RoboChemist: Long-Horizon and Safety-Compliant Robotic Chemical Experimentation

Zongzheng Zhang[*1], **Chenghao Yue**[*1], **Haobo Xu**[*2],
Minwen Liao[1], **Xianglin Qi**[1], **Huan-ang Gao**[1], **Ziwei Wang**[3], **Hao Zhao**[† 1,2]

[*]Equal Contribution; [†]Corresponding author

[1] Beijing Academy of Artificial Intelligence, BAAI
[2] Institute for AI Industry Research (AIR), Tsinghua Univeristy
[3] Nanyang Technological University

zhaohao@air.tsinghua.edu.cn

https://zzongzheng0918.github.io/RoboChemist.github.io/

Abstract: Robotic chemists promise to both liberate human experts from repetitive tasks and accelerate scientific discovery, yet remain in their infancy. Chemical experiments involve long-horizon procedures over hazardous and deformable substances, where success requires not only task completion but also strict compliance with experimental norms. To address these challenges, we propose *RoboChemist*, a dual-loop framework that integrates Vision-Language Models (VLMs) with Vision-Language-Action (VLA) models. Unlike prior VLM-based systems (e.g., VoxPoser, ReKep) that rely on depth perception and struggle with transparent labware, and existing VLA systems (e.g., RDT, π_0) that lack semantic-level feedback for complex tasks, our method leverages a VLM to serve as (1) a planner to decompose tasks into primitive actions, (2) a visual prompt generator to guide VLA models, and (3) a monitor to assess task success and regulatory compliance. Notably, we introduce a VLA interface that accepts image-based visual targets from the VLM, enabling precise, goal-conditioned control. Our system successfully executes both primitive actions and complete multi-step chemistry protocols. Results show **23.57%** higher average success rate and a **0.298** average increase in compliance rate over state-of-the-art VLA baselines, while also demonstrating strong generalization to objects and tasks.

Keywords: Robotic Chemistry, VLA Models, Visual Prompting

1 Introduction

Robotic chemists offer the potential to liberate human experts from repetitive and hazardous laboratory procedures, while accelerating scientific discovery [1, 2, 3, 4, 5, 6, 7]. Yet, the automation of chemical experiments remains highly challenging due to the long-horizon, safety-critical nature of lab protocols, which involve deformable substances, delicate glassware, and hazardous materials. These tasks require not only task success but also strict compliance with procedural norms—such as safe grasping points or heating thresholds—to ensure experimental validity and safety. As shown in Figure 1(c), even seemingly simple actions like stirring or pouring become complex when compositional reasoning, dexterous execution, and regulatory adherence are simultaneously required.

Recent progress in Vision-Language Models (VLMs) [8, 9, 10, 11, 12, 13, 14, 15, 16, 17] and Vision-Language-Action (VLA) models [18, 19, 20, 21, 22, 23] has enabled scalable robotic policies across diverse environments. However, their limitations become pronounced in chemical labs. VLM-based systems like VoxPoser [24] and ReKep [25] rely heavily on depth sensors and object segmentation, which struggle with transparent containers and deformable substances [26]. On the other hand, VLA models such as π_0 [21] and RDT [20] excel at action grounding but lack high-level semantic

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

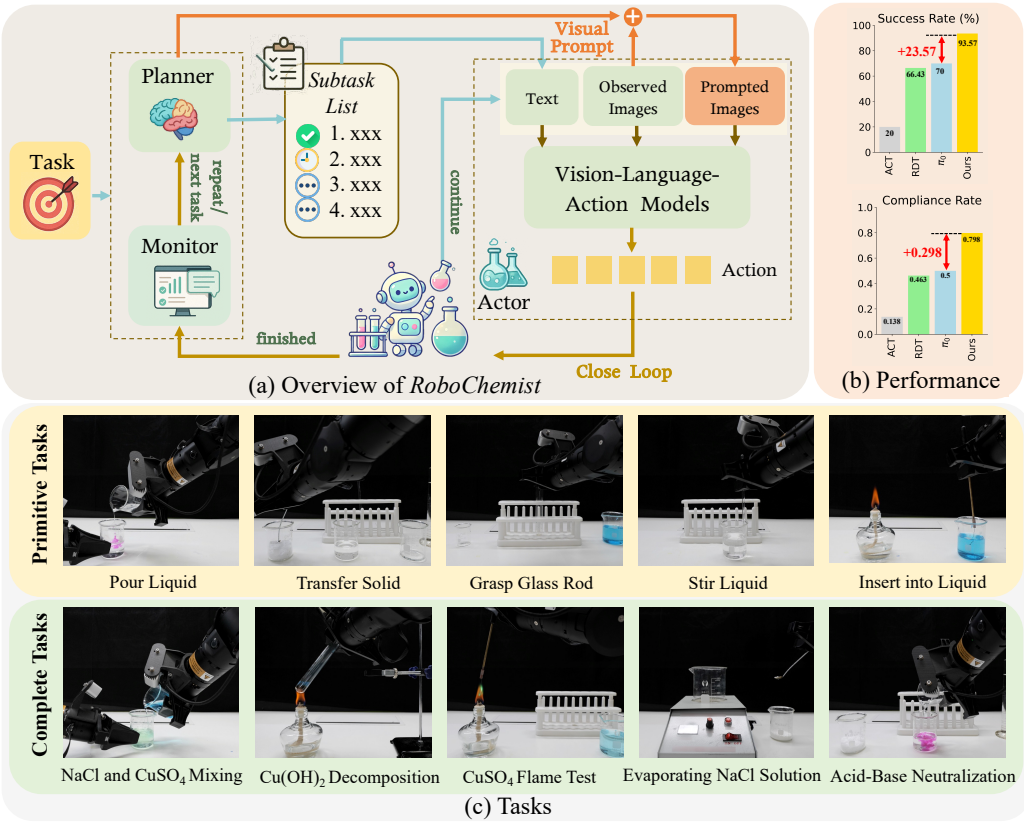

Figure 1: (a) **Overview of *RoboChemist***. The VLM in our system acts as the planner, decomposing high-level tasks into subtasks. Based on each subtask, the VLM generates prompted images through visual prompting and provides them, along with other relevant information, to the VLA models. The VLM also functions as the monitor, assessing the completion status of subtasks, thus ensuring a complete feedback loop in the system. (b) *RoboChemist* outperforms baselines in both primitive tasks and complete chemical experiment tasks. (c) Some tasks performed by *RoboChemist*.

understanding or closed-loop feedback, often resulting in unsafe or failed behaviors in complex tasks. As illustrated in Figure 1(b), such approaches yield low success and compliance rates in primitive lab tasks.

To address these challenges, we propose *RoboChemist*, a dual-loop framework that integrates VLMs and VLAs via visual prompting and semantic supervision. As illustrated in Figure 1(a), *RoboChemist* employs a VLM as: (1) **a planner** that decomposes long-horizon tasks into structured subtasks; (2) **a visual prompt generator** that highlights task-specific grasp or target regions via bounding boxes or keypoints; and (3) **a monitor** that evaluates subtask success and enforces closed-loop corrections. The VLA model executes each primitive task using the prompted image, observed state, and textual instruction, enabling goal-conditioned and safety-compliant control. This design allows the system to reason globally while acting locally, combining high-level semantic structure with low-level dexterity.

We evaluate *RoboChemist* across a wide spectrum of chemical tasks, including both primitive operations (e.g., pouring, stirring, grasping) and multi-step complete experiments (e.g., acid-base neutralization, flame tests)(Figure 1(c)). As shown in Figure 1(b), *RoboChemist* achieves significant gains over prior baselines, with a **23.57%** higher average success rate and a **0.298** average increase in compliance rate. The benefits of the closed-loop architecture are particularly evident in long-horizon tasks, where the outer loop enables condition-based re-execution (e.g., pouring until the solution becomes colorless). Additionally, our system generalizes to unseen reagents, containers, and workflows, demonstrating robust skill transfer without task-specific fine-tuning. These capabili-

ties are further illustrated in Figure 5, showcasing diverse experimental scenarios handled effectively by *RoboChemist*.

To summarize, we make three contributions in this paper: (1) We introduce a dual-loop framework that unifies vision-language reasoning and low-level action grounding, with the VLM acting as planner, prompt generator, and monitor. (2) We propose an instruction-aware visual prompting strategy using Qwen2.5-VL, enabling precise and safe manipulation of transparent and hazardous materials in complex chemical setups. (3) *RoboChemist* outperforms strong VLA and VLM baselines in both atomic and long-horizon tasks, demonstrating superior success, compliance, and generalization.

2 Related Work

Vision-Language-Action Models. Large Language Models (LLMs) [27, 28, 29, 30, 31, 32] and Vision-Language Models (VLMs) [8, 9, 10, 11, 12, 33] have enabled scalable robot foundation models [34, 18, 19]. With robot learning datasets expanding from a few thousand [35, 36] to around one million samples [37, 38], and tasks evolving from single-arm [39] to dual-arm [20, 40], Vision-Language-Action (VLA) architectures are now being applied to embodied tasks [41, 42, 43, 44, 45, 46, 47, 48, 49]. Recent VLA models differ in both perception and control design. While many use 2D visual inputs [21, 50, 51, 52, 53], others incorporate 3D scene representations for richer spatial understanding [54, 55]. On the action side, methods span simple MLP regressors [21, 50, 53], autoregressive policy token decoders [40, 51, 56], and diffusion-based policies [20]. Hybrid architecture [57] and token-efficient designs [22] further aim to balance performance and scalability.

Visual Prompting. Visual prompting [58, 59, 60, 61, 62, 63, 64, 65] has emerged as an effective strategy for guiding vision-language models by embedding task-specific visual cues into inputs. Recent visual prompting methods in robotics extract keypoints from RGB images via object masks, often using segmentation models [66, 67, 68, 69] followed by Farthest Point Sampling (FPS) [70] or K-means [71], as in ReKep [25] and KUDA [72]. MOKA [73] further integrates VLMs to select keypoints from candidates post-hoc. However, these pipelines lack direct use of textual prompts during keypoint generation, limiting adherence to instruction-level constraints—especially important in chemistry. In contrast, our method leverages advanced VLMs like Qwen2.5-VL [12] to produce visual prompts that are instruction-aware from the outset.

Robotic Automation in Chemistry. Robotic automation in chemistry has advanced significantly, covering organic synthesis [4, 7], material synthesis [2], microplate handling [74], and mobile exploratory chemists [1, 75, 3]. However, these systems typically depend on specialized hardware and predefined instructions, restricting flexibility and generalization across tasks [5, 6]. Recent efforts, including Organa [76], ArChemist [77], and liquid handling approaches [78, 79], partially address these limitations but remain predominantly single-arm and scenario-specific. The Chemistry3D benchmark [80] introduces a simulated environment to evaluate general-purpose manipulation in chemistry contexts, failing to address the complexities of real-world chemical experimentation.

In contrast, our work introduces a generalized dual-arm collaborative approach utilizing foundational VLA models. This enables versatile manipulation across diverse chemical experiments without reliance on specialized hardware or rigid task-specific programming, aiming to improve practicality and scalability in robotic laboratory applications.

