

VLA-GAUGE: Systematically Evaluating Generalization and Robustness in Vision-Language-Action Models

Jun Xiao, Rui Ye, Dong Cao, Jinyu Gu, Ying Wang*, and Zhiliang Zhu

Abstract—Vision-Language-Action models (VLAs) have achieved high success rates on familiar manipulation tasks, yet it remains unclear whether such scores reflect genuine zero-shot generalization and perturbation robustness. Existing evaluation frameworks provide limited capability-level diagnosis and combinatorial perturbation coverage, often without connecting diagnosis to targeted model refinement. We introduce VLA-GAUGE, a diagnostic framework that unifies both evaluations under a shared executable task representation. It combines ① paired in-distribution and out-of-distribution tasks across six capability suites, ② single-factor sensitivity analysis with combinatorial perturbation screening and evaluation, and ③ diagnosis-driven refinement through targeted data augmentation and model fine-tuning. We evaluate 12 VLAs spanning different model sizes and action-modeling paradigms under a unified protocol. We find that ① The mean Zero-Shot Score on VLA-GAUGE is only 22.3%, with SimpleVLA-RL scoring 8.0% and ranking second to last among the evaluated models. ② Under three representative combinatorial perturbation configurations, the mean R-Score is 9.3. ③ Diagnosis-driven fine-tuning improves $\pi_{0.5}$'s zero-shot generalization and robustness under a selected perturbation combination on held-out scenarios by 72.4 and 31.8 scoring points, respectively; the latter exceeds the 10.9-point gain from LIBERO-PLUS single-factor fine-tuning despite fewer successful training trajectories. These results show that high reported success rates on familiar manipulation tasks do not guarantee strong zero-shot generalization to unseen tasks or robustness to combinatorial perturbations, while diagnostic evaluation provides actionable signals for targeted model improvement.

Index Terms—Benchmark, Perturbation Robustness, Robotic Manipulation, Vision-Language-Action Models, Zero-Shot Generalization.

I. INTRODUCTION

Vision-Language-Action Models (VLAs) provide a promising path toward building general-purpose robotic policies by jointly modelling *visual observations*, *natural language instructions*, and *robot actions* [1]–[3]. In recent years, with the continued advancement of vision-language pre-training, large-scale robotic datasets, and action generation methods, VLA models have achieved high success rates on familiar manipulation tasks [4]–[6]. However, a real robotic system must not only complete tasks drawn from familiar distributions, but also generalize to unseen objects, instructions, scenarios, and task

compositions, while maintaining stable execution under non-ideal conditions such as visual variation, spatial displacement, and control error. Prior research indicates that changes in lighting, tabletop appearance, and surrounding objects can occur simultaneously in real-world manipulation settings [7]. Related experiments further show that a model may fail under combinatorial perturbations even when it successfully completes the task under each perturbation in isolation [8]. Therefore, high success rates on familiar manipulation tasks are by themselves insufficient to establish zero-shot generalization to unseen task distributions or execution robustness under combinatorial perturbations, and a systematic evaluation of both aspects is needed.

Systematic evaluation of VLA models should go beyond aggregate performance on a specific task distribution to separately assess capability-level zero-shot generalization across distributions and execution robustness under perturbations. Recent robotic manipulation benchmarks have expanded beyond standard task completion to evaluate generalization to unseen objects and environments [9]–[11] and robustness to variations in appearance, layout, viewpoint, and language [4]–[7], [12], [13]. These efforts broaden the scope of evaluation, advance different stages of the diagnostic pipeline, and provide a basis for understanding how models behave under varying task requirements and environmental changes. It remains an open question, however, how evaluation results can be translated into reliable diagnosis of model weaknesses, targeted improvement measures, and independent verification of whether those improvements take effect. Viewed against this complete diagnostic objective, existing evaluations still exhibit the following *four limitations* across different settings: ① **limited capability attribution**: task-level success rates aggregate multiple capability demands into a single scalar, rendering performance differences unattributable to specific weaknesses; ② **inconsistent distributional evaluation**: capability dimensions lack unified, paired in-distribution (ID) and out-of-distribution (OOD) tests; ③ **limited perturbation coverage**: robustness evaluations consider few perturbation types and parameter settings and remain largely static or single factor; and ④ **missing diagnosis-to-refinement loop**: existing benchmarks do not yet integrate fine-grained results from evaluations of model capabilities, ID/OOD performance, and perturbation sensitivity into a closed-loop workflow. Such a workflow guides targeted data augmentation and model optimization based on capability gaps and the conditions under which performance declines, then re-evaluates the model on held-out scenarios.

Addressing these limitations requires a unified diagnostic

*Ying Wang is the corresponding author.

Jun Xiao, Rui Ye, Dong Cao, Ying Wang, and Zhiliang Zhu are with the Software College, Northeastern University, Shenyang, China. E-mail: xiao-jun@mails.neu.edu.cn; yerui@mails.neu.edu.cn; caodong@mails.neu.edu.cn; wangying@swc.neu.edu.cn; ZHUZhiLiang_NEU@163.com.

Jinyu Gu is with the School of Computer Science, Shanghai Jiao Tong University, Shanghai, China, and also with Shanghai Syslong Information Technology Co., Ltd. E-mail: gujinyu@sjtu.edu.cn.

benchmark for VLAs, whose design poses *two core challenges*: ① **Diagnostically attributable task construction**: Multiple capabilities are coupled in robotic manipulation tasks, and changing predefined factors related to a target capability may also alter the demands on other capabilities. A central challenge is to provide sufficient task diversity within each group and reliably attribute performance differences to specific capability bottlenecks while keeping task goals, success criteria, and non-target requirements as consistent as possible (see §III-A). ② **Diagnostic combinatorial perturbation design**: As the number of perturbation factors and the number of possible values per factor increase, the combination space grows rapidly, making exhaustive enumeration costly. Yet overly aggressive screening may miss important dependencies across perturbation categories. Another central challenge is to select combinations that are affordable to evaluate, interpretable for diagnosis, and representative of real-world deployment conditions while calibrating perturbation strength, ensuring task feasibility, and preserving task semantics (see §III-B).

To address these limitations and challenges, we introduce VLA-GAUGE, which uses a unified executable task representation to jointly evaluate zero-shot generalization and perturbation robustness for VLA models while supporting diagnosis-driven model refinement. Its *key innovations* include: ① **Capability-oriented zero-shot generalization evaluation**: We design controlled task variants for six capabilities and construct paired ID and OOD tasks. We keep task goals and success criteria as consistent as possible while controlling non-target factors to reduce confounding across capabilities. This design supports evaluation of how well learned capabilities transfer and helps identify generalization bottlenecks. ② **Perturbation robustness evaluation**: We construct a hierarchical perturbation space that satisfies the requirements of task feasibility and semantic consistency, calibrate perturbation strength through single-factor sensitivity analysis, and select representative Perturbation Factor Combinations (PFCs). This makes it possible to examine the joint influence of multiple perturbation categories at an affordable evaluation cost, and provides a basis for identifying potential performance problems under combinatorial perturbations before real deployment. ③ **Diagnosis-driven model refinement**: Using the capability gaps found in the earlier fine-grained evaluations of model capability, ID and OOD performance, and perturbation sensitivity, we select challenging scenarios. We then combine these scenarios with the matching optimization suggestions to guide targeted data augmentation and model fine-tuning. Finally, we re-evaluate the model on a held-out set to measure how well these changes work, which closes the loop of diagnosis-driven model optimization.

Using VLA-GAUGE, we evaluate the zero-shot generalization and perturbation robustness of 12 VLAs spanning different model sizes and action-modeling paradigms under a unified protocol. Our *key findings* are: ① The mean zero-shot score across all models is only 22.3%, with 32.9% on ID tasks and only 7.2% on OOD tasks. SimpleVLA-RL (e.g., a reported success rate of 99.1% on LIBERO) achieves a zero-shot score of only 8.0% on VLA-GAUGE, ranking second to last among the evaluated models. Spatial reasoning is the

weakest capability dimension overall, with an average score of 13.0%, a weakness already evident under ID conditions. Compositional capability performs worst on OOD tasks, with an average score of only 2.8% and 9 of the 12 models scoring zero. ② Under three representative combinatorial perturbation configurations, the mean R-Score is 9.3; 34 out of 36 model-configuration pairs score at or below 17.7, and no model consistently leads across all three configurations. The mean R-Score falls from 16.5 to 2.9 as the selected combinations move from one to three perturbation categories. ③ Diagnosis-driven fine-tuning improves $\pi_{0.5}$'s zero-shot generalization and robustness under a selected perturbation combination on held-out scenarios by 72.4 and 31.8 scoring points, respectively. The robustness gain substantially exceeds the 10.9-point gain from fine-tuning on single-factor perturbation data from LIBERO-PLUS, despite using fewer successful trajectories. Overall, these results show that high reported in-distribution success rates do not guarantee strong zero-shot generalization to unseen tasks or robustness to combinatorial perturbations, while diagnostic evaluation provides actionable signals for targeted model improvement, validating the need for a unified framework like VLA-GAUGE.

In summary, this paper makes the following contributions:

- **A unified evaluation framework**: We introduce VLA-GAUGE, a diagnostic framework for VLA models based on a unified formulation of executable tasks. It integrates capability-oriented paired ID/OOD evaluation, perturbation robustness evaluation, and diagnosis-driven model refinement into a unified closed loop of evaluation, diagnosis, optimization, and re-evaluation.
- **A systematic empirical study**: We systematically evaluate 12 VLA models under a unified protocol, revealing capability-level bottlenecks in zero-shot generalization and robustness to combinatorial perturbations. We validate the effectiveness of diagnosis-driven model refinement through re-evaluation on held-out scenarios and discuss limitations and future research directions for VLA models and their evaluation.
- **Open-source Artifacts**: We release the complete VLA-GAUGE evaluation suites, training demonstrations, fine-tuned VLA models, and tools for automated generation and evaluation on our website (<https://vla-gauge-site.pages.dev/>). These resources support reproducible evaluation, model optimization, and future research.

II. RELATED WORK

A. Generalization and Robustness Challenges of VLA Models

Recent advances in robot learning have changed the field from task-specific policy networks to general-purpose VLA models [16], [17]. Pioneering works such as $\pi 0$ [18] and *OpenVLA* [3] have contributed to this transition by leveraging large-scale embodied datasets (e.g., *Open-X Embodiment* [19]) and integrating visual encoders (e.g., *SigLIP* [20]) with language models. These models map open-vocabulary instructions and visual observations into executable robot actions [21]. However, existing VLA models still differ substantially in their action representations and decoding mechanisms. They can be categorized into *discrete action models*, *continuous action models*, and *hybrid action models*.

TABLE I: A systematic comparison of the existing benchmarks.

Focus: primary evaluation emphasis; **#Tasks**: number of tasks; **#Objects**: number of objects; **Fine-Grained**: fine-grained capability diagnosis; **ID/OOD**: in-distribution/out-of-distribution zero-shot generalization evaluation; **Robustness**: robustness testing under diverse perturbations; **Optimization**: diagnosis-driven model refinement and re-evaluation. Support level: ✓: full, ✗: partial, ✖: none.

Benchmark	Focus	#Tasks	#Objects	Fine-Grained	ID/OOD	Robustness	Optimization
LIBERO [14]	Knowledge-Transfer	130	67	✗	✖	✖	✖
Behavior-1K [15]	Long-horizon + Dataset	1,000	9,331	✗	✖	✖	✖
COLOSSEUM [7]	Robustness	20	147	✗	✗	✗	✗
SimplerEnv [9]	In-distribution Generalization	10	16	✗	✗	✗	✖
VLA-Test [12]	Robustness	4	74	✗	✓	✗	✖
VLABench [10]	In-distribution Generalization + Long-horizon	100	2,164	✗	✗	✗	✖
VLA-Arena [13]	Robustness	170	122	✗	✗	✗	✖
AGNOSTOS [11]	In-distribution Generalization	23	112	✖	✓	✖	✖
LIBERO-X [6]	Robustness	600	96	✗	✗	✗	✖
LIBERO-Pro [5]	Robustness	40	77	✗	✓	✗	✖
LIBERO-Plus [4]	Robustness	10,030	911	✗	✓	✗	✗
VLA-GAUGE	Zero-shot Generalization + Robustness	6,300	3,089	✓	✓	✓	✓

Although these three action modeling paradigms improve the expressiveness of VLA models, they also introduce different generalization and robustness challenges.

- **Discrete action models**: such as *UniVLA* [22], *Nora* [23], *Fast* [24], *DeepThinkVLA* [25] and *SimpleVLA-RL* [26], quantize continuous control signals into action tokens, which enables them to reuse the autoregressive generation capability of language models. However, this discretization may discard fine-grained motion information and increase sensitivity to token design and training distributions. This often results in degraded generalization when facing unseen object compositions, complex layouts, or tasks that demand precise control.
- **Continuous action models**: such as *OpenVLA-OFT* [27], *RIPT-VLA* [28], and *CronusVLA* [29], directly generate low-level control signals or action fragments, making them closer to real robotic control interfaces. However, they are more susceptible to visual observation shifts, variations in initial states, and accumulated execution errors, which can reduce control stability under out-of-distribution scenarios and perturbed conditions.
- **Hybrid-action models**: such as $\pi 0.5$ [30], combine high-level semantic decision-making with low-level continuous control. However, this combination may amplify inconsistencies between semantic grounding, target selection, and action execution, making failure sources more coupled and difficult to locate.

