ii) *Pushing* - Push the box lid closed;

[1]For the dataset and the accompanying videos: https://sites.google.com/view/rail-lfd

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

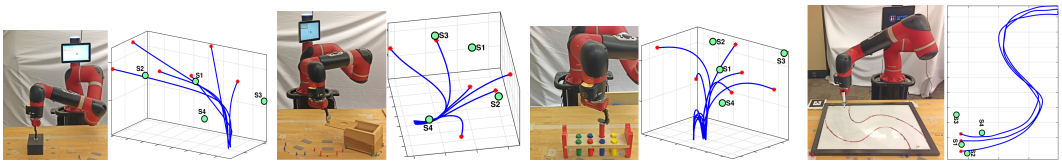

Figure 1: Each inset shows a task execution alongside a plot of an example task dataset. From left to right: *reaching*, *pushing*, *pressing*, and *writing*. Starting positions for reproduction (S1-S4) are shown as green markers.

(iii) *Pressing* - Push down peg $\#1$ and then peg $\#2$; and (iv) *Writing* - Draw an S-shape curve on the whiteboard. In terms of goal position, the *reaching* task posed the strongest requirements, followed by *pressing*, *pushing* and *writing* in that order. On the other hand, the *writing* task was the most constrained in direction of motion, followed by *pressing*, *pushing* and *reaching* respectively.

Participant Selection: To collect training data, we recruited 9 participants with differing levels of robotics experience: *Low, Medium* and *High*, with 3 participants belonging to each category.

Data Recording and Training: Data collection followed an IRB-approved human subjects study protocol. For each recorded trial, the robot was initialized to a preset starting configuration, then participants physically guided the robot through the desired motion. Each participant provided 21 total demonstrations in total (that is, 6 each for *reaching*, *pushing* and *writing*, and 3 for *writing*). For each participant, we combined all the demonstrations for a given task into a training dataset for that task. Over the entire population of participants, this resulted in 45 task datasets[2]. Before training, the datasets were smoothed and time aligned using dynamic time warping (DTW) [15]. Each of our 4 algorithms was then trained on each of the 45 datasets, resulting in 180 task models (one per participant-task-algorithm combination).

Starting Positions for Generalization: To validate generalizability, each task model was evaluated from 4 different starting robot configurations, $S1$-$S4$. Figure 1 visualizes the test starting positions for each task, overlayed over a set of example demonstrations provided by a participant. S1 was selected to be within 90% confidence interval around the mean of the starting configurations of the demonstrations. S2-S4 were selected outside this range, such that $d(S3) > d(S2) > d(S1) > d(S4)$, where $d(.)$ denotes the distance to the target object. S2 and S3 were chosen to be farther away from the target object than S1, and S4 was chosen to be closer to object.

AMT Evaluation: 720 robot task executions were video recorded, resulting from querying 180 trained task models on 4 starting configurations. For each video, 5 Amazon Mechanical Turk [3] workers were asked to provide a rating (between 0 to 3) alongside the reasons for their rating if applicable (i.e., task completion failure, inefficient motion, and unsafe motion). The overall *rating* and *reasons* per video were calculated by taking the median of the responses per video.

3 Performance Across Generalization Scenarios

We study how average rating varies across two independent variables: (1) generalization scenario, and (2) learning method, in each task. For each test condition, the average rating (computed over 9 videos) is visualized in the bar charts in Fig.2. The symbols above each bar indicate the reasons for lower ratings if reported for at least 3 of the 9 videos. Throughout our evaluations in this section, we draw comparisons between algorithms only if there is a statistically significant difference ($p < 0.05$) in their mean performances given by two-way ANOVA test [21] followed by a Tukey's post test [7].

Reaching Task: For starting position S1, all methods except TLGC received near perfect ratings. As the robot starting position was moved farther away from the target object, i.e. S1 through S3, both TLGC and CLFDM exhibited a steady decline in performance, while ProMP and TpGMM exhibit much less significant affect. CLFDM, in particular, received the lowest possible rating in S3. On the other hand, when the distance to target was significantly reduced (S4), both TLGC and CLFDM performed above the minimum acceptable level of 2, while ProMP and TpGMM stayed below this threshold.

Overall, ProMP and TpGMM performed more consistently and above the minimum acceptable performance threshold than TLGC and CLFDM, as long as the starting position stayed a certain minimum distance away from the target object. CLFDM was often found to be inefficient by

[2]The pressing task was trained with two variants (detailed in the following section)

