2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Benchmark for Skill Learning from Demonstration: Impact of User Experience, Task Complexity, and Start Configuration on Performance

M. Asif Rana[1], Daphne Chen[1], Jacob Williams[1], Vivian Chu[1], S. Reza Ahmadzadeh[2], and Sonia Chernova[1]

***Abstract*— We contribute a study benchmarking the performance of multiple motion-based learning from demonstration approaches. Given the number and diversity of existing methods, it is critical that comprehensive empirical studies be performed comparing the relative strengths of these techniques. In particular, we evaluate four approaches based on properties an end user may desire for real-world tasks. To perform this evaluation, we collected data from nine participants, across four manipulation tasks. The resulting demonstrations were used to train 180 task models and evaluated on 720 task reproductions on a physical robot. Our results detail how i) complexity of the task, ii) the expertise of the human demonstrator, and iii) the starting configuration of the robot affect task performance. The collected dataset of demonstrations, robot executions, and evaluations are publicly available. Research insights and guidelines are also provided to guide future research and deployment choices about these approaches.**

I. Introduction

To generate motions required to accomplish a variety of tasks, it is beneficial for robots to have the capability to learn new skills. Learning from demonstration (LfD) [1] provides a viable channel to acquire such skills via human interaction. In a typical LfD setting, a human end user provides trajectory demonstrations from multiple starting locations. A model is then trained over the demonstrations, which can be queried from novel starting positions to generate new robot motions.

From the perspective of an end user, there are multiple desirable properties that a skill learning approach should have, including the ability to:

1) learn skills from demonstrations provided by end users irrespective of all experience levels, with minimal information overload on the user,
2) learn a variety of skills, which may differ in the level of complexity, and
3) reproduce a learned skill in scenarios similar to or different from those encountered during demonstrations.

Given the number and diversity of existing LfD approaches, it is critical that empirical studies be performed to compare the relative strengths of these learning techniques. However, comprehensive evaluation of these approaches based on the criteria mentioned above does not exist to date.

In this work, we evaluate the performance of multiple skill learning approaches and examine how the i) complexity of the task, ii) expertise level of the human demonstrator, and iii) starting configuration of the robot affect performance of each technique. To perform this evaluation, we collected data from nine participants across four different manipulation tasks with varying starting conditions. The resulting demonstrations were used to train 180 task models. Each of the resulting models was then executed on a Rethink Sawyer robot, resulting in 720 videos of robot task reproductions. Finally, we obtained 3600 Amazon Mechanical Turk ratings to evaluate the robot's performance in the videos. Figure 1 provides an overview of our experimental procedure. Additionally, we present an evaluation based on quantitative error metrics obtained by assessing the similarity between the reproduced trajectories and the demonstrations.

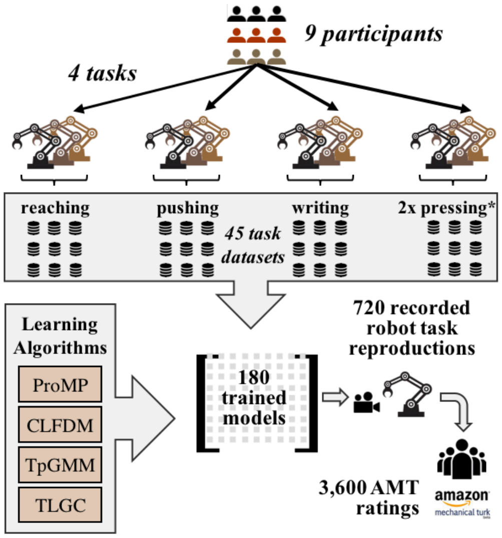


Fig. 1: Overview of the experimental design.

Our results show that the performance of the skill learning approaches — irrespective of their underlying representation — is generally predictable when the new starting condition is closer to the starting position of demonstrations. However, as the generalization scenario differs from the demonstrations, the consistency of an approach's performance across generalization scenarios is highly dependent on the task constraints. Furthermore, we also find that the performance of a given skill learning method is correlated with the experience level of the human providing demonstrations. Lastly, we found that commonly used performance evaluation metrics such as mean squared error are not always able to correctly predict the generalization performance of an approach.