Although the three types of models differ in action representation and decoding mechanisms, they all reveal an unstable mapping among vision-language understanding, action representation, and control execution in VLA models. This indicates that the generalization challenge of VLA models arises from both unseen tasks and their dependence on training data compositions (objects, scenarios, and actions), while their robustness challenge extends across perception, decision-making, and execution, rather than being confined to performance fluctuations under single visual perturbations.

B. Limitations of Current Robotic Manipulation Benchmarks

The above challenges indicate that robotic manipulation benchmarks for generalization and robustness diagnosis should satisfy **four requirements**: ① fine-grained capability diagnosis that attributes model failures to specific stages along the perception, understanding, reasoning, and execution pipeline;

② unified evaluation of zero-shot generalization under both in-distribution (ID) and out-of-distribution (OOD) scenarios; ③ systematic robustness testing under representative perturbations drawn from real-world deployment conditions; and ④ a closed-loop mechanism that leverages diagnosis results to guide model refinement and re-evaluation.

Long-horizon and knowledge-transfer robotic manipulation benchmarks. Benchmarks such as LIBERO [14], CALVIN [31], *RLBench* [32], and *VLABench* [10] provide unified environments for model comparison through standardized task suites, long-horizon manipulation procedures, or knowledge-transfer settings. However, they mainly focus on overall completion rates over predefined tasks, with limited fine-grained failure diagnosis and systematic ID/OOD scenario splits. As a result, it remains difficult to assess the true generalization ability of models under unseen tasks and complex scenario variations.

Generalization-driven robotic manipulation benchmarks. Benchmarks such as *REALM* [33], *SimplerEnv* [9], *Gem-Bench* [34], *AGNOSTOS* [11], and *Maniskill2* [35] extend the generalization evaluation from the perspectives of unseen objects, unseen environments, layout variations, and *Sim2Real* transfer. However, these benchmarks primarily evaluate distribution shifts at the task or environment level, without a unified ID/OOD evaluation across visual perception, instruction understanding, spatial reasoning, and action execution. This makes it difficult to pinpoint the root causes of model degradation in OOD scenarios or to support subsequent refinement and re-evaluation.

Robustness and perturbation robotic manipulation benchmarks. Benchmarks such as LIBERO-PLUS [4], LIBERO-Pro [5], VLA-Arena [13], COLOSSEUM [7], VLA-Test [12], and LIBERO-X [6] extend robustness evaluation through perturbations in object layouts, visual appearances, camera viewpoints, and language expressions. VLA-Test focuses on automatic test generation, while LIBERO-PLUS and COLOSSEUM attempt to connect perturbation evaluation with model improvement. However, most benchmarks remain limited to static or single perturbations, failing to adequately cover dynamic or combinatorial perturbations as encountered in real deployments. They offer no unified mechanism to translate perturbation results into actionable diagnostic optimization.

Table I systematically compares these benchmarks. While they collectively cover long-horizon manipulation, knowledge transfer, in-distribution generalization, and perturbation robustness, each addresses only a subset of these dimensions.

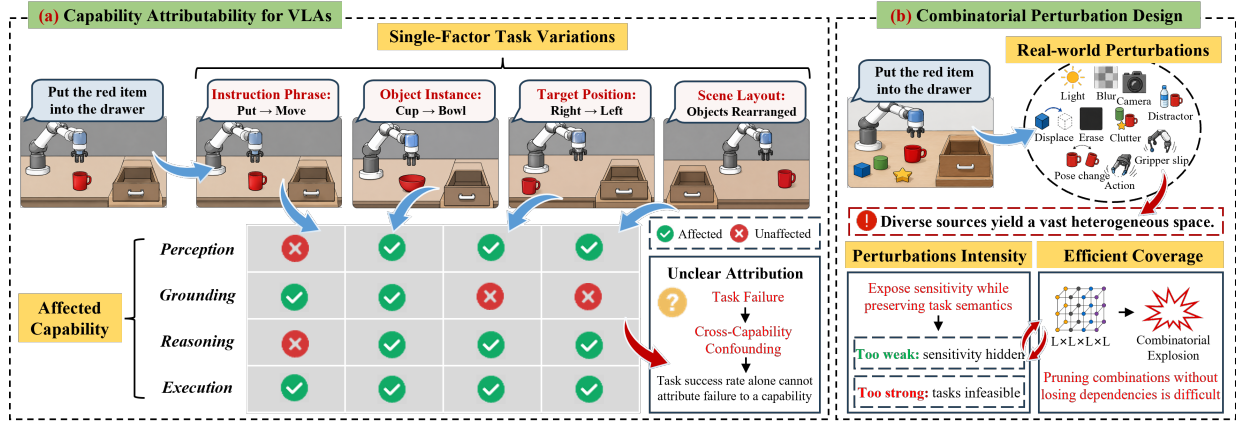


Fig. 1: Examples Illustrating Challenges in VLA Generalization and Robustness Evaluation

More critically, none provides fine-grained diagnosis of the unstable mapping among vision-language understanding, action representation, and control execution in VLA models, nor unified in-distribution versus out-of-distribution analysis across capability axes. Moreover, existing benchmarks offer little support for linking single-factor and combinatorial perturbation evaluation with capability-level diagnosis, targeted model optimization, and re-evaluation on held-out data. In contrast, VLA-GAUGE is the first systematic framework that addresses all limitations together through a unified workflow of evaluation, diagnosis, model optimization, and re-evaluation. Spanning 6,300 task instances and 3,089 objects, the framework unifies three critical assessment axes: in-distribution versus out-of-distribution generalization, robustness under both isolated and combinatorial perturbations, and iterative re-evaluation driven by optimization feedback. This integrated design enables fine-grained, dimension-level diagnosis of model failures across various capability axes and perturbative conditions, while simultaneously providing actionable insights that directly inform and guide subsequent policy optimization.

III. PRELIMINARIES

A. Capability Attributability for VLA Generalization Evaluation

Given a task instruction and current observations, a VLA policy generates action commands through continuous interaction with the environment. Consequently, task completion relies on a multidimensional set of capabilities: visual perception, semantic grounding, world knowledge, spatial reasoning, action reasoning, and their synergistic coordination. Task-level success rates aggregate the requirements of these capabilities into a single metric, which fails to indicate which dimension of capability supports success or to pinpoint which dimension is responsible for failure. Moreover, imitation learning [36] training may cause the policy to exploit recurring correlations between visual observations and action trajectories. Therefore, high performance on familiar tasks does not sufficiently demonstrate the model’s capability generalization.

Multidimensional capability evaluation demands diagnostic attributability. We consider a capability dimension to be attributable when performance variance across a task suite can be reliably traced to predefined variations in that specific

dimension. To enable such interpretation, task objectives, success criteria, motor demands, and other extraneous factors must be held as consistent as possible across comparisons. Simply scaling up the number of tasks or altering surface-level visual attributes does not ensure attributability. Instead, the evaluation suite should be structured around explicit diagnostic targets, with clear specifications on which variables are manipulated and which are controlled.

However, establishing such attributability is inherently difficult in executable manipulation tasks, where success depends on a joint interplay of language instructions, object states, spatial relations, and action constraints. Critically, as illustrated in Figure 1(a), varying a single factor, whether it be an instruction phrase, an object instance, a target position, or a layout configuration, can inadvertently shift multiple capability requirements simultaneously. This creates a fundamental tension: each capability dimension requires sufficient task variation to support generalization testing, yet introducing too much variation across tasks weakens within-suite comparability and invites cross-capability confounding. Conversely, keeping tasks too similar limits diagnostic coverage. Navigating this trade-off is a central benchmark-design challenge.

Challenge #1: Diagnostically Attributable Task Construction.

A diagnostic benchmark must construct executable task suites that vary a predefined capability-related factor while preserving task goals, success conditions, and non-target requirements as consistently as possible. The central challenge is to maintain sufficient within-suite diversity without introducing cross-capability confounding, so that observed performance differences can be reliably attributed to a specific generalization bottleneck.

B. Combinatorial Perturbation Design for VLA Robustness Evaluation

In real-world deployment, VLA policies operate within a closed loop of perception, decision, action, and state transition. A perturbation at any step, whether sensor noise, an unexpected object movement, or actuator deviation, can alter not only the immediate action but also subsequent environment states and future observations. For instance, an early perception bias may lead to misestimation of the target state, which in turn shifts the entire subsequent action trajectory. As a result, the impact of a perturbation can propagate and accumulate over the course of task execution. Open-loop evaluations on static inputs capture only the immediate, single-step reaction to a perturbation, missing these cascading effects. Robustness

therefore must be assessed within the full interactive loop, where propagation and accumulation can be observed.

Single-factor tests remain useful for identifying local sensitivities and calibrating perturbation intensities. However, they do not characterize model behavior when multiple disturbances jointly affect perception, spatial configuration, and control execution. We refer to these conditions as **combinatorial perturbations**. Benchmarking combinatorial perturbations alongside single-factor baselines exposes robustness limits that isolated tests often miss.

We face a central challenge in constructing a diagnostic suite of combinatorial perturbations: four interdependent dimensions must be simultaneously balanced — realism, intensity, confound control, and combinatorial tractability:

- **Realism.** As illustrated in Figure 1(b), perturbations in deployment originate from diverse sources, including sensor noise, environmental variations, and actuation errors, each with distinct parameter ranges. These perturbations must remain realistic while upholding the original task goal, motor requirements, and success criteria.
- **Appropriate intensity.** Perturbation magnitudes should strike a balance between being large enough to elicit sensitivity and small enough to preserve task feasibility and avoid floor effects.
- **Controlled confounding.** The task goal, motor requirements, and success conditions must remain consistent across perturbed and unperturbed settings to ensure that observed performance differences are attributable to the perturbations themselves.
- **Tractable combinatorial coverage.** The space of perturbation combinations must be compressed while preserving cross-category dependencies; otherwise, exhaustive enumeration becomes computationally prohibitive, and the results resist meaningful interpretation.

 **Challenge #2: Diagnostic Combinatorial Perturbation Design.** A robustness benchmark must select perturbation factors, calibrate their intensities, and construct multi-factor configurations that preserve task feasibility and task semantics. The central challenge is to prune the combinatorial space without discarding plausible cross-category dependencies, so that the selected configurations remain computationally tractable, diagnostically interpretable, and representative of real-world deployment.

IV. VLA-GAUGE FRAMEWORK

We introduce VLA-GAUGE, the first unified diagnostic framework for systematically evaluating zero-shot generalization and perturbation robustness in VLA models. Figure 2 illustrates its architecture, which comprises four integrated stages: ① **benchmark construction**, ② **zero-shot generalization evaluation**, ③ **robustness evaluation**, and ④ **diagnosis-driven model refinement**. The framework organizes base scenarios into capability-oriented suites with tailored ID and OOD settings. Evaluating models across these suites yields fine-grained capability profiles and ID-to-OOD generalization gaps, rather than a single aggregate success rate. These diagnostic insights then inform robustness testing under structured perturbations and guide targeted model refinement using verified successful demonstrations. All components are built upon a unified executable task representation that encodes

task semantics, execution conditions, and success criteria, enabling repeated closed-loop evaluations under capability-specific distribution shifts and diverse perturbation conditions.

Specifically, to address **Challenge #1**, VLA-GAUGE organizes 270 base scenarios into six capability-oriented task suites, each with tailored ID and OOD conditions. Within each suite, variants are constructed around a primary diagnostic factor, while keeping the task goal, required motor skills, success criteria, and non-target conditions as consistent as possible. This design yields capability-level performance distributions and ID-to-OOD generalization gaps, rather than collapsing generalization into a single overall success rate. The resulting scenario-level performance also serves as a calibration anchor for subsequent robustness evaluation.

To address **Challenge #2**, VLA-GAUGE compiles perturbation types and their value ranges that have been validated in existing studies [4]–[7], [12], [13], [29], [37]–[40], and organizes them according to their roles within the VLA execution pipeline. The resulting hierarchical framework encompasses three perturbation categories, namely visual information, spatial structure, and control execution, spanning 11 factors (e.g., camera, illumination, and background) and 33 perturbation attributes (e.g., orientation, distance, and position under camera perturbations). This framework captures the diverse and heterogeneous sources of perturbations present in real-world environments. Single-factor sensitivity analysis calibrates perturbation magnitudes within the feasible range of each task, and evaluates model sensitivity while keeping the task goal, motor requirements, and success criteria fixed. Guided by the analysis results, the screening of perturbation categories and factors retains a representative set of multi-factor combinations rather than exhaustively testing all possible permutations. The resulting configurations are then evaluated across all base scenarios and the full model set under closed-loop execution, thereby achieving controlled and reproducible coverage of diverse combinatorial perturbations.

