CVPR JUNE 3-7, 2026 DENVER COLORADO

Cross-Domain Demo-to-Code via Neurosymbolic Counterfactual Reasoning

Video-instructed robotic programming under domain shift

Jooyoung Kim, Wonje Choi, Younguk Song, Honguk Woo*

Department of Computer Science and Engineering, Sungkyunkwan University

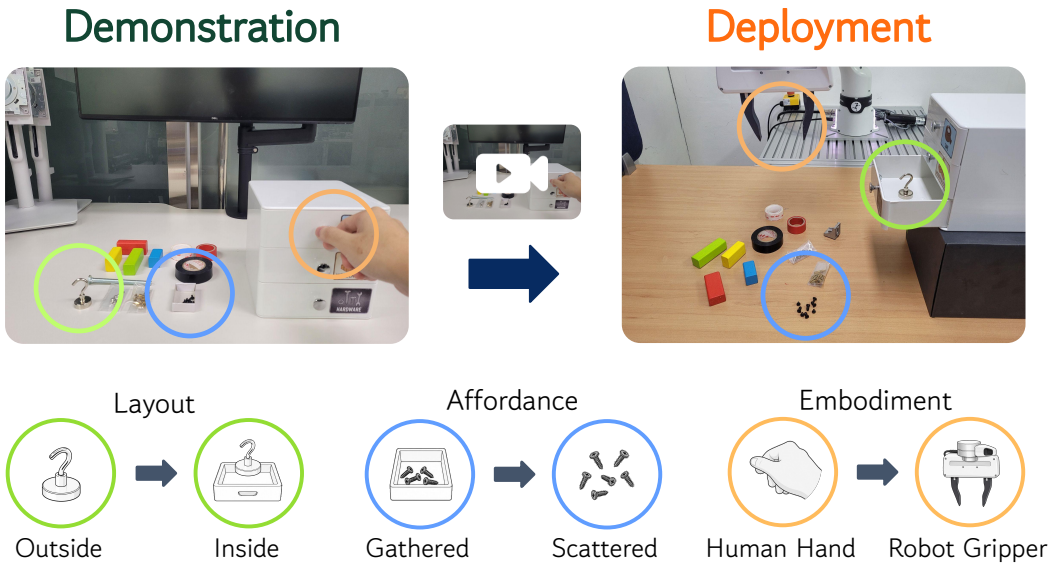
arXiv 2603.18495

1 Problem: Domain Gap in Demo-to-Code

Demo-to-Code
VLMs generate robot policy code from video demonstrations

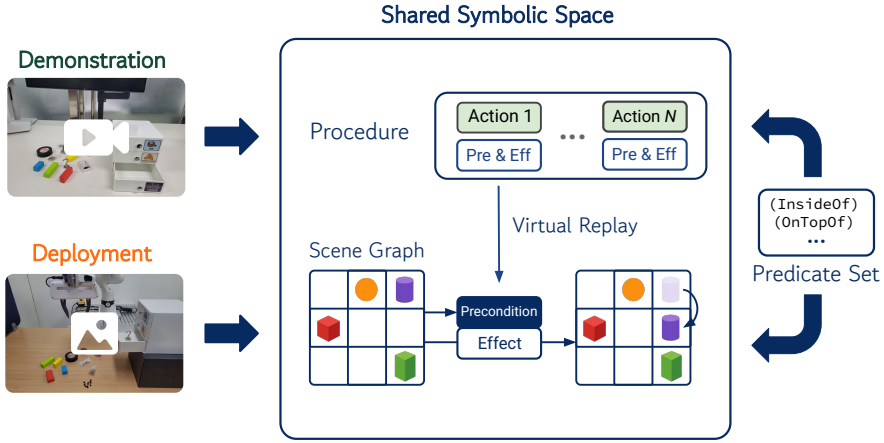
The Challenge
Real deployment often differs from demo in:

- **Layout** (workspace configuration)
- **Affordance** (object states & relations)
- **Embodiment** (human hand → robot gripper)



Difficulty
From raw observations alone, it is *difficult to anticipate* how such physical mismatches will affect the demonstrated procedure, leading to unreliable policy code.