The authors intend for this work to be used by those who study LfD by acting as a reference for experimental design, evaluation metrics, and general best practices. The full dataset of demonstrations, videos of executions, and accompanying evaluations have been made publicly available

[1] Georgia Inst. of Technology, Atlanta, GA. [2] University of Massachusetts Lowell, Lowell, MA. Email for correspondence: asif.rana@gatech.edu

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

to aid future benchmarking efforts[1].

II. Related Work

In this section, we present an overview of motion-based LfD and describe the techniques examined in our study.

A. Overview of Motion-Based LfD

There exist several approaches aimed at learning skills from human demonstrations. While some approaches are aimed at learning reactive (or time-invariant) motions [2]–[6], others learn time-dependent [7]–[10], or sometimes arc-length parameterized movements [11], [12]. Based on the choice of motion generation strategy, the majority of skill learning approaches can be further broadly categorized into four groups: time-invariant (*stable*) dynamical systems [2], [3], [13], time-dependent dynamical systems [7], [9], [10], time-dependent statistical methods [8], [14], [15], and arc-length based geometry preserving techniques [12], [16], [17].

In literature however, these approaches are usually tested in isolation by experts for a specific set of tasks. While the relative advantages and disadvantages of the commonly-used approaches might be known within the LfD community, there do not exist comprehensive guidelines for non-experts outside the community to assist in using these methods. Comprehensive surveys on LfD [1], [18]–[20] do exist, but they mainly focus on summarizing existing LfD approaches, proposing taxonomy, and reporting challenges associated with employing LfD approaches in practice. There is a need to supplement these surveys by comparing and evaluating LfD approaches across several variables that can be encountered in the real world.

Prior work by Lemme *et al.* [21] contributed a valuable benchmarking framework to evaluate the performance of reaching motion generation approaches on a 2D handwriting dataset. Their study evaluates the algorithms' generalization ability in simulation and presents performance metrics on a small scale. Our study is more comprehensive: it covers multiple tasks, incorporates diverse constraints and variables, and is performed on a physical robot. To our knowledge, no benchmarking study exists that independently evaluates a wide range of task execution conditions. Additionally, no prior studies report human ratings of task performance.

B. Techniques Selected for Comparison

The approaches evaluated in this work were chosen to represent the four categories mentioned in the previous subsection. While many other LfD approaches exist, we have selected these four approaches because they are well-known, commonly used, or are most mature based on incremental improvement on prior work. Below, we provide a brief description of each method; see references for full details.

CLFDM [3] – A time-invariant approach which learns a dynamical system mapping positions to velocities. It is assumed here that the final positions of the demonstrated motions are centered at a single goal location, and the dynamical system rollouts are guaranteed to converge at the goal.

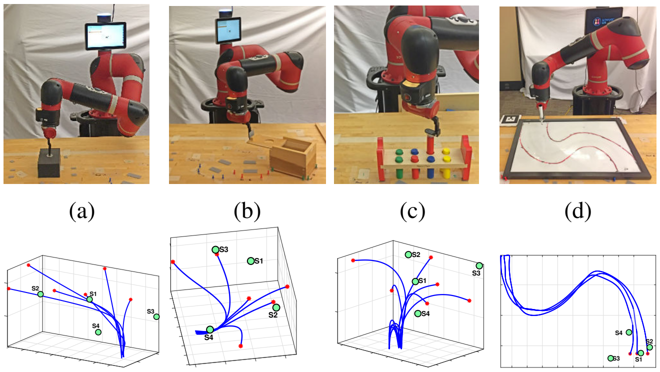

(a)
(b)
(c)
(d)

Fig. 2: From left to right, visualizations of the *reaching*, *pushing*, *pressing*, and *writing* task. The bottom row plots example demonstrations (blue) for each tasks. The red dots denote demonstrated starting positions while the green circles represent *new* initial positions selected for evaluating skill generalization.

ProMP [7] – A time-dependent approach which describes demonstrations as a set of weights. ProMP derives a stochastic control law as a function of these weights, rolling out which reproduces a distribution over trajectories.

TpGMM [14] – An approach which encodes the statistical features of demonstrations as a joint probability density over time and states. The version of TpGMM employed in this work finds new trajectories by solving an LQR problem, resulting in a control law which tracks the demonstration mean with a time-varying gain.

