Robust and Resilient Soft Robotic Object Insertion with Compliance-Enabled Contact Formation and Failure Recovery

Mimo Shirasaka[1*], Cristian C. Beltran-Hernandez[1], Masashi Hamaya[1], Yoshitaka Ushiku[1]

Abstract— Object insertion tasks are prone to failures under pose uncertainties and environmental variations, traditionally requiring manual finetuning or controller retraining. We present a novel approach for robust and resilient object insertion using a passively compliant soft wrist that enables safe contact absorption through large deformations, without high-frequency control or force sensing. Our method structures insertion as compliance-enabled contact formations, sequential contact states that progressively constrain degrees of freedom, and integrates automated failure recovery strategies.

Our key insight is that wrist compliance permits safe, repeated recovery attempts; hence, we refer to it as compliance-enabled failure recovery. We employ a pre-trained vision-language model (VLM) that assesses each skill execution from terminal poses and images, identifies failure modes, and proposes recovery actions by selecting skills and updating goals. In simulation, our method achieved an 83% success rate, recovering from failures induced by randomized conditions—including grasp misalignments up to $5°$, hole-pose errors up to 20 mm, fivefold increases in friction, and previously unseen square/rectangular pegs—and we further validate the approach on a real robot.

I. INTRODUCTION

Peg-in-hole tasks have been widely studied but remain challenging due to their contact-rich nature and tight tolerances [1]. A central difficulty lies in handling uncertainties in part grasping and hole pose. Conventional methods address these uncertainties through precise pose estimation and rigid fixturing, enabling repetitive pick-and-place operations [2], but such approaches demand substantial engineering effort. This study aims to realize an insertion strategy that explicitly accounts for uncertainty.

Compliance-enabled strategies mitigate these challenges by tolerating positional variation. Passive [3], [4], [5] enables stable contact through inherent mechanical deformation, while active compliance [6] relies on precise force sensing and real-time position control to adaptively adjust contact forces and positions. We build on soft robots, which inherently provide passive compliance and accommodate positional uncertainty through large elastic deformations without high-frequency control or precise force sensing.

While soft robots naturally absorb uncertainty, a key question remains: how can we develop robust and resilient control strategies that leverage their compliance? Here, robustness
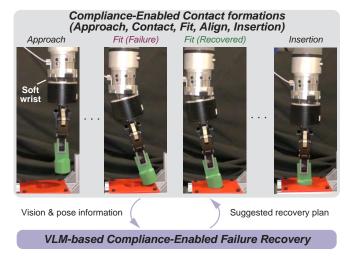
Fig. 1: We propose compliance-enabled contact formations and failure recovery with a soft wrist to achieve more robust and resilient object insertion.

denotes reliable execution under uncertainty, and resilience denotes recovery from failures.

To enhance robustness, we exploit contact formations [7], discrete contact states that constrain the parts' relative motion during assembly. Compliance plays a critical role by enabling stable contact and smooth contact-state transitions, a concept termed *compliance-enabled contact formation* [3]. In peg-in-hole insertion, these formations comprise phases such as approach, contact, fit, alignment, and insertion, each restricting specific degrees of freedom. Following these formations sequentially improves task robustness. Prior work demonstrates the effectiveness of compliance-enabled contact formation [3], [8], [9]; however, failures still occur under severe uncertainties and environmental variations (e.g., friction, part geometry, material properties).

To achieve both robustness and resilience, we integrate a failure recovery strategy into compliance-enabled contact formation. Compliance further facilitates recovery: even if an attempt fails, retries remain safe because compliant interactions limit contact forces, hence we refer to it as *compliance-enabled failure recovery*. A major challenge is accurately identifying failures and selecting appropriate recovery actions. Previous soft-robot recovery methods required training failure detectors [10]. Inspired by recent advances with Large Language Models (LLMs) [11] and Vision-Language Models (VLMs) [12], we employ a VLM to enable compliance-enabled failure recovery without training (Fig. 1). The vision-

language modality analyzes and describes failure cases and proposes recovery plans. Concretely, we predefine goal-conditioned skills for each contact formation and execute them sequentially. A pre-trained VLM (e.g., GPT-4o) evaluates success or failure and explains the outcome; upon failure, it suggests a recovery plan by selecting a sequence of skills and updating goal parameters.

We validate the approach in simulation and on a real robot using a passive compliant soft wrist [13] mounted on a collaborative arm. The method recovers from failures via the VLM in previously unseen settings, including uncertainties in grasping or hole pose and variations in contact friction or peg geometry.

