Imitation Learning for Construction Robot Timber Assembly using Demonstrations from Leader–Follower Teleoperation

Xinhe Yang[1], Lei Huang[2], and Zhengbo Zou[3]

[1]M.S. Student, Mechanical Engineering, Columbia University, U.S.
[2]Ph.D. Student, Civil Engineering and Engineering Mechanics, Columbia University, U.S.
[3]Assistant Professor, Civil Engineering and Engineering Mechanics, Columbia University, U.S.

xy2651@columbia.edu, lh2921@columbia.edu, zhengbo.zou@columbia.edu

Abstract -

Developing autonomous construction robots is crucial for alleviating the compounding pressures of housing supply constraints and persistent labor shortages. Imitation learning has been increasingly adopted in the development of autonomous construction robots in recent years. However, the efficient collection of high-quality expert demonstrations from which robots learn remains a key bottleneck for scaling to a broader range of construction tasks. This paper proposes an end-to-end framework that spans intuitive and efficient demonstration collection using a leader–follower teleoperation system, policy learning from the collected demonstrations, and direct real-robot deployment. The teleoperation system enables a demonstrator to collect expert data on a real robot by maneuvering a small-scale 3D-printed robot-arm manipulator whose joint motions are mapped to a full-scale robot arm. To validate the efficacy of the framework, we train diffusion policies on the collected demonstrations and directly deploy the policies on a real xArm 7 arm. The results show that the policy trained with the full dataset achieves a 100% success rate on a timber truss assembly task.

Keywords -

Construction Robot; Teleoperation; Imitation Learning; Diffusion Policy; Timber Assembly

1 Introduction

The limited availability of workforce with construction expertise is a growing challenge in the U.S. [1, 2], compounding long-standing issues such as cost and schedule overruns [3], eventually limiting the industry's ability to deliver buildings and infrastructure at scale [4]. These labor constraints intensify pressures on productivity and safety; hence, motivating a shift toward construction robotics that can offload dangerous, repetitive, and physically demanding tasks while keeping humans in the loop for oversight in dynamic site conditions [5].

Progress in intelligent construction robots has accelerated rapidly, driven in large part by learning-based control methods, most notably imitation learning (IL) [6, 7, 8] and reinforcement learning (RL) [9, 10]. Although both paradigms have demonstrated strong potential, they differ substantially in their practical requirements for real-world deployment. RL pipelines typically rely on (i) developing a sufficiently accurate simulation "digital twin" of the robot, task setup, and relevant site objects; (ii) specifying a reward function that meaningfully captures desired behavior, often an iterative and labor-intensive design process; and (iii) bridging the sim-to-real gap to ensure the learned policy transfers reliably to physical hardware. In contrast, IL, in particular behavioral cloning (BC), follows a more direct route: collecting expert demonstrations (e.g., via teleoperation or kinesthetic teaching) and training a policy with supervised learning methods to map observations to expert actions, after which the learned policy can be deployed on the real robot without the sim-to-real gap. As a result, compared with RL, BC is often simpler and faster to implement and deploy in practice, making it a natural starting point for learning-based construction robot.

Despite recent advances in BC policy architectures such as Diffusion Policy (DP) [11] and Action Chunking with Transformers (ACT) [12], the collection of high-quality demonstrations remains a primary bottleneck to scaling BC to stronger policies and a broader set of tasks [13]. In the construction context, demonstration data are commonly gathered through teleoperation interfaces such as virtual reality (VR) controllers [7], haptic devices [14], and gesture-based systems [15]. These interfaces typically map a user's input (controller pose, device pose, or inferred hand pose) to an intended end-effector pose, after which inverse kinematics is used to compute the robot arm's joint configuration for execution. While such end-effector–centric teleoperation provides the demonstrator with intuitive control of the end-effector pose, it often provides limited awareness and control of the robot's full-body configuration. As a result, demonstrators may inadvertently generate awkward or suboptimal arm postures (e.g., near joint limits, self-colliding, or poorly conditioned for obstacle clearance), especially in cluttered and dynamically changing site environments.

To address the limited whole-arm awareness in end-

effector–centric teleoperation, we turn to kinematic-isomorphic teleoperation, where the demonstrator directly manipulates a controller that mirrors the robot arm's kinematic structure. We reproduce GELLO [16], a low-cost and open-source teleoperation framework designed specifically to improve the quality and efficiency of demonstration collection for imitation learning by providing an intuitive "embodied" interface for robots. Compared with widely used low-cost alternatives (e.g., VR controllers), GELLO's kinematic correspondence better exposes the robot's joint posture to the human demonstrator, which helps reduce unnatural configurations and improves consistency in collected trajectories.