TLGC [11] – An approach which encodes the geometric features of the demonstrations. TLGC bounds demonstrations by an arc-length parameterized *generalized cylinder*. Given a new initial position, a new trajectory is found by tracking the curvature of the nearest boundary of the cylinder.

III. Experimental Design

This section provides an overview of our experimental design process, including the choice of tasks, human participant selection, as well as methodology for data recording and model evaluation.

A. Robot Tasks

We selected four tasks (Fig. 2) each of which contains unique properties representing different level of position and motion constraint complexity. Human demonstrator ability was kept in mind such that users with minimal experience could demonstrate the task on the robot.

- *Reaching* (Fig. 2a)- Move toward and touch the circle on the gray block . This task poses a hard constraint on the end position.
- *Pushing* (Fig. 2b)- Push the box lid closed. Comparing to the previous task, this task is constrained in the direction of motion towards the end. The position constraint for the endpoint is not as hard as in the *reaching* task.
- *Pressing* (Fig. 2c)- Push down peg #1 and then peg #2 . Compared to *pushing*, this task is more constrained in both the direction of motion and end-positions.
- *Writing* (Fig. 2d)- Draw an S-shaped curve on the whiteboard . Compared to other tasks, this task requires a harder constraint on the direction of motion to follow the curvature of the shape.

[1] https://sites.google.com/view/rail-lfd

B. Participant Selection

We recruited nine participants with different levels of robotics experience. Three participants with *Low* experience had no prior interaction with a robot. Three participants with *Medium* experience had worked with robots but had no experience in robot manipulation, and particularly no experience in kinesthetic teaching. Three participants with *High* experience had some prior experience with LfD and kinesthetic teaching.

C. Data Recording

Data collection with participants followed an IRB-approved human subjects study protocol. Upon arrival, participants were briefed about the goals of the study and taught to interact with the robot using a practice task.

In order to ensure consistency, participants received written instructions that included a verbal description and photos of the goals of each task. During the recording, the robot was first initialized to a preset starting configuration. The participant kinesthetically guided the robot to accomplish the task. In order to assess the quality of the demonstrations, we provided the participants with a visualization of the recorded trajectory in ROS RViz. Participants were allowed to retry the task until they were satisfied that they met the goal. In total, participants provided three demonstrations for the *writing* task with three different starting positions. For the remaining tasks, six demonstrations (3 starting positions × 2 object locations) per participant were collected. This resulted a total of 21 demonstrations per participant. Fig. 2 (*bottom*) shows an example set of demonstrations.

D. Model Evaluation

From the collected demonstrations, we constructed 45 task datasets. Each dataset includes all demonstrations of a specific task (four tasks) performed by a specific participant (nine participants). Note that each participant was asked to demonstrate the *pressing* task twice each time under a different condition (see Section IV-B for more detail), resulting in 9 participants × 5 tasks = 45 datasets.

Each of our four algorithms was then trained on each of the 45 datasets, resulting in 180 task models (one per participant-task-algorithm combination). For evaluation, we executed each of the 180 models under four different starting conditions on a Sawyer robot, resulting in 720 video recordings of robot task executions. To obtain a final evaluation of the robot's performance in each of the videos, we employed five AMT [22] workers to evaluate the quality of each video, resulting in approximately 3600 performance ratings.

E. Amazon Mechanical Turk Evaluation

To ensure that AMT workers evaluating the robot had a consistent understanding of the task goals, workers were shown the same set of instructions as those given to the study participants (i.e., task demonstrators). For each video of the robot's task execution, AMT workers were asked to answer the following questions:

Q1. Please rate the extent to which you agree with the statement: *"The robot efficiently and safely completed the goal(s) of the task."* (Strongly agree; Agree; Disagree; Strongly disagree).

Q2. Please also specify which of the following contributed to your rating in the previous question. (Check all that apply)

- The robot failed to achieve the goals of the task (incomplete).
- The robot performed unnecessary motion (inefficient).
- The robot acted in an unsafe manner (unsafe).

