Systematic Evaluation of Modular Robotic Manipulation Policies via Structured Condition Space: A Study on Precision Pick-and-Place Tasks

Anonymous

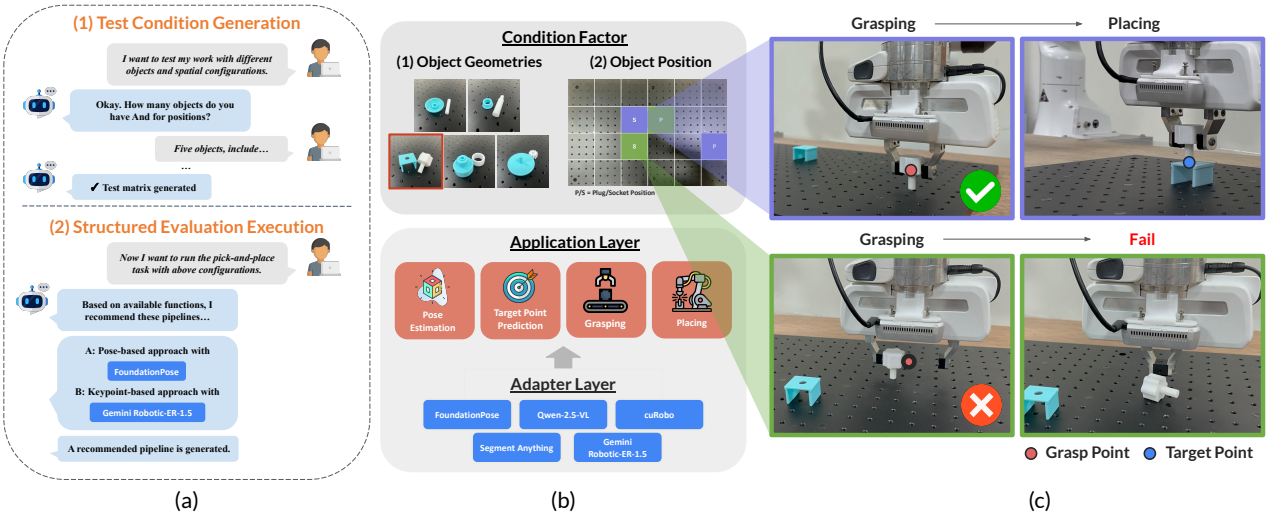

Fig. 1: **Overview of the Proposed Systematic Evaluation Framework for Robotic Manipulation Policies.** (a1) A user interacts with an LLM agent to construct structured, parameterized condition subspaces from high-level evaluation factors. (a2) The agent further compiles robotic manipulation policies within the unified framework. (b) Example condition factors (e.g., object geometry and object position) are shown at the top, while the bottom illustrates the modular architecture comprising an application layer and an adapter layer for policy compilation. (c) The resulting structured condition space enables extensive evaluation and systematic reliability boundary identification. Supplementary video is available at this link[1]

***Abstract*— Foundation models have demonstrated strong potential for robotic manipulation, promising adaptability across diverse tasks and environments. Despite favorable benchmark performance, these systems often exhibit degraded or unstable behavior under variations in object geometry, spatial configuration, sensing conditions, and workspace constraints—challenges that are amplified in real-world deployment. Existing evaluation efforts broaden evaluation across multiple perturbation axes but primarily focus on condition-level sensitivity. This offers limited insight into the precise configurations under which policies become unreliable. We propose a systematic evaluation framework that explicitly constructs a structured condition space with LLM assistance. Leveraging the semantic and commonsense priors encoded in large language models, we decompose high-level evaluation factors into structured, parameterized subspaces, enabling scalable exploration of environmental variations. This design shifts evaluation from coarse condition-level analysis to structured reliability boundary identification. We further introduce a modular architecture that compiles robotic manipulation policies within this unified framework and supports execution analysis across diverse conditions. Experimental results on precision pick-and-place tasks demonstrate that enables fine-grained characterization of performance degradation and failure patterns, providing actionable insights for robustness assessment and real-world deployment.**

I. INTRODUCTION

Robustness remains a central challenge for deploying robotic manipulation systems in real-world settings. Despite promising benchmark results, foundation models—including large perception models [1], [2], [3], Vision-Language models (VLMs) [4], [5], [6], [7], and Vision-Language-Action (VLA) models [8], [9], [10], [11], [12]—often exhibit degraded or unstable behavior when exposed to variations in object geometry, spatial configuration, sensing viewpoint, or workspace constraints [13], [14]. Understanding where and how these systems fail under structured real-world conditions is a necessary step toward improving their robustness.

