Interactive Keyframe Learning (IKL): Learning Keyframes from a Single Demonstration of a Task

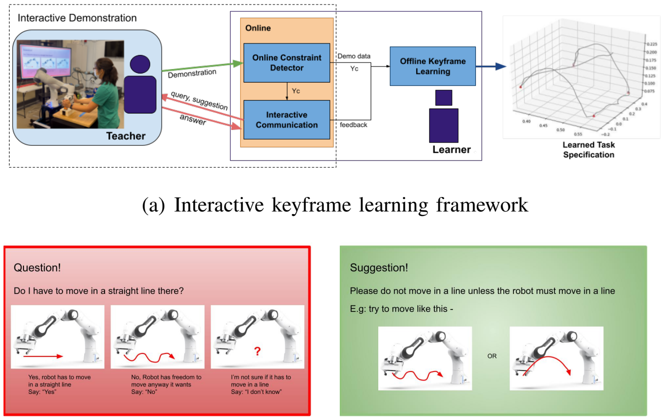
Thavishi Illandara, Julie A. Shah
Computer Science and Artificial Intelligence Laboratory (CSAIL),
Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
$\{thavishi, julie\_a\_shah\}$@$csail.mit.edu$

***Abstract*—Increasing the feasibility of integrating robots into industrial processes requires the robots' programming to be easily accessible to domain experts with little-to-no robotics experience. In this paper, we present Interactive Keyframe Learning (IKL), a method for learning a task specification as an ordered sequence of keyframes in order to capture physical interactions and geometric constraints from a single demonstration provided by a non-expert. IKL infers the human's intent for demonstrated constrained motion online and performs interaction- and constraint-based segmentation offline to reduce the nonessential learned keyframes arising from one-shot learning. Through results from a user study conducted in a real-world setting, we demonstrate the significant benefits of IKL for teaching tasks to robots.**

I. Introduction

Recent advances in robotics have enabled the automation of traditionally manual manufacturing tasks; however, integrating robots into a factory via conventional programming methods remains time- and resource-intensive, as even small changes to a given task require reprogramming and can result in high reintegration costs [12]. Learning from demonstration (LfD) has been explored as a potential solution to this problem, as it can enable domain experts with minimal programming experience to teach robots efficiently [12]. In this work, we focus on learning a high-level task plan from a task demonstration as an ordered sparse set of sequential poses called *keyframes*, a concept introduced by Akgun et al. [1], which captures the physical interactions and local geometric constraints of the task. Learning keyframes will allow the system to generate a compact, agent-independent representation of a multistep task. These representations can then be provided to an existing motion planner equipped with collision avoidance, allowing the planner to optimize for unique objectives, such as power or time efficiency.

Learning a sequenced task plan from a continuous demonstration requires trajectory segmentation. Pérez-D'Arpino and Shah [10] used end-effector poses explicitly defined by the user in order to learn keyframes and geometric constraints; however, in this approach, the quality of the learned sequence hinges upon the operator's knowledge of what constitutes an optimal set of keyframes. On the other hand, automated segmentation techniques can be relatively robust to user expertise and allow users to provide more intuitive trajectory demonstrations. Inspired by the method introduced by Fearnhead and Liu [4], prior work has utilized hidden Markov models (HMMs) and statistical model-based changepoint detection algorithms to perform online segmentation based on changes to specified models or latent variables [9]. However, such techniques are limited to inference over parametrized models, and are unable to recognize trajectory segments that cannot be modeled (such as arbitrary unconstrained motion). Some approaches perform interaction-based segmentation by learning action segments through task events or space characteristics [6]; however, modeling global and local task constraints, such as geometric constraints, is often essential to successful task execution, as demonstrated in prior research [10, 8].


Interactive Demonstration
Teacher
Demonstration
query, suggestion
answer
Online
Online Constraint Detector
Yc
Interactive Communication
Demo data
Yc
feedback
Offline Keyframe Learning
Learner
Learned Task Specification

(a) Interactive keyframe learning framework

Question!

Do I have to move in a straight line there?

Yes, robot has to move in a straight line
Say: "Yes"

No, Robot has freedom to move anyway it wants
Say: "No"

I'm not sure if it has to move in a line
Say: "I don't know"

(b) Question

Suggestion!

