Exploiting Physics and Data for Skill Acquisition from Virtual Reality-based Demonstrations for Construction Tasks

Wangyi Liu[1], Dasharadhan Mahalingam[1], Ci-Jyun Liang[2], and Nilanjan Chakraborty[1]

***Abstract*— In this paper, we present a novel method of manipulation skill acquisition for performing construction activities. We show that construction activities like building a wall can be performed by iteratively repeating manipulation tasks that are represented as a sequence of constant screw motions. Our approach involves setting up a simulated construction activity in a Virtual Reality (VR) environment, where the user can provide demonstrations of the object manipulation skills needed to perform the construction activity. We then exploit the screw geometry of motion to approximate the demonstrated motion as a sequence of constant screw motions. For performing the construction activity, we generate the sequence of manipulation task instances and then compute the joint space motion plan corresponding to each instance using screw linear interpolation (ScLERP). We evaluate our framework by performing the brick wall building activity using a single demonstration of the pick-and-place manipulation skill in VR and executing the activity using a 7 degree-of-freedom (7-DoF) robot in simulation. Through these experiments, we show that our approach is robust to building different types of walls (i.e., walls with different brick layouts) using just a single demonstration of picking and placing a brick over another brick.**

I. Introduction

Robotic manipulation plays an increasingly significant role across a multitude of applications ranging from assistive robotics to industrial automation. In particular, within the construction industry, there is a high demand for robots to perform various tasks involving high precision and complexity, such as bricklaying, installing ceiling tiles, beam alignment, and component assembly. These tasks are challenging due to the kinematic constraints on the robot's end effector during motion. For instance, in bricklaying, each brick must be placed with its bottom face parallel to the ground and correctly oriented relative to the bricks in lower layers (see Figure 1). The whole activity of bricklaying is in essence repeating the motion of picking up a brick and laying it on the floor or on top of another brick. Thus, in principle, if the robot knows how to lay one brick and if it is given the layout of the bricks in the wall, it should be able to build the whole wall. In this paper, *we study this problem of performing an activity that consists of repetitions of the same task through the example of bricklaying*.

[1]The authors are with the Department of Mechanical Engineering, Stony Brook University, Stony Brook, NY, USA. {wangyi.liu, dasharadhan.mahalingam, nilanjan.chakraborty}@stonybrook.edu.

[2]The author is with the Department of Civil Engineering, Stony Brook University, USA. ci-jyun.liang@stonybrook.edu.

This work was partially supported by the US Department of Defense through ALSRP under award No. HT94252410098, a SBU OVPR award, and a SBU LINCATS award.

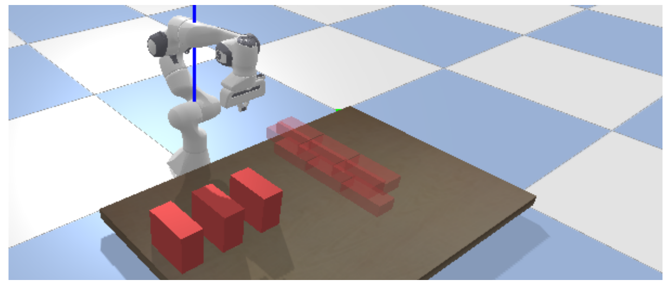


Fig. 1: Construction Activity Setup: Construction of a three-layer wall with a total of nine bricks in a simulation environment. The bricks are stacked in a pile initially (solid red bricks on the left of the image). The generated task instances showing the goal poses of the bricks for constructing a wall, visualized as translucent bricks on the right side of the image.

We consider the bricklaying problem in the context of programming by demonstration [1], where we assume that the demonstration is given in a virtual reality (VR) environment. Note that there are a variety of methods for acquiring demonstrations for robots, namely teleoperation, kinesthetic demonstration, VR-based demonstrations, demonstrations from human workers, and also video demonstrations. Teleoperation requires additional hardware, and kinesthetic demonstrations may be hard to acquire because it is difficult to set up a realistic physical construction scenario. Furthermore, human worker demonstrations are hard to obtain. Video demonstrations do not give direct information of the motion of the objects in $SE(3)$ (the group of rigid body motions). VR-based demonstrations, although they require some additional hardware, namely, a VR headset, are easy and cheap to obtain and give the motion of the object directly. The motion of the object which is a curve in $SE(3)$ contains the constraints that characterize the motion (which we model as one-parameter subgroups of $SE(3)$ or constant-screw motions) [2].