Recent efforts have sought to scale and standardize real-world evaluation of general-purpose policies [15], [16], [17]. However, these works primarily assess condition-level sensitivity—such as object pose variation or semantic rephrasing—without characterizing how performance degrades within each factor. A less explored but equally critical question is under which configurations a policy becomes

[1]Video link: `https://tinyurl.com/robot-eval-video`

unreliable, as this finer-grained characterization provides actionable guidance for improving robustness in deployment.

To this end, we propose a systematic evaluation framework that explicitly constructs a structured condition space with LLM assistance, transforming evaluation from coarse condition-level sensitivity analysis to fine-grained reliability boundary identification. Rather than manually enumerating perturbations, we leverage LLMs to decompose high-level condition factors into structured, parameterized subspaces, enabling scalable exploration of policy reliability boundaries.

Our main contributions are as follows:

1) We propose a systematic evaluation framework based on structured condition-space construction, which organizes operating conditions into explicit, factorized test matrices, enabling fine-grained characterization of performance variations across individual condition factors.
2) We design a three-layer modular architecture with interchangeable adapters, enabling controlled comparisons across pipeline configurations under identical operating conditions.
3) Through extensive real-world evaluation on precision pick-and-place tasks, we uncover structured reliability boundaries and provide actionable insights for improving the robustness of manipulation pipelines.

II. RELATED WORK

A. Real-World Robot Evaluation

The rise of general-purpose manipulation policies has created a growing demand for scalable and standardized evaluation in real-world settings. RoboArena [15] addresses this by proposing a decentralized protocol that aggregates evaluation results across multiple laboratories, enabling large-scale policy comparison without centralizing experimental infrastructure. AutoEval [16] complements this by introducing an automated platform that standardizes execution and environment reset procedures, reducing the human effort required for repeatable real-world testing. STAR-Gen [17] further advances this direction by introducing a structured taxonomy that organizes environmental variations into visual, semantic, and behavioral axes, providing a principled basis for reasoning about what conditions to test. While these works significantly expand the breadth, scalability, and structure of real-world evaluation, they primarily focus on comparing policy performance across diverse conditions. An equally important but less addressed question is how performance varies systematically along a specific condition factor. Our framework addresses this by explicitly constructing a structured condition space, enabling fine-grained characterization of reliability boundaries within each factor.

B. High Precision Manipulation

Precision assembly—such as peg insertion, gear meshing, and plug-socket mating—is a fundamental challenge in industrial robotics, requiring accurate perception and tight geometric tolerances. The NIST Assembly Task Board [18] establishes a standardized benchmark for evaluating such tasks. Following this benchmark, a line of work, Factory [19], [20], [21], progressively advances from simulation to real-world robotic execution on NIST-inspired assets. AutoMate [22] further broadens the scope by introducing 100 diverse assembly geometries, enabling the study of precision assembly policies over a significantly wider range of part configurations.

III. SYSTEMATIC EVALUATION FRAMEWORK

The proposed framework operates across two phases: **(1) condition space construction** and **(2) structured evaluation execution**. In the first phase, the user specifies high-level evaluation goals, and an LLM generates concrete condition instances. In the second phase, a task-specific modular pipeline is composed and executed under each condition, producing stage-level execution signals and task-level outcomes for subsequent analysis.