Building on the reproduced kinematic-isomorphic teleoperation interface and a standardized data pipeline, we aim to lower the practical barrier to scaling BC for construction robots and to provide an end-to-end validation method that spans from teleoperation and dataset curation to policy learning and deployment on a real robot. Specifically, this paper makes the following contributions:

- **Hardware assembly:** We reproduce the GELLO hardware system for construction purposes, providing an embodied teleoperation interface that improves whole-arm awareness during demonstration collection;
- **Full-stack workflow:** We present a complete workflow encompassing demonstration collection with GELLO, synchronized logging and dataset curation, BC-based policy training, and deployment on a real robot; and
- **Real-robot validation:** We validate the efficacy of the collected demonstrations by training BC policies and evaluating them on a timber assembly task, demonstrating that GELLO-collected data is effective for robot arms to learn construction-relevant manipulation tasks.

2 Related Work

IL has been increasingly adopted in construction robotics as a practical route to learning task policies directly from expert behavior. Prior work has applied IL to train robots for a range of construction-relevant tasks, such as panel installation and related manipulation tasks [6, 7], wheel loader bucket filling [17], and tower crane lift path planning [18]. In many of these systems, IL is instantiated as behavioral cloning (BC), where a policy is learned via supervised learning to map observations to expert actions. The BC policy's performance depends strongly on both the quantity and quality of demonstration data: limited coverage and inconsistent demonstrations can induce distribution shift at test time, causing errors to compound during closed-loop execution [19]. Moreover, temporally noisy or non-smooth action sequences can produce jerky control outputs and unstable motion when rolled out on robots, degrading the overall policy performance [20].

However, the demonstration collection process is often described briefly as implementation details compared to other methodological components (e.g., network architectures and learning algorithms), and at times demonstration collection details are not fully specified [17]. For teleoperating robot arms to demonstrate construction operations, common interfaces rely on VR controllers [7], hand gestures [15], and haptic devices [14]. These approaches are typically end-effector–centric: a controller pose (or inferred hand pose) is mapped to a desired end-effector pose, and then inverse kinematics is used to compute joint commands for execution. While intuitive for specifying tool motion, end-effector–centric teleoperation can provide limited awareness and control of the robot's full-arm configuration, leading to demonstrations with undesirable postures (e.g., near joint limits) in construction workspaces. Motivated by these limitations, we adopt kinematic-isomorphic teleoperation and reproduce GELLO [16], which uses an embodied, kinematically matched manipulator to expose whole-arm posture more directly and intuitively, aiming to improve demonstration quality and data collection efficiency.

3 Methodology

In this section, we present an end-to-end framework that starts with robotic hardware assembly (i.e., GELLO) and low-level control via teleoperation for demonstration collection, and proceeds to offline policy learning via BC and policy inference on the real robot. Our framework consists of three stages (as illustrated in Figure 1): First, we assemble a 3D-printed robotic manipulator with a mechanical structure kinematically matched with the target robot used for task execution (i.e., xArm 7). Second, we collect expert demonstrations by teleoperating the robot using the manipulator. During each demonstration, state–action pairs of robot joints are logged synchronously. Third, we implement BC, specifically DP, for policy learning from the collected demonstrations. Finally, we directly deploy the learned policy on the robot and test its performance.

3.1 Stage 1: Manipulator hardware assembly

We follow GELLO [16] and build a kinematic-isomorphic teleoperation manipulator that matches the 7-DoF structure of the target robot, as shown in Figure 1(a). The manipulator has seven serial-connected passive servo joints, each aligned with one of the robot's seven revolute joints. This ensures intuitive joint-to-joint motion mapping. This joint-space mapping eliminates the need for an explicit Denavit–Hartenberg (DH) model, as motion is directly transferred between the joints instead of relying

