

AtomBridge: Agentic VLA Inference Plugin for Long-Horizon Tasks in Scientific Experiments

Yiwen Pang^{1,2}, Bo Zhou^{1,2}, Changjin Li^{1,2}, Xuanhao Wang^{1,2}, Shengxiang Xu^{1,2}, Deng-Bao Wang^{1,2}, Peng Cheng³, Shimin Di^{1,2}, Jingkuan Song³, Min-Ling Zhang^{1,2}

¹School of Computer Science and Engineering & School of Software Engineering
& School of Artificial Intelligence, Southeast University, Nanjing, China

²Pattern Learning and Mining Lab, Southeast University, Nanjing, China

³Tongji University, Shanghai, China
shimin.di@seu.edu.cn

Abstract

Robotic laboratories play a critical role in autonomous scientific discovery by enabling scalable, continuous experimental execution. Recent vision–language–action (VLA) models offer a promising foundation for robotic laboratories. However, scientific experiments typically involve long-horizon tasks composed of multiple atomic tasks. Existing VLA models may fail to perform composed tasks formed by reordering and composing these known atomic actions. This limitation can arise from a skill-chaining gap caused by robot-state mismatch: the terminal robot state of one skill can fall outside the valid initial-state distribution of the next. To address this challenge, we propose AtomBridge, an Agentic VLA Inference Plugin for Long-Horizon Tasks in Scientific Experiments. AtomBridge attaches at inference time to a VLA policy already fine-tuned on atomic tasks, while keeping its weights fixed. At each task boundary, it uses LLM-based transition reasoning and robotic-action code generation to insert transitional actions between consecutive tasks. This plug-and-play design mitigates the skill-chaining gap caused by robot-state mismatch without additional VLA fine-tuning or demonstrations of composed long-horizon sequences. Across scientific manipulation sequences in simulation and a real-world experimental environment, AtomBridge improves execution continuity and per-step atomic-task success. On 8-step composed tasks, AtomBridge improves full-sequence success by 10%~25%.

1 Introduction

In recent years, the AI4Science paradigm (Wang et al. 2023) has evolved from single-model approaches (Jumper et al. 2021; Bi et al. 2023) toward autonomous scientific discovery (Novikov et al. 2025; Volk et al. 2023; Dai et al. 2025; Koscher et al. 2023). Emerging research aims to directly embed artificial intelligence into scientific workflows to systematically enhance the entire research lifecycle (Koscher et al. 2023; Szymanski et al. 2023; Boiko et al. 2023). A robotic laboratory plays a critical role in enabling autonomous discovery (Burger et al. 2020; Szymanski et al. 2023; Coley et al. 2019). It creates experimental environments with high-precision control, strong reproducibility, and large-scale execution capabilities, substantially accelerating scientific progress while reducing labor and time costs. Moreover, robots capable of continuous, unattended operation significantly reduce researchers’ exposure to hazardous

experimental conditions. Autonomous laboratories have successfully discovered correct synthesis routes for 41 theoretically predicted materials through large-scale automated material synthesis experiments (Szymanski et al. 2023).

Current robotic laboratories typically follow the automation paradigm based on manually programmed, fixed experimental procedures (Coley, Eyke, and Jensen 2020). This choice is largely motivated by the high-precision requirements of scientific experiments. For instance, in high-throughput chemical screening (Macarron et al. 2011), a single experimental cycle typically consists of sample preparation and reaction execution, for which fixed programs for robotic arms provide stability guarantees. However, such rigid workflows limit the robot’s ability to reconfigure task sequences or respond to unforeseen situations, making them ineffective in open-ended experimental settings (Ochiai et al. 2025). As an example, the parameters of a thermal cycler—such as cycle numbers, temperatures, and durations—must be flexibly adjusted according to experimental objectives, templates, and primers (the scientific instrument in Fig. 1), and thus cannot be pre-encoded as a single robot trajectory.

Recent vision–language–action (VLA) models (Black et al. 2024; Kim et al. 2025) provide a flexible interface for reconfigurable laboratory manipulation. Unlike fixed-program systems, this paradigm allows robots to interpret natural-language instructions and switch among learned manipulation skills. As the set of atomic experimental procedures is generally fixed and finite, VLA models can be trained or fine-tuned using data collected only for *atomic tasks*. Then, to support diverse experimental protocols, these atomic tasks can be dynamically composed and reordered. However, we observe a phenomenon with the VLA model when performing these long-horizon tasks composed of atomic tasks: *even though the VLA model fine-tuned on atomic tasks can handle these atomic tasks smoothly, it still cannot correctly complete composed tasks made up of these known atomic tasks*. As illustrated in Fig. 1 (a), a fine-tuned VLA model can independently execute open thermal cycler and place PCR plate, but fails to correctly execute the composed task open thermal cycler-place PCR plate. This limitation can reflect a *skill-chaining gap* caused by robot-state mismatch (Konidaris and Barto 2009;

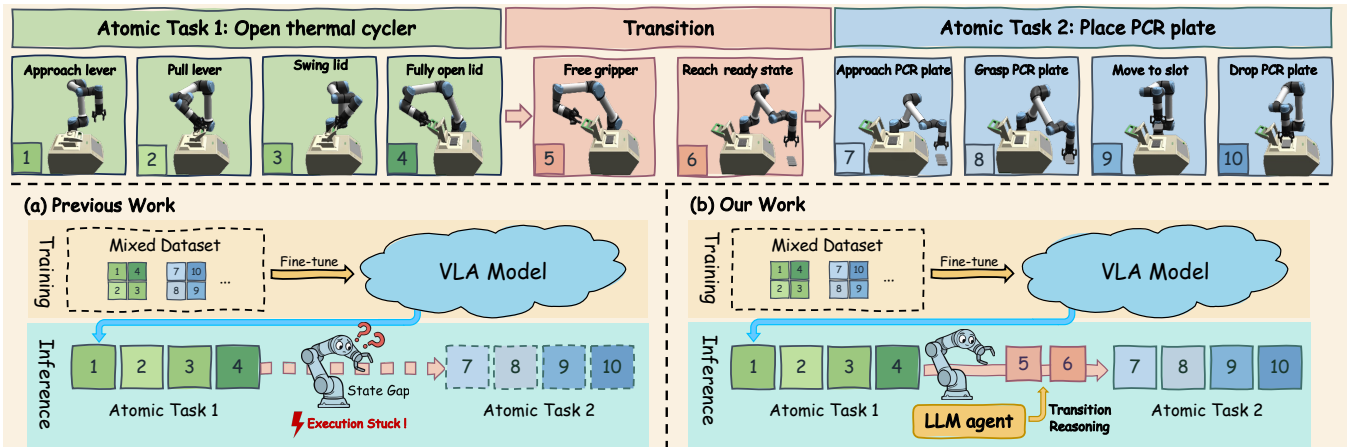


Figure 1: Given a VLA model fine-tuned for atomic tasks in scientific experiments, it may fail to complete the composed task because of a robot-state mismatch between consecutive atomic tasks. In this paper, we propose an LLM-based agentic inference plugin to generate transitional robotic execution code, improving execution of the long-horizon tasks.