Please do not move in a line unless the robot must move in a line

E.g: try to move like this -

(c) Suggestion

Fig. 1. Interactive keyframe learning. Fig. 1(a) depicts our proposed interactive keyframe learning framework that learns a task specification (marked as red circles) from a single interactive demonstration. Fig. 1(b) illustrates the question posed to the user during the interactive demonstration, and Fig. 1(c) displays the suggestion provided.

Previous segmentation literature [10, 8] has explored learning from a single demonstration to mitigate the time- and resource-intensive nature of providing multiple demonstrations. For example, Liu et al. [8] learn articulated constraint segments represented as task space regions (TSRs) [2] from a single demonstration. However, the accuracy of the learned constraints depends on the assumption that the teacher will only demonstrate constrained motion when doing so is crucial to task success. To mitigate this, we leverage human feedback elicited during the demonstration to improve the accuracy of the constraints learned. Additionally, we enable learner feedback to the teacher as a suggestion in order to improve their teaching of unconstrained motion, taking a step toward improving the teacher's mental model of the learner.

In this paper, we propose Interactive Keyframe Learning (IKL), a proof-of-concept system for learning a task's physical interactions and move-in-line constraints from a single demonstration as an ordered sequence of keyframes encoded as TSRs. We introduce an interactive demonstration framework that allows a robot, through human-robot communication, to learn a human's intent for the demonstrated constrained motion and evaluate its impact on the teaching workload and ability to reduce the over-constrained nature of specifications that occur when learning from a single demonstration via a user study conducted in a real-world setting.

II. The Interactive Keyframe Learning Framework

A set of sequential keyframes can encode a multistep task to create an agent-independent task specification [10]; however, multiple specifications may exist for a given task. IKL aims to learn S_E, the specification consisting only of the keyframes essential for task success, with a set of keyframes considered essential if removing any of the keyframes results in an erroneous task specification. We limit the scope of this work to the move-in-line constraint introduced by Perez-D'Arpino and Shah [10], and physical interactions resulting in gripper state changes. Learning constraints from a single demonstration can result in over-constrained specifications [10]. To mitigate this, IKL learns the move-in-line constraints in two stages, as shown in Fig. 1(a): (1) the interactive demonstration and (2) offline keyframe learning.

A. Interactive Demonstration

The interactive demonstration framework consists of an online constraint detector and interactive communication, as shown in Fig. 1(a). During teaching, the constraint detector observes the end-effector poses (position and orientation) as a sequence of frames, $X_{ee} = \{x_{ee}^i\}_{i=1:M_t}$, where M_t is the number of frames recorded until time t and infers the latent binary state variable, $Y_c[t]$, denoting the constrained nature of demonstrated motion at time step t. Here, $Y_c[t] = 0$ and $Y_c[t] = 1$ indicate unconstrained and constrained motion, respectively. Informed by $Y_c[t]$, interactive communication begins with the learner's query, followed by the teacher's answer, and ends with a suggestion from the learner; this dialog occurs each time $Y_c[t]$ toggles from 0 to 1. Once the interactive demonstration ends, the teacher's answers (Fb), the inferred state variable ($\{Y_c\}$), and the recorded demonstration data X_{ee} along with the gripper state (open or close) at every frame x_{ee}^i serve as input for the offline learning phase.

1) Online Constraint Detector: The online constraint detector infers latent-state variable $Y_c[t]$ from the end-effector poses, $X_{ee} = \{x_{ee}^i\}_{i=1:M_t}$. First, the poses are *transformed* to $\{d_i\}_{i=1:M_t''}$, which represents the distance error of x_{ee} to constrained line motion; then models are fit to d_i to generate the *switching probability*, $\Pr(\bar{B}|d_i)$, defined as the probability of the switching the constraint state given d_i, and, finally, the latent state, $Y_c[t]$ is *inferred*. As the scope of this work is limited to the straight-line constraint, there are only two states in which the end-effector can be at instance i: constrained ($Y_c^i = 1$) or unconstrained ($Y_c^i = 0$). We employ a logistic regression model, with distance error $\{d_i\}$ as the independent variable, to model states of motion, Y_c^i. The parameters of this model are learned by fitting a logistic regression model to labeled distance errors computed using demonstration data of a straight line and a circle. This demonstration data is recorded during a simple calibration step, wherein the human moves the end-effector in a straight line and then in a circle.