Contributions: We present a novel three-step approach for completing complex construction tasks: (a) Given a specific task, we build a simple environment in the virtual reality (VR) environment, move the object with the controller to finish the task, and collect the object's path in $SE(3)$. (b) We segment the motion of the end effector in the task space into a sequence of constant screws and extract the essential constraints based on the object poses. (c) For the new task instances that arises for completing the activity, we replicate the segmented screws with the new object poses. Then, we compute the motion plan based on ScLERP [3], which automatically ensures that the constant screw constraints embedded in the demonstrated motion are satisfied. We use the example of bricklaying to illustrate our approach.

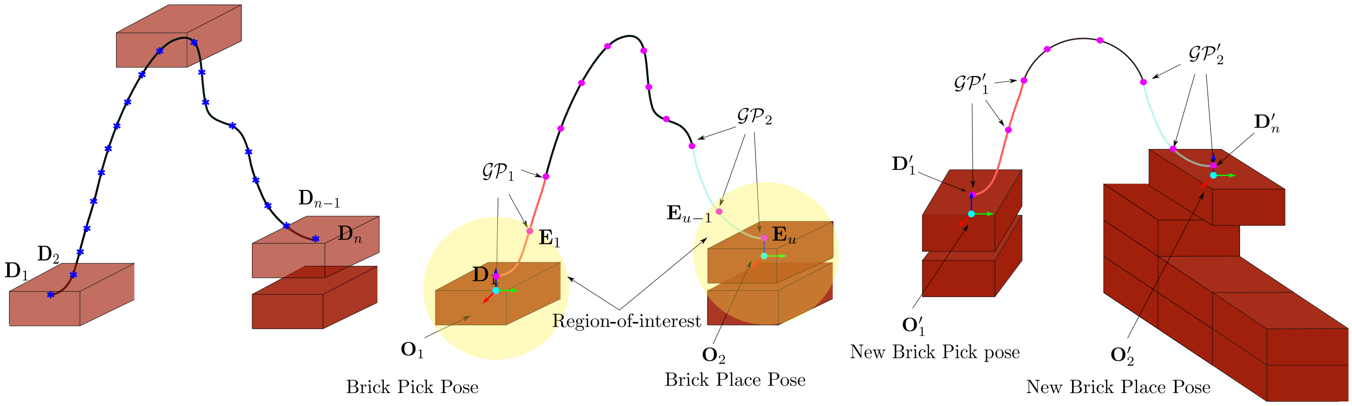


Fig. 2: Schematic Sketch of Motion Estimation for Building a Wall: Left - The collected demonstrations $\mathbf{D}$ in VR which consists of a sequence of $SE(3)$ poses which are represented with blue markers. **Center** - Segmenting the demonstration $\mathbf{D}$ as a sequence of constant screw motions, $\{\mathbf{D}_1, \mathbf{E}_1, \ldots, \mathbf{E}_u\}$, the *"Key Segments"* ($\mathcal{GP}_1$ and $\mathcal{GP}_2$) are determined based on the region-of-interest centered at the initial and final object poses, $\mathbf{O}_1$ and $\mathbf{O}_2$ respectively. **Right** - Given new initial and final object poses, $\mathbf{O}'_1$ and $\mathbf{O}'_2$, the *"Guiding Poses"* $\mathcal{GP}'_1$ and $\mathcal{GP}'_2$ are determined by transforming $\mathcal{GP}_1$ and $\mathcal{GP}_2$ with respect to the new object poses.

II. Related Work

Learning from Demonstration (LfD) [1] is a learning paradigm which focuses on using human guided demonstrations to train robots. In the context of automating construction, LfD approaches such as reinforcement learning (RL) and imitation learning (IL) have been employed with the aim of learning policies that can perform tasks autonomously [4]–[6]. These approaches require large amounts of data to train a successful policy. The ability to create highly controllable and flexible environments that are safer than real world construction environments have made VR environments an attractive platform for collecting such demonstrations. VR platforms have significant advantages in the development and simulation of realistic construction environments [7] thus advancing the field of autonomous construction [8]–[10]. However, such solutions employ RL and IL approaches that still require a lot of demonstrations for training [11]–[13]. In this work we focus on exploiting the screw geometry of motion to efficiently make use of demonstrations collected in a VR environment for automating construction tasks. The authors in [2] solve the problem of motion generation for complex manipulation tasks using kinesthetic demonstrations. This work builds upon those results to show that activities consisting of repetitive complex manipulation tasks can also be performed using demonstrations which are collected in a VR environment.