3 Method

In this section, we first present an overview of the *RoboChemist* pipeline in Sec. 3.1. Sec. 3.2 covers the motivation and approach using visual prompting, while Sec. 3.3 focuses on the design and implementation of the closed-loop system.

3.1 Overview

Figure 1(a) illustrates the overall pipeline of *RoboChemist*, consisting of a closed-loop system formed by the VLM and VLA model. When the researcher specifies a chemical experiment task along with the apparatus and reagents, the VLM acts as a planner, decomposing the task into a sequence of executable primitive tasks. For each subtask, considering the specific operation proto-

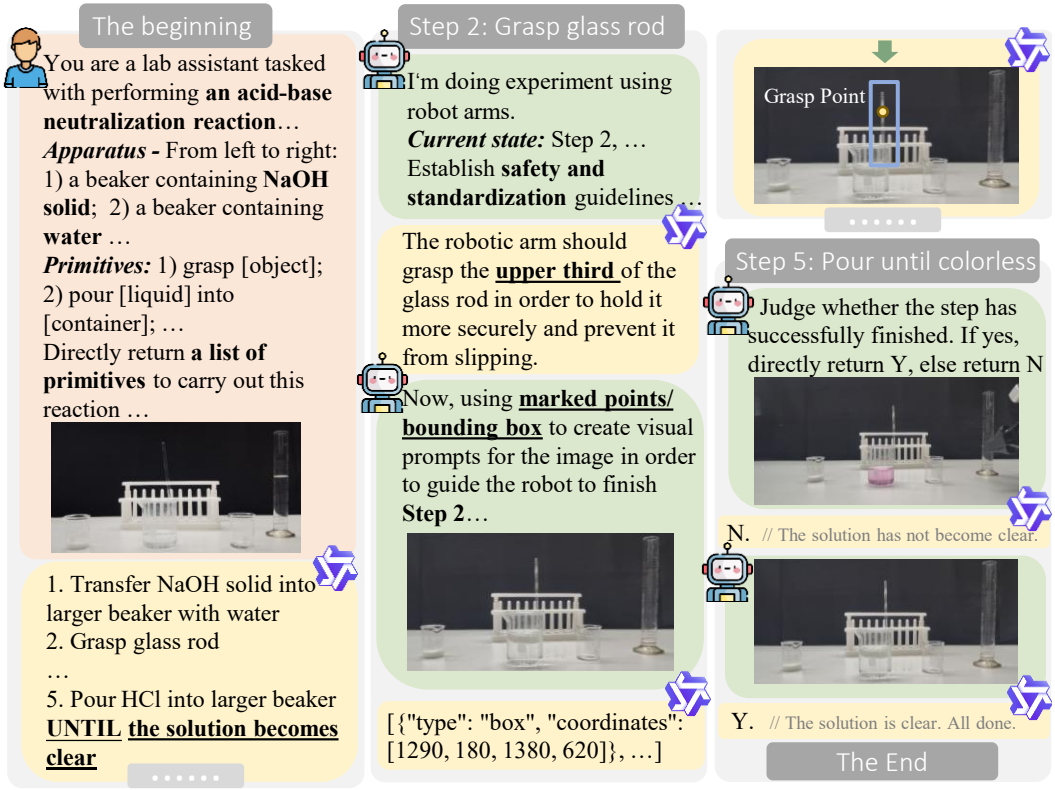

Figure 2: **Illustration of the acid-base neutralization reaction experiment.** 1) **The Beginning**: The researcher provides the complete task instructions, initial scene setup and available primitive tasks, followed by *RoboChemist*'s task decomposition. 2) **Step 2**: During the glass rod grasping step, *RoboChemist* uses visual prompting to highlight the bbox and grasp point to adhere to safety guidelines, providing reference for subsequent actions. 3) **Step 5**: *RoboChemist*'s closed-loop system observes the experiment's state and continues adding acid until the solution turns colorless, at which point the task is completed and the experiment is terminated.

cols of chemical experiments, the VLM generates detailed guidelines, then uses visual prompting to highlight grasp points, target points, or bounding boxes in the front view (as shown in Figure 2). This reference image, along with observations from other camera viewpoints and the corresponding language instructions, is provided to the VLA model to execute each primitive task. After each subtask, the VLM functions as a monitor, evaluating the current state. If the subtask is completed successfully, the system proceeds to the next step; otherwise, the current task is repeated until successful completion. This forms a complete closed-loop feedback system, ensuring both safety and improved success rates, with high interpretability.

3.2 Visual Prompting

Motivation. While VLA models exhibit strong capabilities in perceiving scenes, their abilities remain limited when conditioned solely on text prompts as instructions [81, 44], especially in environment with many visually similar objects, like chemical experiments involving multiple containers. In such cases, visual prompting can largely alleviate the issue. Here, we compare the performance of standalone VLA model with our visual-prompt-based method, *RoboChemist*, on a liquid pouring task, which requires transferring liquid between two specific cups among a set of two or three. Results in Table 1 indicate that performance degrades as the number of cups increases, underscoring the necessity of visual prompting.

Success Rate	Two Cups	Three Cups
w/o visual prompt	16/20	5/20
w/ visual prompt	**19**/20	**17**/20

Table 1: Comparison of different methods for pouring liquid between cups.

However, while there are several visual prompt-based approaches, most of them face challenges in reconstructing transparent objects or ensuring safe, controlled execution [26]. To illustrate this, we designed an experiment where a robotic arm is instructed to heat solid over a flame, as shown in Figure 3. In this experiment, ReKep [25] struggles to reconstruct the transparent test tube, leading to a failed grasp. Methods that pre-compute grasp candidates without considering textual instructions are generally effective in standard settings but may fail to account for safety and procedural requirements crucial in chemistry experiments. As shown in Figure 3(b), MOKA [73] selects the center as the grasp point, resulting in an unsafe action.

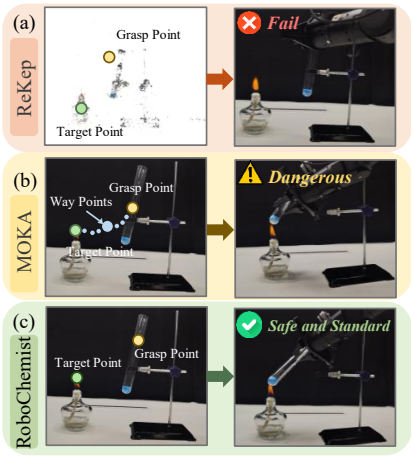

Figure 3: Comparison of different methods for grasping a test tube containing solid for heating over a flame.

Proposed Method. To overcome the limitations of existing methods, we introduce a framework to self-generate task-specific safety guidelines and visual prompts. The method starts by providing the current state of the experiment—including the specific primitive task to be performed and images of the lab bench—as input to a VLM to generate detailed safety guidelines. These generated guidelines include considerations such as safe grasping regions on transparent labware, or chemical handling precautions—all without human intervention. Next, the model is tasked with using these contexts to produce visual prompts in the form of bounding boxes or key points. Thanks to the fine-grained grounding ability of Qwen-2.5-VL [12], we are able to reliably identify critical points/regions—even in visually ambiguous contexts.

The proposed method demonstrates its end-to-end simplicity with high precision. As illustrated in Figure 3(c), the grasp point is placed on the upper body of the test tube—avoiding the heated area—while the target point ensures correct positioning for material heating. A full example of the visual prompt generation process is illustrated in Figure 2, where in Step 2, the robotic arm is tasked with grasping a glass rod. By bridging the gap between perception, instruction generation, and actionable spatial grounding, we establish a new paradigm for safe, instruction-aware robotics in chemically sensitive environments.

3.3 Closed-Loop System Design

Inner Loop Enhancement with Unsuccessful Trials. The VLA models naturally have an inner closed loop, obtaining real-time images and state information as inputs and generating actions to interfere the environment. However, the loop would end after a failed attempt. To enhance the self-correcting ability and robustness of the loop, we introduce scenarios into the training data where, after an unsuccessful trial, the system automatically attempts a second execution. This approach, rather than relying solely on successful one-shot tasks, improves the VLA model's feedback capabilities, resulting in a stronger inner loop.

Outer Loop Establishment with Monitors. The inner loop lacks an explicit feedback mechanism, making it unable to distinguish between success and failure or support long-horizon behaviors that require ongoing monitoring. *RoboChemist* introduces an outer loop integrating the VLM as a monitor. After each inner loop, the current scene images are fed to the VLM, which evaluates whether each primitive task has achieved its intended goal and enables recovery from failures. For complex, multi-repetition tasks, the outer loop can also provide ongoing feedback. For instance, in Step 9 of Figure 2, the initial attempt to pour acid does not fully neutralize the base—evidenced because the phenolphthalein indicator remaining pink. Therefore, the outer loop triggers a re-execution of the pouring primitive until the solution clear. Through this mechanism, the outer loop composes multiple discrete actions into a coherent sequence that approximates continuous behavior.

Additionally, the outer loop functions as a planner, decomposing complex tasks into sequences of primitive subtasks. As shown at the beginning of Figure 2, *RoboChemist* uses the known apparatus configuration to break down an acid-base neutralization reaction into a series of primitives.

4 Experiment

4.1 Experiment Setup

Hardware Platform. The system uses the Cobot Magic ALOHA, a dual-arm robot with 7 degrees of freedom per arm. It is equipped with four Depth Camera D435 units, but we only utilize RGB data from cameras on the two wrists, front, and a top-down viewpoint. An NVIDIA RTX 4090 GPU handles data collection and model inference.

Data Collection. For VLA model fine-tuning, we collect 400 data samples for each primitive task; the full list of primitive tasks is provided in Sec. 4.2. To ensure task generalization, we introduce diversity in both the experimental scene setup and object selection for each basic action. For example, in the *liquid pouring* task, we vary the liquid color (e.g., blue, red, colorless, etc.), volume (e.g., small, medium, large, etc.), container type (e.g., beakers, test tubes, graduated cylinders, etc.), and the surrounding environment layout. The data collection frequency aligns with the inference frequency, both set at 20 Hz. All data is stored in Parquet or HDF5 format for training.

Baselines. To evaluate *RoboChemist*, we consider advanced baselines in robotic bimanual manipulation: ACT [82], RDT [20], and π_0 [21]. All models are fine-tuned using our collected data under the same setup. ACT is a state-of-the-art method in bimanual manipulation, utilizing action chunking with transformers. RDT-1B uses the DiT architecture, the latest dual-arm foundation model. π_0, using the VLA architecture, demonstrates strong generalization across tasks and robotic platforms.

Metric. We evaluate each primitive task using two metrics: **Success Rate** (SR) and **Compliance Rate** (CR). For each task, we conduct 20 trials. SR is calculated as the ratio of successful trials to total trials. To assess the precision of actions in chemical experiments, we introduce CR. For example, in the *Grasp Glass Rod* task, a failed grasp is scored as 0, a grasp at an incorrect position (e.g., not at the 1/3 point of the rod) is scored as 0.5, and a successful and compliant grasp is scored as 1. Detailed CR settings for additional tasks can be found in the Appendix A.1.