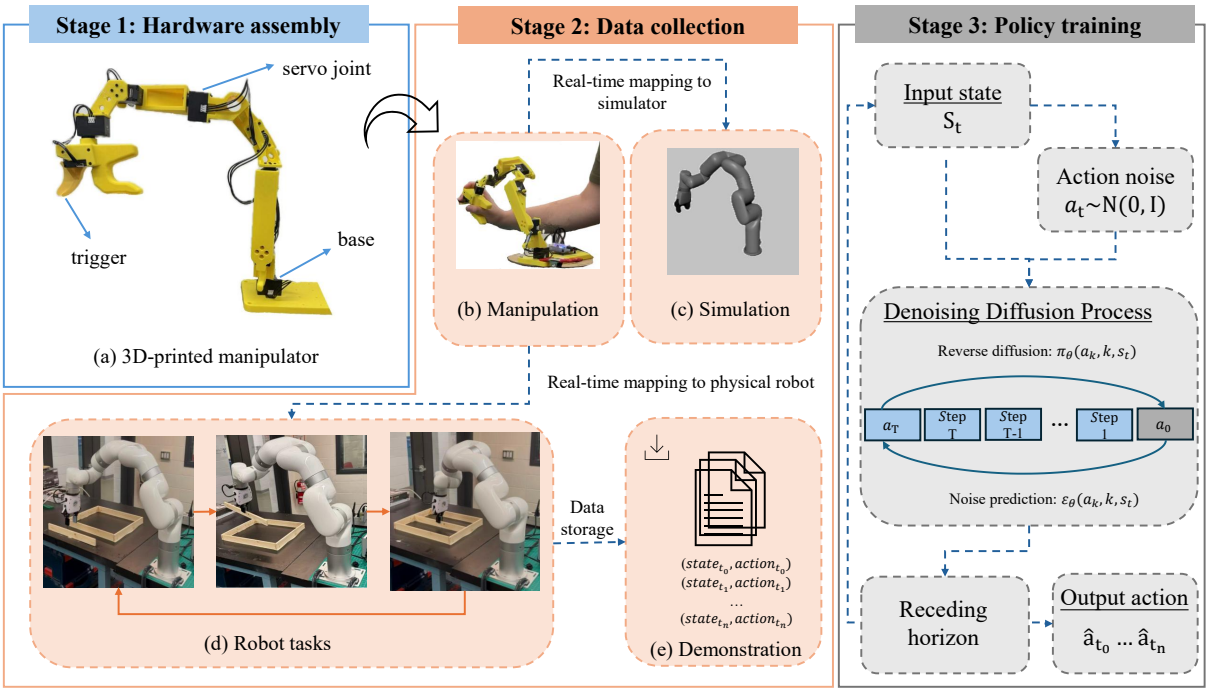

Figure 1. The overall stages of the proposed approach.

on forward or inverse kinematics. The eighth servo motor is mounted beside the trigger and continuously reads the trigger's rotation angle that is normalized to the range $[0, 1]$. This angle is transmitted to the target robot as its gripper control signal, actuating the opening and closing of the end-effector. Because passive joints tend to sag into undesirable configurations under gravity, we install tension rubber bands as simple joint regularizer to provide restoring torques, biasing the device toward a natural home posture aligned with the robot and reducing awkward poses and collisions.

3.2 Stage 2: Data collection via teleoperation

After assembling the manipulator, we implement a robot teleoperation system where we control the manipulator to teleoperate a virtual robot arm in MuJoCo [21] and a physical robot arm in the real world simultaneously. During teleoperation, joint data from the manipulator (i.e., GELLO) are transmitted via serial communication and mapped in real time to the simulated robot (i.e., xArm 7 in Mujoco), as shown in Figure 1(b) and Figure 1(c). For physical robot (i.e., xArm 7) operation, the host computer reads the manipulator joint data via serial communication. It maps this data to the controller through TCP protocol, which relays the control command to the physical robot. In both the virtual environment and the real world, the robots are synchronized with the manipulator at each timestep.

We utilize the manipulator to teleoperate the target robot for collecting expert demonstrations of robot tasks. For illustration purposes, we ground the proposed framework to the construction task of picking a timber truss and placing it in the middle of the frame. The base of the manipulator is secured on the operating table. The participant repeatedly demonstrates the task using the manipulator from a fixed observation perspective.

Our example task is decomposed into four segments, as seen in Figure 2. First, the demonstrator maneuvers the manipulator to guide the robot arm from its initial position toward the truss. Second, the demonstrator gradually pulls the trigger to control the robot arm to grasp the truss. Third, the demonstrator transfers the truss towards the frame and places it vertically into the center of the frame. Finally, the demonstrator operates the manipulator to guide the robot arm back to its initial position for reset. The trajectories of the robot arm (i.e., sequences of state-action pairs) are recorded during teleoperation and only successful teleopereated rollouts are saved as demonstration data for subsequent policy learning.