Contribution: We develop a compliance-enabled contact-formation and failure-recovery framework for peg-in-hole insertion, combining a soft wrist with VLM-based reasoning.

II. RELATED WORKS

A. Robust and resilient soft robots

Soft robots conform to object geometry and absorb disturbances, and have therefore been used in applications such as grippers and assistive robots [14], [15], [16]. Prior work on resilience in soft robotics has focused on damage tolerance and self-healing systems [17]. While substantial effort has addressed robustness and resilience at the mechanical aspect, this work targets robust and resilient *control* for peg-in-hole insertion by leveraging soft robots' inherent compliance. The following subsections discuss compliance-enabled contact formation and failure recovery, which together enhance robustness and resilience.

B. Compliance-enabled contact formation

Contact formations enhance robust contact-rich manipulation. Model-based approaches employ velocity control to achieve the target constraint forces [3] for each contact formation. Learning-based techniques update impedance control parameters using Bayesian optimization [8] and reinforcement learning [9]. Additionally, learning from demonstrations has been used to extract skills for contact-rich manipulation tasks [18], [19] Hamaya et al. segmented a peg-in-hole task into several contact formations and employed reinforcement learning [20]. Wang et al. proposed contact configuration identification for hole search using action primitives [21]. Zhang et al. introduced a reinforcement learning method composed of discrete and continuous action spaces to select insertion primitives and low-level control parameters [22]. Saito et al. proposed training action primitives based on the contact state transitions for in-hand manipulation [23]. Kim et al. proposed using tactile sensors to localize the contact line between the peg and hole [24].

Although the methods described above demonstrated robustness in contact-rich manipulations, explicit failure recovery has not been addressed. A recent study by Wu et al. designed a finite-state machine for recovery, which switches skills automatically based on high-frequency force sensor readings and controller [25]. Our study proposes using a soft

wrist and VLM to suggest recovery plans automatically in contact formation-based manipulation.

C. Learning-based error detection and recovery

Numerous researchers have investigated error detection and recovery strategies. Anomaly detection methods benefit robots by alerting them to dangerous or unseen states. Romeres et al. used a Gaussian process-based anomaly detection method when the measured force deviates significantly from the expected range [26]. Park et al. proposed a vision-audio-force-based error detection method composed of an LSTM-based autoencoder [27].

Learning-based approaches enable robots to obtain recovery strategies automatically. Chang et al. employed a Petri net for failure recovery; when the robot fails and visits an unknown state, it asks for human demonstrations to correct the error [28]. Hegemann et al. proposed a method for learning skill graphs based on human labeling and employed multimodal sensors for error detection in grasping and mobile manipulation tasks [29]. Thananjeyan et al. proposed a recovery reinforcement learning method that executed a recovery policy when the agent entered unsafe areas [30]. Vats et al. proposed an efficient learning method that incrementally discovers failure cases and subsequently learns recovery skills [31]. Saxena et al. developed a vision-based failure predictor and corresponding recovery maneuvers for aerial robots [32].

Recent strategies based on LLM or VLM have demonstrated self-recovery in various tasks. Dai et al. proposed combining a VLM with a language-guided visuomotor policy, where the VLM detects errors and suggests recovery strategies using rich language instructions [12]. Cornelio et al. integrated ontologies, logical rules, and LLM-based planners for online error detection and recovery in simulated kitchen environments [33]. Shinha et al. introduced an LLM-based fast-slow reasoning method for aerial robots, which employs fast anomaly detection in the LLM's embedding space and slow autoregressive generation to determine the criticality of anomalies [11]. Liu et al. developed a framework for robot failure explanation and correction in which an LLM verifies the success of each subgoal based on vision and audio summaries [34]. Shirasaka et al. proposed a self-recovery prompting pipeline for promptable robot systems, which refines prompts based on past experiences and active communication with the operator to recover from failures [35]. Wang et al. combined task and motion planning with an LLM to automatically generate recovery symbolic plans [36].

