

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. Finally, Sec. A.7 describes the specific VLA architecture employed in our work.

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

4. Pouring Liquid:

- Objective: Pour a liquid from one container into another.

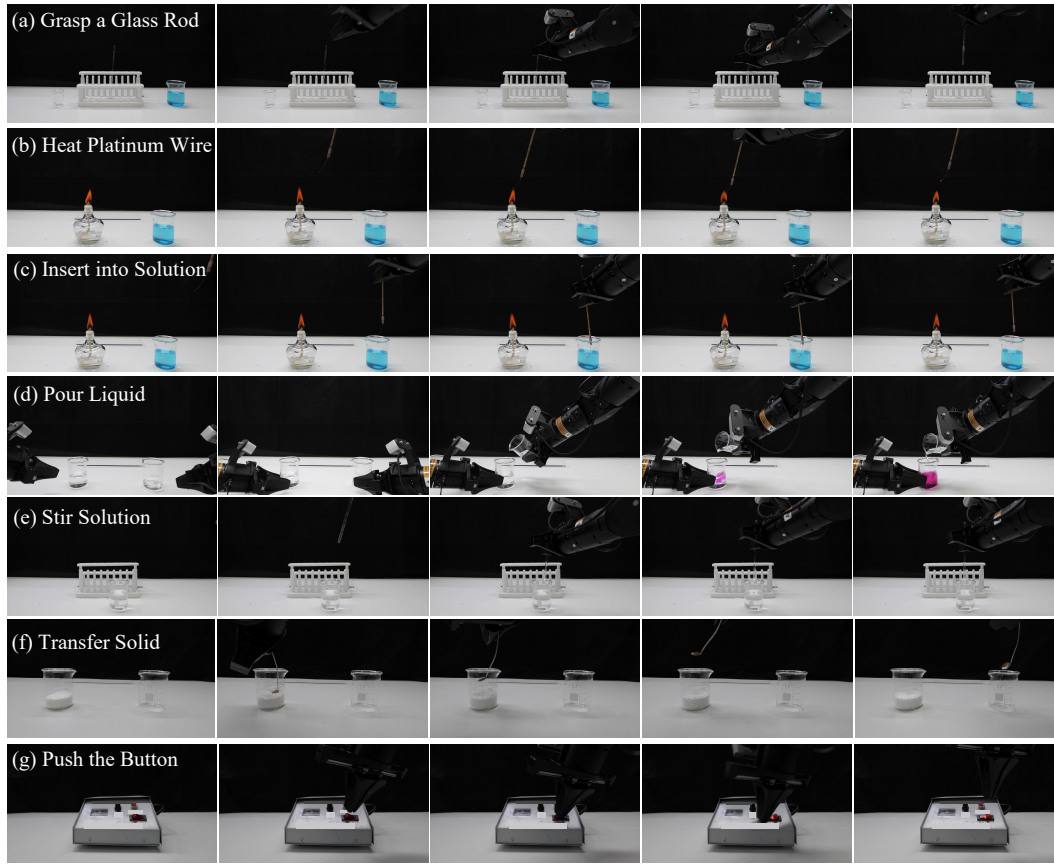


Figure 6: Visualization of primitive tasks.

- 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:

- 614 – Does not press: 0
- 615 – Pressed but not activated: 0.25
- 616 – Pressed but too forcefully, causing movement of the equipment: 0.5
- 617 – Correct press with appropriate force: 1

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

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

620 The Compliance Rate (CR) is the average score of the 20 trials, where each trial is assigned a score
 621 based on the compliance with the task criteria. For example, in the "Grasping a Glass Rod" task,
 622 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
 623 score of 1. Then the Compliance Rate is calculated as:

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

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

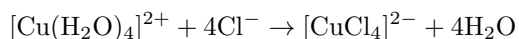
626 This calculation method is used to derive the data presented in Table 2 for the primitive tasks,
 627 where both Success Rate and Compliance Rate are computed for each task. **Note:** The last two
 628 rows in Table 2 correspond to different variants of *RoboChemist*. *RoboChemist w/o CL* represents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

637 A.2 Details of Complete Chemical Tasks and Evaluation Protocols

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

643 1. Mixing NaCl and CuSO₄ Solutions

- 644 • **Objective:** Mix sodium chloride solution with copper sulfate solution and observe the
 645 resulting color change.
- 646 • **Primitive tasks:** Pour one beaker of liquid into another.
- 647 • **Observation and Explanation** This is a coordination reaction in which hydrated copper
 648 ions, initially blue in color, react with chloride ions to form the complex ion $[\text{CuCl}_4]^{2-}$. The
 649 reaction proceeds as:



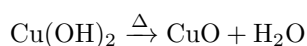
650 As the reaction occurs, the solution color transitions from blue to cyan and then to
 651 green. The final green hue corresponds to the formation of sodium tetrachlorocuprate,
 652 $\text{Na}_2[\text{CuCl}_4]$, indicating successful complexation (as Figure 7 shows).