Lee et al. 2022; Jung et al. 2025): although the VLA model has learned each atomic task, the terminal robot state of the pre-task can fall outside the valid initial-state distribution of the post-task, preventing the required inter-skill transition.

To address this challenge, we propose AtomBridge, an agentic, plug-and-play inference plugin for long-horizon scientific manipulation. This bridge primarily inserts transitional actions between consecutive atomic tasks within a long-horizon task sequence, thereby ensuring smoother task transitions. Our contributions are as follows:

- We identify the skill-chaining gap caused by the absence of explicit transitions between atomic tasks, and introduce the *ready state*: a pre-manipulation state that lies within the valid initial-state distribution of the next atomic task.
- We introduce AtomBridge, an agentic and plug-and-play inference plugin that detects task completion, retrieves the next task’s ready state, and generates constrained transition actions. It attaches to a fixed atomic-task VLA policy without collecting additional data or re-training the VLA.
- In our simulation test, AtomBridge improves 53 of 60 post-initial atomic-task comparisons and obtains full-sequence success in 11 of 12 task-model configurations, whereas all matched baselines obtain zero. In our real thermal-cycler experiment, it increases the five-step full-sequence success rate from 0% to 40%.

2 Related Work

Robotic Labs. Robotic Labs combine manipulators, mobile platforms, and automated instruments to reduce manual experimental labor (Boyd 2002). Representative systems automate thin-film research and high-throughput perovskite preparation (MacLeod et al. 2020; Zhao et al. 2021), use language models and active sensing to plan solid-state synthesis (Szymanski et al. 2023), and explore photocatalytic parameter spaces with mobile robots (Burger et al. 2020). Earlier systems such as Adam further coupled automated

experiments with hypothesis testing to identify gene functions (King et al. 2009).

Vision-Language-Action Models. VLA models map visual observations and language instructions to robot actions, using vision-language representations (Zhang et al. 2024; Radford et al. 2021) to support generalization across instructions and scenes. Diffusion Policy (Chi et al. 2025) and RDT (Liu et al. 2025) generate continuous action trajectories through iterative denoising, while π_0 (Black et al. 2024) and $\pi_{0.5}$ (Black et al. 2025) use flow matching with a PaliGemma backbone (Beyer et al. 2024). In particular, LabVLA (Ren et al. 2026) is a VLA model trained specifically for scientific laboratory scenarios. After large-scale robot pretraining and task-specific fine-tuning, these models can execute diverse manipulation skills, but composing independently trained skills remains difficult.

Long-Horizon Challenges. Long-horizon manipulation faces multiple challenges. First, small perception or control errors can accumulate across many time steps, so later actions begin from increasingly inaccurate states (Hou et al. 2024). Closed-loop feedback (Ebert et al. 2018), replanning (Yokoyama et al. 2024), and recovery (Sung, Wang, and Stone 2023) are used to correct these deviations during execution. Second, a high-level goal must be decomposed into executable atomic tasks. Hierarchical policies make this decomposition explicit (Dietterich 2000), while recent VLAs learn task decompositions, subtask tokens, or phase-aware representations: $\pi_{0.5}$ adds task-decomposition pairs during vision-language pretraining (Black et al. 2025). LohoVLA predicts atomic-task tokens before action tokens (Yang et al. 2025). LongVLA attends to different visual information across movement phases (Fan et al. 2025). Third, the resulting atomic policies must be connected reliably. This *skill-chaining gap* arises when one policy’s terminal-state distribution does not overlap with the next policy’s valid initial-state distribution, producing out-of-distribution or physically invalid starting states. Existing solutions learn backward-reachable options (Konidaris and Barto 2009), relay or goal-