Model Training and Inference. We fine-tune the π_0 [21] model with collected data, training each task for 30K steps on four L20 GPUs. During training, in addition to the images from the four camera viewpoints used for data collection, we apply Qwen2.5-VL-72B-Instruct [12] visual prompting to label the grasp and target points in each primitive task's initial experimental scene. These images, along with the language instructions and the proprioceptive state, are used as input to fine-tune the VLA model. To diversify the language instructions, we generate various command variations using GPT-4o [83]. Inference is performed in real-time on an NVIDIA RTX 4090 GPU at 20 Hz.

4.2 Chemical Tasks

Primitive Tasks. To enable a wider range of chemical experiments, we decompose standard chemical tasks into 7 primitive tasks, including grasping a glass rod, heating a platinum wire, inserting platinum wire into a solution, pouring liquid, stirring a solution with a glass rod, transferring the solid, and pressing a button. For a more detailed illustration, please refer to the Appendix A.1. Table 2 compares *RoboChemist* and baselines on these primitive tasks in terms of Success Rate (SR) and Compliance Rate (CR), with all models fine-tuned using the same data. *RoboChemist* outperforms all baselines in both SR and CR across the tasks. *RoboChemist* without the closed-loop system (only the addition of a visual prompting reference image) shows improvement compared to π_0 [21], demonstrating the effectiveness of visual prompting. Furthermore, *RoboChemist* with the outer-loop structure achieves higher success rates than the version without the outer loop, highlighting the effectiveness of the closed-loop system in enhancing primitive task completion.

Complete Tasks. After training the primitive tasks, we proceed to execute complete experimental tasks. We select five complete chemical experiment tasks (detailed description please refer to Appendix A.2), where each task consists of 1 to 5 primitive tasks in order. Figure 4 shows the experi-

Primitive Task	Grasp Glass Rod		Heat Platinum Wire		Insert into Solution		Pour Liquid		Stir the Solution		Transfer the Solid		Press the Button	
Method	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑
ACT [82]	55	0.325	20	0.063	10	0.050	25	0.288	15	0.075	15	0.063	0	0.100
RDT [20]	20	0.100	60	0.363	80	0.775	90	0.675	75	0.400	75	0.513	65	0.413
π_0 [21]	40	0.200	55	0.325	80	0.800	80	0.475	85	0.600	80	0.525	70	0.575
RoboChemist w/o CL	85	0.750	70	0.575	85	0.850	80	0.663	95	0.650	85	0.538	75	0.613
RoboChemist w/ CL	**95**	**0.875**	**90**	**0.800**	**95**	**0.950**	**95**	**0.800**	**100**	**0.825**	**95**	**0.675**	**85**	**0.663**

Table 2: **Primitive tasks results.** We evaluate baselines and *RoboChemist* on seven primitive tasks using Success Rate (SR) and average Compliance Rate (CR). CL denotes Closed-Loop architecture.

Complete Task	Mix NaCl and $Cu(SO_4)$		Heat-induced Decompose $Cu(OH)_2$				Flame Test for $CuSO_4$ Solution							
Primitive Task	Pour Liquid		Grasp Test Tube		Heat Test Tube		Grasp Platinum Wire		Heat Platinum Wire		Insert into Solution		Heat Platinum Wire	
Method	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑
ACT [82]	25	0.250	45	0.250	30	0.150	50	0.500	10	0.062	10	0.100	0	0.000
RDT [20]	80	0.650	10	0.050	5	0.025	20	0.200	10	0.075	10	0.100	5	0.038
π_0 [21]	80	0.650	40	0.250	35	0.200	40	0.400	25	0.200	20	0.200	15	0.125
RoboChemist	**95**	**0.775**	**90**	**0.900**	**80**	**0.450**	**90**	**0.900**	**80**	**0.750**	**80**	**0.800**	**65**	**0.650**
Complete Task	Evaporate NaCl Solution				Acid-Base Neutralization Reaction									
Primitive Task	Transfer NaCl Solid		Press the Button		Transfer NaOH Solid		Grasp Glass Rod		Stir the Solution		Add Phenolphthalein		Add HCl Acid	
Method	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑
ACT [82]	15	0.063	0	0.100	20	0.075	10	0.050	5	0.025	0	0.000	0	0.000
RDT [20]	75	0.513	50	0.363	75	0.513	20	0.100	15	0.075	5	0.038	0	0.013
π_0 [21]	80	0.525	55	0.400	80	0.525	35	0.200	30	0.175	20	0.150	5	0.013
RoboChemist	**95**	**0.675**	**80**	**0.600**	**100**	**0.675**	**95**	**0.850**	**90**	**0.650**	**80**	**0.625**	**40**	**0.300**

Table 3: **Complete tasks results.** We evaluate baselines and *RoboChemist* on five complete chemical experiments using Success Rate (SR) and Compliance Rate (CR). For tasks that require multiple steps, the metrics for each subsequent step are calculated based on the success of the previous step.

mental process and phenomena as *RoboChemist* completes the tasks. Table 3 shows the performance metrics for each primitive task in order. As the experiments progress, other baselines experience a decline in success rate; however, *RoboChemist* performs consistently well in tasks composed of 2-3 primitive tasks, and even successfully completes more complex tasks involving up to five steps. For unseen tasks, such as grasping a test tube, *RoboChemist* also demonstrates strong generalization.

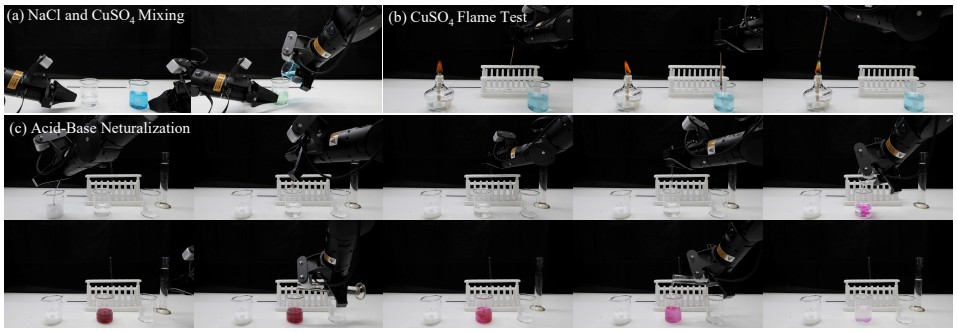

Figure 4: **Visualization of Complete Chemical Experiments.** (a) Complexation reaction: mixing NaCl and $CuSO_4$ to form a light green complex; (b) Copper flame test: dipping a platinum wire into $CuSO_4$ solution and observing a green flame upon heating; (c) Acid-base neutralization: preparing a NaOH solution and using phenolphthalein as an indicator to determine neutralization by HCl.

4.3 Effectiveness of Visual Prompting

To evaluate the effectiveness of our proposed visual prompting method, we compare it with the π_0 [21] baseline and other visual prompt-based methods, including ReKep [25] and MOKA [73]. In all experiments, we generate visual prompts from the same input instructions and use the resulting images as references for the fine-tuned VLA model. Table 4 presents the quantitative results. In the case of ReKep [25], the RGB-D cameras struggle with the reconstruction of transparent objects, leading to the lowest success rates. MOKA [73], not relying on text-based visual prompts, shows relatively lower compliance. In contrast, *RoboChemist*, using a VLM-based visual prompt, achieves the best results, demonstrating the superiority of our method in effectively utilizing visual cues.

4.4 Generalizability

Leveraging the VLM's open vocabulary capability [84] and the VLA's generalization ability, *RoboChemist* demonstrates remarkable adaptability, successfully performing both primitive and complete tasks across a wide range of scenarios. Please refer to Appendix A.4 for more details.

Primitive Task	Grasp Glass Rod		Heat Platinum Wire		Insert into Solution		Pour Liquid		Stir the Solution		Transfer the Solid		Press the Button	
Method	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑
π_0 [21]	40	0.200	55	0.325	80	0.800	**80**	0.475	85	0.600	80	0.525	70	0.363
ReKep [25] + π_0 [21]	35	0.200	30	0.150	**85**	0.825	55	0.375	90	0.500	75	0.313	**75**	0.400
MOKA [73] + π_0 [21]	65	0.350	55	0.338	**85**	**0.850**	75	0.500	**95**	0.600	**85**	0.363	**75**	0.475
RoboChemist w/o CL	**85**	**0.750**	**70**	**0.575**	**85**	**0.850**	**80**	**0.663**	**95**	**0.650**	**85**	**0.538**	**75**	**0.613**

Table 4: Comparison of visual prompt-based methods.

Primitive Task	**Training Data**	**Generalized Tasks**
Pick-and-Place	Grasp a glass rod	Grasp and place test tubes, beakers, graduated cylinders, etc
Stir	Stir the solution with a glass rod	Stir solid reagents or use different objects to stir
Heat	Heat a platinum wire	Heat various types of materials, such as iron wire, magnesium wire, test tubes, etc
Pour liquid	Pour liquid from one container to another	Measure liquid solvents
Insert	Insert platinum wire into a solution	Insert any solid reagent, a thermometer, or place a test tube into cooling liquid.

Table 5: Primitive Tasks Generalization.

Generalization in Primitive Tasks. As shown in Table 5, *RoboChemist* generalizes its learned primitive tasks to various objects and materials. For instance, grasping a glass rod was trained and successfully extends to picking and placing test tubes, beakers, and other containers. These examples highlight its robust capability to apply basic skills to diverse experimental scenarios.

Generalization in Complete Tasks. Although only seven primitive tasks were trained, *RoboChemist* leverages their generalization and the extensive chemical knowledge stored in the VLM to perform a wide range of complete tasks. This ability not only allows it to complete the specified tasks but also enables it to infer and summarize experimental observations, drawing conclusions based on the outcomes. Figure 5 demonstrates how *RoboChemist* can execute the four basic types of chemical reactions—combination, decomposition, displacement, and double displacement reactions. It also explores the physical and chemical properties of different metal elements, such as flame reactions and comparing the reactivity of metals through displacement reactions with solutions.

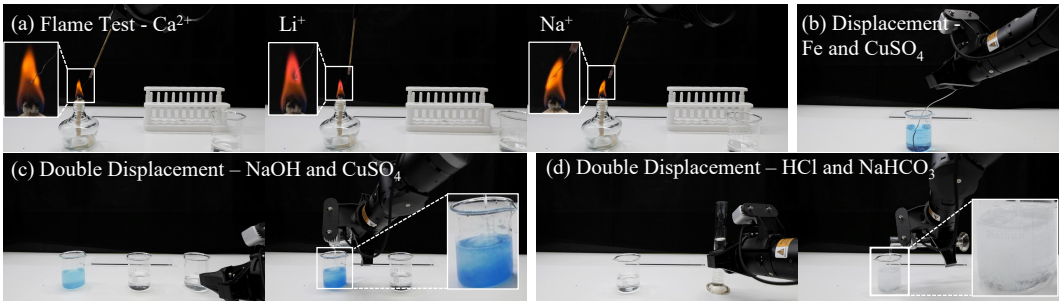

Figure 5: **Visualization of Generalization in Complete Tasks.** (a) Flame tests: Ca^{2+} (brick-red), Li^+ (purplish-red), and Na^+ (yellow); (b) Displacement reaction: Fe displaces Cu from $CuSO_4$ solution; (c) Double displacement reaction: NaOH and $CuSO_4$ form $Cu(OH)_2$ precipitate; (d) Double displacement reaction: HCl and $NaHCO_3$ generate CO_2 gas bubbles.

