ConstructSkill: Benchmarking Imitation Learning for Construction Manipulation with ManiSkill and Pretrained Encoders

Shyam Prasad Reddy Kaitha[1], Zhaofeng Hu[2], Hongrui Yu[1*], and Ci-Jyun Liang[2]

***Abstract*— Construction operates in different configurations, imposing great challenges for pre-programmed robots. As robot learning provides a better solution, limited benchmarks are usable for construction. The authors present ConstructSkill, an open simulation benchmark for construction manipulation, built on ManiSkill with a Franka Panda and RGB-only observations. Each task declares its assets, initial-state distribution and success criterion behind a fixed policy interface, and a scripted generator produces its demonstrations, minimizing policy generalization costs The reference policy is minimal by construction: a frozen ViT backbone with two linear heads, so that what varies across runs is the visual representation. Instantiated on brick pick-and-place, the reference study spans four backbones at ten training seeds each and four ablations of that policy at three. Representation quality governs grasping, Ten of Ten seeds against Three of Ten, while placement remains seed-dependent for every backbone and offline action error does not predict closed-loop success. The benchmark reports per-seed outcomes by default, and is the instrument we use next to ask whether vision–language–action (VLA) policies reduce that seed variance. `https://www.youtube.com/watch?v=tZJ3TwPIq2U`**

I. Introduction

Construction sites contain a large volume of physically demanding material-handling and assembly tasks, including positioning bricks and blocks, staging components at a fixture, repositioning materials that recur across a project but never in exactly the same configuration twice, and is poorly served by the fixed automation of factory settings [1]. Learning from demonstration is an attractive way to specify such operations, because a skilled worker can show the skill far more cheaply than an engineer can encode it, and construction-specific systems built on this premise have been demonstrated across a range of recurring site operations - ceiling tile installation [1], [4], excavator control [2], and material handover [3]. Whether such policies succeed reliably, measured as the rate at which they grasp and place across demonstrations on a fixed task is an empirical question, and one that has to be settled in simulation before it can be evaluated on site. Therefore, it is important to benchmark learning-based construction robots.

The benchmarks that drive progress in robot learning, such as RLBench [5], LIBERO [6] are useful because each fixes what is not under test, leaving the method as the only free variable, both are built around household and tabletop manipulation. We conducted a systematic review of 22 manipulation benchmarks, screening for construction-relevant tasks, and found none targeting construction operations, where object dimensions and receiving-structure geometry are binding constraints rather than incidental details. Construction manipulation is instead evaluated on bespoke per-paper setups [1], [2], [7], each with its own robot, object, and definition of success. Physics simulators for construction assemblies exist [8], but a simulator supplies dynamics, not tasks. The missing layer is the one above it, and no such benchmark exists for learning-based construction manipulation.

To address this issue, we proposed ConstructSkill. ConstructSkill is a simulation benchmark for construction settings, built on ManiSkill [9] and its SAPIEN backend, with a Franka Panda and RGB-only observations from a single fixed 224×224 view. Each task supplies its own assets, initial-state distribution, and success criterion behind a fixed policy interface, so a new operation can be added without changing the policy side. Each task is paired with a scripted demonstration generator with optional noise injection, so datasets are produced without human teleoperation. The suite is currently instantiated on brick pick-and-place in which placement is tolerance-bound insertion and is designed to accommodate further construction primitives.

The suite's reference policy a frozen pretrained backbone with two linear heads follows an approach with an established record: features from large image-pretrained models, used without fine-tuning, are competitive with representations learned from scratch on control data [10], and the most systematic comparison of such encoders found no single representation dominant across embodied tasks [11], a conclusion visible only because it was measured over a standard set of environments.

A reference study on the instantiated task spans 52 runs over eight configurations: four frozen ViT backbones at ten training seeds each, and four ablations of the reference policy at three. Representation quality governs grasping DINOv2 ViT-S/14 reaches the lift-proxy in 10 of 10 seeds against 3 of 10 for ImageNet-supervised ViT-Tiny/16, but placement remains seed-dependent for every backbone, and the configuration with the lowest offline action error fails at rollout in every seed. Most consequentially, at three seeds the placement ordering between the two strongest backbones was the reverse of what ten seeds show, a hazard documented in offline manipulation learning [12] so ConstructSkill makes multi-seed evaluation the default rather than an extra.

We contribute:

*Corresponding author.