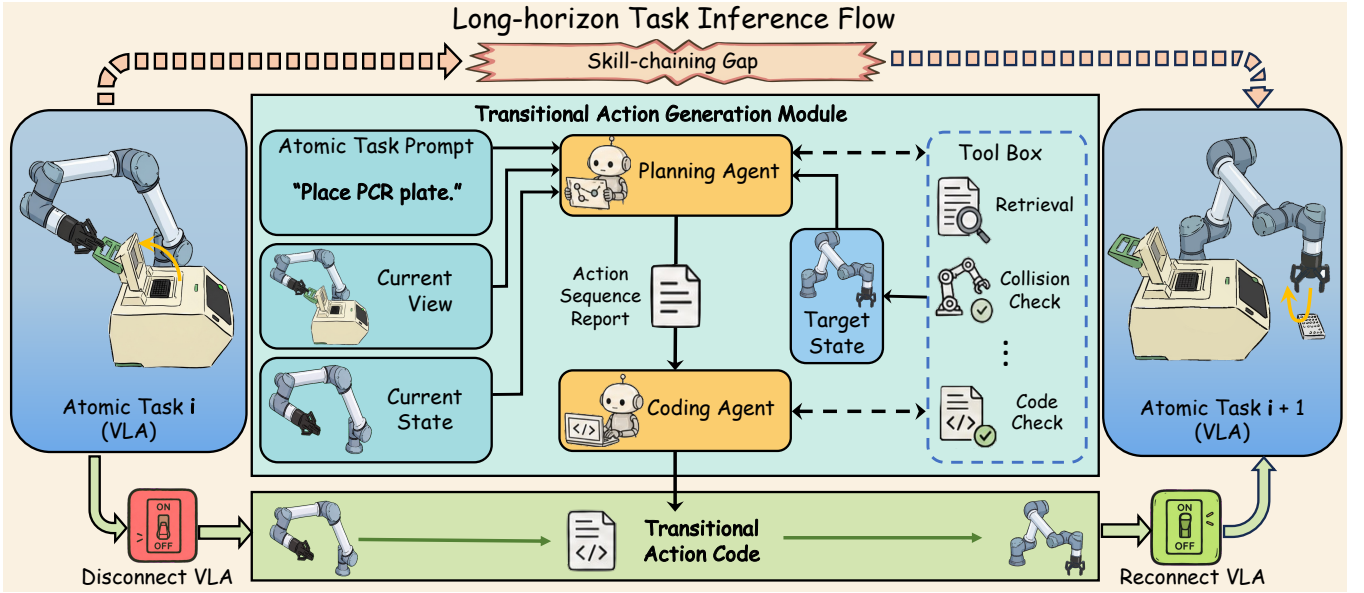


Figure 2: Overview of AtomBridge. Fixed VLA policies execute the atomic tasks in a long-horizon sequence. At each task boundary, the intervention switch detects completion or a task-specific timeout. AtomBridge then retrieves a pre-contact ready state for the next task from its atomic-task memory, and agents will plan and generate transitional action code. After executing the transition, the next atomic VLA policy resumes. This completion–transition–resume cycle repeats without modifying the VLA policy.

conditioned policies (Gupta et al. 2020), or state-distribution alignment (Clegg et al. 2018; Lee et al. 2022). Alternatively, they progressively fine-tune sub-policies (Chen et al. 2023) or plan over learned skill priors and connector policies at inference time (Mishra et al. 2023; Jung et al. 2025). These approaches address decomposition or state compatibility, but do not explicitly generate task-conditioned transitions between fixed VLA policies.

3 AtomBridge: Agentic Plugin

Let $\mathcal{A} = \{\tau^j, \dots, \tau^k\}$ be the set of atomic task descriptions, and let π be a fixed VLA policy trained only on demonstrations of tasks in $\mathcal{A}$. A long-horizon task of length N is an arbitrary ordered sequence $\mathcal{T} = [\tau_1, \dots, \tau_N]$ with $\tau_i \in \mathcal{A}$. For a demonstration trajectory of task τ , let $\mathcal{D}_\tau = \{(I_t, s_t)\}_{t=0}^{L-1}$, where I_t is the RGB image frame at time step t , s_t is the corresponding robot state, and L is the trajectory length. Let r be the last frame before the robot begins the task-specific operation. Its *ready-state set* is

$$\mathcal{R}_\mathcal{D}(\tau) = \{s_t \mid 0 \leq t \leq r\}. \quad (1)$$

Thus, a ready state is any state in which the task has not yet begun and the robot has not started to manipulate the target. Ready state is inherently non-unique and visually gradual. Pooling these sets across demonstrations induces the empirical ready-state distribution $p_\mathcal{D}^{\text{ready}}(s \mid \tau)$. When π executes τ_i , its terminal state s_{cur} is immediately passed to the next policy invocation:

$$s_{\text{cur}} \sim p_\pi^{\text{end}}(s \mid \tau_i) \longrightarrow \pi(\tau_{i+1}; s_{\text{cur}}), \quad (2)$$

$$p_\pi^{\text{end}}(s \mid \tau_i) \neq p_\mathcal{D}^{\text{ready}}(s \mid \tau_{i+1}).$$

where p_π^{end} is the terminal-state distribution induced by π .

The mismatch between $p_\pi^{\text{end}}(s \mid \tau_i)$ and $p_\mathcal{D}^{\text{ready}}(s \mid \tau_{i+1})$ is the skill-chaining gap. We define the required *transition* as an action sequence $\mathbf{a}_i^{\text{tr}}$ that moves the robot to a valid pre-contact state for τ_{i+1} :

$$s_{\text{cur}} \xrightarrow{\mathbf{a}_i^{\text{tr}}} s_{\text{ready}}, \quad s_{\text{ready}} \sim p_\mathcal{D}^{\text{ready}}(s \mid \tau_{i+1}). \quad (3)$$

AtomBridge retrieves s_{ready} from an independent demonstration of τ_{i+1} and generates $\mathbf{a}_i^{\text{tr}}$ only at task boundaries. It therefore composes fixed VLA policies without additional VLA training (Fig. 2).

3.1 Intervention Switch

At discrete observation step t , the switch receives the initial image frame I_0 , current image frame I_t , and task description τ_i . We use a frozen CLIP image-text encoder with trainable parameters θ outputs the completion probability $p_t = f_\theta(I_0, I_t, \tau_i)$. Let w be the required number of consecutive positive predictions, η the completion threshold, H_i the maximum execution time for τ_i . The binary switch decision z_t is activated when the latest w predictions exceed η , or when $t \geq H_i$:

$$z_t = \begin{cases} 1, & \left(\bigwedge_{j=t-w+1}^t p_j \geq \eta \right) \vee (t \geq H_i), \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Here, j indexes the prediction steps in the latest window. The temporal condition suppresses transient false positives.