2 Idea: Counterfactual Reasoning in Symbolic Space



Shared Symbolic Space

Abstract both demo and deployment scene into symbolic representations
(objects, predicates, actions with preconditions & effects)

Virtual Replay

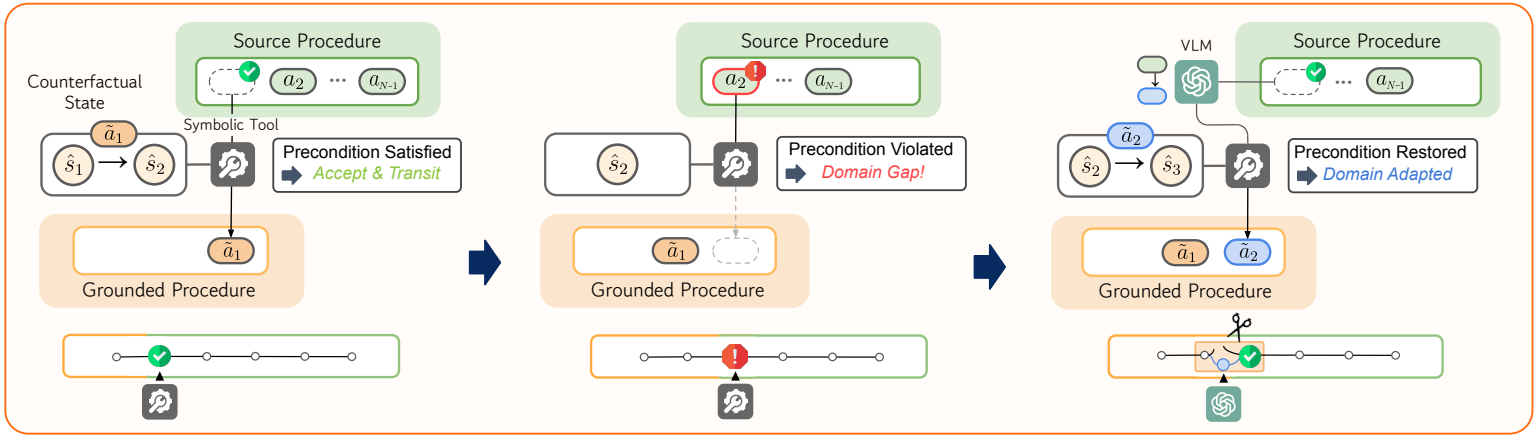
Symbolically replay the demo procedure under deployment conditions to find *where and why it fails*

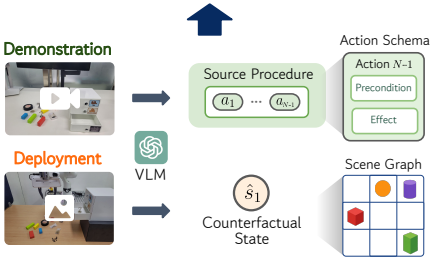
Counterfactual Reasoning

"What would happen if this demo were executed in that environment?"

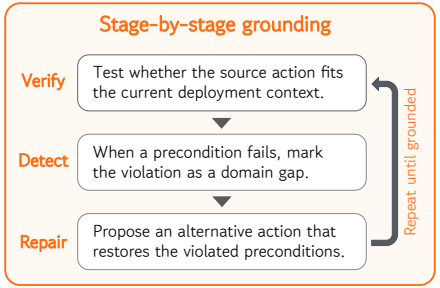
→ **Identify & repair procedural incompatibilities via VLM + symbolic verification loop**

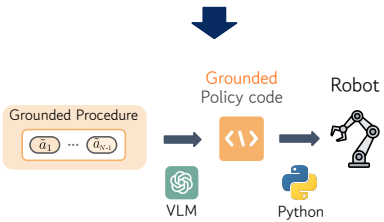
3 NeSyCR: Neurosymbolic Counterfactual Reasoning





Demo video and deployment scene are converted to a symbolic representation.





Grounded procedure is compiled into policy code.

3.1 Symbolic World Model Construction

Symbolic State Translation

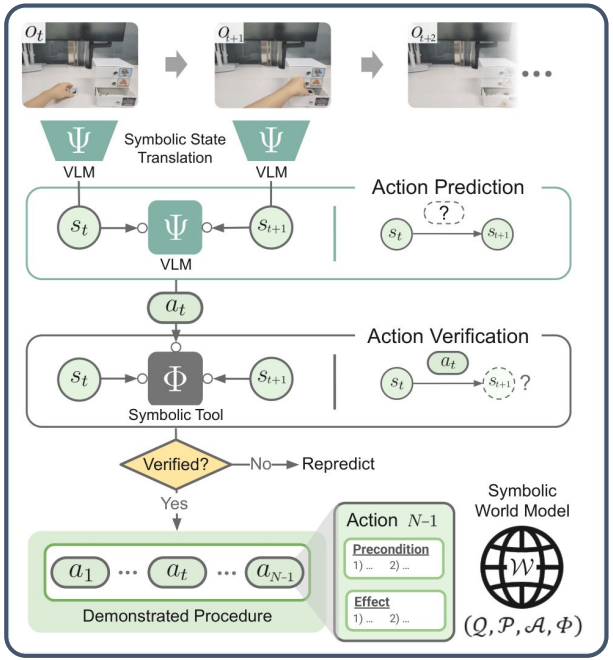
VLM extracts grounded scene graphs from key frames: objects, spatial relations, affordances

