AHA: A VISION-LANGUAGE-MODEL FOR DETECTING AND REASONING OVER FAILURES IN ROBOTIC MANIPULATION

Jiafei Duan[1,2] **Wilbert Pumacay**[3] **Nishanth Kumar**[1,4]
Yi Ru Wang[1,2] **Shulin Tian**[5] **Wentao Yuan**[1,2]
Ranjay Krishna[2,6] **Dieter Fox**[1,2] **Ajay Mandlekar**[*1] **Yijie Guo**[*1]
[1]NVIDIA, [2]University of Washington, [3]Universidad Católica San Pablo, [4]MIT
[5]Nanyang Technological University, [6]Allen Institute for Artificial Intelligence

aha-vlm.github.io

ABSTRACT

Robotic manipulation in open-world settings demands not only the execution of tasks but also the ability to detect and learn from failures during execution. While recent advances in vision-language models (VLMs) and large language models (LLMs) have enhanced robots' spatial reasoning and problem-solving capabilities, these models often struggle to recognize and reason about failures, limiting their effectiveness in real-world applications. We introduce AHA, an open-source VLM specifically designed to detect and reason about failures in robotic manipulation through natural language. By framing failure detection as a free-form reasoning task, AHA identifies failures and generates detailed explanations adaptable across various robots, tasks, and environments in both simulation and real-world scenarios. To fine-tune AHA, we developed `FailGen`, a scalable simulation framework that procedurally generates AHA dataset—the first large-scale dataset of robotic failure trajectories—by perturbing successful demonstrations from the RLBench simulator. Despite being trained solely on the AHA dataset, AHA generalizes effectively to real-world failure datasets, different robotic systems, and unseen tasks. It surpasses the second-best model by 10.3% and exceeds the average performance of all six compared models—including five state-of-the-art VLMs and one model employing in-context learning—by 35.3% across multiple metrics and datasets. Moreover, we integrate AHA into three VLM/LLM-assisted manipulation frameworks. Its natural language failure feedback enhances error recovery and policy performance through methods such as improving reward functions with Eureka reflection, optimizing task and motion planning, and verifying sub-task success in zero-shot robotic manipulation. Our approach achieves an average task success rate 21.4% higher than GPT-4 models.

1 INTRODUCTION

In recent years, foundation models have made remarkable progress across various domains, demonstrating their ability to handle open-world tasks (Driess et al., 2023; Alayrac et al., 2022; Achiam et al., 2023; Zhang et al., 2023). These models, including large language models (LLMs) and vision-language models (VLMs), have shown proficiency in interpreting and executing human language instructions (Ouyang et al., 2022), producing accurate predictions and achieving strong task performance. However, despite these advancements, key challenges remain—particularly with hallucinations, where models generate responses that deviate from truth. Unlike humans, who can intuitively detect and adjust for such errors, these models often lack the mechanisms for recognizing their own mistakes (Lin et al., 2021; Chen et al., 2021; Heyman, 2008).

Learning from failure is a fundamental aspect of human intelligence. Whether it's a child learning to skate or perfecting a swing, the ability to reflect on and adjust based on feedback is essential for

[*]Equal advising

improvement (Young, 2009; Gopnik, 2020; Heyman, 2008). In machine learning, this process is mirrored through techniques like Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022; Christiano et al., 2017), where human oversight helps guide models toward desired outcomes. This feedback loop plays a critical role in aligning generative models with real-world objectives. However, a crucial question persists: How can we equip these models with the capability to detect and learn from their own errors without human in the loop?

This need is particularly pressing in robotics, where foundation models such as VLMs and LLMs are increasingly used to address open-world tasks. Recent advancements have enabled these models to tackle spatial reasoning, object recognition, and multimodal problem-solving—skills vital for robotic manipulation (Reid et al., 2024; OpenAI, 2024; Yuan et al., 2024; Chen et al., 2024; Wang et al., 2023b). VLMs and LLMs are already being integrated to automate reward generation for reinforcement learning (Ma et al., 2023; 2024), develop task plans for motion planning (Curtis et al., 2024), and even generate zero-shot robot trajectories (Huang et al., 2023; 2024a; Duan et al., 2024; Huang et al., 2024b). While these models excel at task execution, they often face challenges in detecting and reasoning over failures—skills that are crucial for navigating dynamic and complex environments. For example, if a robot drops an object mid-task, a human observer would immediately recognize the error and take corrective action. How can we empower robots with similar capabilities, allowing them not only to perform tasks but also to detect and learn from their mistakes?

In this work, we introduce AHA, an open-source vision-language model (VLM) that detects and reasons about failures in robotic manipulation using natural language. By framing failure detection as a free-form reasoning task, our model not only identifies failures but also generates detailed explanations. This approach allows AHA to adapt to various robots, camera viewpoints, tasks, and environments in both simulation and real-world scenarios. To fine-tune the VLM, we developed `FailGen`, an automated data pipeline that procedurally generates the AHA dataset, a large-scale dataset of simulated robotic manipulation failures. Despite being fine-tuned only on the AHA dataset, AHA demonstrates **strong generalization to real-world failure datasets, different robotic systems, and unseen tasks**, as evaluated on three separate datasets not included in the fine-tuning. `FailGen`'s flexible data generation pipeline integrates seamlessly with various simulators, enabling scalable procedural generation of failure demonstrations.

Upon fine-tuning AHA, we also benchmarked it against six state-of-the-art VLMs, both open-source and proprietary, evaluating across four metrics on three datasets. AHA outperformed GPT4o model by more than 20.0% on average across datasets and metrics, and by over 43.0% compared to LLaVA-v1.5-13B (Liu et al., 2023a), the base model from which AHA is derived. This demonstrates AHA's exceptional ability to detect and reason about failures in robotic manipulation across embodiment and domains. Moreover, AHA integrates seamlessly into VLM-guided robotic systems, providing failure feedback to improve reward functions through Eureka reflection, enhancing task and motion planning, and verifying sub-task success in zero-shot robotic manipulation. Across three downstream tasks, our approach achieved an average task success rate 21.4% higher than GPT4 models, highlighting the **effectiveness of AHA in providing accurate natural language failure feedback to aid with improving downstream task performance through error correction**.

In summary, our contributions are threefold: (1) developing `FailGen`, a scalable simulation framework for procedural generation of failure demonstrations, and curating the AHA dataset, the first large-scale robot failure dataset; (2) AHA, a new open-source VLM for reasoning about failures in manipulation tasks, outperforming six proprietary and open-source models; and (3) integration of AHA into three VLM/LLM-assisted manipulation frameworks, demonstrating that its natural language failure feedback improves error recovery and policy performance in downstream tasks.