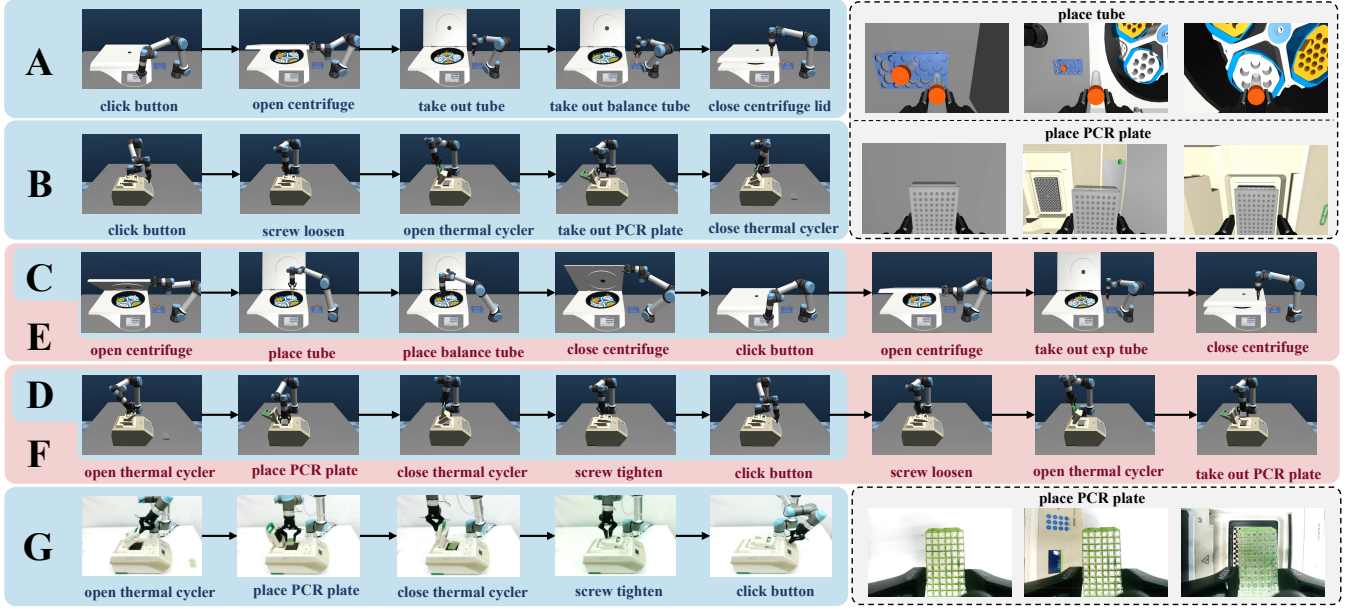


Figure 3: Definitions of the long-horizon scientific operation tasks. Tasks A, B, C, D and G are long-horizon task sequences of length 5, whereas Tasks E and F are long-horizon task sequences of length 8. The atomic tasks inside the dotted box require higher operation precision.

When $z_t = 1$, AtomBridge pauses the current policy, executes $\mathbf{a}_i^{\text{tr}}$, resets the history, and starts the next policy. Training labels are supplied by task-specific physics success predicates, and splits are made by complete trajectory. Model and training details are deferred to the supplementary material.

3.2 Transition Generation

For each atomic task, AtomBridge stores one offline demonstration containing its description, observations, and robot states. This offline demonstration is called *memory*, denoted by $\mathcal{M}$. For the next task, the ready-state retrieval tool searches its matched memory for a usable ready state close to the last pre-operation frame r :

$$s_{\text{ready}} = \text{Retrieve}(\mathcal{M}, \tau_{i+1}). \quad (5)$$

A task-conditioned visual judge and a fast sliding-window search algorithm identify the onset of contact or manipulation. Given the next-task description and an earlier-later pair of memory frames, the task-conditioned visual judge determines whether either frame has exceeded ready state, for example through target contact, manipulation, or an unusably close gripper configuration. The deterministic fast sliding-window search begins from a mid-trajectory window, accepts a non-exceeded later endpoint, or contracts and shifts the window left under a fixed judgment budget to return a usable pre-contact state near r . The visual-judge prompt and the retrieval algorithm details are provided in the supplementary material.

Using the retrieved ready state together with calibrated camera views V and calibration parameters κ , the planning agent determines the transition from s_{cur} to s_{ready} . The cod-

ing agent converts them into template Python commands:

$$\mathbf{a}_i^{\text{tr}} = \text{Code}(\text{Plan}(s_{\text{cur}}, s_{\text{ready}}, V, \kappa)). \quad (6)$$

The host accepts only fixed motion commands (e.g., `open_gripper`, `translate`, `rotate`) that satisfy schema, limit, transition-only, and collision checks. Implementation details are provided in the supplementary material.

4 Experiments

4.1 Experimental Setup

Baseline Settings. We use two VLA models, π_0 (Black et al. 2024) and $\pi_{0.5}$ (Black et al. 2025), as base policies. For each policy, we construct two baselines. *Vanilla* executes the atomic-task policies consecutively without inserting a transition. *MP* inserts transitions generated from conventional motion planning by bidirectional joint-space RRT-Connect (Jung et al. 2025) between consecutive atomic tasks, but uses no LLM agent or agent-generated code. Its ready state is randomly sampled from the initial states of the matched next atomic task. If the task description cannot be matched, it will randomly sample a initial state from the memory as the ready state. *AtomBridge* uses the same base policies, and uses Qwen3.5-397B-A17B as agents’ base model. Its plugin-equipped variants are denoted AtomBridge- π_0 and AtomBridge- $\pi_{0.5}$.

Data Collection. Across simulation and real-world settings, we collect 100 demonstrations for each of 19 atomic tasks. The tasks manipulate scientific instruments and consumables such as centrifuges, thermal cyclers, PCR plates, and centrifuge tubes, etc. The simulation scenes are built on the Mujoco engine by using Autobio (Lan et al. 2025) 3D assets. We compose these atomic tasks into seven long-horizon

