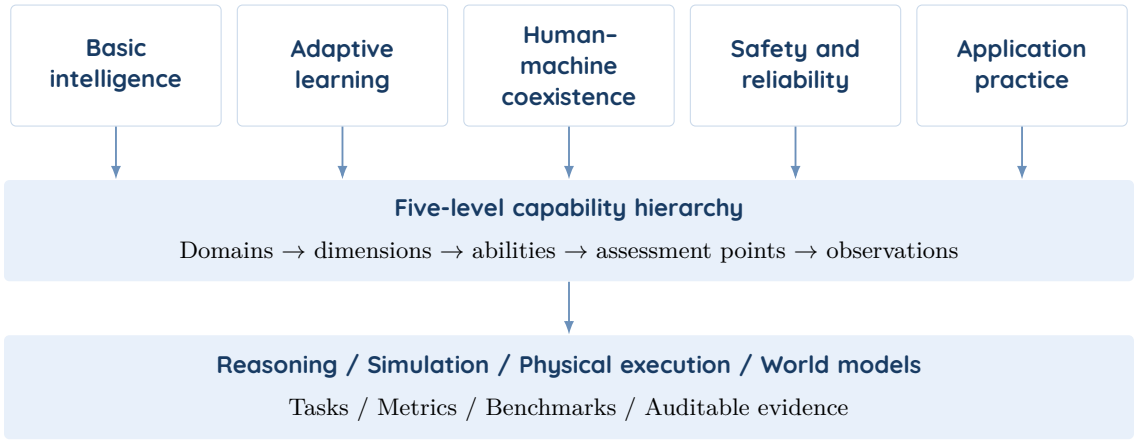
REAL: A Unified Evaluation of Physical AI

REAL Team
Beijing Academy of Artificial Intelligence (BAAI)
See Contributions and Author List

Abstract

Physical AI evaluation requires a system that relates capability definitions to observable task evidence. We present REAL, a unified capability–task–metric framework organized around basic intelligence, adaptive learning, human–machine coexistence, safety and reliability, and application practice. The framework organizes evaluation design through 14 dimensions, 55 abilities, and 158 capability indicators. Its five-level hierarchy of domains, dimensions, abilities, assessment points, and observations guides their operationalization as task specifications. Four leaderboards instantiate the system through embodied reasoning, simulated manipulation, physical manipulation, and world prediction. Theoretical applicability, scheduled scope, and completed experiments are reported separately. Shared task identities, frozen protocols, auditable artifacts, coverage, and uncertainty support reproducible comparisons, with dedicated simulation-to-real alignment and model-role studies. The report describes the complete capability system and its operational mapping; empirical findings require verified model submissions and measured results.



REPORT CONTENTS

Contents

1 Introduction 1

1.1 What is Physical AI? 1

1.2 Why Physical AI matters 1

1.3 Physical AI approaches and the scope of this report 2

1.4 Why evaluation is needed and what makes it difficult 3

1.5 Scope and intended contributions 4

1.6 Related work 5

1.6.1 World Models 5

1.6.2 Embodied Manipulation Models 6

1.6.3 Vision–Language Models 7

1.6.4 Physical AI Benchmarks 7

1.6.5 Unified Physical AI Evaluation 8

2 REAL Physical AI Evaluation System 9

2.1 The complete capability evaluation system 9

2.1.1 Design logic of the five domains 9

2.1.2 Five levels from capability to observable evidence 10

2.1.3 Embodied reasoning evaluation system 11

2.1.4 Simulated manipulation evaluation system 12

2.1.5 Real-world manipulation evaluation system 12

2.1.6 World-model evaluation system 13

2.2 Evaluation subjects 13

2.3 Evaluation methods 15

2.4 Metric system 16

2.5 Evaluation tasks 16

2.5.1 Embodied reasoning tasks 17

2.5.2 Simulated manipulation tasks 17

2.5.3 Real-world manipulation tasks 17

2.5.4 World-model tasks 18

2.6 Evaluation workflow 18

2.7 Evaluation environments 18

2.7.1 Embodied reasoning environment 19

2.7.2 Simulated manipulation environment 19

2.7.3 Real-world manipulation environment 19

2.7.4 World-model environment 20

2.8 Evaluation data 20

2.8.1 Embodied reasoning data 21

2.8.2 Simulated manipulation data 22

2.8.3 Real-world manipulation data 22

2.8.4 World-model data 22

2.9 Leaderboards and scoring 23

2.9.1 Freezing the evaluation setting 23

2.9.2 Run accounting and coverage 23

2.9.3 Binary outcomes and second-level aggregation 24

2.9.4 World-model additive scoring 24

2.9.5 Criterion-level evidence and uncertainty 25

2.9.6 Judge calibration and statistical reporting 26
2.10 Fairness and validity 26
2.11 Distinctive evaluation design 27
2.12 Operation and maintenance 27
3 Embodied Reasoning Model Evaluation 27
3.1 Scope (Evaluation subjects and system mapping) 27
3.2 Benchmark Design (Task and data design) 28
3.3 Evaluation Protocol (Input, output, and evaluation methods) 29
3.4 Experimental Setup (Metrics and experimental settings) 30
3.5 Results and Discussion (Results and diagnostic analysis) 30
3.6 Ablations and Case Studies (Ablations and case analysis) 32
4 Simulated Manipulation Model Evaluation 32
4.1 Evaluation subjects and system mapping 33
4.2 Task and data design 33
4.3 Input, output, and evaluation methods 33
4.4 Metrics and experimental settings 33
4.5 Results and diagnostic analysis 34
4.6 Ablations and case analysis 34
5 Real-World Manipulation Model Evaluation 35
5.1 Evaluation subjects and system mapping 35
5.2 Task and data design 35
5.3 Input, output, and evaluation methods 35
5.4 Metrics and experimental settings 36
5.5 Results and diagnostic analysis 36
5.6 Ablations and case analysis 37
6 World Model Evaluation 37
6.1 Scope and evaluation interfaces 37
6.2 Benchmark construction and data 38
6.3 Evaluation protocol and metrics 39
6.4 Models and experimental settings 41
6.5 Results and diagnostic analysis 42
6.6 Ablations and evidence-based cases 42
7 Simulation-to-Real Alignment Evaluation 43
7.1 Matched tasks and comparison units 43
7.2 Alignment measurements and uncertainty 43
7.3 Cross-leaderboard analysis and ablations 44
7.4 Experimental report and failure analysis 44
8 GPT6 as a VLM: Simulation and Real-World Experiments 44
8.1 VLM role and execution interface 45
8.2 Simulated manipulation experiments 45
8.3 Real-world manipulation experiments 46
8.4 VLM ablations and auxiliary-route comparisons 47
9 Summary and Outlook 47
9.1 Summary 47
9.2 Limitations and Responsible Interpretation 48
9.3 Reproducibility and Release Plan 48
Contributions and Author List 55

Appendix 57
A Capability System Index and Version Notes 57
B Qualitative Evaluation Examples 58
B.1 Embodied reasoning cases 58
B.2 Real-world manipulation cases 59
B.3 Simulated manipulation cases 59
B.4 World-model cases 59
B.5 Embodied reasoning — level-3 ability breakdown 59
C Task and Run Record Templates 62
D Publication Completion Checklist 63
E Complete Evaluation Data and Artifact Index 63
F Repository Sources for This Draft 63

1 Introduction

Physical AI connects intelligent capabilities to perception, decisions, and actions in real environments. This chapter defines Physical AI and explains its importance before introducing the technical approaches, evaluation challenges, and scope of REAL.

1.1 What is Physical AI?

Writing owner: 高达辉 and 史可为 **To complete:** Review the definition and supply examples from each leaderboard.

Physical AI is an intelligent system that understands its environment, takes action, and completes tasks in the physical world. It gathers information through perception, predicts how the environment may change and what its actions may cause, and controls physical systems such as robots to act toward task goals. Through sustained interaction with the environment and humans, it adjusts its behavior based on actual outcomes and human feedback. By accumulating and learning from these experiences, it improves its ability to complete tasks and adapt to new environments.

For example, to move a cup from a table to a designated location, a robot must first locate the cup and its destination, determine how to grasp, move, and place it, and predict whether the cup might slip or encounter an obstacle along the way. During execution, it must check whether its grasp is secure and whether the cup has reached the right location. If the cup moves or a person changes the request, the robot must adjust its actions accordingly. Evaluating Physical AI therefore requires examining whether it understands the environment and predicts action outcomes correctly, as well as whether it completes tasks, respects constraints, and responds to changes during execution.

This report focuses on robot-related Physical AI models. It examines how reasoning models understand tasks and form plans, how manipulation policies generate actions, and how world models predict environmental changes and action outcomes. We use matched tasks to analyze how these models support task completion, conducting controlled, repeatable tests in simulation and then using physical robot trials to examine execution in real environments.

1.2 Why Physical AI matters

Writing owner: 高达辉 and 史可为 **To complete:** Supply application motivations and primary evidence for each setting.

The development of artificial intelligence (AI) is increasingly moving from the digital to the physical, from silicon-based to carbon-based systems, and from individual agents to collectives. In the digital AI stage, AI systems represented by large language models operate primarily within computers, learning knowledge, understanding information, and generating content in the digital world. In the Physical AI era, AI begins to enter the physical world: it perceives environments, predicts changes, takes actions, and learns and evolves through sustained interaction with humans and the real world, extending the boundaries of human physical capabilities. Looking ahead, AI will extend further into living and microscopic worlds, moving from understanding and acting on macroscopic physical environments toward understanding, modeling, and intervening in biological, neural, and even molecular-scale complex systems, extending or reshaping carbon-based life. From the digital world to the physical world and onward to living

and microscopic worlds, AI continues to expand the worlds that intelligence can understand and act upon, while reshaping the relationships among humans, intelligence, and the environment.

Human–AI coexistence will become an inevitable part of the future. As AI enters the physical world, how can it truly be applied across all aspects of that world? BAAI's view is: Physical AGI. The development of "Physical AI" technology is flourishing, but how far is it from practical deployment? How far is it from "Physical AGI"? The field currently lacks systematic research into this question and cannot yet provide an answer.

To this end, BAAI starts from "real needs" and is guided by an "evaluation framework" to pursue in-depth research into core Physical AI technologies and build Physical AI infrastructure. Through evaluation, it identifies the gap between technology and needs, thereby guiding technological development and advancing the realization of Physical AGI.

1.3 Physical AI approaches and the scope of this report

Writing owner: 高达辉 and 史可为 **To complete:** Confirm actual model interfaces, tested components, and coverage.

The evolution of Physical AI can be summarized as a progression from visual feedback control through language-guided action generation and world prediction toward foundation models that integrate understanding, prediction, and action [5]. **Visual servoing** uses visual observations to correct motion, typically combining perception, planning, and control modules for predefined tasks. **End-to-end vision–language–action models (VLAs)** connect visual and language understanding to action generation so that robots can act on human intent. **World action models (WAMs)** introduce prediction of action consequences to inform action selection and closed-loop decisions. **Foundation world models** point toward integrated multimodal understanding, prediction, and action as a shared basis for interaction across tasks and environments.

The roadmap also describes five capability levels: interaction in constrained environments (L1), interaction in open environments (L2), interaction with rollout and prediction (L3), general physical interaction (L4), and autonomous evolution (L5). These levels express development objectives and directions; the capabilities of a particular system still require evaluation. Section 1.6 examines representative methods and their training, prediction, and execution mechanisms.

REAL organizes four leaderboards in the order used throughout this report: **embodied reasoning, simulated manipulation, real-world manipulation, and world models**. Chapters 3–6 pair each leaderboard's capability scope and protocol with its experimental report. Chapter 7 studies matched simulation and physical execution; Chapter 8 defines the planned GPT6 VLM study with separate simulation and real-world manipulation experiments.

1.4 Why evaluation is needed and what makes it difficult

Writing owner: 高达辉 and 史可为 **To complete:** Supply observed evaluation difficulties and representative failure cases.

01 Foundations and challenges

- **Evaluation supports continual improvement**: Data, models, and evaluation are three essential elements of continual AI development. Evaluation translates task requirements into testable capability objectives and diagnoses model performance, informing data development, training optimization, and model improvement within an iterative feedback loop.
- **Physical interaction increases evaluation complexity**: Physical AI capabilities emerge through continuous interaction with the environment. Performance depends jointly on spatial relations, object properties, dynamics, and interaction feedback. Evaluation must therefore examine reasoning, action execution, and environmental responses together, while addressing the difficulty of reproducing experimental conditions and the cost of physical execution.
- **Evaluation requires its own methodology**: Evaluation defines capability objectives and development directions, measures current performance, and provides actionable evidence for improvement. This requires systematic research into capability frameworks, task design, metrics, measurement methods, and evaluation tools. A single test result cannot substitute for a complete assessment.

02 Gaps in evaluation frameworks

- **Fragmented benchmarks limit comparability**: Existing benchmarks differ in task design, capability scope, data distributions, and evaluation conventions, making their results difficult to compare directly. Without a common reference framework, performance on isolated tasks provides an incomplete basis for judging overall capability.
- **Top-level design determines coverage and measurement quality**: Combining existing task collections does not necessarily produce a complete evaluation framework. Systematic design must begin with capability definitions, establish relationships among capabilities, tasks, and metrics, identify gaps and redundancy in coverage, and justify the validity of measurement methods.
- **Credibility depends on independence and expertise**: Authority and fairness require an independent evaluation stance, methodological expertise, and transparent, consistent rules. As a neutral, third-party nonprofit organization, BAAI should maintain independence from evaluated participants and establish credibility through public protocols, verifiable evidence, and consistently applied evaluation procedures.

03 Value of unified evaluation

- **A common yardstick advances technology and industry**: Unified, trustworthy evaluation standards provide a shared basis for choosing technical approaches, allocating research resources, and comparing outcomes, reducing repeated validation and ineffective investment. Continually identifying capability limitations and gaps relative to application needs can improve the efficiency and quality of technological development.
- **Evaluation standards build influence in the field**: By proposing testable capability definitions and evaluation standards, REAL can contribute to shared understanding and technical agendas, establishing a strategic basis for long-term research. Its influence should rest on methodological validity, reliable evidence, and reproducible results.
- **Professional services support sustainable operation**: Trustworthy, expert, timely, and comprehensive evaluation can support an industry-facing REAL service platform. Such a platform can inform investment and financing decisions, data transactions, and technical consulting, while exploring advertising, sponsorship, and platform partnerships. The continuing provision of valuable evidence and professional services can generate sustained returns for long-term research and platform development.

Evaluation challenges and comparison boundaries

The central challenge in evaluating Physical AI is to establish a reliable connection between capability judgments and observable evidence. Correct verbal explanations, visually plausible prediction videos, and successful physical interactions provide different kinds of evidence and cannot directly substitute for one another. Training and adaptation budgets, observation permissions, hardware, environment resets, and assessment errors also affect measurements and their comparability.