A. Modular Architecture and Pipeline Execution

We adopt a three-layer modular architecture that separates decision-making, functional abstraction, and model integration. The Orchestration Layer, powered by a large language model (LLM), dynamically composes a task-specific manipulation pipeline and manages data flow between stages. The Application Layer defines functional modules that can be composed into pipelines, while the Adapter Layer provides standardized wrappers around diverse third-party models, enabling a plug-and-play mechanism for model substitution without modifying pipeline logic. To support systematic evaluation across condition instances, manipulation functions are equipped with built-in checkers that monitor execution status and return a binary pass/fail signal. If a stage fails, the pipeline triggers an early exit. In addition, recorders can be attached to stages to capture relevant data—such as RGB-D images, point clouds, and robot states—after execution. These mechanisms enable precise failure localization and structured data logging across different test conditions.

B. Structured Condition-Space Construction

Evaluation planning is formulated as the structured construction of a test matrix over explicitly defined condition factors, where each factor represents a dimension of variation that may influence system performance, such as scene configuration, viewpoint, or background. Each factor is associated with a finite set of discrete values or sampling ranges. The Cartesian combination of selected factor values defines the trial configuration space. An LLM assists in transforming factor definitions into concrete condition instances—for example, discretizing workspace position into predefined spatial regions or enumerating object geometry variants—enabling scalable and reproducible exploration of configuration boundaries.

For each trial, the physical environment is configured according to the corresponding condition values and reset between trials. The composed manipulation pipeline is then executed autonomously, with stage-level checkers monitoring

execution status. Execution outcomes are automatically associated with the corresponding condition in the test matrix, ensuring traceable logging per condition.

IV. Case Study: Modular Pipelines for Precision Pick-and-Place

We conduct a case study on precision plug-and-socket assembly to validate the proposed framework on a real-world manipulation task requiring accurate perception under varying object geometries and spatial configurations. The subject of evaluation in this study is not an end-to-end learned manipulation policy, but two modular, training-free pipelines for precision assembly that share identical orchestration logic and manipulation functions while differing only in their perception component, enabling controlled comparison of perception robustness within a consistent execution backbone.

A. Task Definition

The precision pick-and-place task requires the robot to grasp a plug and transport it to an insertion-ready pose above the corresponding socket, achieving accurate spatial alignment without physical insertion. At the start of each trial, both objects are placed at random positions in a stable, upright orientation on the workspace surface. Successful execution requires reliable localization of both objects, stable grasp execution, and accurate placement. The experimental setup is described in **Appendix A**.

B. Pipeline Configurations

We adopt two pipeline configurations that share identical orchestration logic, manipulation functions, and execution sequence (plug localization, socket localization, grasping, and placing), differing only in their perception adapters: a pose-based approach (Pipeline A) using SAM2 [23] and Foundation Pose [3] for coarse-to-fine 6D pose estimation, and a keypoint-based approach (Pipeline B) using Gemini Robotics-ER 1.5 [7] for direct 2D keypoint prediction. Details are provided in the **Appendix B**.

C. Test Matrix Construction

The test matrix defines the experimental conditions under which the two pipelines are evaluated, organized along two primary condition factors: **object geometry** (difference in plug-socket form factors) and **spatial configuration** (placement of the plug and socket within the tabletop workspace). For object geometry, five pairs of plug-socket assemblies are selected from the AutoMate dataset [22] to represent diverse geometric characteristics, differing in overall size, symmetry, surface structure, and grasp affordances (Fig. 2).

For spatial configuration, the tabletop workspace is discretized into a grid and partitioned into three distance-based zones (D1–D3) according to Euclidean distance from the board center (Fig. S5). The resulting test matrix spans 5 object geometries and 20 spatial position pairs per object, yielding 100 unique condition instances, enabling systematic characterization of robustness across diverse operating conditions.

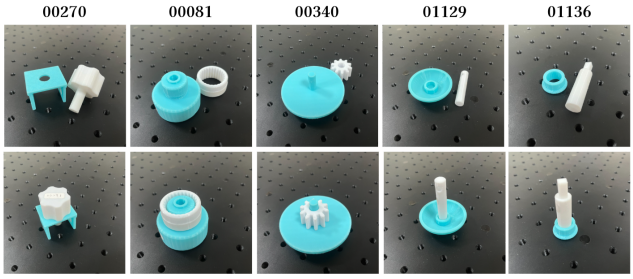

Fig. 2: 3D-printed plug-socket assemblies selected from the AutoMate dataset [22] in this study.