Task	Method	Success Rate Per Atomic Task								Full-Seq. Success
		1	2	3	4	5	6	7	8	
A	π_0 (vanilla)	20/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	$\pi_{0.5}$ (vanilla)	20/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	π_0 (MP)	20/20	<u>6/20</u>	0/20	0/20	0/20	–	–	–	0/20
	$\pi_{0.5}$ (MP)	20/20	<u>3/20</u>	0/20	0/20	<u>1/20</u>	–	–	–	0/20
	π_0 (AtomBridge)	20/20	20/20	<u>5/20</u>	0/20	15/20	–	–	–	0/20
	$\pi_{0.5}$ (AtomBridge)	20/20	20/20	10/20	5/20	15/20	–	–	–	3/20
B	π_0 (vanilla)	20/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	$\pi_{0.5}$ (vanilla)	20/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	π_0 (MP)	20/20	<u>14/20</u>	<u>3/20</u>	0/20	<u>1/20</u>	–	–	–	0/20
	$\pi_{0.5}$ (MP)	20/20	<u>10/20</u>	<u>3/20</u>	0/20	<u>3/20</u>	–	–	–	0/20
	π_0 (AtomBridge)	20/20	20/20	20/20	14/20	14/20	–	–	–	14/20
	$\pi_{0.5}$ (AtomBridge)	20/20	20/20	<u>14/20</u>	<u>10/20</u>	<u>8/20</u>	–	–	–	<u>7/20</u>
C	π_0 (vanilla)	16/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	$\pi_{0.5}$ (vanilla)	18/20	0/20	0/20	0/20	0/20	–	–	–	0/20
	π_0 (MP)	15/20	0/20	0/20	0/20	<u>8/20</u>	–	–	–	0/20
	$\pi_{0.5}$ (MP)	18/20	<u>1/20</u>	<u>2/20</u>	<u>3/20</u>	<u>7/20</u>	–	–	–	0/20
	π_0 (AtomBridge)	16/20	<u>6/20</u>	<u>4/20</u>	<u>14/20</u>	<u>6/20</u>	–	–	–	<u>2/20</u>
	$\pi_{0.5}$ (AtomBridge)	18/20	13/20	12/20	18/20	12/20	–	–	–	5/20
D	π_0 (vanilla)	18/20	0/20	12/20	0/20	0/20	–	–	–	0/20
	$\pi_{0.5}$ (vanilla)	18/20	0/20	12/20	0/20	0/20	–	–	–	0/20
	π_0 (MP)	17/20	<u>1/20</u>	<u>13/20</u>	<u>13/20</u>	<u>2/20</u>	–	–	–	0/20
	$\pi_{0.5}$ (MP)	20/20	<u>4/20</u>	<u>13/20</u>	<u>5/20</u>	<u>5/20</u>	–	–	–	0/20
	π_0 (AtomBridge)	19/20	<u>16/20</u>	<u>11/20</u>	14/20	18/20	–	–	–	<u>6/20</u>
	$\pi_{0.5}$ (AtomBridge)	19/20	17/20	14/20	14/20	<u>9/20</u>	–	–	–	8/20
E	π_0 (vanilla)	17/20	0/20	0/20	0/20	0/20	0/20	0/20	0/20	0/20
	$\pi_{0.5}$ (vanilla)	17/20	0/20	0/20	0/20	0/20	0/20	0/20	0/20	0/20
	π_0 (MP)	17/20	<u>1/20</u>	<u>1/20</u>	0/20	13/20	<u>1/20</u>	0/20	0/20	0/20
	$\pi_{0.5}$ (MP)	18/20	<u>1/20</u>	<u>4/20</u>	<u>2/20</u>	<u>8/20</u>	0/20	0/20	<u>3/20</u>	0/20
	π_0 (AtomBridge)	16/20	<u>6/20</u>	<u>3/20</u>	<u>12/20</u>	<u>4/20</u>	<u>15/20</u>	<u>8/20</u>	<u>13/20</u>	<u>2/20</u>
	$\pi_{0.5}$ (AtomBridge)	18/20	13/20	12/20	17/20	<u>7/20</u>	18/20	10/20	16/20	5/20
F	π_0 (vanilla)	19/20	0/20	<u>15/20</u>	0/20	0/20	0/20	0/20	0/20	0/20
	$\pi_{0.5}$ (vanilla)	19/20	0/20	<u>13/20</u>	0/20	0/20	0/20	0/20	0/20	0/20
	π_0 (MP)	19/20	<u>5/20</u>	<u>11/20</u>	<u>12/20</u>	<u>1/20</u>	<u>3/20</u>	<u>2/20</u>	0/20	0/20
	$\pi_{0.5}$ (MP)	20/20	<u>2/20</u>	<u>13/20</u>	<u>5/20</u>	<u>9/20</u>	0/20	<u>2/20</u>	0/20	0/20
	π_0 (AtomBridge)	19/20	<u>16/20</u>	<u>10/20</u>	<u>12/20</u>	<u>16/20</u>	<u>7/20</u>	<u>12/20</u>	10/20	<u>2/20</u>
	$\pi_{0.5}$ (AtomBridge)	20/20	20/20	16/20	16/20	20/20	8/20	18/20	10/20	4/20

Table 1: Per-atomic-task and full-sequence success rates for Tasks A–F. Dashes indicate atomic-task positions not present in the five-step tasks A–D.

task suites (A–G in Fig. 3). Tasks C, D, and G represent experiment-execution workflows. Tasks A and B represent experiment-collection workflows. Tasks E and F represent complete experimental workflows. Detailed collection protocols, state randomization, and task-success annotations are provided in the supplementary material.

Evaluation Metrics. For each method and long-horizon task suite, we run $N = 20$ episodes without resetting the environment between atomic tasks. Atomic task i in episode n is successful, $s_i^{(n)} = 1$, only if its task-specific state predicates are satisfied at both the start and the end of execution. Otherwise, $s_i^{(n)} = 0$. For example, `open lid` requires a closed, locked lid at the start and a fully open lid at the end.

For a suite of K atomic tasks, the entries under *Success Rate Per Atomic Task* in Tab. 1 are computed as

$$SR_i = \frac{1}{N} \sum_{n=1}^N s_i^{(n)}. \quad (7)$$

Thus, an entry of $a/20$ means that atomic task i succeeded in a of the 20 continuous episodes. The *Full-Seq. Success* column requires every atomic task in the same episode to succeed:

$$FSR = \frac{1}{N} \sum_{n=1}^N \prod_{i=1}^K s_i^{(n)}. \quad (8)$$

Accordingly, $a/20$ in this column denotes a complete task sequences with no failed atomic task.

4.2 Scientific Operation Results

Tab. 1 shows that AtomBridge improves most later-step atomic-task success rates relative to the matched transition baseline. Across the 60 Atomic 2 onward comparisons, 53 increase, two are unchanged, and five decrease. The decreases occur in isolated task-model pairs, including Atomic E.5, where AtomBridge- π_0 decreases from 13/20 to 4/20.

The full-sequence results show a clearer long-horizon benefit. All baseline configurations obtain 0/20 full-sequence success, whereas AtomBridge obtains non-zero full-sequence success in 11 of the 12 task-model configurations, ranging from 2/20 to 14/20. AtomBridge- $\pi_{0.5}$ achieves a higher full-sequence success rate than AtomBridge- π_0 in five of the six long tasks. Task B is the exception, where AtomBridge- π_0 reaches 14/20, and AtomBridge- $\pi_{0.5}$ reaches 7/20. These results indicate that AtomBridge can improve execution continuity across atomic tasks, while also showing that transition handling alone does not eliminate failures in individual atomic policies or guarantee end-to-end success.

