RoboChallenge Zero: A Continually Evolving Benchmark for Zero-Shot Evaluation on Real Robots

RoboChallenge Team

Abstract

RoboChallenge Zero measures how reliably embodied systems complete new manipulation tasks on real robots without task-specific demonstrations or tuning. We designed the benchmark to retest the same system version each month on new tasks and on existing tasks with different objects, layouts, or instructions. The same task goals and success criteria apply to end-to-end vision-language-action policies and agents built on vision-language models. Full task completion is the primary outcome, and we score partial progress separately.

In the first release, 11 multimodal agents succeeded in 148 of 330 rollouts (44.85%) across six tasks. They placed the larger of two blocks in 49 of 55 rollouts and wrote "Z0" in 4 of 55. No agent completed the cable task in 55 rollouts, although one untied the knot without laying out the cable as instructed.

1 Introduction

As teams improve their systems, a stable set of manipulation tasks makes it possible to track progress across submissions. Table30 V2[7] serves this role, but gains on familiar tasks can reflect both transferable skills and refinements made for those particular tasks. We run RoboChallenge Zero alongside Table30 V2 to measure how well a submitted system handles a newly introduced task without task-specific demonstrations or tuning.

Even a task introduced for zero-shot evaluation becomes familiar once teams have seen the instructions and reviewed their rollouts. Monthly updates give us fresh tasks on which to test a fixed system version. Revisiting earlier tasks under changed conditions also lets us examine whether a skill carries over when, for example, the same instruction refers to an object in a different layout. We compare systems within each release so that they face the same task set.

Real-robot rollouts reveal what happens after a system selects an action. A robot may reach for the correct object, miss the grasp, and need to use its next observation to reposition the gripper. This matters for agents that select or compose robot skills[1, 5] and for vision-language-action policies that generate actions directly[3, 4, 6, 10]. Building on RoboArena [2], AutoEval [9], and RoboChallenge [8], the RoboChallenge Zero protocol defines success for both approaches by what the robot accomplishes in each rollout.

The main contributions of this report are:

- **A real-robot zero-shot evaluation protocol**: no task-specific training data is provided, and standardized real-robot execution with result review ensures comparable results.
- **A unified multi-system result framework**: compatible with native VLA policies and multimodal large-model agents, reporting success rate, progress score, execution time, and step count.
- **A zero-shot real-robot baseline for 11 multimodal models**: based on 330 real-robot rollouts, characterizing the capability distribution and systematic limitations of current models.

2 RoboChallenge Zero Benchmark

2.1 Benchmark Definition

RoboChallenge Zero evaluates whether an embodied system can complete real-robot manipulation tasks without task-specific adaptation. The tasks may involve unseen objects, spatial configurations, manipulation procedures, or combinations of robot skills. Each task is evaluated through multiple independent rollouts to account for the stochastic nature of real-world perception and control.

Task instructions and grading criteria are public, but no demonstration data is provided, and submitted systems are expected not to be trained or fine-tuned on data from the benchmark tasks. During evaluation, task-specific prompt changes, online parameter updates, and human guidance are avoided in principle; operators are mainly responsible for robot initialization, scene preparation, and resets.

This protocol is designed to measure the capability of the submitted system at evaluation time; thus, improvement obtained through benchmark-specific adaptation falls outside its scope. Online adaptation and in-context demonstrations may be studied in future auxiliary settings, but are excluded from the strict zero-shot results. RoboChallenge Zero complements Table30 V2; the two benchmarks measure related but distinct aspects of embodied intelligence (Table 1).

Table 1: Comparison between RoboChallenge Zero and Table30 V2.

	Table30 V2	RoboChallenge Zero
Primary purpose	Evaluate generalist robot policies on a stable real-world task suite	Measure zero-shot generalization beyond the development distribution
Data condition	Task demonstration data provided	No task-specific training data provided
Main output	Comparable progress on a specific task suite	Zero-shot capability and efficiency

2.2 Evaluated Systems

RoboChallenge Zero evaluates the behavior of the complete submitted system and allows different systems to adopt different architectures. Submissions are described using three categories:

- **Scope of participation.** RoboChallenge Zero evaluates three types of systems against the same task objectives and success criteria. Native VLA policies map observations and instructions directly to robot actions. VLM/LLM agents plan actions and invoke robot skills through tools. Integrated embodied systems combine models with perception modules, memory, planners, skills, and custom controllers. This release reports results for 11 VLM/LLM agents. Native VLA submissions are open.