V. Evaluation Results

A. Metrics Definition

We define four metrics to evaluate pipeline performance. Since the plug–socket interfaces are rotationally symmetric, all metrics are defined in terms of 3D positional deviation. **Plug Position Error** and **Socket Position Error** measure the 3D Euclidean distance (mm) between the predicted and ground-truth positions, quantifying perception accuracy at the localization stage. **Grasp Success Rate** is the proportion of trials in which a stable grasp is achieved, determined by gripper width after closure. **Final Execution Error** measures the 3D Euclidean distance (mm) between the plug's final placed position and the target pose, annotated within the same coordinate frame, eliminating the cross-sensor bias. Human annotators review the recorded results after trial completion. Errors exceeding 100mm are excluded as outliers, and final execution error is computed only for successful grasps (Pipeline A: 459/500; Pipeline B: 232/500).

B. Per-Object Results (Geometry Effects)

Across all five objects, Pipeline A achieves consistently lower mean positional errors than Pipeline B (e.g., plug position error: 12.33 mm vs. 23.13 mm) and a substantially lower Final Execution Error (7.66 mm vs. 17.29 mm), with significantly smaller standard deviations indicating greater stability (see **Appendix C** for detailed results). Object geometry directly influences failure modes: thin and slender objects (#01129, #01136) leave little margin for localization error, resulting in low grasp success rates in Pipeline B (36% and 37%), while flatter objects (#00081, #00340) increase the probability of displacement during grasping even under accurate localization in Pipeline A. These geometry-specific failure patterns demonstrate that structured condition-space evaluation can reveal reliability boundaries inaccessible to condition-level analysis.

C. Per-Position Results (Spatial Effect)

Pipeline A achieves uniformly high grasp success rates across most spatial configurations, with the exception of the rightmost column (X=5), where success rates drop to 64% despite no corresponding increase in plug position error, suggesting that robot kinematic constraints rather than perception errors are the primary failure source. This insight

TABLE I: Success Rate Across Object IDs

Object ID	#00081	#00271	#00340	#01129	#01136	**Overall**
Pipeline A	82%	99%	81%	99%	98%	91.8%
Pipeline B	51%	62%	46%	36%	37%	45.3%