3.3 Stage 3: Policy learning via behavioral cloning

After collecting and curating the demonstration dataset, we train robot policies with BC. We adopt Diffusion Policy [11] as the backbone of the neural network due to its strong performance on contact-rich and long-horizon manipulation tasks. In contrast to Multi-Layer Perceptron (MLP)-based BC methods, which typically regress a single next action from the current observation and struggles with multimodal distributions in expert behavior, DP mod-

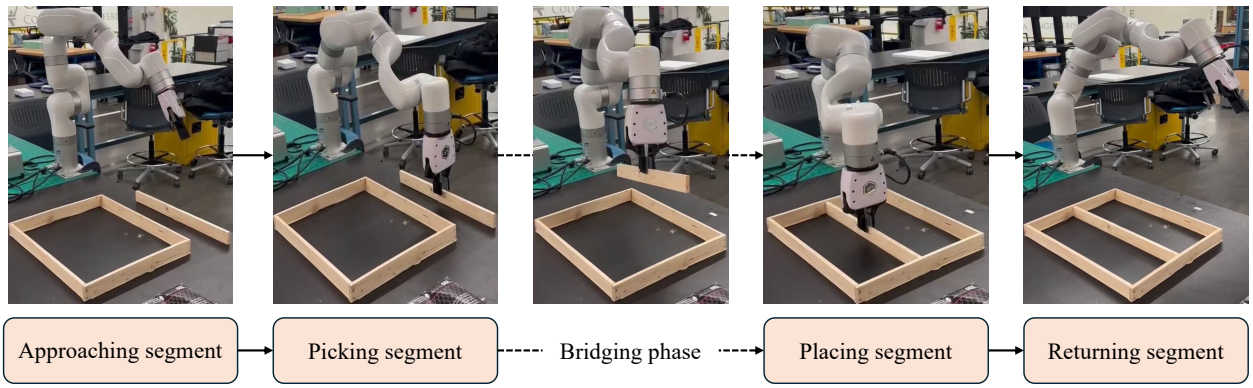

Figure 2. Four segments of each demonstration.

Algorithm 1 Diffusion Policy Training and Execution

Require: Demonstration dataset $\mathcal{D} = \{(s_i, \Delta q_i)\}_{i=1}^{N}$
Ensure: Diffusion policy parameters θ

procedure TrainDiffusionPolicy($\mathcal{D}$)
  Initialize diffusion policy parameters θ
  for each training iteration **do**
    Sample state–action pair $(s_i, \Delta q_i) \sim \mathcal{D}$
    Sample diffusion timestep $k \sim \mathcal{U}(1, T)$
    Sample noise $\epsilon \sim \mathcal{N}(0, I)$
    Generate noisy action
    $\Delta q_k = \sqrt{\bar{\alpha}_k}\,\Delta q_0 + \sqrt{1-\bar{\alpha}_k}\,\epsilon$
    Predict noise $\hat{\epsilon} = \epsilon_\theta(\Delta q_k, k, s_t)$
    Update θ by minimizing the loss
    $\mathcal{L} = \mathbb{E}_{k,\epsilon}\left[\|\hat{\epsilon} - \epsilon\|^2\right]$
  end for
end procedure

— Rollout Execution —

procedure ExecutePolicy(θ)
  for each control timestep t **do**
    Observe current state s_t
    Initialize action noise $\Delta q_T \sim \mathcal{N}(0, I)$
    for $k = T, T-1, \ldots, 1$ **do**
      Denoise action:
    $\Delta q_{k-1} \sim \pi_\theta(\Delta q_k, k, s_t)$
    end for
    Obtain predicted action sequence $\widehat{\Delta q}_{0:H-1}$
    Execute first action $\widehat{\Delta q}_0$ ▷ Receding horizon
  end for
end procedure

els the conditional action distribution using a generative denoising process and is commonly trained to predict short action chunks over multiple future steps. This formulation better captures the variability across demonstrations and reduces the tendency to average trajectories, resulting in more coherent rollouts in closed-loop control.

As illustrated in Figure 1, the diffusion policy generates actions by performing conditional denoising in the action space. At each control timestep, the current robot state s_t (i.e., joint position in our case) is provided as the conditioning input. An initial action is sampled from Gaussian noise, which is then iteratively denoised through a reverse diffusion process parameterized by the policy network. We choose the joint position change in each timestep as our action, which is defined as $a_t = \Delta q_t$.