While these methods recover from failures across diverse tasks, they primarily target rigid-robot pick-and-place and mobile navigation. A notable exception in soft robotics proposes a tactile-based predictive network that classifies grasp success or failure [10], but it requires data collection and model training. In contrast, we address peg-in-hole insertion by combining a pretrained VLM-based failure-recovery strategy with compliance-enabled contact formation using a soft wrist.
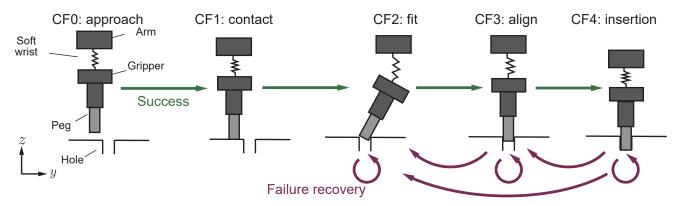
Fig. 2: Compliance-enabled contact formation: We employ the contact formations for peg-in-hole tasks by following the prior studies [8], [20]. It consists of approach, contact, fit, align, and insertion contact formations. We consider applying a failure recovery strategy in fit, align, and insertion contact formations.

III. FRAMEWORK OVERVIEW

We aim to robustly and resiliently complete peg-in-hole tasks, even when faced with uncertainties in grasp or hole pose estimations and environmental variations such as unseen surface friction and peg shapes. This study uses a rigid robot arm and a two-finger gripper with a soft wrist [13]. This soft wrist has three springs and allows large and 6 DoF deformation. An RGB camera is placed to see the peg and the hole.

Following the prior studies [8], [20], we prepare manipulation skills and the goal information of the rigid robot arm's poses p_g and images I_g for the predefined contact formations. The terminal information, which consists of the robot arm pose p_e and the image I_e, is received at the timestep when the skill executions are completed or halted.

Each $n \in N$ skill executes a position control sequentially given p_g^n. A VLM evaluates the success or failure of the n skill based on the goal and terminal information.

If it fails, the VLM suggests a recovery plan that involves reorganizing the skill sequences and updating the goal parameters as necessary. Secs. IV and V describe more detailed information on the contact formations for peg-in-hole tasks and failure recovery strategies, respectively.

IV. MANIPULATION SKILLS IN COMPLIANCE-ENABLED CONTACT FORMATION

This section introduces compliance-enabled contact formation for peg-in-hole tasks. Following the previous studies [8], [20], we assume a 2D-plane peg insertion and a fixed height of the hole surface for simplicity; the arm pose is y, z, and roll θ. As shown in Fig. 2, peg-in-hole tasks have five contact formations (CF): approach, contact, fit, align, and insertion. Only the dimensions of interest are evaluated and updated in each CF.

CF0 (Approach): The robot moves close to the board. The contact does not occur. The goal pose is given as:

$$p_g^{n=0} = [y, z, \theta].$$

CF1 (Contact): The robot goes straight down until it touches the board's surface. Only the z-axis goal pose is

provided or updated.

$$p_g^{n=1} = [-, z, -],$$

where $-$ represents the goal pose of the previous skill.

CF2 (Fit): The robot slides the peg horizontally until the peg's tip touches the hole's edge:

$$p_g^{n=2} = [y, -, -].$$

In CF2, as the soft wrist allows stable contact for the z-axis, we can only consider the y-axis goal pose.

CF3 (Align): The robot rotates the peg to align it vertically:

$$p_g^{n=3} = [y, -, -].$$

Due to its compliance, the soft wrist can automatically rotate the peg with only y-axis translations.

CF4 (Insertion): The robot moves the peg down to a certain depth of the hole. We consider only the z-axis motion:

$$p_g^{n=4} = [-, z, -].$$

We sequentially execute the skills of CF0 to CF4. When the CF4 is completed, the overall task is completed. Due to the fixed height of the hole surface, CF0 and CF1 are always successful. When the robot fails in a CF, the recovery plan suggests the subsequent skill sequences starting from CF2 to CF4.

V. COMPLIANCE-ENABLED FAILURE RECOVERY WITH VLM

This section introduces our method for compliance-enabled failure recovery, which consists of a success check and a failure recovery process. The overview is shown in Fig. 3.

First, a VLM judges success or failure based on the terminal robot arm pose p_e and the image I_e. If the VLM judges that the skill execution of the contact formation fails, the VLM-based failure recovery is executed. The VLM analyzes the reasons for failure, suggests recovery skill sequences, and updates goal parameters as necessary.