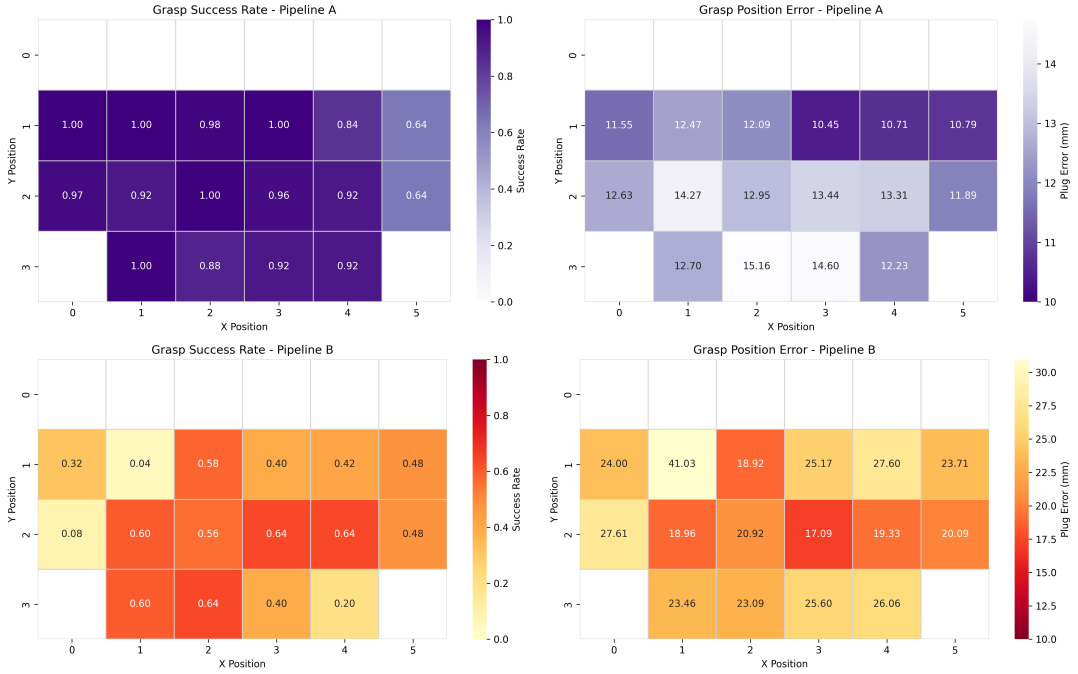

Fig. 3: Spatial performance heatmaps for Pipeline A (top) and B (bottom), showing grasp success rate (left) and the plug position error (right) across the 4x6 grid.

is only discoverable by jointly analyzing perception error and task success across spatial conditions: when perception error is acceptable but success rate remains poor, kinematic constraints should be investigated first.

For Pipeline B, regions with lower position error consistently correspond to higher grasp success rates, indicating that perception accuracy directly drives task success. Spatial performance heatmaps for both pipelines are shown in Fig. 3. The comparison across distance categories (D1–D3) in Figure S6 shows that Pipeline A maintains consistently high accuracy across all distance levels, while Pipeline B exhibits progressive degradation as distance increases, reflecting the limitation of its single-stage perception strategy.

D. Observations for Real-world Deployment

Based on experiments across two pipelines and five object types, we highlight the following observations to inform real-world deployment: (1) **Pipeline selection by geometry:** Pipeline A is recommended for tasks requiring high localization accuracy; Pipeline B exhibits larger localization error and may not be suitable for precision-critical tasks involving thin or slender objects. (2) **Spatial zone restriction:** Pipeline B may benefit from being restricted to near-to-mid distance zones (D1–D2), or augmented with an active approach mechanism to compensate for distance-induced accuracy loss. (3) **Failure diagnosis:** When perception error is acceptable but grasp success rate remains low, kinematic constraints may be a primary failure source worth investigating. (4) **Resource trade-offs:** Pipeline A requires local GPU resources but offers higher precision and spatial robustness; Pipeline B operates via cloud-based API with no local GPU requirement, offering a lighter deployment path at the cost of reduced accuracy.

VI. Conclusion

We present a systematic evaluation framework based on structured condition-space construction, enabling fine-grained characterization of where and how manipulation pipelines become unreliable across individual condition factors, and allowing practitioners to identify reliability boundaries and derive actionable improvement directions. In future work, extending LLM involvement beyond condition-space construction to the analysis stage would enable automated interpretation of evaluation results and generation of deployment guidelines, completing the evaluation loop. Combined with the extensibility of the layered architecture, we hope this work can serve as a foundation for systematic robustness evaluation across a wider range of real-world manipulation tasks.

APPENDIX

A. Experimental Setup

Our experimental platform consists of a 7-DoF Franka Emika Panda robot with the default gripper, a wrist-mounted RealSense D435 RGB-D camera for conducting experiments, and a fixed Orbbec Femto Bolt RGB-D camera for recording. 3D-printed assemblies from the AutoMate dataset [22] are placed on a 400 × 600 mm optical breadboard serving as the tabletop workspace, as shown in Fig. S4. To enable systematic spatial variation, the workspace is partitioned into three distance-based zones according to radial distance from the board center. These predefined regions provide a structured basis for sampling plug and socket positions during subsequent test matrix construction.

B. Pipeline Configurations

Both pipelines share identical orchestration logic and execution sequence: plug localization, socket localization, grasping, and placing. The grasp pose is derived from a canonical grasp defined in the AutoMate [22] dataset. The only difference lies in the perception adapters used to localize the plug and socket.