2 RELATED WORK

Failure Detection in Robotic Manipulation. Failure detection and reasoning have long been studied in the Human-Robot Interaction (HRI) community (Ye et al., 2019; Khanna et al., 2023) and in works leveraging Task and Motion Planning (TAMP) (Garrett et al., 2020). With the recent widespread adoption of LLMs and VLMs in robot manipulation systems—either for generating reward functions or synthesizing robot trajectories (Ma et al., 2023; 2024) in a zero-shot manner—the importance of detecting task failures has regained prominence (Huang et al., 2023; Duan et al., 2024; Skreta et al., 2024; Ha et al., 2023). Most modern approaches focus on using off-the-shelf VLMs or LLMs as

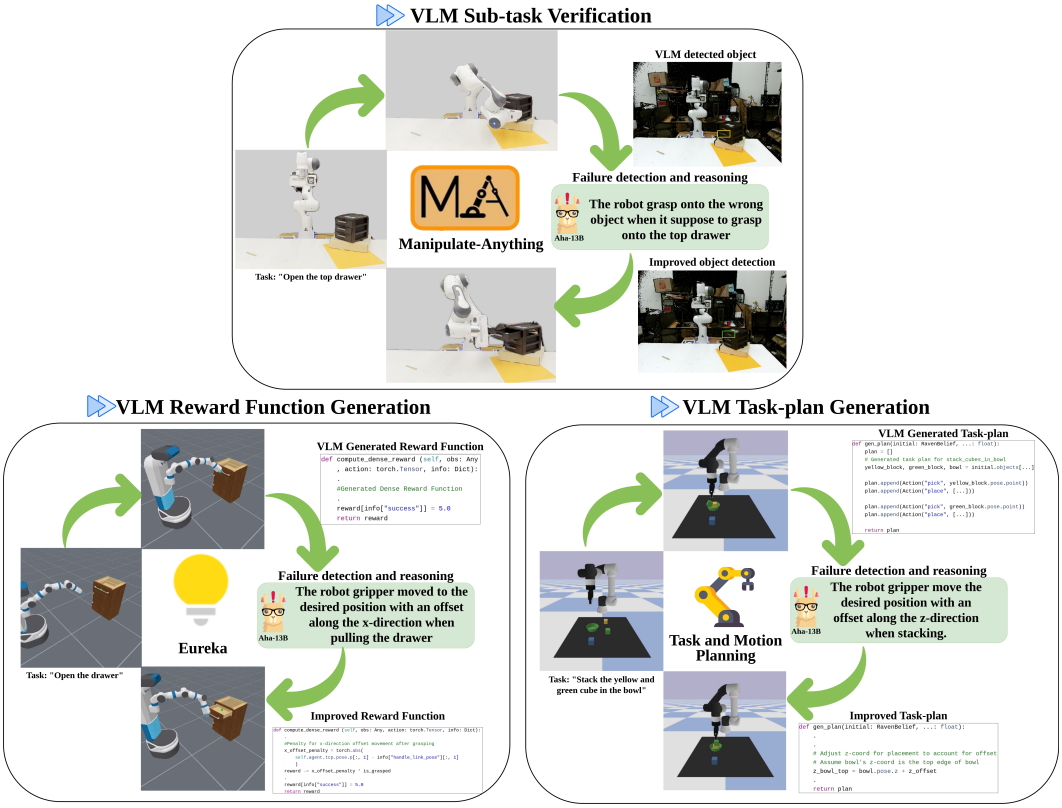


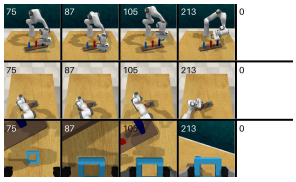
Figure 1: AHA is a Vision-Language Model designed to detect and reason about failures in robotic manipulation. As an instruction-tuned VLM, it can enhance task performance in robotic applications that utilize VLMs for reward generation, task planning, or sub-task verification. By incorporating AHA into the reasoning pipeline, these applications can achieve accelerated and improved performance.

success detectors (Ma et al., 2022; Ha et al., 2023; Wang et al., 2023a; Duan et al., 2024), and some employ instruction-tuning of VLMs to detect failures (Du et al., 2023). However, these methods are often limited to binary success detection and does not provide language explanations for why failures occur. Our framework introduces failure reasoning in a new formulation, generating language-based explanations of failures to aid robotics systems that leverage VLMs and LLMs in downstream tasks.

Foundation Models for Robotic Manipulation. In recent years, leveraging foundation models for robotic manipulation has become an active area of research. This interest is driven by the effectiveness of VLMs in interpreting open-world semantics and their adaptability to cross-task generalization (Duan et al., 2022; Hu et al., 2023; Firoozi et al., 2023; Urain et al., 2024). Two main approaches have emerged in this domain. The first approach utilizes VLMs and LLMs in a promptable manner. Many works focus on designing sophisticated visual prompts to aid low-level action generation, enabling robots to perform tasks based on prompts derived from visual inputs (Liu et al., 2024a; Huang et al., 2024a;b). The second approach involves instruction-tuning specialized VLMs for domain-specific tasks in robotics (Li et al., 2024). For instance, RoboPoint (Yuan et al., 2024) has been instruction-tuned for spatial affordance prediction, while Octopi (Yu et al., 2024) is an instruction-tuned VLM designed for physical reasoning over objects using tactile image inputs. These specialized VLMs are general-purpose, capable of generalizing beyond their training data distribution, and can be seamlessly integrated into downstream manipulation pipelines. Our approach aligns with the second line of thought. We develop a scalable method for generating large-scale instruction-tuning data in simulation to fine-tune general-purpose VLMs specialized in detecting and reasoning about failures in robotic manipulation. Moreover, our problem formulation extends beyond manipulation tasks, making it transferable to other domains in robotics.

Data Generation in Robotics There have been many methods in robotic manipulation that automate data generation of task demonstrations at scale (Mandlekar et al., 2023; Hoque et al., 2024), whether

Table 1: **AHA datasets for instruction-tuning.** We combined the AHA dataset, our large-scale robotic manipulation failure dataset, with VQA and object detection data. By incorporating this diverse data mix into the fine-tuning process, AHA is able to reason about failures in robotic manipulation across different domains, embodiments, and tasks.

