Assisted Teleoperation for Scalable Robot Data Collection

Shivin Dass[1†], Karl Pertsch[1†], Hejia Zhang[1], Youngwoon Lee[1], Joseph J. Lim[2], and Stefanos Nikolaidis[1]

***Abstract*— Large-scale data is an essential component of machine learning as demonstrated in recent advances in natural language processing and computer vision research. However, collecting large-scale robotic data is much more expensive and slower as each operator can control only a single robot at a time. To make this costly data collection process efficient and scalable, we propose a novel assisted teleoperation system, which automates part of the demonstration collection process using a learned assistive policy. The assistive policy autonomously executes repetitive behaviors in data collection and asks for human input only when it is uncertain about which subtask or behavior to execute. We conduct teleoperation user studies both with a real robot and a simulated robot fleet and demonstrate that our assisted teleoperation system reduces human operators' mental load while improving data collection efficiency. Further, it enables a single operator to control multiple robots in parallel, which is a first step towards scalable robotic data collection.**

I. INTRODUCTION

Recently, many works have shown impressive robot learning results from diverse, human-collected demonstration datasets [1]–[4]. They underline the importance of scalable robot data collection. Yet, the current standard approach for demonstration collection, human teleoperation, is tedious and costly: tasks need to be demonstrated repeatedly and each operator can control only a single robot at a time. Research in teleoperation has focused on exploring different interfaces, such as VR controllers [5] and smart phones [1], but does not address the aforementioned bottlenecks to scaling data collection. Thus, current teleoperation systems are badly equipped to deliver the scalability required by modern robot learning pipelines.

Our goal is to improve the scalability of robotic data collection by providing assistance to the human operator during teleoperation. We take inspiration from other fields of machine learning, such as semantic segmentation, where costly labeling processes have been substantially accelerated by providing human annotators with learned assistance systems, e.g., in the form of rough segmentation estimates, that drastically reduce the labeling burden [6], [7]. Similarly, we propose to train assistive *policies*, that can automate control of repeatedly demonstrated behaviors and ask for user input only when facing a novel situation or when unsure which behavior to execute. Thereby, we aim to reduce the mental load of the human operator and enable scalable teleoperation by allowing a single operator to perform data collection with multiple robots in parallel.

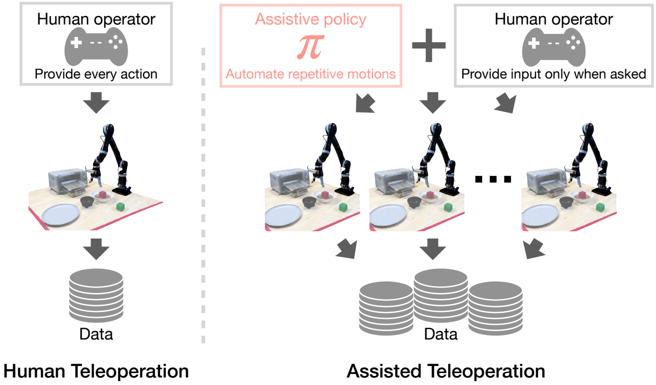


Fig. 1: Policy-assisted teleoperation enables large-scale data collection by minimizing human operator inputs and mental efforts with an assistive policy, which autonomously performs repetitive subtasks. This allows a human operator to simultaneously manage multiple robots.

In order to build an assistive system for robotic data collection, we need to solve two key challenges: (1) we need to learn assistive policies from diverse human-collected data, which is known to be challenging [8], and (2) we need to learn when to ask for operator input while keeping such interventions at a minimum. To address these challenges, we propose to use a hierarchical stochastic policy that can learn effectively from diverse human data. Further, we use the policy's stochastic predictions to estimate its uncertainty about how to act in the current scene and which task to pursue. Then, we use this estimate to elicit operator input only if the assistive policy is uncertain about how to proceed.

The main contribution of this paper is a novel assisted teleoperation system, which enables scalable robotic data collection using a hierarchical assistive policy. We evaluate the effectiveness of our approach in a user study in which operators collect datasets of diverse kitchen-inspired manipulation tasks with a real robot. We find that our proposed *assisted* teleoperation approach reduces operators' mental load and improves their demonstration throughput. We further demonstrate that our approach allows a single operator to control data collection with multiple robots simultaneously in a simulated manipulation environment – a first step towards more scalable robotic data collection.

*This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant (No.2019-0-00075, Artificial Intelligence Graduate School Program, KAIST) and National Research Foundation of Korea (NRF) grant (NRF-2021H1D3A2A03103683), funded by the Korea government (MSIT).