Pipeline A: Pose-based Approach. Extending the perception pipeline introduced in AutoMate [22], this pipeline formulates object localization as full 6D pose estimation using a coarse-to-fine strategy: Qwen2.5-VL [6] prov ides coarse semantic localization, followed by SAM2 [23] segmentation and FoundationPose [3] for 6D pose estimation. The estimated poses are converted into grasp and placement targets for motion planning.

Pipeline B: Keypoint-based Approach. Inspired by VLM-based keypoint prediction applications in robotic manipulation [24], [25], [26], this pipeline formulates perception as manipulation-oriented keypoint localization. Gemini Robotics-ER 1.5 [7] directly predicts 2D keypoints from an RGB image, which are unprojected into 3D coordinates using depth and refined along the Z axis using the CAD model and estimated table height. The resulting 3D keypoints are converted into grasp and placement targets.

TABLE II: Comparison of the two perception pipelines.

	Pose-based	Keypoint-based
Localization Output	6D pose (SE(3))	3D keypoint
Estimation Process	Two-stage (coarse-to-fine)	Single-stage
Semantic Segmentation	Required	Not required
CAD Model Usage	For pose estimation	For Z refinement

C. Detailed Evaluation Results

Tables III–V provide detailed per-object quantitative results across all metrics for both pipelines. Note that systematic cross-sensor bias may be introduced in Plug and Socket Position Errors, as predicted positions are obtained from the wrist-mounted RealSense D435 while ground-truth is annotated using the Orbbec Femto; this does not affect cross-pipeline comparison since both pipelines share identical camera configurations.

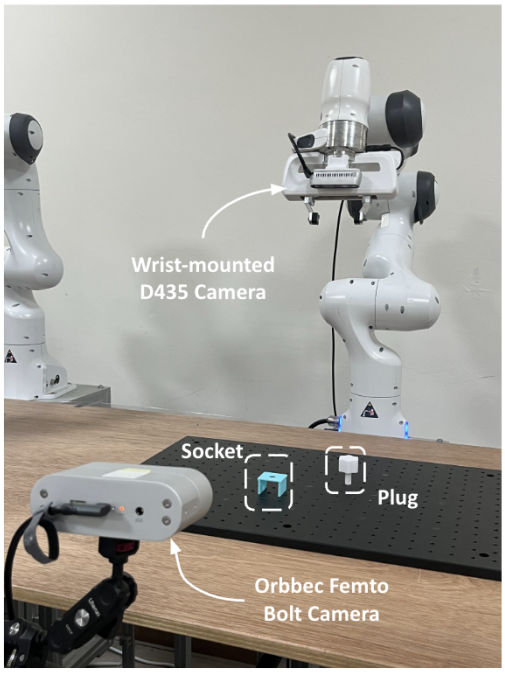

Fig. S4: **Experimental setup.** A Franka Emika Panda robot with a wrist-mounted RealSense D435 camera and an Orbbec Femto Bolt camera is used for recording.

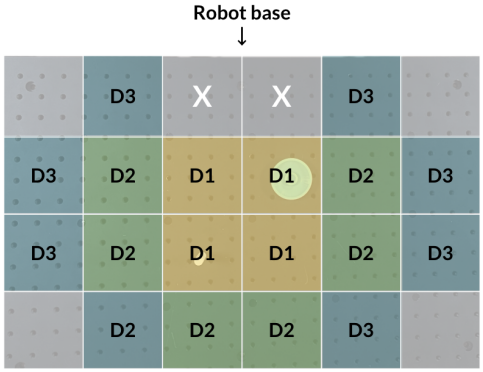

Fig. S5: Tabletop workspace partitioned into three distance-based zones (D1–D3). Cells marked with crosses (X) are excluded due to proximity to the robot base.