Source	The AHA dataset (Train)	VQA (Liu et al., 2023a)	LVIS (Gupta et al., 2019)
			
Quantity	49K	665K	100K
Query	For the given sub-tasks, first determine it has succeed by choosing from ["yes", "no"] and then explain the reason why the current sub-tasks has failed.	What is the cat doing in the image?	Find all instances of drawer.
Answer	No, The robot gripper rotated with an incorrect roll angle	The cat is sticking its head into a vase or container, possibly drinking water or investigating the interior of the item.	[(0.41, 0.68, 0.03, 0.05), (0.42, 0.73, 0.04, 0.08), ...]

for training behavior cloning policies, instruction-tuning VLMs (Yuan et al., 2024), or curating benchmarks for evaluating robotic policies in simulation (Xie et al., 2024; Pumacay et al., 2024). A well-known example is MimicGen (Mandlekar et al., 2023), which automates task demonstration generation via trajectory adaptation by leveraging known object poses. Additionally, works like RoboPoint use simulation to generate general-purpose representations for robotic applications, specifically for fine-tuning VLMs. Similarly, systems like The Colosseum Pumacay et al. (2024) automate data generation for curating benchmarks in robotic manipulation. Our approach aligns more closely with RoboPoint, as we also leverage simulation to generate data for instruction-tuning VLMs. However, unlike RoboPoint, we focus on synthesizing robotic actions in simulation rather than generating representations like bounding boxes or points.

3 THE AHA DATASET

We leveraged `FailGen` to procedurally generate the AHA dataset from RLBench tasks (James et al., 2020) and used it for the instruction-tuning of AHA. In this section, we begin by categorizing common failure modes in robotics manipulation and defining a taxonomy of failures in Section 3.1. Next, we explain how this taxonomy is used with `FailGen` to automate the data generation for the AHA dataset in simulation in Section 3.2.

3.1 FAILURE MODES IN ROBOTIC MANIPULATION

To curate an instruction-tuning dataset of failure trajectories for robotic manipulation tasks, we began by systematically identifying prevalent failure modes. Our approach involved a review of existing datasets, including DROID (Khazatsky et al., 2024) and Open-X Embodiment (Padalkar et al., 2023), as well as an analysis of policy rollouts from behavior cloning models. We examined failures occurring in both teleoperated and autonomous manipulations. Building upon prior works, such as REFLECT (Liu et al., 2023d), we formalized a taxonomy encompassing seven distinct failure modes commonly observed in robotic manipulation: incomplete grasp, inadequate grip retention, misaligned keyframe, incorrect rotation, missing rotation, wrong action sequence, and wrong target object.

Incomplete Grasp (`No_Grasp`) Failure: `No_Grasp` is an object-centric failure that occurs when the gripper reaches the desired grasp pose but fails to close before proceeding to the next keyframe.

Inadequate Grip Retention (`Slip`) Failure: `Slip` is an object-centric failure that happens after the object has been successfully grasped. As the gripper moves the object to the next task-specific keyframe, the grip loosens, causing the object to slip from the gripper.

Misaligned keyframe (`Translation`) Failure: This action-centric failure occurs when the gripper moves toward a task keyframe, but a translation offset along the X, Y, or Z axis causes the task to fail.

Incorrect Rotation (`Rotation`) Failure: `Rotation` is an action-centric failure that occurs when the gripper reaches the desired translation pose for the sub-task keyframe, but there is an offset in roll, yaw, or pitch, leading to task failure.

Missing Rotation (`No_Rotation`) Failure: `No_Rotation` is an action-centric failure that happens when the gripper reaches the desired translation pose but fails to achieve the necessary rotation (roll, yaw, or pitch) for the sub-task, resulting in task failure.

Wrong Action Sequence (`Wrong_action`) Failure: `Wrong_action` is an action-centric failure that occurs when the robot executes actions out of order, performing an action keyframe before the correct one. For example, in the task `put_cube_in_drawer`, the robot moves the cube toward the drawer before opening it, leading to task failure.

Wrong Target Object (`Wrong_object`) Failure: `Wrong_object` is an object-centric failure that occurs when the robot acts on the wrong target object, not matching the language instruction. For example, in the task `pick_the_red_cup`, the gripper picks up the green cup instead, leading to task failure.

3.2 Implementation of the AHA dataset

The AHA dataset is generated with `RLBench`, utilizing its keyframe-based formulation to dynamically induce failure modes during task execution. RLBench natively provides keyframes for task demonstrations, which enables flexibility in both object manipulation (handling tasks with varying objects) and the sequence of actions (altering the execution order of keyframes). Building on this foundation, we leverage `FailGen`, our custom environment wrapper to wrap around `RLBench` that allows for task-specific trajectory modifications through keyframes perturbations, object substitutions, and reordering of keyframe sequences. This framework systematically generates failure trajectories aligned with the taxonomy defined in Section 3.1, yielding a curated dataset of 49k failure-question pairs.

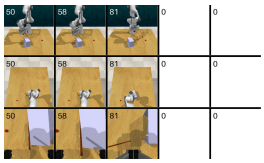
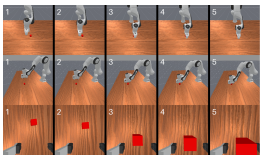
To generate the AHA dataset, we systematically sweep through all keyframes in each RLBench task, considering all potential configurations of the seven failure modes that could result in overall task failure. By leveraging the success condition checker in the simulation, we procedurally generate `YAML`-based configuration files by sweeping through each failure mode across all keyframes. These files provide details on potential failure modes, parameters (such as distance, task sequence, gripper retention strength, etc.), and corresponding keyframes that `FailGen` should perturb to induce failure. Additionally, we incorporate language templates to describe what the robot is doing between consecutive keyframes. Using these descriptions along with the failure modes, we can systematically curate question-answer pairs for each corresponding failure mode.

For specific failure modes, `No_Grasp` is implemented by omitting gripper open/close commands at the relevant keyframes, effectively disabling gripper control. `Slip` introduces a timed release of the gripper shortly after activation. `Translation` and `Rotation` perturb the position and orientation of a keyframe, respectively, while `No_Rotation` constrains the keyframe's rotational axis. `Wrong_Action` reorders keyframe activations to simulate incorrect sequencing, and `Wrong_Object` reassigns the keyframes intended for one object to another, maintaining the relative pose to mimic improper object manipulation. Using this pipeline, we also successfully generated a failure dataset from ManiSkill (Mu et al., 2021) and adapted RoboFail (Liu et al., 2023d) for the evaluation of AHA. This further demonstrates the generalizability and versatility of `FailGen` in generating failure cases across different simulation environments.

4 Method

This section outlines the failure reasoning problem formulation (Sec.4.1) used to fine-tune and evaluate AHA. Next, we discuss the curated data mix used for co-finetuning AHA (Sec.4.2). Finally, we detail the instruction fine-tuning pipeline and the model architecture selection for AHA (Sec.4.3).