1. Success check

VLM

Overall success check

Prompt

- Use position and vision data analysis
- Balance reliability across modalities
- Prioritize safety over majority vote
- Justify decisions with short reasoning

Position check

Vision check

Compression limit: **Failure**

Overall success check: **Failure**

Overall success check: **Success**

Execute next skill

2. Failure recovery

VLM

Phase 1: Failure analysis and recovery suggestion

Prompt

- Analyze WHAT, WHY, and HOW TO RECOVER
- Use visual and positional data
- Consider previous log
- Update constraints : ±0.005 (z), ±0.01 (y)

Phase 2: Generate executable recovery plans

Prompt

- Read the Phase 1 failure analysis thoroughly
- Create a recovery skill sequence
- Update only the necessary parameter for goal pose
- Output the final plan in function call format

Generated recovery plan

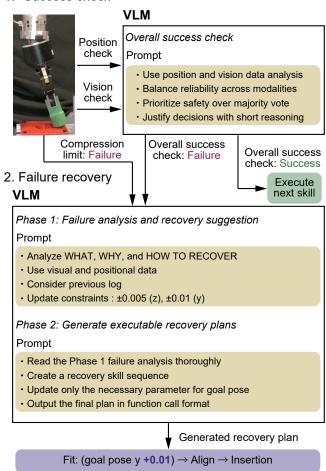Fit: (goal pose y **+0.01**) → Align → Insertion

Fig. 3: Overview of our failure recovery strategy: 1) Success check, where a VLM evaluates the skill execution. 2) Failure recovery planning, where the VLM analyzes the failure and generates recovery plans.

Fit: peg pose estimation
Align: peg pose & angle estimation

Insertion: image comparison

Peg Angle: 7.00 degrees
Peg Center: (308, 130)
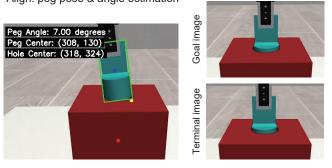Hole Center: (318, 324)

Goal image

Terminal image

Fig. 4: Vision success check: we applied vision-based pose and angle estimation using OpenCV in fit and align, and image comparison that is fed to a VLM in insertion.

A. Success check

Vision and pose-based success check: To verify success in the Fit and Align CFs, we apply vision-based analysis using OpenCV. The peg and hole are segmented using HSV color thresholding: the peg is detected as a green (or cyan in simulation) region, and the hole as a red region. We extract the largest contour for each object to estimate its bounding boxes and center coordinates. For pose estimation, we fit a minimum-area rectangle to the peg contour using `cv2.minAreaRect`, from which the tilt angle of the peg is computed. Success is determined based on visual measurements of the peg's position and orientation relative to the hole. For the Fit subtask, we compare the y-coordinates of the peg's center P_{center} and its bottom-right edge P_{edge} against the hole's center H_{center}. For the Align subtask, we evaluate (1) the horizontal offset between P_{center} and H_{center}, and (2) whether the peg's tilt is within $\pm 5°$ of vertical. The criteria for both subtasks are summarized in Table I.

In the Insertion subtask, the peg becomes visually hidden once it enters the hole, making it difficult for OpenCV to determine whether the operation was successful. Therefore, we provide a goal image and success criteria in the prompt and use a VLM to determine vision-based success. The visual success verification overview is shown in Fig. 4.

We also apply a pose-based success check to determine whether the robot reaches the goal pose. The success condition is determined as: $\|p_g - p_e\| \leq p_{th}$, where p_{th} is a threshold for the pose error, set to 0.001 m.

Compression limit: During skill execution, the compression limit is reached when the soft-wrist springs are compressed along the z-axis beyond a predefined threshold. When this occurs, the skill terminates immediately, and the failure-recovery procedure is activated.

Overall success check by VLM: We use a VLM to perform an overall analysis of the execution results for each skill. In precision tasks, multiple sensors may provide conflicting answers (i.e., differing decisions regarding the success or failure of each subtask) due to their fragility in accurate recognition. We leverage the VLM's strong reasoning ability to integrate multisensory analysis results and make a final decision on subtask success.

B. Failure recovery with VLM

The failure recovery system consists of a two-stage analysis with VLM. In the first phase, a VLM analyzes the failure and suggests a recovery plan, including a recovery motion as a relative translation. Based on this analysis, the second VLM agent generates a recovery plan in an executable format with absolute goal positions reflecting the suggested relative movement in the previous analysis. Below, we provide a detailed explanation of the analysis flow.