- **Hardware platforms.** The W1 is a bimanual mobile platform with Airbot Play arms and parallel grippers. The ARX5 is a table-mounted, six-degree-of-freedom arm. Both use Intel RealSense RGB-D cameras to view the workspace from above and from each wrist. The ARX5 also has a side camera.

- **Experimental procedure.** After standard robot and scene setup, we issue the task instruction and the system runs autonomously. Native VLA policies submit action sequences through the RoboChallenge remote-robot interface. VLM/LLM agents run for up to 40 model calls on single-arm tasks or 50 on bimanual tasks. At each step, the agent receives camera images and robot state, with additional depth and torque data available. It specifies a target end-effector pose and a gripper command through tool calls. The controller executes the motion using inverse kinematics and returns the next observation. We record video and model outputs and review the video to assess the run against the task criteria. Rollouts that fail because of hardware or benchmark infrastructure issues may be rerun.

- **Results reporting.** The unified leaderboard provides category-specific, open-weight, and efficiency views. We encourage submitters to document their prompts, available skills, external APIs, and compute resources so that readers can interpret the results in the context of the entire system.

2.3 Monthly Task Refresh and Longitudinal Evaluation

RoboChallenge Zero follows a monthly update cycle. Each release adds new tasks and refreshes existing ones through changes to objects, layouts, and instructions. These updates continually broaden capability coverage and reduce the opportunity to overfit a fixed task suite.

The same model versions can continue to participate across monthly releases, yielding longitudinal results on both new and refreshed tasks. Any change in model weights, prompts, tools, or skill libraries constitutes a new system version. The results identify both the system version and the benchmark release, and the leaderboard comparisons are made within the same release.

Future releases will also introduce additional robot platforms to extend evaluation across embodiments and hardware capabilities. The interfaces and requirements for the new platforms will be documented before evaluation, and the results will identify the platform used.

2.4 Metrics

Each model performs five independent rollouts per task. Four measures are reported:

- **Success rate** (primary): the proportion of rollouts that complete the objective of the task;
- **Progress score**: each rollout earns 0–10 points in 2–4 task-specific scoring criteria. For each model, we sum its five rollout scores on each task (up to 50 points per task), then average the six task totals. The overall progress score ranges from 0 to 50;
- **Execution time**: the duration of a rollout from start to end;
- **Step count**: the number of model calls in a rollout.

As in Table30, full task completion determines success.

3 Initial Evaluation

3.1 Experimental Setup

We evaluated 11 multimodal models under the same zero-shot real-robot protocol: `claude-fable-5-2`, `claude-opus-5-5`, `deepseek-v4.1-flash`, `gemini-4-neon`, `gemini-4-argon`, `gpt-6.2-astra`, `gpt-6-astra`, `gpt-6.1-sol`, `kimi-k3.1-pro`, `qwen4-max`, and `qwen3.8-max`.

All models share the same prompt template and tool interface. The evaluation comprises six manipulation tasks (Table 2) on two robot platforms . Each model performs five independent rollouts per task, yielding 30 rollouts per model and 330 real-robot rollouts in total. The robot and scene are reset between rollouts.

3.2 Overall Performance

Across the six tasks, the 11 models succeeded in 148 of 330 rollouts (44.85%). Success rates ranged from 6.67% to 70.00% (Fig. 1). The top model, `gpt-6.2-astra`, completed 21 of 30 rollouts, one more than `claude-fable-5-2`.

Table 2: The RoboChallenge Zero task suite.

ID	Platform	Instruction	Primary demands
T1	W1	Knock over the shorter water bottle first, then the taller one.	Height comparison and ordered execution
T2	W1	Hold the paper in place with one hand and write "Z0" with the other.	Bimanual coordination and precise trajectories
T3	W1	Untangle the cable and lay it on the table without loops or overlaps.	Deformable-object perception and manipulation
T4	ARX5	Place the larger block on the plate.	Size comparison and grasp-and-place execution
T5	ARX5	Pick up the doll wearing blue and place it in the box.	Attribute-based selection and grasping
T6	ARX5	Move exactly one stick to make "6 + 4 = 4" correct.	Mathematical reasoning and repositioning