Each video was evaluated by five AMT workers and an overall *rating* per video/execution was calculated by taking the median of the responses to the first question. To get a quantitative measure of the evaluator rating, we mapped the answers to numerical values: Strongly agree = 3, Agree = 2, Disagree = 1, and Strongly disagree = 0. We consider a task reproduction to be acceptable to the evaluators if the rating is 2 or above. Answers to the second question were only considered if the participant selected a rating below "Strongly agree" in response to the first question.

The selected keywords, *incomplete*, *inefficient*, and *unsafe*, are suitable to define the characteristics of the task execution quality from an end user's point of view. Our reasoning is that a robot that cannot complete a task efficiently can impose great burden on the user, and a successful human-robot team requires a smooth and predictable task execution.

IV. Data Processing and Validation Scenarios

This section provides an overview of the data processing and parameter tuning methods used in our evaluation, as well as the design of the starting robot configurations used in evaluating the generalization of the chosen approaches.

A. Data Preprocessing

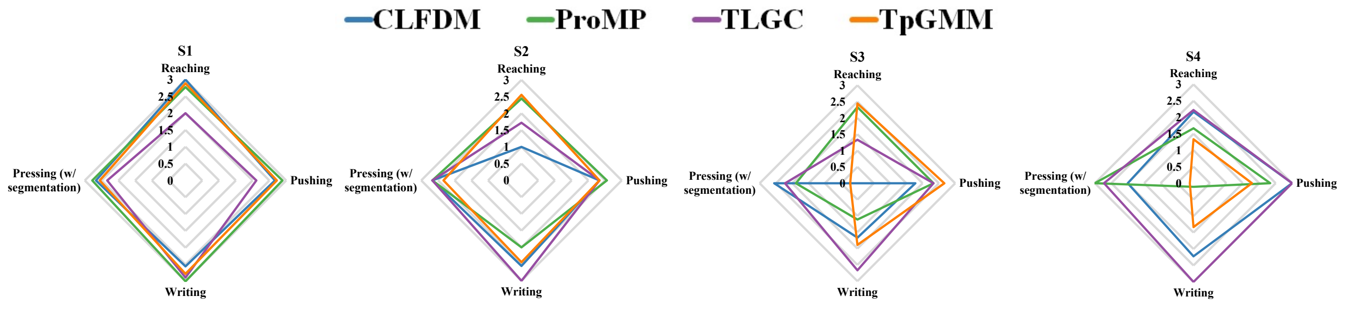
Captured human demonstrations consist of robot end-effector poses over time. First, we applied a low-pass moving average filter to remove high-frequency noise from the raw data. Additionally, we estimated the velocities of the end-effector using 1st-order finite differencing. Finally, for methods that require time-aligned trajectories, we also warped the demonstrations to be of the same time duration using dynamic time warping (DTW) [23].

B. Motion Segmentation for Pressing Task

Since the *pressing* task includes two consecutive pressing primitives, we conducted experiments of the task once without and once with motion segmentation [24]–[27]. Specifically, we created another variation of the task dataset by passing the demonstrations through a motion segmentation routine [28], which divided each trajectory into two segments. For a given approach, we trained a model per segment, reproduced the task segments separately, and stitched the reproduced segments together to be executed by the robot as one trajectory. Throughout the paper we clarify which variant of *pressing* is being used, and we evaluate the effect of segmentation on performance in Section V-C.

C. Parameter Tuning

To mimic a realistic operational context for the robot, we chose to use only a single common set of parameters for each algorithm. More specifically, we tuned a parameter set for each *algorithm*, but did not tune unique parameters *per task*,



(a) (b) (c) (d)

Fig. 3: Radar plots reporting average ratings against executed tasks, for a particular starting position. The average is computed over nine datapoints corresponding to the nine recorded videos, where each video represents a model query at the given generalization scenario.