[1]Shivin Dass, Karl Pertsch, Hejia Zhang, Youngwoon Lee, and Stefanos Nikolaidis are with the University of Southern California, USA {sdass, pertsch, hejiazha, lee504, nikolaid}@usc.edu

[2]Joseph J. Lim is with Korea Advanced Institute of Science and Technology and NAVER AI Lab, South Korea joe.lim@kaist.ac.kr

[†]Equal contribution.

II. RELATED WORK

Robot Teleoperation. Demonstrations have played a key role in robot learning for many decades [9]–[11], thus many approaches have been explored for collecting such demonstrations. While initially kinesthetic teaching was common [12] in which a human operator directly moves the robot, more recently teleoperation has become the norm [1], [4], [5], [13], [14], since separating the human operator and the robot allows for more comfortable human control inputs and is crucial for training policies with image-based inputs. Research into teleoperation systems has focused on exploring different interfaces like VR headsets [4], [5], joysticks [2] and smartphones [1]. Yet, none of these works explores active assistance of the human operator during teleoperation. Others have investigated controlling high-DoF manipulators via low-DoF interfaces through learned embedding spaces [15], [16] to allow people with disabilities to control robotic arms. In contrast, our approach trains assistive policies that automate part of the teleoperation process with the goal of enabling more scalable data collection.

Shared Autonomy. The idea of sharing efforts between humans and robots when solving tasks has a rich history in the human-robot-interaction (HRI) community [17]–[26]. Approaches for such *shared autonomy* typically rely on a pre-defined set of goals and aim to infer the intent of the human operator to optimally assist them. Crucially, in the context of data collection, we cannot assume that all goals are known a priori, since a core goal of data collection is to collect previously unseen behaviors. Thus, instead of inferring the operator's intent over a fixed goal set, we leverage the model's estimate over its own uncertainty to determine when to assist and when to rely on operator input.

Interactive Human Robot Learning. In the field of robot learning, many approaches have explored leveraging human input in the learning loop and focused on different ways to decide when to leverage such input. Based on the DAgger algorithm [27], works have investigated having the human themselves decide when to intervene [28], using ensemble-based support estimates [29], using discrepancies between model output and human inputs [30], [31] or risk estimates based on predicted future returns [32]. Yet, all these approaches focus on training a policy for a single task, not on collecting a diverse dataset. Thus, they are not designed to learn from multi-modal datasets or estimate uncertainty about the desired task. We show in our user study that these are crucial for enabling scalable robot data collection.

Assisted Robot Data Collection. Clever et al. [33] aims to assist in robot demonstration collection via a learned policy. They visualize the projected trajectory of the assistive policy to enable the human operator to intervene if necessary. However, they focus on collection of single-task, short-horizon demonstrations and require the operator to constantly monitor the robot to decide when to intervene. In contrast, our system can collect diverse, multi-task datasets and learn when to ask the user for input, enabling more scalable data collection, e.g., with multiple robots in parallel.

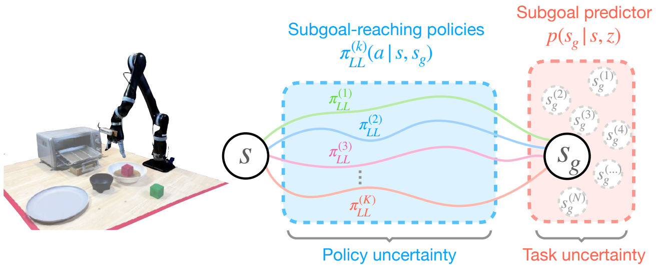


Fig. 2: Our assistive policy is hierarchical: a high-level subgoal predictor $p(s_g|s,z)$ and a low-level subgoal-reaching policy $\pi_{LL}(a|s,s_g)$. To decide when to follow the assistive policy, we measure uncertainty of both high-level (subgoal predictor) and low-level (subgoal-reaching policy) decisions. The task uncertainty is estimated using the subgoal predictor's variance, and the policy uncertainty is estimated as a disagreement among an ensemble of subgoal-reaching policies.

III. APPROACH

In this paper, we aim to improve the scalability of robotic data collection by learning an assistive policy to automate part of teleoperation (in Section III-B) and ask for user inputs only when necessary (in Section III-C).

A. Problem Formulation

An *assistive policy* $\pi(a|s)$ produces actions a, e.g., robot end-effector displacements, given states s, e.g., raw RGB images. To enable scalable data collection of a dataset $\mathcal{D}$, the policy should control the robot and minimize required human inputs, which allows the human operator to divert attention away from the robot over contiguous intervals, e.g., to attend to other robotic agents collecting data in parallel.