III. Mathematical Preliminaries

In this section, we present a brief review of the mathematical background required to understand this work.

Screw Displacement: Chasles-Mozzi theorem states that the general Euclidean displacement/motion of a rigid body from the origin $\mathbf{I}$ to $\mathbf{T} = (\mathbf{R}, \boldsymbol{p}) \in SE(3)$ can be expressed as a rotation θ about a fixed axis $\mathcal{S}$, called the *screw axis*, and a translation d along that axis. Plücker coordinates can be used to represent the screw axis by $\boldsymbol{\omega}$ and $\boldsymbol{m}$, where $\boldsymbol{\omega} \in \mathbb{R}^3$ is a unit vector that represents the direction of the screw axis, $\boldsymbol{m} = \boldsymbol{r} \times \boldsymbol{\omega}$, and $\boldsymbol{r} \in \mathbb{R}^3$ is an arbitrary point on the screw axis. Thus, the screw parameters are defined as $\boldsymbol{\omega}, \boldsymbol{m}, h, \theta$, where h is the pitch of the screw and θ is its magnitude. In general, for pure rotation and general screw motion, h is finite, while for pure translation, $h = \infty$. If $\mathbf{R} \neq \mathbf{I}$, then by using the standard procedure to obtain the rotation axis and magnitude from the rotation matrix $\mathbf{R}$, we can determine $\boldsymbol{\omega}$ and θ. The pitch is given by $h = \boldsymbol{\omega}^T \boldsymbol{v}$ and $\mathbf{m} = \boldsymbol{v} - h\boldsymbol{\omega}$, where $\boldsymbol{v} = \left[(\mathbf{I} - e^{\hat{\omega}\theta})\hat{\omega} + \theta\boldsymbol{\omega}\boldsymbol{\omega}^T\right]^{-1} \boldsymbol{p}$. If $\mathbf{R} = \mathbf{I}$, then the motion is pure translation, where $h = \infty$ and $\boldsymbol{m} = \mathbf{0}$ by definition. We can obtain θ and $\boldsymbol{\omega}$ from $\theta = ||\boldsymbol{p}||$ and $\boldsymbol{\omega} = \boldsymbol{p}/||\boldsymbol{p}||$. **A *constant screw motion* is a motion where the parameters $\boldsymbol{\omega}, \boldsymbol{m}$, and h stay constant throughout the motion.**

Given the screw parameters $\boldsymbol{\omega}, \boldsymbol{m}, h$, the screw displacement for a motion of magnitude θ can be obtained using the matrix exponential, $\mathbf{T} = e^{\widehat{\boldsymbol{\xi}}\theta}$. Here, $\widehat{\boldsymbol{\xi}} \in se(3)$ and $\boldsymbol{\xi} \in \mathbb{R}^6$ are the unit twist and unit twist coordinates associated with the motion. They are defined as,

$$\widehat{\boldsymbol{\xi}} = \begin{bmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{m} + h\boldsymbol{\omega} \\ 0 & 0 \end{bmatrix}, \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{m} + h\boldsymbol{\omega} \\ \boldsymbol{\omega} \end{bmatrix} \text{ for } h \neq \infty \quad (1)$$

$$\widehat{\boldsymbol{\xi}} = \begin{bmatrix} \mathbf{I} & \boldsymbol{\omega} \\ 0 & 0 \end{bmatrix}, \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{0} \end{bmatrix} \text{ for } h = \infty \quad (2)$$

Task Instance: Objects affecting the generation of motion plans for manipulation tasks are defined as task-related objects and the set $\mathcal{O} = \{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_u\}$ containing the $SE(3)$ poses of all the task-related objects is defined as a task instance.

Demonstration: A demonstration of a manipulation task is a sequence of $SE(3)$ poses $\mathcal{D} = \langle \mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_v \rangle$ that defines the motion of the manipulated object.