Failure analysis and recovery suggestion: To generate a concise recovery plan, an accurate understanding of the situation is essential. Thus, upon detecting a failure, the framework summarizes the situation, focusing on the failure. We establish an analysis-to-suggestion thought pathway by prompting the VLM to consider the situation from three perspectives: 1) What happened?, 2) Why did it occur?, and 3) How can the framework recover from the failure? The recovery suggestion includes motion instructions in terms of

TABLE I: OpenCV Success Criteria

Fit

P_{center}	P_{edge}	Result Message
$\leq H_{center}$	$> H_{center}$	success
$\leq H_{center}$	$\leq H_{center}$	p_{gy} was too small.
otherwise	–	p_{gy} was too big.

Align

Peg relative pose	Tilt Angle	Result Message
On the hole	Within $\pm 5°$	Align success
On the hole	$> 5°$	peg is tilted to the left.
On the hole	$< -5°$	peg is tilted to the right.
Left of the hole	Within $\pm 5°$	p_{ey} was too small.
Right of the hole	Within $\pm 5°$	p_{ey} was too big.
Left of the hole	$> 5°$	peg is tilted to the left. p_{gy} was too small.
Right of the hole	$> 5°$	p_{gy} was too big. also tilted to the left.
Left of the hole	$< -5°$	p_{gy} was too small. also tilted to the right.
Right of the hole	$< -5°$	p_{gy} was too big. also tilted to the right.

direction and values relative to the current goal parameters. By guiding the VLM to reason through the situation, we enable the recovery framework to generate a well-founded strategy, thereby enhancing both the explainability of the recovery process and the transparency of the framework's workflow.

Generate executable recovery plans: In the second phase of failure recovery, the VLM translates the suggested recovery strategy into a sequence of executable skills necessary to complete the task (i.e., successfully inserting the peg into the hole). To achieve this, we utilize the VLM's function-calling capability, prompting it to update the goal position as needed. These updates reflect the recovery adjustments proposed in the previous phase, ensuring a structured and adaptive execution plan.

VI. EXPERIMENTS

We performed experiments in a simulation and with a real robot. The robotic system consisted of a Universal Robot UR5e, a Robotiq Adaptive Gripper 2-Finger 85, and a soft wrist [13]. A built-in force/torque sensor of UR5e was used only to check the compression limit of the wrist's springs, and was not used for control. The soft wrist comprises three springs that allow six degrees of freedom deformations. As for VLM, we used GPT-4o (version 1.64.0).

The goal of the experiments is to answer the following questions: (1) Can our method improve the success rates by recovery?, (2) Can our method recover from failures in uncertain and unseen environments?, and (3) Can our method be used in real-world environments?

A. Simulation experiments

To answer the first and second questions, we performed simulation experiments.

Simulator setup: We used a simulator developed by a previous study [37]. The simulation is based on the Robo-

suite framework with the MuJoCo physics engine [38]. The previous study developed a simplified soft wrist composed of four degrees of freedom: z, roll, pitch, and yaw. The stiffness parameters were the same as those of the previous study [37]. We assume that the peg was fixed to the gripper and no slippage occurred during the tasks. The tolerance between the peg and hole was 2 mm. We used a red and green colored peg and hole and applied OpenCV for pose estimation from vision, introduced in Sec. V-A.

Default condition and initial goal parameters: We designed the initial goal pose p_g in the default condition where the circular peg was aligned in the gripper, and the hole was fixed in a specific position. We controlled the robot with a keyboard and recorded p_g^n in n contact formations.

Randomized conditions: To see the robustness of the failure recovery, we test our method in uncertain environments. We uniformly randomized the in-grasp angle of the peg within $\pm 5°$, the uncertainty of the position of the hole within ± 20 mm from the default poses. The hole's surface friction was randomized up to a 5-times increase compared to the default condition. We also tested our method with unseen peg shapes, including squares and rectangles.

Metrics and baselines: The robot sequentially executed skills given the default goal pose p_g and our recovery method in the four conditions, including (1) no randomization, (2) peg and hole pose randomization, (3) friction randomization, and (4) all randomization (peg and hole pose and friction). As a baseline, we executed the skills with the default goal pose and without our recovery method.

The trials were performed 30 times, and the success rates of the overall task completions were evaluated. Overall success was defined as when the peg was inserted into the hole to a depth of 10 mm. Overall failures were considered when a timeout (i.e., recovery skill executions failed consecutively 10 times) occurred.