a) Transformation: The process of inferring the latent constraint state occurs online, updating $Y_c[t]$ with each new end-effector pose observation, x_{ee}. First, $\{x_{ee}^i\}_{i=1:M_t}$ is filtered using the position vector of the frame, x_{ee}^i, $\boldsymbol{p}_i = (x_i, y_i, z_i)$, such that $\|\boldsymbol{p}_{j+1} - \boldsymbol{p}_j\| = D_F$. Here, D_F is a predefined distance value, and $\boldsymbol{p}_j$ is the position vector of frame x_F^j in the new filtered sequence of frames, $X_F = \{x_F^j\}_{j=1:M_t'}$. Next, the filtered trajectory, X_F, is converted to the distance errors, $\{d_i\}_{i=1:M_t''}$, by computing the perpendicular distance from $\boldsymbol{p}_{j+2}$ to the straight line fitted to $\boldsymbol{p}_{j+1}$ and $\boldsymbol{p}_j$.

b) Switching probability: We assume constraint states are independent in this step; therefore, the probability of the constraint state switching given d_i and $Y_c^{i-1} = B$, $\Pr(Y_c^i = \bar{B}|d_i, Y_c^{i-1} = B)$, is the logistic regression model classification probability of $Y_c^i = \bar{B}$ given d_i, $\Pr(\bar{B}|d_i)$.

c) Inference: Due to instrument and human motion noise, simply thresholding the switching probability values can lead to incorrect high-frequency switching of state values; therefore, we drew inspiration from Khoramshahi et Billard [7] and introduced an energy tank that governs state switching. The energy of the tank, T_i, is defined as $T_i = T_{i-1} + \Pr(\bar{B}|d_i)^2 - T_d$, where T_d is the constant dissipated energy. When $T_i \geq T_s$, where T_s is the threshold that triggers a state switch, the state of the system is switched ($Y_c^i = 1 - Y_c^{i-1}$) and the energy of the tank is reinitialized to zero — i.e., $T_i = 0$. When $T_i < T_s$, $Y_c^i = Y_c^{i-1}$.

2) Interactive Communication.: Interactive communication is a dialogue between the teacher and learner with a [query, answer, suggestion] format designed to capture the human's intent for the demonstrated straight-line motion. Following the guidelines established by Cakmak et al. [3], we adopt closed-form, physically grounded feature queries under the *queries made only under certain conditions* mode. Only a single feature-based query about the validity of the detected constraint is required; here, the feature is the inferred constraint state of motion $Y_c[t]$ at time t, and the query is *"Do I have to move in a straight line there?"* The answers to this query (*"Yes," "No,"* and *"I don't know"*) are presented to the teacher each time it is posed (as shown in Fig. 1(b)), which occurs when $Y_c[t-1] = 0$ and $Y_c[t] = 1$. To physically ground these queries, we pose them at the time of occurrence and present images on a screen, illustrating each answer's impact on learning. Based on the teacher's answers, we allow the learner to provide a suggestion to improve the teacher's understanding of the learner, as illustrated in Fig. 1(c). The suggestion about demonstrating unconstrained motion, *"Please do not move in a line unless the robot must move in a line,"* is only given if the teacher's answer is either *"No"* or *"I don't know."* The teacher can accept or decline the suggestion; we assume the learner cannot access the teacher's decision.

B. Offline Keyframe Learning

Offline keyframe learning (OKL) utilizes demonstration data X_{ee}, human feedback Fb, and the constraint state inferred online, $\{Y_c\}$, to learn S_E. First, X_{ee} is segmented based on interactions captured by frames corresponding to a gripper state change. Next, the constraint regions detected online are modified to reflect the feedback received by removing the constraint segments corresponding to the feedback *"No."*.OKL then performs constraint-based segmentation for each remaining constraint region by performing two least-square fittings and denoising the fitting errors via total variation denoising (TVD) [11]. Finally, the boundary frames of the constraint and interaction segments are encoded as keyframes to create S_E.