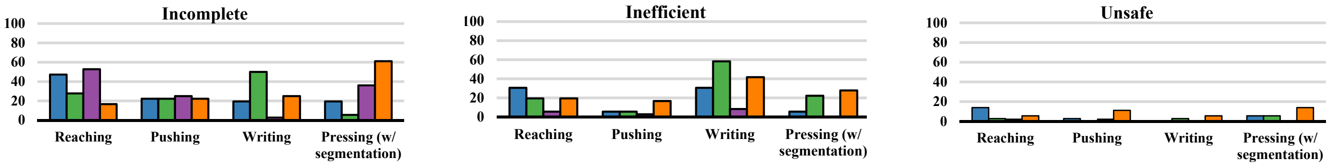


Fig. 4: Bar charts reporting subjective user feedback, whereby each bar represents the number of times a given reason was cited – as a percentage of the total number of robot executions for a given task-algorithm combination.

since this would be impractical in real-world settings. We performed the tuning process on the LASA dataset [29] and a small randomly selected subset of robot demonstrations. We manually tuned the parameters of each method until we observed consistently good performance across the test set.

D. Starting Positions for Generalization

Each task model was evaluated from four different initial configurations, $S1$-$S4$, to validate the generalizability of the learned models. Figure 2 (bottom row) visualizes the initial positions for each task. S1 was selected to be within 90% confidence interval around the mean of the initial positions of the demonstrations. S2-S4 were selected outside this range, such that $d(S3) > d(S2) > d(S1) > d(S4)$, where $d(.)$ denotes the Euclidean distance to the target object. S2 and S3 were chosen to be farther away from the target object, while S4 was chosen to be closer to the object.

V. Generalization Performance across Starting Positions and Tasks

In this section, we study how the average rating for each skill learning method varies across two independent variables: (1) starting position and (2) task.

A. Trends across starting positions

We see larger variations in average performance of approaches across tasks when the distance between the robot's starting position and the target location is progressively increased (S1 through S3), as shown in Fig. 3a through 3d. In general, as evident from Fig. 7(*top*), we noticed worsening performance with increasing distance. This is particularly noticeable for the *writing* and *reaching* tasks in 3b and 3c. However, when the target distance is significantly decreased (S4), CLFDM and TLGC performed consistently in an acceptable manner across tasks, while ProMP and TpGMM generally under-performed. Overall, TLGC was least affected by the changes in starting positions for the *pushing*, *writing*, and *pressing* tasks. However, on the *reaching* task, where the other approaches performed generally well, TLGC performed the worst and often at an unacceptable level.

B. Task-wise evaluation and subjective user feedback

Analysis in this subsection is based on Fig. 7 (*bottom*) in conjunction with subjective user feedback from Fig. 4. The video accompanying this paper shows some of the failure/success cases mentioned here.

For the *reaching* task, TLGC is hypothesized to have accrued low ratings due to robot executions which often stopped a short distance from the target. Users often marked these executions as *incomplete*. CLFDM was found to not generalize well for starting positions S2 and S3 which are farther from the target, and had a high percentage *incomplete*, *inefficient*, and sometimes *unsafe* ratings. We hypothesize that this is due to often long, unpredictable paths generated by CLFDM. Furthermore, due to this unpredictability, the robot often collided with the table and hence failed to complete the task; thus the evaluators frequently marked the executions as *incomplete* and *unsafe*.

On *pushing*, although all approaches were consistent on average across starting positions, we noticed several failure cases. TpGMM was sometimes perceived as *inefficient* and *unsafe* when starting too far from (S3) or too close to the box (S4). During some of these executions, the robot pushed farther than necessary into the box and dismounted it.

For *writing*, only TLGC generalized across starting positions. CLFDM was observed to be the second most consistent across starting positions, except when starting away from the final position (S3). CLFDM often drew a longer L-shaped curve instead of the desired S-shape, which was marked as *inefficient* and *incomplete* although it was mostly smooth and safe. Executions by ProMP were frequently marked as *incomplete* and *inefficient* since it was often observed to draw non-smooth curves when starting farther away, i.e. S2-S3

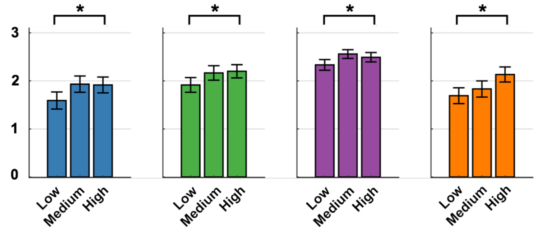


Fig. 6: Average ratings grouped by algorithm (CLFDM, ProMP, TLGC, TpGMM) over the experience level of the demonstrators.

or illegible shapes when starting closer (S4). When starting from S3, TpGMM was also found to draw an S-shaped curve with relatively sharp edges. Lastly, for both TpGMM and ProMP, the robot was frequently observed to go back a short distance from S4 before drawing, often penalized by evaluators for being *inefficient*.