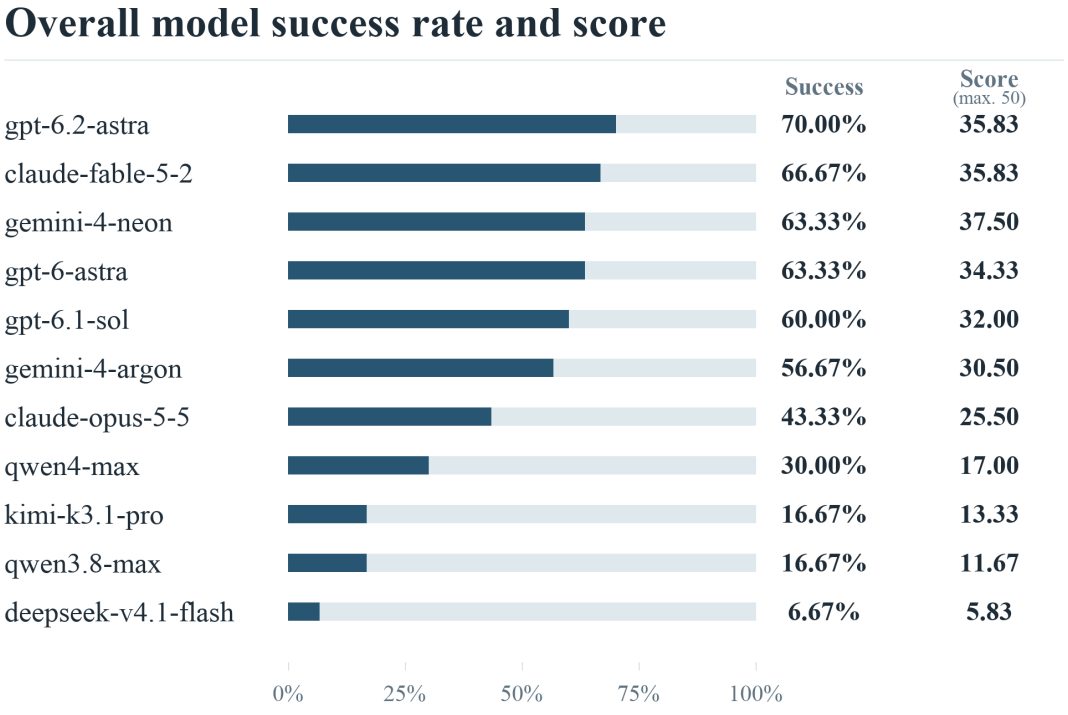


Figure 1: Success rate of each model (30 rollouts per model).

3.3 Task-Level Results

Task difficulty varies more sharply than the performance of the leading systems (Fig. 2). Placing the larger block (T4) is the most consistently completed task at 89.09% (49/55), with 7 of the 11 models succeeding in all five rollouts, suggesting that coarse grasp-and-place behavior is already within the capability range of most current systems. Knocking over bottles (T1, 65.45%), picking up the doll (T5, 61.82%), and fixing the equation (T6, 45.45%) provide greater differentiation, as they require height comparison with ordered execution, color recognition with deformable-object grasping, and selection of the target through mathematical reasoning, respectively. T6 admits two valid solutions: moving the vertical stick of the plus sign to obtain 8-4=4, or moving the middle stick of the 6 to obtain 0+4=4; different models arrived at different solutions (Fig. 5).

Writing "Z0" (T2) and untangling the cable (T3) are the most challenging tasks. T2 has a success rate of 7.27% (4/55), with successful rollouts only of `gpt-6.2-astra`, `claude-fable-5-2`, and `gemini-4-neon`, whose written results are shown in Fig. 4. The task requires converting a semantic instruction into a continuous and spatially precise trajectory. T3 is not completed in any of the 55 rollouts; it requires reasoning about deformable-object state, contact, and the consequences of intermediate actions. The sharp decline in these tasks indicates that the principal bottleneck lies in grounding high-level intent into reliable closed-loop physical behavior.

The task × model heatmap (Fig. 3) further shows that no system is uniformly strongest; the differences between models are driven mainly by T1, T2, T5, and T6.