IV. Problem Statement

Consider that we are given a single demonstration $\mathcal{D}$ of an object manipulation skill that is required to perform an activity along with the corresponding task instance $\mathcal{O}$. The problem that we are trying to solve can be stated as: **Given a demonstration $\mathcal{D}$ of a manipulation task and its corresponding task instance $\mathcal{O}$, determine the sequence of task instances $\langle \mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_k \rangle$ and the joint space motion plan $\mathcal{M} = \langle \Theta_1, \Theta_2, \ldots, \Theta_l \rangle$ required to transfer**

the manipulation task to the new task instances for successfully completing an activity. For the brick wall building activity, $\mathcal{D}$ defines the motion that a brick goes through during the stacking process and $\mathcal{O} = \{\mathbf{O}_1, \mathbf{O}_2\}$ consists of the initial pose $\mathbf{O}_1$ and the final pose $\mathbf{O}_2$ of the brick. Coincidentally, $\mathbf{O}_1 = \mathbf{D}_1$ and $\mathbf{O}_2 = \mathbf{D}_v$.

The above problem can be solved by solving the following three three sub-problems:

Constraint Extraction: Given a demonstration $\mathcal{D}$, we need to extract the essential task constraints, i.e., the constraints that characterize the task and should be satisfied by all task instances.

Determination of Task Instances: For constructing a wall, we also need to determine the sequence of task instances which specify the order and pose at which the bricks are placed. We assume that we are given a pose $\mathbf{B} \in SE(3)$ denoting the starting pose of the wall and the specification of the wall in terms of straight, curved or corner along with the number of bricks in each layer (β), number of layers (α), and the horizontal offset between adjacent layers (δ). Using this information, we need to determine the sequence of task instances that need to be executed to complete the construction activity.

Motion estimation: After computing the sequence of task instances, we now have to determine the joint space motion plan that can successfully execute the motion while satisfying the extracted task constraints.

V. Solution Approach

Demonstration Acquisition: We collect demonstrations of manipulation tasks in a VR environment. The VR environment was built in Unreal Engine 4 (UE4) [14] and allows the user to interact with objects in the VR environment by means of a VR headset. We used the Meta Quest 2 [15] headset in our work. The environment is set up such that it contains two bricks, one that the user can pick up and manipulate and the other is placed at another location denoting the start of the wall. The user can provide a demonstration by picking up the movable brick and placing it next to the first brick. The motion of the brick that is being manipulated is recorded and used as the demonstration. This demonstration implicitly captures the constraints required to stack bricks without colliding with the neighboring bricks.

Extraction of Task Constraints: From the provided demonstration, we then extract the task constraints as a sequence of constant screw motion constraints. By following the approach proposed in [2], we are able to extract the motion invariants and are able to determine the object motion for a new task instance. These constraints are independent of the choice of coordinate frame and allow us to successfully transfer the extracted constraints to a new task instance.

Generation of Task Instances: In this work, we assume that the bricks are stacked in a known initial location. Consider that the dimension of the brick along its length, breadth and width are $\ell, b,$ and h respectively. Let ϵ_ℓ and ϵ_h be the spacing required between adjacent bricks in the same layer and adjacent layers. Given the starting pose of the wall, $\mathbf{B}$, depending on the type of wall, we can also determine the goal poses of the bricks as,

$$\mathbf{T}_{i,j} = \begin{cases} \mathbf{B}, & i = 1, j = 1 \\ \mathbf{T}_{i-1,1}\mathbf{Z}(h+\epsilon_h)\mathbf{X}(\Delta(i,\delta))\mathbf{R}(\theta), & i > 1, j = 1 \\ \mathbf{T}_{i,j-1}\mathbf{X}(\ell+\epsilon_\ell)\mathbf{R}(\theta), & i \geq 1, j > 2 \end{cases} \tag{3}$$

Where $\mathbf{X}(t)$ and $\mathbf{Z}(t)$ are $SE(3)$ transformations that represent translation along the vectors $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^{\mathbf{T}}$ and $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathbf{T}}$ respectively by a magnitude of t meters and $\mathbf{R}(t)$ is a $SE(3)$ transformation that represents pure rotation about the axis $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathbf{T}}$ by the magnitude t.