DP consists of an offline training phase and an online execution phase, as summarized in Algorithm 1. During training, the policy learns to predict the injected Gaussian noise at randomly sampled diffusion steps using a mean squared error objective. During online execution, the policy repeatedly generates a short-horizon action sequence via conditional denoising and executes only the first predicted joint movement. Then the horizon shifts and the workflow repeats until the task is completed.

4 Experiment and Results

We design experiments to (1) demonstrate the feasibility of collecting high-quality expert demonstrations using GELLO, and (2) evaluate the performance of policies trained on the demonstrations, and (3) evaluate how policy performance scales with the number of demonstrations.

4.1 Experiment setup

We use a 7-degree-of-freedom (DoF) xArm 7 robot arm to execute timber assembly. Across demonstrations, the initial pose of the timber is perturbed by up to ±1 cm, while the robot base, manipulator base, and timber frame remain fixed. During demonstration collection, the human operator stands at a fixed location and avoids unnecessary body motion to reduce unintended robot actions. The dimensions of the timber truss are 39.5 cm×1.0 cm×3.8 cm. The dimensions of the frame are 41.6 cm × 33.8 cm × 3.8 cm. And the tolerance of assembly is 0.5 cm.

To evaluate this long-horizon task in a structured manner, we decompose each episode into four segments: *approach*, *pick*, *place*, and *return*, as shown in Figure 2. We define segment success criteria as follows: **Approach**: the

(a) Failure case 1 (b) Failure case 2 (c) Failure case 3 (d) Singularity case

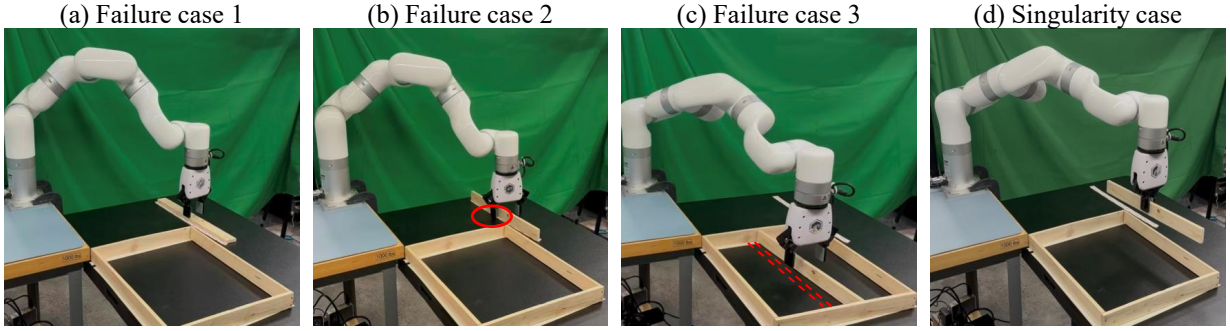

Figure 3. (a) The gripper knocks over the timber truss during picking; (b) The gripper collides with the table, triggering an emergency stop; (c) The object is placed at an incorrect location; (d) Singularity scenario encountered during keyboard-based teleoperation.

robot moves from the initial pose to a pre-grasp pose above the timber center; **Pick**: the robot grasps the timber and lifts it clear of the table without triggering a safety stop (e.g., due to collisions with the table or the truss frame); **Place**: the robot transfers the timber into the frame and places it vertically at the target location without triggering a safety stop due to collisions with the truss frame or the table; **Return**: the robot releases the timber and retreats to the initial pose without triggering a safety stop.

4.2 Results

4.2.1 Teleoperation Comparison

We compare the performance of GELLO with a conventional teleoperation interface (keyboard) for this task. The keyboard-based control enables linear position and orientation control of the robot's end-effector through command inputs. For the complete four-segment task, Table 1 summarizes the success rates and execution time over 10 teleoperation trials for both interfaces.

As shown in Table 1, the keyboard-based teleoperation shows lower success rate compared to GELLO. Failure cases are more likely to occur during the picking and placing segments. Figure 3 (a), (b), (c) illustrates several representative failure scenarios.

In addition, the average teleoperation time with GELLO is significantly lower compared to keyboard-based teleoperation, indicating higher efficiency in both task execution and data collection. One key reason is that joint-limit conditions are more likely to occur during linear end-effector navigation under keyboard-based teleoperation, which increases the difficulty of operation. When such situations arise, the operator must spend additional time exploring alternative feasible configurations, resulting in increased teleoperation time. Figure 3 (d) illustrates singularity scenarios that occur under keyboard-based teleoperation, whereas Figure 4 shows that GELLO allows the operator to naturally avoid such configurations.