5 Conclusion

In this paper, we present *RoboChemist*, a dual-loop framework combining Vision-Language Models (VLMs) with Vision-Language-Action (VLA) models for safe and interpretable robotic chemical experiments. The VLM serves as a planner, visual prompt generator, and monitor—decomposing protocols, guiding VLA models, and verifying outcomes with corrective feedback. This closed-loop design handles hazardous and complex labware while ensuring safety and procedural compliance. Evaluations on primitive actions and multi-step tasks show *RoboChemist* surpasses prior methods in success and compliance. It generalizes well to new reagents, containers, and workflows, highlighting its potential to enhance lab automation with less human input and greater reliability.

6 Limitations

While *RoboChemist* demonstrates strong capabilities, several limitations remain. First, the gripper used for execution is not specifically designed for chemical tasks, limiting its ability to handle certain instruments; incorporating a more specialized robotic hand would expand the range of tasks that can be performed. Second, the system is currently unable to autonomously assemble chemical experimental setups, which require high precision and careful handling of various components. Tasks like connecting delicate glassware or positioning sensitive equipment demand more intricate manipulation capabilities that go beyond the current robotic system's design. Achieving this would require the integration of more advanced manipulation techniques and finer control over the robot's movements. Lastly, *RoboChemist*'s current architecture is not suited for tasks requiring strict quantification and precise time control. Future work includes integrating more capable hardware, extending the model to support fine-grained temporal and quantitative control, and testing in real-world laboratory settings.

References

[1] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, et al. A mobile robotic chemist. *Nature*, 583(7815):237–241, 2020.

[2] N. J. Szymanski, B. Rendy, Y. Fei, R. E. Kumar, T. He, D. Milsted, M. J. McDermott, M. Gallant, E. D. Cubuk, A. Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91, 2023.

[3] T. Dai, S. Vijayakrishnan, F. T. Szczypiński, J.-F. Ayme, E. Simaei, T. Fellowes, R. Clowes, L. Kotopanov, C. E. Shields, Z. Zhou, et al. Autonomous mobile robots for exploratory synthetic chemistry. *Nature*, pages 1–8, 2024.

[4] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[5] S. Steiner, J. Wolf, S. Glatzel, A. Andreou, J. M. Granda, G. Keenan, T. Hinkley, G. Aragon-Camarasa, P. J. Kitson, D. Angelone, et al. Organic synthesis in a modular robotic system driven by a chemical programming language. *Science*, 363(6423):eaav2211, 2019.

[6] S. H. M. Mehr, M. Craven, A. I. Leonov, G. Keenan, and L. Cronin. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512): 101–108, 2020.

[7] C. W. Coley, D. A. Thomas III, J. A. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, et al. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453):eaax1566, 2019.

[8] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[9] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[10] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[11] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, A. Wang, R. Fergus, Y. LeCun, and S. Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37: 87310–87356, 2024.

[12] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[13] X. Liu, B. Tian, Z. Wang, R. Wang, K. Sheng, B. Zhang, H. Zhao, and G. Zhou. Delving into shape-aware zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2999–3009, 2023.

[14] P. Li, B. Tian, Y. Shi, X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang. Toist: Task oriented instance segmentation transformer with noun-pronoun distillation. *Advances in Neural Information Processing Systems*, 35:17597–17611, 2022.

[15] B. Jin, Y. Zheng, P. Li, W. Li, Y. Zheng, S. Hu, X. Liu, J. Zhu, Z. Yan, H. Sun, et al. Tod3cap: Towards 3d dense captioning in outdoor scenes. In *European Conference on Computer Vision*, pages 367–384. Springer, 2024.

[16] H. Chi, H.-a. Gao, Z. Liu, J. Liu, C. Liu, J. Li, K. Yang, Y. Yu, Z. Wang, W. Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *arXiv preprint arXiv:2505.23757*, 2025.

[17] K. Ding, B. Chen, Y. Su, H.-a. Gao, B. Jin, C. Sima, W. Zhang, X. Li, P. Barsch, H. Li, et al. Hint-ad: Holistically aligned interpretability in end-to-end autonomous driving. *arXiv preprint arXiv:2409.06702*, 2024.

[18] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[19] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[20] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *International Conference on Learning Representations*, 2025.

[21] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Conference on Robot Learning*. PMLR, 2024.

[22] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[23] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[24] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[25] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[26] Y. R. Wang, Y. Zhao, H. Xu, S. Eppel, A. Aspuru-Guzik, F. Shkurti, and A. Garg. Mvtrans: Multi-view perception of transparent objects. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3771–3778. IEEE, 2023.

[27] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[28] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[29] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[30] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[31] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[32] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[33] Z. Zhang, X. Li, S. Zou, G. Chi, S. Li, X. Qiu, G. Wang, G. Zheng, L. Wang, H. Zhao, et al. Chameleon: Fast-slow neuro-symbolic lane topology extraction. *arXiv preprint arXiv:2503.07485*, 2025.

[34] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[35] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[36] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In *Robotics: Science and Systems*, New York City, USA, 2022.

[37] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu,

J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[38] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, Delft, Netherlands, 2024.

[39] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[40] AgiBot-World-Contributors, Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, S. Jiang, Y. Jiang, C. Jing, H. Li, J. Li, C. Liu, Y. Liu, Y. Lu, J. Luo, P. Luo, Y. Mu, Y. Niu, Y. Pan, J. Pang, Y. Qiao, G. Ren, C. Ruan, J. Shan, Y. Shen, C. Shi, M. Shi, M. Shi, C. Sima, J. Song, H. Wang, W. Wang, D. Wei, C. Xie, G. Xu, J. Yan, C. Yang, L. Yang, S. Yang, M. Yao, J. Zeng, C. Zhang, Q. Zhang, B. Zhao, C. Zhao, J. Zhao, and J. Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[41] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, et al. Vision-language foundation models as effective robot imitators. *International Conference on Learning Representations*, 2024.

[42] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3D world. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20413–20451. PMLR, 2024.

[43] Z. Durante, B. Sarkar, R. Gong, R. Taori, Y. Noda, P. Tang, E. Adeli, S. K. Lakshmikanth, K. Schulman, A. Milstein, et al. An interactive agent foundation model. *arXiv preprint arXiv:2402.05929*, 2024.

[44] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[45] J. Zheng, J. Li, D. Liu, Y. Zheng, Z. Wang, Z. Ou, Y. Liu, J. Liu, Y.-Q. Zhang, and X. Zhan. Universal actions for enhanced embodied foundation models. *arXiv preprint arXiv:2501.10105*, 2025.

[46] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

[47] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

[48] Z. Wu, Y. Zhou, X. Xu, Z. Wang, and H. Yan. Momanipvla: Transferring vision-language-action models for general mobile manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[49] A. Jiang, Y. Gao, Z. Sun, Y. Wang, J. Wang, J. Chai, Q. Cao, Y. Heng, H. Jiang, Y. Dong, et al. Diffvla: Vision-language guided diffusion planning for autonomous driving. *arXiv preprint arXiv:2505.19381*, 2025.

[50] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[51] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[52] K. Ding, B. Chen, R. Wu, Y. Li, Z. Zhang, H.-a. Gao, S. Li, G. Zhou, Y. Zhu, H. Dong, et al. Preafford: Universal affordance-based pre-grasping for diverse objects and environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7278–7285. IEEE, 2024.

[53] J. Liu, M. Liu, Z. Wang, P. An, X. Li, K. Zhou, S. Yang, R. Zhang, Y. Guo, and S. Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110, 2024.

[54] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3D-VLA: A 3D vision-language-action generative world model. In *Proceedings of the 41st International Conference on Machine Learning*, pages 61229–61245. PMLR, 2024.

[55] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[56] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[57] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[58] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022.

[59] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[60] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[61] S. Yoo, E. Kim, D. Jung, J. Lee, and S. Yoon. Improving visual prompt tuning for self-supervised vision transformers. In *International Conference on Machine Learning*, pages 40075–40092. PMLR, 2023.

[62] W. Liu, X. Shen, C.-M. Pun, and X. Cun. Explicit visual prompting for low-level structure segmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19434–19445, 2023.

[63] F. Li, Q. Jiang, H. Zhang, T. Ren, S. Liu, X. Zou, H. Xu, H. Li, J. Yang, C. Li, et al. Visual in-context prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12871, 2024.

[64] M. Cai, H. Liu, S. K. Mustikovela, G. P. Meyer, Y. Chai, D. Park, and Y. J. Lee. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923, 2024.

[65] C. Xu, Y. Zhu, H. Shen, B. Chen, Y. Liao, X. Chen, and L. Wang. Progressive visual prompt learning with contrastive feature re-formation. *International Journal of Computer Vision*, 133 (2):511–526, 2025.

[66] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[67] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[68] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[69] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[70] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003.

[71] K. Krishna and M. N. Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

[72] Z. Liu, M. Zhang, and Y. Li. Kuda: Keypoints to unify dynamics learning and visual prompting for open-vocabulary robotic manipulation. *arXiv preprint arXiv:2503.10546*, 2025.

[73] K. Fang, F. Liu, P. Abbeel, and S. Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. *Robotics: Science and Systems (RSS)*, 2024.

[74] Y. Harazono, H. Shimono, K. Hata, T. Mitsuyama, and T. Horinouchi. Evaluation of microplate handling accuracy for applying robotic arms in laboratory automation. *SLAS technology*, 29 (6):100200, 2024.

[75] N. Yoshikawa, A. Z. Li, K. Darvish, Y. Zhao, H. Xu, A. Kuramshin, A. Aspuru-Guzik, A. Garg, and F. Shkurti. Chemistry lab automation via constrained task and motion planning. *arXiv preprint arXiv:2212.09672*, 2022.

[76] K. Darvish, M. Skreta, Y. Zhao, N. Yoshikawa, S. Som, M. Bogdanovic, Y. Cao, H. Hao, H. Xu, A. Aspuru-Guzik, et al. Organa: a robotic assistant for automated chemistry experimentation and characterization. *Matter*, 8(2), 2025.

[77] H. Fakhruldeen, G. Pizzuto, J. Glowacki, and A. I. Cooper. Archemist: Autonomous robotic chemistry system architecture. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6013–6019. IEEE, 2022.

[78] D. Knobbe, H. Zwirnmann, M. Eckhoff, and S. Haddadin. Core processes in intelligent robotic lab assistants: Flexible liquid handling. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2335–2342. IEEE, 2022.