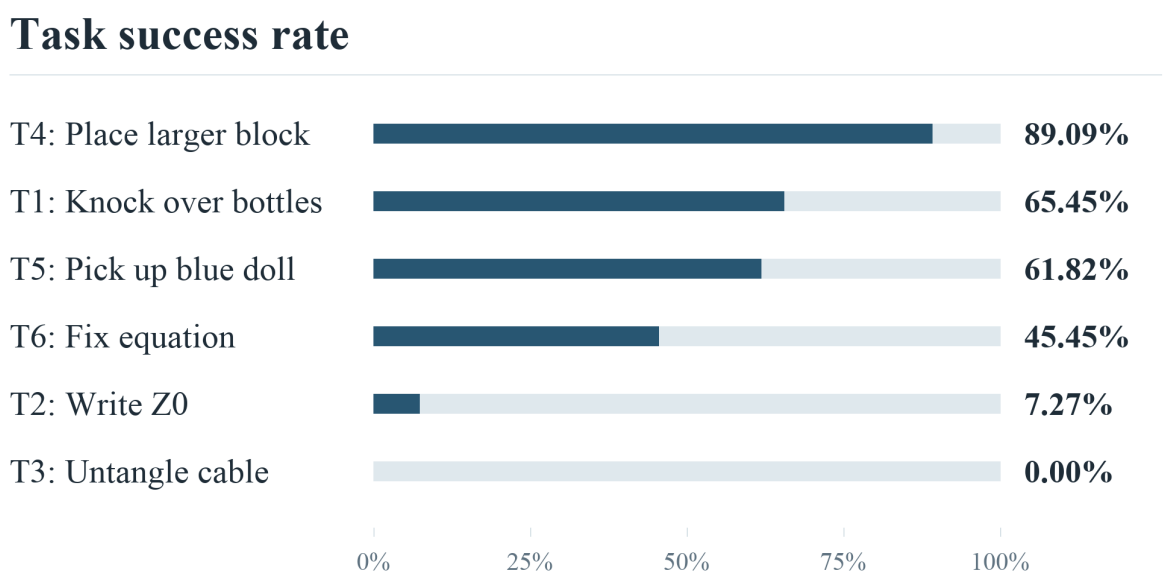
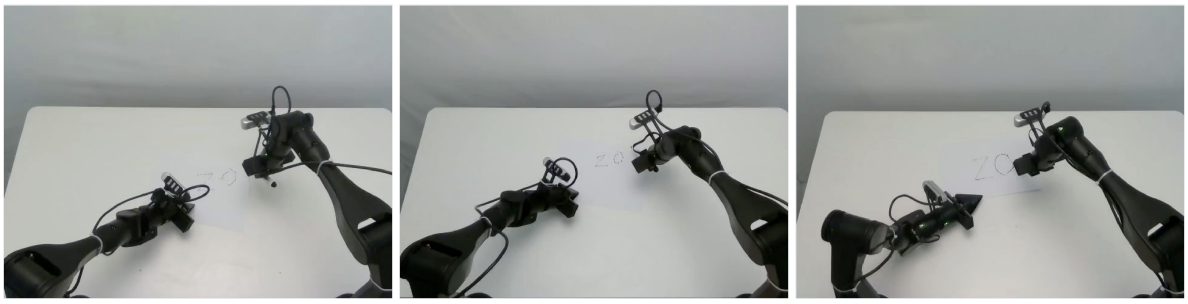


Figure 2: Success rate of each task.

Task × model success rate

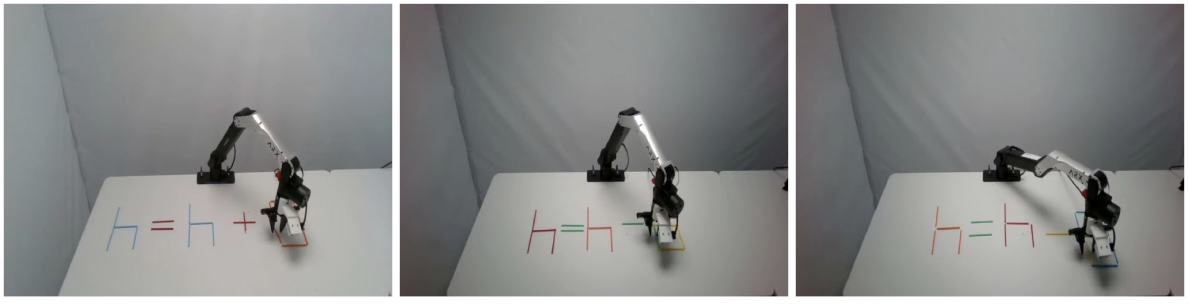
	T1: Knock over bottles	T2: Write Z0	T3: Untangle cable	T4: Place larger block	T5: Pick up blue doll	T6: Fix equation
gpt-6.2-astra	100%	20%	0%	100%	100%	100%
claude-fable-5-2	80%	20%	0%	100%	100%	100%
gemini-4-neon	100%	40%	0%	100%	100%	40%
gpt-6-astra	100%	0%	0%	100%	100%	80%
gpt-6.1-sol	80%	0%	0%	100%	100%	80%
gemini-4-argon	80%	0%	0%	100%	100%	60%
claude-opus-5-5	80%	0%	0%	100%	40%	40%
qwen4-max	60%	0%	0%	80%	40%	0%
kimi-k3.1-pro	20%	0%	0%	80%	0%	0%
qwen3.8-max	20%	0%	0%	80%	0%	0%
deepseek-v4.1-flash	0%	0%	0%	40%	0%	0%

Figure 3: Success rate of each model on each task.