Finally, VLA-GAUGE identifies challenging scenarios in which all 12 models achieve zero success under the initial diagnostic rollout configurations. For the zero-shot generalization and combinatorial perturbation robustness branches, we separately split the corresponding scenarios into training and test subsets that share no tasks or scenario configurations. Guided by the diagnostic results of each branch, we then apply data augmentation to the training subsets only. From the resulting training scenarios, we collect demonstrations verified as successful. The target model is then fine-tuned on these demonstrations and evaluated on the corresponding held-out test subset to determine whether the weaknesses revealed by diagnosis have been effectively mitigated. This process of evaluation, optimization, and re-evaluation forms a closed-loop diagnostic pipeline that integrates weakness identification, targeted fine-tuning, and independent validation.

A. Diagnostic Benchmark Construction

This diagnostic benchmark is built upon a unified executable task representation, enabling fine-grained capability attribution in VLA model evaluation. To faithfully replicate the diverse conditions encountered by robots in real-world environments, the benchmark encompasses three evaluation

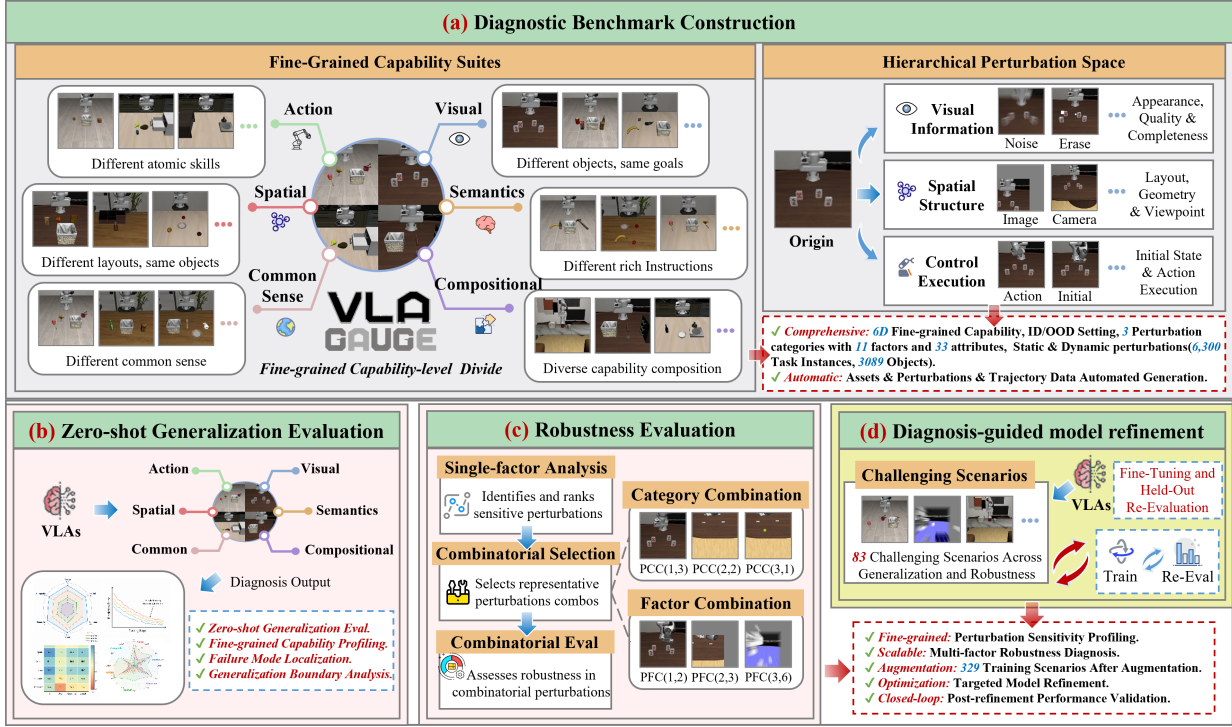


Fig. 2: Overview of the VLA-GAUGE Evaluation Framework

settings: clean-task, single-factor perturbation, and combinatorial perturbations. As illustrated in Figure 2(a), the construction proceeds along two parallel branches. The first branch, **fine-grained capability suites**, defines capability-specific in-distribution and out-of-distribution tasks while rigorously controlling non-target variables. The second branch, **hierarchical perturbation space**, generates controlled single-factor and combinatorial perturbations anchored to the same base tasks. Together, these two branches form a unified diagnostic space that redefines model assessment, shifting the focus from aggregate success-rate comparisons to capability attribution and perturbation robustness diagnostics.

Unified Executable Task Representation. Before constructing the two parallel branches described above, we first build a unified representation of their common foundation, namely the robot manipulation task. Following the general definitions established in previous work [3], [41], we represent each base task as *a quadruple* $\mathcal{T} = (\ell, \mathcal{O}, \mathcal{C}, \mathcal{E})$, corresponding to the ① *instruction*, ② *scenario layout*, ③ *success criteria*, and ④ *execution-condition configuration*, respectively, thereby enabling direct execution and repeated instantiation of every benchmark instance in a robot simulation environment:

- **Instruction.** A natural-language instruction describing the robot’s task objective.
- **Scenario layout.** It specifies the initial states and spatial relationships of manipulable, articulated, and fixed objects.
- **Success criteria.** These define the logical conditions used to determine whether the task has been successfully completed.
- **Execution-condition configuration.** It specifies environmental attributes (e.g., camera, light, and background) and robot attributes (e.g., gripper state and execution delay).

Although the Behavior Domain Definition Language (BDDL) [14] can describe task objects, initial states, and goal conditions to construct base tasks, it has *two limitations*: ①

Limited task representational scope, which does not fully encode execution conditions or environmental attributes; ② *Lack of unified task-variant modeling*, which cannot generate controlled variants from base tasks while preserving task semantics and success conditions. Therefore, we introduce the Variant-augmented Behavior Domain Definition Language (V-BDDL), which introduces an optional unified variant block (i.e., `:variant`). V-BDDL represents a task’s execution-condition configuration as $\mathcal{E} = \{\rho_i\}_{i=1}^n$, with each variable encoded as a structured perturbation item $\rho_i = (c_i, f_i, u_i, v_i, \mu_i)$, where

- **Category** (c_i) is determined by the primary stage of the VLA pipeline affected by the perturbation (e.g., perception, reasoning, or execution).
- **Factor** (f_i) denotes a specific perturbation source within a category (e.g., camera, noise, or image).
- **Attribute** (u_i) denotes a configurable attribute of a perturbation factor (e.g., orientation, distance, or position under the camera factor).
- **Parameter value** (v_i) specifies a perturbation attribute’s concrete setting (e.g., angular values from 2° to 10° for the yaw, pitch, and roll components of camera orientation).
- **Static and dynamic perturbations** (μ_i) specify how a perturbation item varies during task execution (e.g., static or dynamic).

An empty execution-condition configuration $\mathcal{E} = \emptyset$ denotes a clean base task, whereas configurations satisfying $|\{f_i\}| = 1$ or $|\{f_i\}| > 1$ instantiate single-factor or combinatorial perturbation variants, respectively. This design not only preserves BDDL’s original task representation capabilities but also unifies the modeling and instantiation of base and perturbed tasks, providing a unified executable task formalism for constructing fine-grained capability suites and a hierarchical perturbation space. Additionally, we convert assets from existing work [9],

[10], [15], [32], [42] into object definitions compatible with ROBOSUITE [43], forming a shared pool of 3,089 candidate objects. Building on these foundations, VLA-GAUGE automates asset configuration and the generation of perturbations and trajectories, yielding 6,300 task instances across subsequent evaluations. Below we elaborate the construction of the two benchmark branches.

A.1. Fine-Grained Capability Suite Construction.

Considering the processing demands of robotic manipulation, including visual, language, and action modalities, as well as the semantic and world knowledge acquired during pretraining, we categorize VLA model capabilities into six dimensions: ① *visual perception*, ② *semantic grounding*, ③ *common sense and world knowledge*, ④ *spatial reasoning*, ⑤ *action reasoning*, and ⑥ *compositional capability*. These dimensions systematically characterize the generalization of VLA models in closed-loop manipulation. Based on the unified executable task representation, we construct 270 clean base scenarios and organize them evenly into six fine-grained capability suites. This design uses the resulting performance profiles to attribute generalization failures to specific dimensions, thus revealing model weaknesses and informing subsequent diagnosis-driven refinement.

Each capability suite follows a *unified variable control principle*: ① **Variation**. Tasks are constructed around a single primary diagnostic variable, and only conditions directly related to this variable are modified. ② **Control**. Non-target variables, including task objectives, success criteria, and other manipulation requirements, are kept as consistent as possible. This principle reduces confounding capabilities and attribution ambiguity, allowing performance differences to better reflect the target capability.

Following the variable control principle above, the six capability suites are constructed as follows:

- **Visual perception**. This suite varies object category, shape, texture, and visually similar distractors to assess whether models can identify, localize, and distinguish task-relevant objects from visual observations.
- **Semantic grounding**. This suite varies linguistic cues (e.g., referring expressions, contextual descriptions, and implicit intents) to assess whether models can interpret diverse expressions and map them to executable goals and manipulation targets.
- **Common sense and world knowledge**. This suite varies required knowledge types (e.g., object functions, category relations, everyday uses, and conventions) to assess whether models can infer unstated targets or intents using knowledge acquired during pretraining.
- **Spatial reasoning**. This suite varies object positions, relative distances, directional relations, and reference object configurations to assess whether models can interpret scenario layouts and spatial relations to determine target objects and task constraints.
- **Action reasoning**. This suite varies atomic skill requirements (e.g., pick&place, open&close, twist, and stack) and local action dependencies, while limiting subgoals and sequence length, to assess whether models can select and sequence appropriate skills.

- **Compositional capability**. This suite varies subgoal count, skill combinations, and long-horizon dependencies across task stages to assess whether models can coordinate multiple skills across subgoals during multistep execution.

Compared with existing benchmarks with coupled capability requirements, this fine-grained decomposition ① more directly attributes generalization failures to individual capabilities and ② uses capability-specific clean-task baselines to distinguish underlying capability deficits from sensitivity to single and combinatorial perturbations.

Within each capability suite, we adopt the LIBERO training distribution shared by all evaluated models as the reference distribution, and build two evaluation settings on top of it.

- **In-distribution (ID)**. This setting preserves the task structures and object layout rules and adjusts the target diagnostic variable of the suite only within the value ranges and composition rules that this distribution already covers, so that it measures the zero-shot generalization of a model to unseen task instances drawn from the same distribution.
- **Out-of-distribution (OOD)**. This setting, again guided by the diagnostic objective of each suite, introduces task structures, object categories, scenario layouts, instruction forms, or task compositions that the reference distribution does not cover, so that it measures the zero-shot generalization of a model to unseen task instances under these novel conditions.

The six suites thus yield 12 subsuites for comparing ID performance and ID-to-OOD generalization gaps across capabilities. Consequently, VLA-GAUGE enables the identification of capability bottlenecks and the characterization of generalization decay across each capability from ID to OOD regimes, while maintaining a clean baseline without perturbations. This facilitates the downstream attribution of task failures to whether they stem from inherent capability limitations or from sensitivity to environmental perturbations.

A.2. Hierarchical Perturbation Space Construction.

In real-world deployment, we systematically screen perturbation factors that have been implemented and validated in existing studies of robot manipulation [29], [37], [38], [39], VLA evaluation [13], [6], [5], [4], [12], [7] and embodied navigation [40]. Each factor must satisfy the following *selection criteria*: ① it is reported in at least two studies; and ② it acts through a clearly identifiable visual, spatial, or control pathway. Based on the unified executable task representation, we encode each retained perturbation as a structured perturbation item $\rho_i \in \mathcal{E}$ and construct the hierarchical perturbation space of VLA-GAUGE. This space contains **3 perturbation categories, 11 perturbation factors, 33 perturbation attributes, and 268 validated parameter values, while supporting static and dynamic perturbations as well as single and combinatorial perturbation configurations**. Its hierarchical organization bounds the candidate space for subsequent combinatorial perturbation screening, reducing reliance on inefficient manual enumeration.

The perturbation space covers only execution-condition changes that preserve task semantics; linguistic variation and knowledge-based reasoning are assessed by the semantic grounding and common sense and world knowledge suites, respectively. Within this scope, we classify the perturbation

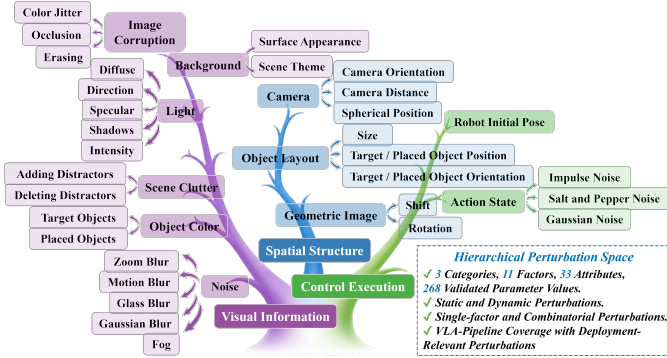


Fig. 3: Hierarchical Structure of the VLA-GAUGE Perturbation Space

items into three categories based on where they act in the closed loop of VLAs; their factors and perturbation attributes are shown in Figure 3.