Table 2: **Examples of detection and reasoning over failures by various VLMs.** We sampled one evaluation data point from each of the three evaluation datasets and compared the reasoning predictions generated by different state-of-the-art VLMs alongside our AHA-13B. Our model consistently provides the most semantically similar reasoning to the ground-truth answers for the given queries.

Evaluation Datasets	AHA dataset (Train)	Maniskill-Fail	RoboFail (Liu et al., 2023d)
			
Question	At the current sub-task, the robot is picking up the hockey stick. First, determine whether it has succeeded by choosing from ["yes", "no"]. If not, explain why it failed?	At the current sub-task, the robot is picking up the red cube. First, determine whether it has succeeded by choosing from ["yes", "no"]. If not, explain why it failed?	At the current sub-task, the robot is placing the apple into the bowl. First, determine whether it has succeeded by choosing from ["yes", "no"]. If not, explain why it failed?
LLaVA 1.5-13B	The image displays a time-lapse sequence of a robotic arm grabbing a hockey stick.	Based on the image, it depicts a different sub-task of a robot grasping a red block.	For the pick-up task, i would say the robot arm has succeeded in picking.
GPT4o	No, the robot arm fails to hold the hockey stick securely.	Yes, the robot arm has successfully picked up the red block.	No, the robot gripper failed to pick up the bowl in the second frame.
Gemini 1.5 Flash	Yes.	Yes, Yes, Yes.	Frame 1, no the gripper is not grasping the bowl and move it.
AHA-13B (Ours)	No, the robot slip the object out of its gripper.	No, the robot gripper fails to close the gripper.	No, this is not the right action sequence for the task.
Ground-truth Answer	No, the robot slip the object out of its gripper.	No,the robot gripper fails to close.	No, this is not the right action sequence for the task.

4.1 FAILURE REASONING FORMULATION

Unlike previous works (Liu et al., 2023d; Skreta et al., 2024; Duan et al., 2024) that primarily focus on detecting task success as binary classification problem, we approach failure reasoning by first predicting a binary success condition ("Yes" or "No") of the given sub-task based on a language specification and an input image prompt. If the answer is "No," the VLM is expected to generate a concise, free-form natural language explanation detailing why the task is perceived as a failure.

To formulate failure reasoning, we prompt the VLMs to analyze the trajectory failures at the current sub-task and provide reasoning for *why* or *what* led to the failure. We define manipulation task trajectories as a series of sub-tasks $\{S_0, S_1, S_2, \ldots, S_t\}$, where each sub-task is represented by two consecutive keyframes. For example, in a task like "stacking cubes," a sub-task could represent a primitive action, such as 'picking up the cube.' For the input formulation used in VLMs for instruction fine-tuning and evaluation, we required a query prompt with an input image for prompting the VLMs. The query prompt was generated using a template corresponding to the current sub-task the robot is performing. To capture the temporal relationships within the action sequence, the input image was constructed by selecting a single frame that represents the robot's trajectory up to the current sub-task and concatenating it with frames from other viewpoints in the rollout sequence, this is depicted in Table 2.

This input frame is built by concatenating all keyframes up to the current sub-task in temporal order, from left to right, with any remaining keyframes replaced by white image patches. To mitigate occlusions, we also included all the available camera viewpoints, concatenating them alongside the temporal sequence, and provide a detailed task description in the prompt, as illustrated in Table 1 (left image). The image data is structured as a matrix $\mathbf{I}$, where each row corresponds to a different camera viewpoint $\{V_0, V_1, \ldots, V_n\}$ and each column captures the temporal sequence of keyframes $\{S_0, S_1, S_2, \ldots, S_t\}$. The matrix $\mathbf{I}$ is defined as follows:

$$\mathbf{I} = \begin{pmatrix} I_{V_0 S_0} & I_{V_0 S_1} & \cdots & I_{V_0 S_t} \\ I_{V_1 S_0} & I_{V_1 S_1} & \cdots & I_{V_1 S_t} \\ \vdots & \vdots & \ddots & \vdots \\ I_{V_n S_0} & I_{V_n S_1} & \cdots & I_{V_n S_t} \end{pmatrix}$$

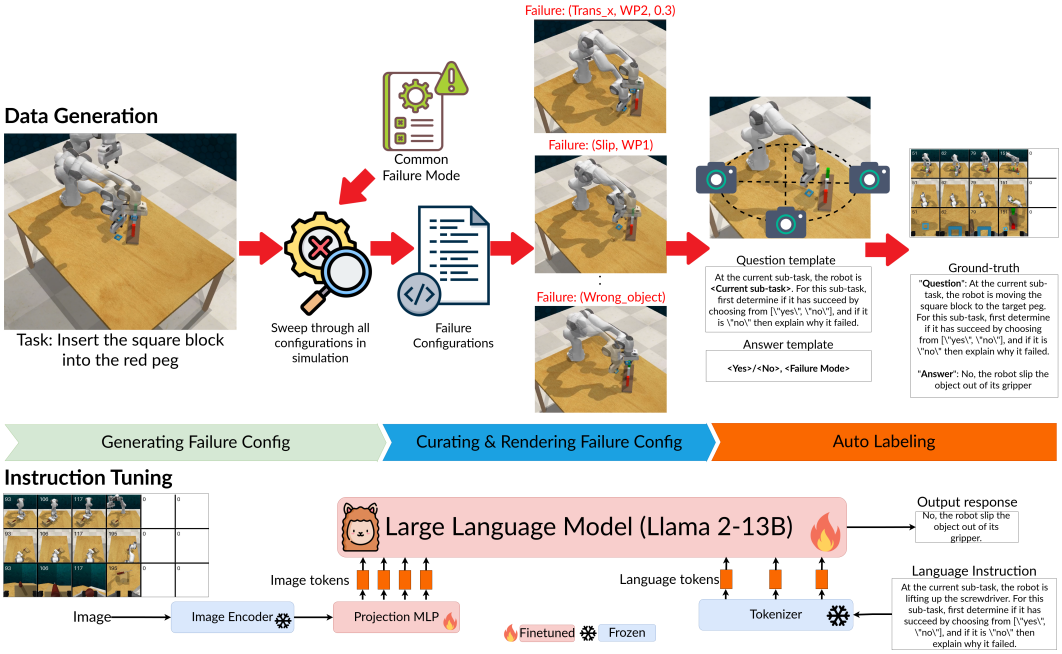


Figure 2: **Overview of AHA Pipeline**. (Top) The data generation for AHA is accomplished by taking a normal task trajectory in simulation and procedurally perturbing all keyframes using our taxonomy of failure modes. Through `FailGen`, we systematically alter keyframes to synthesize failure demonstrations conditioned on the original tasks. Simultaneously, we generate corresponding query and answer prompts for each task and failure mode, which are used for instruction-tuning. (Bottom) The instruction-tuning pipeline follows the same fine-tuning procedure as LLaVA-v1.5 Liu et al. (2023a), where we fine-tune only the LLM base model—in this case, LLaMA-2-13B and the projection linear layers, while freezing the rest of the mode

Where $I_{V_i S_j}$ represents the image from viewpoint V_i at sub-task S_j, this formulation for curating images serves as a general approach for formatting all datasets used for fine-tuning and evaluation. This structured input enables consistent handling of data across different tasks and viewpoints. Overall, our failure reasoning problem is to prompt VLM with sub-task discription and keyframe trajectory image to predict the success condition and language description of failure reason for each sub-task, as shown in Table 2.