[1]Dept. of Civil and Environmental Engineering, Virginia Tech, Blacksburg, VA, USA {skaitha, hryu42}@vt.edu

[2]Dept. of Civil and Environmental Engineering, Stony Brook University, Stony Brook, NY, USA {zhaofeng.hu, ci-jyun.liang}@stonybrook.edu

- an open, extensible simulation benchmark for learning-based construction manipulation, in which each task declares its assets, initial-state distribution and success criterion behind a fixed policy interface, so a new operation is added without touching the policy side;
- a per-seed evaluation protocol, and the reference study that shows why it is needed: offline action error does not predict closed-loop success, and at three seeds the placement ordering between the two strongest backbones reverses.
- scripted demonstration generators with optional noise injection, which produce image action datasets for any task in the suite without human demonstration, addressing a data bottleneck that construction robot learning otherwise meets through teleoperation.

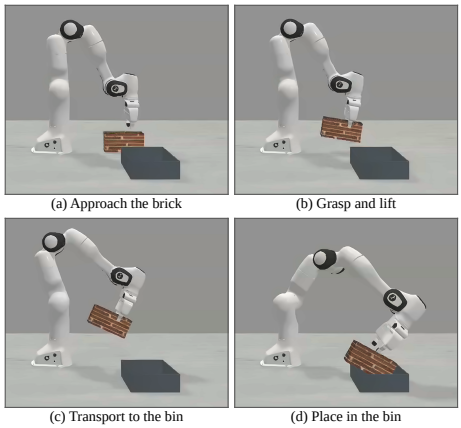

Fig. 1. Constructskill structure.Instantiated brick pick-and-place scene with the Franka Panda, the textured brick, the bin, and the fixed 224×224 RGB view.

II. Methodology

ConstructSkill is built on ManiSkill [9] and its SAPIEN backend. Every task places a 7-DoF Franka Panda with a parallel gripper in a scene, exposes a single fixed 224×224 RGB view as the sole observation, and is driven through `pd_ee_delta_pose`, of which the suite uses the translation and gripper channels. The Franka Panda [5], [6], [9] was chosen as the platform used by roughly two-thirds of the benchmarks screened above. Those three declarations asset, initial-state distribution, and success criterion are made on an environment class inheriting the shared machinery: robot placement a rest keyframe applied at every reset, target-region construction, and the containment test defining success so a new pick-and-place task requires only an asset-loading method and, where the layout differs, four geometry constants.

The task instantiated here is brick pick-and-place, registered as `ConstructionPickPlace-ViT-v1`. A textured box primitive of $0.21 \times 0.10 \times 0.065$ m must be moved into an open-topped bin of interior $0.25 \times 0.20 \times 0.15$ m with 5 mm walls. The camera is fixed at $[-1.0m, -1.3m, 0.9m]$, targets $[0.0m, -0.35m, 0.2m]$, and has a $\pi/2$ field of view. The environment's initial-state distribution randomises the object over ± 0.10 m in x and y and over a full turn of yaw; this task instance overrides it to a fixed pose due to Panda robot's jaw geometric limitations.

Each task declares a scripted multi-primitive controller alongside its assets, initial-state distribution and success criterion, since the manipulation strategy is task-specific while the data format is not. The brick generator is a six-phase state machine PREGRASP, DESCEND, GRASP, LIFT, CARRY, RELEASE driven by proportional control toward each phase's target, with distance-threshold transitions, a short dwell at GRASP, and a 0.40 m lift.

Generators optionally inject noise following DART [13]: zero-mean Gaussian noise with $\sigma = 0.01$ m per axis is added to the *executed* delta, clipped to ± 0.08 m, so a tail draw cannot displace the end-effector far enough to abandon the phase target while the label stored for that frame remains the clean control. TNoise is applied during PREGRASP, DESCEND, LIFT and CARRY and withheld at GRASP and RELEASE so contact is never perturbed. The released dataset is 80 demonstrations, approximately 37,000 image–action pairs. visits.

A pretrained ViT backbone is held frozen, in evaluation mode regardless of the training flag, and two linear heads read its pooled feature: one predicts a chunk of eight future translation deltas scaled by $0.1 \tanh(\cdot)$, the other eight gripper logits trained under binary cross-entropy. Four backbones are provided (ViT-Tiny/16 [16], DINOv2 ViT-S/14 [14], CLIP ViT-B/16 [15]), and a second ViT-Tiny variant using the $(0.5, 0.5, 0.5)$ normalization statistics of its AugReg checkpoint [16] in place of ImageNet statistics.