For the *pressing* task, TpGMM was severely affected by variations in starting positions. TpGMM frequently carried out extraneous motions for S3 and S4, often failing to press any of the pegs. Moreover, TpGMM occasionally followed a pressing motion but stayed above the pegs. Such executions were often rated *incomplete* and *inefficient*. ProMP was sometimes marked *inefficient*, which can be attributed to jerky, extraneous motions when started far from the pegs.

C. Effect of motion segmentation

We conducted an additional evaluation to test our hypothesis regarding the adverse effect of learning on unsegmented data on the *pressing* task's performance. We trained each algorithm on unsegmented data and performed the same crowdsourced rating in Section III-E. Fig. 5 shows a bar chart comparing performance with vs. without segmentation. Each bar shows the average rating *without* motion segmentation subtracted from the average rating *with* the segmentation routine. We observed that ProMP, and especially CLFDM, suffer significantly when segmentation is not used. This is an expected result for CLFDM, which is known to be incapable of learning self-intersecting motions [3]. This behavior is in fact expected for all LfD approaches which learn first-order dynamical systems from demonstrations [2], [4]–[6].

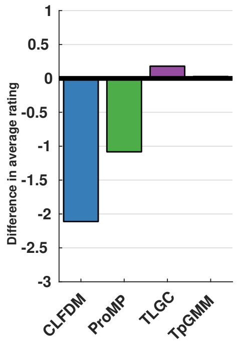


Fig. 5: Difference in ratings on the *pressing* task w/ and w/o segmentation.

VI. Performance Across Experience Level

In this section, we present an analysis on the dependence of the evaluator ratings, averaged over all the tasks and starting positions, on the experience level of the demonstrators. Fig. 6 provides a visualization of the results.

All the methods show similar increase in average rating from low to high experience when each algorithm is individually observed across experience levels. To corroborate this trend, we also carried out a two-way ANOVA analysis for the approaches against the experience levels. We found that the experience level has a statistically significant effect on average ratings ($p = 0.0389 < 0.05$), while no statistically significant interaction effect was found between the two variables ($p = 0.95 > 0.05$). We further carried out Tukey's range test, which determined that there was a statistically significant effect on performance between the 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. A secondary analysis was also carried out on the reasons the evaluators provided for their ratings. This showed that there was a statistically significant difference between user experience levels low and high ($p < 0.05$) for a video being marked as inefficient. This means that the evaluators considered the lower-rated videos corresponding to the low experience demonstrators to be more inefficient on average.

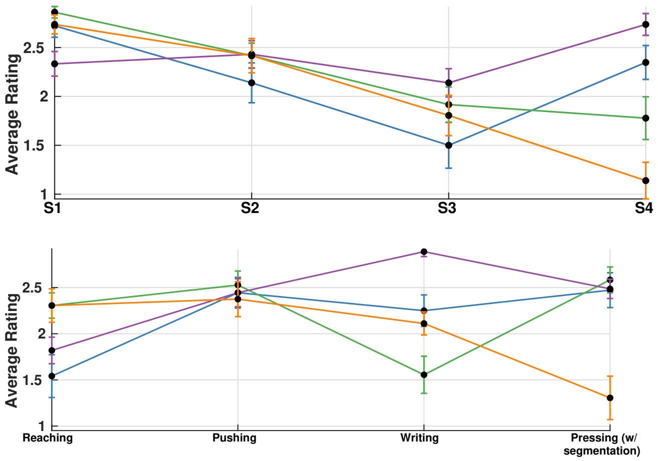


Fig. 7: Plot of average user rating against *top*: starting positions and *bottom*: tasks.