Results: The success rates for each condition are shown in Table II. The values in each contact formation (Fit, Align, and Insertion) denote the average number of additional executions triggered by the failure recovery. "F" denotes framework success (i.e., the VLM's judgment of task completion), while "G" denotes ground-truth success/failure. The baseline without recovery achieves 47% success under the all randomization setting. In contrast, our recovery-enabled method improves the success rate to 83%, underscoring the significant contribution of recovery. In addition, the results show that the proposed framework generalizes robustly to previously unseen peg shapes (square and rectangular); notably, it achieves 80% success under the all randomization setting with the square peg.

The recovery transitions of the circular peg are presented in Table III. Although the majority of the transitions are the same contact formations, it should be noted that among the recovery strategies called upon failures in Align execution, 9.5% of them were Fit.
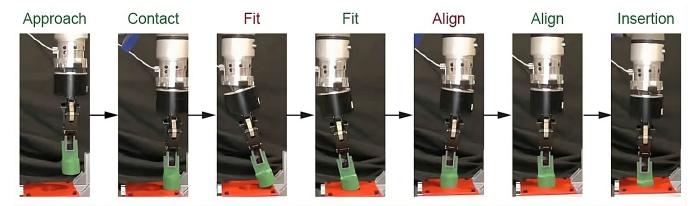
Fig. 5: Snapshots of a peg-in-hole trial: The robot failed in the Fit and Align contact formations, but it recovered using our method. The contact formations colored with green and purple above each snapshot describe the successful and failed executions, respectively.

TABLE II: Performance under different conditions for each peg shape (Sim)

Peg Shape	Randomized Conditions	Fit	Align	Insert	Succ. [F, G]
		Avg. # extra executions			
◯ Baseline (w/o recovery)	None	—	—	—	100, 100
	Friction	—	—	—	43, 43
	Hole Pose & Peg Angle	—	—	—	57, 57
	All	—	—	—	47, 47
◯	None	0.0	0.0	0.0	100, 100
	Friction	0.6	0.1	0.0	100, 100
	Hole Pose & Peg Angle	0.7	1.1	0.7	93, 90
	All	0.8	1.0	1.1	87, 83
▢ (Unseen)	None.	0.0	0.0	0.0	100, 100
	Friction	0.0	0.3	0.0	100, 100
	Hole Pose & Peg Angle	0.4	1.3	0.5	93, 93
	All	0.8	2.1	1.1	80, 80
▭ (Unseen)	None	0.0	0.0	0.0	100, 100
	Friction	0.0	0.0	0.0	100, 100
	Hole Pose & Peg Angle	0.7	2.1	0.8	77, 77
	All	0.6	2.5	1.5	67, 67

TABLE III: Transition Matrix — Circular Peg (Sim)

From\To	Fit	Align	Insert
Fit	100.0%	0.0%	0.0%
Align	9.5%	90.5%	0.0%
Insert	0.0%	1.8%	98.2%

B. Real robot experiments

To answer the third question, we evaluated the proposed recovery framework on a real robot using a round-shaped peg. We used an Intel RealSense D435f camera. Experiments were conducted under seven combinations of conditions, varying peg tilt angles (0° and 5°) and hole positions (0, 15, and 20 mm). The success rates in five trials for each condition are shown in Table IV. The results confirmed that,

TABLE IV: Performance under different combinations of hole position and peg angle in real robot experiments

Hole	Angle	Fit	Align	Insert	Success [F, G]
		Avg. # extra executions			
0	0	0.0	0.2	0.6	100, 100
10	0	0.2	0.4	0.2	100, 100
15	0	1.4	1.4	0.4	100, 100
20	0	2.2	2.8	0.0	80, 80
10	5	2.0	0.8	0.2	100, 100
15	5	2.0	0.4	0.2	100, 100
20	5	3.0	1.8	1.4	80, 100

with recovery, a success rate of over 80% was achieved in all conditions. Similar to the simulation, the number of additional executions per subtask increased, indicating that the condition change necessitated error recovery for task completion. Fig. 5 shows a peg-in-hole trial with our recovery.

VII. DISCUSSION

A. Discussion about experimental results

All randomization setting with a circular peg:
In the all randomization condition (hole pose, peg angle, and friction) with a circular peg, the success rate was 83%. Log analysis suggests that the main factor limiting further improvement was activation of the wrist's compression limit during repeated Insertion attempts. This occurred when small misalignments of the peg persisted, and the Align success check, affected by vision estimation error, classified the alignment as successful. After one or more retries, the system reached the predefined maximum number of trials (set to 10 in this study).