III. Evaluations

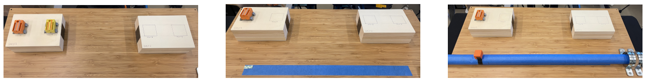
In a within-subject study, we evaluated our proposed framework against two baseline methods: keyframe demonstrations (KD) [1, 10] and the articulated constraints learning approach [8], which was augmented with keyframes inferred by gripper commands (mACL).

A. Hypotheses

To learn S_E, IKL employs human feedback to reduce excess keyframes and constraint errors; in contrast, KD utilizes user-defined keyframes without constraint information, and mACL cannot account for the possibility of unintentionally over-constrained demonstrations. Thus, we hypothesized that (1) keyframe and (2) pose accuracies will be *greater* with our technique than both baselines, (3) constraint accuracy will be *greater* with our technique than the mACL baseline, (4) teaching workload will be *lower* with our technique than both baselines, and (5) teaching efficiency will be *greater* with our technique than both baselines. We computed all accuracy measures as the intersection-over-union (IoU) between the corresponding feature of the learned specification and the ground truth. We measured teaching workload according to the NASA-TLX workload scale [5] and determined teaching efficiency by dividing keyframe accuracy by teaching workload.

B. Experimental Design

We designed the experiment to collect kinesthetic demonstrations of three tasks, as shown in Fig. 2: a pick–and-place task with no constraints (Task 1), an inspection task with an explicit constraint (Task 2), and an assembly task with an implicit constraint (Task 3). We defined "explicit constraints" as those explicitly written in the task description provided to participants, whereas "implicit constraints" were required in order to execute the task correctly but not explicitly written.



(a) Pick-and-place task (b) Inspection task (c) Assembly task

Fig. 2. Task setup. 2(a): The pick-and-place task is to move both objects onto the other shelf. 2(b): The inspection task is to first move the object along the blue line and then place it on the other shelf. 2(c): The assembly task is to first move the object to the other shelf, then grasp the orange block on the pipe and slide the pipe through the lamp.

Our setup comprised a Franka Emika Panda robot arm and a graphical user interface (GUI) to display task descriptions and robot queries. Query answers and gripper commands were given through speech. In our experiment, we defined the distance value (D_F) and TVD parameter as 5 cm and 0.6, respectively, and all remaining parameters were inferred from participants' calibration and demonstration data.

The experiment comprised 12 participants ranging in age from 20 to 62 years (M = 28.33, SD = 10.73), including six men and six women. Seven participants indicated no prior experience in robotics. Each participant began with a demographic questionnaire and training session, followed by a calibration phase during which they moved the robot along a predefined line and circle. Calibration was followed by a primary phase consisting of nine tasks: three modes balanced between subjects using a Latin square design — IKL, KD, and mACL — with three tasks per mode. The experimenter gave the following instruction to the participants on how to provide a demonstration under the IKL and mACL teaching modes: *"When moving the robot, try to move the robot in a line only when you have to move in a straight line to perform the task."* For KD teaching mode, the users were told to *"Give the least number of steps you think the robot needs to perform the task correctly."* Participants responded to the NASA-TLX questionnaire [5] after each task. The keyframe, constraint, and NASA-TLX data collected was analyzed using MATLAB's linear mixed-effects modeling function (film), which incorporated a participant's age, sex, experience in robotics and using a joystick controller, and the chronological order of modes used.

C. Results and Discussion

KD cannot learn constraint labels for keyframes; therefore, we analyzed success rates under two conditions, as reported in Table I. Here, condition (a) utilizes the original learned specification, and condition (b) employs a modified learned specification incorporating an accurate constraint label for the user-defined keyframes. As expected, under condition (a), KD reported 0% success rates for Tasks 2 and 3, and IKL reported the highest success rates for all tasks under both conditions.

The linear mixed-effects model analysis supported *Hypothesis 1 for keyframe accuracy*, indicating a highly significant increase in IKL's accuracy compared with KD ($p < 0.01$) and mACL ($p < 0.01$). As KD cannot learn constraints, we examined the keyframe pose performance that did not penalize the constraint labels of the learned keyframes and

TABLE I
TASK SPECIFICATION LEARNING SUCCESS RATES AS PERCENTAGES.