Table 1. Teleoperation results: GELLO vs keyboard.

Trial	GELLO	Keyboard
Trial 1	39 s	Fail
Trial 2	38 s	Fail
Trial 3	37.27 s	165.56 s
Trial 4	36.87 s	Fail
Trial 5	39.74 s	163.15 s
Trial 6	43.04 s	158 s
Trial 7	39.02 s	160 s
Trial 8	43.40 s	Fail
Trial 9	39.32 s	190 s
Trial 10	43.94 s	176 s
Success rate	100%	60%
Average time	39.96 s	168.79 s

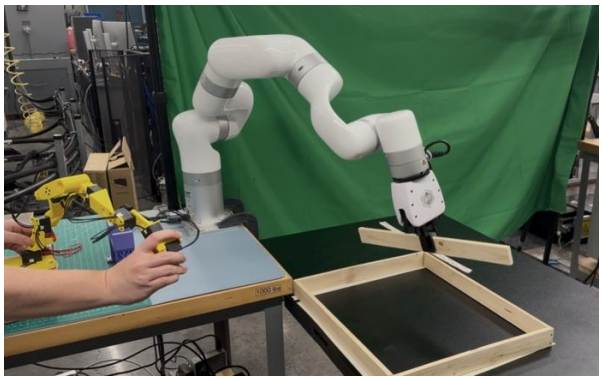

Figure 4. GELLO-based teleoperation naturally avoids singular configurations in Figure 3 (d).

4.2.2 Policy Performance

We use GELLO to collect 50 demonstrations in total and utilize the entire dataset to train the diffusion policy. We then test the learned policy by running 10 rollouts on the xArm 7 robot. The result shows that the robot arm successfully completes the timber assembly task in all trials, achieving a 100% success rate. Additionally, we observe that the robot trajectory generated by the learned policy

is smoother than demonstrations. This improvement is attributed to the policy's ability to eliminate unnecessary motions and reduce shaking during operations.

Furthermore, we curate smaller training datasets by randomly selecting episodes from the collected demonstrations. Specifically, We consider training sets consisting of 20%, 40%, 60%, 80%, and 100% of the complete dataset, which includes 10, 20, 30, 40, and 50 demonstrations, respectively. For each smaller training set, we train a policy from scratch and evaluate it over 10 rollouts. We break down the performance of each policy according to segments, as shown in Table 2.

As seen in Table 2, when 40 expert demonstrations are used for training, failures begin to appear in the placing segment, where the timber truss is not precisely positioned at the center of the target frame. Nevertheless, the robot's motion trajectory remains consistent, which means that the execution is not stopped by collisions or severe oscillations. Therefore, the robot is still able to execute the remaining task sequence of returning.

When the number of expert demonstrations is reduced to 30 or fewer, the robot arm's trajectories encounter safety stops due to collisions or excessive oscillation. In these cases, the robot fails to return to the initial position after the placing segment. This behavior is primarily caused by imprecise policy during the critical transition when the robot slowly places the timber, opens the gripper, and lifts the end-effector upward. The policy struggles to effectively coordinate these interrelated actions, resulting in either unintended contact with the table or excessive lifting motions that trigger emergency stop mechanisms. This situation is illustrated by the return segment success rate dropping to 0%, even when the placing segment is completed successfully.

When the number of expert demonstrations is extremely limited (i.e., only 20 or 10 demonstrations), the robot arm is still able to successfully approach the timber truss from the initial configuration, resulting in a success rate of 100% in approaching. However, both the picking and placing segments show significantly reduced performance, with success rates below 50%. Specifically, during the placing segment, the final placement location shows a significant deviation from the center of the target frame. Additionally, failures in the picking segment occurre more frequently, as unintended contact with the table leads to interruptions in the trajectory. These results indicate that while the approaching behaviors can be learned from a limited number of demonstrations, precise and contact-rich manipulation behaviors require more number of expert demonstrations.

5 Discussion and Conclusion

This paper presents an end-to-end framework for collecting robot arm demonstrations using a teleoperation manipulator with a kinematic structure consistent with that of the target robot arm, and for learning autonomous robot policies from the collected data for manipulation tasks in construction. Through a robotic construction task involving timber truss assembly, we validate the feasibility and effectiveness of the proposed demonstration collection and policy learning framework.