[79] D. Schober, R. Güldenring, J. Love, and L. Nalpantidis. Vision-based robot manipulation of transparent liquid containers in a laboratory setting. In *2025 IEEE/SICE International Symposium on System Integration (SII)*, pages 1193–1200. IEEE, 2025.

[80] S. Li, Y. Huang, C. Guo, T. Wu, J. Zhang, L. Zhang, and W. Ding. Chemistry3d: Robotic interaction benchmark for chemistry experiments. *arXiv preprint arXiv:2406.08160*, 2024.

[81] W. Zhao, P. Ding, Z. Min, Z. Gong, S. Bai, H. Zhao, and D. Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. In *The Thirteenth International Conference on Learning Representations*.

[82] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[83] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[84] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022.

RoboChemist: Long-Horizon and Safety-Compliant Robotic Chemical Experimentation

A Appendix

In this appendix, we provide comprehensive supplementary materials and detailed technical insights to support and extend the results presented in the main paper. Sec. A.1 details the primitive tasks, including specific compliance criteria and evaluation metrics used to measure success rate and compliance rate. Sec. A.2 describes the five complete chemical experiments used in our evaluation,elaborating on chemical principles, experimental protocols, and observation criteria for multi-step tasks. Sec. A.3 lists key prompts utilized for task decomposition, safety guideline generation, visual prompting, and task completion verification. Sec. A.4 presents additional evaluation demonstrating RoboChemist's capability for generalization across primitive tasks and complete chemical procedures. Sec. A.5 compares various visual prompting strategies to assess their effectiveness in guiding robotic manipulations. Sec. A.6 quantitatively evaluates the impact of training data configurations with varied proportions of successful trials and second-attempt exploratory trials on the robustness and self-correcting capabilities of the inner loop. Sec. A.7 describes the specific VLA architecture employed in our work. Sec. A.8 provides an error decomposition across five modules. Finally, Sec. A.9 evaluates the generalization of one task under diverse variations, showing that VLM feedback and visual prompting enable robust transfer.

A.1 Details of Primitive Tasks, Compliance Criteria, and Evaluation Protocols

This section provides a detailed description of the seven primitive tasks performed by *RoboChemist*, including the task objectives and the compliance criteria used for evaluation. The performance of each primitive task is evaluated using two metrics: Success Rate (SR) and Compliance Rate (CR), where SR represents the ratio of successful trials to total trials, and CR measures how well the task adheres to predefined experimental norms.

1. **Grasping a Glass Rod:**
   - Objective: Grasp a glass rod successfully.
   - Compliance Criteria:
     - Grasp fails (does not grasp the rod): 0
     - Grasp is made but not at the correct position (e.g., not at the 1/3 point of the rod): 0.5
     - Correct grasp at the 1/3 position of the rod: 1
2. **Heating Platinum Wire:**
   - Objective: Heat a platinum wire in a flame to a specific region.
   - Compliance Criteria:
     - Does not heat: 0
     - Heats at the wrong location and does not return: 0.25
     - Heats at the wrong location but returns: 0.5
     - Heats at the correct location but does not return: 0.75
     - Heats at the correct location and returns: 1
3. **Inserting Platinum Wire into Solution:**
   - Objective: Insert a platinum wire into a solution.
   - Compliance Criteria:
     - Does not reach the top of the beaker: 0
     - Reaches the beaker but does not insert into the solution: 0.5
     - Correctly inserts into the solution: 1

Figure 6: Visualization of primitive tasks.

4. **Pouring Liquid:**
   - Objective: Pour a liquid from one container into another.
   - Compliance Criteria:
     - Fails to grasp: 0
     - Grasped but spills completely: 0.25
     - Grasped but spills slightly: 0.75
     - Successfully grasps and pours liquid with control: 1
5. **Stirring Solution with a Glass Rod:**
   - Objective: Stir a solution to ensure proper mixing.
   - Compliance Criteria:
     - Does not contact liquid: 0
     - Stirred but uneven, hitting the container wall: 0.5
     - Correct stirring with adequate speed and uniform direction: 1
6. **Transferring Solid:**
   - Objective: Transfer a solid from one container to another.
   - Compliance Criteria:
     - Does not transfer solid: 0
     - Solid transferred but spills out: 0.25
     - Solid transferred with slight spillage: 0.75
     - Solid transferred without spillage: 1
7. **Pressing a Button:**

- Objective: Press a button to initiate a process.
- Compliance Criteria:
  - Does not press: 0
  - Pressed but not activated: 0.25
  - Pressed but too forcefully, causing movement of the equipment: 0.5
  - Correct press with appropriate force: 1

Based on this, the Success Rate (SR) is calculated by performing the task 20 times and dividing the number of successful trials by the total number of trials. Specifically, the Success Rate is given by:

$$SR = \frac{\text{Number of successful trials}}{\text{Total number of trials}}.$$

The Compliance Rate (CR) is the average score of the 20 trials, where each trial is assigned a score based on the compliance with the task criteria. For example, in the "*Grasping a Glass Rod*" task, suppose there are n_1 trials where the score is 0, n_2 trials with a score of 0.5, and n_3 trials with a score of 1. Then the Compliance Rate is calculated as:

$$CR = \frac{(0 \times n_1) + (0.5 \times n_2) + (1 \times n_3)}{n_1 + n_2 + n_3}.$$

In this example, $n_1 + n_2 + n_3 = 20$. The CR is the average of the scores obtained in these 20 trials, representing how well the actions align with the experimental norms.

This calculation method is used to derive the data presented in Table 2 for the primitive tasks, where both Success Rate and Compliance Rate are computed for each task. **Note:** The last two rows in Table 2 correspond to different variants of *RoboChemist*. *RoboChemist w/o CL* represents the version without the outer closed-loop; that is, the VLM is not used to assess whether each primitive task is successfully completed, while the visual prompting module is still included. In contrast, *RoboChemist w/ CL* denotes the complete system. In this configuration, the VLM acts as a monitor: after each primitive execution, it evaluates the current scene and determines whether the task has succeeded. If not, it instructs the VLA model to reattempt the primitive until the task is deemed successful, forming a full closed-loop feedback mechanism that enhances both robustness and compliance. By comparing the performance of these two variants, we observe the significant role of the closed-loop system in improving the success and compliance of primitive task execution.

A.2 Details of Complete Chemical Tasks and Evaluation Protocols

This section provides detailed descriptions of the five complete chemical experiment tasks used in our evaluation. Each task is composed of 1 to 5 primitive tasks arranged sequentially, and is designed to cover a diverse range of chemical operations and reaction types. For each task, we describe the objective, underlying reaction principle, expected observable phenomena, and the language prompts used to initiate the sequence.

1. Mixing NaCl and $CuSO_4$ Solutions

- **Objective:** Mix sodium chloride solution with copper sulfate solution and observe the resulting color change.
- **Primitive tasks:** Pour one beaker of liquid into another.
- **Observation and Explanation** This is a coordination reaction in which hydrated copper ions, initially blue in color, react with chloride ions to form the complex ion $[CuCl_4]^{2-}$. The reaction proceeds as:

  $$[Cu(H_2O)_4]^{2+} + 4Cl^- \rightarrow [CuCl_4]^{2-} + 4H_2O$$

  As the reaction occurs, the solution color transitions from blue to cyan and then to green. The final green hue corresponds to the formation of sodium tetrachlorocuprate, $Na_2[CuCl_4]$, indicating successful complexation (as Figure 7 shows).

Figure 7: Visualization of mixing NaCl and $CuSO_4$ solutions.

2. Thermal Decomposition of $Cu(OH)_2$

- **Objective:** Heat a test tube containing solid $Cu(OH)_2$ and observe the resulting decomposition process.
- **Primitive tasks:** Grasp the test tube containing $Cu(OH)_2$ $\rightarrow$ Heat over flame.
- **Explanation and Observation:** Copper(II) hydroxide ($Cu(OH)_2$) is a pale blue solid. Upon heating, it decomposes into copper(II) oxide (CuO), which is black, and water vapor. During the reaction, mist forms on the inner wall of the test tube due to condensation of steam. The chemical equation is:

$$Cu(OH)_2 \xrightarrow{\Delta} CuO + H_2O$$

This reaction reflects the thermal instability of $Cu(OH)_2$ and the stability of CuO at elevated temperatures (as Figure 8 shows).

Figure 8: Visualization of thermal decomposition of $Cu(OH)_2$.

3. Flame Test of $CuSO_4$ Solution

- **Objective:** Identify the presence of Cu^{2+} ions in copper sulfate through a flame test, investigating its physical property via characteristic flame emission.
- **Primitive tasks:** Grasp platinum wire $\rightarrow$ Dip into $CuSO_4$ solution $\rightarrow$ Heat platinum wire in flame.
- **Observation and Explanation:** When a compound containing Cu^{2+} ions (e.g., $CuSO_4$) is heated in a flame, it emits a characteristic blue-green color. This flame color originates from the emission spectrum of Cu^{2+} ions, which release photons primarily in the 450–500 nm wavelength range when thermally excited. The resulting blue-green light is a reliable indicator of the presence of copper ions and is commonly used in qualitative elemental analysis (as Figure 9 shows).

Figure 9: Visualization of flame test of $CuSO_4$ solution.

4. Evaporation of NaCl Solution

- **Objective:** Evaporate an impure NaCl solution to separate soluble salt from insoluble impurities.
- **Primitive tasks:** Transfer solid NaCl into a beaker of water $\rightarrow$ Press the heater button to initiate evaporation.

- **Explanation and Observation:** The primary component of the crude salt is sodium chloride (NaCl), which dissolves readily in water to form a solution, while insoluble impurities such as sand or other salts remain at the bottom. Upon pressing the heater button, the solution begins to heat and eventually boils, producing visible bubbles. As heating continues, the water gradually evaporates, increasing the salt concentration. Once the solution reaches saturation, solid NaCl crystals begin to precipitate. When all water has evaporated, white NaCl crystals remain in the beaker, typically hard in texture and free from soluble contaminants (as Figure 10 shows).

Figure 10: Visualization of evaporation of NaCl solution.