We further evaluated physical execution in the thermal-cycler scene of Task G (Fig. 3). As shown in Tab. 2, AtomBridge- $\pi_{0.5}$ succeeded in 4/10 trials for Atomic Tasks 3-5 and for the full sequence, whereas the vanilla policy achieved 0/10 in each of these cases. Both methods succeeded in 5/10 trials on Atomic Task 2, consistent with the close match between the end state of Atomic Task 1 and the initial state of Atomic Task 2 in the collected demonstrations. These real-environment results support the benefit of AtomBridge specifically at task boundaries that require a state transition.

Method	Success Rate Per Atomic Task					Full-Seq. Success
	1	2	3	4	5	
$\pi_{0.5}$ (vanilla)	5/10	5/10	0/10	0/10	0/10	0/10
$\pi_{0.5}$ (AtomBridge)	5/10	5/10	4/10	4/10	4/10	4/10

Table 2: Per-atomic-task and full-sequence success rates for the five-step Task G in a real environment.

4.3 Completion Model Evaluation

We evaluate the completion model on 10 episodes for each of the seven thermal-cycler atomic tasks (70 episodes and 34,872 stride-10 frames). The model receives the initial frame, current frame, and task description. Simulator success predicates provide frame-level labels. For all reported experiments, we use the shared default threshold of 0.05. Figure 4 reports mean per-episode accuracy, precision, and recall, together with the temporal distributions of ground-truth positive, false-positive, and false-negative frames over 20 normalized execution bins.

Across all frames, the model achieved 92.2% accuracy, 59.1% precision, and 84.7% recall. It was nearly exact for persistent end states: closing the lid reached 99.8% accuracy with 96.2% precision and 100% recall, and removing the PCR plate reached 99.9% accuracy with 99.6% precision and 99.9% recall. In contrast, placement, button pressing, and knob tightening retained perfect recall but had lower precision, indicating early completion predictions rather than missed completions.

The open-lid task obtained 96.7% accuracy but zero precision and recall. This is because collection terminates immediately after the lid opens, leaving only a short positive segment and no post-completion robot motion.

Method	Key Tools		Eval.	
	Ready retrieval	Collision Check	D.2 Success	Contact Pair-Samples
No-check	✓	×	10/20	165.35
No-retrieval	×	✓	0/20	82.50
check+retrieval	✓	✓	19/20	18.00

Table 3: Ablation results on the long task D.1→D.2. A check mark indicates an enabled tool.

Model	Transition Generation Time	Full-Seq. Time	D.2 Success
GPT-5.2	38s (78%)	49s (100%)	10/10
Qwen3.5-397B-A17B	185s (81%)	227s (100%)	8/10
Qwen3.5-35B-A3B	69s (60%)	115s (100%)	7/10
Qwen3.5-9B	55s (56%)	99s (100%)	5/10

Table 4: Comparison of the proprietary GPT-5.2 model and three open-source Qwen3.5 variants on the D.1→D.2 task. Parenthetical values report each duration as a proportion of the complete long-horizon task time, and D.2 success rates are measured over 10 trials.

The heatmaps localize these errors. Ground-truth positives concentrate at the end of every trajectory. False positives occur before this terminal region for PCR-plate placement, button pressing, and knob tightening, whereas false negatives concentrate at the end of lid opening and knob loosening. Thus, the completion signal is reliable for detecting a candidate task boundary, but ambiguous visual changes require temporal confirmation. All reported experiments use the default threshold together with a short consecutive-positive window. Increasing this window can mitigate premature predictions and we set it to 10 for all experiments. In principle, selecting task-type-specific window lengths could further improve intervention effectiveness. If the completion model does not fire, as in the open-lid case, the task-specific maximum execution time triggers the intervention switch as a fallback.

4.4 Ablation Studies

We study the D.1→D.2 transition because it combines instrument opening with high-precision placement. As shown in Tab. 3, the ablation compares the complete pipeline with No-check and No-retrieval variants. Because collision check and ready-state retrieval are key tools in the whole pipeline. For each setting, we collect 20 valid trials in which D.1 succeeds and report the conditional D.2 success rate and transition *Contact Pair-Samples*. The conditional D.2 success rate is the success rate of D.2 under the condition that D.1 succeeds. Meanwhile, *Contact Pair-Samples* is a metric to quantify collision severity during transitions. This metric counts the sampled contact pairs between the robot arm and environment objects across execution time. It aggregates contact incidence and duration because multiple pairs may be recorded at each sampled time step. Lower values there-