Accordingly, the REAL evaluation design first specifies capability applicability, task configurations, information access, and evidence-recording requirements. It then reports overall performance and capability-specific results within groups whose participation conditions are consistent. Cross-leaderboard analysis requires matched tasks and comparable conditions; the current version does not provide a combined ranking across the four leaderboards.

1.5 Scope and intended contributions

Writing owner: 高达辉 and 史可为 **To complete:** Verify implemented contributions and their supporting artifacts.

This working report develops three connected contributions:

1. **A capability–task–metric evaluation system.** Five domains and a complete five-level hierarchy connect capability definitions and diagnostic tasks to benchmark, simulated, and physical evidence. Outcome, adaptation, efficiency, and reliability measurements have explicit applicability and comparison boundaries.
2. **Four complete leaderboard reports.** Each leaderboard projects the unified system through task data, input and output contracts, methods, metric settings, results, and ablations.

3. **Auditable alignment and role studies.** Versioned manifests, denominator accounting, evidence bounds, statistical uncertainty, matched simulation-to-real experiments, and role-specific model comparisons support reproducible interpretation.

> **Release status.** Local `uniwmbench` materials provide the detailed world-model foundation. Other leaderboard configurations, alignment experiments, and the GPT6 study remain proposed. Planned coverage, implementation readiness, and completed measurements are reported separately.

1.6 Related work

Writing owner: 高达辉 and 史可为 **To complete:** Coordinate the five assignments below and consolidate the coverage comparison supplied by each owner.

This section first reviews world models, embodied manipulation models, and vision–language models, then examines Physical AI benchmarks and the connections needed for unified evaluation. The benchmark review compares tested interfaces, task and data scope, metrics, evaluation conditions, and capability coverage. Benchmark versions, official leaderboard links, and access dates accompany the final comparison; citation does not imply integration into REAL.

1.6.1 World Models

Writing owner: 史可为 **To complete:** Review world-model approaches, representations, conditioning, memory, and uses in planning; distinguish environment prediction from policy optimization.

World models learn representations of environmental states and their evolution to support prediction and action selection. Approaches differ in whether they predict latent states, video, or visual representations, whether predictions are conditioned on actions, and how they support policy learning or online planning.

Latent dynamics and learning in imagination. DreamerV3 learns latent transitions, rewards, and continuation probabilities from interaction trajectories, then trains a behavior policy and value function on imagined trajectories produced by the model [20]. Moving part of policy learning into the internal model allows interaction experience to be reused. Prediction errors can affect imagined trajectories and value estimates, so evaluation needs both multistep prediction checks and policy returns in the actual environment.

Generative video world modeling. Cosmos provides a Physical AI world foundation model platform with video processing, visual tokenizers, and autoregressive and diffusion models, supporting post-training for specific settings [1]. Predicted videos provide observable futures for research into scene expansion and interaction consequences. Action evaluation additionally requires checking how controls condition the model and whether different actions from the same initial state produce the corresponding physical changes. Visual realism and accurate action consequences are distinct objectives.

Representation prediction and online planning. V-JEPA 2 first learns video representations through action-free self-supervision, then uses robot trajectories to train an action-conditioned predictor, V-JEPA 2-AC [3]. During planning, it predicts future representations for candidate action sequences and evaluates their distance to a goal-image representation. The system executes one action before observing and replanning. This connects video pretraining

to closed-loop control while making long-horizon prediction error, goal-representation quality, and planning cost separate evaluation concerns.

World models describe possible state changes following candidate actions, whereas policies select the actions to execute. They may be trained jointly or integrated within one system, but predictive quality and downstream task benefits still need separate reporting. REAL therefore records conditioning inputs, state and memory representations, prediction horizons, action interfaces, and planning uses before selecting tasks that match the observable outputs.

1.6.2 Embodied Manipulation Models

Writing owner: 刘璐 and 侯祥锐 **To complete:** Review manipulation policy and VLA approaches, training and adaptation, and observation/action interfaces in simulation.

Writing owner: 郭美衫 **To complete:** Supply physical robot implementations, embodiment adaptation, and the roles of external skills and controllers.

Embodied manipulation models map vision, language, and robot state to action commands. Approaches extend from feedback control with explicit geometry to learning action distributions from demonstrations and using vision–language pretraining for broader semantic and task knowledge.

Visual feedback control. Classical visual servoing constructs control signals from the error between current visual features or estimated poses and their targets, correcting motion through repeated observation [9]. Image-based and position-based approaches define errors in image-feature and three-dimensional task spaces, respectively. These methods provide explicit mechanisms for reaching targets and tracking trajectories; open-ended instruction understanding and task decomposition generally belong to higher-level modules.

Generative action policies. Diffusion Policy models visually conditioned action generation as a denoising diffusion process. It learns potentially multimodal action distributions from demonstrations and combines action-sequence prediction with receding-horizon execution [10]. Sequence generation directly models temporal relationships across a motion segment. Execution depends on the action horizon, observation-update frequency, and sampling cost, which must be reported together with the underlying controller.

Vision–language pretraining and action learning. OpenVLA combines vision–language pretraining with robot demonstrations, discretizes continuous actions into tokens, and generates commands autoregressively. It also provides adaptation methods for target robots and tasks [26]. π_0 instead adds an action expert to a pretrained vision–language model, using conditional flow matching to generate continuous action chunks from images, language, and robot state [6]. These approaches exemplify discrete action tokenization and continuous action generation. Comparisons require aligned action spaces, execution frequencies, and adaptation budgets.

Heterogeneous co-training and generalization. Building on π_0, $\pi_{0.5}$ combines multi-robot data, web data, high-level semantic subtask prediction, and low-level action learning to study knowledge transfer for long-horizon manipulation in unseen environments [47]. This direction extends evaluation beyond success on familiar tasks to capability retention across changes in objects, scenes, tasks, and embodiments. REAL therefore records pretraining and post-training data, action representations, task splits, and external controllers, using separate simulation and physical protocols to examine adaptation and execution.

1.6.3 Vision–Language Models

Writing owner: 武若楠 **To complete:** Review VLM approaches to grounded perception, spatial and physical reasoning, and planning; explain the interface between model decisions and downstream execution.

Vision–language models support environment understanding, spatial reasoning, and task-level decisions, connecting to physical execution through skills or control interfaces. The central objective is to ground language knowledge in current observations so that answers, targets, and plans reflect the environment.

Fusing embodied observations with language. PaLM-E encodes vision and continuous sensor states as embeddings alongside text, using multitask training to study visual question answering, embodied reasoning, and robot task planning [16]. This unified input allows instructions to be interpreted in the current state, while downstream policies execute task-level plans. Semantic correctness, environmental feasibility, and execution success therefore require separate evaluation.

Spatial reasoning for manipulation. RoboRefer incorporates depth information for spatial referring in robotic scenes and combines supervised training with reinforcement learning for multistep spatial reasoning and target localization [63]. This grounds abstract language requirements in spatial locations that can serve as targets for grasping or navigation modules. Evaluation should examine localization error and reasoning stability under occlusion, viewpoint changes, and composed relations, while separating the contributions of localization and control to the final outcome.

Language planning with skill feasibility. As a hierarchical decision method, SayCan combines a language model's assessment of a skill's relevance to the task with feasibility estimated by its value function, selecting the next action from an existing skill library [2]. It shows how embodied skills ground language knowledge in the environment, with low-level motion supplied by those skills. Evaluating embodied systems that use VLMs or language models therefore requires recording visible observations, available skills, execution feedback, and replanning rules to attribute improvements to the reasoning model or its surrounding components.

1.6.4 Physical AI Benchmarks

Writing owner: 史可为 **To complete:** Compare world-model benchmarks and leaderboards: prediction horizons, physical consistency, action conditioning, memory, and planning evidence.

World-model benchmarks. VBench measures video generation quality, including subject consistency and motion smoothness [22]. WorldModelBench evaluates instruction following and physical adherence in generated videos [28]. HappyWorld-Bench organizes video, spatial, and embodied world-model tracks under a shared capability framework [4]. Its embodied track evaluates generated action-conditioned videos, making its evidence distinct from executed robot actions. Our comparison examines observable prediction quality, declared control interfaces, and downstream utility; visual consistency alone does not establish action-conditioned accuracy or planning performance.

Writing owner: 刘璐 and 侯祥锐 **To complete:** Compare simulated manipulation benchmarks and leaderboards: task diversity, interfaces, adaptation budgets, generalization splits, success criteria, and simulation-to-real evidence.

Simulated manipulation benchmarks. LIBERO studies knowledge transfer in lifelong robot learning [32]; RoboCasa supplies simulation environments for everyday manipulation [38]. SIMPLER studies evaluation of real-world policies in simulation and reports paired simulation and physical evaluations [30]. Our review treats task suites, observations, controllers, adaptation allowances, and success definitions as part of each result. Existing simulation-to-real studies also require REAL to establish the validity of its own matched settings.

Writing owner: 郭美衫 **To complete:** Compare physical manipulation benchmarks and leaderboards: embodiments, task/object coverage, resets, trial counts, intervention, efficiency, reliability, and execution evidence.

Real-world manipulation benchmarks. RoboChallenge develops an online real-robot evaluation system and compares embodied policies using its Table30 benchmark [58]. Our comparison records hardware and camera configurations, controller interfaces, initial-state variation, intervention rules, and the trials underlying reported outcomes. Physical results establish evidence under those conditions; transfer to other robots, tasks, or environments requires additional evaluation. Model-level results and results of the complete policy stack must also be distinguished.

Writing owner: 武若楠 **To complete:** Compare VLM and embodied reasoning benchmarks and leaderboards: perception, spatial/physical reasoning, planning, answer scoring, and executable decisions.

VLM and embodied reasoning benchmarks. MMBench evaluates multimodal abilities through a structured bilingual benchmark [33]; MMMU emphasizes multimodal understanding and reasoning with subject knowledge [61]. EmbSpatial-Bench evaluates egocentric spatial understanding in embodied scenes [17]. These benchmarks supply distinct evidence about perception and reasoning. This report separately tests answer correctness, feedback-based decision making, and physical execution to examine how these forms of evidence relate.

1.6.5 Unified Physical AI Evaluation

Writing owner: 高达辉 and 史可为 **To complete:** Synthesize the model and benchmark reviews into a cited coverage comparison. Each leaderboard owner supplies evidence for covered, partially covered, and unexamined capabilities and the corresponding REAL tasks.

From these differences in evaluation interfaces, we identify three connections to examine: reasoning to executable decisions, predicted dynamics to useful control, and simulated outcomes to physical outcomes. This is our analytical framing for the comparison, rather than a claim that every existing benchmark omits these connections. Owners will check long-horizon memory, causal and action-conditioned prediction, recovery and adaptation, generalization, human–machine coexistence, safety and reliability, and application constraints against the five-domain capability system. A capability absent from a benchmark is unexamined by that benchmark; it is not thereby shown to be absent from the model.

The comparison will distinguish explicitly tested coverage, partial evidence, and unexamined coverage, with a source and protocol for each judgment. REAL is positioned as a shared capability–task–metric system with four interface-specific leaderboards and matched alignment studies. Proposed tasks, implemented protocols, and completed measurements remain separate, so the final related-work comparison will state which gaps the current release actually tests and which remain planned.

2 REAL Physical AI Evaluation System

2.1 The complete capability evaluation system

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Verify the full hierarchy and supply capability definitions and task mappings for each board.

REAL treats its evaluation system as a shared methodological foundation for all four leaderboards. The REAL presentation specifies 5 first-level domains, 14 second-level dimensions, 55 third-level abilities, and 158 capability indicators [5]. This report uses level-4 assessment points to operationalize indicators and level-5 observations to specify task evidence. The presentation supplies the capability structure and indicator totals; the complete indicator inventory and formulas remain to be supplied. The hierarchy describes what the benchmark seeks to measure; task design, scoring formulas, and benchmark sources make those constructs executable.

2.1.1 Design logic of the five domains

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide evidence and examples for the organizing questions without inferring completed coverage.

The five domains organize complementary questions about capability and deployment. Table 1 relates these questions to the supplied second-level composition. The questions are an explanatory summary of the domain definitions.

Basic intelligence concerns the chain from understanding the environment to predicting consequences, deciding what to do, and carrying out motion. Its world understanding, world prediction, autonomous decision-making, and embodied-motion branches connect cognition to action.

Adaptive learning concerns performance as conditions or experience change. Generalization and transfer examines reuse across instructions, objects, scenes, tasks, skill compositions, and embodiments. Learning and evolution examines continued acquisition, exploration, memory, and reflection.

Human–machine coexistence concerns operation with people and other robots. Multi-robot collaboration, human–robot collaboration, and interaction connect task allocation and coordinated motion to behavior understanding, social norms, preference alignment, dialogue, and nonverbal expression.

Safety and reliability concerns risk prevention and performance retention. Safety covers physical safety, cognitive safety, proactive safety, and adversarial attacks. Robustness concerns performance under hardware, task, and temporal changes.

Application practice concerns usable capability in a deployed system. It connects knowledge and skill coverage to deployment effort, adaptation, training and inference resources, portability, and stability across tasks and training runs, as well as long-term operation. This domain makes operational cost and consistency explicit.

Table 1. Five capability domains and their organizing questions.

Level 1	Organizing question	Level-2 composition
Basic intelligence	Can the system understand, predict, decide, and act?	World understanding; World prediction; Autonomous decision-making; Embodied motion
Adaptive learning	Can the system transfer capabilities and continue learning?	Generalization and transfer; Learning and evolution
Human–machine coexistence	Can the system cooperate and communicate with people and other robots?	Multi-robot collaboration; Human–robot collaboration; Human–robot interaction
Safety and reliability	Can the system operate safely and reliably under risks and disturbances?	Safety; Robustness
Application practice	Can the system be deployed with practical resources and deliver stable service?	Skill and knowledge coverage; Application efficiency; Application stability

Questions summarize the supplied domain definitions. The five domains describe the complete evaluation system; a release may schedule only a subset.

2.1.2 Five levels from capability to observable evidence

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Check hierarchical definitions, observational points, and the correspondence between constructs and evidence.

Level 1 specifies the capability domain; level 2 organizes related dimensions; level 3 states the ability and its meaning; level 4 identifies a testable assessment point; level 5 provides observable aspects or controlled variations for that point. The supplied definition column constrains task interpretation at level 3. Table 2 explains the levels through an illustrative path, and the appendix index in table 21 lists all level-3 abilities compiled from the presentation.

Table 2. Five levels of the REAL capability system.