To train the assistive policy π we assume access to a pre-collected dataset $\mathcal{D}_{\text{pre}}$ of diverse agent experience, e.g., from scripted policies, previously collected data on different tasks or human play [34]. Crucially, we explicitly require our approach to handle scenarios in which the newly collected dataset $\mathcal{D}$ contains behaviors that are *not* present in $\mathcal{D}_{\text{pre}}$. Thus, it is *not* possible to fully automate data collection given the pre-training dataset. Instead, the system needs to request human input for unseen behaviors while providing assistance for known behaviors.

B. Learning Assistive Policies from Multi-Modal Data

Learning the assistive policy $\pi(a|s)$ from the diverse, multi-task data $\mathcal{D}_{\text{pre}}$ is challenging [8], since it is often highly multi-modal and requires imitation of long demonstrations.

In our approach, we build on prior work in imitation of long-horizon, multi-task human data [35]. We propose to use a hierarchical policy with a subgoal predictor $p(s_g|s,z)$ and a low-level subgoal reaching policy $\pi_{\text{LL}}(a_t|s_t,s_g)$ (see Figure 2). We condition the subgoal predictor on a stochastic latent variable z to allow prediction of the *full distribution* of possible subgoals. For example, in a multi-modal dataset in which the robot moves veggies into the oven in half of the trajectories and places them on a plate in the other half,

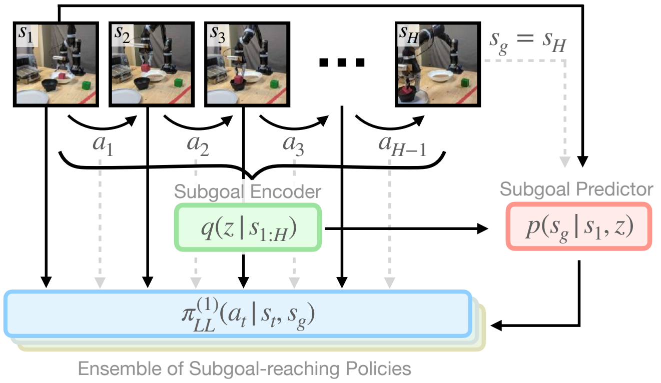


Fig. 3: Our hierarchical assistive policy is trained using a pre-collected dataset $\mathcal{D}_{\text{pre}}$. From a sampled trajectory $(s_1, a_1, \ldots, a_{H-1}, s_H)$ of length H, a subgoal predictor $p(s_g|s_1, z)$ is trained as a conditional VAE to cover a multi-modal subgoal distribution, where $s_g = s_H$. Then, an ensemble of subgoal-reaching policies $\pi_{LL}^{(k)}(a_t|s_t, s_g)$ are trained to predict the ground truth actions.

z captures whether to predict a subgoal with veggies in the oven or on the plate.

We train the subgoal predictor as a conditional variational auto-encoder over subgoals [36]: given a randomly sampled starting state s_t from the pre-training dataset and a subgoal state $s_{t+H} = s_g$ H steps later in the trajectory, we use a learned inference network $q(z|s_t, s_g)$ to encode s_t and s_g into a latent variable z. We then use the subgoal predictor $p(s_g|s_t, z)$ to decode back to the original subgoal state. During training we apply a subgoal reconstruction loss, as well as a regularization loss on the latent variable z. Finally, the subgoal reaching policy is trained via simple behavioral cloning. We summarize the components of our training model in Figure 3. Our final training objective is:

$$\max_{\theta,\phi,\mu} \; \mathbb{E}_{\substack{(s,a,s_g)\sim\mathcal{D}_{\text{pre}} \\ z\sim q(\cdot|s,s_g)}} \underbrace{p_\theta(s_g|s,z)}_{\text{subgoal reconstruction}} + \underbrace{\pi_{\text{LL},\phi}(a|s,s_g)}_{\text{behavioral cloning}} \\ - \underbrace{\beta D_{\text{KL}}\big(q(z|s,s_g), p(z)\big)}_{\text{latent regularization}} + \qquad (1)$$

We use θ, ϕ, μ to denote the parameters of the subgoal predictor, goal reaching policy, and inference network, respectively. β is a regularization weighting factor, D_{KL} denotes the Kullback-Leibler divergence, and we use a unit Gaussian prior $p(z)$ over the latent variable.

To execute our assistive policy $\pi(a|s)$, we first sample a latent variable z from the unit Gaussian prior, then pass z and s through the subgoal predictor $p(s_g|s, z)$ to generate a subgoal and then use the goal-reaching policy to predict executable actions $\pi_{\text{LL}}(a|s, s_g)$.

C. Deciding When to Request User Input

A core requirement of our approach is that it can actively ask for operator input while minimizing the number of such requests. This is crucial, since it allows the operator to divert their attention to other tasks, e.g., controlling other robots, while the assistive policy is controlling the robot. Thus, a key question is: when should the policy ask for human inputs?