4.2 SYNTHETIC DATA FOR INSTRUCTION-TUNING

To facilitate the instruction-tuning of AHA, we needed to systematically generate failure demonstration data. To achieve this, we developed `FailGen`, an environment wrapper that can be easily applied to any robot manipulation simulator. `FailGen` systematically perturbs successful robot trajectories for manipulation tasks, transforming them into failure trajectories with various modes of failure as depicted in Figure 2 (Top image). Using `FailGen`, we curated the AHA dataset (Training) dataset by alternating across 79 different tasks in the RLBench simulator, resulting in 49k failure image-text pairs. Furthermore, following proper instruction-tuning protocols for VLMs (Liu et al., 2023a) and building on prior works (Brohan et al., 2023; Yuan et al., 2024), co-finetuning is crucial to the success of instruction fine-tuning of VLMs. Therefore, in addition to the AHA dataset, we co-finetuned AHA with general visual question-answering (VQA) datasets sourced from internet data, which helps models retain pre-trained knowledge. Specifically, we included the VQA dataset (Liu et al., 2023a), containing 665k conversation pairs, and the LVIS dataset (Gupta et al., 2019), which comprises 100k instances with predicted bounding box centers and dimensions, as summarized in Table 1.

Table 3: **Quantitative Evaluation on Failure Detection and Reasoning.** AHA-13B was evaluated and benchmarked against three open and three closed-source VLMs and one visual prompting baseline across three evaluation datasets. AHA-13B outperformed all other VLMs on every evaluation dataset and nearly every evaluation metric, with the exception of the AHA (Test) dataset, where GPT-4o exceeded AHA-13B by less than 3%.

Models	Evaluation Datasets	Evaluation Metrics			
		ROUGE$_L$ ↑	Cosine Similarity ↑	Binary Success(%) ↑	LLM Fuzzy Match ↑
LLaVA-v1.5-13B (Liu et al., 2023a)	**AHA dataset (Test set)**	0.061	0.208	0.080	0.648
	ManiSkill-Fail	0.000	0.208	0.022	0.270
	RoboFail (Liu et al., 2023d)	0.000	0.203	0.000	0.404
LLaVA-NeXT-34B (Liu et al., 2024b)	**AHA dataset (Test set)**	0.013	0.231	0.017	0.626
	ManiSkill-Fail	0.001	0.195	0.007	0.277
	RoboFail (Liu et al., 2023d)	0.018	0.188	0.017	0.351
Qwen-VL (Bai et al., 2023)	**AHA dataset (Test set)**	0.000	0.161	0.000	0.426
	ManiSkill-Fail	0.037	0.301	0.116	0.034
	RoboFail (Liu et al., 2023d)	0.000	0.159	0.000	0.050
Gemini-1.5 Flash (Reid et al., 2024)	**AHA dataset (Test set)**	0.120	0.231	0.371	0.566
	ManiSkill-Fail	0.003	0.121	0.014	0.032
	RoboFail (Liu et al., 2023d)	0.000	0.042	0.000	0.393
GPT-4o	**AHA dataset (Test set)**	0.251	0.308	0.500	**0.784**
	ManiSkill-Fail	0.142	0.335	0.688	0.453
	RoboFail (Liu et al., 2023d)	0.114	0.318	0.554	0.438
GPT-4o-ICL (5-shot)	**AHA dataset (Test set)**	0.226	0.380	0.611	0.776
	ManiSkill-Fail	0.341	0.429	0.971	0.630
	RoboFail (Liu et al., 2023d)	0.236	0.429	0.571	0.418
AHA-7B	**AHA dataset (Test set)**	0.434	0.574	0.691	0.695
	ManiSkill-Fail	**0.609**	0.680	1.000	0.532
	RoboFail (Liu et al., 2023d)	0.204	0.394	0.625	0.439
AHA-13B (Ours)	**AHA dataset (Test set)**	**0.446**	**0.583**	**0.702**	0.768
	ManiSkill-Fail	0.600	**0.681**	**1.000**	**0.633**
	RoboFail (Liu et al., 2023d)	**0.280**	**0.471**	**0.643**	**0.465**

Table 4: **Ablation on Different Base LLMs for Fine-Tuning.** We fine-tuned AHA-13B using both LLaMA-2-13B and Vicuna-1.5-13B as base LLM models. The quantitative results show that the average performance difference between the two models is less than 2.5%, indicating that our failure formulation and the AHA dataset are effective regardless of the base model selection.

Models	AHA dataset (Test)				ManiSkill-Fail				RoboFail			
	ROUGE$_L$ ↑	Cos Sim ↑	BinSucc(%) ↑	Fuzzy Match ↑	ROUGE$_L$ ↑	Cos Sim ↑	BinSucc(%) ↑	Fuzzy Match ↑	ROUGE$_L$ ↑	Cos Sim ↑	BinSucc(%) ↑	Fuzzy Match ↑
AHA-13B (Llama-2)	**0.446**	0.583	0.702	**0.768**	**0.600**	**0.681**	**1.000**	0.633	0.280	**0.471**	**0.643**	0.465
AHA-13B (Vicuna-1.5)	0.458	**0.591**	**0.709**	0.695	0.574	0.657	**1.000**	**0.851**	**0.290**	0.468	**0.661**	**0.605**

4.3 INSTRUCTION FINE-TUNING

We followed the instruction-tuning pipeline outlined by (Liu et al., 2023b). As depicted in Fig. 2, our model architecture includes an image encoder, an linear projector, a language tokenizer, and a transformer-based language model. The image encoder processes images into tokens, which are projected by a 2-layer linear into the same space as the language tokens. These multimodal tokens are then concatenated and passed through the language transformer. All components are initialized with pre-trained weights. During fine-tuning, only the projector and transformer weights are updated, while the vision encoder and tokenizer remain frozen. The model operates autoregressively, with the objective of predicting response tokens and a special token marking the boundary between instruction and response.