Level	Scale	Role and path example
1: Domain	5	Defines the overall goal; e.g., basic intelligence.
2: Dimension	14	Organizes related capabilities; e.g., world understanding.
3: Ability	55	Defines the construct and its meaning; e.g., causal understanding.
4: Assessment point	158	Specifies a testable aspect; e.g., logical causality.
5: Observation	Per point	Specifies observable evidence or variations; e.g., failure diagnosis and causal attribution.

Counts follow the REAL presentation. Level 4 uses its declared indicator total; the complete inventory and formulas remain to be supplied. The path illustrates task design using the legacy table.

The 158 indicators are the total declared in the presentation, not that many completed metric definitions or implementations. Owners must supply indicator names and capability assignments, task construction, scoring and aggregation rules, and data sources. The legacy indicator inventory remains a reference for task design and cross-version mapping.

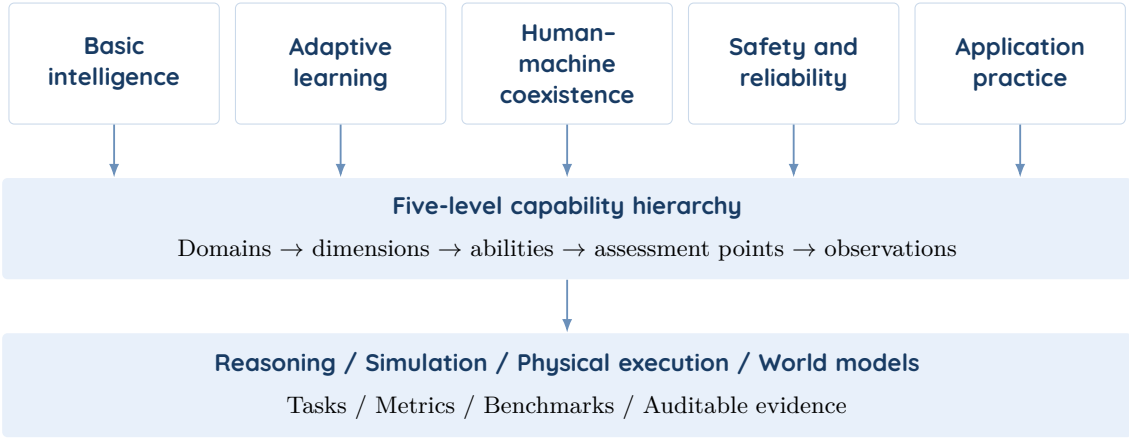


Figure 1. The complete REAL capability system. Five capability domains are operationalized through a five-level hierarchy and mapped to four evaluation interfaces, with task, metric, and evidence specifications supporting execution.

2.1.3 Embodied reasoning evaluation system

Writing owner: 武若楠 **To complete:** Confirm selected capability paths, eligible answer interfaces, and the evidence and aggregation rules for reasoning.

Counting conventions. The PPT reasoning projection contains 33 level-3 abilities and 84 indicators (table 5). The 28 level-4 points and 47 ability groups below describe the existing question collection; their mapping to the PPT taxonomy still requires item-level verification.

Existing question coverage. The reasoning board covers 28 level-4 evaluation points across two level-1 domains. Under basic intelligence, it tests world understanding (instruction comprehension, object properties and affordances, spatiotemporal and causal reasoning, physical

laws), world prediction (kinematic, dynamic, scene, and spatial-geometric consistency), and autonomous decision-making (task planning, feasibility judgment, error recovery). Under adaptive learning, it tests reflection, autonomous exploration, and short-term memory. Human–machine coexistence, safety, reliability, and application practice are not scheduled for this board.

Scoring. Approximately 95% of questions are scored automatically: multiple-choice and true/-false items by direct answer comparison, numerical items (e.g., distance and size estimation) on a 0–1 scale via relative error, and pointing or localization items by checking whether the response falls within the target region. The remaining approximately 5% (570 questions from three data sources) are scored by an LLM judge.

Aggregation. Each ability's score is the mean per-question accuracy across all its questions, regardless of data source. The overall score is the equal-weight average of all 47 ability scores—not weighted by question count, since question counts (22–847 per ability) reflect data availability rather than importance. Level-1 through level-4 node scores are likewise equal-weight averages of the ability scores under each node, used for diagnostic profiling; they are not further aggregated upward into the overall score, because the hierarchy is unbalanced (e.g., adaptive learning contains only 3 abilities versus 44 under basic intelligence). If a model lacks any ability score, all nodes containing that ability and the overall score are left blank.

2.1.4 Simulated manipulation evaluation system

Writing owner: 刘璐 and 侯祥锐 **To complete:** Confirm the simulation capability map, observable policy interfaces, success predicates, and comparison groups.

Measurement scope. The simulation projection measures capabilities through closed-loop manipulation. The scheduled scope connects instruction following, object and spatial understanding, and manipulation to held-out object, background, and task conditions, goal changes, and deployment-efficiency measurements. The full taxonomy remains the reference for selecting and explaining these points.

Evidence unit. An episode links a task and initial-state distribution to policy observations, issued actions, simulator transitions, terminal state, and an executable success predicate. Intermediate subgoals and constraint checks support diagnosis without replacing the declared task outcome.

Reporting structure. Separate zero-shot and adapted policies, observation privileges, embodiments, and action interfaces. Report task-family and capability results alongside adaptation and resource budgets. Map every scored measurement to a declared point and disclose shared tasks across points so that repeated use of one rollout does not silently duplicate its contribution.

2.1.5 Real-world manipulation evaluation system

Writing owner: 郭美衫 **To complete:** Confirm the real-world capability map, hardware groups, autonomous-success rules, and safety and deployment measurements.

Measurement scope. The physical-execution projection tests whether capabilities produce verified outcomes on a declared robot. Alongside manipulation and scheduled generalization, it records physical and interaction safety, goal-change adaptation, and deployment-efficiency

requirements from the supplied system. Coverage is tied to the tasks and hardware actually available.

Evidence unit. A physical trial connects a measured initial state, instruction, policy and controller configuration, synchronized observations and actions, and independently checked final state. Intervention, stopping, reset, and fault records accompany the outcome. Completion after human assistance is distinguished from autonomous success.

Reporting structure. Compare submissions within frozen robot, sensor, controller, information-access, and adaptation groups. Report success, constraint violations, intervention, time, and resource measurements with their own denominators. A linked simulation trial can support alignment analysis, while the physical trial supplies the evidence for this leaderboard.

2.1.6 World-model evaluation system

Writing owner: 史可为 **To complete:** Confirm the prediction capability map, control-interface eligibility, criterion evidence, and task-to-score-item attribution.

Measurement scope. The prediction projection tests instruction-conditioned state evolution, object and scene persistence, physical and geometric consistency, and the scheduled generalization and safety points. Each probe connects a full capability path to a visible event, state change, or controlled comparison.

Evidence unit. Conditioning observations, instructions, supported controls, and their timestamps define the requested prediction. The generated video or state sequence is checked for output validity, control realization, and criterion-specific evidence. Visual plausibility alone does not establish that an action-conditioned physical response occurred.

Reporting structure. Keep robot-action, camera or interactive-control, and language-conditioned comparisons distinct. The current release uses the 33-item, 3,300-point additive rule in Section 2.9.4; these are scoring items rather than task or rollout counts. Report evidence coverage, uncertainty, validity, and generation cost alongside the score, preserving the complete taxonomy above this release view.

2.2 Evaluation subjects

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply model rosters, interfaces, eligibility, and system components.

Model families and leaderboards describe different attributes: a family denotes the declared model interface, whereas a leaderboard denotes the evidence collected. A model enters a leaderboard only after satisfying its input and output contract. Tables 3 and 4 define the roles and four evaluation settings. Model-only, policy-stack, and whole-robot submissions record the external planner, controller, perception, and recovery components that contribute to the outcome.

Table 3. Model-interface roles used in the evaluation design.

Type	Expansion	Role to evaluate
VLM	Vision-Language Model	Grounded visual understanding, reasoning, and instruction alignment
VLA	Vision-Language-Action Model	Instruction-to-action mapping and embodied execution
WAM	World Action Model	Action-conditioned prediction and physical state evolution
WM	World Model	Physical regularities and long-horizon state prediction

Roles follow the supplied outline. Eligibility follows the actual submitted interface; a family name alone does not establish leaderboard coverage.

Table 4. Four leaderboards and their evaluation interfaces.

Leaderboard	Input and required output	Primary outcome
Embodied reasoning	Observations and instructions → grounded answers, decisions, or plans	Grounded correctness
Simulated manipulation	Simulated observations and instructions → closed-loop rollouts	Simulated task success
Real-world manipulation	Sensor streams and instructions → actions on a physical robot	Physical task success
World models	Initial observations and controls → predicted observations or states	Additive capability score

Primary metrics and eligibility are frozen separately for each leaderboard.

The presentation specifies the four leaderboard projections in table 5. The world-model projection contains 52 level-3 abilities and 153 indicators. These projections describe evaluation design, while implementation and measured coverage require separate checks against execution records.

Table 5. Leaderboard projections specified in the REAL presentation.

Leaderboard	Domains	Dimensions	Abilities	Indicators
Reasoning (REAL-ER)	5	10	33	84
Simulation (REAL-MS)	5	13	47	131
Physical execution (REAL-MR)	5	13	51	141
World models (REAL-WM)	5	13	52	153

Source: slides 14, 20, 25, and 26. These are framework projections, not measured coverage or task counts. Indicator totals are stated in the slides; a complete indicator inventory and formulas remain to be supplied.

Table 6 preserves the November schedule from the October 22, 2026 table to trace existing tasks and world-model scoring settings. Its level-4 points use a different counting unit from the new abilities and indicators; their item-level mapping remains to be established. The 33-item, 3,300-point world-model rule does not define the size of the new taxonomy. Owners must confirm interface eligibility and task mappings before freezing the executable set.

Table 6. Legacy November schedule: level-4 points by domain and leaderboard.

Capability domain	Reasoning	Simulation	Physical	World models
Basic intelligence	22	8	8	18
Adaptive learning	2	3	3	11
Human–machine coexistence	0	0	0	0
Safety and reliability	0	1	7	4
Application practice	0	9	9	0
Total	24	21	27	33

Counts are derived from the 2026-10-22 table, pending mapping to the PPT taxonomy. A point may occur in several leaderboards. They are neither task counts, measured coverage, nor score ceilings. Zero means not listed for November, without negating theoretical applicability.

2.3 Evaluation methods

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Specify executable checks, judges, human review, and evidence requirements.

Benchmark, simulation, and physical evaluation form complementary evidence paths. Reasoning tasks check grounded answers and plans, manipulation tasks measure closed-loop execution, and world-model tasks evaluate controlled predictions. Each task preserves its complete capability path and definition. Use exact or executable checks when reference information permits; otherwise use frozen criterion-level judgments with timestamped evidence and calibrated human review. Matched cross-setting analysis follows Chapter 7. Table 7 traces basic intelligence, world understanding, causal understanding, logical causality, and failure diagnosis to board-

specific tasks and evidence. A shared construct can use different outputs; grading must respect the submitted interface.

Table 7. One capability path connected to board-specific evidence.

Leaderboard	Example task and evidence design	Writing owner(s)
Reasoning	Identify a failure cause from observations and compare the answer with a frozen causal rubric.	武若楠
Simulation	Use controlled interventions, state logs, and recovery outcomes to test the same causal hypothesis.	刘璐 and 侯祥锐
Physical execution	Verify the causal hypothesis through matched physical trials with recorded resets and interventions.	郭美衫
World models	Compare predicted responses to matched interventions, recording timestamped evidence and validity.	史可为

Shared path: basic intelligence → world understanding → causal understanding → logical causality → failure diagnosis. These are task-design examples; the owners must supply executable protocols, metrics, and benchmarks before experiments.

2.4 Metric system

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply metric definitions, formulas, units, denominators, and applicability.

For each level-4 point, owners specify the question or interaction format, grading method, metric and denominator, aggregation, and adopted or newly built benchmark. Outcome and process measurements are reported together. Success rate (SR) uses the frozen binary success predicate and denominator in Section 2.9. The presentation defines PSR as average process success rate, capturing partial progress through completed task stages. TEQ denotes task execution quality, combining time efficiency, path efficiency, and trajectory smoothness relative to human demonstrations on successful tasks or successfully completed stages. Stage definitions and scores for PSR, and component normalization and weights for TEQ, still need to be frozen before these metrics enter official rankings. Application measurements include execution time, inference latency, intervention frequency, and adaptation data, compute, and wall-clock cost. Learning efficiency is assessed across declared adaptation budgets; time endpoints and resource units must be specified. These measurements accompany capability scores without silently mixing cost and accuracy into an undeclared scalar.

2.5 Evaluation tasks

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply frozen task inventories, difficulty rules, and capability assignments.

Task specifications record identifier, version, leaderboard, complete capability path and definition, instruction, observations, controls, budget, success criteria, and evaluator configuration. Desktop, workstation, and scene-scale tasks describe operating scope, while declared horizon, precision, and interaction requirements describe difficulty. Scope alone does not establish a difficulty ranking. Matched tasks share a concept identifier and disclose remaining differences. New tasks require validation and a new release manifest before entering a score.

2.5.1 Embodied reasoning tasks

Writing owner: 武若楠 **To complete:** Supply question families, grounded references, difficulty factors, example cases, and validated task counts.

Task families. Specify single-observation understanding and grounding, video-based temporal and causal judgments, history-dependent memory questions, and structured planning or failure-recovery problems. Instruction probes can vary negation, conditions, and referring expressions; spatial or physical probes must identify the evidence that determines a valid response.

Task specification. A question records its source observation, permitted history, instruction, response schema, answer or criterion references, and capability path. For multi-turn tasks, fix the turn sequence, state available at each turn, information-request permissions, and stopping rule. Planning tasks declare subgoal dependencies and constraints that can be checked from the input.

Controlled variation. Describe difficulty through distractors, visibility, temporal extent, causal depth, and plan length. Keep paraphrases, counterfactual variants, and questions derived from one episode linked to a common source identifier. Review answerability and ambiguity before freezing the set; when the observation is insufficient, define the accepted uncertainty or abstention response instead of requiring an unsupported guess.

2.5.2 Simulated manipulation tasks

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply executable tasks, initialization distributions, success functions, variation factors, and episode budgets.

Task families. Candidate families include basic grasping, placing and pushing, precise assembly or insertion, continuous multi-step manipulation, and mobile manipulation where the embodiment supports it. The frozen inventory identifies the implemented families and their capability paths.

Task specification. Define the robot and objects, initial-state distribution, instruction, permitted observations, action interface, subgoals, terminal success predicate, constraints, horizon, reset, and failure conditions. An episode starts from a sampled initial state and ends under the declared success, failure, or timeout rule. Record intermediate progress separately from the final outcome.