(a) gemini-4-neon (b) claude-fable-5-2 (c) gpt-6.2-astra

Figure 4: Successful results of three models on the writing task (T2).



(a) gemini-4-neon (b) gpt-6.2-astra (c) claude-fable-5-2

Figure 5: Different solutions of three models on the equation task (T6).

3.4 Progress and Efficiency

The success rate captures whether the final objective is achieved, and the progress score further distinguishes near-complete executions from early failures. This is particularly important for T2 and T3. Among the 55 rollouts of T3, only one rollout by gemini-4-neon scores 7 points: the knot is untied, but the cable is not laid out as required, so the success condition is not met; all other rollouts score 0. Current models can thus make partial progress on deformable-object tasks, while full completion remains out of reach.

The completion time and the step count provide complementary views of efficiency (Figs. 6 and 7). Successful rollouts of T4 and T5 consume about 480k and 518k tokens on average and take 13–14 minutes. The four successful rollouts of T2 average 1,735.6k tokens, 23.8 minutes, and 43.8 steps, close to the call limit, indicating that fine trajectory tasks require substantially more perception–reasoning cycles. Time and step count are reported alongside success and are not combined into a composite score.

Fig. 8 places each model's success rate against the average token consumption of its successful rollouts. Within model families, newer releases generally show improved embodied manipulation capability. Figure 9 estimates the mean API cost of successful rollouts using model-specific input and output token prices. The estimated cost varies substantially across models and does not increase consistently with the success rate, suggesting that stronger performance does not necessarily require proportionally higher inference spending. Figure 10 compares the duration distributions of successful and failed rollouts. Successful rollouts are concentrated at 5–10 minutes (43/148, 29.05%) and all finish within 60 minutes. Failed rollouts are more dispersed, peaking at 15 to 20 minutes (25/182, 13.74%), and 30 of them (16.5%) exceed 60 minutes.

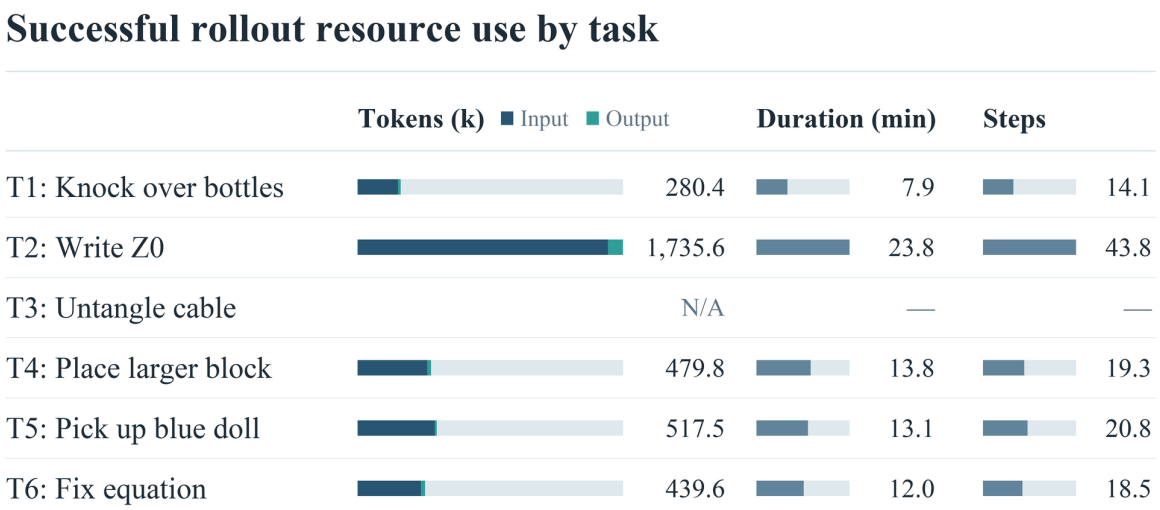


Figure 6: Resource use of successful rollouts by task.

Successful rollout resource use by model

	Tokens (k) ■ Input ■ Output		Duration (min)		Steps	
gpt-6.2-astra		274.6		5.6		15.3
claude-fable-5-2		433.5		11.2		15.7
gemini-4-neon		539.0		13.4		17.6
gpt-6-astra		387.6		8.2		20.4
gpt-6.1-sol		490.8		10.8		23.1
gemini-4-argon		217.9		5.9		11.1
claude-opus-5-5		754.0		11.7		23.1
qwen4-max		630.1		28.0		23.3
kimi-k3.1-pro		782.5		19.1		28.4
qwen3.8-max		653.1		42.1		24.0
deepseek-v4.1-flash		920.1		23.2		30.5

Figure 7: Resource use of successful rollouts by model.