$$\mathbf{X}(t) = \left(\mathbf{I}, \begin{bmatrix}t & 0 & 0\end{bmatrix}^T\right), \quad \mathbf{Z}(t) = \left(\mathbf{I}, \begin{bmatrix}0 & 0 & t\end{bmatrix}^T\right) \tag{4}$$

$$\mathbf{R}(t) = \begin{bmatrix} \cos t & -\sin t & 0 & 0 \\ \sin t & \cos t & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Here, $\Delta(x, y)$ is a real valued function which determines the offset between bricks in adjacent layers and is defined as,

$$\Delta(x, y) = \begin{cases} y, & \text{if } x \text{ is even} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The curvature of the wall is determined by θ. For a straight wall, $\theta = 0$. Setting $\delta = 0$ results in the layers being aligned.

By modifying equation (3) for the case $i \geq 1, j > 2$ we can also determine the goal pose of the bricks for a corner wall,

$$\mathbf{T}_{i,j} = \mathbf{T}_{i,j-1}\mathbf{X}(\Delta(i+1, \ell'+\epsilon_\ell))\mathbf{Y}(\Delta(i, \ell'+\epsilon_\ell))\mathbf{R}(\theta) \tag{7}$$

with $\theta = 90^\circ$ at the corner and $\theta = 0^\circ$ everywhere else. Here $\ell' = (\ell + b)/2$, and $\mathbf{Y}(t)$ is an $SE(3)$ transformation that represents pure translation along the vector $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^{\mathbf{T}}$.

Equations (3) and (7) determine the goal pose of the brick given the indices i and j which denote which layer the brick is in and where the brick is positioned in a given layer respectively. Here $1 \leq i \leq \beta$ and $1 \leq j \leq \alpha$.

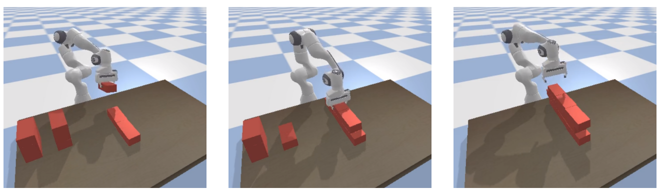
Using $\mathbf{T}_{i,j}$, we can now determine the sequence of the task instances. Each task instance $\mathcal{O}_k$ consists of the initial pose and the final pose of the brick. Since we assume that the bricks are stacked initial at a known location and the goal pose can be determined using $\mathbf{T}_{i,j}$, we can easily determine the sequence of task instances, $\langle \mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_k \rangle$.

Motion Generation: Once the sequence of task instances have been determined, we can transfer the constant screw constraints that were extracted from the demonstration by following approach proposed in [2]. This defines the motion of each brick in terms of a sequence of $SE(3)$ poses, $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_w$ where each pair of consecutive poses, $\mathbf{G}_i$ and $\mathbf{G}_{i+1}$ define a constant screw constraint. We can then determine a feasible grasp pose that would allow us to perform this motion using the approach proposed in [16] and then use Screw Linear Interpolation (ScLERP) combined with Jacobian pseudo-inverse to compute a motion plan in the joint space [3]. This ensures that the generated motion plan satisfies the extracted constant screw constraints.

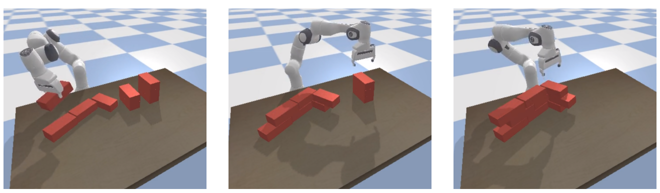
A schematic sketch describing the solution approach is shown in Figure 2.

VI. Experimental Results

In this section, we provide experimental results of performing the construction activity of building a wall using our proposed framework. We collected a single demonstration of the manipulation task in the VR environment and then transfer that demonstration to construct five walls, each of different specifications (Figures 1, 3). For more details please refer to the supplemental video (`https://youtu.be/6saerS5M68E`). All the experiments are carried out in the PyBullet simulation environment using a 7 DoF Franka Emika Panda robot.