Controlled variation. Construct matched variants for scheduled object shape, background, new task types, and goal-change conditions. State the event time and permitted replanning for goal changes. Distinguish single-factor shifts from joint shifts; fix seeds and repeat budgets before comparing policies. Training tasks, adaptation examples, and held-out evaluation episodes are enumerated independently.

2.5.3 Real-world manipulation tasks

Writing owner: 郭美衫 **To complete:** Supply physical task cards, initial-state tolerances, success checks, trial budgets, and matched simulation identifiers.

Task families. Instantiate the available basic, precise, long-horizon, and mobile-manipulation operations on the actual hardware. Task cards identify the object instance, goal configuration, physical constraints, and capability path; a task family is included only when its execution and outcome can be checked.

Task specification. Record workspace geometry, object placement and orientation tolerances, instruction, observation and action permissions, maximum duration, and final-state verification. Each trial includes an initial-state check, autonomous execution, outcome inspection, and reset record. Freeze the treatment of drops, constraint violations, policy stops, human interventions, and infrastructure faults before running the task.

Controlled variation. Use documented object and background variants, new task types, and goal-change events within the scheduled scope. Preserve repeat identifiers and trial order so that reset drift can be analyzed. For alignment studies, attach the corresponding simulation concept and task identifiers and record the physical properties or execution conditions that remain unmatched.

2.5.4 World-model tasks

Writing owner: 史可为 **To complete:** Supply conditioning packets, control timelines, criterion evidence, valid-output checks, and paired task variants.

Task families. Design prediction probes for instruction compliance, object persistence through occlusion or revisits, scene and geometric consistency, contact and motion evolution, and scheduled generalization and safety conditions. Each probe specifies which observable event would support its capability claim.

Task specification. Record initial observations or history, instruction, control interface, timestamped control sequence, prediction horizon, expected events, scoring criteria, and available reference measurements. Distinguish tasks with synchronized reference trajectories from those judged by explicit event or consistency criteria. Each task maps to its declared scoring item.

Controlled variation. Where an interface permits it, compare matched initial conditions with one changed instruction, action, viewpoint, object, or scene factor. Check realization of the requested intervention before interpreting its consequences. Declare the occlusion, departure, and revisit intervals for persistence probes. Repeated seeds estimate prediction variation and remain repeat runs of the same task; legitimate stationary intervals follow the task protocol rather than a universal motion requirement.

2.6 Evaluation workflow

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide adaptation, inference, reset, evaluation, and audit workflows.

Register models and allowed interfaces, freeze tasks and budgets, run validity checks, execute, score, audit evidence, and publish status counts with results. Reasoning uses the frozen benchmark interface. Simulation and physical policies declare whether post-training occurs and separate it from held-out evaluation; physical trials additionally define deployment and initial-state restoration. World-model evaluation preserves conditioning and control timelines through generation and scoring. Zero-shot and adapted submissions are separate settings.

2.7 Evaluation environments

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide actual hardware or simulator versions, APIs, controls, and dependencies.

An environment specification separates inference, observations, controls, and execution or evaluation resources. The four interfaces below require different configurations; each result links to its frozen environment manifest. Actual hardware, software revisions, and integration status are supplied by the responsible owner.

2.7.1 Embodied reasoning environment

Writing owner: 武若楠 **To complete:** Supply inference hardware or API revisions, media processing, prompts, tool permissions, and timing settings.

Inference environment. Separate local-weight and API submissions. Record the checkpoint or service revision, inference software, available compute, numerical precision, context limit, decoding parameters, and request budgets. Fix system prompts, answer schemas, retry rules, and timeout handling within a comparison group.

Observation environment. Freeze image preprocessing and video decoding, resolution, frame selection and timestamps, and the amount of history visible to the model. State-assisted, tool-assisted, and visual-only settings are identified separately. Multi-turn cases define session initialization, memory retention, permitted information requests, and termination.

Reproducibility. Store processed-input fingerprints, prompts, responses, tool traces, and request or inference timing where permitted. Keep answer keys and evaluator-only annotations outside model inputs. The environment manifest identifies the settings needed to replay a local run or audit a service call.

2.7.2 Simulated manipulation environment

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply simulator builds, assets, robot and controller configurations, physics parameters, seeds, and compute hardware.

Simulator and embodiment. Record the simulator and task-suite versions, asset revisions, robot geometry, joint and actuator settings, gripper, controller, and action conventions. Isaac Sim is a candidate from the supplied outline; the actual adopted simulator and integration status require confirmation.

Physics and observations. Freeze simulation time step, solver settings, control rate, contact and friction parameters, and reset distributions. Record camera placement, rendering and image settings, proprioception, and any privileged state. Align controller limits and observation access within comparison groups; separate training randomization from the held-out test distribution.

Execution environment. Record compute hardware, inference precision, the configuration of parallel environments, policy scheduling, observation delay, and synchronous or asynchronous stepping. Preserve environment and reset seeds, configuration snapshots, success-function versions, and transition logs. Nominally identical seeds are accompanied by measured run variation rather than an assumption of deterministic simulation.

2.7.3 Real-world manipulation environment

Writing owner: 郭美衫 **To complete:** Supply actual robot and sensor models, calibration, controllers, deployment topology, workspace, and reset specifications.

Hardware and workspace. Identify robot, gripper, end effector, sensors, object instances, and workspace geometry. The outline's Franka, Songling, G1, and gripper configurations are candidate leads; exact configurations and availability remain to be supplied. Record camera intrinsics and extrinsics, robot-to-camera calibration, sensor synchronization, and initial-state tolerances.

Control and deployment. Fix coordinate frames, units, command bounds, control rate, action horizon, interpolation, observation delay, and the lower-level controller. State which components run on the robot, edge host, or remote service, including communication and latency handling. Record operating limits, stop and intervention mechanisms, and permissions for automatic recovery.

Execution environment. Freeze reset and outcome-inspection procedures, trial ordering, object replacement rules, and operator responsibilities. Log calibration changes, maintenance events, hardware faults, and deviations from the initial-state tolerance. These records explain differences between physical repeats and support matched simulation comparisons.

2.7.4 World-model environment

Writing owner: 史可为 **To complete:** Supply generation backends, weights or service revisions, control adapters, output specifications, and evaluator environments.

Generation environment. Record model and code revisions, weights, hardware, precision, sampling settings, random seeds, and inference scheduling. Specify image preprocessing, history length, output resolution, frame rate, duration, and any chunked or autoregressive generation. Model-specific settings are disclosed rather than silently standardized through unsupported options.

Control environment. Version the adapter that converts benchmark actions or camera commands into native inputs. Record coordinates, units, timing, resampling, scaling, and any approximation. The manifest identifies supported control groups and preserves the original and translated control timelines.

Evaluation environment. Separate generation from reference measurements and scoring. Freeze video decoding, frame sampling, metric models, judge configuration, and calibration sets. When a simulator or recorded trajectory provides a reference, identify its version and alignment procedure. Future reference states and private criteria are evaluator inputs and do not become additional conditioning information.

2.8 Evaluation data

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide sources, licenses, splits, task counts, and submission records.

Data specifications distinguish model-visible inputs, adaptation data, evaluator-only references, and generated evaluation artifacts. Every source records rights, collection or generation procedures, identifiers, grouped splits, and quality checks. The inventory summarizes board-specific units; the following subsections define their content and statistics. Candidate sources and historical repository counts do not establish released task volume.

> **No results are populated in this draft.** TBD denotes an unconfirmed configuration; ··· denotes a result that has not been measured or supplied. Rows labeled "Model A/B/C" are placeholders and do not identify evaluated models. All tables require verified artifacts before supporting empirical claims.

The evaluation snapshot should contain an immutable task manifest, source and license records, split assignments, metric configurations, and the adapter versions used to generate results. Report both task count and evaluation episode count: repeated trials, paired interventions, and multi-round tasks can make these quantities differ substantially.

Table 8. Leaderboard-specific data units, split groups, and release statistics.

Leaderboard	Data unit	Split groups	Counts to report
Embodied reasoning	Media, question, answer and evidence	Source episode, scene, object, template	Unique media, questions, multi-turn cases, responses
Simulated manipulation	Scene configuration, demonstration, rollout	Asset family, scene, template, parameters	Tasks, assets, demo episodes and steps, test episodes
Real-world manipulation	Capture session, demonstration, physical trial	Session, object, scene, embodiment	Tasks, sessions, demo duration, trials and statuses
World models	Conditioning packet, controls, reference evidence	Source scene, initial state, sequence, template	Tasks, conditions, control variants, seeds, generated sequences

Owners supply values from the frozen manifest, by split and capability. Task, sample, demonstration, and evaluation-run counts are distinct; repeated runs and invalid statuses remain explicit.

Each submission record includes its model identifier, checkpoint or service revision, parameter count when available, training-data disclosure, adaptation regime, input modalities, compute configuration, and inference budget. Baseline selection should cover the interfaces present in each leaderboard. Related-work citations do not serve as a confirmed baseline roster.

2.8.1 Embodied reasoning data

Writing owner: 武若楠 **To complete:** Supply media and question sources, answer annotations, grouped splits, licenses, and sample statistics.

Composition and provenance. A sample links an image, video segment, or observation history to a question, response schema, answer or rubric, and supporting evidence. Record benchmark or collection source, scene and object identifiers, timestamps, annotation procedure, and rights. Mark newly authored, adapted, and derived questions separately.

Splitting and quality. Group questions and variants by source episode, scene, object instance, and template before assigning development, calibration, and test splits. Define which factors are held out. Review answer consistency, grounding, ambiguity, and annotation agreement; keep reference answers and private evidence outside the submitted inference input.

Statistics. Report unique media sources, questions, multi-turn cases, and evaluation responses separately, with counts by split, capability, question format, and difficulty. Several questions derived from one video remain correlated samples for uncertainty estimation.

2.8.2 Simulated manipulation data

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply asset and scene sources, demonstrations, transition fields, held-out factors, and episode statistics.

Composition and provenance. Separate scene and asset configurations, demonstration episodes, adaptation data, and evaluation rollouts. An episode records observations, actions, available proprioception, timestamps, terminal flags, and outcome checks. Full simulator state can support the evaluator; its availability to the policy follows the declared observation setting. Track asset licenses, generation parameters, and suite revisions.

Splitting and quality. Define held-out asset families, scenes, task templates, and parameter ranges at their source level. Keep complete trajectories together and prevent evaluation seeds or hidden configurations from entering adaptation data. Validate resets, action bounds, time alignment, success predicates, and termination records before accepting episodes.

Statistics. Report tasks, scene and asset counts, demonstration episodes and steps, adaptation examples, and evaluation episodes by split and variation factor. Distinguish the number of task definitions from repeated seeds and parallel rollouts; include failed and unresolved runs in the release accounting.

2.8.3 Real-world manipulation data

Writing owner: 郭美衫 **To complete:** Supply collection sessions, demonstration and trial records, sensor synchronization, grouped splits, and usable-data statistics.

Composition and provenance. Separate demonstrations or adaptation trajectories from held-out physical trials. Candidate collection routes include leader–follower operation and demonstrations on declared non-test embodiments. Record session, robot and object identifiers, camera streams, calibration, proprioception, issued and measured actions where available, timestamps, intervention events, and verified outcomes. Identify unavailable sensor fields.

Splitting and quality. Declare the generalization claim and group by collection session, object instance, scene, or embodiment accordingly. Neighboring frames and repeated clips from one trajectory stay in the same split. Check sensor alignment, action units, calibration validity, missing segments, and outcome annotations. Record source rights and any restrictions on releasing recordings.

Statistics. Report tasks, collection sessions, demonstration trajectories and duration, adaptation examples, and physical trial counts. Give autonomous, intervened, invalidated, and unresolved trial counts separately, with the declared rules linking those statuses to metric denominators.

2.8.4 World-model data

Writing owner: 史可为 **To complete:** Supply conditioning and reference sources, control annotations, grouped splits, quality checks, and task versus generation counts.

Composition and provenance. Each conditioning packet contains initial observations or history, instructions, control signals and timestamps, and output requirements. Evaluator resources contain available reference recordings or states, expected events, and criterion annotations. Tag captured, simulated, and authored or generated sources. A generated initial image is a conditioning source and does not by itself provide physical ground truth.

Splitting and quality. Group shared source scenes, initial states, trajectories, assets, and task templates before assigning adaptation, calibration, and test splits. Keep private reference futures out of model inputs. Check media decoding, conditioning/control alignment, supported interfaces, reference provenance, and criterion consistency. Freeze the task snapshot and retain fingerprints linking canonical tasks, display copies, and inference inputs.

Statistics. Report task definitions, conditioning packets, control variants, repeated seeds, generated sequences, and total duration or frames separately. Break counts down by capability, control group, horizon, split, and validity status. The 33 scoring items describe coverage; the frozen manifest determines actual data and generation volume.

2.9 Leaderboards and scoring

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Confirm comparison groups, aggregate views, weights, and missing-result rules.

The four leaderboards provide an overall view and capability-specific views within each frozen comparison group. This implements the planned overall/sub-board presentation without equating scores across different interfaces. A future cross-leaderboard aggregate would require separate eligibility, missing-coverage, and weighting rules; it is not defined by the current release. The common accounting rules and world-model-specific additive rule follow below.

The November world-model table supplies the additive scoring rule for that leaderboard. Other rules below form a proposed common reporting contract; release-specific implementations must be checked before results are published.

2.9.1 Freezing the evaluation setting

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Confirm frozen tasks, versions, permissions, and budgets for each board.

Each release fixes task membership, split assignments, input permissions, model adaptation rules, inference budgets, success criteria, metric versions, and aggregation weights. A comparison group contains only submissions satisfying the same essential task interface and information constraints. Different robot hardware, privileged state access, or control interfaces are explicit grouping variables. Task and rubric revisions produce a new version and invalidate affected cached evaluations.

2.9.2 Run accounting and coverage

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide status mappings, failure policy, counts, and coverage denominators.

Record eligible runs before execution. Distinguish model outcomes from infrastructure faults and evaluator failures. The proposed status vocabulary is `scored`, `model_failure`,

`evaluation_error`, `infrastructure_error`, and `unsupported`. Existing backend statuses are mapped explicitly rather than assumed to use these names.

Let N_{eligible} denote the fixed eligible run count and N_{resolved} the runs with a valid outcome under the frozen failure policy. Report coverage as

$$C = \frac{N_{\text{resolved}}}{N_{\text{eligible}}}. \tag{1}$$

Task timeouts, malformed model outputs, and policy execution failures are retained as unsuccessful outcomes when prescribed by that policy. Evaluator and infrastructure errors require resolution or explicit missing-result disclosure. Interface eligibility is determined before the comparison group is frozen. Within the world-model schedule, a declared non-participating scoring item is retained with score zero, as required by the supplied scoring table. It cannot be dropped from the total or used to renormalize the score. A pending measurement is different from non-participation and leaves the result provisional. Publish status counts as well as score denominators.