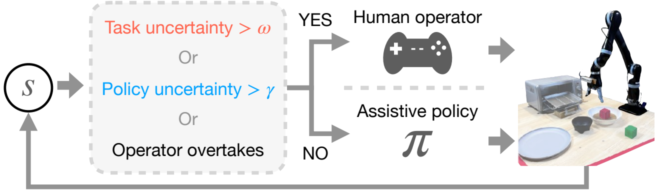


Fig. 4: Our approach asks for human inputs when the assistive policy is uncertain about which subtask or action to take. If both the task uncertainty and policy uncertainty are lower than their thresholds, our assistive policy can reliably perform a subtask, reducing the workload of the human operator.

Intuitively, the policy should request help when it is uncertain about what action to take next. This can occur in two scenarios: (1) the policy faces a situation that is not present in the training data, so it does not know which action to take, or (2) the policy faces a seen situation, but the training trajectories contain multiple possible continuations and the policy is not sure which one to pick. The latter scenario commonly occurs during the collection of diverse datasets, since trajectories for different tasks often intersect. For example, in a kitchen environment, multiple tasks could start by tossing veggies in a pan, but then diverge into putting the pan on the stove or in the oven. During teleoperation, human inputs are required for such situations to decide which task to continue with. This is in contrast to single-task demonstration collection, e.g., during DAgger training, where such uncertainty over tasks usually does not occur.

Our hierarchical model allows us to separately estimate both classes of uncertainty. To estimate whether a given state is unseen, we follow prior work on out-of-distribution detection [29], [37] and train an ensemble of K low-level reaching policies, all on the same data $\mathcal{D}_{\text{pre}}$ but with different initializations and batch ordering. By measuring the disagreement $D(a^{(1)}, \ldots, a^{(K)})$ between the actions predicted by these ensemble policies, i.e., the mean of the variance of actions in each dimension, we can recognize unseen states as those states with high disagreement [37].

If we determine that a state is seen in the training data, we can estimate the policy's certainty about the task by sampling from the distribution of subgoals produced by the subgoal predictor and computing the inter-subgoal variance $Var(s_g^{(1)}, \ldots, s_g^{(N)})$. In cases with multiple possible task continuations, this variance will be high since the distribution of possible subgoals will widen.

We leverage both uncertainty estimates to decide on whether the assistive policy should continue controlling the robot or whether it should stop and ask for human input. We found a simple thresholding scheme sufficient, with threshold parameters γ, ω for the ensemble disagreement and subgoal variance, respectively. Future work can investigate more advanced mixing schemes, e.g., with auto-tuned thresholds [32] or hysteresis between switching from robot to human and back [31]. We also include a human override H_t that allows the human to actively take control at any point during the teleoperation, e.g., in order to demonstrate a new task from a seen state, in which case the assistive policy would not ask for human input automatically. In summary, we *continue* executing the assistive policy if:

$$Assist = \neg \underbrace{(D(a^{(1)}, \ldots, a^{(K)}) > \gamma)}_{\text{OOD state}} \wedge \neg \underbrace{(Var(s_g^{(1)}, \ldots, s_g^{(N)}) > \omega)}_{\text{uncertain task}} \wedge \neg \underbrace{H_t}_{\text{human override}} \tag{2}$$

IV. EXPERIMENTS

We aim to answer the following questions: (1) Does using our assisted teleoperation approach reduce the mental load of human operators? (2) Does it allow the operators to divert their attention to other tasks during teleoperation? (3) Can our system scale robotic data collection by allowing a single operator to teleoperate multiple robotic systems in parallel? To answer these questions, we perform two user studies in which (1) users teleoperate a real robot arm to perform diverse manipulation tasks, and (2) users teleoperate *multiple* simulated robotic arms simultaneously.

A. User Study Design

To evaluate the effectiveness of our assisted teleoperation, we conduct a user study ($N = 16$) in which users teleoperate a Kinova Jaco 2 robot arm to collect diverse robot manipulation demonstrations for kitchen-inspired long-horizon tasks, e.g., "place ingredients in bowl" and "place bowl in oven" (see Figure 5). Users teleoperate the robot's end-effector via joystick and buttons on a standard gamepad controller.

The users are also asked to solve simple side tasks during teleoperation to measure their ability to divert attention and conduct other tasks. Specifically, they are shown a string of randomly selected characters, one of which is different from the others (e.g., 000100), and asked to specify the index of the odd character. To ensure that the users can only attend to teleoperation *or* the side task at a time, they perform teleoperation purely via a live-stream video of the robot setup, without being able to see the physical robot. Crucially, they can only see *either* the video stream *or* the side task and can switch between the two views with a button press.