- **Visual information perturbations.** These perturbations target the appearance, quality, or integrity of visual observations to assess perceptual robustness to visual variation or degradation.
- **Spatial structure perturbations.** These perturbations target object layouts, geometric relations, reference frames, or viewpoints to assess the robustness of a model’s spatial reasoning under changes in spatial configuration.
- **Control execution perturbations.** These perturbations target initial robot states or action execution to assess a model’s closed-loop control robustness to state shifts or execution uncertainty.

Based on the parameter ranges reported in these studies, we generate candidate values for each perturbation attribute. Continuous parameter ranges (e.g., camera distance) are divided into subintervals and sampled uniformly within, whereas discrete or categorical parameter ranges (e.g., target objects or background) are sampled randomly from their value sets. Each candidate value is then instantiated in ROBOSUITE and evaluated under *three validation criteria*: ① **Perturbation effectiveness**: the instantiated perturbation must alter the intended simulator, observation, or control property, with perceptible visual changes manually verified by comparing observations before and after injection. ② **Physical feasibility**: the perturbed task must initialize correctly, maintain valid robot and object states and stable simulation, and remain executable. ③ **Semantic consistency**: the perturbation must preserve task instruction, goal predicates, required skill type, and success criteria. Only candidate values satisfying all three criteria are retained, thereby controlling confounding so that performance changes remain attributable to the perturbations themselves.

We further define static and dynamic perturbations $\mu_i \in \{\text{static}, \text{dynamic}\}$ for each perturbation item ρ_i : ① **Static perturbation**. The value of the selected parameter is set at task initialization and remains fixed throughout execution. ② **Dynamic perturbation**. The parameter value of the corresponding attribute is updated over time during execution.

To apply the set of execution conditions $\mathcal{E}$ to a base task T , we define a perturbation operator $\mathcal{P}_{\mathcal{E}}$, which instantiates the perturbed task $\tilde{T} = \mathcal{P}_{\mathcal{E}}(T)$ via the V-BDDL (:variant) interface. The simulation parser implements this mapping by

normalizing each perturbation item $\rho_i \in \mathcal{E}$ and routing it to the corresponding simulation module according to its target. Thus, the hierarchical perturbation space converts heterogeneous real-world perturbations into reproducible perturbed tasks with controllable parameters. Compared with existing benchmarks that only evaluate perturbations in isolation, it supports systematic robustness diagnosis across perturbation categories and parameter settings, covering static and dynamic perturbations and their combinations.

B. Benchmark Evaluation Protocol

To date, no existing VLA benchmark has combined the evaluation of zero-shot generalization and robustness under combinatorial perturbations within a single framework. VLA-GAUGE addresses this gap through two complementary protocols: ① **Zero-shot generalization evaluation**, assessing how well the model transfers its learned capabilities from ID to OOD settings and diagnosing generalization failures via paired ID/OOD tasks across six capability suites. ② **Perturbation robustness evaluation**, quantifying sensitivity to single-factor perturbations and robustness under combinatorial perturbations against clean-task baselines. Across both protocols, model parameters remain frozen during evaluation. Below, we elaborate on both protocol implementations.

B.1. Zero-Shot Generalization Evaluation

As shown in Figure 2(b), VLA-GAUGE evaluates the zero-shot generalization performance of each model across the six capability dimensions and reports its ID, OOD, and overall zero-shot score.

$$ZS_y = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} SR_y(d), \quad y \in \{\text{ID}, \text{OOD}, \text{ALL}\} \quad (1)$$

Here, $\mathcal{D}$ denotes the set formed by the six capability dimensions, SR_y denotes the model’s mean success rate under distribution y , and $\text{ALL} = \text{ID} \cup \text{OOD}$. We evaluate each model with 50 closed-loop rollouts per scenario. All models share predefined initial states and random seeds and use the execution time limits specified by the corresponding suites. A rollout is successful if all goal predicates are satisfied within the time limit.

To reduce the cost of the subsequent perturbation robustness evaluation, we select the two models with the highest overall zero-shot score ZS_{ALL} to form a fixed reference model set $\mathcal{M}_{\text{ref}}$. Following LIBERO-PLUS [4], we classify a scenario as **Easy**, **Medium**, or **Hard** according to whether both reference models, only one of them, or neither reaches the success rate threshold of 0.6. Within each difficulty level of every capability dimension, we take the scenarios whose average success rate over the two reference models lies closest to the median of that group as anchors, and these anchors constitute the anchor scenario set $\mathcal{S}_{\text{anchor}}$ used for calibration in the subsequent perturbation robustness evaluation.

B.2. Perturbation Robustness Evaluation

As shown in Figure 2(c), VLA-GAUGE builds on the calibrated anchor scenarios and the hierarchical perturbation space, and carries out the evaluation in three successive stages: ① **Single-factor sensitivity analysis**, which calibrates perturbation strength and identifies the perturbation factors to which

a model is sensitive. ② **Combinatorial perturbation screening**, which chooses representative perturbation combinations at both the category level and the factor level. ③ **Combinatorial perturbations evaluation**, which quantifies model robustness when several perturbation factors act jointly. All three stages adopt the R-Score [29] to measure how well a model preserves its performance under perturbation.

Stage 1: Single-Factor Sensitivity Analysis.

During single-factor sensitivity analysis, VLA-GAUGE evaluates the validated parameter values $\mathcal{V}(u)$ for each attribute u in turn on $\mathcal{S}_{anchor}$ using $\mathcal{M}_{ref}$, leaving all other attributes at their default values. It selects two types of sensitive values, which serve as the basis for sensitivity scoring and for ranking attributes and factors: ① **the overall degradation value set** $\mathcal{V}_{deg}$ contains candidate values that yield the lowest mean R-Score across the reference models, capturing the greatest overall performance degradation; and ② **the model-discriminative value set** $\mathcal{V}_{disc}$ contains candidate values that maximize the gap between the reference models' mean R-Scores to distinguish their robustness responses.

$$\mathcal{V}_\eta(u) = \begin{cases} \arg \min_{v \in \mathcal{V}(u)} \bar{R}(u, v), & \eta = \text{deg}, \\ \arg \max_{v \in \mathcal{V}(u)} \Delta \bar{R}(u, v), & \eta = \text{disc}. \end{cases} \quad (2)$$

Here, $\bar{R}$ denotes the mean R-Score for the reference models on the fixed anchor set when attribute u is set to v . $\Delta \bar{R}$ denotes the absolute difference between the mean R-Score of the two reference models. All tied optimal values are retained, and the two sets of sensitive values are merged into $\mathcal{V}_{sens}$, which supports a unified sensitivity score and the subsequent **combinatorial perturbation screening**. For each attribute, the R-Scores associated with its calibrated sensitive value set are averaged to obtain the attribute score, and the attribute scores within the same factor are then averaged to obtain the factor score. **Lower scores indicate greater perturbation sensitivity**. These scores yield **three rankings** for the subsequent screening phase: ① attributes within each factor, ② factors within each category, and ③ factors globally across categories.

Stage 2: Combinatorial Perturbations Screening.

Building on the calibrated sensitive value sets and the ranking produced in Stage 1, combinatorial perturbation screening proceeds at the category level and then at the factor level. Both levels adopt the same attribute configuration: each perturbation factor contributes the perturbation attribute with the lowest score, one value is drawn at random from the calibrated sensitive value set of that attribute and then held fixed throughout the screening process, and all remaining attributes stay at their default settings.

Perturbation Category Screening. All perturbation category combinations (PCCs) formed from the three perturbation categories are enumerated. Within each combination, the lowest-scoring factor of every included category serves as the representative of that category, and the combination is instantiated with the attribute configuration described above. Candidate combinations that contain the same number of perturbation categories are then screened by the criterion given below, which yields the representative PCC set used for the subsequent factor-level screening.

Perturbation Factor Screening. Within each retained PCC,

the factor with the highest score, that is, the lowest sensitivity, is selected from every category to construct the initial perturbation factor combination (PFC). The remaining factors are then added one by one in descending order of factor score, which yields a nested sequence of candidates. The PFCs in each sequence are screened separately according to the criterion given below, and the retained combinations constitute the representative set of combinatorial perturbations in VLA-GAUGE.

For a candidate perturbation combination $\mathcal{E}$, the mean R-Score of reference model m over the fixed scenario set $\mathcal{S}_{anchor}$ is converted into a degradation score $D_{\mathcal{E}}(m) = \max(0, 1 - \bar{R}_{\mathcal{E}}(m)/100)$. A combination reaches the saturation boundary only when both reference models have a mean R-Score of zero, i.e., $D_{\mathcal{E}}(m) = 1$ for every $m \in \mathcal{M}_{ref}$. When a candidate combination is expanded, expansion stops as soon as this boundary is reached. Representative combinations are then selected using the criterion below. Following MMA [44], we further define, for unsaturated candidates, **the Harmonic Mean of the Degradation Scores (D-HM)**:

$$\text{D-HM}_{\mathcal{E}} = \frac{|\mathcal{M}_{ref}|}{\sum_{m \in \mathcal{M}_{ref}} (D_{\mathcal{E}}(m) + \epsilon)^{-1}} - \epsilon \quad (3)$$

Here, $\epsilon = 10^{-6}$. This metric is constrained by the smaller degradation score, so it favors combinations that cause substantial performance degradation for every reference model. Both screening levels retain the unsaturated candidate with the highest D-HM, and if all candidates under consideration are saturated, the candidate with the highest mean degradation score is retained instead. All tied optimal candidates are kept. **Stage 3: Combinatorial Perturbations Evaluation.**

VLA-GAUGE applies the representative PFCs selected in Stage 2 to all base scenarios and reports the R-Score of each model under every PFC, which quantifies its robustness to combinatorial perturbations. Each model runs 50 closed-loop trials on each task instance, and every trial selects a static or a dynamic perturbation with equal probability.

Under static perturbation, every attribute retains throughout the rollout the pre-initialized value determined in Stage 2. Under dynamic perturbation, each mutable attribute that has at least two calibrated sensitive values resamples a value from its own calibrated sensitive value set at steps 0, $\lfloor H_t/3 \rfloor$, and $\lfloor 2H_t/3 \rfloor$, where H_t denotes the maximum number of execution steps specified by the suite to which task t belongs. Attributes with only one calibrated sensitive value remain static throughout.

For a given PFC, base scenario, and rollout index, all models share the same perturbation mode (static or dynamic) and the same sequence of values. Sampling is independent across rollouts, and every perturbation configuration satisfies the semantic consistency requirements specified in A.2.

To quantify the overall performance degradation induced by each representative PFC, we build on RMSD [33] and define the **Degradation Root Mean Square Deviation (D-RMSD)**:

$$\text{D-RMSD}_{\mathcal{E}} = \sqrt{\frac{1}{|\mathcal{M}| |\mathcal{S}_{\mathcal{E}}|} \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{S}_{\mathcal{E}}} (\Delta_{\mathcal{E}}(m, t))^2} \quad (4)$$

Here, $\mathcal{M}$ denotes the full set of evaluated models, and $\mathcal{S}_{\mathcal{E}}$ denotes the set of task instances generated by applying

a representative PFC to the base scenarios. The quantity $\Delta_{\mathcal{E}}(m, t) = \max(0, \text{SR}_{\mathcal{O}}(m, t) - \text{SR}_{\mathcal{E}}(m, t))$ denotes the non-negative drop in success rate relative to the unperturbed condition. **A larger D -RMSD indicates a more severe overall performance degradation induced by that PFC.**

C. Diagnosis-Driven Model Refinement

Data Construction. We use the capability diagnosis results from the zero-shot generalization branch and the combinatorial perturbation robustness branch to determine which tasks, scenario configurations, and perturbation states the optimization data needs to cover. For each of the two evaluation branches, we collect all scenarios in which every evaluated model achieves zero success under the initial diagnostic rollout configurations and organize them by the six capability dimensions, which yields two groups of challenging scenarios. We then apply a fixed random seed to split each group into a training subset and a test subset that overlap in neither task nor scenario configuration, and we perform data augmentation on the training subset only.