5. Acid-Base Neutralization with Phenolphthalein Indicator

- **Objective:** Neutralize a sodium hydroxide (NaOH) solution by gradually adding hydrochloric acid (HCl) until the solution reaches neutrality, as indicated by phenolphthalein.
- **Primitive tasks:** Transfer solid NaOH into a beaker of water → Grasp a glass rod → Stir the NaOH solution to dissolve the solid → Add phenolphthalein indicator → Pour HCl into the NaOH solution until the solution becomes colorless.
- **Explanation and Observation:** Sodium hydroxide is a strong base and dissolves in water to form a colorless alkaline solution. Phenolphthalein is added as an acid-base indicator, turning the solution red in basic conditions. As hydrochloric acid is gradually added, H^+ ions react with OH^- ions to form water, reducing the pH of the solution. When neutrality is achieved, the pink color disappears and the solution becomes colorless (as Figure 11 shows). This indicates the end point of the acid-base neutralization reaction:

$$\mathrm{HCl} + \mathrm{NaOH} \rightarrow \mathrm{NaCl} + \mathrm{H_2O}$$

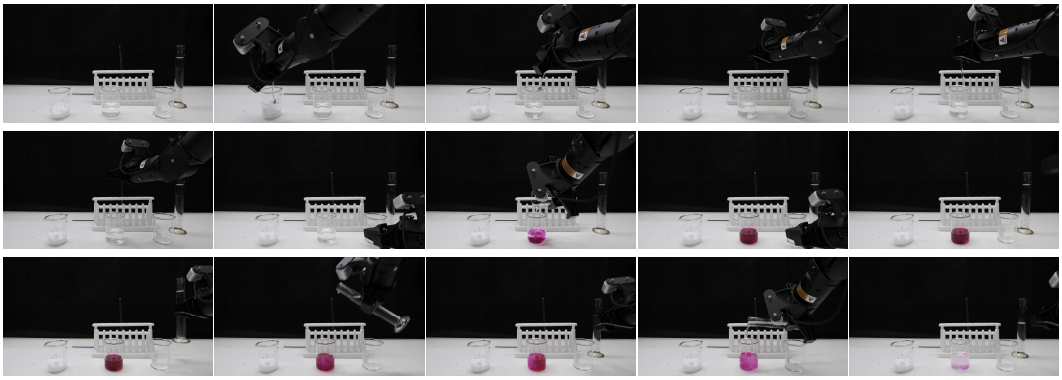

Figure 11: Visualization of acid-base neutralization with phenolphthalein indicator.

Each experiment is prompted through natural language instructions, and *RoboChemist* decomposes the tasks and monitors the progress via its dual-loop system. To quantitatively evaluate the execution of complete tasks, we follow a stepwise evaluation protocol aligned with the sequential structure of chemical experiments. As shown in Table 3, each complete task is composed of a series of primitive tasks. We conduct 20 independent trials for each complete task. If any primitive task within the sequence fails, all subsequent steps are considered unsuccessful for that trial. Therefore, the success rate (SR) for a complete task is computed as the proportion of trials in which all constituent primitive tasks are successfully completed. Similarly, the compliance rate (CR) for each primitive step is

averaged only over the trials where that specific step was reached. Consequently, the final step's SR in each row also reflects the overall SR of the complete task, serving as a comprehensive indicator of the whole task's success.

A.3 Key Prompts

A.3.1 Task Decomposition

We present examples of our proposed prompt for task decomposition in each chemical experiment. For different chemical experiments, only the apparatus and task need to be modified.

1. Mixing NaCl and $CuSO_4$ Solutions

You are a lab assistant tasked with mixing NaCl and $CuSO_4$ solutions to form sodium tetrachlorocuprate using the materials shown in the image. The items, from left to right, are:

- A beaker with sodium chloride solution.
- A beaker with copper sulfate solution.

Task: Mixing NaCl and $CuSO_4$ solutions to form sodium tetrachlorocuprate.

Goal: Provide a step-by-step list of primitives (simple lab actions) required to carry out this reaction safely and correctly using the available materials.

Output Format: Directly return a list of primitive actions only containing the following steps without any explanations. If given until [condition] sentence, the robot will repeat the operation until the condition is achieved:

- Grasp [rod-like object]
- Pour [liquid] from [container] into [container] until [condition]
- Stir [mixture]
- Transfer [solid] from [container] to [container]
- Dip [object] into the [solution] in [container]
- Heat [object] over a flame
- Press the button of [object]

2. Thermal Decomposition of $Cu(OH)_2$

You are a lab assistant tasked with performing the thermal decomposition of copper(II) hydroxide using the materials shown in the image. The items, from left to right, are:

- A lit alcohol lamp.
- A test tube containing copper(II) hydroxide.

Task: Performing the thermal decomposition of copper(II) hydroxide.

...

3. Flame Test of $CuSO_4$ Solution

```
You are a lab assistant tasked with performing the flame test of
copper(II) hydroxide using the materials shown in the image.  The
items, from left to right, are:
     • A lit alcohol lamp.
     • Platinum wire.
     • A test tube containing copper(II) hydroxide.

Task:  Performing the flame test of copper(II) hydroxide.

...
```

4. Evaporation of NaCl Solution

```
You are a lab assistant tasked with evaporating an impure NaCl
solution to separate soluble salt from insoluble impurities using
the materials shown in the image.  The items, from left to right,
are:
     • An evaporator with a power button.
     • A breaker with NaOH solid.

Task:  Evaporating an impure NaCl solution to separate soluble salt
from insoluble impurities

...
```

5. Acid-Base Neutralization with Phenolphthalein Indicator

```
You are a lab assistant tasked with performing an acid-base neutrali-
zation reaction using the materials shown in the image.  The items,
from left to right, are:
     • A beaker with NaOH solid.
     • A beaker with water and a glass rod.
     • A beaker with phenolphthalein indicator.
     • A graduated cylinder containing hydrochloric acid (HCl).
     • A glass rod in a test tube rack.

Task:  Performing an acid-base neutralization reaction.

...
```

A.3.2 Safety Guideline Generation

We present examples of our proposed prompt for safety guideline generation prior to performing primitive tasks. For different primitive tasks, only the name of each task needs to be modified.

```
I am doing experiment using robot arms.  Regarding to step [NUMBER],
[PRIMITIVE], establish safety and standardization guidelines
necessary for conducting the chemistry experiment for robot arms,
such as where are the items related to this step, where grasp points
of robot arm are, etc.  These guidelines should be only related
```

```
to items mentioned before.  Summarize these in one or two clear
sentences.  If such guidelines are not applicable or available,
return None.
```

A.3.3 Visual Prompt Generation

We present examples of our proposed prompt for visual prompt generation prior to performing primitive tasks. For different primitive tasks, only the name of each task needs to be modified.

```
Now, given the image of current state of experiment, using marked
points/bounding box to create visual prompts for the image in
order to guide the robot to finish step [NUMBER], [PRIMITIVE].
Remember that only mark the points/bounding box that robot can
currently see/operate in the current state and do not predict future
things/items.  Return a list of point/bounding box coordinates in
the image of current state in the format of a list, like ["type":
"box", "coordinates":[xmin, ymin, xmax, ymax], "type":"point",
"coordinates":[x, y]].  If there is nothing to return, return an
empty list.
```

A.3.4 Task Completion Verification

We present examples of our proposed prompt for task completion verification after performing primitive tasks. For different primitive tasks, only the name of each task needs to be modified.

```
Judge whether the step [NUMBER], [PRIMITIVE], has successfully
finished from the image provided.  If yes, directly return Y, else
return N.
```

A.4 Extended Analysis of Generalization

Primitive Task Generalization. To further validate the generalization capability of *RoboChemist* at the primitive task level, we design and evaluate a set of novel tasks that are not seen during training but are compositionally similar to the original primitives (as shown in Figure 12). These tasks introduce variations in object geometry and physical interaction while preserving the core manipulation intent. We conduct 20 trials for each task, using the same evaluation criteria as in the main paper—Success Rate (SR) and Compliance Rate (CR). As shown in Table 5, *RoboChemist* demonstrates strong transferability, successfully applying learned manipulation policies to novel scenarios with consistent safety and execution quality.

Figure 12: Visualization of primitive task generalization.

Training Task	Grasp Glass Rod				Stir the Solution		Heat Platinum Wire		Insert Platinum Wire into Solution			
Generalized Task	Place Glass Rod		Grasp Test Tube		Stir Solid Reagents		Heat Test Tube		Insert a Thermometer		Place Test Tube into Cooling Liquid	
Method	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR ↑	SR(%) ↑	CR
ACT [82]	10	0.050	40	0.250	15	0.075	5	0.025	10	0.025	10	0.050
RDT [20]	0	0.000	10	0.050	60	0.350	35	0.175	75	0.500	60	0.300
π_0 [21]	35	0.200	40	0.250	85	0.450	55	0.425	80	0.550	65	0.450
RoboChemist	**55**	**0.400**	**90**	**0.850**	**100**	**0.785**	**70**	**0.675**	**95**	**0.950**	**85**	**0.850**

Table 6: Primitive task generalization results.

Complete Task Generalization. Beyond the generalization of individual primitives, *RoboChemist* shows the ability to generalize to novel complete chemistry tasks composed of sequences of primitive actions. This is enabled by the VLM's capacity to reason over unseen goals and plan appropriate action sequences using its chemical knowledge. In particular, the system successfully performs complete experiments corresponding to the four fundamental types of chemical reactions: combination, decomposition, displacement, and double displacement.

(a) Combination Reaction: *RoboChemist* pours water into a container of calcium oxide (CaO), initiating a vigorous exothermic reaction that forms calcium hydroxide:

$$\mathrm{CaO} + \mathrm{H_2O} \rightarrow \mathrm{Ca(OH)_2}$$

During execution, the calcium oxide rapidly reacts with water, releasing a large amount of heat. This thermal energy generates a hissing sound and partial vaporization of the liquid. The reaction produces a white solid—calcium hydroxide—which partially dissolves, forming a milky suspension. This example demonstrates the system's ability to manage high-temperature reactions and recognize key visual and auditory indicators of successful chemical transformation (as Figure 13 shows).

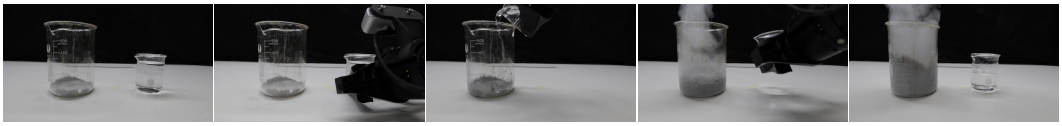

Figure 13: Visualization of combination reaction.

(b) Decomposition Reaction: *RoboChemist* performs a catalytic decomposition of hydrogen peroxide ($\mathrm{H_2O_2}$) by introducing Manganese(II) hydroxide ($\mathrm{Mn(OH)_2}$) as a catalyst. The reaction proceeds as:

$$2\mathrm{H_2O_2} \xrightarrow{\mathrm{Mn(OH)_2}} 2\mathrm{H_2O} + \mathrm{O_2} \uparrow$$

Upon the addition of $\mathrm{Mn(OH)_2}$, *RoboChemist* observes a rapid generation of oxygen gas, visible as dense bubbling at the liquid surface. This demonstrates the system's ability to manage reactive liquid mixing, recognize real-time gas evolution, and safely interpret catalyst-driven decomposition reactions (as Figure 14 shows).

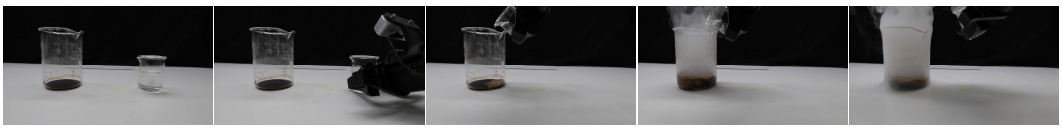

Figure 14: Visualization of decomposition reaction.