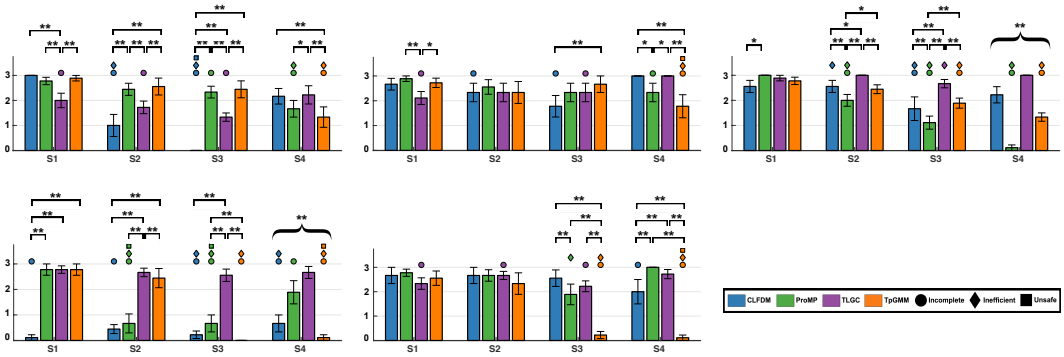

Figure 2: Bar plots of average user ratings for *reaching* (top-left), *pushing*(top-center), *writing*(top-right), *pressing* without segmentation (bottom-left), and pressing with segmentation (bottom-center). The legend is shown in bottom-right. $^{*}p < 0.05$, $^{**}p < 0.01$.

evaluators on S2 or unsafe due to collision with the table for S3. TLGC was consistently found to stop a small distance away from the target position, which was penalized on most occasions by evaluators.

Pushing Task: For the pushing task, no statistically significant difference in performance of algorithms was found ($p > 0.05$) across all conditions. However, CLFDM for S3, and TpGMM for S4, on average, performed below the minimum acceptable level. CLFDM and TLGC received perfect ratings when starting close to the target object S4.

In summary, ProMP and TLGC consistently remained above the acceptable threshold for all conditions. However, at least one failure case was observed each where the robot failed to reach the object. Moreover, unlike *reaching*, ProMP did not show a decline in performance for S4 here. TLGC, similar to *reaching*, was often penalized for failing to close the box fully, except for S4. Further, TpGMM was perceived as inefficient and unsafe when starting from S4 when the robot acted in an unsafe manner including a few occasions in which the robot dragged the box out of its mounting.

Writing Task: For S1, starting close to the demonstration mean, all the methods were rated highly, with CLFDM marginally performing worse than the rest. Increase in distance to the target object, i.e. S1 through S3, correlated with a performance decline for ProMP and TpGMM, with ProMP exhibiting a steeper decline than TpGMM. For starting position closer to the object than S1, i.e. for S4, the performance of ProMP and TpGMM were severely affected, while CLFDM and TLGC did not exhibit a significant difference in performance relative to S1.

Overall, TLGC was the only algorithm in this task to consistently perform on average in an acceptable manner and better than other methods. Although TLGC often produced extraneous motions for S3, casting doubts on the predictability of its outcome, the motions were mostly smooth. On the other hand, ProMP was often observed to draw non-smooth curves which were sometimes illegible. CLFDM, for S3 in particular, mostly occasions reproduced longer L-shaped curve instead of the desired S-shape, while TpGMM often drew sharper edges. For S4, both TpGMM and ProMP were frequently observed to go back a short distance before drawing a shape.

Pressing Task: Unlike the other tasks, the pressing motion can require certain methods to run a trajectory segmentation [5] routine before training. This a true in particular for CLFDM, which otherwise can not learn self-intersecting motions [12]. Hence, we evaluate performance for two training routines on pressing task, first without any segmentation, and second, with a perfect segmentation procedure.

Without task segmentation, as expected, CLFDM was found to consistently perform significantly below the minimum acceptable level. For other methods, the executions were highly rated when generated from closer to the mean of starting positions of demonstrations (S1). However unexpectedly, ProMP exhibited a sharp performance decline for S2 and S3 relative to S1. On the other hand, while TpGMM showed acceptable performance for S2, further increase in distance dropped its average rating to the lowest possible level. Starting closer to the target object (S4) saw all the methods except TLGC performing on average in an unacceptable manner.

With segmentation, the performance of CLFDM was found to improve significantly compared to that without segmentation. However, unexpectedly ProMP also showed a significant improvement in

performance for S2, S3 and S4. A possible reason for this could be that ProMP was able to better fit the skill for smaller durations since it approximates it with a finite set of bases [17]. While we found no statistically significant difference in the performance of algorithms for S1 or S2, notable differences emerged for S3 and S4. Similar to the unsegmented case, TpGMM performed significantly below acceptable level and worse than other methods for both these cases.

To summarize, irrespective of segmentation, TLGC was the only method which consistently performed in an acceptable manner across all scenarios. Often, however, TLGC would not fully pressing down the pegs which possibly prevented it from getting perfect ratings. When ProMP was rated low, it was also observed to be either colliding with the object, carrying out jerky and/or extraneous motions, and/or pressing either one or none of the pegs down. TpGMM, particularly for S3 and S4, was frequently observed to carry out extraneous motions while often failing to press any of the pegs.