In the zero-shot generalization branch, we retain the ID/OOD partition used in the evaluation and augment the challenging training scenarios to cover conditions associated with diagnosed capability weaknesses. Following the capability definitions in A.1, we construct variants using the following rules:

- **Visual perception.** We vary the target object’s shape, color, or surface texture while keeping the manipulation goal unchanged. For instance, in a cup placement task, we change the cup’s surface texture from a solid color to a striped pattern. If the instruction refers to the original texture, we update the corresponding description accordingly.
- **Spatial reasoning.** We move either the target object or its movable reference object at a time, translating it to the left, to the right, toward the robot arm, or away from the robot arm while keeping all other objects in place. We generate four variants by moving the target object and four by moving the reference object, yielding eight variants per original scene. When necessary, we update the spatial descriptions in the instruction accordingly.
- **Semantic grounding.** We rewrite instructions by adding contextual information, changing referring expressions, or conveying intent implicitly, while keeping the scene and task goal unchanged. For instance, we rewrite “Place the red cup in the tray” as “The red cup should be in the tray.”
- **Common sense and world knowledge.** We construct object selection tasks that require identifying the target object based on its function, category membership, or physical properties. For instance, in a scene containing a spoon, a pen, and a toy, we use the instruction “Place the object used to eat soup in the tray.”
- **Action reasoning.** We expand the range of manipulated object instances while keeping the atomic action type unchanged. For instance, in a drawer-opening task, we replace the drawer with another instance whose handle is in a different position.
- **Compositional capability.** We modify the scene configurations and instructions of existing compositional tasks to

create variants that require the joint use of multiple basic capabilities. For instance, in the task “Open the drawer and place the cup inside,” we position the target cup to the left of a bowl and revise the instruction to “Open the drawer and place the red cup located to the left of the bowl into the drawer.” The resulting task combines object recognition, spatial reasoning, and sequential manipulation.

In the combinatorial perturbation robustness branch, we select perturbation combinations based on the evaluation results. For each challenging training scenario, we use the parameter configuration retained during screening as the baseline. Each variant starts from this baseline configuration and changes only one attribute by randomly sampling a value different from the original from the corresponding subset of sensitive parameter values in B.2; all other attributes remain unchanged. For instance, given a combination of target object size, action noise, and image translation, we generate three new combinatorial perturbation configurations by changing each attribute once in a separate variant. We apply these configurations to the corresponding challenging training scenarios while keeping the task goals and success criteria unchanged.

Across the selected challenging training scenarios and their augmented counterparts, we collect the trajectories of the zero-shot generalization branch with a *SpaceMouse*, and those of the combinatorial perturbation robustness branch through automated replay with the VLA-GAUGE toolkit. The trajectories of a given task cover a variety of initial configurations and execution paths, which reduces the reliance of the model on a particular initial state or a fixed action pattern. After collection, task success is verified automatically against the goal predicate of each scenario, and trajectories that contain abnormal pauses, ineffective actions, or redundant paths are removed manually.

Refinement Protocol. VLA-GAUGE selects the model with the highest average performance on the zero-shot generalization and combinatorial perturbation robustness evaluations as the target model. The two optimization branches start from the same original checkpoint and train separate LoRA [45] adapters on the generalization training set and the robustness training set respectively. The training setup and the LoRA hyperparameters follow the official configuration of the target model. A checkpoint is saved every 10,000 steps during training, and the test sets serve only for the final evaluation. For held-out evaluation, the original and fine-tuned models use a new, shared set of initial states and random seeds, independent of those used for scenario selection.

V. EXPERIMENTS

We propose five research questions in our evaluation:

🔦 **RQ1 (Overall Zero-Shot Generalization Performance):** What is the overall zero-shot generalization performance of existing VLA models?

🔦 **RQ2 (Zero-Shot Performance Distribution and Capability Bottlenecks):** What zero-shot performance distribution and capability bottlenecks do existing VLA models exhibit across VLA-GAUGE’s six capability dimensions?

🔦 **RQ3 (Sensitivity to Single-Factor Perturbations):** How sensitive are representative VLA models to different single-factor perturbations in VLA-GAUGE?

🔦 **RQ4 (Robustness to Combinatorial Perturbations):** How robust are existing VLA models to combinatorial perturbations in VLA-GAUGE?

🔦 **RQ5 (Effectiveness of Diagnosis-Driven Fine-Tuning):** To what extent does VLA-GAUGE’s diagnosis-driven fine-tuning improve the target VLA model’s zero-shot generalization and combinatorial robustness?

A. Experiment Setup

1) **Data Composition.** VLA-GAUGE primarily includes data for three stages: zero-shot generalization evaluation, perturbation robustness evaluation, and diagnosis-driven fine-tuning. The zero-shot generalization dataset comprises 270 unperturbed base scenarios across six capability suites. Each suite contains 45 scenarios categorized as ID or OOD. We select one easy, one medium, and one hard scenario from each suite, yielding 18 anchor scenarios. The robustness evaluation sequentially tests 268 validated parameter values across all attributes, followed by 7 PCCs and then 15 PFCs. The 3 selected representative PFCs are then applied to all base scenarios, yielding 810 task instances under combinatorial perturbations. During diagnosis-driven fine-tuning, we first select 27 challenging scenarios for zero-shot generalization and 56 for robustness to combinatorial perturbations. Following the train-test isolation setting in [46], we partition the selected scenarios into training and test subsets at the scenario level, yielding 21 training and 6 test scenarios for the generalization branch and 45 training and 11 test scenarios for the robustness branch before augmentation. We collect 50 successful trajectories per training scenario, resulting in a total of 3,300 training demonstrations across the two branches before augmentation. We then apply diagnosis-driven data augmentation only to the training subsets. The augmentation methods and the final sizes of the training sets are detailed in V-F.

2) **Models and Metrics.** We assess the zero-shot generalization and perturbation robustness of 12 VLA models using VLA-GAUGE: ① **five large models**, *UniVLA* [22], *OpenVLA-OFT* [27], *RIPT-VLA* [28], *SimpleVLA-RL* [26] and *CronusVLA* [29]; ② **seven small models**, $\pi_{0.5}$ [30], *Nora* [23], *GR00T1.5* [47], *DeepThinkVLA* [25], *X-VLA* [48], *Evo-1* [49] and *VLA-Adapter-Pro* [50]. These models span discrete, continuous, and hybrid action paradigms, as detailed in II-A.

We use five metrics: ① **%Success Rate (SR)**: the average binary success measure across all rollouts of each task; ② **%Zero-Shot Score (ZS)**: the mean performance across ID and OOD tasks within each suite, with both metrics assessing zero-shot generalization. ③ **R-Score (R)**: performance retention relative to the unperturbed condition, used throughout robustness evaluation; ④ **D-HM**: cross-model consistency of performance degradation under combinatorial perturbations, used for perturbation screening. ⑤ **%D-RMSD**: the overall degradation induced by combinatorial perturbations, used to select diagnosis-driven fine-tuning data. We evaluate fine-tuning with the corresponding evaluation-branch metrics.

3) **Evaluation Configuration.** The evaluation follows the protocol in IV-B. For the initial zero-shot generalization and perturbation robustness evaluations, we use the officially released model checkpoints without additional training or adaptation. For each scenario, all models share predefined

TABLE II: Zero-shot generalization performance of VLA models on VLA-GAUGE.

Vis., Sem., CW, Spa., Act., and Comp. denote visual perception, semantic grounding, common sense and world knowledge, spatial reasoning, action reasoning, and compositional capability, respectively. **ZS** and **Average** denote per-model means across capabilities and per-subset means across models, respectively. All scores are percentages. The **top-1** and **top-2** results are highlighted, respectively.

Models	Params	Generalization Capability						ZS
		Vis.	Sem.	CW.	Spa.	Act.	Comp.	
Large Models								
UniVLA [22]	9B	29.3	18.7	23.6	10.2	35.9	23.6	23.6
OpenVLA-OFT [27]	7B	38.2	10.2	36.4	16.9	22.7	21.3	24.3
RIPT-VLA [28]	7B	30.2	6.7	28.4	5.3	15.3	23.1	18.2
SimpleVLA-RL [26]	7B	11.1	2.2	8.0	2.2	13.6	11.1	8.0
CronusVLA [29]	7B	24.4	28.9	29.3	8.4	18.2	22.7	22.0
Small Models								
π 0.5 [30]	3B	60.0	57.8	61.8	47.6	72.7	41.8	57.0
Nora [23]	3B	11.1	9.3	8.4	2.2	6.8	1.8	6.6
GR00T1.5 [47]	3B	15.1	16.0	17.3	4.9	14.5	17.3	14.2
DeepThinkVLA [25]	2.9B	39.1	39.6	37.8	21.8	33.2	29.8	33.6
X-VLA [48]	0.9B	17.3	14.7	9.8	22.7	37.3	35.6	22.9
Evo-1 [49]	0.77B	30.2	28.0	30.7	8.0	28.2	23.6	24.8
VLA-Adapter-Pro [50]	0.5B	16.4	14.7	7.6	6.2	10.5	19.6	12.5
Average	–	26.9	20.6	24.9	13.0	25.7	22.6	22.3

initial states and random seeds and perform 50 closed-loop rollouts. Each suite’s rollout horizon is set to 1.1 times the average human completion steps following *LIBERO-X* [6].

Following the model-selection criterion in IV-C, we selected $\pi_{0.5}$ as the fine-tuning target and trained separate LoRA adapters for the generalization and robustness branches. Training was conducted on eight NVIDIA A100 80GB GPUs. The training configuration included: ① *Action horizon*: 10; ② *Batch size*: 4; ③ *Training duration*: 5 epochs; ④ *Learning rate*: $5e-5$ initial rate with cosine decay; ⑤ *Warmup ratio*: 3%; ⑥ *LoRA ranks*: 16 and 32 for the backbone and action expert, respectively.

B. 🔦 RQ1: Overall Zero-Shot Generalization Performance

Methodology. We comprehensively assess the overall zero-shot generalization and cross-distribution performance gaps of VLA models. Following the unified evaluation protocol in B.1, we keep model parameters frozen and evaluate 12 VLA models under in-distribution (ID) and out-of-distribution (OOD) conditions across the six capability suites of VLA-GAUGE, and report the overall, ID, and OOD **%Zero-Shot Score (ZS)** for each model. Under the ID condition, VLA-GAUGE preserves the task structures, object layout rules, and instruction forms of the reference training distribution, and constructs unseen task instances within the existing value ranges and composition rules according to the evaluation requirements of the six capabilities. Under the OOD condition, VLA-GAUGE introduces task structures, object categories, scenario layouts, instruction forms, or task compositions that the reference training distribution does not cover. The detailed settings of VLA-GAUGE appear in A.1. We also group models into large and small categories based on parameter count and compare their overall, ID, and OOD scores to examine the relationship between model size and zero-shot generalization performance on VLA-GAUGE.

Results. Table II shows that the 12 VLA models achieve overall zero-shot scores of 6.6% to 57.0% on VLA-GAUGE, with a mean of only 22.3%. $\pi_{0.5}$ ranks first at 57.0%, while the remaining 11 models score no higher than 33.6%. *SimpleVLA-RL* achieves a zero-shot score of only 8.0% on VLA-GAUGE, ranking second to last among the evaluated models.

TABLE III: In-distribution and out-of-distribution zero-shot generalization performance across VLA models.

ID/OOD: in-distribution/out-of-distribution evaluation settings; The *Average1.* and *Average2.* denote per-model means across capabilities and per-subset means across models. The *top-1* and *top-2* results are highlighted, respectively. Near-zero ID/OOD scores appear in *red*. The overall results and the results specific to the ID/OOD settings are obtained from separate evaluation runs.

Models	Vis.		Sem.		CW.		Spa.		Act.		Comp.		Average1.	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
Large Models														
UniVLA [22]	47.0	10.9	32.2	4.5	40.9	5.5	16.4	2.4	56.2	6.7	30.3	0.0	37.2	5.0
OpenVLA-OFT [27]	71.3	3.6	16.5	3.6	67.8	3.6	22.9	8.2	17.7	0.0	27.4	0.0	37.3	3.2
RIPT-VLA [28]	58.3	0.9	13.0	0.0	53.9	1.8	9.3	0.0	20.8	0.0	29.7	0.0	30.8	0.5
SimpleVLA-RL [26]	25.4	2.8	3.5	0.0	16.7	2.7	1.5	0.0	15.4	0.0	11.5	0.0	12.3	0.9
CronusVLA [29]	41.7	4.5	53.0	3.6	53.0	3.6	12.9	0.0	30.8	0.0	28.0	0.0	36.6	2.0
Small Models														
$\pi 0.5$ [30]	71.3	49.1	67.0	42.7	67.8	46.4	47.9	47.1	75.4	65.6	55.4	22.0	64.1	45.5
Nora [23]	21.7	0.0	17.4	0.1	16.5	0.0	3.6	0.0	11.5	0.0	2.3	0.0	12.2	0.0
GR00T1.5 [47]	30.4	0.0	35.7	0.0	33.9	0.0	7.9	0.0	25.4	0.0	20.0	0.0	25.6	0.0
DeepThinkVLA [25]	57.4	20.0	60.9	17.3	55.7	19.1	25.7	16.5	45.4	15.6	38.3	0.0	47.2	14.8
X-VLA [48]	33.9	0.0	29.6	5.5	21.7	2.7	20.7	22.4	56.2	6.7	42.3	10.0	34.1	7.9
Evo-1 [49]	50.4	7.3	40.9	9.1	43.5	9.1	12.1	4.7	43.8	6.7	26.9	2.0	36.3	6.5
VLA-Adapter-Pro [50]	31.3	0.9	28.7	0.0	14.8	0.0	10.7	0.0	17.7	0.0	25.1	0.0	21.4	0.2
<i>Average2.</i>	45.0	8.3	33.2	7.2	40.5	7.9	16.0	8.4	34.7	8.4	28.1	2.8	32.9	7.2

Finding 1: The overall zero-shot generalization of the evaluated VLA models remains limited on the tasks covered by VLA-GAUGE. These results suggest that insufficient generalization may hinder reliable deployment in unseen environments.