653 2. Thermal Decomposition of Cu(OH)₂



Figure 7: Visualization of mixing NaCl and CuSO₄ solutions.

- **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 → 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:



This reaction reflects the thermal instability of Cu(OH)₂ and the stability of CuO at elevated temperatures (as Figure 8 shows).

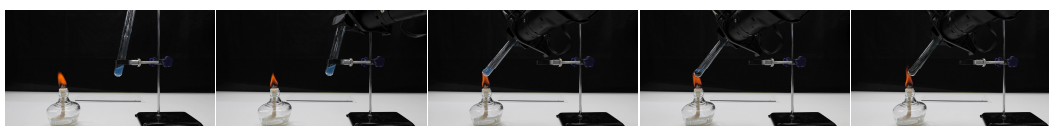


Figure 8: Visualization of thermal decomposition of Cu(OH)₂.

3. Flame Test of CuSO₄ Solution

- **Objective:** Identify the presence of Cu²⁺ ions in copper sulfate through a flame test, investigating its physical property via characteristic flame emission.
- **Primitive tasks:** Grasp platinum wire → Dip into CuSO₄ solution → Heat platinum wire in flame.
- **Observation and Explanation:** When a compound containing Cu²⁺ ions (e.g., CuSO₄) is heated in a flame, it emits a characteristic blue-green color. This flame color originates from the emission spectrum of Cu²⁺ 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 CuSO₄ 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 → Press the heater button to initiate evaporation.

679 • **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
 680 such as sand or other salts remain at the bottom. Upon pressing the heater button, the
 681 solution begins to heat and eventually boils, producing visible bubbles. As heating con-
 682 tinues, the water gradually evaporates, increasing the salt concentration. Once the solution
 683 reaches saturation, solid NaCl crystals begin to precipitate. When all water has evaporated,
 684 white NaCl crystals remain in the beaker, typically hard in texture and free from soluble
 685 contaminants (as Figure 10 shows).
 686

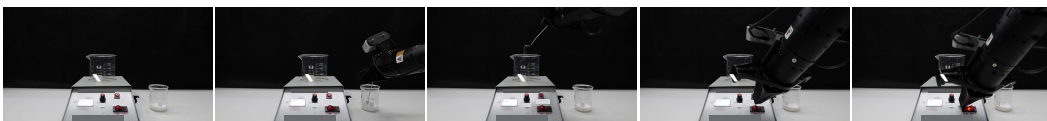


Figure 10: Visualization of evaporation of NaCl solution.

687 5. Acid-Base Neutralization with Phenolphthalein Indicator

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

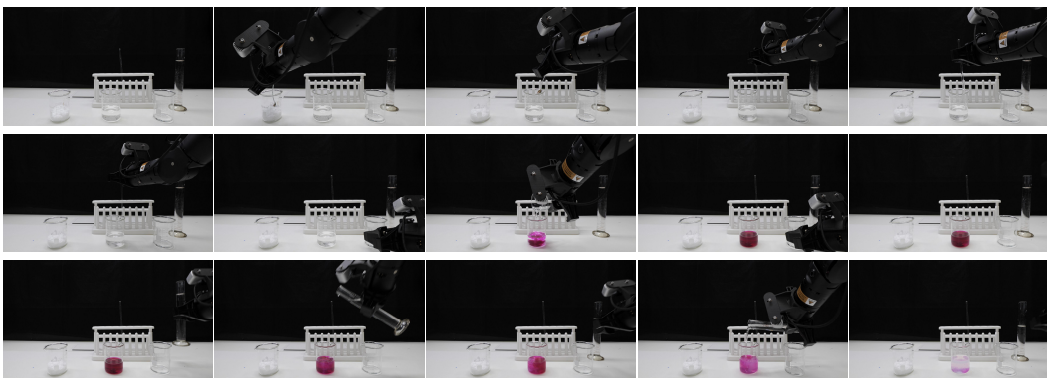
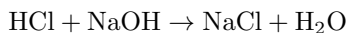


Figure 11: Visualization of acid-base neutralization with phenolphthalein indicator.