In conclusion, we see that higher demonstrator experience positively affects performance across all algorithmic conditions. Interestingly, little difference in performance is observed between participants with high and medium levels of experience (participants with kinesthetic teaching experience vs. participant with general robotics experience). This observation indicates that having prior knowledge about robots, sensing, or sensitivity to noise is potentially more important than having specific experience with kinesthetic teaching. This insight could direct future work on developing training guidelines to quickly increase novices' expertise. Additionally, an extension may study whether providing supplementary directions (e.g. about speed, waypoints, and direction of motion) to novice users beyond the baseline instruction improves overall performance.

VII. Quantitative Metric Evaluations

While the evaluations reported thus far are based on a qualitative measure of performance, existing LfD literature employs quantitative metrics for this purpose. A widely-used metric is the mean squared error (MSE) [21], which measures the deviation of reproduced motions from demonstrated trajectories. We examine whether there is a correlation between

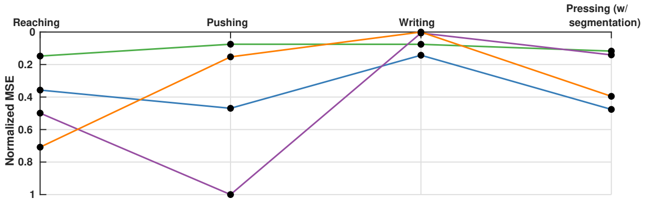


Fig. 8: Normalized mean squared error for different algorithms across all tasks. Note that the vertical axis direction is flipped.

the MSE and the evaluator ratings.

Figure 8 reports the MSE scores, averaged over starting positions and demonstrators, plotted against the tasks. The vertical axis represents MSE scores, normalized to lie in the range 0 to 1. Note that the direction of the vertical axis for MSE scores has been reversed such that moving up the vertical axis corresponds to improvement in performance in terms of MSE. To compare against the user ratings, we make use of the average user ratings against the tasks plotted in Fig. 7(*bottom*). For each task, we ranked the approaches in terms of the MSE scores and the user ratings respectively and compared the two rankings.

Overall, despite a common assumption to the contrary, we observe that MSE is not an accurate predictor of generalization performance of a skill learning approach. This is particularly evident for the *writing* task. For this task, the AMT users were observed to care more about the shape of the executed motion as opposed to its position profile. However, MSE only measures deviations in positions from the demonstrations. Hence, while all the approaches were predicted to perform well according to MSE, only TLGC was able to draw an S-shape curve on most occasions and hence get high ratings. Furthermore, we also observe that MSE gives little information about the capability of a model to achieve the task goals. In particular, for the *pushing* task, we see that all the approaches were rated highly since they mostly achieved the goal of closing the box towards the end of execution. The users were observed to care less about the trajectory while approaching the box. However, MSE considers the entire length of the trajectories, therefore approaches that fit the data better received higher scores.

VIII. Conclusions and Discussion

We have presented a large-scale evaluation of four skill learning approaches across four real-world tasks. Our conclusions are based on 720 robot task executions and 3600 ratings provided by AMT users who evaluated the robot trajectories on safety, efficiency, and success in achieving the task goals. Here, we share algorithm-specific observations to guide users in selecting the appropriate method for their use case.

A. Algorithmic Observations

For those planning to use a dynamics-based approach such as CLFDM, it may be useful to note that while such methods guarantee reaching a target location, they cannot ensure *safety* or *efficiency* of executions. However, both these factors have great real-world significance, as noted by evaluators who rated CLFDM on *reaching* and *writing*. CLFDM is also more sensitive to changes in distance from the target. Segmenting the task can mitigate this, particularly for tasks with a strong position and direction-of-motion constraint.

Time-parameterized approaches, e.g. ProMP and TpGMM, can be suitable on tasks which impose minimal direction-of-motion constraint alongside end-position constraint (e.g., *pushing* and *reaching*). However, starting very close to the goal can immensely affect performance. This is because time-parameterized approaches are not robust to large spatio-temporal perturbations. One should take care to ensure the robot does not start too close to the final position unless a majority of the provided demonstrations are in the vicinity of this desired starting position.

For tasks with strong constraints in the direction of motion, a geometric approach like TLGC can be more suitable. We conclude this by observing the consistency of TLGC's performance on the *writing* and *pressing* tasks. This is primarily because TLGC explicitly encodes the shape of the demonstrated motions and minimizes deviations from this shape during reproduction.