5 EXPERIMENTAL RESULTS

In this section, we first evaluate the detection and reasoning performance of AHA against six state-of-the-art VLMs, both open-source and proprietary, including those utilizing in-context learning. We perform these evaluations across three diverse datasets, covering out-of-domain tasks, various simulation environments, and cross-embodiment scenarios. Next, we assess AHA's generalization capabilities and its retention of general world knowledge, a key attribute expected from VLMs. We evaluate whether this knowledge is preserved after fine-tuning on domain-specific data. Lastly, we explore the potential for AHA to enhance downstream robotic manipulation tasks.

Table 5: **Quantitative Evaluation on Standard VQA Benchmarks.** AHA-13B performs on par with LLaVA-13B Liu et al. (2023a), the VLM from which AHA adapts its fine-tuning strategy.

	MMBench (Liu et al., 2023c)	ScienceQA (Lu et al., 2022)	TextVQA (Singh et al., 2019)	POPE (Li et al., 2023)	VizWiz (Gurari et al., 2018)
LLaVA-13B (LLama-2) (Liu et al., 2023a)	**67.70**	**73.21**	**67.40**	**88.00**	53.01
AHA-13B (LLama-2)	65.20	71.94	65.20	85.74	**53.45**

5.1 EXPERIMENTAL SETUP

To quantitatively evaluate AHA's detection and reasoning capabilities for failures in robotic manipulation, we curated two datasets and adapted an existing failure dataset for benchmarking. To ensure a fair comparison of free-form language reasoning, we also employed four different evaluation metrics to measure semantic similarity between sentences.

Benchmarks We curated three datasets to evaluate AHA's reasoning and failure detection capabilities, benchmarking against other state-of-the-art VLMs. The first dataset, AHA dataset (Test), includes 11k image-question pairs from 10 RLBench tasks, generated similarly to the fine-tuning data via `FailGen` (Section 3.2) but without overlapping with the tasks from the finetuning dataset. It evaluates AHA's ability to generalize to novel, out-of-domain tasks. The second dataset, ManiSkill-Fail, comprises 130 image-question pairs across four tasks in ManiSkill (Mu et al., 2021), generated using `Failgen` wrapper on Maniskill simulator. This dataset assesses AHA's performance in a different simulator and under changing viewpoints. Lastly, we adapted a failure benchmark from the RoboFail dataset (Liu et al., 2023d) (which is the one of the fewer robot failure dataset ever being curated), which features real-world robot failures in seven UR5 robot tasks. This allows for evaluation across real-world trajectories and different embodiment's.

Baselines We compare AHA against three state-of-the-art open-source VLMs: LLaVA-1.5, LLaVA-NeXT, and Qwen-VL, as well as two proprietary VLMs: GPT-4o and Gemini 1.5 Flash. Additionally, we evaluated in-context learning (GPT-4o-ICL) by providing 5 input-output pairs from our each of the evaluation datasets as demonstrations.

Evaluation Metrics To ensure fair and accurate evaluation across the three datasets and all baseline methods on success detection and free language reasoning, we utilize four evaluation metrics. First, the **ROUGE-L score** assesses the quality of generated text in tasks like summarization and machine translation by measuring the similarity between the candidate text and a reference text, focusing on the Longest Common Subsequence (LCS) of words. Second, we employ **Cosine Similarity** rather than distance to evaluate the similarity between text documents, sentences, or word embeddings by representing them as high-dimensional vectors to avoid the "curse of dimensionality". Third, **LLM Fuzzy Matching** uses an external language model to assess the semantic similarity between the reference sentence and predicted sentences in a teacher-student prompting format; specifically, we leverage an unseen language model, `claude-3-sonnet` from Anthropic for evaluating the LLM Fuzzy Matching score. Lastly, we evaluate the model's predictions against the ground truth for success detection using a **Binary success rate**.

5.2 QUANTITATIVE EXPERIMENTAL RESULTS

We contextualize the performance of AHA by conducting a systematic evaluation of failure reasoning and detection across these three datasets, general VQA datasets, and performed ablation studies.

AHA generalizes across embodiments, unseen environments, and novel tasks. To ensure fairness and eliminate bias in the detection and reasoning capabilities of AHA, we evaluated it on three different datasets that were never seen during fine-tuning, each designed to test a specific form of generalization. First, on the AHA dataset (test) dataset, AHA demonstrated its ability to **generalize reasoning across tasks and new behaviors within the same domain, outperforming the second-best performing VLM, GPT-4o**, by an average margin of 0.126 across all evaluation metrics. Second, we assessed AHA-13B on a dataset generated by the `Failgen` wrapper in a different simulation domain, ManiSkill, showing that our model outperforms GPT-4o-ICL by an average of 0.134 across all metrics. Lastly, to **demonstrate generalization to real-world robots and different embodiments, we evaluated AHA-13B on RoboFail (Liu et al., 2023d), where it outperforms GPT-4o-ICL by 0.049.**

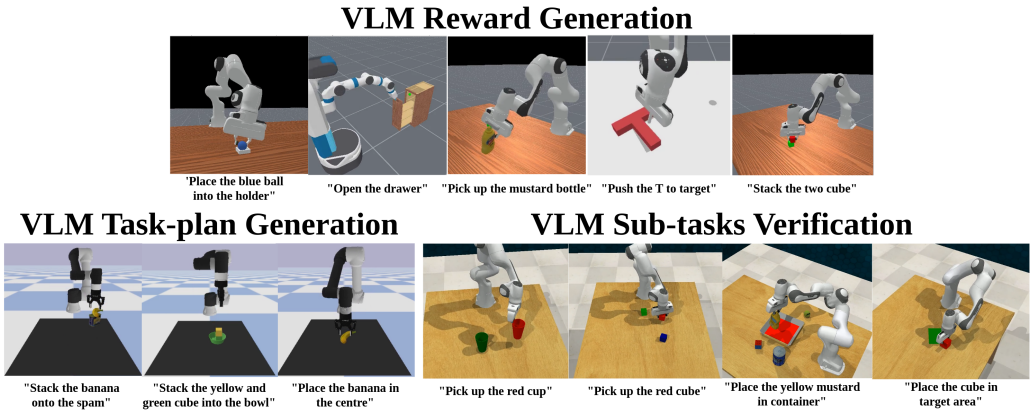


Figure 4: **Downstream Robotic Application.** We demonstrated that AHA can be integrated into existing LLM/VLM-assisted robotic applications to provide failure reasoning and feedback, helping to accelerate and improve task success rates in these systems.