(a)



(b)

Fig. 3: Brick Wall Construction: (a) Straight wall construction (b) Corner wall construction

For the brick size we chose standard USA-size with $\ell = 19.4\ cm$, $b = 9.2\ cm$, and $h = 5.4\ cm$. We conducted our approach with a single demonstration for 5 experiments:

(1) **Straight wall 1:** A three-layer wall of nine bricks is assembled parallel to the x-axis.
(2) **Straight wall 2:** Construction of a wall identical to the previous experiment but rotated 30° with respect to the x-axis.
(3) **Straight wall 3:** The wall is built parallel to the y-axis, again re-using the original pick-up pose and same configuration as experiment 1 and 2.
(4) **Curved wall:** A two-layer, 120° circular wall of six bricks is erected.
(5) **Corner wall:** A three-layer wall corner consisting of nine bricks (two intersecting runs of 6+3 bricks) is assembled.

VII. Conclusion

In this work, we propose a novel framework that extracts manipulation constraints from demonstrations which are collected in VR to complete complex construction activities. From a single task-space demonstration, our algorithm extracts coordinate-invariant screw constraints and reuses them to synthesize motion plans for multiple sub-tasks.

Currently, our work is tested in a simulation environment. We plan to verify the approach on real robots under real construction conditions. Although this paper focuses on brick laying, our method is general enough to apply to tasks like ceiling tile installation, which we want to study in the future. Currently, we assume a fixed-base manipulator with a limited workspace; future work will consider scenarios with a mobile manipulator performing the task.

References

[1] A. G. Billard, S. Calinon, and R. Dillmann, "Learning from humans," *Springer Handbook of Robotics*, pp. 1995–2014, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_74

[2] D. Mahalingam and N. Chakraborty, "Human-guided planning for complex manipulation tasks using the screw geometry of motion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7851–7857.

[3] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9480–9487.

[4] C.-J. Liang, V. R. Kamat, and C. C. Menassa, "Teaching robots to perform quasi-repetitive construction tasks through human demonstration," *Automation in Construction*, vol. 120, p. 103370, 2020.

[5] A. A. Apolinarska, M. Pacher, H. Li, N. Cote, R. Pastrana, F. Gramazio, and M. Kohler, "Robotic assembly of timber joints using reinforcement learning," *Automation in Construction*, vol. 125, p. 103569, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580521000200

[6] B. Belousov, B. Wibranek, J. Schneider, T. Schneider, G. Chalvatzaki, J. Peters, and O. Tessmann, "Robotic architectural assembly with tactile skills: Simulation and optimization," *Automation in Construction*, vol. 133, p. 104006, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092658052100457X

[7] K. Duan, Z. Zou, and T. Y. Yang, "Training of construction robots using imitation learning and environmental rewards," *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 40, no. 9, p. 1150–1165, Dec. 2024. [Online]. Available: https://doi.org/10.1111/mice.13394

[8] R. Li and Z. Zou, "Enhancing construction robot learning for collaborative and long-horizon tasks using generative adversarial imitation learning," *Advanced Engineering Informatics*, vol. 58, p. 102140, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034623002689

[9] J. Ho and S. Ermon, "Generative adversarial imitation learning," *CoRR*, vol. abs/1606.03476, 2016. [Online]. Available: http://arxiv.org/abs/1606.03476

[10] X. Wang, C.-J. Liang, C. Menassa, and V. Kamat, "Interactive and immersive process-level digital twin for collaborative human-robot construction work," *Journal of Computing in Civil Engineering*, vol. 35, 11 2021.

[11] H. You, Y. Ye, T. Zhou, and J. Du, "Force-based robotic imitation learning: A two-phase approach for construction assembly tasks," 2025. [Online]. Available: https://arxiv.org/abs/2501.14942

[12] K. Duan and Z. Zou, "Learning from demonstrations: An intuitive vr environment for imitation learning of construction robots," 2023. [Online]. Available: https://arxiv.org/abs/2305.14584

[13] L. Huang and Z. Zou, "Deep reinforcement learning-based construction robots collaboration for sequential tasks," in *Proceedings of the 1st Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2022)*, Philadelphia, PA, USA, May 2022, pp. 48–51.

[14] E. Games, "Unreal engine," https://www.unrealengine.com/, accessed: April 2025.

[15] Meta, "Quest," https://www.meta.com/quest/, accessed: April 2025.

[16] A. Patankar, K. Phi, D. Mahalingam, N. Chakraborty, and I. Ramakrishnan, "Task-oriented grasping with point cloud representation of objects," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.