Table III further compares the evaluated models’ zero-shot performance under ID and OOD conditions in VLA-GAUGE. Across the six capability suites, the 12 models achieve a mean ID score of only 32.9% on unseen task instances. Under OOD conditions, the mean score drops further to 7.2%, which is $\downarrow 25.7$ percentage points below ID score. In the 72 model-capability evaluations, 71 yield lower OOD scores than the corresponding ID scores, and 32 yield OOD scores of 0.

Finding 2: Even when the task structures, layout rules, and instruction forms of the reference training distribution are preserved, the evaluated models still obtain low average ID zero-shot scores on the unseen task instances of VLA-GAUGE, and their scores drop further under the OOD condition. This highlights the need to assess generalization to unseen task instances within the reference distribution as well as under distribution shifts.

As shown in Figure 4 and Table II, the 3B-parameter $\pi 0.5$ ranks first with an overall zero-shot score of 57.0%, outperforming all five models with 7B to 9B parameters. Table III further shows that $\pi 0.5$ achieves the highest mean scores under both ID and OOD conditions. One possible explanation is its joint training on heterogeneous data and its hybrid design that combines discrete representation learning with continuous action generation. Therefore, to improve zero-shot generalization, future work should consider broader task and scenario coverage in training data and more effective mechanisms for integrating visual, language, and action information, rather than relying solely on parameter scaling.

Finding 3: Among the 12 models evaluated on VLA-GAUGE, the zero-shot score does not improve consistently with parameter scale, so parameter scale alone is insufficient to explain the differences in zero-shot generalization across models.

C. π RQ2: Zero-Shot Performance Distribution and Capability Bottlenecks

Methodology. To examine how generalization performance varies across the six capability dimensions of VLA-GAUGE

and identify capability bottlenecks, we use the unified evaluation results from V-B and the six capability suites defined in A.1. For each dimension, we compute the mean overall, ID, and OOD zero-shot scores across the 12 models and count the models with an OOD score of zero. We define the dimension with the lowest mean overall score as the overall capability bottleneck, and the dimension with both the lowest mean OOD score and the most models scoring zero under OOD conditions as the OOD capability bottleneck. Both bottlenecks represent performance weaknesses relative to the other capability dimensions within this evaluation. We measure performance degradation under OOD conditions as the mean ID score minus the mean OOD score for each dimension. As described in A.1, the three suites for visual perception, semantic grounding, and common sense and world knowledge share scenario layouts and differ only in task instructions. We compare their mean overall, ID, and OOD scores to examine performance differences under different instruction requirements.

Results. Table II and Figure 4(a)–(b) show clear differences in mean zero-shot performance across the six capability dimensions for the 12 models. Spatial reasoning has a mean zero-shot score of only 13.0%, the lowest among the six dimensions and at least 7.6 percentage points below that of any other dimension. For 8 of the 12 models, spatial reasoning is the lowest-scoring dimension or ties for the lowest score.

As shown in Table III and Figure 4(c)–(d), spatial reasoning also has the lowest mean ID score among the six dimensions, only 16.0%. This weakness may be related to most evaluated models’ primary reliance on 2D RGB inputs and limited coverage of spatial relations and layout variations in the training data. Future work may prioritize improving spatial reasoning under ID conditions by drawing on existing methods for integrating depth maps and point clouds and expanding training data coverage of target object positions, directional descriptions, and variations in reference objects. These changes can be evaluated to test whether 3D information and spatial training data improve zero-shot spatial reasoning.

Finding 4: Spatial reasoning constitutes the overall capability bottleneck of the evaluated VLA models, and this weakness is already evident under the ID condition.

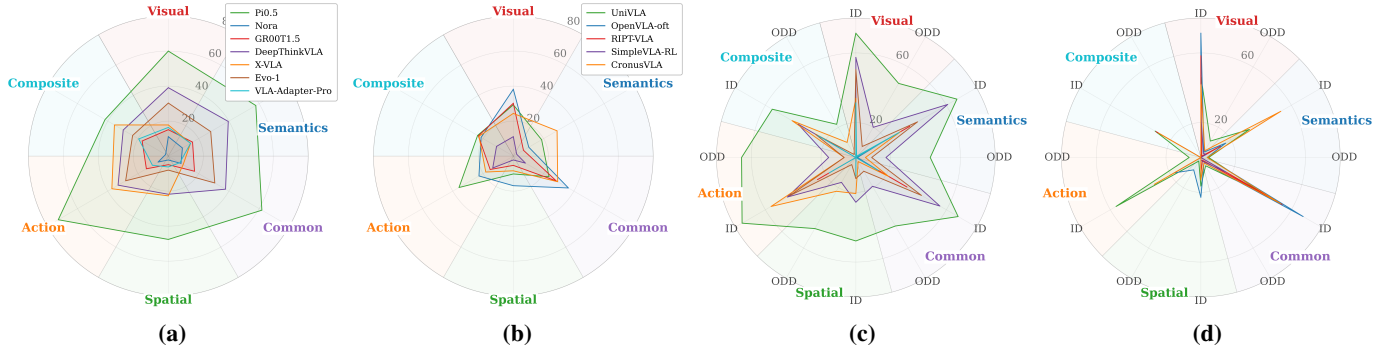


Fig. 4: Radar distributions of VLA models across six capability dimensions. (a) Small-scale models. (b) Large-scale models. (c) ID/OOD performance of small-scale models. (d) ID/OOD performance of large-scale models.

With all other task conditions held fixed, the models perform poorly on spatial tasks that involve changes in the position of the target object, in the directional description within the instruction, and in the reference object.

Under OOD conditions, compositional capability has the lowest mean score, only 2.8%. 9 of the 12 models score zero in this dimension, the largest number across the six dimensions. In comparison, visual perception, semantic grounding, common sense and world knowledge, and action reasoning have higher mean OOD scores, but all remain in the single digits and are well below their corresponding ID scores. We further observe that common sense and world knowledge achieves a mean score close to that of visual perception under both ID and OOD conditions, whereas semantic grounding scores lower than both.

These patterns may reflect insufficient compositional task examples in the evaluated models’ training data, along with limited coverage of long or implicitly phrased instructions and subtask execution. Future work may prioritize adding compositional task examples that cover different capability combinations and strengthening training on subtask execution. It can also strengthen training in understanding long or implicitly phrased instructions and add examples that require common sense and world knowledge to interpret task requirements and select appropriate actions, with the aim of improving zero-shot performance under OOD conditions. In the diagnosis-driven fine-tuning experiments in V-F, we further test whether this data augmentation approach improves zero-shot generalization.

Finding 5: Compositional capability constitutes the out-of-distribution capability bottleneck of the evaluated VLA models, whose OOD zero-shot performance is particularly weak on multi-step tasks that require coordinating several subgoals, combining different manipulation skills, and handling long-horizon dependencies across task stages. Under the OOD condition, the models also show limited understanding of instructions that contain long descriptions or implicit expressions, together with limited ability to execute the corresponding subtasks.

D. 🟡 RQ3: Sensitivity to Single-Factor Perturbations

Methodology. Following the selection rules in B.1, we use the zero-shot generalization results from V-B and V-C to select the reference model set $M_{ref} = \{\pi_{0.5}, \text{DeepThinkVLA}\}$ and an 18-scenario anchor scenario set S_{anchor} , respectively. Under the single-factor evaluation protocol in B.2, we examine

33 perturbation attributes associated with 11 perturbation factors across three categories, covering 268 validated non-default parameter settings. We apply each setting to all 18 anchor scenarios, changing only the corresponding perturbation attribute while keeping all other attributes at their default values. This yields 4,824 perturbed tasks, each evaluated with both reference models.

We used each model’s unperturbed performance in the same scenario as the baseline and quantified performance retention under perturbation using *R-Score* (R). Lower R-Score values indicate greater sensitivity to the perturbation. Using this metric, we compare the magnitude of performance changes across perturbation attributes for the two reference models and examine how their performance varies with parameter settings.

Results. We select 6 of the 33 perturbation attributes for focused analysis, as shown in Figure 5. These attributes cover all three perturbation categories and clearly illustrate the magnitude of performance degradation, performance trends across parameter settings, and differences between the reference models.

As shown in Figure 5(a) and 5(b), perturbations to the robot’s initial state and action noise markedly affect the performance of both reference models. For robot initial state, the mean R-Score across the two models ranges from 13.6 to 89.0 depending on the parameter setting. With a 475° offset in the robot’s initial state, both models reach their respective minimum R-Score. *DeepThinkVLA*’s R-Score drops to zero, indicating greater sensitivity under this setting. For action noise, the mean R-Score ranges from 30.4 to 90.9 across parameter settings. As with initial state perturbations, some action noise settings cause substantial performance degradation.

Finding 6: The sensitivity of the two reference models to changes in the robot initial state and to action noise varies markedly with the parameter setting, and some settings cause severe performance degradation. These results suggest that developers deploying VLA models should pay particular attention to how the robot initial state is configured, and should limit the adverse effect of action noise on task execution.

For image shift, Figure 5(c) shows that the mean R-Score across the two reference models decreases overall from 102.5 to 64.2 as the shift magnitude increases, despite occasional increases. For target object scale, Figure 5(d) shows that both models maintain high R-Score at scale factors of 0.8 and 0.9. As the scale factor increases from 0.9 to 1.2, the mean

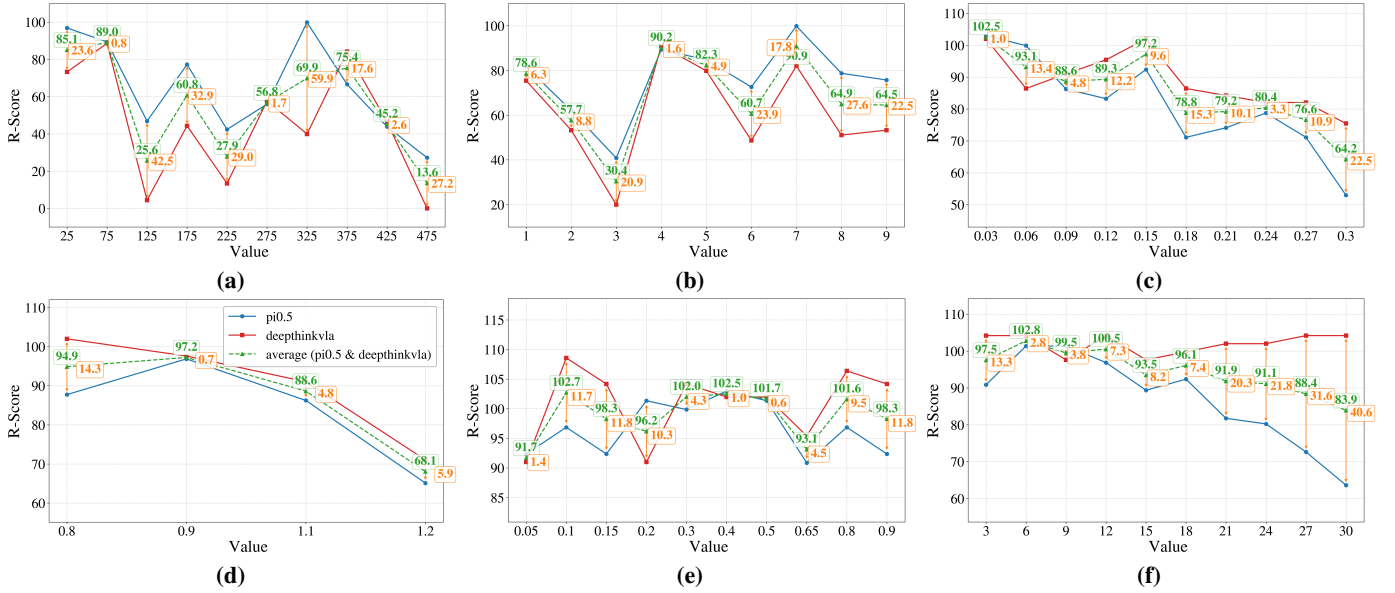


Fig. 5: Sensitivity of the two reference models to six selected perturbation attributes. (a) Robot initial state. (b) Action noise. (c) Image shift. (d) Target object scale. (e) Light diffuse. (f) Image rotation.

R-Score decreases steadily, reaching a minimum of 68.1. One possible reason is that image shifts move some task-relevant information outside the model’s field of view, leaving incomplete visual information, whereas enlarging the target object changes its apparent size and may make perception and grasping more difficult. In contrast, the mean R-Score for light diffuse remains between 91.7 and 102.7, as shown in Figure 5(e).