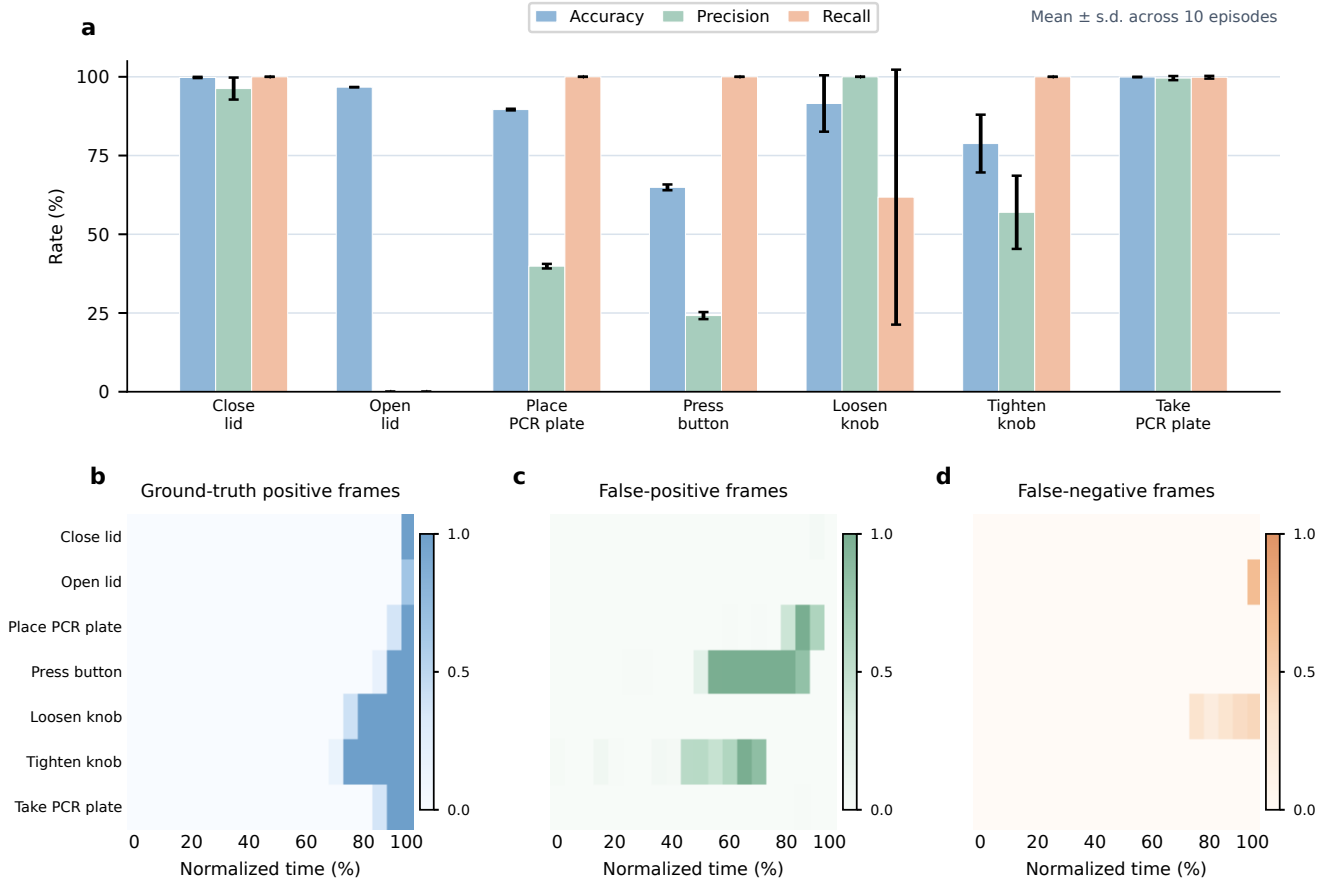


Figure 4: Completion-model evaluation over ten episodes for each thermal-cycler atomic task. **a**, Mean frame-level accuracy, precision, and recall. Error bars denote standard deviations across episodes. **b–d**, Fractions of ground-truth positive, false-positive, and false-negative frames in 20 normalized execution-time bins.

fore indicate fewer accumulated contact pair samples under this sampling scheme, not a smaller physical collision area or guaranteed safety. This compound comparison measures ready-state target guidance as a whole and does not isolate retrieval alone. The complete pipeline achieves the highest conditional D.2 success rate and the fewest contact pair samples.

4.5 Latency and Generation Quality

As shown in Tab. 4, we compare the proprietary GPT-5.2 model with three open-source Qwen3.5 variants on D.1→D.2 over 10 trials. VLA inference uses a single NVIDIA 5070 GPU. All LLMs are accessed through remote inference services, so the reported transition generation times include both server-side inference and network communication latency. GPT-5.2 achieves the highest observed D.2 success rate (10/10) and the shortest transition generation time (38s), which accounts for 78% of the 49s task time. Within the open-source Qwen3.5 family, D.2 success decreases from 8/10 for the 397B model to 7/10 for the 35B model and 5/10 for the 9B model. Transition generation time also decreases from 185s to 69s and 55s, respectively. These open-source results indi-

cate a latency-quality trade-off within the Qwen3.5 family. Because each model is evaluated over only 10 trials, the cross-family results are preliminary. Moreover, the reported timings represent end-to-end remote deployment measurements rather than hardware-normalized model-inference times.

5 Conclusion

This paper proposes AtomBridge, an inference plugin that composes fine-tuned atomic VLA policies without additional VLA fine-tuning or long-horizon demonstrations. Across the evaluated simulated sequences, AtomBridge improves per-step success for most subsequent atomic tasks and increases execution continuity. Ultimately improving the execution success rate for full-sequence tasks.

However, AtomBridge does not explicitly reason about or restore non-robot states, such as the states of manipulated objects and laboratory instruments. Future work could extend AtomBridge to perceive and restore object and instrument states, thereby addressing a broader portion of the skill-chaining gap. In addition, it is necessary to reduce remote LLM inference latency and improve the robustness of visual completion detection under occlusion and real-world

domain shifts.