2.9.3 Binary outcomes and second-level aggregation

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫 **To complete:** Freeze success predicates, task aggregation, and dimension weights for the three boards.

For a resolved binary trial r of task i, define $y_{ir} = \mathbb{I}[\text{the success predicate is satisfied}]$. The per-task success estimate is

$$\widehat{p}_i = \frac{1}{R_i} \sum_{r=1}^{R_i} y_{ir}, \tag{2}$$

where R_i is the number of resolved trials and is reported alongside the scheduled trial count. For the reasoning and manipulation leaderboards, a candidate reporting rule first averages tasks within each second-level dimension d, then combines dimensions using weights frozen before model comparison:

$$S_d = \frac{1}{|\mathcal{T}_d|} \sum_{i \in \mathcal{T}_d} s_i, \qquad S_{\text{board}} = \sum_{d \in \mathcal{D}} w_d S_d, \qquad \sum_d w_d = 1. \tag{3}$$

Here $s_i \in [0, 100]$ is the normalized task score, such as $100\widehat{p}_i$. The task sets in this expression must be complete for an official ranking. The weights for these three leaderboards remain to be finalized. This proposed task-macro rule does not replace the November world-model additive rule below. Publish micro averages as diagnostics when useful, and label them separately from the dimension-level report.

2.9.4 World-model additive scoring

Writing owner: 史可为 **To complete:** Confirm the scheduled item mapping and verify additive score implementation.

Let $q_{mj} \in [0, 100]$ be model m's score for underlying scoring item j, and let $\mathcal{J}_d$ be the fixed item set for second-level dimension d. In the order shown in table 9, the item counts are $(8, 10, 11, 4)$ and sum to 33. The dimension subtotal and overall score are

$$B_{md} = \sum_{j \in \mathcal{J}_d} q_{mj}, \qquad T_m = \sum_d B_{md}, \qquad T_{\max} = 3300. \tag{4}$$

Table 9. Second-level dimensions for the November world-model evaluation.

Dimension	What is evaluated	Max. score
World understanding	Grounded comprehension of instructions, objects, spatial relations, physical knowledge, and causes	800
World prediction	Coherent evolution of motion, interaction, scenes, and spatial geometry	1,000
Generalization and transfer	Performance under changes in instructions, objects, scenes, and skill combinations	1,100
Safety	Physical operating constraints and ambiguous human instructions	400
Total	Fixed November scoring scope	**3,300**

This score table summarizes the four scheduled subtotals over 33 underlying items. The complete capability system is defined in Chapter 2; scheduled scope does not indicate completed evaluation.

Intermediate hierarchy levels add their descendants without taking a mean. Thus, the four second-level ceilings are 800, 1,000, 1,100, and 400. These are the only capability subtotals displayed in the main result table.

If a percentage view is needed, normalize by the corresponding fixed ceiling:

$$P_{md} = 100\frac{B_{md}}{100|\mathcal{J}_d|}, \qquad P_m = 100\frac{T_m}{3300}. \tag{5}$$

The normalized overall score therefore weights dimensions in proportion to $(8, 10, 11, 4)$, rather than averaging the four percentages equally. Under the source convention, a hyphen indicates declared non-participation and contributes zero; an empty result is pending and must not be silently converted to zero. Draft result cells use $\cdots$ to avoid conflating these states.

2.9.5 Criterion-level evidence and uncertainty

Writing owner: 史可为 **To complete:** Provide rubric weights, timestamped evidence, unknown handling, and bounds.

The earlier world-model evidence design distinguishes `pass`, `fail`, and `unknown` for individual criteria. In its evidence-reporting setting, judges return criterion status and timestamped evidence, while deterministic code computes scores. The following bounds are a candidate evidence report, not the current answers-mode scoring implementation. For task i, let $a_{ik} \geq 0$ be the weight of criterion k and $A_i = \sum_k a_{ik} > 0$. Define

$$L_i = \frac{100}{A_i}\sum_k a_{ik}\mathbb{I}[z_{ik} = \text{pass}], \tag{6}$$

$$U_i = L_i + \frac{100}{A_i}\sum_k a_{ik}\mathbb{I}[z_{ik} = \text{unknown}], \tag{7}$$

$$M_i = \frac{L_i + U_i}{2}. \tag{8}$$

$[L_i, U_i]$ describes uncertainty from insufficient criterion evidence. It is *not* a statistical confidence interval. Unknown outcomes trigger targeted evidence collection under the same sam-

pling rule for all models. If uncertainty remains, report the bounds, midpoint, and criterion-weight coverage. The earlier design used five-point task rubrics; the normalized expression above also supports other positive weight totals. The active ordinary-question protocol and event/metric backends are specified in Section 6; in particular, answers-mode dependent unknowns currently contribute zero rather than an automatically published midpoint. Targeted resampling and interval-based release checks require a frozen implementation before they can be claimed as executed.

The mapping from task-level evidence to each underlying scoring-item score must be frozen before evaluation. Additive dimension bounds are then computed by applying equation (4) to the corresponding lower and upper item bounds. Evidence coverage is reported alongside scores and does not change the fixed 3,300-point denominator.

The candidate publication rule for the world-model board compares the rankings obtained from aggregated midpoint and lower-bound scores. If their ordering differs, collect additional evidence or publish the ordering as provisional. Even matching orders do not establish statistical significance. Binary manipulation outcomes use their task success predicates; they do not automatically inherit video-rubric midpoint scoring.

2.9.6 Judge calibration and statistical reporting

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply judge calibration, human review, repeats, and statistical settings.

For judge-assisted evaluation, freeze the judge revision, prompt, decoding configuration, frame sampling rule, and retry policy. Blind the judge to model identity. Compare criterion judgments with independently annotated human labels on a stratified calibration subset, reporting agreement, disagreement categories, and adjudication procedures. Temporal disputes require evidence at the relevant timestamps. Judge errors are tracked separately from model errors.

For repeated trials, report sample sizes and a declared confidence-interval procedure. Where tasks share a source scene, trajectory, or intervention pair, resample at the independent group level. Paired comparisons use shared task groups and seeds when available. Report effect sizes and uncertainty before interpreting small score differences. Bootstrap settings, confidence level, and random seeds remain TBD until the statistical analysis is frozen.

2.10 Fairness and validity

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Document blinding, budget alignment, contamination checks, and reproducibility.

Separate benchmark construction, model submissions, and result adjudication, and blind judges to model identity. Restrict test answers and evaluation access under a versioned release policy. Within each comparison group, align hardware, information access, adaptation budgets, and stopping criteria. Record official implementations, default parameters, and declared deviations; loss convergence alone does not equalize training budgets. Preserve environment images, code and model revisions, configurations, and rollouts or API request records where permitted. Record known training exposure, audit contamination, and version task updates. Evaluator calibration and statistical reporting follow Section 2.9.

2.11 Distinctive evaluation design

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply evidence-based comparisons with relevant benchmarks.

Three design goals guide the framework: deployment relevance, diagnostic coverage, and verifiable findings. Deployment relevance connects understanding and learning to execution, cost, and operating constraints. Diagnostic coverage maps tasks to capabilities with explicit non-overlapping score attribution. Verifiable findings require independent checks, reproducible artifacts, and declared uncertainty. Comparisons with existing benchmarks must state the compared scope and supporting evidence before claiming priority or superior coverage.

2.12 Operation and maintenance

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Specify submission, updates, issue review, and maintenance for each board.

Model developers use failure diagnosis to guide iteration; application teams compare performance and resource requirements within relevant settings. Version task, data, environment, adapter, and metric releases together, record change impact, and recompute affected cached results. Publish reproducible materials and result explanations under the applicable licenses, retain protected test components under a declared policy, and provide submission, issue reporting, and review procedures. Community task contributions enter the benchmark after source review, validation, and versioned acceptance.

3 Embodied Reasoning Model Evaluation

Leaderboard lead(s): 武若楠

3.1 Scope (Evaluation subjects and system mapping)

Writing owner: 武若楠 **To complete:** Confirm the reasoning board's coverage and that the November schedule is consistent with it.

The reasoning board evaluates whether a model can produce correct answers, judgments, or plans from embodied observations. This round covers 28 level-4 evaluation points drawn from two of the five level-1 capability domains:

- **Basic intelligence** (24 points), spanning three level-2 dimensions:
  - *World understanding* (13 points) — instruction understanding (content comprehension, intent understanding, reference resolution, context understanding), object understanding (visual attributes, physical attributes, affordances, agent–object interaction), spatiotemporal understanding (spatial, structural, temporal), scientific understanding (physical laws), and causal understanding (logical causality).
  - *World prediction* (8 points) — dynamic consistency (prediction of force effects, material properties), scene consistency (subject consistency, semantic consistency), spatial geometric consistency (3D occupancy and reconstruction, multi-view consistency), and kinematic consistency (motion trajectory prediction). A reflection point (task assessment) is also included under this dimension.

  - *Autonomous decision-making* (3 points) — task planning (task decomposition, feasibility judgment) and task execution (error recovery).

- **Adaptive learning** (4 points), under the learning and evolution dimension: reflection (task assessment), autonomous exploration (information gathering, environment exploration and modeling), and memory (short-term memory).

The remaining three level-1 domains—human–machine coexistence, safety and reliability, and application practice—are part of the full capability system but are not scheduled for the reasoning board this round. Generalization and transfer (under adaptive learning) is also not included. These capabilities will be incorporated in future evaluation rounds as task design and data collection progress.

3.2 Benchmark Design (Task and data design)

Writing owner: 武若楠 **To complete:** Provide the task list, capability paths, data sources, and splits, with validated task counts.

Task types. The reasoning board covers the following task types: instruction and reference understanding, spatial reasoning, affordance judgment, physical and causal reasoning, temporal reasoning, and planning. Each task traces to a specific evaluation point in the full system: tasks are constructed from the level-3 capability definitions and level-5 observation notes, with an explicit link to the corresponding level-4 point.

Sample structure. A sample contains an image or video observation history, a question or instruction, and any state information the model is allowed to use. The output is one of: a multiple-choice or true/false selection, a numerical value, a point or region coordinate, or a short answer assessed by an LLM judge.

Runtime conditions. Observation duration, context length, tool access, answer attempts, and inference budget are declared per task. Text-only, visual, and state-assisted settings are labeled and reported separately.

Data. The evaluation set contains 11,029 questions drawn from 25 public benchmark subsets, covering 47 level-5 abilities. No question is counted toward more than one ability. Question counts per ability range from 22 to 847 (mean 234.7). Reference answers and private evidence are kept out of the model-visible input.

The data sources are organized by level-2 dimension:

- *Object understanding*: affordance and interaction from ASIMOV [50], PIO [57], RoboBench [35], and MV-RoboBench [42]; visual and physical attributes from RoboBench, EPIC-Kitchens [13], and PhysBench [11].
- *Spatiotemporal understanding*: spatial layout and relative distance from RoboBench, VSI-Bench [59], MV-RoboBench, EPIC-Kitchens, and OpenEQA [36]; structural topology from CausalPhys [53]; temporal reasoning from RoboBench and TVBench [12].
- *Instruction understanding*: intent and content understanding from RoboBench, ASIMOV, EAI [29], NegVQA [62], and MomaGraph [25]; reference resolution from PIO and RefSpatial [63].
- *Causal understanding*: causal chains, counterfactual, intervention, and attribution from CausalPhys and EgoTaskQA [23]; failure analysis from RoboBench.

- *Scientific understanding*: physical laws from PhyX [51].
- *World prediction*: state-change and semantic consistency from EgoTaskQA, ERIQ [41], SWITCH [31], RoboBench, MomaGraph, MV-RoboBench, EAI, and PIO; object permanence from CoreCognition [40] and IntPhys 2 [7]; kinematic consistency from CausalPhys and VLM4D [64]; dynamics from PhysBench and CausalPhys; spatial-geometric consistency from MV-RoboBench, OmniSpatial [24], and VSI-Bench.
- *Learning and evolution*: autonomous exploration from CrossPoint [56], ASIMOV, AQuA [43], and MM-UPD [37]; object attribute probing from ROMA [45]; short-term memory from EMBER-Bench [44].

ROMA requires the evaluated model to probe object attributes through multi-sensory interaction. To accommodate the audio-input limitations of the evaluated VLMs, the submitted protocol converts acoustic information to textual descriptions using Qwen3.5-Omni-Plus (`qwen/qwen3.5-omni-plus`), which receives only the audio stream and no other modality, and is prohibited from speculating about the sound source. This reduces leakage of visual or contextual cues so that the evaluated model reasons from the sound descriptions. Models are allowed to retry during interaction, so the task also measures the ability to recover from incorrect actions.

3.3 Evaluation Protocol (Input, output, and evaluation methods)

Writing owner: 武若楠 **To complete:** Provide the actual interfaces, execution protocols, scoring implementations, and judge settings.

Scoring. The primary metric is per-question accuracy. Approximately 95% of questions are scored automatically: multiple-choice and true/false items by direct answer comparison, numerical items (e.g., VSI-Bench distance and size estimation, 437 questions) on a 0–1 scale via relative error, and pointing or localization items by checking whether the response falls within the target region. One exception is low-visual-quality judgment (#46), where "clear" and "unclear" categories are each weighted at 50% to prevent trivially high scores from always predicting the majority class. If a model abstains, abstention rate and answer coverage are reported alongside accuracy.

The remaining approximately 5% (570 questions from three sources) use model-assisted scoring:

- RoboBench planning questions (#14, #28): scored by GPT-4o (`gpt-4o-2024-08-06`, accessed via OpenRouter), consistent with the benchmark's official protocol. These items were originally judged by Qwen3.8-Max but were re-scored with GPT-4o after 2026-10-20 due to the earlier judge's lenient grading.
- AQuA (#47) and OpenEQA (#6): scored by Qwen3.8-Max. Since Qwen3.8-Max is itself an evaluated model, self-evaluation bias may affect these items.
- EAI (#16, #31): scored by the benchmark's official rubric; #31 is evaluated by running the VirtualHome simulator directly. No LLM judge is used.

Reporting. Accuracy is reported by capability, task type, and generalization split, together with sample counts, latency, and any judge-assisted assessment.

3.4 Experimental Setup (Metrics and experimental settings)

Writing owner: 武若楠 **To complete:** Provide baseline versions, observation groups, question splits, repeated-response budgets, and scoring and uncertainty settings.