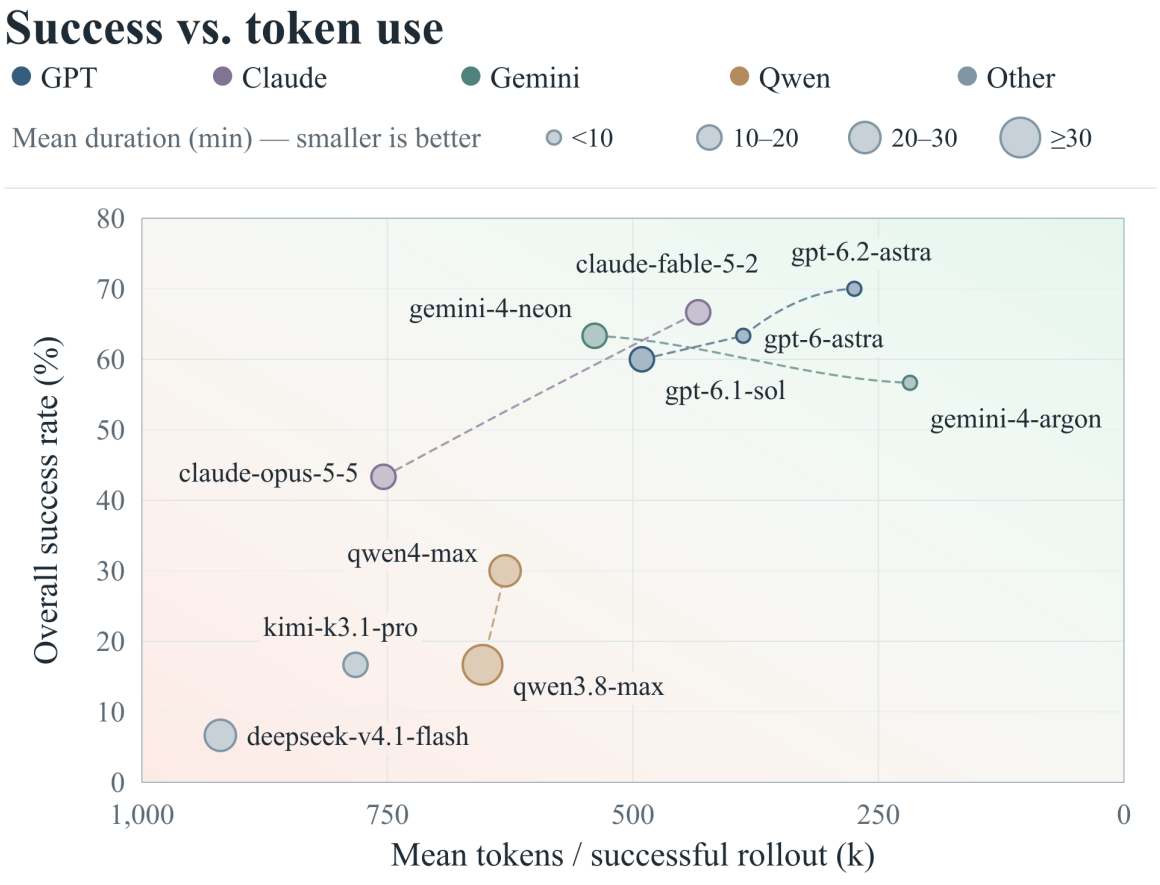


Figure 8: Success rate versus mean token consumption per successful rollout.