Finding 7: Within the tested parameter ranges, the two reference models are more sensitive to large image shifts and to enlargement of the target object, whereas changes in diffuse illumination affect them relatively little. Future work can explore strengthening contextual reasoning under incomplete visual observations and combining it with multi-scale visual augmentation, so as to improve the adaptability of a model to image shifts and to variations in target scale.

Figure 5(f) shows that as the image rotation angle increases from 18° to 30° , $\pi 0.5$ ’s R-Score continues to decrease, whereas DeepThinkVLA’s remains high with a slight overall increase. At 30° , the R-Score gap between the two models reaches 40.6, its maximum within the tested range.

Finding 8: The two reference models can differ markedly in their sensitivity to the same image rotation setting. Within the tested range of large rotation angles, $\pi 0.5$ is considerably more sensitive to image rotation than DeepThinkVLA. This suggests that model developers should pay close attention to how image rotation in the visual input affects model performance.

E. 🌱 RQ4: Robustness to Combinatorial Perturbations

Methodology. To comprehensively evaluate the robustness of existing VLA models to combinatorial perturbations, we sequentially conducted combinatorial perturbation screening and robustness evaluation.

Combinatorial Perturbation Screening. We follow the Stage 2 rules in B.2 to screen perturbation combinations. Using the V-D evaluation results, we identify sensitive value sets for each perturbation attribute and rank perturbation factors

and their attributes by their impact on reference model performance. We first construct and evaluate 7 candidate perturbation category combinations (PCCs), then construct and evaluate 15 candidate perturbation factor combinations (PFCs) within the selected category combinations.

In $PCC(m, n_1)$ and $PFC(m, n_2)$, m denotes the number of perturbation categories used, n_1 and n_2 denote the category combination index and factor combination index, respectively, for that category count. For example, $PCC(1, 1)$ uses only visual information perturbations. $PFC(2, 2)$ combines spatial structure and control execution perturbations, instantiated through Target Object Scale, Action Noise, and Image Shift.

We use *D-HM* to assess whether a candidate combination causes pronounced performance degradation in all reference models. Both screening stages use the anchor scenario set S_{anchor} and reference model set M_{ref} and follow the same rule. After excluding candidates for which all reference models have an R-Score of zero, we select the combination with the highest D-HM separately among candidates involving one, two, or three perturbation categories. The selected representative PFCs are used for subsequent evaluation of robustness to combinatorial perturbations, with their configurations held fixed throughout evaluation.

Combinatorial Perturbation Robustness Evaluation. Following the Stage 3 evaluation protocol in B.2, we apply each of the three selected representative PFCs to all 270 unperturbed base scenarios, yielding 810 task instances with combinatorial perturbations. We evaluate all 12 VLA models on these instances. Using each model’s unperturbed performance as the baseline, we use *R-Score* (*R*) to measure its performance retention under combinatorial perturbations. We also use %*D-RMSD* to quantify the overall performance degradation caused by each perturbation combination.

Results. As shown in Figure 6(a), $PCC(1, 3)$ and $PCC(2, 2)$ have the highest D-HM among candidates using one and two perturbation categories, respectively. $PCC(3, 1)$ is the only candidate using three categories. None of these combinations

TABLE IV: Overall robustness under representative PFCs across VLA models.

Average1. and *Average2.* denote per-model means across representative PFCs and column-wise means across models, respectively. *D-RMSD* is computed from the original PFC success rates relative to ZS. The *top-1* and *top-2* results are highlighted, respectively. Near-zero R-Scores appear in red.

Models	Params	PFC(1,2)	PFC(2,3)	PFC(3,6)	Average1.
Large Models					
UniVLA [22]	9B	9.3	8.9	0.8	6.4
OpenVLA-OFT [27]	7B	10.3	9.9	2.9	7.7
RIPT-VLA [28]	7B	8.2	4.4	1.6	4.8
SimpleVLA-RL [26]	7B	7.5	2.5	1.3	3.8
CronusVLA [29]	7B	17.3	3.2	1.8	7.4
Small Models					
$\pi 0.5$ [30]	3B	43.0	17.7	4.9	21.9
Nora [23]	3B	10.6	10.6	1.5	7.6
GR00T1.5 [47]	3B	14.1	17.6	7.7	13.1
DeepThinkVLA [25]	2.9B	12.8	13.4	3.9	10.0
X-VLA [48]	0.9B	52.4	3.5	0.9	18.9
Evo-1 [49]	0.77B	6.5	5.2	4.4	5.4
VLA-Adapter-Pro [50]	0.5B	6.4	4.8	3.2	4.8
<i>Average2.</i>	–	16.5	8.5	2.9	9.3
<i>D-RMSD</i>	–	19.3	22.7	24.8	–

yields zero R-Score for both reference models, so all three are retained under the screening rules. Under PCC(1,3), which contains only control execution perturbations, $\pi 0.5$ and *DeepThinkVLA* achieve R-Score of only 34.8 and 15.5, respectively. Both models have even lower R-Score under PCC(2,2) and PCC(3,1). These results show that both reference models struggle to maintain their unperturbed performance under these representative PCCs in the anchor scenarios.

Figure 6(b) further presents the evaluation results for the PFC candidates within the three representative PCCs. PFC(2,4) and PFC(3,9) are excluded because each yields zero R-Score for both reference models. Among the remaining candidates, PFC(1,2), PFC(2,3), and PFC(3,6) have the highest D-HM for combinations involving one, two, and three perturbation categories, respectively. We retain these three PFCs for subsequent evaluation of robustness to combinatorial perturbations.

PFC(1,2) includes Action noise and Robot initial state; PFC(2,3) includes Target object scale, action noise, image shift, and camera Vertical; PFC(3,6) includes Target object scale, background, action noise, image occlusions, light direction, image shift, placed object color, and noise. Both reference models have low R-Score under all three selected PFCs, indicating that these combinations cause marked performance degradation in both models.

As shown in Table IV, the evaluated models achieve an overall mean R-Score of only 9.3. In the 36 evaluation results, the two highest R-Scores are 52.4 and 43.0, while the remaining 34 are not higher than 17.7. Across PFC(1,2), PFC(2,3), and PFC(3,6), the mean R-Score decreases and D-RMSD increases at each step.

Finding 9: The evaluated VLA models exhibit insufficient overall robustness under the three representative combinatorial perturbations, and the overall performance loss becomes progressively more severe as the selected combinations move from a single perturbation category to three categories. VLA-GAUGE organizes its representative combinations by the number of perturbation categories involved, which makes it possible to assess the joint effect of multiple perturbation types under controlled conditions and thereby provides a basis for identifying potential performance problems under combinatorial perturbations before real deployment.

We also find that model robustness rankings vary across representative perturbation combinations. *X-VLA* achieves the highest R-Score under PFC(1,2), but scores only 3.5 and 0.9 under PFC(2,3) and PFC(3,6), respectively. In contrast, *GR00T1.5* achieves an R-Score close to that of the leading model $\pi 0.5$ under PFC(2,3) and ranks first under PFC(3,6) with an R-Score of 7.7.

Finding 10: None of the evaluated models ranks first under all three groups of representative PFCs. Therefore, assessing the combinatorial perturbation robustness of a VLA model requires considering its performance across different compositions jointly.

Figure 7 shows that the mean R-Scores across the 12 VLA models are generally low across the six capability dimensions under the three PFCs. Spatial reasoning and action reasoning have higher mean R-Score than the other four dimensions under all three PFCs, whereas compositional capability has the lowest mean R-Score under PFC(2,3) and PFC(3,6). One possible explanation is that the evaluated models’ training data do not adequately cover diverse combinations of perturbation settings. Future work may use multiple parameter settings for each factor in a perturbation combination to construct training examples covering diverse perturbation states. This strategy aims to improve robustness to combinatorial perturbations across all six capability dimensions, with particular emphasis on compositional capability.

Finding 11: All six capability dimensions of the evaluated models degrade under combinatorial perturbations. Relatively strong performance on an individual capability does not imply comparably strong combinatorial perturbation robustness on compositional tasks that require several capabilities to be exercised together.

F. 🧠 RQ5: Effectiveness of diagnosis-driven Fine-Tuning

Methodology. We follow the diagnosis-driven optimization protocol in IV-C to test whether VLA-GAUGE’s fine-grained diagnostics help construct effective training data. We select $\pi 0.5$ as the target model because it achieves the highest average performance in the zero-shot generalization and combinatorial perturbation robustness evaluations. Based on the results in V-E, we select PFC(2,3), which has the intermediate D-RMSD value among the three representative PFCs, to optimize and evaluate robustness to combinatorial perturbations.

In the zero-shot generalization branch, we construct training data following the augmentation rules in IV-C, guided by the diagnostic results and optimization recommendations from V-B and V-C. The specific settings are as follows:

- To address the shortfall in spatial reasoning performance under ID conditions, we start from 4 challenging ID training scenarios and sample translation distances uniformly from 1 to 5 cm. We generate 8 new variants per original scenario, yielding 32 additional training scenarios in total.
- To address the performance shortfall in compositional capabilities under OOD conditions, we start from 5 challenging OOD training scenarios. For each original scenario, we construct compositional tasks involving 2, 3, and 4 basic capabilities, adding one new scenario per category and yielding 15 additional training scenarios in total.
- To address the overall shortfall in zero-shot generalization across the remaining four capability dimensions, we

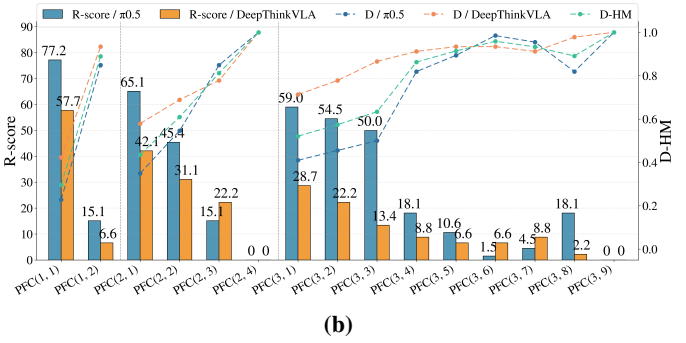
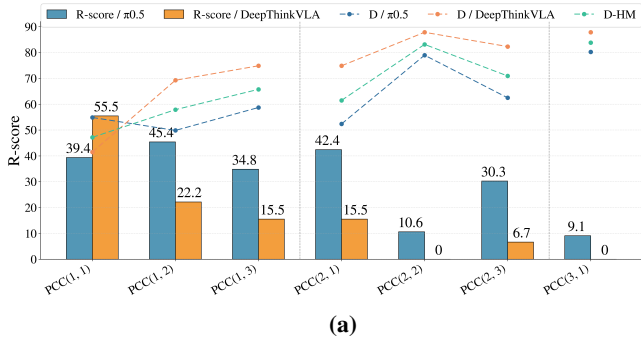


Fig. 6: Results of combinatorial perturbation selection. (a) perturbation category combination selection. (b) perturbation factor combination selection.

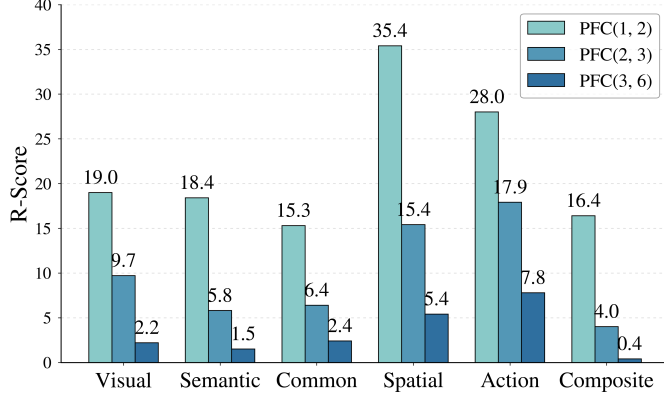


Fig. 7: Capability robustness under representative combinatorial perturbations.

augment the 12 existing challenging training scenarios. For semantic grounding, visual perception, and common sense and world knowledge, we generate one variant per original scenario using each of the three augmentation methods defined for the corresponding capability. For action reasoning, we add 3 scenarios per original scenario, each using a different instance of the manipulated object. This yields 12 additional scenarios from the 4 semantic grounding scenarios and 24 additional scenarios from the 8 scenarios covering the other three capabilities.

In the combinatorial perturbation robustness branch, we address the insufficient robustness of the evaluated models across all six capability dimensions revealed by V-E. Taking the PFC(2,3) parameter configuration retained during screening as the baseline, we follow the augmentation rules in IV-C and randomly sample, from the candidate set of each of its four perturbation attributes, one replacement value distinct from the original. Replacing one attribute value at a time thus yields four new combinatorial perturbation states. Applying each of these four states to the 45 existing challenging training scenarios yields 180 new training scenarios, giving 225 scenarios in total together with the original ones.