References

- Beyer, L.; Steiner, A.; Pinto, A. S.; Kolesnikov, A.; Wang, X.; Salz, D.; Neumann, M.; Alabdulmohsin, I.; Tschannen, M.; Bugliarello, E.; et al. 2024. PaliGemma: A versatile 3B VLM for transfer. *CoRR*.
- Bi, K.; Xie, L.; Zhang, H.; Chen, X.; Gu, X.; and Tian, Q. 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970): 533–538.
- Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M. R.; Finn, C.; Fusai, N.; Galliker, M. Y.; et al. 2025. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning*.
- Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π_0 : A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.
- Boiko, D. A.; MacKnight, R.; Kline, B.; and Gomes, G. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992): 570–578.
- Boyd, J. 2002. Robotic laboratory automation. *Science*, 295(5554): 517–518.
- Burger, B.; Maffettone, P. M.; Gusev, V. V.; Aitchison, C. M.; Bai, Y.; Wang, X.; Li, X.; Alston, B. M.; Li, B.; Clowes, R.; et al. 2020. A mobile robotic chemist. *Nature*, 583(7815): 237–241.
- Chen, Y.; Wang, C.; Fei-Fei, L.; and Liu, K. 2023. Sequential Dexterity: Chaining Dexterous Policies for Long-Horizon Manipulation. In *Conference on Robot Learning*, 3809–3829. PMLR.
- Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11): 1684–1704.
- Clegg, A.; Yu, W.; Tan, J.; Liu, C. K.; and Turk, G. 2018. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(6): 1–10.
- Coley, C. W.; Eyke, N. S.; and Jensen, K. F. 2020. Autonomous discovery in the chemical sciences part I: Progress. *Angewandte Chemie International Edition*, 59(51): 22858–22893.
- Coley, C. W.; Thomas III, D. A.; Lummiss, J. A.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; et al. 2019. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453): eaax1566.
- Dai, Y.; Chan, H.; Vriza, A.; Fan, J.; Kim, F.; Wang, Y.; Liu, W.; Shan, N.; Xu, J.; Weires, M.; et al. 2025. Adaptive AI decision interface for autonomous electronic material discovery. *Nature Chemical Engineering*, 1–11.
- Dietterich, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13: 227–303.
- Ebert, F.; Dasari, S.; Lee, A. X.; Levine, S.; and Finn, C. 2018. Robustness via Retrying: Closed-Loop Robotic Manipulation with Self-Supervised Learning. In Billard, A.; Dragan, A.; Peters, J.; and Morimoto, J., eds., *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, 983–993. PMLR.
- Fan, Y.; Bai, S.; Tong, X.; Ding, P.; Zhu, Y.; Lu, H.; Dai, F.; Zhao, W.; Liu, Y.; Huang, S.; et al. 2025. Long-VLA: Unleashing Long-Horizon Capability of Vision Language Action Model for Robot Manipulation. In *Conference on Robot Learning*, 2018–2037. PMLR.
- Gupta, A.; Kumar, V.; Lynch, C.; Levine, S.; and Hausman, K. 2020. Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. In *Conference on Robot Learning*, 1025–1037. PMLR.
- Hou, Y.; Ma, J.; Sun, H.; and Wu, F. 2024. Effective Offline Robot Learning With Structured Task Graph. *IEEE Robotics and Automation Letters*, 9(4): 3633–3640.
- Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.
- Jung, H.; Lee, D.; Park, H.; Kim, J.; and Kim, B. 2025. From planning to policy: distilling Skill-RRT for long-horizon prehensile and non-prehensile manipulation. *arXiv e-prints*, arXiv-2502.
- Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; et al. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning*, 2679–2713. PMLR.
- King, R. D.; Rowland, J.; Oliver, S. G.; Young, M.; Aubrey, W.; Byrne, E.; Liakata, M.; Markham, M.; Pir, P.; Soldatova, L. N.; et al. 2009. The automation of science. *Science*, 324(5923): 85–89.
- Konidaris, G.; and Barto, A. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in neural information processing systems*, 22.
- Koscher, B. A.; Canty, R. B.; McDonald, M. A.; Greenman, K. P.; McGill, C. J.; Bilodeau, C. L.; Jin, W.; Wu, H.; Vermeire, F. H.; Jin, B.; et al. 2023. Autonomous, multiproperty-driven molecular discovery: From predictions to measurements and back. *Science*, 382(6677): eadi1407.
- Lan, Z.; Jiang, Y.; Wang, R.; Xie, X.; Zhang, R.; Zhu, Y.; Li, P.; Yang, T.; Chen, T.; Gao, H.; et al. 2025. Autobio: A simulation and benchmark for robotic automation in digital biology laboratory. *arXiv preprint arXiv:2505.14030*.
- Lee, Y.; Lim, J. J.; Anandkumar, A.; and Zhu, Y. 2022. Adversarial Skill Chaining for Long-Horizon Robot Manipulation via Terminal State Regularization. In *Conference on Robot Learning*, 406–416. PMLR.
- Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2025. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. In *The Thirteenth International Conference on Learning Representations*.

- Macarron, R.; Banks, M. N.; Bojanic, D.; Burns, D. J.; Cirovic, D. A.; Garyantes, T.; Green, D. V.; Hertzberg, R. P.; Janzen, W. P.; Paslay, J. W.; et al. 2011. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, 10(3): 188–195.
- MacLeod, B. P.; Parlane, F. G.; Morrissey, T. D.; Häse, F.; Roch, L. M.; Dettelbach, K. E.; Moreira, R.; Yunker, L. P.; Rooney, M. B.; Deeth, J. R.; et al. 2020. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20): eaaz8867.
- Mishra, U. A.; Xue, S.; Chen, Y.; and Xu, D. 2023. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, 2905–2925. PMLR.
- Novikov, A.; Vü, N.; Eisenberger, M.; Dupont, E.; Huang, P.-S.; Wagner, A. Z.; Shirobokov, S.; Kozlovskii, B.; Ruiz, F. J.; Mehrabian, A.; et al. 2025. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.
- Ochiai, K.; Tahara-Arai, Y.; Kato, A.; Kaizu, K.; Kariyazaki, H.; Umeno, M.; Takahashi, K.; Kanda, G. N.; and Ozaki, H. 2025. Automating Care by Self-maintainability for Full Laboratory Automation. *arXiv preprint arXiv:2501.05789*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.
- Ren, B.; Liu, X.; Chen, X.; Liu, Y.; Li, C.; Gao, D.; Su, Z.; Xing, J.; Xue, Z.; Li, R.; et al. 2026. LabVLA: Grounding Vision-Language-Action Models in Scientific Laboratories. *arXiv preprint arXiv:2606.13578*.
- Sung, Y.; Wang, Z.; and Stone, P. 2023. Learning to Correct Mistakes: Backjumping in Long-Horizon Task and Motion Planning. In Liu, K.; Kulic, D.; and Ichnowski, J., eds., *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, 2115–2124. PMLR.
- Szymanski, N. J.; Rendy, B.; Fei, Y.; Kumar, R. E.; He, T.; Milsted, D.; McDermott, M. J.; Gallant, M.; Cubuk, E. D.; Merchant, A.; et al. 2023. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990): 86–91.
- Volk, A. A.; Epps, R. W.; Yonemoto, D. T.; Masters, B. S.; Castellano, F. N.; Reyes, K. G.; and Abolhasani, M. 2023. AlphaFlow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nature Communications*, 14(1): 1403.
- Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; et al. 2023. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60.
- Yang, Y.; Sun, J.; Kou, S.; Wang, Y.; and Deng, Z. 2025. LoHoVLA: A Unified Vision-Language-Action Model for Long-Horizon Embodied Tasks. *arXiv preprint arXiv:2506.00411*.
- Yokoyama, N.; Clegg, A.; Truong, J.; Undersander, E.; Yang, T.-Y.; Arnaud, S.; Ha, S.; Batra, D.; and Rai, A. 2024. ASC: Adaptive Skill Coordination for Robotic Mobile Manipulation. *IEEE Robotics and Automation Letters*, 9(1): 779–786.
- Zhang, J.; Huang, J.; Jin, S.; and Lu, S. 2024. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8): 5625–5644.
- Zhao, Y.; Zhang, J.; Xu, Z.; Sun, S.; Langner, S.; Hartono, N. T. P.; Heumueller, T.; Hou, Y.; Elia, J.; Li, N.; et al. 2021. Discovery of temperature-induced stability reversal in perovskites using high-throughput robotic learning. *Nature Communications*, 12(1): 2191.