AHA retains common sense knowledge. We evaluated AHA-13B's performance on various VQA benchmarks and present the results in Table 5 . AHA-13B **performs comparably to LLaVA-v1.5-13B (LLama-2) (Liu et al., 2023a)** , with only a 1.5% margin difference as depicted in Table 5. Notably, LLaVA-v1.5-13B is a VLM trained on the same pre-trained weights as AHA-13B but fine-tuned on VQA data. This indicates that AHA-13B is capable of functioning as a general purpose VLM, in addition to excelling at failure reasoning.

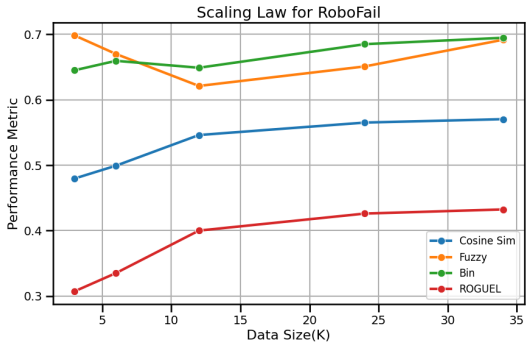


Figure 3: **Scaling law with the AHA dataset**. Scaling of effect of model performance with varying domain specific fine-tuning data.

AHA's performance scales with data size. We evaluated Aha's performance using a range of AHA data for instruction fine-tuning, spanning [3k, 6k, 12k, 34k, 48k, 60k], and co-trained individual checkpoints corresponding to these data sizes as shown in Figure 3. The model was then assessed on the ManiSkill-Fail dataset across four evaluation metrics. A linear fit of the results showed an average slope of 0.0021 across all metrics, indicating a **clear scaling effect with fine-tuning on our procedurally generated data pipeline**. This suggests that further scaling of the generated data may lead to improved model performance.

5.3 DOWNSTREAM ROBOTICS TASKS

We demonstrate that AHA's failure detection and reasoning capabilities are useful across a wide spectrum of downstream robotics applications. This includes automatic reward generation for reinforcement learning applications (Ma et al., 2023), automatic task plan generation for task and motion planning applications (Curtis et al., 2024), and as an improved verification step for automatic data generation systems (Duan et al., 2024). Videos and detailed improved reward function, task plan, example videos from each applications and etc can be found on the project page: aha-vlm.github.io/.

AHA enables efficient reward synthesis for reinforcement learning. To evaluate this downstream task, we adapted Eureka's (Ma et al., 2023) implementation to the ManiSkill simulator, which offers more state-based manipulation tasks. We strictly followed the Eureka reward function generation and reflection pipeline, modifying it by incorporating perception failure feedback via either AHA-13B or GPT-4o (acting as a baseline) to enhance the original LLM reflection mechanism. Instead of only including a textual summary of reward quality based on policy training statistics for automated reward editing, we further incorporated explanations of policy failures based on evaluation rollouts. We

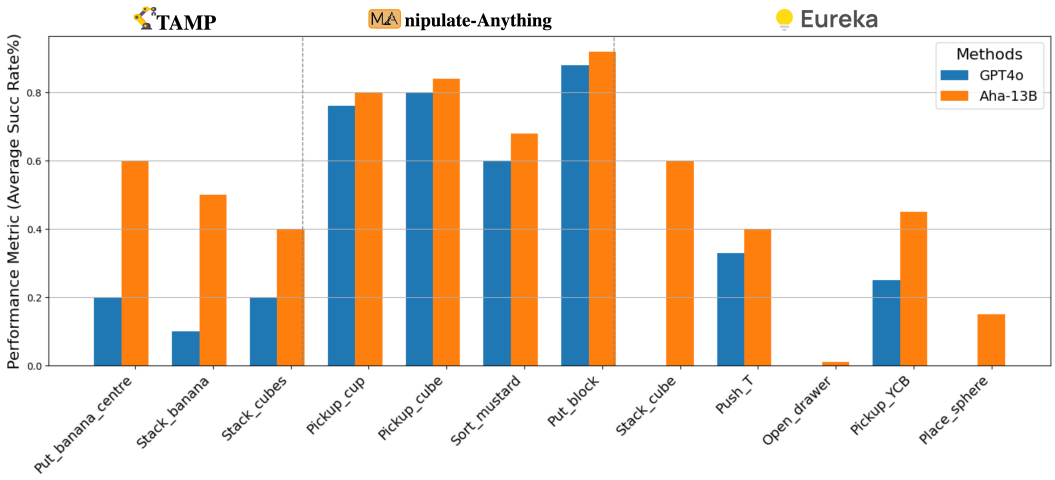


Figure 5: **Downstream Robotic Application Performance.** AHA-13B outperforms GPT-4o in reasoning about failures within these robotic applications, leading to improved performance of the downstream tasks.

evaluated our approach on five reinforcement learning tasks from ManiSkill, ranging from tabletop to mobile manipulation. To systematically assess the reasoning capabilities of different VLMs under budget constraints, we sampled one reward function initially and allowed for iterations over two sessions of GPT API calls. Each policy was trained using PPO over task-specific training steps and evaluated across 1,000 test steps. During policy rollouts, we employed either AHA-13B or GPT-4o for reward reflection to improve the reward function. Comparing the evaluated policy success rates using different failure feedback VLMs, we observed that AHA-13B provided intuitive, human-level failure reasoning that aided in modifying and improving generated dense reward functions. This resulted in success across all five tasks within the budget constraints, and our approach **outperformed GPT4o by a significant margin of 22.34% in task success rate** shown in Figure 5.

AHA refines task-plan generation for TAMP. To demonstrate AHA's utility within a planning system, we incorporated our approach into PRoC3S (Curtis et al., 2024). The PRoC3S system solves tasks specified in natural language by prompting an LLM for a Language-Model Program (LMP) that generates plans, and then testing a large number of these plans within a simulator before executing valid plans on a robot. If no valid plan can be found within a certain number of samples (100 in our experiments), the LLM is re-prompted for a new LMP given failure information provided by the environment. Importantly, as is typical of TAMP methods, the original approach checks for a finite set of failures (inverse kinematics, collisions, etc.) from the environment, and returns any sampled plan that does not fail in any of these ways. We incorporated a VLM into this pipeline in two ways: (1) we prompt the VLM with visualizations of failed plan executions within the simulator, ask it to return an explanation for the failure, and feed this back to PRoC3S' LLM during the LMP feedback stage, (2) after PRoC3S returns a valid plan, we provide a visualization of this to the VLM and ask it to return whether this plan truly achieves the natural language goal, with replanning triggered if not. We compared GPT-4o and AHA-13B as the VLM-based failure reasoning modules within this implementation of PRoC3S across three tasks (shown in Figure 4). Each task was evaluated over 10 trials, with a maximum of 100 sampling steps and three feedback cycles provided by either GPT-4o or AHA-13B. The success rate for each task was recorded. As shown in Figure 5, utilizing AHA-13B for **failure reasoning significantly improved the task success rate and outperforming GPT-4o by a substantial margin of 36.7%**.