Experiment results illustrate that, compared to conventional teleoperation interfaces, GELLO provides a more intuitive and physically grounded interaction paradigm by directly conveying joint limitations to the operator through its kinematically aligned structure. This implicit feedback allows the operator to perceive and avoid infeasible configurations during manipulation, leading to improved robustness and efficiency in task execution. Besides, when a sufficient number of expert demonstrations are collected using the proposed pipeline, a DP can be successfully trained to enable the robot arm to complete the task autonomously with smooth and stable motion trajectories. Compared with the expert demonstrations, the learned policy produces trajectories with reduced unnecessary motion and oscillations, highlighting the advantage of DP in modeling multimodal demonstration data and generating temporally consistent actions during closed-loop execution.

Furthermore, our segment-wise evaluation reveals how the quantity of demonstration affects the performance of the policy in different phases of the task. The reaching segment remains strong even when there are few demonstrations available. However, the picking segments decline dramatically as the number of demonstrations decreases. The placement and return segment is particularly at risk of failure because it relies heavily on precise timing and coordination within safety constraints. These results indicate that although DP can model complex behaviors, their effectiveness ultimately depends on sufficient demonstration coverage, especially for contact-rich and precision-critical manipulation tasks.

From the perspective of failure cases, we observe and evaluate failure cases on each segment. In the picking segment, when the dataset size is reduced to 30 demonstrations, the policy begins to occasionally fail to establish a reliable grasp on the truss. With further reduction to extremely small datasets (i.e., 10 and 20 demonstrations), the grasp pose exhibits large deviations, resulting the gripper to collide with the table and trigger an emergency stop. Such early-stage safety stops directly prevent task completion and consequently lead to failures in subsequent stages. In the placement segment, when the dataset size is reduced to 30 demonstrations, the placement accuracy degrades and the truss is not fully assembled into the frame (e.g., noticeable offset or partial placement outside the frame). When the number of demonstrations is further reduced, collisions during the picking segment occur

Table 2. Segment-wise success rates under different numbers of expert demonstrations.

# of demos	Approaching segment	Picking segment	Placing segment	Returning segment	Overall task
50	100% (10/10)	100% (10/10)	100% (10/10)	100% (10/10)	100% (10/10)
40	100% (10/10)	100% (10/10)	80% (8/10)	100% (10/10)	80% (8/10)
30	100% (10/10)	70% (7/10)	50% (5/10)	0% (0/10)	0% (0/10)
20	100% (10/10)	50% (5/10)	10% (1/10)	0% (0/10)	0% (0/10)
10	100% (10/10)	20% (2/10)	0% (0/10)	0% (0/10)	0% (0/10)

more frequently, resulting in the absence of valid placing and consequently leading to a further decrease in success rate of this segment. In the returning segment, once the dataset size drops to 30 demonstrations, all test trials fail after placing, as the robot knocks into the table during the release-and-lift transition, resulting in a safety stop and preventing the arm from returning back to the initial pose.

Future work will integrate onboard cameras on the robot arm to provide visual observations during execution. Incorporating visual feedback would enable the proposed framework to learn visuo-motor policies and thus generalize to more diverse and complex tasks under richer scene variation. In addition, the approach can be extended to bimanual manipulation by teleoperating two robot arms with paired manipulators, enabling more precise, coordinated, and dexterous construction operations.

References

[1] Mohammed A. Albattah, Paul M. Goodrum, and Timothy R. B. Taylor. New metric of workforce availability among construction occupations and regions. *Practice Periodical on Structural Design and Construction*, 24(2): 04019003, 2019. doi:10.1061/(ASCE)SC.1943-5576.0000413. URL https://doi.org/10.1061/(ASCE)SC.1943-5576.0000413.

[2] Ahmed Shiha and Islam H. El-adaway. Localized multivariate statistical assessment of united states construction labor shortages. *Journal of Construction Engineering and Management*, 151(8): 04025097, 2025. doi:10.1061/JCEMD4.COENG-15702. URL https://ascelibrary.org/doi/abs/10.1061/JCEMD4.COENG-15702.

[3] Abhishek Bhargava, Panagiotis Ch. Anastasopoulos, Samuel Labi, Kumares C. Sinha, and Fred L. Mannering. Three-stage least-squares analysis of time and cost overruns in construction contracts. *Journal of Construction Engineering and Management*, 136(11): 1207–1218, 2010. doi:10.1061/(ASCE)CO.1943-7862.0000225. URL https://doi.org/10.1061/(ASCE)CO.1943-7862.0000225.