None was trained for control; the reference policy holds them frozen precisely to test the inherited assumption that features learned for image understanding suffice for manipulation [10], [11]. The three checkpoints span the pretraining objectives in common use: ImageNet label supervision [16], self-supervision [14], and image–text alignment [15]. Architecture family is common to all three, so a difference in rollout behaviour is attributable to the supervision signal. Control is closed-loop: each frame produces a fresh chunk, and the ensemble delta is issued through `pd_ee_delta_pose`.

At inference the policy predicts a fresh chunk every step and combines overlapping predictions by temporal ensembling as in ACT [17]: the last eight chunks are retained, and each chunk still covering the current step contributes with weight $\exp(-0.1k)$ for k steps since it was predicted. The gripper logit is thresholded at 0.5 and mapped to a binary variable of open or close.

III. Experimental Setup

The four backbones ViT-Tiny/16, DINOv2 ViT-S/14, CLIP ViT-B/16, and ViT-Tiny with AugReg normalisation statistics are each trained at ten seeds, 0–9. Four ablations of the reference policy, all on DINOv2 ViT-S/14, are each trained at three seeds, 0–2: without action chunking and temporal ensembling, with a regression gripper head replacing the BCE head, and without demonstration-time noise injection.

TABLE I

BACKBONES, TEN TRAINING SEEDS EACH (SEEDS 0–9).

	Seeds (n/10)		Mean episodes		Val. MSE
Backbone	Lift	Place	Lift	Place	($\times 10^{-4}$)
DINOv2 ViT-S/14	**10**	**7**	9.7	6.1	0.92 [0.885–1.01]
CLIP ViT-B/16	6	5	6.0	4.5	**0.82** [0.79–0.85]
ViT-Tiny/16	3	1	3.0	0.5	1.47 [1.42–1.62]
ViT-Tiny (AugReg)	2	1	2.0	1.0	1.45 [1.41–1.52]

Seeds (n/10) counts seeds meeting the criterion at least once; mean episodes averages all 10×10 rollouts; Val. MSE is the translation-chunk error on the held-out split, mean [range] over seeds.

TABLE II

ABLATIONS OF THE REFERENCE POLICY, THREE TRAINING SEEDS EACH.

	Seeds (n/**3**)		Mean episodes		Val. MSE
Configuration	Lift	Place	Lift	Place	($\times 10^{-4}$)
Baseline (3 seeds)	3	1	9.0	3.3	0.90
No action chunking	1	1	3.3	3.3	0.98 [0.90–1.11]
No temporal ensembling	1	1	3.3	3.3	0.90
Regression gripper head	3	2	10.0	6.7	0.90
No noise injection	**0**	**0**	**0.0**	**0.0**	**0.31**

All ablations use DINOv2 ViT-S/14. The baseline row is that backbone's seeds 0–2, matched to the ablations' seed count I.

All other settings are held fixed. The seed counts are deliberately asymmetric: three for the ablations against ten for the backbones so an ablation effect has to be large to be resolvable at all, and only one is.

Ten of the 80 demonstrations are held out, split by episode identifier, i.e, every eighth episode is held out, so the split does not coincide with any drift in generation order, and never by frame, because consecutive frames within an episode are near duplicates and a frame-level split leaks held-out states into the training set.

Two success metrics are used. *Lift-proxy* requires the brick to reach $z \geq 0.15\,\mathrm{m}$ at any step, isolating grasp and lift from placement. *Placement* requires the brick centre to lie within $0.125\,\mathrm{m}$ in x, $0.10\,\mathrm{m}$ in y and $0.10\,\mathrm{m}$ in z of the bin centre, with the gripper open, for five consecutive steps. The bounds are read from the bin geometry at call time rather than set by hand; the z bound admits a brick resting against the bin rim as well as one lying flat inside.

The brick spawns at a fixed pose and the robot resets to a fixed keyframe, so the ten evaluation episodes of a run repeat a single trajectory up to simulator non-determinism, and outcomes are correspondingly near-bimodal. Each seed therefore contributes close to a single sample.

Backbone comparisons are made on seeds succeeding out of ten and tested with Fisher's exact test, two-tailed, which computes the exact probability of the observed split under independence and does not rely on the large-sample approximation a chi-square test would require at this count. A small p therefore indicates that a difference of that size is unlikely to arise from which seeds happened to be drawn, rather than that one backbone is better in general.