AHA improves task verification for zero-shot robot data generation. To demonstrate AHA's utility in zero-shot robot demonstration generation, we integrated our approach into the `Manipulate-Anything` framework. This open-ended system employs various Vision-Language Models (VLMs) to generate diverse robot trajectories and perform a wide range of manipulation tasks without being constrained by predefined actions or scenarios. A critical component of `Manipulate-Anything` is its sub-task verification module, which analyzes past and current frames to decide whether a sub-task has been achieved before proceeding or re-iterating over the

previous sub-task. We replaced the original VLM (GPT-4V) in the sub-task verification module with AHA-13B and evaluated performance across four RLBench tasks (Figure 4), conducting 25 episodes for each task. Our results show that **substituting the sub-task verification module's VLM with AHA improved reasoning accuracy and overall task success by an average of 5%**.

6 CONCLUSION

Limitations AHA currently outputs language reasoning that is closely aligned with the failure scenarios in the fine-tuning data. However, we aim to capture more open-ended failures, such as those arising from large pretrained policies like OpenVLA (Kim et al., 2024), RT2 (Brohan et al., 2023), and Octo (Team et al., 2024). Additionally, while `FailGen` systematically curates failure data from simulations, distilling large pretrained policies to perform diverse tasks in simulation and sampling various failure modes would allow us to generate more open-ended failure examples. This could significantly enhance the instruction-tuning of AHA.

Conclusion In conclusion, this work presents AHA, an open-source vision-language model that significantly advances the ability of robotic systems to detect and reason about failures in manipulation tasks through natural language. By framing failure detection as a free-form reasoning task, AHA not only identifies failures but also provides detailed explanations that are adaptable across various robots, tasks, and environments in both simulation and real-world scenarios. The development of `FailGen` and the curation of the AHA dataset have been instrumental in fine-tuning AHA, enabling the generation of a large and diverse dataset of robotic failure trajectories for robust training. Our extensive evaluations demonstrate that AHA surpasses the second-best model by 10% and exceeds the average performance of all six compared models—including five state-of-the-art VLMs and one model employing in-context learning—by 30% across multiple metrics and datasets. When integrated into three different VLM/LLM-assisted manipulation frameworks, AHA's natural language failure feedback leads to significant improvements in error recovery and policy performance, achieving an average task success rate 21.4% higher than that of GPT-4 models. These results underscore the effectiveness of AHA in enhancing downstream task performance through accurate error detection and correction. This work highlights the critical importance of enabling robots to recognize and learn from their failures—a key step toward developing truly intelligent and autonomous systems. By providing a scalable framework for failure detection and reasoning, we open new avenues for research in robotic learning and adaptation. Future work will explore the integration of AHA into more complex robotic systems and tasks, as well as the expansion of `FailGen` to include a wider range of failure scenarios. We believe that this approach will significantly contribute to the advancement of robotic manipulation in open-world settings, ultimately bringing us closer to robots that can seamlessly operate alongside humans in real-world environments.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Aidan Curtis, Nishanth Kumar, Jing Cao, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Trust the proc3s: Solving long-horizon robotics problems with llms and constraint satisfaction, 2024. URL `https://arxiv.org/abs/2406.05572`.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. *arXiv preprint arXiv:2303.07280*, 2023.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*, 2024.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *arXiv preprint arXiv:2312.07843*, 2023.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the international conference on automated planning and scheduling*, volume 30, pp. 440–448, 2020.

Alison Gopnik. Childhood as a solution to explore–exploit tensions. *Philosophical Transactions of the Royal Society B*, 375(1803):20190502, 2020.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5356–5364, 2019.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pp. 3766–3777. PMLR, 2023.

Gail D Heyman. Children's critical thinking when learning from others. *Current directions in psychological science*, 17(5):344–347, 2008.

Ryan Hoque, Ajay Mandlekar, Caelan Garrett, Ken Goldberg, and Dieter Fox. Intervengen: Interventional data generation for robust and data-efficient robot imitation learning. *arXiv preprint arXiv:2405.01472*, 2024.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Zhibo Zhao, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *arXiv preprint arXiv:2403.08248*, 2024a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024b.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Parag Khanna, Elmira Yadollahi, Mårten Björkman, Iolanda Leite, and Christian Smith. User study exploring the role of explanation of failures by robots in human robot collaboration tasks. *arXiv preprint arXiv:2303.16010*, 2023.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023c.

Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*, 2023d.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Yecheng Jason Ma, William Liang, Hung-Ju Wang, Sam Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. Dreureka: Language model guided sim-to-real transfer. *arXiv preprint arXiv:2406.01967*, 2024.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

OpenAI. Hello gpt-4o, May 2024. URL `https://openai.com/index/hello-gpt-4o`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. Replan: Robotic replanning with perception and language models. *arXiv preprint arXiv:2401.04157*, 2024.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Julen Urain, Ajay Mandlekar, Yilun Du, Mahi Shafiullah, Danfei Xu, Katerina Fragkiadaki, Georgia Chalvatzaki, and Jan Peters. Deep generative models in robotics: A survey on learning from multimodal demonstrations. *arXiv preprint arXiv:2408.04380*, 2024.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. *arXiv preprint arXiv:2310.01361*, 2023a.

Yi Ru Wang, Jiafei Duan, Dieter Fox, and Siddhartha Srinivasa. Newton: Are large language models capable of physical reasoning? *arXiv preprint arXiv:2310.07018*, 2023b.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2024.

Sean Ye, Glen Neville, Mariah Schrum, Matthew Gombolay, Sonia Chernova, and Ayanna Howard. Human trust after robot mistakes: Study of the effects of different forms of robot communication. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–7. IEEE, 2019.

H Peyton Young. Learning by trial and error. *Games and economic behavior*, 65(2):626–643, 2009.

Samson Yu, Kelvin Lin, Anxing Xiao, Jiafei Duan, and Harold Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv preprint arXiv:2405.02794*, 2024.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.