Models. We evaluate 21 models in two groups.The first group consists of leading general-purpose vision-language models (VLMs): Doubao-Seed-2-1-Pro [8], Qwen3.8-Max-0902, Qwen3.8-27B, and Qwen3.8-Flash [49], MiMo-v2.6-Flash and MiMo-v2.6-Pro [48], MiniMax-M3 [27], GLM-5.3-Flash [19], Gemma-4-26B-A4B [18], and DeepSeek-v4.1-Flash [15]. The second group consists of open-source models specifically trained for embodied reasoning: Pelican-72B [46], RynnBrain-122B and RynnBrain-9B [14], SpatialAxiom [34], CosmosR2-32B and CosmosR2-8B [39], HY-A3B [54], ZDTaichu [52], EmbodiedR-15 [60], MiMo-ER-7B [21], and UniFoLM-ER-1 [55]. This selection allows direct comparison between state-of-the-art general-purpose VLMs and models purpose-built for embodied scenarios.

Comparison groups. Each group fixes model versions, inputs, prompts, history, tool access, output format, and call budget. Zero-shot and adapted submissions are reported separately.

Main experiments. All 21 models are evaluated on the same set of 11,029 questions spanning 47 level-5 abilities under 28 level-4 evaluation points. Each ability's score is the mean per-question accuracy; the overall score is the equal-weight average of all 47 ability scores. Node scores at each level (level-1 through level-4) are equal-weight averages of the ability scores they contain, used for diagnostic comparison rather than hierarchical roll-up.

Statistics. All 47 ability scores carry equal weight; question counts per ability range from 22 to 847 (mean 234.7). Parse failures and timeouts are retained under the declared accounting rule. The 95% confidence interval half-width, estimated from per-ability binomial variance and combined across abilities, is approximately ± 1.1 points; models whose overall scores differ by less than 2 points should not be considered meaningfully separated.

Resources. Inference hardware, precision, call counts, token usage where available, and wall-clock time are reported. Resource differences are explicit comparison conditions.

3.5 Results and Discussion (Results and diagnostic analysis)

Writing owner: 武若楠 **To complete:** Populate the primary table, capability and format breakdowns, generalization results, confidence intervals, and grounded error cases.

Table 10. Embodied reasoning leaderboard — overall and level-2 dimension scores. **1st**[1], 2nd[2], *3rd*[3] in each column.

Model	Overall	World understanding	World prediction	Decision-making	Learning & evolution
Qwen3.8-Max-0902	**74.3**[1]	**73.4**[1]	**73.6**[1]	**77.6**[1]	**77.7**[1]
Doubao-Seed-2-1-Pro	71.4[2]	70.6[2]	71.4[2]	77.0[2]	69.0
Qwen3.8-Flash	*69.5*[3]	*69.5*[3]	*68.2*[3]	71.2	*71.2*[3]
Qwen3.8-27B	68.1	67.6	66.4	*72.1*[3]	71.8[2]
GLM-5.3-Flash	64.5	64.1	63.4	67.5	66.2
MiMo-v2.6-Pro	59.4	60.2	55.5	64.3	60.0
DeepSeek-v4.1-Flash	58.7	59.9	54.5	65.0	55.7
MiniMax-M3	58.4	57.5	56.6	67.9	56.9
MiMo-v2.6-Flash	56.8	57.4	52.7	64.5	55.9
Gemma-4-26B-A4B	52.7	53.7	44.9	65.2	55.3
Open-source embodied reasoning models					
SpatialAxiom	51.8	57.1	55.9	48.1	37.3
Pelican-72B	50.7	55.4	56.7	54.9	32.2
RynnBrain-122B	50.5	59.2	51.1	46.9	29.4
ZDTaichu	50.2	55.5	52.5	58.5	33.5
RynnBrain-9B	50.1	54.5	53.2	61.0	33.5
CosmosR2-32B	48.7	53.0	52.6	61.6	30.6
HY-A3B	48.2	59.5	53.3	51.6	16.6
MiMo-ER-7B	47.8	51.4	50.2	51.9	36.1
EmbodiedR-15	44.4	50.5	49.7	44.8	25.1
UniFoLM-ER-1	43.9	50.2	47.8	24.4	33.3
CosmosR2-8B	42.8	47.0	50.2	40.0	26.2

Models are ranked by the submitted overall scores (0–100). A visual separator marks the boundary between general-purpose VLMs and open-source embodied reasoning models. Aggregates remain provisional where ability scores are missing; see the missing-data discussion.

Main results. Qwen3.8-Max-0902 achieves the highest overall score (74.3), followed by Doubao-Seed-2-1-Pro (71.4) and Qwen3.8-Flash (69.5). In the current table, all general-purpose multimodal LLMs score higher overall than the embodied reasoning specialists. The highest-scoring specialist, SpatialAxiom (51.8), remains below even the lowest-scoring general-purpose model, Gemma-4-26B-A4B (52.7).

Domain-level analysis. Embodied reasoning specialists show weaker performance on adaptive learning. General-purpose VLMs average 63.8 on learning and evolution, while open-source embodied models average only 30.3—a gap of over 30 points. HY-A3B scores 57.1 on basic intelligence but only 16.6 on adaptive learning. Within the embodied group, memory scores are near or at zero for several models (HY-A3B: 0.0), and environment exploration scores are

frequently below 30. This suggests that embodied reasoning specialists, while competitive on perceptual understanding tasks, have limitations in multi-turn interaction and self-correction. A detailed level-3 breakdown is provided in Appendix Table 22.

Dimension-level patterns. Within basic intelligence, world understanding and world prediction scores tend to be close for a given model, while autonomous decision-making varies more widely. Task execution in particular shows high variance across models (range: 22.0–96.0), suggesting it is a strong differentiator.

Under world understanding, scientific understanding is an outlier: most general-purpose models score above 90, likely because the questions are well-covered by pretraining data. Spatiotemporal understanding is the weakest dimension for most models, with spatial understanding scores frequently below 55.

Missing data. Memory scores are missing for the general-purpose models; scientific understanding is also missing for HY-A3B. The current tables nevertheless contain aggregate scores for these models, which have not yet been reconciled with the complete-coverage rule in Section 2.1.3. The submitted numerical values are retained as provisional results. The owner must confirm coverage, aggregation denominators, and uncertainty estimates against the frozen per-question records before final comparison.

Error analysis. Errors are attributed to specific causes: missed visual evidence, reference-resolution mistakes, temporal or causal errors, unsupported assumptions, and invalid output formats. Where a plan is executed, plan quality and the linked manipulation outcome are reported as separate observations. Detailed error breakdowns are pending.

Effect of embodied training. Several open-source embodied models are fine-tuned from publicly available base VLMs. To isolate the effect of embodied-specific training, we compare each specialist against its corresponding base model on the same evaluation set. This comparison aims to answer two questions: (1) does embodied training improve performance on embodied reasoning tasks, and (2) does it degrade capabilities present in the base model, such as general instruction following or causal reasoning? A radar chart of selected base–fine-tuned pairs will visualize per-ability gains and losses. Specific model pairs and results will be reported once the experiments are complete.

3.6 Ablations and Case Studies (Ablations and case analysis)

Writing owner: 武若楠 **To complete:** Provide controlled observation, history, tool, and adaptation ablations, with correct and failed cases tied to source evidence.

Questions and scoring are held fixed while frame access, history length, tool permissions, or adaptation conditions are varied one factor at a time. Cases are included where extra context helps, fails to help, or introduces conflicting interpretations. Input evidence, answer, reference or rubric, and any downstream execution result are shown side by side. Case selection follows declared sampling or failure categories, with identifiers linking each case to the complete result set.

4 Simulated Manipulation Model Evaluation

Leaderboard lead(s): 刘璐 and 侯祥锐

4.1 Evaluation subjects and system mapping

Writing owner: 刘璐 and 侯祥锐 **To complete:** Confirm the projection from the full system to this board's scheduled points and interfaces.

This leaderboard evaluates closed-loop policies through the simulation entries in the shared capability system. The supplied November scope also includes adaptation, reliability, and deployment measurements. Interface qualification and actual task availability are confirmed separately.

4.2 Task and data design

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply tasks, complete capability paths, data sources, and splits.

Identify simulator suites, assets, tasks, initial distributions, held-out changes, and random seeds. Link success predicates and observation records to the full capability path. Adaptation and evaluation splits must remain separate.

4.3 Input, output, and evaluation methods

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply actual interfaces, execution protocols, scoring implementations, and judge settings.

Objective. Evaluate closed-loop manipulation under controlled environment and initialization distributions. Simulation supports repeated trials and systematic variations in objects, layouts, visual conditions, and task composition.

Interface. Fix the simulator build, assets, robot embodiment, observation modalities, action space, control rate, physics settings, and episode horizon. Policies use the same permitted information within a comparison group. Methods with privileged simulator state are labeled and reported separately from methods using only deployable observations.

Evaluation. Use a frozen success predicate, termination policy, seed list, and reset distribution. Share evaluation seeds across policies where possible. Separate development environments from held-out test environments. Generalization splits should identify the changed factor and prevent overlap through shared scene assets, source trajectories, or task templates when these would reveal the test condition. Training and adaptation budgets accompany every submission.

Required reporting. Provide per-suite success, performance on unseen conditions, run-to-run variation, and compute requirements. **To finalize:** adopted simulator suites and versions, environment licenses, held-out splits, success predicates, and evaluation episode counts. Simulation results support claims about the specified simulated setting; transfer to physical execution requires matched real-world evidence.

4.4 Metrics and experimental settings

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply eligible policies and executors, training/adaptation budgets, matched seeds, condition shifts, and success statistics.

Comparison groups. Instantiate Sections 2.1.4, 2.5.2, 2.7.2, and 2.8.2. Freeze embodiment, observation privilege, action interface, controller, horizon, success predicate, and evalua-

tion seeds. Separate zero-shot deployment, fixed-budget adaptation, and any further training. Record demonstration volume, optimization budget, and policy-stack components.

Main experiments. Compare task success on declared in-distribution and held-out conditions using matched initial states or seeds where possible. Measure object-shape, background, new-task, and goal-change conditions in the scheduled scope. Adaptation studies report performance against a declared data or compute budget. The VLM-to-executor simulation study in Section 8.2 uses the same task and outcome specifications.

Statistics and resources. Freeze repeat budgets, per-task and aggregate weights, and the treatment of reset or infrastructure failures. Report episode counts, capability coverage, uncertainty, execution time, inference latency, and scheduled training, adaptation, memory, and compute measurements.

4.5 Results and diagnostic analysis

Writing owner: 刘璐 and 侯祥锐 **To complete:** Populate the simulation table, per-task and held-out results, adaptation curves, uncertainty, and failure breakdowns.

Use the frozen release records in Section 2.8 and accounting rules in Section 2.9. Present observation and adaptation groups separately; actual configurations, counts, and measurements remain pending.

Table 11. Simulated manipulation leaderboard.

Model	Success rate (%) ↑	ID success (%)	OOD success (%)	Evaluated episodes	Coverage (%)
Model A	...	...	...	...	...
Model B	...	...	...	...	...
Model C	...	...	...	...	...

ID and OOD denote the declared in-distribution and held-out condition splits. Results are pending.

Primary and generalization analysis. Show per-task success and the declared aggregate alongside object, background, task, and goal-change conditions. Analyze adaptation curves at matched budgets and sensitivity to initialization. Check whether gains persist across tasks rather than following episode-count imbalance.

Failure diagnosis. Separate perception or grounding errors, subgoal ordering, grasp or contact control, recovery failures, and environment faults. For goal changes, inspect the post-change response and eventual outcome. Simulation-to-real interpretation requires the paired physical evidence in Chapter 7.

4.6 Ablations and case analysis

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply controlled observation, action, history, adaptation and executor ablations, with synchronized simulation evidence.

Change one observation, control, history, or adaptation factor while preserving test seeds and success criteria. For modular systems, isolate the planner, skill executor, and recovery loop where their interfaces allow a controlled comparison. Show matched initial states, actions, state transitions, subgoals, and success-function outputs for representative successes and failures.

5 Real-World Manipulation Model Evaluation

Leaderboard lead(s): 郭美衫

5.1 Evaluation subjects and system mapping

Writing owner: 郭美衫 **To complete:** Confirm the projection from the full system to this board's scheduled points and interfaces.

This leaderboard evaluates physical execution against the shared capability system, including scheduled safety, adaptation, reliability, and deployment measurements. Every result identifies the tested robot, policy stack, information access, and actual operating constraints.

5.2 Task and data design

Writing owner: 郭美衫 **To complete:** Supply tasks, complete capability paths, data sources, and splits.

Supply task inventories, robot and gripper configurations, camera and calibration records, initial-state tolerances, resets, and demonstrations. Preserve the full capability path, physical outcome evidence, interventions, and invalidated trials.

5.3 Input, output, and evaluation methods

Writing owner: 郭美衫 **To complete:** Supply actual interfaces, execution protocols, scoring implementations, and judge settings.

Objective. Evaluate task execution through physical robot interaction. Candidate task families include pick-and-place, articulated-object operation, tool use, precision placement, and multi-step manipulation. The released suite must identify which families and embodiments are actually available.

Interface. A policy receives the permitted cameras, task instruction, and declared proprioception, and outputs actions through an embodiment-specific adapter. The protocol fixes action units, coordinate frames, bounds, command frequency, action horizon, observation preprocessing, and episode termination. Robot model, gripper, camera placement, calibration procedure, controller, and workspace configuration are part of the comparison setting.

Evaluation. Primary evidence is the executed trajectory and verified task outcome. Define initial-state tolerances, reset procedures, trial ordering, repeat counts, maximum duration, and permitted recovery before evaluation. Record policy-triggered stops and human interventions explicitly; an intervention does not count as autonomous success. Hardware or infrastructure faults require a documented invalidation rule and remain visible in the run accounting.

Required reporting. Report task success, constraint violations, intervention frequency, completion time, trial counts, and uncertainty within each hardware setting. Separate zero-shot

deployment from task-specific adaptation. **To finalize:** robots, sensors, task inventory, trial budget, outcome verification, and physical operating constraints.

5.4 Metrics and experimental settings

Writing owner: 郭美衫 **To complete:** Supply hardware comparison groups, deployment/adaptation budgets, trial ordering, repeats, outcome inspection, and resource measurements.

Comparison groups. Instantiate Sections 2.1.5, 2.5.3, 2.7.3, and 2.8.3. Freeze the robot, gripper, sensors, controller, workspace, information permissions, and adaptation regime. Declare which differences in policy or executor interfaces require separate groups. The owner provides the actual model roster and the deployment and adaptation records.

Main experiments. Evaluate autonomous completion on matched task cards, initial-state tolerances, and repeat budgets. Record trial order and any counterbalancing procedure. Include scheduled object, background, new-task, and goal-change conditions, together with physical and interaction constraints. The VLM-driven physical experiment in Section 8.3 follows these same task and outcome rules. Matched simulation comparisons use Chapter 7.

Statistics and resources. Measure autonomous success, violations, interventions, recovery, execution and reset time, and scheduled deployment costs. Freeze timing endpoints and the handling of faulted or interrupted trials. Report repeated-trial uncertainty within each hardware group and retain all trial statuses in the accounting.