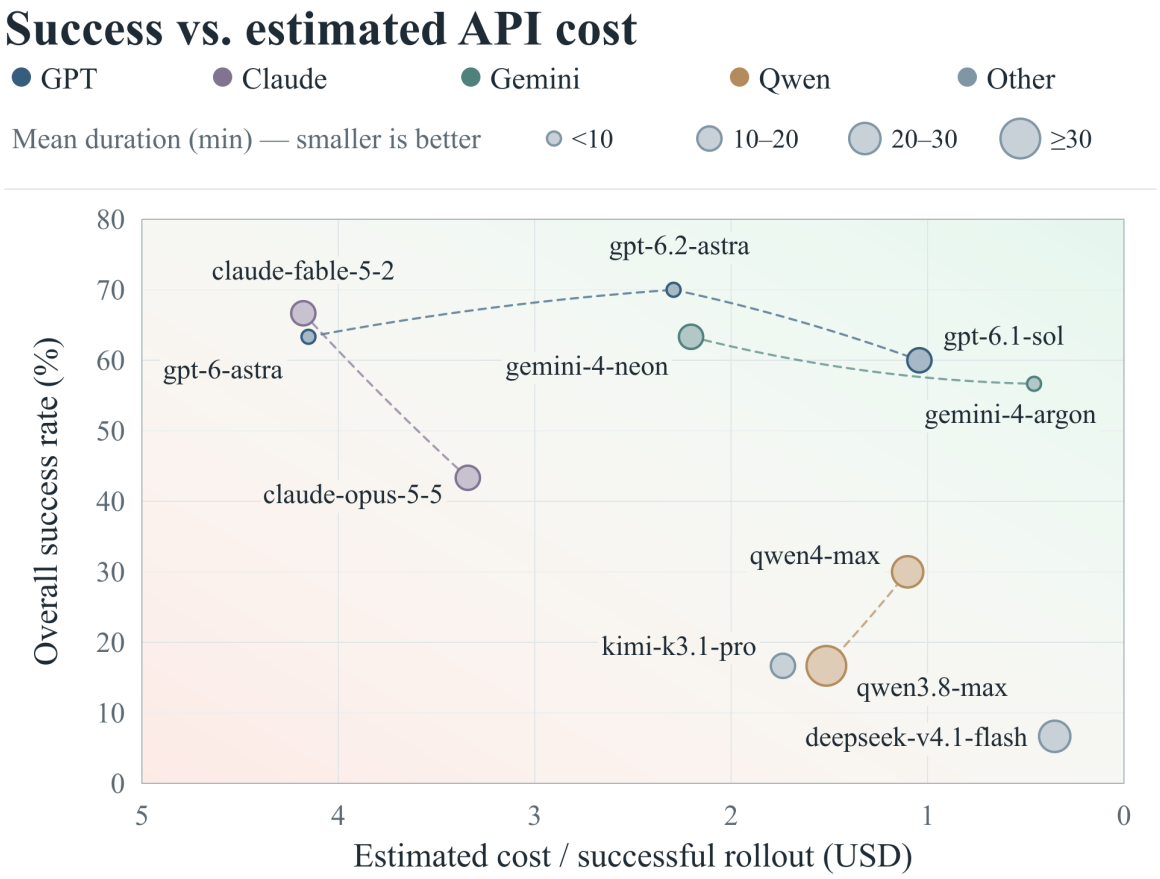


Figure 9: Success rate versus estimated mean API cost per successful rollout.

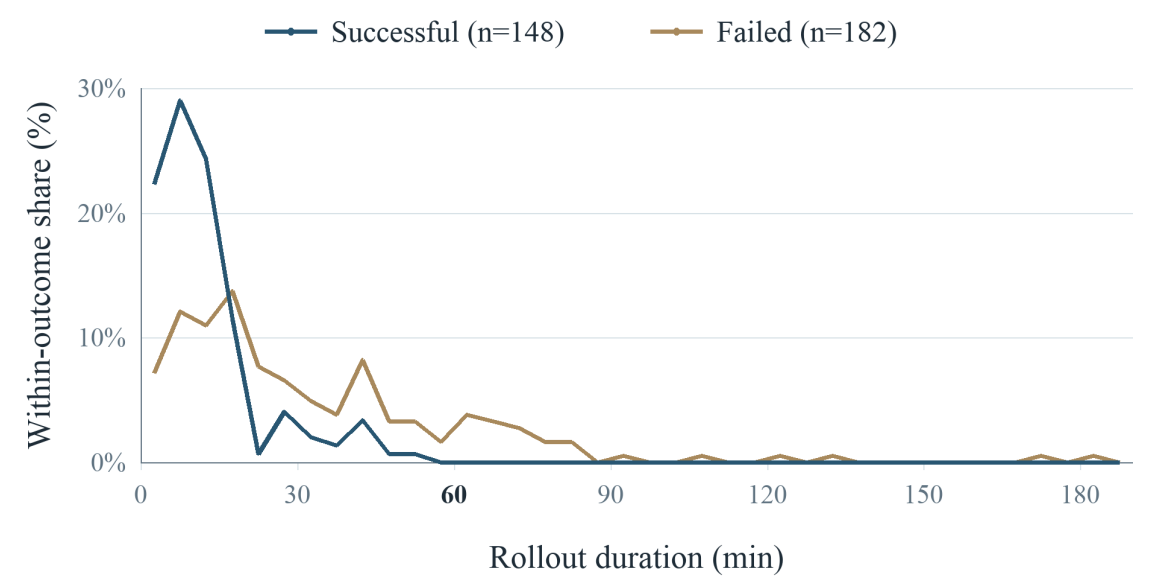


Figure 10: Duration distributions of successful and failed rollouts.