To train our assistive policy, we collect a pre-training dataset of 120 demonstrations. Crucially, during the user study the operators need to collect *unseen* long-horizon tasks.

We compare our approach to (1) teleoperation without assistance, the current standard approach to collection robot demonstration data and (2) ThriftyDAgger [32], the closest prior work to ours for interactive human-robot data collection. ThriftyDAgger is designed to minimize human inputs during single-task demonstration collection by requesting human input only in *critical* states where a learned value function estimates low probability for reaching the goal. We initially implemented ThriftyDAgger with a flat policy as in the original work, but found it could not learn to model the multi-modal trajectories in our training dataset, leading to poor performance. Thus, we compare to an improved version of ThriftyDAgger that uses the same hierarchical policy as ours, which we found better suited to learn from the multi-modal data.

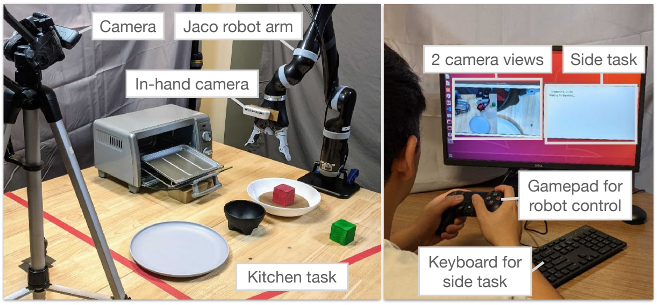


Fig. 5: User study setup. **(left)** A Kinova Jaco arm, front-view and in-hand cameras, and objects for kitchen-inspired tasks are placed on the workspace. **(right)** A human operator can watch a monitor, which shows either the camera inputs or a side task. The operator uses a gamepad to control the robot, and uses a keyboard to solve the side task.

Our goal is to measure the user's mental workload during teleoperation as well as their ability to divert their attention to other tasks, which is crucial for scaling teleoperation by allowing a single user to teleoperate multiple robots (see Section IV-C). After each teleoperation session, we administer the NASA TLX survey [38], [39], which measures the user's perceived workload. We aggregate the responses to obtain a single score for each user. Participants also answer a survey adopted from previous work [22], which assesses their perception of the robot's intelligence, their satisfaction and trust in the system (Table II).

B. User Study Analysis

TABLE I: Average number of completed side tasks and teleoperation time per demonstration during the real-robot teleoperation user study.

Approach	Avg. Num. of completed side tasks	Avg. teleop time (seconds)
Unassisted	0.25 (±0.66)	109.5 (±31.4)
ThriftyDAgger	**13.06 (±9.63)**	105.9 (± 29.5)
Ours	**15.88 (±7.11)**	**85.0 (±18.2)**

1) Comparison of Assisted Methods with Unassisted Baseline: We first compare the standard unassisted teleoperation baseline to the two methods with assistance in Table I: unassisted teleoperation requires the user's full attention, thus they are unable to solve any side task during teleoperation. In contrast, assisted approaches allow the user to solve additional side tasks without sacrificing teleoperation speed.

2) Comparison of Proposed Method with ThriftyDAgger: Now we compare the proposed method with the assisted baseline ThriftyDAgger. First, we statistically evaluate the user responses to the surveys to elicit the participants' trust, satisfaction and perception of the robot's intelligence, as well as their mental workload when teleoperating with the two approaches.

During the study, participants agreed more strongly that they trusted the robot to perform the correct action at the