IV. RESULTS

On brick pick-and-place, performance is reported over the task's two primitives, grasp–lift and placement (Section III). The four backbones separate on the first and not on the second. DINOv2 ViT-S/14 reaches the lift-proxy in 10 of 10 training seeds, against 3 of 10 for ImageNet-supervised ViT-Tiny/16 (Fisher exact, two-tailed, $p = 0.0031$), 2 of 10 for the AugReg-normalised variant ($p = 0.0007$), and 6 of 10 for CLIP ViT-B/16 ($p = 0.087$, reported as counts rather than as a separation at this sample size). Placement is achieved in 7, 5, 1 and 1 of 10 seeds over the same four backbones, with DINOv2 against CLIP at $p = 0.65$. Representation quality therefore governs the grasp primitive but does not carry through to the placement primitive, and Figure 2(a) shows why: outcomes are near-bimodal, and the seeds that fail placement are not the seeds that fail grasping. This is the reference policy's failure mode — perception suffices for grasping and does not suffice for placing.

Offline action error does not predict closed-loop success on either primitive. CLIP attains the lowest held-out validation MSE at every seed, 0.79–0.85×10^{-4} against DINOv2's 0.885–1.01×10^{-4}, and grasps in four fewer seeds. Figure 2(b) plots all forty (backbone, seed) pairs; there is no trend to fit.

Demonstration-time noise injection is necessary, and is the only ablation whose effect exceeds seed variance: 0 of 3 seeds on every metric against the baseline's 3 of 3 on the lift-proxy. It also posts the sweep's best validation loss, 0.31×10^{-4}, but validates on the noise-free dataset, so the success numbers are the comparable part. Noise-free training fits the demonstrated trajectory exactly and cannot recover from its own drift. No other ablation is resolvable at three seeds.

Two negative results are worth recording. Correcting ViT-Tiny's normalisation statistics to those of its AugReg checkpoint does not rescue it grasp-lift falls from 3 of 10 seeds to 2 of 10 so the weakness is in the features, not the preprocessing. Removing temporal ensembling changes grasp-lift in two of three seeds but leaves placement identical in all three.

The measurement result is the consequential one. At three seeds the placement ordering was CLIP ahead of DINOv2; at ten it is DINOv2 in 7 of 10 seeds against CLIP's 5. The ordering reversed with nothing added but seeds. A three-seed comparison the usual unit of evidence in this literature would have supported the opposite conclusion here, which is why the benchmark reports per-seed outcomes by default, and why the ablations in Table II carry no verdict beyond the noise-injection row.

V. LIMITATIONS AND CONCLUSION

The current task instance carries constraints that bound what the reference study can be taken to show.

- *Gripper aperture against brick geometry.* Only one of the brick's three face pairs fits the Panda's jaws (Section II), so spawn orientation cannot be randomised; in-place reorientation would require the rotation channels of `pd_ee_delta_pose` and is future work.