5.5 Results and diagnostic analysis

Writing owner: 郭美衫 **To complete:** Populate the hardware-specific table, task and condition results, intervention and constraint records, uncertainty, and deployment measurements.

Use the release records in Section 2.8 and the accounting rules in Section 2.9. Keep hardware and adaptation groups separate; actual robot configurations, trial counts, and measurements remain pending.

Table 12. Real-world manipulation leaderboard.

Model	Success rate (%) ↑	Evaluated trials	Intervention rate ↓	Violation rate ↓	Completion time ↓
Model A	...	...	...	...	...
Model B	...	...	...	...	...
Model C	...	...	...	...	...

Report each hardware setting separately. Model labels and measurements are pending.

Primary and generalization analysis. Present autonomous success by task and condition, with intervention, constraint, time, and deployment measurements alongside it. Examine repeat variation and reset differences. Report failed, assisted, invalidated, and unresolved trials explicitly rather than hiding them behind a final-state success label.

Failure diagnosis. Distinguish visual grounding, planning, controller or contact errors, communication delay, reset deviations, and hardware faults. For matched simulation pairs, identify where the execution diverges and which physical differences could explain the observed gap.

5.6 Ablations and case analysis

Writing owner: 郭美衫 **To complete:** Supply feasible controlled component and observation ablations, repeated trials, and traceable initial/execution/final-state evidence.

Keep hardware and task conditions fixed when varying observation access, high-level planning, executor settings, or permitted recovery. Record component changes and any remaining unmatched factors. Show synchronized initial, execution, and terminal observations with action, intervention, and outcome records; connect selected cases to their simulation counterparts when available.

6 World Model Evaluation

Leaderboard lead(s): 史可为

6.1 Scope and evaluation interfaces

Writing owner: 史可为 **To complete:** Freeze the task-to-item crosswalk, eligible control groups, and model revisions before publishing the November leaderboard.

The world-model board evaluates the evolution of a world under declared observations, instructions, and controls. Its evidence is a predicted video or state sequence: a plausible image alone does not establish that the requested intervention occurred or that its consequences are correct. Video-quality benchmarks and world-model benchmarks provide useful complementary views [22, 28]; REAL additionally binds each finding to a capability path, an intervention, and an observable outcome.

Framework coverage. The PPT world-model projection spans five domains, 13 dimensions, 52 level-3 abilities, and 153 indicators (table 5). This describes framework scope; task implementation and completed measurements require separate verification.

Legacy schedule. The legacy November projection contains 33 level-4 scoring items across three level-1 domains. Basic intelligence contributes 8 world-understanding items and 10 world-prediction items; adaptive learning contributes 11 generalization-and-transfer items; safety and reliability contributes 4 safety items. Their ceilings are 800, 1,000, 1,100, and 400, respectively. These subtotals preserve legacy task groups; their item-level mapping to the new indicators remains to be checked. The fixed total is 3,300, using addition rather than the reasoning board's equal-weight ability average.

Interface groups. We distinguish robot-action-conditioned prediction, camera or interactive-control-conditioned prediction, and language-conditioned video generation. A task can expose an initial image, permitted history, scene constraints, timed instructions, and an eligible control signal. Action arrays, native camera trajectories, discrete keyboard commands, and textual descriptions are separate information conditions. A textual approximation is recorded as such; it does not establish native action or camera-control support. Multi-view tasks additionally require the declared views and synchronization. Interface eligibility is determined before comparison, with unsupported items handled under Section 2.9.4.

6.2 Benchmark construction and data

Writing owner: 史可为 **To complete:** Confirm source-grouped splits, asset permissions, input eligibility, and the frozen release manifest; retain the implementation inventory separately.

Task unit. A case specifies an initial observation, public conditioning, a temporal plan, expected output streams, and a private evaluation definition. A branch is a generation run within a case; a view is an output stream within a multi-view case. Neither branches, individual judge questions, nor judge requests are counted as additional cases. Generation inputs are explicitly separated from reference answers, reference events, RGB-D evaluation assets, masks, and scoring prompts. Public action controls remain available only when the declared input mode permits them.

Composition and provenance. The active catalog contains 497 basic-intelligence cases, 350 adaptive-learning cases, and 300 safety cases. Sources include locally authored intervention and manipulation scenarios, imported reference-event cases and their variants, WBench natural-scene depth probes, and the supplied kinematics, TriWorldBench, and ORBIA subsets. Source images, identifiers, edits, reference assets, and upstream code revisions are retained. Generated or edited initial scenes are distinguished from unmodified source imagery; source provenance does not establish training-data independence.

Table 13. Active world-model evaluation routes. Counts describe the implementation snapshot, not completed model evaluations.

Route	Cases	Measurement and scope
Question answering	575	275 cases have question protocols; 300 safety cases lack a scoring profile.
Event matching	400	Reviewed event annotations and joint F1 at temporal IoU ≥ 0.5.
Camera control	12	Video-estimated camera trajectories against declared controls.
Scene consistency	60	Subject appearance, background appearance, and hard-cut continuity.
Depth consistency	20	DA3 cross-frame depth reprojection; internal geometry, without depth GT.
Kinematics	40	Trajectory overlap and rules; velocity/acceleration remain native errors.
Three-view prediction	20	TriWorldBench head/left/right predictions; seven original metrics.
Cross-view geometry	20	ORBIA T5 covisible-reprojection feature branch, not its full overall score.
Total	1,147	Each discovered case has one route; multiple views and event questions are not additional cases.

Configuration availability is not generation coverage or metric validation. Raw-error and dataset-specific metrics are not silently converted to the fixed November score.

The 400 event cases are routed by their reference-event annotation, not merely by the presence of a reference video. The 20 TriWorldBench cases retain the head, left, and right views; the removed six-view cases are excluded. The 20 ORBIA cases use their declared camera or alternative input mode and corresponding RGB-D references. Reference assets are not silently

supplied to a single-image generator, and one image is not duplicated to impersonate independent views.

Quality and splits. The catalog and its 32 groups are implementation units, not a completed test split. Before freezing a release, group related cases by source scene, reference episode, intervention pair, and base/variant family; keep each group within one split. Validate observable interventions, non-target constraints, time windows, and the distinction between necessary dependencies and independent violations. Changes to a rubric invalidate affected scores; changes to public conditioning also invalidate affected generations. The checked-in snapshot stores case identifiers and metadata/judge hashes, allowing these counts to be reproduced without treating a changing working tree as a published benchmark.

6.3 Evaluation protocol and metrics

Writing owner: 史可为 **To complete:** Freeze judge revisions, frame sampling, calibration, raw-to-item mappings, and unresolved-evidence handling for the release.

Execution and validity. Preserve the original initial image, timeline, and conditioning fields in the generation record. Record resizing or cropping, coordinate conventions, control translation, temporal quantization, frame rate, and actual output duration. Stateful generation and last-frame continuation are different experimental conditions. Do not create missing duration by replay, repeated frames, or time rescaling. Decode all required streams and verify their generation identity before scoring. Camera tasks may contain valid stationary intervals; physical-response and revisit tasks require their own observable intervention or departure checks.

Ordinary question answering. The 275 configured cases use authored yes/no questions and deterministic local scoring. Positive and negative polarity are resolved by code. For an ordinary ungated case, the score is given by Equation 9, where K_i counts scored questions and z_{ik} is the resolved criterion status. In the current answers protocol, a behavioral failure or dependent `unknown` contributes zero; auxiliary gates have no extra scoring weight. Task-specific no-action gates and contact/support branches retain their own definitions rather than being replaced with an unconditional average. Media faults, invalid inputs, and absent judge responses are technical states, not behavioral zeroes.

The local default judge identifier is `qwen/kimi-k3`; this records configuration, not a new verification of provider availability or model capability. The default answers mode returns ordered yes/no answers; evidence mode additionally retains structured judgments and frame evidence. The current ordinary-question sampler uses 1 FPS with a maximum of 32 frames. At that cap it can omit the later part of a long video; the earlier proposal for targeted resampling is not evidence that a second pass is already performed. A release must audit temporal coverage and freeze a consistent remedy before ranking.

$$s_i^{\mathrm{QA}} = \frac{1}{K_i}\sum_{k=1}^{K_i}\mathbb{I}[z_{ik} = \mathrm{pass}], \qquad K_i > 0, \tag{9}$$

$$s_i^{\mathrm{event}} = \frac{2M_i}{N_i^{\mathrm{ref}} + N_i^{\mathrm{pred}}}, \qquad \mathrm{tIoU} \geq 0.5, \tag{10}$$

$$s_i^{\mathrm{appearance}} = A_i \min(1, D_i/0.1213), \tag{11}$$

$$s_i^{\mathrm{depth}} = \frac{1}{1+e_i}. \tag{12}$$

Reference-event evaluation. Generated video is annotated, independently reviewed, and, when requested by the annotation workflow, checked with SAM before revision and validation. The annotation stage receives neither reference-event answers nor the generation prompt. Reference and predicted events are cropped to the declared observed interval in original seconds. Candidate pairs must satisfy temporal IoU ≥ 0.5 and semantic agreement. A one-to-one assignment first maximizes the number of accepted matches M_i, then their total temporal IoU; Equation 10 reports joint F1. If both event sets are empty, the score is undefined, not a perfect match. Preserve micro/macro summaries, match evidence, and explicit generalization-eligibility flags; do not infer eligibility from a base/variant label.

Appearance and continuity. Subject appearance uses DINO features; background appearance uses the WBench CLIP implementation. Both compare the generated first frame and adjacent frames and apply the local dynamic factor in Equation 11, where A_i is appearance similarity and D_i the RAFT-derived dynamic degree. This is a local composition, not the unmodified WorldArena or WBench main score. The threshold 0.1213 awaits calibration for this composition. Under the inherited dynamic mapping, zero optical flow still gives a small positive factor, so this is not a strict frozen-video gate. Full-frame appearance does not isolate object identity or recover three-dimensional background truth. The semantic-consistency CV subset uses TransNetV2 hard-cut detection: no cut scores one, any retained cut scores zero. Its score measures segment continuity, not semantic or causal correctness.

Camera and geometry. For camera tasks, VIPE estimates a common-coordinate trajectory from the generated video. The declared controls define the comparison trajectory; they cannot replace the measured trajectory. The normalized score is $1 - (\mathrm{nATE}_t + \mathrm{nATE}_r)/2$, with the native 0–100 report retained. It is trajectory conformity rather than calibrated metric localization. Depth evaluation jointly estimates depth and cameras with DA3 at approximately 3 FPS, reprojects points from other frames into a target frame, and averages per-frame median relative depth residuals into e_i; Equation 12 is the main score. Report valid-frame coverage and auxiliary RGB reprojection PSNR. Estimated internal consistency is not metric depth accuracy, and a frozen scene can still score well.

Native-unit and multi-view measurements. The supplied trajectory subset reports $100 \times \mathrm{IoU} \times$ rule correctness. Its velocity/acceleration subset retains two-dimensional reference-normalized errors in reference-diagonal units per second and per second squared, without inventing a percentage mapping; a single image does not uniquely determine initial velocity. TriWorldBench retains seven original metrics, including nPSNR, SSIM, appearance, and VLM measures, and labels the exact episode/metric range of any dataset aggregate. ORBIA reports the T5 covisible-reprojection feature branch, not the full T5 or overall score, preserving its native frame count and 16 FPS. These measurements require their own compatible input protocols and do not automatically supply November item scores.

6.4 Models and experimental settings

Writing owner: 史可为 **To complete:** Freeze participating configurations, native interface support, generation budgets, seeds, and the final task-to-item aggregation.

Model roster. Table 14 lists the 24 models in the project model list, using its AC-WM, WAM, and Video categories. Model names follow that list without adding inference configurations or experimental variants. Each experiment records the selected checkpoint and adapter revisions, input mode, and comparison conditions.

Table 14. World models and video-generation baselines from the project model list.

Category	Model names
AC-WM	XGEN-JING; Zing-0.5; EchoWM; EchoFlash（Echo-WM Flash Preview）; AlayaWorld; YUME; Happy Oyster; LingBot-World v2（fast）; HelixWorld 1.0（Preview v1）; SANA-WM
WAM	Cosmos3-Super; Cosmos3-Nano; WoW-14B-2M（WoW-1-Wan-14B-2M）; GigaWorld-0; Video Prediction Policy 2（VPP2）
Video	HY-video 1.5（HunyuanVideo-1.5）; Kling 3.0（可灵 3.0）; Wan3（Wan 3.0）; Seedance 2.5; MiniMax-H3; Wan2.2 I2V A14B; LingBot-Video 30B-A3B; Happy Horse 1.1（HappyHorse 1.1）; Wan2.2 TI2V 5B

Names and categories follow the project model list. Input-interface eligibility and checkpoint versions are specified separately for each experiment.

Matched generation conditions. Compare models only within compatible observation and control groups. Preserve task duration and information access while recording each model's native resolution, FPS, sampling steps, precision, and continuation strategy. Do not hide differences by presenting fixed-camera text generation as native trajectory following. For a multi-view task, disclose whether prediction is joint or independent across views. Repeated runs use recorded seeds where supported; the number of repeats and budget must be fixed before comparison, not selected after inspecting outcomes.

Aggregation contract. The implementation first averages available scalar case scores within a capability group and reports a coverage-dependent partial mean. Some backends report native measurements with `score=null`; completed measurement is therefore not the same as a scalar score. Neither the partial mean nor its completed-group variant is the fixed November total. To publish that total, freeze the case/criterion crosswalk and compatible normalization for every scheduled item, obtain $q_{mj} \in [0, 100]$, and apply Section 2.9.4 without dropping missing items or changing the 3,300-point denominator. Pending measurements stay unresolved; declared non-participation follows the source table's zero-score convention. Raw errors and dataset-specific aggregates remain separate unless a justified conversion is frozen.

Statistics and resources. Report eligible, generated, valid, measured, and scalar-scored counts separately. Preserve case, video, judge, code, and metric identities when resuming or aggregating results; file modification time alone is insufficient. Estimate uncertainty by source-group resampling, keeping variants and repeated seeds together. Confidence levels, resampling repetitions, and judge calibration remain release parameters until frozen. Report wall-clock generation/evaluation time, hardware, precision, peak memory where available, token or API use, and retries. No confidence interval or cost number is inferred from registry membership.

6.5 Results and diagnostic analysis

Writing owner: 史可为 **To complete:** Complete the frozen-scope generation and scoring runs, resolve missing profiles and item mappings, and then populate model scores and uncertainty.

Audited implementation results. A fresh catalog scan confirms 1,147 cases, 32 groups, 275 configured ordinary-question cases, 400 event cases, and 172 dedicated-metric cases. Another 300 cases, all in safety, have no complete dimension profile. The remaining 847 cases are not blocked by that particular configuration issue; this does not demonstrate available predictions, dependencies, or successful metric execution.