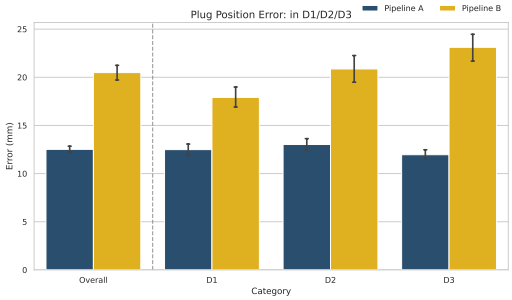

Fig. S6: Comparison of Plug Position Error Across D1–D3 for Pipeline A and B.

TABLE III: Plug Position Error Across Object IDs

Object ID	#00081		#00271		#00340		#01129		#01136		Overall	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Pipeline A	10.95	2.10	17.75	3.05	11.21	2.78	12.00	1.96	9.72	2.12	12.33	3.72
Pipeline B	23.46	16.35	21.92	12.34	23.66	13.61	23.78	14.50	22.84	12.38	23.13	13.94

TABLE IV: Socket Position Error Across Object IDs

Object ID	#00081		#00271		#00340		#01129		#01136		Overall	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Pipeline A	17.39	3.88	17.38	4.02	17.32	4.76	14.60	3.32	13.05	3.38	15.95	4.30
Pipeline B	18.80	8.23	20.03	11.45	22.77	11.53	18.09	9.45	19.49	8.92	19.82	10.12

TABLE V: Final Position Error Across Object IDs

Object ID	#00081		#00271		#00340		#01129		#01136		Overall	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Pipeline A	7.89	2.74	6.21	2.65	7.51	5.94	10.61	3.84	6.06	2.82	7.66	4.1
Pipeline B	18.7	8.93	17.46	13.04	19.34	11.69	16.42	15.22	13.34	9.8	17.29	12.05

REFERENCES

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

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[3] J. K. S. B. Bowen Wen, Wei Yang, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[4] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[5] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 49 250–49 267, 2023.

[6] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[7] G. R. Team, A. Abdolmaleki, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, A. Balakrishna, N. Batchelor, A. Bewley, J. Bingham *et al.*, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025.

[8] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[9] 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.

[10] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "π0. 5: A vision-language-action model with open-world generalization. arxiv 2025," *arXiv preprint arXiv:2504.16054*, 2025.

[11] 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," *arXiv preprint arXiv:2311.01378*, 2023.

[12] 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.

[13] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *RSS*, 2024.

[14] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, J. Fu, J. Gong, and X. Qiu, "Libero-plus: In-depth robustness analysis of vision-language-action models," *arXiv preprint arXiv:2510.13626*, 2025.

[15] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos *et al.*, "Roboarena: Distributed real-world evaluation of generalist robot policies," in *Proceedings of the Conference on Robot Learning (CoRL 2025)*, 2025.

[16] Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine, "Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world," *arXiv preprint arXiv:2503.24278*, 2025.

[17] J. Gao, S. Belkhale, S. Dasari, A. Balakrishna, D. Shah, and D. Sadigh, "A taxonomy for evaluating generalist robot manipulation policies," *IEEE Robotics and Automation Letters (RA-L)*, 2026.

[18] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji, "Benchmarking protocols for evaluating small parts robotic assembly systems," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 883–889, 2020.

[19] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu *et al.*, "Factory: Fast contact for robotic assembly," 2022.

[20] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. Narang, "Industreal: Transferring contact-rich assembly tasks from simulation to reality," in *Robotics: Science and Systems*, 2023.

[21] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt2: Learning precise manipulation from few demonstrations," *RSS*, 2024.

[22] B. Tang, I. Akinola, J. Xu, B. Wen, A. Handa, K. Van Wyk, D. Fox, G. S. Sukhatme, F. Ramos, and Y. Narang, "Automate: Specialist and generalist assembly policies over diverse geometries," in *Robotics: Science and Systems*, 2024.

[23] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

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

[25] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

[26] Z. Zhang, C. Yue, H. Xu, M. Liao, X. Qi, H.-a. Gao, Z. Wang, and H. Zhao, "Robochemist: Long-horizon and safety-compliant robotic chemical experimentation," 2025.