[4] Associated General Contractors of America and NCCER. 2025 workforce survey analysis, 2025. URL https://www.agc.org/sites/default/files/users/user21902/2025%20Workforce%20Survey%20Analysis%20(3).pdf.

[5] Daniel Castro-Lacouture. *Construction Automation and Smart Buildings*, pages 1035–1053. Springer International Publishing, Cham, 2023. ISBN 978-3-030-96729-1. doi:10.1007/978-3-030-96729-1_48. URL https://doi.org/10.1007/978-3-030-96729-1_48.

[6] Ci-Jyun Liang, Vineet R. Kamat, and Carol C. Menassa. Teaching robots to perform quasi-repetitive construction tasks through human demonstration. *Automation in Construction*, 120:103370, 2020. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2020.103370. URL https://www.sciencedirect.com/science/article/pii/S092658052030950X.

[7] Lei Huang, Zihan Zhu, and Zhengbo Zou. To imitate or not to imitate: Boosting reinforcement learning-based construction robotic control for long-horizon tasks using virtual demonstrations. *Automation in Construction*, 146:104691, 2023. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2022.104691. URL https://www.sciencedirect.com/science/article/pii/S0926580522005611.

[8] Lei Huang, Weijia Cai, Zihan Zhu, and Zhengbo Zou. Dexterous manipulation of construction tools using anthropomorphic robotic hand. *Automation in Construction*, 156:105133, 2023. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2023.105133. URL https://www.sciencedirect.com/science/article/pii/S092658052300393X.

[9] Aleksandra Anna Apolinarska, Matteo Pacher, Hui Li, Nicholas Cote, Rafael Pastrana, Fabio Gramazio, and Matthias Kohler. Robotic assembly of timber joints using reinforcement learning. *Automation in Construction*, 125:103569, 2021. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2021.103569. URL https://www.sciencedirect.com/science/article/pii/S0926580521000200.

[10] Kangkang Duan and Zhengbo Zou. Safety-constrained deep reinforcement learning control for human–robot collaboration in construction. *Automation in Construction*, 174:106130, 2025. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2025.106130. URL https://www.sciencedirect.com/science/article/pii/S0926580525001700.

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

[12] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.016. URL https://www.roboticsproceedings.org/rss19/p016.html.

[13] Ken Goldberg. Good old-fashioned engineering can close the 100,000-year "data gap" in robotics. *Science Robotics*, 10(105):eaea7390, 2025. doi:10.1126/scirobotics.aea7390. URL https://www.science.org/doi/abs/10.1126/scirobotics.aea7390.

[14] Qi Zhu, Tianyu Zhou, and Jing Du. Haptics-based force balance controller for tower crane payload sway controls. *Automation in Construction*, 144:104597, 2022. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2022.104597. URL https://www.sciencedirect.com/science/article/pii/S0926580522004678.

[15] Kangkang Duan and Zhengbo Zou. Morphology agnostic gesture mapping for intuitive teleoperation of construction robots. *Advanced Engineering Informatics*, 62:102600, 2024. ISSN 1474-0346. doi:https://doi.org/10.1016/j.aei.2024.102600. URL https://www.sciencedirect.com/science/article/pii/S1474034624002489.

[16] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163, 2024. doi:10.1109/IROS58592.2024.10801581. URL https://ieeexplore.ieee.org/document/10801581.

[17] Daniel Eriksson and Reza Ghabcheloo. Comparison of machine learning methods for automatic bucket filling: An imitation learning approach. *Automation in Construction*, 150:104843, 2023. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2023.104843. URL https://www.sciencedirect.com/science/article/pii/S0926580523001036.

[18] Zikang Wang, Chun Huang, Boqiang Yao, and Xin Li. Integrated reinforcement and imitation learning for tower crane lift path planning. *Automation in Construction*, 165:105568, 2024. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2024.105568. URL https://www.sciencedirect.com/science/article/pii/S0926580524003042.

[19] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), April 2017. ISSN 0360-0300. doi:10.1145/3054912. URL https://doi.org/10.1145/3054912.

[20] I-Chun Arthur Liu, Shagun Uppal, Gaurav S. Sukhatme, Joseph J Lim, Peter Englert, and Youngwoon Lee. Distilling motion planner augmented policies into visual control policies for robot manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 641–650. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/liu22b.html.

[21] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi:10.1109/IROS.2012.6386109.