B. Research Insights

This subsection provides general, algorithm-independent research insights learned from this benchmarking effort. We hope this knowledge will guide researchers in developing more robust techniques.

- Approaches with different model representation perform differently on tasks with various constraints. Our evaluations suggest that none of the approaches worked well across every task. While TLGC, the approach with a geometric representation, worked well for tasks with strong constraints in the direction of motion (e.g., *writing*), ProMP, with a time-parameterized probabilistic representation, was found to be most consistent on tasks with positional (e.g., goal location) constraints.
- Generalization quality decreases as the new starting positions go farther from the original starting positions. None of the approaches were able to consistently generalize to such starting positions.
- Task complexity affects the approaches' generalization ability. Our results show that algorithms generalize better for tasks with simpler constraints, usually struggling on tasks with directional and positional constraints.
- For long-horizon tasks with multiple position constraints (e.g. via-points) alongside constraints on the direction of motion, motion segmentation can be beneficial.
- Higher user experience level positively impacts the performance of the approaches. Our findings also show that algorithm performance is affected by the *quality* of demonstrations from users with varying experience.
- Conventional metrics may not be good predictors of performance. We have found that the quantitative MSE does not reliably predict performance across many tasks.

IX. Acknowledgements

The authors thank Harish Ravichandar for contributing insightful comments on this study. This work was supported in part by NSF IIS 1564080 and ONR N000141612835.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[3] S. M. Khansarizadeh and A. Billard, "Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[4] N. Perrin and P. Schlehuber-Caissier, "Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems," *Systems & Control Letters*, vol. 96, pp. 51–59, 2016.

[5] H. C. Ravichandar, I. Salehi, and A. P. Dani, "Learning partially contracting dynamical systems from demonstrations." in *Conference on Robot Learning*, 2017, pp. 369–378.

[6] K. Neumann and J. J. Steil, "Learning robot motions with stable dynamical systems under diffeomorphic transformations," *Robotics and Autonomous Systems*, vol. 70, pp. 1–15, 2015.

[7] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in neural information processing systems*, 2013, pp. 2616–2624.

[8] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Trans. on Systems, Man, and Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[9] M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, "Towards robust skill generalization: Unifying learning from demonstration and motion planning," in *Conference on Robot Learning*, 2017, pp. 109–118.

[10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[11] S. R. Ahmadzadeh and S. Chernova, "Trajectory-based skill learning using generalized cylinders," *Frontiers in Robotics and AI*, vol. 5, pp. 1–18, 2018.

[12] Y. Meirovitch, D. Bennequin, and T. Flash, "Geometrical invariance and smoothness maximization for task-space movement generation," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 837–853, 2016.

[13] H. Ravichandar and A. Dani, "Learning position and orientation dynamics from demonstrations via contraction analysis," *Autonomous Robots*, pp. 1–16, 2018.

[14] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[15] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *arXiv preprint arXiv:1708.08638*, 2017.

[16] T. Nierhoff, S. Hirche, and Y. Nakamura, "Spatial adaption of robot trajectories based on laplacian trajectory editing," *Autonomous Robots*, vol. 40, no. 1, pp. 159–173, 2016.

[17] S. R. Ahmadzadeh, M. A. Rana, and S. Chernova, "Generalized cylinders for learning, reproduction, generalization, and refinement of robot skills." in *Robotics: Science and systems*, vol. 1, 2017.

[18] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[19] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *arXiv preprint arXiv:1907.03146*, 2019.

[20] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 21, 2017.

[21] A. Lemme, Y. Meirovitch, S. M. Khansari-Zadeh, T. Flash, A. Billard, and J. J. Steil, "Open-source benchmarking for learned reaching motion generation in robotics," *Paladyn, Journal of Behavioral Robotics*, vol. 6, no. 1, 2015.

[22] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[23] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[24] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[25] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[26] O. Kroemer, C. Daniel, G. Neumann, H. Van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 1503–1510.

[27] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, "Movement segmentation using a primitive library," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3407–3412.

[28] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[29] S. M. Khansari-Zadeh, "Lasa handwriting dataset," https://bitbucket.org/khansari/lasahandwritingdataset.