3.5 Discussion and Limitations

The results lead to the following observations.

Zero-shot real-robot capability remains strongly task-dependent. The gap between T4 (89.09%) and T3 (0%) exceeds the gap between the strongest and weakest models (70.00% and 6.67%). Success on familiar grasping and placement behaviors does not imply competence on precise or contact-rich manipulation.

Fine-grained perception errors are an observed failure mode. In T1, some models judge the two bottles to be of equal height or confuse which one is shorter, and consequently knock them over in the wrong order. In T3, models struggle to infer the crossing structure of the cable from images and cannot determine which segment to grasp to undo the knot; In some rollouts, they pull the knot tighter. These errors occur before execution and determine whether the subsequent manipulation is meaningful. The perceptual competence that models exhibit in general visual question answering transfers poorly to fine-grained attribute comparison and deformable-object state estimation in real scenes.

Spatial grounding and collision-free execution remain challenging despite access to depth maps. On the bimanual W1 tasks, observed failures include gripper–table contact that stops the arm and collisions between the two arms during coordinated motion. In some cases, the model identifies the task goal correctly but fails to execute a valid motion. These observations show that RGB images and depth maps alone do not ensure collision-free execution. Successful motion also requires accounting for the robot's spatial occupancy, inter-arm clearance, and the trajectory between target poses. The current evaluation does not isolate the contributions of model reasoning, calibration, trajectory generation, and collision handling to these failures.

Reliable physical execution remains challenging in the evaluated systems. Current models can usually interpret the objective and select a plausible action, but they remain less reliable at tracking physical state, producing precise motion, and recovering from execution errors. The low success rates on T2 and T3 illustrate this gap most clearly.

Success alone is insufficient for comparing embodied systems. Progress, time, step count, safety, and system scope are necessary to distinguish reliable and efficient behavior from success achieved through costly or highly engineered execution.

These findings support the role of RoboChallenge Zero as a continually evolving benchmark. As individual tasks become saturated, harder object configurations, disturbances, and task compositions can be introduced while preserving the zero-shot protocol. Versioned results make this evolution explicit and keep scores obtained on different task distributions from being treated as directly interchangeable.

4 Conclusion

RoboChallenge Zero provides a practical framework for measuring zero-shot embodied intelligence in the physical world. It combines an evaluation protocol without task-specific training data, real-robot execution, category-aware reporting, and result review. Together with Table30 V2, it enables RoboChallenge to measure both progress on a stable generalist task suite and generalization beyond known benchmark conditions. The 330 real-robot rollouts in this release establish a zero-shot baseline for 11 multimodal models, revealing large differences across task types, systematic limitations in fine-grained visual perception and 3D spatial awareness, and the gap between semantic reasoning and physical execution.

We hope the benchmark helps the community move from isolated demonstrations toward comparable, reproducible evidence of what embodied AI systems can actually do.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, et al. Do as I can, not as I say: Grounding language in robotic affordances. In Conference on Robot Learning (CoRL), 2022.

[2] Pranav Atreya, Karl Pertsch, Tony Lee, et al. RoboArena: Distributed real-world evaluation of generalist robot policies. arXiv preprint arXiv:2506.18123, 2025.

[3] Kevin Black, Noah Brown, Danny Driess, et al. $\pi 0$: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

[4] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In Conference on Robot Learning (CoRL), 2024.

[5] Jacky Liang, Wenlong Huang, Fei Xia, et al. Code as policies: Language model programs for embodied control. In IEEE International Conference on Robotics and Automation (ICRA), 2023.

[6] Physical Intelligence, Kevin Black, Noah Brown, et al. $\pi 0.5$: A vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[7] RoboChallenge Dexmal Team. Table30 V2: Evaluating generalized models by real robots at scale. Technical report, 2026.

[8] Adina Yakefu, Bin Xie, Chongyang Xu, et al. RoboChallenge: Large-scale real-robot evaluation of embodied policies. arXiv preprint arXiv:2510.17950, 2025.

[9] Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. AutoEval: Autonomous evaluation of generalist robot manipulation policies in the real world. arXiv preprint arXiv:2503.24278, 2025.

[10] Brianna Zitkovich, Tianhe Yu, Sichun Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning (CoRL), 2023.