(c) Displacement Reaction: As Figure 5(b) shows, *RoboChemist* inserts an iron wire into a $\mathrm{CuSO_4}$ solution. The reaction proceeds as:

$$\mathrm{Fe} + \mathrm{CuSO_4} \rightarrow \mathrm{FeSO_4} + \mathrm{Cu}$$

This causes reddish solid copper to precipitate on the iron surface, demonstrating iron's higher reactivity. Similarly, RoboChemist performs acid-metal reactions such as:

$$\mathrm{Zn} + 2\mathrm{HCl} \rightarrow \mathrm{ZnCl_2} + \mathrm{H_2} \uparrow$$

where bubbles form as hydrogen gas evolves and zinc dissolves, showcasing its ability to execute and interpret multiple forms of displacement processes (as Figure 15 shows).

(d) Double Displacement Reaction: As Figure 5(c)-(d) shows, *RoboChemist* executes precipitation and gas-generation reactions through compound exchange. For example:

$$2\mathrm{NaOH} + \mathrm{CuSO_4} \rightarrow \mathrm{Cu(OH)_2} \downarrow + \mathrm{Na_2SO_4}$$

forming a distinct blue precipitate of $\mathrm{Cu(OH)_2}$. Another example includes:

$$\mathrm{NaHCO_3} + \mathrm{HCl} \rightarrow \mathrm{NaCl} + \mathrm{H_2O} + \mathrm{CO_2} \uparrow$$

Figure 15: Visualization of displacement reaction between Zn and HCl.

where CO_2 gas is produced, visible as vigorous bubbling, highlighting RoboChemist's capacity to identify evolving gases and reaction completion.

In addition, *RoboChemist* can also distinguish between different metal ions based on flame color, revealing generalization to physical property inference. As shown in Figure 5(a), RoboChemist performs flame tests by dipping a platinum wire into various solutions and observing the resulting flame color when heated. These colors serve as indicators of metal identity, including Ca^{2+} (brick-red), Li^{+} (purplish-red), Na^{+} (yellow), Mn^{2+} (yellow-green), and Sr^{2+} (magenta).

This demonstrates that once *RoboChemist* masters a representative instance of a given experimental category—such as a flame test—it can generalize this capability to a broader class of similar procedures. Such compositional generalization underscores the system's ability to integrate procedural knowledge (via VLM) and low-level execution (via VLA) to generalize from basic actions to diverse, multi-step chemistry tasks. The system not only completes complex workflows but also interprets and responds to observed phenomena, showcasing its robust reasoning and adaptability. Please refer to the project page for qualitative demonstrations of generalization performance.

A.5 Comparison of Visual Prompting Methods

This section provides detailed descriptions of the visual prompting strategies used in our comparison experiments (as shown in Table 4), specifically focusing on ReKep [25], MOKA [73], and our proposed VLM-based approach. All three methods generate visual prompts that are used as auxiliary supervision for fine-tuning a π_0 [21] policy model. This standardized setup enables a fair evaluation of the prompt generation quality across different frameworks.

In **ReKep**, visual prompts are created in the form of Relational Keypoint Constraints (ReKep). These constraints encode desired spatial relations between semantically meaningful 3D keypoints extracted from RGB-D input. ReKep first uses Large Vision Models (e.g., DINOv2) to generate dense keypoint candidates, then employs a Vision-Language Model (e.g., GPT-4o) to generate Python functions expressing constraints over these keypoints. For example, the constraint might require one keypoint (e.g., spout) to align vertically above another (e.g., cup center), or define a 3D vector between two points to encode a specific rotation. Although ReKep can flexibly describe complex spatio-temporal tasks, its performance suffers in scenes involving transparent or textureless objects, where depth-based keypoint proposals become unreliable. Additionally, since the constraints are encoded as code, they must still be transformed into explicit labeled examples (e.g., target positions) to be used for training π_0.

MOKA, on the other hand, formulates visual prompting as a mark-based problem. It uses open-vocabulary detectors (e.g., Grounding DINO) and segmentation models (e.g., SAM) to locate objects of interest, and then leverages Vision-Language Models (e.g., GPT-4V) to generate mark sets on RGB images — each mark corresponding to a 2D location (or region) semantically tied to the instruction. These marked images are treated as visual prompts to supervise a visuomotor agent. Importantly, MOKA avoids explicit 3D reconstruction and instead operates directly in 2D pixel space. While this allows for flexibility and ease of deployment, it lacks fine-grained geometric reasoning and results in lower task compliance, especially for cases requiring precise spatial alignment or multi-step reasoning.

In contrast, our method uses a vision-language model to directly annotate 2D visual prompts by pointing to semantically and geometrically relevant locations on input images. These annotated keypoints are automatically derived from the instruction-image pair using a VLM with built-in

grounding ability. Unlike ReKep or MOKA, our approach does not rely on RGB-D input, depth reconstruction, or open-vocabulary object detectors, and produces supervision that is both semantically aligned and geometrically grounded. We then use these VLM-derived visual prompts as reference targets to fine-tune a VLA model, which can then serve as a policy for π_0 training.

Quantitative results in Table 4 highlight that ReKep achieves limited success due to the difficulty of keypoint localization in challenging scenes (e.g., transparent objects), and MOKA exhibits moderate compliance as it lacks explicit spatial reasoning. In contrast, *RoboChemist*, powered by VLM-based 2D prompting, consistently outperforms these methods, demonstrating the advantage of our direct, image-based visual prompting pipeline.

A.6 Inner Loop Enhancement: Implementation and Evaluation

To investigate the impact of mixed success data on the performance of our VLA model, we conduct a controlled study using four distinct training configurations for each primitive task. Each configuration consists of a total of 400 training samples for a given task, but the proportion of successful trials and those with a second attempt after an initial failure varies:

- **Config 1 (400/0 - All Successful):** 400 successful trials with no failed attempts.
- **Config 2 (300/100 - Majority Successful):** 300 successful trials and 100 trials where a second attempt is made after an initial failure.
- **Config 3 (200/200 - Balanced):** 200 successful trials and 200 trials where a second attempt is made after an initial failure.
- **Config 4 (0/400 - All Second Attempt):** 400 trials where a second attempt is made after an initial failure.

Primitive Task	Grasp Glass Rod	Heat Platinum Wire	Insert into Solution	Pour Liquid	Stir the Solution	Transfer the Solid	Press the Button	Average
Config 1	**40**	50	75	**85**	**85**	**80**	55	67.14
Config 2	**40**	**55**	**80**	80	**85**	**80**	70	**70**
Config 3	35	**55**	70	65	65	70	**75**	62.14
Config 4	25	20	15	5	30	15	45	22.14

Table 7: Impact of Different Training Configurations on Primitive Task Success Rates.

We fine-tune the π_0 model using each of these four data configurations and quantitatively evaluate their impact on the success rate of the seven primitive tasks. Table 7 presents the success rates for each task under the four configurations. The results demonstrate that Config 2 (300/100) achieves the highest average success rate (70%), indicating the optimal balance between successful and exploratory trials. Config 1 (400/0) follows closely, while Config 3 (200/200) and Config 4 (0/400) exhibit significantly lower performance, suggesting that too many exploratory trials degrade the model's effectiveness.

A.7 Details of Used VLA Architecture

We utilized π_0 as the backbone of the VLA Architecture. Here, we present more details about π_0 design.

π_0 builds on a pre-trained vision–language model (VLM) backbone (specifically, PaliGemma), inheriting its rich semantic and visual reasoning capabilities. Image observations from the robot's cameras are encoded into embeddings that share the same space as language tokens. To turn the VLM into a controller, π_0 augments this backbone with two robotics-specific components: (1) proprioceptive state inputs (e.g., joint angles) and (2) continuous action outputs. Both are projected into the transformer's token sequence. Crucially, π_0 uses a separate "action expert" — a distinct set of transformer weights — to process these robotics tokens. Rather than discretizing actions, π_0 employs conditional flow matching to model the distribution of continuous action chunks. At each timestep, the model predicts a short sequence (chunk) of future actions by learning a denoising vector field that is integrated via a fixed-step Euler solver at inference. This diffusion-style approach

gives π_0 high precision and the ability to represent complex, multimodal action distributions needed for dexterous tasks.

π_0 is trained on data from multiple robot platforms (single-arm, dual-arm, mobile bases), each with different action/state dimensions. To accommodate this diversity, all configuration and action vectors are zero-padded to a common size, and missing camera slots are masked. This cross-embodiment strategy lets a single π_0 model generalize across varied hardware setups without per-robot specialization.

Regarding *RoboChemist*, we introduce an image containing a visual prompt, which consists of annotated points or bounding boxes indicating keypoints and objects. These visual prompts are generated based on the objectives of primitive chemical experiments and corresponding safety guidance. Under the guidance of these visual prompts, the robot is able to perform experimental tasks with greater accuracy and enhanced safety. Specifically, we directly annotate the visual prompts on the RGB image and include them as an additional input channel to the image encoder in the VLA model. We also explicitly indicate in the instruction whether a given image includes a visual prompt. Furthermore, to ensure instruction diversity, we employ GPT-4o to generate multiple variations of natural language instructions for each primitive task, as detailed below.

1. **Grasping a Glass Rod:**

```
1) "In the image input, the last image is used as a reference
image, with the [COLOR] target point being the location
where the robotic gripper grasps the glass rod.  The [COLOR]
bounding box surrounds the glass rod, indicating the region of
interest.  Using the right arm of the robotic arm, carefully
grasp the glass rod and lift it gently."
2) "In the last image of the input sequence, the [COLOR]
target point indicates the designated grasp location on the
glass rod.  The [COLOR] bounding box encloses the glass rod,
marking the region to focus on.  Using the right manipulator,
precisely approach and grasp the rod at the indicated point,
ensuring a secure and stable hold."
3) "Refer to the last image provided, in which the [COLOR]
target point specifies the grasp location on the glass rod.
The [COLOR] bounding box highlights the glass rod's region of
interest.  The right robotic arm should be used to perform a
precise and stable grasp at the indicated position."
4) "As shown in the final image input, the [COLOR] point
represents the target location for grasping the glass rod.
The [COLOR] bounding box defines the region of the glass rod.
Utilize the right arm of the robot to perform a careful and
firm grasp at this location, ensuring the rod is securely
held."
5) "The last image in the input sequence provides the
reference for grasping, with the [COLOR] point indicating
the target position on the glass rod.  The [COLOR] bounding
box clearly identifies the region of the glass rod.  The task
is to control the robot's right arm to grasp the rod at this
point and maintain a stable grip."
```