	KD		mACL	IKL
	Con.(a)	Con.(b)		
Task1	50.0	50.0	33.3	66.7
Task2	0.0	33.3	41.7	75.0
Task3	0.0	41.7	16.7	58.3
Overall	13.9	41.7	30.6	66.7

found support for *Hypothesis 2 for pose accuracy*, suggesting IKL's pose accuracy was significantly higher than that of KD ($p < 0.05$) and mACL ($p < 0.01$).

Although the analysis of *Hypothesis 3 for constraint accuracy* indicated an increase in constraint accuracy for IKL compared with mACL, the results were statistically insignificant. However, the comparison of distributions of missed and incorrect constraint lengths per trial suggested improved constraint accuracy for IKL compared with mACL. For instance, 50% of task specifications learned via IKL had less than 2.18 cm of constraint length errors, whereas those for mACL were spread over 23.65 cm. Additionally, the total number of constraint errors for IKL (19 errors) was much lower than that for mACL (46). We also discovered that IKL was able to prevent 51.3% of incorrectly demonstrated constrained motion segments through human feedback, while 73.7% of constraint errors for IKL were due to incorrect human feedback. These findings suggest the importance of leveraging human feedback to reduce constraint errors, and that increasing feedback accuracy will directly improve IKL's constraint accuracy.

Analysis of the NASA-TLX workload scores partially supported *Hypothesis 4 for teaching workload* by indicating a highly significant increase in workload for KD ($p < 0.01$) and an insignificant decrease in workload for mACL ($p > 0.05$) compared with IKL. These results suggest that switching from user-defined keyframes to a continuous demonstration framework significantly reduced workload, whereas interactive communication introduced to continuous demonstrations increased teaching workload. The analysis also supported *Hypothesis 5 for teaching efficiency*, suggesting a significant increase in IKL's teaching efficiency compared with KD ($p < 0.01$) and mACL ($p < 0.05$). These results indicate that although IKL yielded a greater teaching workload, it improved the efficiency of teaching tasks to a robot compared with KD and mACL.

IV. CONCLUSION

Our proposed interactive keyframe learning framework (IKL) learns a task specification encoded as an ordered sequence of keyframes in order to capture a task's physical interactions and move-in-line constraints from a single demonstration. Our human-subject experiment reported significant performance gains compared with two state-of-the-art keyframe and constraint learning techniques. We found that although human-robot communication during demonstrations can increase a teacher's workload, it significantly improves overall teaching efficiency. Furthermore, we discovered the positive impact of human feedback on constraint learning and that IKL's constraint accuracy could be improved further through design improvements in human-robot communication.

ACKNOWLEDGMENTS

This work was supported by Mitsubishi Electric Automation, Inc. We greatly appreciate anonymous reviewers for their constructive and rigorous reviews, and Nadia Figueroa for her invaluable advice on improving this paper.

REFERENCES

[1] B. Akgün, M. Cakmak, K. Jiang, and A. Thomaz. Keyframe-based Learning from Demonstration. *International Journal of Social Robotics*, 2012.

[2] D. Berenson, S. Srinivasa, and J. Kuffner. Task Space Regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 2011.

[3] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012.

[4] P. Fearnhead. Exact and Efficient Bayesian Inference for Multiple Changepoint Problems. *Statistics and Computing*, 2006.

[5] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index):Results of Empirical and Theoretical Research. In *Human Mental Workload*. 1988.

[6] J. Huang, D. Fox, and M. Cakmak. Synthesizing Robot Manipulation Programs from a Single Observed Human Demonstration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[7] M. Khoramshahi and A. Billard. A Dynamical System Approach for Detection and Reaction to Human Guidance in Physical Human–Robot Interaction. *Auton. Robots*, 2020.

[8] Y. Liu, F. Zha, L. Sun, J. Li, M. Li, and X. Wang. Learning Articulated Constraints From a One-Shot Demonstration for Robot Manipulation Planning. *IEEE Access*, 2019.

[9] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. Online Bayesian changepoint detection for articulated motion models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[10] C. Pérez-D'Arpino and J. A. Shah. C-LEARN: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[11] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 1992.

[12] L. Sanneman, C. Fourie, and J. A. Shah. The State of Industrial Robotics: Emerging Technologies, Challenges, and Key Research Directions, 2020.