Both optimization branches follow the collection and quality verification procedure described in IV-C, recording multi-view visual observations, end-effector states, and action sequences synchronously at 20 Hz. Each training scenario contributes 50 successful trajectories, so the two types of optimization data together comprise 16,450 successful trajectories. We compare the zero-shot generalization score and the combinatorial perturbation robustness score under PFC(2,3) before and after fine-tuning $\pi_{0.5}$. For the robustness branch, we additionally

report fine-tuning on the single-factor perturbation data of LIBERO-PLUS as a reference, a dataset that contains more successful trajectories than the training set of our robustness branch.

Results. As shown in Figure 8(a), our method increases the target model’s mean zero-shot generalization score by $\uparrow 72.4$ points and its mean robustness score under PFC(2,3) by $\uparrow 31.8$ points. In comparison, fine-tuning on LIBERO-PLUS single-factor perturbation data improves the mean robustness score by only $\uparrow 10.9$ points.

Finding 12: The optimization data constructed under the diagnosis-driven procedure of VLA-GAUGE effectively improves both the zero-shot generalization of the target model and its combinatorial perturbation robustness under PFC(2,3). Notably, compared with the single-factor fine-tuning baseline, our approach achieves a larger robustness gain under the PFC(2,3) combinatorial perturbation condition while using fewer successful training trajectories.

Figure 8(b) and Figure 8(c) show marked score gains across all six capability dimensions in both branches. In the zero-shot generalization branch, spatial reasoning, compositional capability, and semantic grounding are key optimization targets, with score gains of $\uparrow 83.0$, $\uparrow 32.4$, and $\uparrow 78.0$ points, respectively. The PFC(2,3) robustness branch achieves gains of $\uparrow 26.1$ to $\uparrow 45.2$ points across the six dimensions.

Finding 13: The optimization data constructed under the diagnosis-driven procedure of VLA-GAUGE improves the capabilities on which the target model performs relatively weakly, in both the zero-shot generalization branch and the PFC(2,3) robustness branch, which demonstrates the practical value of fine-grained diagnosis for guiding model optimization.

VI. DISCUSSIONS

Data Leakage: Because the full pretraining data of the 12 evaluated VLA models cannot be thoroughly verified, we cannot entirely rule out the possibility that their pretraining content overlaps with the evaluation tasks in VLA-GAUGE. To reduce the risk of data leakage in our evaluation procedure, we use the officially released model checkpoints without additional training or adaptation and keep all model parameters frozen during the initial evaluations. In the diagnosis-driven fine-tuning stage, we split the challenging scenarios of each of the two evaluation branches into training and test sets that share no tasks or scene configurations, and we perform data augmentation and construct the optimization data using only the training scenarios.

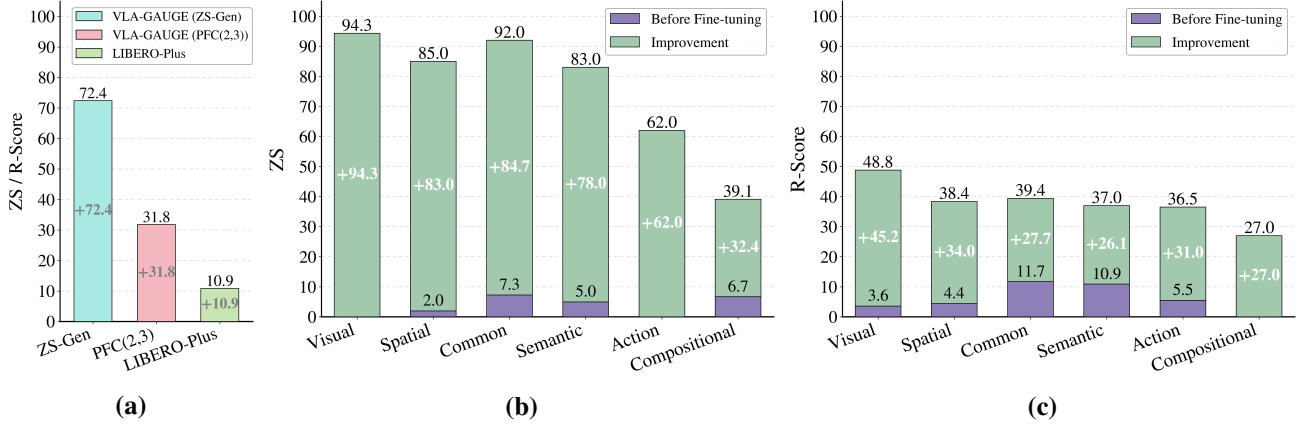


Fig. 8: Diagnosis-driven fine-tuning gains in zero-shot generalization and perturbation robustness. (a) Overall comparison of the fine-tuning gains of VLA-GAUGE on zero-shot generalization and on PFC(2,3), together with the robustness gain from fine-tuning on LIBERO-PLUS single-factor perturbation data. (b) Zero-shot generalization gains across capabilities. (c) Robustness gains under PFC(2,3) across capabilities.

Threats to Validity: One potential threat is that VLA-GAUGE is built on the LIBERO simulation environment, so its evaluation scope may not fully cover other robot embodiments or the diverse conditions of real-world deployment. To broaden the coverage of the simulated evaluation, we construct a hierarchical perturbation space that spans three types of perturbations, namely visual information, spatial structure, and control execution perturbations, and we evaluate 12 VLA models on 270 base scenarios covering six capability dimensions, together with their perturbed instances. Future work can extend the evaluation to other robot embodiments and to real-robot tasks, which would test whether the capability gaps and perturbation sensitivities observed here still hold under these conditions.

VII. CONCLUSION

We present VLA-GAUGE, the first unified diagnostic framework for systematically assessing the zero-shot generalization and perturbation robustness of VLA models. The framework integrates six fine-grained capability suites with paired in-distribution (ID) and out-of-distribution (OOD) splits and a hierarchical perturbation space, and it closes the optimization loop through diagnosis-driven data construction and fine-tuning. Our large-scale empirical study of 12 VLA models shows that high reported in-distribution success rates do not guarantee strong zero-shot generalization to unseen tasks or robustness to combinatorial perturbations. The diagnosis-driven fine-tuning experiments further show that fine-grained diagnostic results, together with the optimization suggestions derived from them, effectively guide the construction of training data and improve both the zero-shot generalization and combinatorial perturbation robustness of the target model. We therefore urge the community to give these two capabilities greater weight in the evaluation and development of VLA models.

REFERENCES

- [1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.
- [2] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv preprint arXiv:2303.03378*, 2023.
- [3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [4] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei *et al.*, “Libero-plus: A progressive robustness benchmark for vision-language-action models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 38 574–38 583.
- [5] X. Zhou, Y. Xu, G. Tie, Y. Chen, G. Zhang, D. Chu, P. Zhou, and L. Sun, “Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization,” *arXiv preprint arXiv:2510.03827*, 2025.
- [6] G. Wang, C. Zhang, Q. Liu, J. Zhang, J. Cai, J. Liu, and X. Liu, “Libero-x: Robustness litmus for vision-language-action models,” *arXiv preprint arXiv:2602.06556*, 2026.
- [7] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, “The colosseum: A benchmark for evaluating generalization for robotic manipulation,” *arXiv preprint arXiv:2402.08191*, 2024.
- [8] H. Sawada and S. Kasahara, “Beyond single-axis testing: Paired evaluation of compound robustness in vision-language-action policies,” *arXiv preprint arXiv:2609.15940*, 2026.
- [9] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, “Evaluating real-world robot manipulation policies in simulation,” *arXiv preprint arXiv:2405.05941*, 2024.
- [10] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang *et al.*, “Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 11 142–11 152.
- [11] J. Zhou, K. Ye, J. Liu, T. Ma, Z. Wang, R. Qiu, K.-Y. Lin, Z. Zhao, and J. Liang, “Exploring the limits of vision-language-action manipulation in cross-task generalization,” *Advances in Neural Information Processing Systems*, vol. 38, pp. 139 899–139 927, 2026.
- [12] Z. Wang, Z. Zhou, J. Song, Y. Huang, Z. Shu, and L. Ma, “Vlatest: Testing and evaluating vision-language-action models for robotic manipulation,” *Proceedings of the ACM on Software Engineering*, vol. 2, no. FSE, pp. 1615–1638, 2025.
- [13] B. Zhang, J. Li, J. Shen, Y. Cai, Y. Zhang, Y. Chen, J. Dai, J. Ji, and Y. Yang, “Vla-arena: An open-source framework for benchmarking vision-language-action models,” *arXiv preprint arXiv:2512.22539*, 2025.
- [14] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.
- [15] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, “Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 80–93.

- [16] X. Li, M. Zhang, Y. Geng, H. Geng, Y. Long, Y. Shen, R. Zhang, J. Liu, and H. Dong, "Manipllm: Embodied multimodal large language model for object-centric robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18061–18070.
- [17] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu *et al.*, "Vision-language foundation models as effective robot imitators," in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 26 703–26 721.
- [18] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, " π_0 : A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.
- [19] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.
- [20] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11975–11 986.
- [21] Z. Li, Z. Li, K. Zhou, and J. Gu, "Soma: Strategic orchestration and memory-augmented system for vision-language-action model robustness via in-context adaptation," *arXiv preprint arXiv:2603.24060*, 2026.
- [22] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, "Univla: Learning to act anywhere with task-centric latent actions," *arXiv preprint arXiv:2505.06111*, 2025.
- [23] C.-Y. Hung, Q. Sun, P. Hong, A. Zadeh, C. Li, U. Tan, N. Majumder, S. Poria *et al.*, "Nora: A small open-sourced generalist vision language action model for embodied tasks," *arXiv preprint arXiv:2504.19854*, 2025.
- [24] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.
- [25] C. Yin, Y. Lin, W. Xu, S. Tam, X. Zeng, Z. Liu, and Z. Yin, "Deep-thinkvla: Enhancing reasoning capability of vision-language-action models," *arXiv preprint arXiv:2511.15669*, 2025.
- [26] H. Li, Y. Zuo, J. Yu, Y. Zhang, Z. Yang, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui *et al.*, "Simplevla-rl: Scaling vla training via reinforcement learning," *arXiv preprint arXiv:2509.09674*, 2025.
- [27] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.
- [28] S. Tan, K. Dou, Y. Zhao, and P. Krähenbühl, "Interactive post-training for vision-language-action models," *arXiv preprint arXiv:2505.17016*, 2025.
- [29] H. Li, S. Yang, Y. Chen, X. Chen, X. Yang, Y. Tian, H. Wang, T. Wang, D. Lin, F. Zhao *et al.*, "Cronusvla: Towards efficient and robust manipulation via multi-frame vision-language-action modeling," *arXiv preprint arXiv:2506.19816*, 2025.
- [30] P. Intelligence, K. Black, N. Brown, J. Darphinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, " $\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.
- [31] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.
- [32] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.
- [33] M. Sedlacek, P. Yefanov, G. Ponimatkin, J. Bardhan, S. Pilc, M. Fourmy, E. Kazakos, C. G. Snoek, J. Sivic, and V. Petrik, "Realm: A real-to-sim validated benchmark for generalization in robotic manipulation," *IEEE Robotics and Automation Letters*, 2026.
- [34] R. Garcia, S. Chen, and C. Schmid, "Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8996–9002.
- [35] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.
- [36] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.
- [37] H. Zhang, S. Zhang, J. Jin, Q. Zeng, R. Li, and D. Wang, "Robustvla: Robustness-aware reinforcement post-training for vision-language-action models," *arXiv preprint arXiv:2511.01331*, 2025.
- [38] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3153–3160.
- [39] E. Xing, A. Gupta, S. Powers, and V. Dean, "Kitchenshift: Evaluating zero-shot generalization of imitation-based policy learning under domain shifts," in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.
- [40] P. Chattopadhyay, J. Hoffman, R. Mottaghi, and A. Kembhavi, "Robustnav: Towards benchmarking robustness in embodied navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 691–15 700.
- [41] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.
- [42] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.
- [43] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.
- [44] L. Yang, R.-Y. Zhang, Y. Wang, and X. Xie, "Mma: Multi-modal adapter for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 23 826–23 837.
- [45] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
- [46] S. Toyer, R. Shah, A. Critch, and S. Russell, "The magical benchmark for robust imitation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 284–18 295, 2020.
- [47] J. Björck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.
- [48] J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng *et al.*, "X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model," *arXiv preprint arXiv:2510.10274*, 2025.
- [49] T. Lin, Y. Zhong, Y. Du, J. Zhang, J. Liu, Y. Chen, E. Gu, Z. Liu, H. Cai, Y. Zou *et al.*, "Evo-1: Lightweight vision-language-action model with preserved semantic alignment," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026, pp. 13 397–13 406.
- [50] Y. Wang, P. Ding, L. Li, C. Cui, Z. Ge, X. Tong, W. Song, H. Zhao, W. Zhao, P. Hou *et al.*, "Vla-adapter: An effective paradigm for tiny-scale vision-language-action model," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 40, no. 22, 2026, pp. 18 638–18 646.