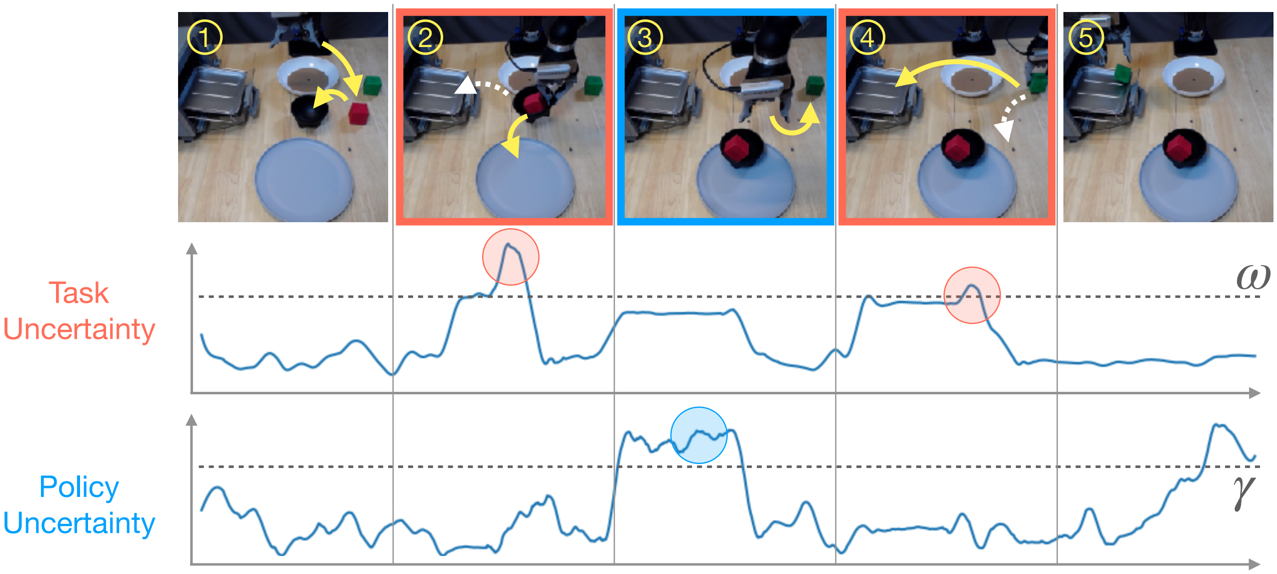


Fig. 6: Visualization of our assistive teleoperation policy on a task from the real-robot user study: *place red block in bowl; place bowl on plate; put green block in oven*. The policy autonomously executes familiar behaviors, but asks for user input in frames (2) and (4) to determine where to place bowl and green block (white vs. yellow arrow). In these cases, the task uncertainty surpasses the threshold ω since the subgoal predictor produces subgoals for both possible targets. Further, the policy asks for user input in frame (3) since the required transition between placing the bowl and picking up the green block was not part of its training data. Thus, the policy uncertainty estimate surpasses its threshold γ.

correct time for the proposed approach (Wilcoxon signed-rank test, $p = 0.001$). Further, they found the robot to be significantly more intelligent with the proposed method (repeated-measures ANOVA, $F(1, 15) = 5.14, p = 0.039$, Cronbach's $\alpha = 0.95$) and were significantly more satisfied with their collaboration with the robot ($F(1, 15) = 5.05, p = 0.040$, $\alpha = 0.91$). Finally, during the NASA TLX survey, participants showed a lower mental workload using the proposed approach compared to the baseline ($F(1, 15) = 5.52, p = 0.033$).

A key factor in the subjective differences between the two approaches is their ability to elicit user feedback at appropriate times: when the robot is at a decision point between two possible task continuations (see Figure 6, frames (2) and (4)). ThriftyDAgger's risk-based objective is not sensitive to such decision points and thus it rarely asks for user feedback. It instead executes one of the possible subtasks at random. In our study we found that this lead to erroneous skill executions in 48 % of cases. Such errors require tedious correction by the user, deteriorating their trust in the system and their teleoperation efficiency. In contrast, our approach leverages its estimate of task uncertainty (see Figure 6) to correctly elicit user feedback in 82 % of cases, leading to higher perceived levels of trust and reduced mental load.

C. Scaling Data Collection to Multiple Robots

In the previous section we showed that our assistive teleoperation approach allows users to divert their attention to other tasks during data collection. An important application of this is multi-robot teleoperation, in which a single operator performs data collection with multiple robots in parallel and periodically attends to different robots. To test this, we conduct a teleoperation study (N = 10) with *multiple* simulated robots in the realistic physics simulator [14] (see Figure 7). A human user is asked to collect demonstrations for a block stacking task with multiple robots *in parallel* via the same gamepad interface used in the real robot study. The user can switch control between different robots with a button press. We measure the total number of collected demonstrations across the robot fleet in a fixed time frame of T = 4 minutes.

We compare the number of collected demonstrations with increasing numbers of simultaneously teleoperated robots in Figure 8a. Our approach enables strong scaling of data collection throughput. As expected, the scaling is not linearly proportional, i.e., four robots do not lead to four times more demonstrations collected. This is because simultaneous teleoperation of a larger fleet requires more context switches between the robots, reducing the effective teleoperation time. Yet, our method enables effective parallel teleoperation, leading to higher demonstration throughput with larger robot

TABLE II: Post-execution survey (Likert scales with 7-option response format)

Trust:
Q1. I trusted the robot to do the right thing at the right time.

Robot intelligence ($\alpha = 0.95$)**:**
Q2. The robot was intelligent.
Q3. The robot perceived accurately what my goals are.
Q4. The robot and I worked towards mutually agreed upon goals.
Q5. The robot's actions were reasonable.
Q6. The robot did the right thing at the right time.

Human satisfaction ($\alpha = 0.91$)**:**
Q7. I was satisfied with the robot and my performance.
Q8. The robot and I collaborated well together.
Q9. The robot was responsive to me.