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

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

710 A.3 Key Prompts

711 A.3.1 Task Decomposition

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

714 1. Mixing NaCl and CuSO₄ Solutions

You are a lab assistant tasked with mixing NaCl and CuSO₄ 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 CuSO₄ 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]

715

716 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.

...

717

718 3. Flame Test of CuSO₄ 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.

...

719

720 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

...

721

722 5. Acid-Base Neutralization with Phenolphthalein Indicator

You are a lab assistant tasked with performing an acid-base neutralization 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.

...

723

724 A.3.2 Safety Guideline Generation

725 We present examples of our proposed prompt for safety guideline generation prior to performing
726 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

727

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.



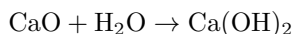
Figure 12: Visualization of primitive task generalization.

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(%) \uparrow CR \uparrow		SR(%) \uparrow CR \uparrow		SR(%) \uparrow CR \uparrow		SR(%) \uparrow CR \uparrow		SR(%) \uparrow CR \uparrow		SR(%) \uparrow CR	
ACT [74]	10	0.050	40	0.250	15	0.075	5	0.025	10	0.025	10	0.050
RDT [15]	0	0.000	10	0.050	60	0.350	35	0.175	75	0.500	60	0.300
π_0 [16]	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:



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).

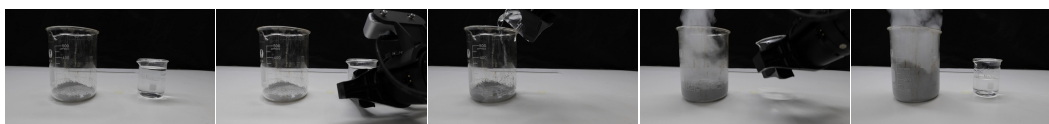
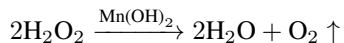


Figure 13: Visualization of combination reaction.

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



Upon the addition of 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).

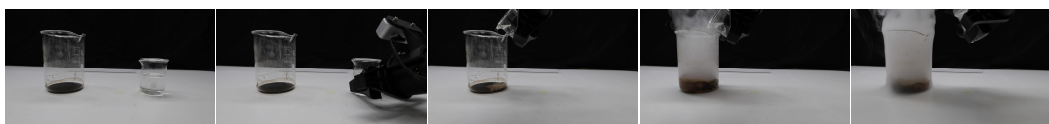
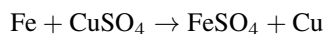
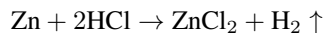


Figure 14: Visualization of decomposition reaction.

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

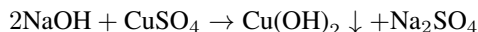


This causes reddish solid copper to precipitate on the iron surface, demonstrating iron’s higher reactivity. Similarly, *RoboChemist* performs acid-metal reactions such as:



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:



forming a distinct blue precipitate of Cu(OH)_2 . Another example includes:

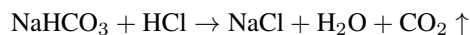




Figure 15: Visualization of displacement reaction between Zn and HCl.

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

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

782 This demonstrates that once *RoboChemist* masters a representative instance of a given experimen-
 783 tal category—such as a flame test—it can generalize this capability to a broader class of similar
 784 procedures. Such compositional generalization underscores the system’s ability to integrate proce-
 785 dural knowledge (via VLM) and low-level execution (via VLA) to generalize from basic actions
 786 to diverse, multi-step chemistry tasks. The system not only completes complex workflows but also
 787 interprets and responds to observed phenomena, showcasing its robust reasoning and adaptability.
 788 Please refer to the [supplemental video](#) for qualitative demonstrations of generalization performance.

789 A.5 Comparison of Visual Prompting Methods

790 This section provides detailed descriptions of the visual prompting strategies used in our comparison
 791 experiments (as shown in Table 4), specifically focusing on ReKep [20], MOKA [65], and our
 792 proposed VLM-based approach. All three methods generate visual prompts that are used as auxiliary
 793 supervision for fine-tuning a π_0 [16] policy model. This standardized setup enables a fair evaluation
 794 of the prompt generation quality across different frameworks.

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

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

815 In contrast, our method uses a vision-language model to directly annotate 2D visual prompts by
 816 pointing to semantically and geometrically relevant locations on input images. These annotated
 817 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

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

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

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

875 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."

876

877 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

878

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."

879

880

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."

881

882

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."

883

5. Stirring Solution with a Glass Rod:

884

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."

885

- 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."

886

887

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."

888

889

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."

890

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."