Symbolic Dynamics Reconstruction

For each consecutive state pair (s_t, s_{t+1}):

1. VLM predicts action operator a_t
2. Symbolic tool verifies: Φ(s_t, a_t) ⊨ s_{t+1}
3. If fails, VLM repredicts until verified

Result: Verified W = (Q, P, A, Φ)



3.2 Neurosymbolic Counterfactual Adaptation

Counterfactual Identification

VLM generates counterfactual state $\hat{s}_1$ from target observation; symbolic tool forward-simulates the demo procedure under this altered initial state

Counterfactual Exploration

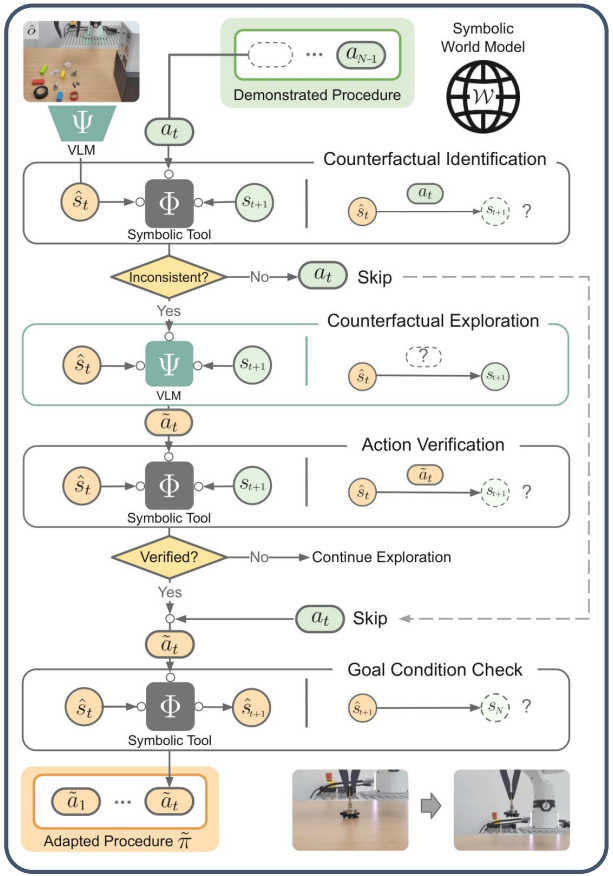
For each inconsistent action at:

1. VLM proposes alternative action ãt
2. Symbolic tool verifies causal validity
3. If invalid, continue exploration

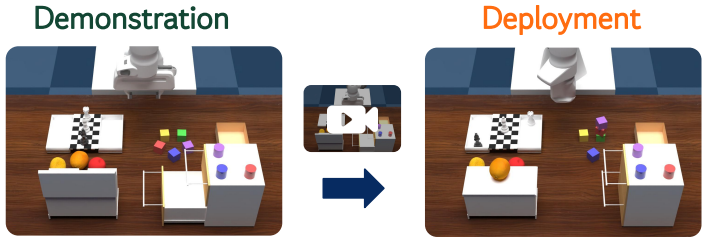
Goal Condition Check

Iterate until adapted procedure reaches goal: ŝ_{t+1} ⊨ sN

Result: Adapted $\tilde{\pi} \rightarrow$ Code policy $\pi\theta = \Psi(\tilde{\pi})$

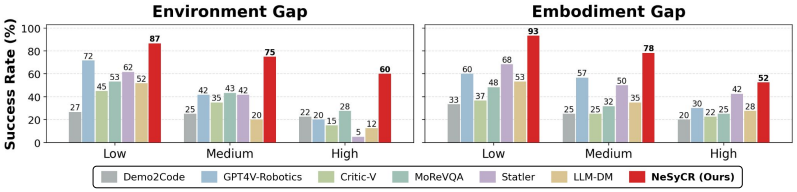


4 Experimental Setup & Result

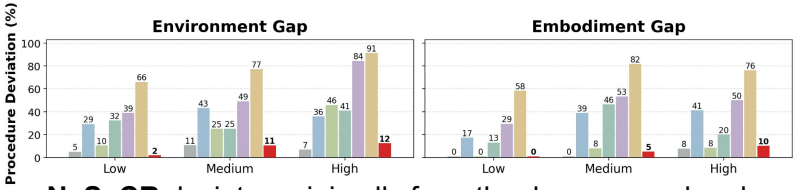


We evaluate on cross-domain scenarios with *environment and embodiment gaps* built in the Genesis simulator.