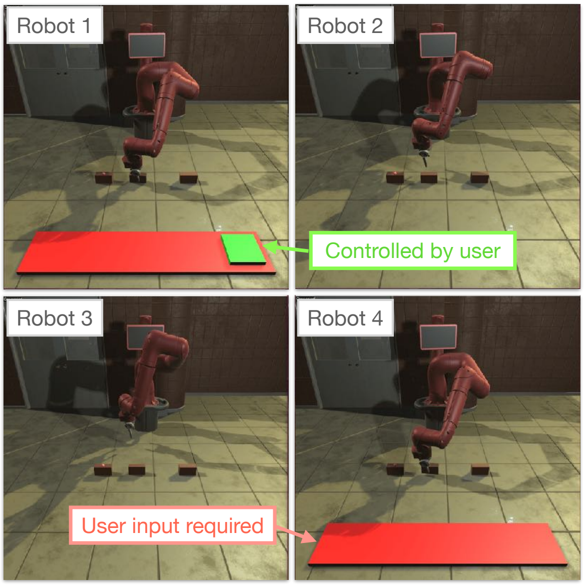
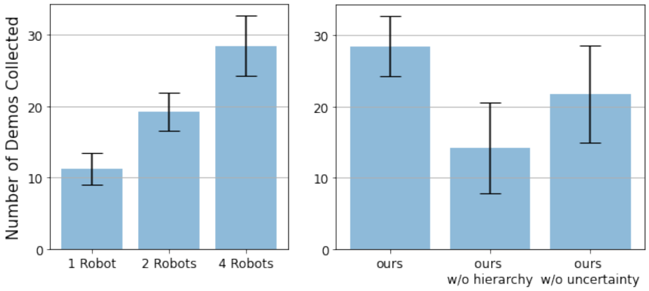


Fig. 7: We use up to four simulated robots to collect demonstrations. The assistive policy asks for human input using the red indicator. The green indicator represents which environment is being controlled by a user.



(a) Scaling experiment

(b) Ablation study

Fig. 8: Average number of demonstrations collected in 4 minutes using multiple robots in simulation. **(a)** With our assisted teleoperation, users can manage multiple robots simultaneously and collect more demonstrations with four robots. **(b)** The ablated systems with four robots show inferior demonstration collection throughput.

fleets. In contrast, with standard unassisted teleoperation, a single operator can only control a single robot such that the demonstration throughput would not increase even if a larger fleet of robots was available.

D. Ablation Study

Our approach has two key components: the hierarchical policy (Section III-B) and the uncertainty based requesting of user input (Section III-C). To evaluate their importance, we perform ablation studies in the 4-robot teleoperation setup from Section IV-C. We compare our full method against two ablations: (1) **ours w/o hierarchy**, which trains an ensemble of *flat* stochastic policies $\pi(a|s)$, and (2) **ours w/o uncertainty**, which removes the uncertainty-based requesting of user input. The flat policy ablation “ours w/o hierarchy” uses the ensemble disagreement and action distribution variance to determine policy and task uncertainty, respectively.

We report the number of collected demonstrations per method within $T = 4$ min averaged across $N = 10$ users in Figure 8b. Both ablations perform worse than our approach. Specifically, we find that the flat policy is neither able to accurately model the multi-modal training data nor to accurately estimate the task uncertainty. As a result, the assistive policy does not ask for user input in critical decision states and exhibits frequent control failures, leading to reduced teleoperation throughput. Similarly, the ablation without uncertainty estimation does not elicit user input in critical states and thus requires the user to correct it whenever it tries to complete an incorrect task. As a result, the demonstrations collected with the ablated methods are less optimal, requiring an average 398 and 375 steps until task completion for “ours w/o hierarchy” and “ours w/o uncertainty”, respectively, compared to 322 steps for our method.

E. Implementation Details

For real robot experiments, we use both front-view and in-hand camera images of size $244 \times 244 \times 3$ as an observation. But, instead of using raw images, we use a 2048-dimensional pre-trained visual representation from R3M [40]. In addition, we include the end-effector pose and gripper state of the robot in the observation space. The action space consists of a 3D translational action and 1D discrete gripper action. The conditional-VAE for the subgoal predictor consists of encoder and decoder with 5-layer MLP with 128 hidden units, and ReLU activation. We use an ensemble of $K = 5$ LSTM subgoal-reaching policies with 128 hidden units and the Adam optimizer [41]. We use the same models for simulated experiments but the observation space consists of robot joint positions, velocities, and object poses.