2. **Heating Platinum Wire:**

```
1) "In the image input, the last image is used as a reference
image, with the [COLOR] target point for the platinum wire
head to extend into the alcohol burner flame.  Using the
```

```
right robotic arm, hold the platinum wire and carefully extend
it into the outer flame of the Bunsen burner until it glows
red-hot."
2) "The final image in the input serves as a reference,
with the [COLOR] marker specifying the target location for
introducing the platinum wire tip into the Bunsen burner
flame.  The right robotic manipulator is used to securely
hold the wire and extend it into the outer flame region until
red-hot."
3) "Refer to the last input image, where the [COLOR] target
point marks the location for inserting the platinum wire tip
into the flame of the Bunsen burner.  The robot's right arm
should be used to hold the wire and steadily guide it into the
outer flame until visible incandescence is achieved."
4) "In the last image provided, the [COLOR] point indicates
the desired position for extending the platinum wire tip into
the Bunsen burner flame.  The right robotic arm is tasked with
holding the wire and positioning it within the outer flame
zone until it becomes red-hot."
5) "The [COLOR] marker in the final input image denotes the
target region for positioning the platinum wire head within
the Bunsen burner flame.  The right arm of the robot is
employed to grasp and insert the wire into the outer flame
carefully, heating it until it glows red."
```

3. **Inserting Platinum Wire into Solution:**

```
1) "In the last image, the [COLOR] bounding box surrounds the
beaker and the [COLOR] target point marks the liquid level
inside it.  Using the right robotic arm, carefully grasp the
platinum wire and gently extend it into the beaker to dip it
into the liquid."
2) "The last image in the input sequence serves as a
reference, where the [COLOR] bounding box outlines the beaker
and the [COLOR] marker denotes the target liquid level inside
it.  The right robotic arm is used to securely hold the
platinum wire and gently insert it into the liquid up to the
specified depth."
3) "As shown in the final input image, the [COLOR] bounding
box highlights the beaker and the [COLOR] target point
indicates the liquid surface level.  The robot's right
manipulator is employed to grasp the platinum wire and immerse
it into the liquid to the designated level."
4) "Refer to the last image in the input, where the [COLOR]
bounding box encloses the beaker and the [COLOR] target point
represents the desired immersion depth corresponding to the
liquid level.  The platinum wire is held by the right robotic
arm and is carefully dipped into the liquid accordingly."
5) "In the final image of the input, the [COLOR] bounding box
frames the beaker and the [COLOR] point indicates the liquid
level to which the platinum wire should be submerged.  The
right robotic arm is used to delicately lower the wire into
the beaker until the required depth is reached."
```

4. **Pouring Liquid:**

```
1) "In the last image, the [COLOR] bounding box around the
left beaker and the [COLOR] bounding box around the right
beaker are shown, each containing its respective [COLOR] grasp
point (one on the left beaker, one on the right).  The [COLOR]
point on the right beaker indicates the position for the
robotic arm to grasp the beaker.  With the left arm holding
the left beaker (at the [COLOR] left-beaker point inside its
[COLOR] bounding box) and the right arm grasping the right
beaker (at the [COLOR] right-beaker point inside its [COLOR]
bounding box), carefully pour the liquid from the right beaker
into the left until fully transferred."
2) "The last image serves as a reference, showing the [COLOR]
bounding box around the left beaker, the [COLOR] bounding box
around the right beaker, and their corresponding [COLOR] grasp
points.  The [COLOR] marker on the right beaker denotes the
designated grasp location.  The robotic system uses its left
arm to stabilize the left beaker (holding it at the [COLOR]
left-beaker point) while the right arm lifts and tilts the
right beaker (grasped at the [COLOR] right-beaker point) to
transfer the liquid completely into the left one."
3) "In the final image, you can see the [COLOR] bounding box
around the left beaker and the [COLOR] bounding box around the
right beaker, each highlighting a [COLOR] grasp point.  The
[COLOR] point on the right beaker indicates where to grasp.
The robot uses its left manipulator to hold the left beaker
steady (at its [COLOR] point) while the right manipulator
grasps the right beaker (at the [COLOR] point) and pours its
contents into the left until the transfer is complete."
4) "Refer to the last image, which shows a [COLOR] bounding
box around the left beaker and a [COLOR] bounding box around
the right beaker, each with an associated [COLOR] point.
The [COLOR] marker on the right beaker identifies the
designated grasp position.  The dual-arm system coordinates
both manipulators|left for stabilizing the receiving beaker
(at the [COLOR] left-beaker point) and right for pouring (at
the [COLOR] right-beaker point)|to execute a complete liquid
transfer."
5) "In the final image of the input, the [COLOR] bounding
box around each beaker and their corresponding [COLOR] grasp
points are displayed (one on the left, one on the right).  The
[COLOR] point on the right beaker indicates where to grasp.
The robot is instructed to use its left arm to hold the left
beaker steady (at the [COLOR] left-beaker point) and its right
arm to grasp the right beaker (at the [COLOR] right-beaker
point), then carefully pour the liquid into the left beaker
until the transfer is complete."
```

5. **Stirring Solution with a Glass Rod:**

```
1) "In the last image, the [COLOR] bounding box highlights
the beaker.  Use the right arm to grasp the spatula and stir
inside that box."
2) "The final image shows a [COLOR] box around the beaker.
Command the right arm to pick up the spatula and stir within
this box."
```

```
3) "In the last frame, a [COLOR] bounding box encloses the
beaker.  Have the right manipulator grasp the spatula and stir
inside that region."
4) "Referencing the last image, you'll see a [COLOR] box
around the beaker.  Instruct the right arm to hold the spatula
and stir within the boxed area."
5) "In the final image, a single [COLOR] bounding box marks
the beaker.  Use the right arm to grasp the spatula and stir
inside the box."
```

6. **Transferring Solid:**

```
1) "In the last image, the [COLOR] boxes highlight the left
(solid) and right (liquid) cups, each with a [COLOR] point.
The left [COLOR] point is where to scoop solid; the right
[COLOR] point marks the liquid surface.  Use the right arm
to grasp the spatula, scoop at the left cup's [COLOR] point,
and pour into the right cup's [COLOR] point."
2) "The last image shows [COLOR] boxes around both cups and
[COLOR] markers for scoop and pour points|the left for solid,
the right for liquid.  With the right arm, grasp the spatula,
scoop at the left [COLOR] point, then deposit into the right
[COLOR] point."
3) "In the final image, two [COLOR] boxes enclose the cups,
each with a [COLOR] point:  left for scooping solid, right for
the liquid level.  The right manipulator holds the spatula,
scoops at the left [COLOR] point, and pours at the right
[COLOR] point."
4) "Refer to the last image's [COLOR] boxes and [COLOR]
points|left at the solid's scoop location, right at the liquid
level.  The right arm grabs the spatula, scoops at the left
[COLOR] point, and delivers into the right [COLOR] point."
5) "In the final image, [COLOR] boxes and points mark the
scoop (left) and pour (right) locations.  Use the right arm
to pick up the spatula, scoop at the left [COLOR] point, and
transfer into the right [COLOR] point."
```

7. **Pressing a Button:**

```
1) "In the image input, the last image is used as a reference
image, with the [COLOR] target point indicating the location
of the switch.  Using the right arm of the robotic arm,
carefully extend to the red switch and flick it to the left
to turn it on."
2) "In the final image of the input, the [COLOR] target point
indicates the location of the switch.  Using the right robotic
arm, the system carefully extends toward the switch and flicks
it to the left to activate it."
3) "The last image in the input sequence serves as a
reference, where the [COLOR] marker denotes the switch
position.  The right manipulator is employed to approach the
switch and toggle it leftward to turn it on."
4) "Refer to the last image in the input, where the [COLOR]
point marks the switch location.  The robot's right arm is
tasked with extending to the switch and flipping it to the
left to power it on."
```

```
5) "In the final reference image, the [COLOR] marker
identifies the location of the switch.  The robotic system
extends its right arm to engage the switch by flicking it to
the left, thereby switching it on."
```

A.8 Error Breakdown

We identify core failure sources across five modules: task decomposition, visual prompting, outcome monitoring, VLA violating prompt constraints, and VLA action failure with no recovery. Figure 17 summarizes statistics from 20 failure cases. The most common errors arise from visual prompt misalignment in cluttered scenes, followed by VLM monitoring and VLA constraint violations. We are addressing these issues by incorporating **multi-view prompt validation**, **sensor-assisted monitoring**, and **stricter prompt supervision during training**, leading to gains in robustness and task success.

A.9 Generalization on environment, and embodiment

We evaluated *RoboChemist*'s generalization across 7 challenging variations, including different platforms, occlusions, cluttered scenes, lighting conditions, table textures, spatial height changes, and embodiments (Figure 16). We evaluated the *Pour Liquid* task and observed strong performance enabled by VLM-based closed-loop feedback and visual prompting (Figure 18).

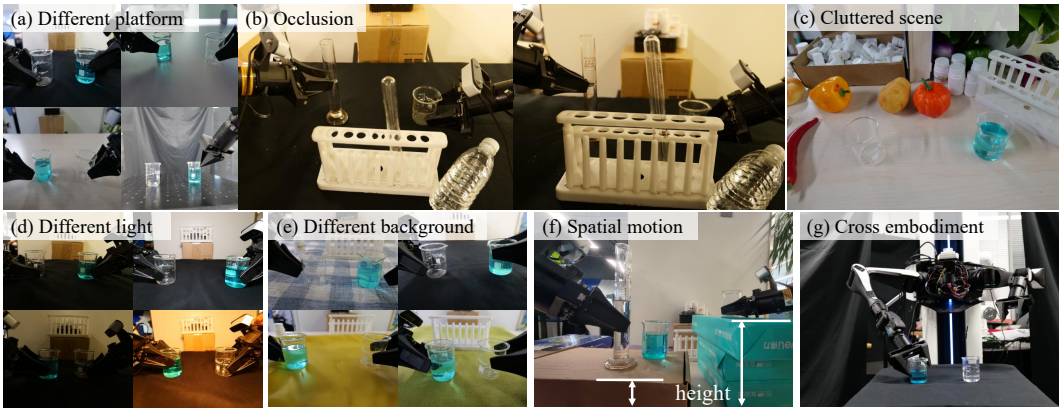

Figure 16: Visualization of primitive task generalization.

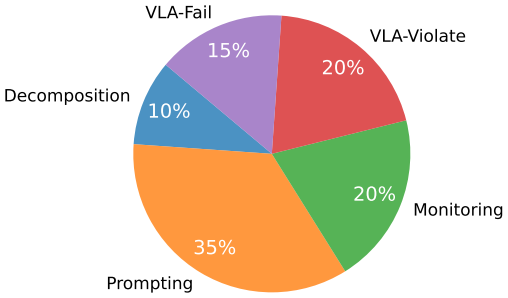

Figure 17: Error breakdown.

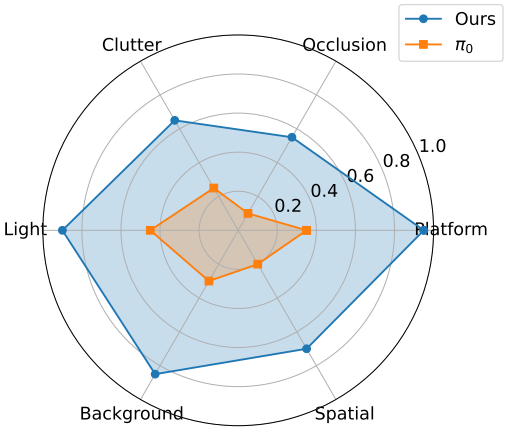

Figure 18: Generalization evaluation.