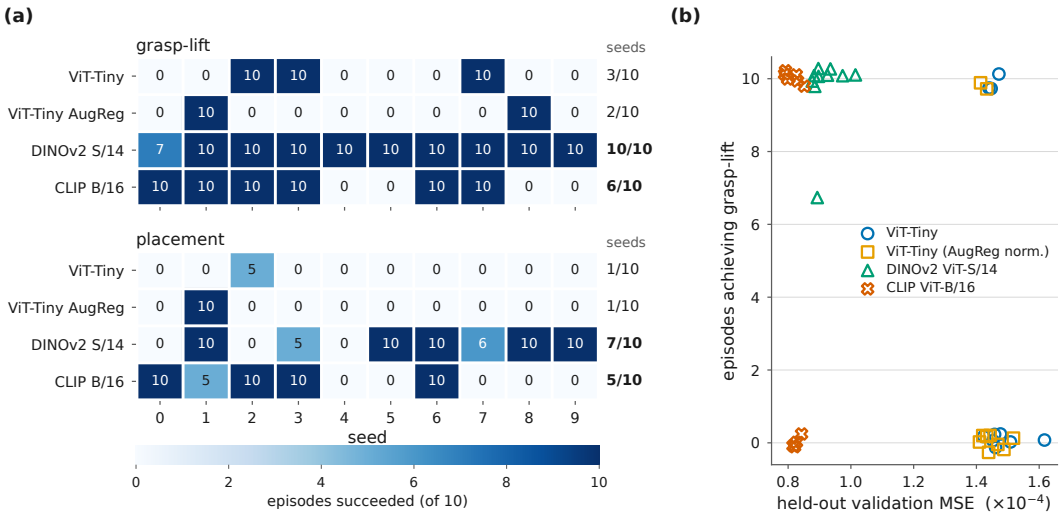

Fig. 2. (a) Per-seed outcomes for the four backbones under the lift-proxy and placement criteria; cells give episodes succeeded of ten, right-margin labels give seeds succeeded of ten. (b) Held-out validation MSE against episodes achieving grasp–lift, one point per (backbone, seed): offline error does not predict closed-loop success.

- *Fixed spawn pose.* The initial state is fixed, so a seed's ten episodes are near-identical and the seed count, not the episode count, is the effective sample size.
- *Placement tolerance.* Object and receptacle dimensions are task-declared constants rather than fixed properties of the benchmark, so placement tolerance is a design axis the suite exposes; the current instance uses a single pairing.
- *Seed asymmetry.* The ablations ran at three seeds against the backbones' ten, so the multi-seed protocol this paper argues for is met by the backbone comparison and not yet by the ablation table; only noise injection is resolvable at that count.
- *Scope.* Simulation only, one task and one object.

The reference policy's failure mode is that perception suffices for grasping and does not suffice for placing, and placement varies across training seeds on a task whose initial state is fixed. This locates the bottleneck in policy architecture and task conditioning rather than in visual features. Whether vision-language-action policies reduce that variance is the natural next question, and the multi-seed protocol is the instrument for answering it. Position randomisation at fixed yaw, receptacle geometries that admit tighter placement tolerances, and further construction primitives are the other planned extensions.

References

[1] 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, art. 103370, 2020.

[2] Y. Li, S. Liu, M. Wang, S. Li, and J. Tan, "Teleoperation-driven and keyframe-based generalizable imitation learning for construction robots," *Journal of Computing in Civil Engineering*, vol. 38, no. 6, 2024.

[3] H. Yu, V. R. Kamat, C. C. Menassa, W. McGee, Y. Guo, and H. Lee, "Mutual physical state-aware object handover in full-contact collaborative human-robot construction work," *Automation in Construction*, vol. 150, art. 104829, 2023.

[4] H. Yu, V. R. Kamat, and C. C. Menassa, "Cloud-based hierarchical imitation learning for scalable transfer of construction skills from human workers to assisting robots," *Journal of Computing in Civil Engineering*, vol. 38, no. 4, 2024.

[5] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "RLBench: the robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[6] B. Liu *et al.*, "LIBERO: Benchmarking knowledge transfer for lifelong robot learning," in *Advances in Neural Information Processing Systems 36 (NeurIPS), Datasets and Benchmarks Track*, 2023.

[7] Z. Hu, H. Yu, V. Chandramouli, and C.-J. Liang, "Sample-efficient robot skill learning for construction tasks: benchmarking hierarchical reinforcement learning and vision-language-action models," arXiv:2512.14031, 2025.

[8] H. Wen, R. Liu, W. Piao, S. Li, and C. Liu, "BrickSim: a physics-based simulator for manipulating interlocking brick assemblies," arXiv:2603.16853, 2026.

[9] S. Tao *et al.*, "Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3," in *Proc. Robotics: Science and Systems (RSS) XXI*, 2025.

[10] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta, "The unsurprising effectiveness of pre-trained vision models for control," in *Proc. 39th Int. Conf. on Machine Learning (ICML)*, PMLR vol. 162, pp. 17359–17371, 2022.

[11] A. Majumdar *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" in *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

[12] A. Mandlekar *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," in *Proc. Conf. on Robot Learning (CoRL)*, PMLR vol. 164, pp. 1678–1690, 2022.

[13] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg, "DART: noise injection for robust imitation learning," in *Proc. 1st Conf. on Robot Learning (CoRL)*, PMLR vol. 78, pp. 143–156, 2017.

[14] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[15] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. on Machine Learning (ICML)*, PMLR vol. 139, pp. 8748–8763, 2021.

[16] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your ViT? Data, augmentation, and regularization in vision transformers," *Transactions on Machine Learning Research*, 2022.

[17] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. Robotics: Science and Systems (RSS) XIX*, 2023.