V. CONCLUSION

A large-scale robot demonstration dataset is key to enabling the next breakthrough in robot learning. As a step towards large-scale robot data, we propose an efficient and scalable system for robotic data collection, which automates part of human teleoperation using a learned assistive policy and actively asks for human input in critical states. This allows a human operator to handle multiple robots simultaneously and significantly improve data collection throughput. The user study supports that our assisted teleoperation system requires infrequent inputs from users and users feel less mental burden when collecting robot demonstrations. We further show significantly improved data collection throughput of our system in the multi-robot control experiments.

For simplicity, we assume access to pre-collected data $\mathcal{D}_{\text{pre}}$ to train the assistive policy. However, our assistive policy can, in theory, be learned from scratch and continuously improved as more data is collected. We leave this as a future work. Moreover, deploying the multi-robot data collection in the real world is a plausible future work.

References

[1] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *CoRL*, 2018.

[2] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang, "Scaling data-driven robotics with reward sketching and batch reinforcement learning," *RSS*, 2019.

[3] Y. Lu, K. Hausman, Y. Chebotar, M. Yan, E. Jang, A. Herzog, T. Xiao, A. Irpan, M. Khansari, D. Kalashnikov, and S. Levine, "Aw-opt: Learning robotic skills with imitation and reinforcement at scale," in *CoRL*, 2021.

[4] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," in *RSS*, 2022.

[5] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *ICRA*, 2018.

[6] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *CVPR*, 2017.

[7] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *CVPR*, 2018, pp. 859–868.

[8] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *CoRL*, 2021.

[9] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *NIPS*, 1989, pp. 305–313.

[10] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics*, vol. 59, no. BOOK_CHAP, 2008.

[11] 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.

[12] H. B. Amor, E. Berger, D. Vogt, and B. Jung, "Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction," in *Annual conference on artificial intelligence*, 2009.

[13] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," *CoRL*, 2019.

[14] Y. Lee, E. S. Hu, Z. Yang, A. Yin, and J. J. Lim, "IKEA furniture assembly environment for long-horizon complex manipulation tasks," *ICRA*, 2021. [Online]. Available: https://clvrai.com/furniture

[15] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *ICRA*, 2020.

[16] H. J. Jeon, D. P. Losey, and D. Sadigh, "Shared autonomy with learned latent actions," *RSS*, 2020.

[17] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," in *RSS*, 2015.

[18] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, 2021.

[19] P. Berthet-Rayne, M. Power, H. King, and G.-Z. Yang, "Hubot: A three state human-robot collaborative framework for bimanual surgical tasks based on learned models," in *ICRA*. IEEE, 2016, pp. 715–722.

[20] A. Pichler, S. C. Akkaladevi, M. Ikeda, M. Hofmann, M. Plasch, C. Wögerer, and G. Fritz, "Towards shared autonomy for robotic tasks in manufacturing," *Procedia Manufacturing*, vol. 11, pp. 72–82, 2017.

[21] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration," *Science Robotics*, vol. 2, no. 7, p. eaan5074, 2017.

[22] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2017, pp. 294–302.

[23] M. Johns, B. Mok, D. Sirkin, N. Gowda, C. Smith, W. Talamonti, and W. Ju, "Exploring shared control in automated driving," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 91–98.

[24] B. D. Argall, "Autonomy in rehabilitation robotics: An intersection," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, p. 441, 2018.

[25] M. Fontaine and S. Nikolaidis, "A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy," *arXiv preprint arXiv:2012.04283*, 2020.

[26] M. C. Fontaine and S. Nikolaidis, "Evaluating human–robot interaction algorithms in shared autonomy via quality diversity scenario generation," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 11, no. 3, pp. 1–30, 2022.

[27] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *AISTATS*, 2011, pp. 627–635.

[28] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *ICRA*, 2019.

[29] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer, "Ensembledagger: A bayesian approach to safe imitation learning," in *IROS*, 2019.

[30] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *AAAI*, 2017.

[31] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg, "Lazydagger: Reducing context switching in interactive imitation learning," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021.

[32] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," *CoRL*, 2021.

[33] H. M. Clever, A. Handa, H. Mazhar, K. Parker, O. Shapira, Q. Wan, Y. Narang, I. Akinola, M. Cakmak, and D. Fox, "Assistive tele-op: Leveraging transformers to collect robotic task demonstrations," *arXiv preprint arXiv:2112.05129*, 2021.

[34] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," in *CoRL*, 2020.

[35] A. Mandlekar, F. Ramos, B. Boots, L. Fei-Fei, A. Garg, and D. Fox, "Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," *ICRA*, 2020.

[36] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NIPS*, 2015.

[37] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, 2017.

[38] S. G. Hart, "Nasa task load index (tlx)," 1986.

[39] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

[40] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," in *CoRL*, 2022.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.