Accuracy of success classification: The framework's classifications (F) were generally consistent with the ground truth (G), reflecting the effect of the verification mechanisms. This indicates that the method provides reliable success judgments across diverse and randomized conditions. Occasional discrepancies were observed; for instance, the framework classified a task as successful when the peg was not fully inserted.

Robustness against friction variation: Although a previous study [3] reported that compliance-enabled contact

formation has robustness to variations in hole pose and peg angle, it struggled with higher friction conditions. In contrast, the proposed recovery framework achieved a 100% success rate under the increased friction condition.

Alignment failures with unseen peg shapes: In the case of unseen peg shapes, the success rate declined (by 7% for the square peg and 20% for the rectangular peg, as shown in Table II). The primary factor for this decline was the increased likelihood of the robot reaching the maximum number of predefined recovery attempts in the Align subtask.

For example, under the *Hole Pose & Peg Angle* condition, the average number of executions for Align increased by 0.2 times for the square peg and 0.8 times for the rectangular peg compared to the circular peg. Log analysis revealed that failures predominantly occurred when the randomized angle exceeded $4.0°$. In many of these failures, as the system attempted to recover from misalignment, the peg frequently overshot the hole, attempted to correct, and overshot again. The challenge was to fit the peg into the hole such that, by leveraging the compliance of the wrist, the peg could be aligned to an angle close to $0°$. When the initial angle was large, this became increasingly difficult, as the vertical height of the peg shortened with tilt, making proper fitting more challenging.

Real robot experiments: Real robot experiments achieved a high success rate, with one failure observed when repeated misalignments in the Align subtask led the system to reach the maximum number of retries. Analysis revealed that this was caused by an excessively large parameter update suggested by the VLM. This suggests that such failures could be mitigated by narrowing the allowable range for parameter adjustments at each step.

B. Limitations and future work

Our method demonstrated successful recoveries in randomized and unseen conditions; however, some limitations remain.

This study used pre-designed skill sets. They can be replaced with more intelligent learning-based policies that may reduce the number of recovery retries. Nevertheless, even with more intelligent policies, some failures are likely to occur. Thus, our method will remain necessary across different skill sets. For future work, we will consider integrating learning-based skills described above into our compliance-enabled contact formation and failure recovery framework.

VIII. CONCLUSIONS

This study proposes integrating the compliance-enabled failure recovery using VLMs into compliance-enabled contact formation using a soft wrist for robust and resilient peg-in-hole insertion. The simulation and real robot experiments demonstrated that our method effectively recovered from failures and improved success rates under uncertain and unseen conditions.

REFERENCES

[1] J. Xu, Z. Hou, Z. Liu, and H. Qiao, "Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies," *arXiv preprint arXiv:1904.05240*, 2019.

[2] K. Ota, D. K. Jha, S. Jain, B. Yerazunis, R. Corcodel, Y. Shukla, A. Bronars, and D. Romeres, "Autonomous robotic assembly: From part singulation to precise assembly," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 13 525–13 532.

[3] A. S. Morgan, Q. Bateux, M. Hao, and A. M. Dollar, "Towards generalized robot assembly through compliance-enabled contact formations," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 8010–8016.

[4] T. Nishimura, Y. Suzuki, T. Tsuji, and T. Watanabe, "Peg-in-hole under state uncertainties via a passive wrist joint with push-activate-rotation function," in *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 67–74.

[5] Q. Zhang, Z. Hu, W. Wan, and K. Harada, "Compliant peg-in-hole assembly using a very soft wrist," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 17–24, 2023.

[6] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, vol. 10, no. 19, p. 6923, 2020.

[7] M. Skubic and R. A. Volz, "Identifying contact formations from sensory patterns and its applicability to robot programming by demonstration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 1996, pp. 458–464.

[8] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5844–5850.

[9] N. Vuong, H. Pham, and Q.-C. Pham, "Learning sequences of manipulation primitives for robotic assembly," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 4086–4092.

[10] K. Gilday, T. George-Thuruthel, and F. Iida, "Predictive learning of error recovery with a sensorized passivity-based soft anthropomorphic hand," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200390, 2023.

[11] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone, "Real-time anomaly detection and reactive planning with large language models," in *Robotics: Science and Systems*, 2024.

[12] Y. Dai, J. Lee, N. Fazeli, and J. Chai, "Racer: Rich language-guided failure recovery policies for imitation learning," *arXiv preprint arXiv:2409.14674*, 2024.