This include different workspace layouts, object configurations, and robot morphologies.

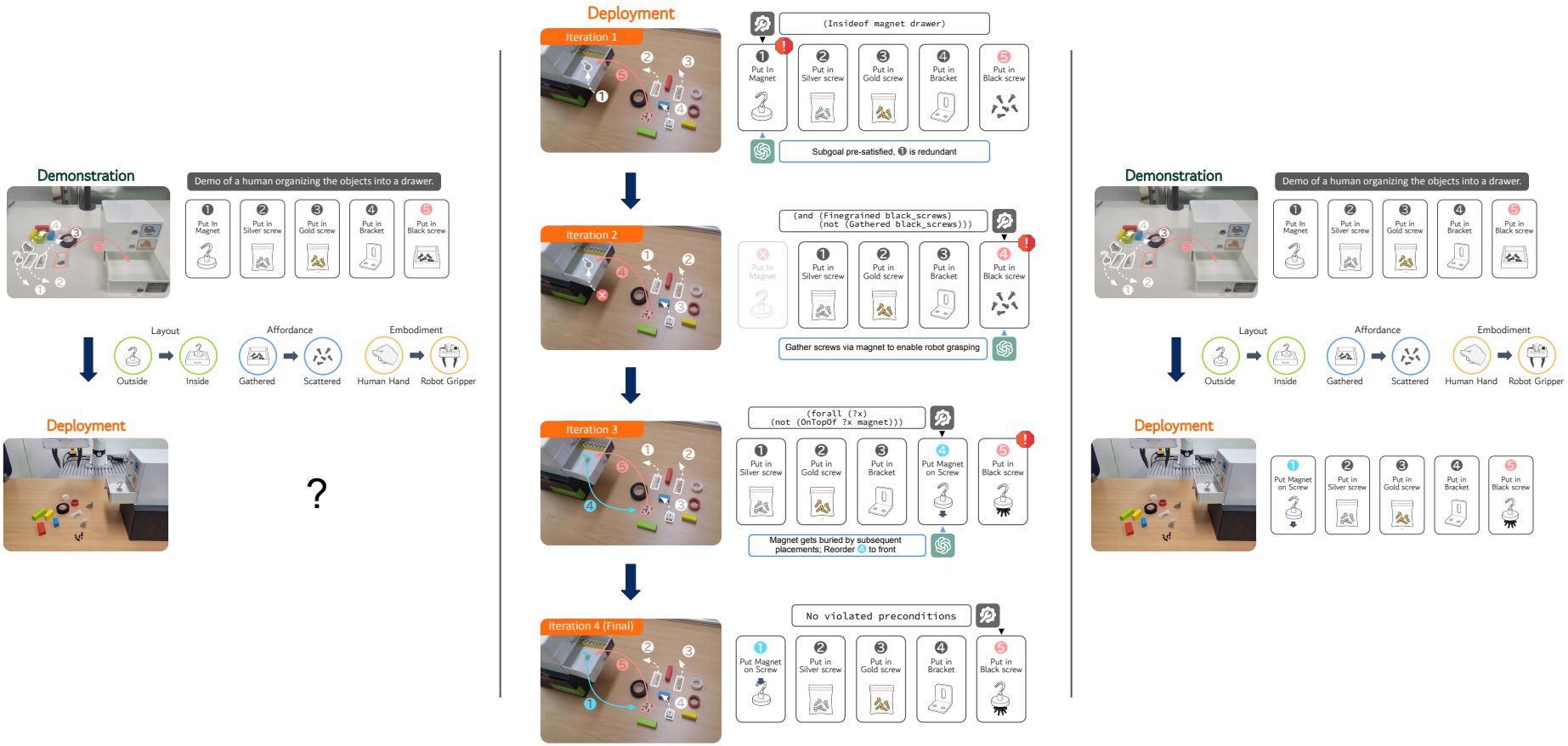


- **NeSyCR** adapts well across every level of *domain gap*



- **NeSyCR** deviates minimally from the demo procedure by precisely editing only the necessary parts

5 Visualization



Any Questions?

{**onsaemiro**, wjchoi1995, syw2045, hwoo}@skku.edu

Department of Computer Science and Engineering, Sungkyunkwan University