4 Performance Across Experience Level

We present an analysis on the dependence of evaluator rating, averaged over all tasks and starting positions, on the experience level of the demonstrators (Fig. 3). By way of a two-way ANOVA analysis followed by Tukey's range test, we found that irrespective of the learning approach, there was a statistically significant difference in algorithm performance between low and high experience levels ($p < 0.05$). However no statistically significant difference in performance was found for low and medium, or medium and high experience levels. Carrying out a secondary analysis on the reasons provided by evaluators showed that robot executions from models learned from low experienced participants were statistically more likely to be marked inefficient on average than those from highly experienced participants ($p < 0.05$).

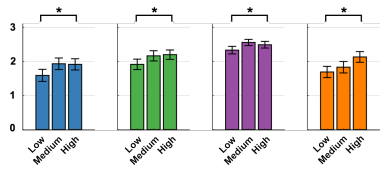

Figure 3: Average rating, by algorithm, averaged over experience level of demonstrator.

In conclusion, we see that experience level of the demonstrators affects performance across all algorithmic conditions. Interestingly, little difference in performance is observed between participants in the High condition and participants in the Medium condition, who had robotics experience but no kinesthetic teaching experience. This indicates that prior knowledge of robots, sensing, or sensitivity of many robotic systems to noise is potentially more important in predicting performance than experience with kinesthetic teaching itself. This insight could help guide future work on developing training guidelines for novice robot users that could help those with Low experience more quickly reach Medium or High performance.

5 Discussion and Conclusions

In this work, we have presented a large scale evaluation of four skill learning approaches across four real-world tasks. We found that there is no single technique among the ones tested that was consistent in performance across all types of tasks. TLGC delivered better performance on tasks with strong constraints in direction of motion, e.g., *writing*. On the other hand, ProMP and TpGMM performed better on tasks with just positional constraints, e.g. *reaching*. A possible reason could be that TLGC's trajectory generation routine depends explicitly on matching curvature of demonstrations, while TpGMM/ProMP seek to minimize the positional mismatch at certain time instances. Furthermore, we found that no one algorithm could guarantee successful execution on a given task across all starting configurations. Failure cases, inefficient behaviors, and unsafe trajectories were observed across many conditions, which poses a problem for robot deployment in real-world settings. This can be attributed to the well-known covariate shift problem in machine learning [10]. An important aspect here is that covariate shift problem poses itself more often for learning approaches that parametrize the skill based on time, i.e. ProMP and TpGMM, which often generalized worse than CLFDM and TLGC when starting closer to the target object.

Our findings also show that the performance of all four algorithms is impacted by the experience level of the user, even when users are instructed in teaching and have opportunities to practice. Further analysis of the causes of this effect should be performed, with the goal of identifying mitigation strategies that potentially improve performance. We hope that insights from this research would guide future research toward developing more robust approaches capable of handling a variety of tasks that a robot can encounter in the real world.

References

[1] Seyed Reza Ahmadzadeh, Muhammad Asif Rana, and Sonia Chernova. Generalized cylinders for learning, reproduction, generalization, and refinement of robot skills. In *Robotics: Science and systems*, volume 1, 2017.

[2] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[3] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[4] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.

[5] Sylvain Calinon, Darwin G Caldwell, and Aude Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. Citeseer.

[6] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(2):286–298, 2007.

[7] David B Duncan. Multiple range and multiple f tests. *Biometrics*, 11(1):1–42, 1955.

[8] Yanlong Huang, Leonel Rozo, João Silvério, and Darwin G Caldwell. Kernelized movement primitives. *arXiv preprint arXiv:1708.08638*, 2017.

[9] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[10] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3:1–12, 2008.

[11] S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[12] S Mohammad Khansari-Zadeh and Aude Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

[13] Andre Lemme, Yaron Meirovitch, Seyed Mohammad Khansari-Zadeh, Tamar Flash, Aude Billard, and Jochen J Steil. Open-source benchmarking for learned reaching motion generation in robotics. *Paladyn, Journal of Behavioral Robotics*, 6(1), 2015.

[14] Yaron Meirovitch, Daniel Bennequin, and Tamar Flash. Geometrical invariance and smoothness maximization for task-space movement generation. *IEEE Transactions on Robotics*, 32(4):837–853, 2016.

[15] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[16] Thomas Nierhoff, Sandra Hirche, and Yoshihiko Nakamura. Spatial adaption of robot trajectories based on laplacian trajectory editing. *Autonomous Robots*, 40(1):159–173, 2016.

[17] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in neural information processing systems*, pages 2616–2624, 2013.

[18] Muhammad Asif Rana, Mustafa Mukadam, S Reza Ahmadzadeh, Sonia Chernova, and Byron Boots. Towards robust skill generalization: Unifying learning from demonstration and motion planning. In *Conference on Robot Learning*, pages 109–118, 2017.

[19] Harish Ravichandar and Ashwin Dani. Learning position and orientation dynamics from demonstrations via contraction analysis. *Autonomous Robots*, pages 1–16, 2018.

[20] Markus Schneider and Wolfgang Ertel. Robot learning by demonstration with local gaussian process regression. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 255–260. IEEE, 2010.

[21] Lars St, Svante Wold, et al. Analysis of variance (anova). *Chemometrics and intelligent laboratory systems*, 6(4):259–272, 1989.