[13] F. von Drigalski, K. Tanaka, M. Hamaya, R. Lee, C. Nakashima, Y. Shibata, and Y. Ijiri, "A compact, cable-driven, activatable soft wrist with six degrees of freedom for assembly tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 8752–8757.

[14] G. Agarwal, N. Besuchet, B. Audergon, and J. Paik, "Stretchable materials for robust soft actuators towards assistive wearable devices," *Scientific Reports*, vol. 6, no. 1, p. 34224, 2016.

[15] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Robust proprioceptive grasping with a soft robot hand," *Autonomous Robots*, vol. 43, no. 3, pp. 681–696, 2019.

[16] M. Li, A. Pal, A. Aghakhani, A. Pena-Francesch, and M. Sitti, "Soft actuators for real-world applications," *Nature Reviews Materials*, vol. 7, no. 3, pp. 235–249, 2022.

[17] R. A. Bilodeau and R. K. Kramer, "Self-healing and damage resilience for soft robotics: A review," *Frontiers in Robotics and AI*, vol. 4, p. 48, 2017.

[18] A. T. Le, M. Guo, N. van Duijkeren, L. Rozo, R. Krug, A. G. Kupcsik, and M. Bürger, "Learning forceful manipulation skills from multi-modal human demonstrations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 7770–7777.

[19] M. Okada, M. Komatsu, R. Okumura, and T. Taniguchi, "Learning compliant stiffness by impedance control-aware task segmentation and multi-objective bayesian optimization with priors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 8155–8162.

[20] M. Hamaya, R. Lee, K. Tanaka, F. Von Drigalski, C. Nakashima, Y. Shibata, and Y. Ijiri, "Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 7747–7753.

[21] C. Wang, H. Luo, K. Zhang, H. Chen, J. Pan, and W. Zhang, "Pomdp-guided active force-based search for robotic insertion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 10 668–10 675.

[22] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *International Conference on Robotics and Automation*, 2022, pp. 9881–9887.

[23] D. Saito, A. Kanehira, K. Sasabuchi, N. Wake, J. Takamatsu, H. Koike, and K. Ikeuchi, "Apricot: Action primitives based on contact-state transition for in-hand tool manipulation," in *IEEE-RAS International Conference on Humanoid Robots*, 2024, pp. 827–834.

[24] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *International Conference on Robotics and Automation*, 2022, pp. 10 241–10 247.

[25] Y. Wu, F. Wu, L. Chen, K. Chen, S. Schneider, L. Johannsmeier, Z. Bing, F. J. Abu-Dakka, A. Knoll, and S. Haddadin, "1 khz behavior tree for self-adaptable tactile insertion," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 16 002–16 008.

[26] D. Romeres, D. K. Jha, W. Yerazunis, D. Nikovski, and H. A. Dau, "Anomaly detection for insertion tasks in robotic assembly using gaussian process models," in *European Control Conference*, 2019, pp. 1017–1022.

[27] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[28] G. Chang and D. Kulić, "Robot task error recovery using petri nets learned from demonstration," in *International Conference on Advanced Robotics*, 2013, pp. 1–6.

[29] P. Hegemann, T. Zechmeister, M. Grotz, K. Hitzler, and T. Asfour, "Learning symbolic failure detection for grasping and mobile manipulation tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 4302–4309.

[30] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery RL: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[31] S. Vats, M. Likhachev, and O. Kroemer, "Efficient recovery learning using model predictive meta-reasoning," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 7258–7264.

[32] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5824–5829.

[33] C. Cornelio and M. Diab, "Recover: A neuro-symbolic framework for failure detection and recovery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 12 435–12 442.

[34] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," in *Conference on Robot Learning*, 2023, pp. 3468–3484.

[35] M. Shirasaka, T. Matsushima, S. Tsunashima, Y. Ikeda, A. Horo, S. Ikoma, C. Tsuji, H. Wada, T. Omija, D. Komukai *et al.*, "Self-recovery prompting: Promptable general purpose service robot system with foundation models and self-recovery," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 17 395–17 402.

[36] S. Wang, M. Han, Z. Jiao, Z. Zhang, Y. N. Wu, S.-C. Zhu, and H. Liu, "LLM3: Large language model-based task and motion planning with motion failure reasoning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 12 086–12 092.

[37] Y. Fuchioka, C. C. Beltran-Hernandez, H. Nguyen, and M. Hamaya, "Robotic object insertion with a soft wrist through sim-to-real privileged training," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 9159–9166.

[38] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*.