Table 15. World-model leaderboard: second-level capability scores.

Model	Total / 3,300	SECOND-LEVEL DIMENSIONS			
		World understanding / 800	World prediction / 1,000	Generalization and transfer / 1,100	Safety / 400
Frozen release	...	...	...	...	...

A complete current-scope model result has not been established by the inspected artifacts. A light ellipsis means pending measurement, not zero. Model names are listed in Table 14; partial or native-unit measurements are not a 3,300-point ranking. Declared non-participation contributes zero under the fixed November rule.

Model-result availability. The inspected local artifacts do not establish a complete, current 33-item leaderboard. The existing unified summary belongs to an earlier task scope and cannot be relabeled as results for the current catalog. Reference-copy, controlled-fixture, and feature-comparison checks in the supplied kinematic and multi-view reports validate parts of the software or metric. They are not generated-model scores. The main table consequently retains pending results rather than constructing rankings from incompatible partial artifacts.

Diagnostic findings and interpretation. Three boundaries matter for subsequent experiments. First, instruction realization must be separated from response correctness: failure to release, load, contact, or revisit does not establish a physical-law outcome. Independent non-target violations must remain observable even when a desired action fails. Second, appearance and depth proxies can reward static or geometrically self-consistent but incorrect videos; report motion/control evidence and coverage alongside their native scores. Third, missing protocols, stale generations, uncertain annotations, and evaluator failures have different causes and should not be merged into a model-failure count.

Generalization and extended coverage. Compare matched source groups and preserve explicit base/variant links and eligibility flags. Current variant coverage is uneven and is not evidence for every one of the 11 scheduled transfer items. Likewise, the older WMBench360 plan's long-term object persistence, frozen V-JEPA reference similarity, and two-alternative realism judgments are prospective diagnostic routes, not completed experiments or additions to the November denominator. The current catalog and metric definitions take precedence over historical task counts and proposed scorer behavior.

6.6 Ablations and evidence-based cases

Writing owner: 史可为 **To complete:** Run the prespecified controlled comparisons, attach timestamped success/failure evidence, and report unresolved cases with their actual coverage.

Controlled comparisons. Hold source cases, seeds, and judge settings fixed while changing one factor at a time: native controls versus their declared text approximation; maintained history versus last-frame continuation; prediction horizon within supported interfaces; ordinary-question sampling density and temporal coverage; or appearance-only versus the declared dynamic composition. Frozen-video and known geometric-perturbation controls test evaluator sensitivity separately from model performance. Calibration or ablation results are published only after those experiments run; no gains are assumed here.

Case reporting. Each case shows the public initial observation and control timeline, actual generation record, predicted sequence, permitted reference evidence, criterion or event outcome, and timestamps. Include successful, failed, and unresolved cases, with the sampling rule declared in advance. Contact and material cases identify the intervention before judging the response; sequence and causal cases compare event order under the original timeline; camera/depth cases distinguish motion realization from geometric self-consistency. Multi-view examples show all required streams and their alignment.

Reproducibility boundary. The snapshot, model roster, and source hashes are included in the manuscript package. They support reconstruction of the inventory and the claims about implementation state. They do not substitute for the frozen release manifest, original videos, judge records, calibration labels, or completed model experiments needed for a publishable performance comparison.

7 Simulation-to-Real Alignment Evaluation

This chapter tests which findings transfer between matched simulated and physical settings. Alignment is an experimental hypothesis supported by paired evidence; visual resemblance alone does not establish equivalent dynamics or difficulty.

7.1 Matched tasks and comparison units

Writing owner: 刘璐 and 侯祥锐; 郭美衫 **To complete:** Jointly provide matched sim/real tasks, initial states, embodiments, and exclusions.

Assign a shared concept identifier to matched goals, objects, instructions, and initial-state tolerances. Record embodiment, observations, action semantics, control rate, contact parameters, and termination differences. Evaluate comparable policy versions and adaptation regimes; label unavoidable mismatches. Report matched sample size and unmatched exclusions before comparing scores.

7.2 Alignment measurements and uncertainty

Writing owner: 刘璐 and 侯祥锐; 郭美衫 **To complete:** Jointly define paired metrics, coordinates, timing, budgets, and uncertainty.

For each matched setting, compare task success, constraint violations, completion time, and failure categories. Report the paired simulation-to-real success difference with a declared uncertainty procedure. Geometric or trajectory deviations require a common coordinate system, time alignment, and reference measurements. Inspect whether model ordering is preserved only across models eligible for the same matched set. Any rank agreement statistic and its unit of resampling must be frozen before results are examined.

7.3 Cross-leaderboard analysis and ablations

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide each board's matched evidence, comparison factors, and ablation results.

Reasoning-to-action analysis links answers or plans to actual execution; prediction- to-action analysis links world-model intervention responses to downstream outcomes. Shared concepts support these studies, but do not guarantee identical information or difficulty. Report matched counts and uncertainty, and interpret correlations within the declared setting.

Planned ablations isolate observation access, model adaptation, simulator parameters, deterministic versus judge-assisted checks, and macro versus micro aggregation. World-model temporal sampling and prediction-horizon ablations belong to the corresponding fixed control group. Change one declared factor per comparison and preserve failure accounting.

7.4 Experimental report and failure analysis

Writing owner: 刘璐 and 侯祥锐; 郭美衫 **To complete:** Jointly populate paired results and diagnose transfer and execution failures.

Table 16. Matched simulation-to-real evaluation template.

Policy / setting	Sim SR (%)	Real SR (%)	Real–sim gap (pp)	Matched tasks	Coverage (%)
	...	...	...	...	...
Model A					
	...	...	...	...	...
Model B					

Use the same matched task set and a declared aggregation rule for both columns. Report trial counts and uncertainty with the full results; all cells are pending.

To complete: paired task manifest, physical and simulator configurations, policy versions, repeat budget, uncertainty procedure, and verified artifacts. Analyze perception, contact and dynamics, control timing, resets, and evaluator errors separately before drawing transfer conclusions.

8 GPT6 as a VLM: Simulation and Real-World Experiments

This study evaluates GPT6 primarily through a vision–language interface: interpreting observations, grounding instructions, proposing task-level actions, and updating decisions from feedback. A fixed adapter and executor translate those outputs into simulated or physical actions. GPT6 remains the study label; the exact model revision, supported inputs, and access configuration must be confirmed. The following specifies the experiments to conduct, without assuming a native motor-control interface or reporting unmeasured capability.

8.1 VLM role and execution interface

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫 **To complete:** Supply the VLM input and response schemas, prompts, skill vocabulary, adapter, executor, and feedback protocol.

Role and inputs. Supply the task instruction, current camera views or permitted video history, progress record, and declared state information. Evaluate object and relation grounding, task decomposition, subgoal selection, progress assessment, and recovery decisions. Any object-pose estimator or additional perception module is named as part of the system.

Outputs and execution. Freeze a structured response identifying the target object, selected skill, parameters in a declared coordinate frame, and expected outcome. The adapter checks format and allowed parameters, then dispatches to a fixed skill library, motion planner, or controller. Define malformed-response handling and permitted repair attempts. The VLM contribution and the execution components are reported separately.

Feedback loop. Compare an initial plan with execution-time visual reassessment. Freeze observation refresh, retained history, decision frequency, call budget, latency handling, and stopping conditions. These conditions test visual decision-making and closed-loop task management; physical action performance is attributed to the specified VLM-plus-executor system.

8.2 Simulated manipulation experiments

Writing owner: 刘璐 and 侯祥锐 **To complete:** Supply simulator tasks, fixed executors, VLM prompts and calls, comparison conditions, seeds, and simulation results.

Tasks and environment. Use the frozen simulation tasks and settings in Sections 2.5.2, 2.7.2, 2.8.2, and 4.4. Select eligible basic, precise, and multi-step tasks and the scheduled condition shifts. The skill library, controller, perception backend, and observation permissions remain fixed when comparing VLM decision conditions. Privileged-state-assisted conditions receive their own label.

Execution protocol. Present the allowed visual observation and instruction, obtain a structured skill or subgoal, execute it, and return the new observation and declared progress feedback. Fix the skill vocabulary, parameter schema, maximum decisions, retries, and episode horizon. State whether simulation pauses during remote inference or advances with elapsed time; these yield different latency conditions.

Main comparisons. Compare one-shot VLM planning and visual feedback-driven replanning on matched tasks and seeds. Add another eligible VLM with the same executor, and an eligible native policy baseline as a separately documented route. Use object, background, new-task, and goal-change conditions from the frozen scope. Actual baseline versions and budgets remain to be supplied.

Measurements and diagnosis. Report task success, subgoal completion, grounding or parameter errors, valid structured outputs, recovery, calls, and latency. Inspect whether a failure originated in visual understanding, subgoal selection, response parsing, or low-level execution. Populate the following simulation-specific table from verified rollouts.

Table 17. GPT6 VLM study: simulated manipulation comparisons.

Condition	Task SR (%)	Subgoal completion	Valid responses	Model calls	Decision latency
VLM, one-shot	...	...	...	...	...
VLM, visual feedback	...	...	...	...	...
Native policy baseline	...	...	...	...	...

Planned conditions; versions, executors, denominators, and measured values remain to be supplied. Decision-specific fields apply only to compatible interfaces.

8.3 Real-world manipulation experiments

Writing owner: 郭美衫 **To complete:** Supply robot deployment, fixed skills and controllers, VLM interaction records, repeated trials, and physical outcomes.

Tasks and deployment. Use Sections 2.5.3, 2.7.3, 2.8.3, and 5.4 to select available physical tasks, hardware, calibration, and trial protocols. Reuse the structured VLM interface where feasible, while recording any embodiment-specific skill or parameter transformation. Fix the perception, skill-execution, and controller components for VLM comparisons.

Execution protocol. Initialize the physical task, provide current camera observations and the instruction, execute the selected skill, and return post-action observations for progress assessment or replanning. Declare remote inference delay, command queuing, maximum calls, retry permissions, timeouts, and the handling of stale observations. Stopping, intervention, reset, and outcome verification follow the physical leaderboard protocol.

Main comparisons. Compare one-shot and feedback-driven VLM execution under matched task cards, initial-state tolerances, and repeat budgets. Use a fixed-executor VLM baseline where available and document any native-policy comparison in its eligible group. For matched simulated tasks, compare failures and transfer gaps using Chapter 7; disclose remaining object, sensor, controller, and dynamics differences.

Measurements and diagnosis. Report autonomous task success, intervention, constraint violations, completion time, calls, and latency with trial counts and uncertainty. Separate correct VLM decisions followed by executor failure from visually ungrounded or inappropriate decisions. Populate the following hardware-specific table using synchronized trial evidence.

Table 18. GPT6 VLM study: real-world manipulation comparisons.

Condition	Autonomous SR (%)	Intervention rate	Violation rate	Completion time	Model calls
VLM, one-shot	...	...	...	...	...
VLM, visual feedback	...	...	...	...	...
Native policy baseline	...	...	...	...	...

Report each hardware and adaptation group separately, with trial counts and uncertainty. Rows define comparisons rather than measured capability.

8.4 VLM ablations and auxiliary-route comparisons

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply matched VLM ablations, auxiliary reasoning/coding conditions, resource controls, and component-attribution evidence.

Primary ablations. In simulation and physical execution separately, vary visual feedback, history or memory, grounding information, subgoal decomposition, and recovery permissions while holding tasks and the executor fixed. Evaluate the change in both decision quality and task outcomes. Report information and call-budget differences rather than combining unmatched conditions into one ranking.

Auxiliary reasoning and coding. Reasoning is the VLM's task-level decision process in the primary conditions. A code-assisted route is an optional comparison: freeze its tools, execution environment, development/test separation, and repair budget, and score the generated artifact and downstream execution separately. It does not replace the primary visual decision experiment.

Table 19. GPT6 VLM ablations and an optional code-assisted comparison.

Usage mode	Task outcome	Valid outputs (%)	Latency ↓	Measured cost
VLM, one-shot	...	...	...	...
VLM, visual feedback	...	...	...	...
Code-assisted (optional)	...	...	...	...

Populate simulation and physical results separately, with matched tasks, executors, and declared call budgets. Model versions, units, and measurements remain pending.

Interpretation. Compare model revisions, prompts, calls, latency, resource cost, and executor differences together with outcomes. Simulation and physical results remain separate, with matched pairs supporting the alignment analysis. Attribute claims to the measured VLM-plus-executor configuration and isolate the model contribution through controlled component comparisons.

9 Summary and Outlook

9.1 Summary

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Summarize verified findings and practical implications for each leaderboard.

REAL organizes embodied-intelligence evaluation into four leaderboards for reasoning, simulated manipulation, physical manipulation, and world modeling. The proposed framework connects them through shared capability definitions, versioned protocols, and auditable evidence while preserving the meaning of each setting's outcomes. This working report provides the methodological foundation and reporting structure for the technical report. The next milestone is a frozen benchmark release with validated configurations and measured results.

Overall and capability-specific reports remain within each leaderboard, while simulation-to-real alignment and model-role studies provide controlled cross-setting interpretation. Release milestones are a frozen taxonomy, tasks and settings, calibrated evaluation, and validated measurements.

9.2 Limitations and Responsible Interpretation

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply coverage limits, comparability constraints, measurement errors, and exposure.

Coverage. A taxonomy can be broader than the evaluated task set. The release must distinguish planned capability coverage from validated tasks and completed runs. The current world-model materials are a foundation for that leaderboard; they do not establish the readiness of all four leaderboards.

Comparability. Robot hardware, sensor placement, controller design, privileged simulator state, and model adaptation can change measured performance. Scores are interpreted within the declared comparison group. Matched tasks support cross-setting diagnosis only to the extent justified by their alignment.

Measurement. Visual judges may miss brief events or infer unsupported physical properties. Physical quantities unavailable from the observation should remain unknown unless supplied by an independent measurement. Evidence bounds and repeated-trial confidence intervals answer different questions and are reported separately.

Data provenance and generalization. Public tasks and assets may overlap with model training data. Record known provenance and exposure, and avoid claims of unseen-condition generalization without a defensible split. Scene-based, edited, generated, and simulated assets require distinct provenance labels.

Operational scope. Physical tests record the enforced operating limits, interventions, and failures relevant to the tasks. Benchmark success supports a claim about the evaluated conditions; broader deployment claims require additional evidence. Dataset, asset, model, and redistribution licenses must be documented for the released artifacts.

9.3 Reproducibility and Release Plan

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply reproducible commands, dependencies, artifacts, licenses, and release items.

The report should be accompanied by task and split manifests, model and adapter identifiers, evaluation configurations, run-status summaries, and per-task artifacts where redistribution is permitted. Reproduction instructions must state the hardware setting, environment dependencies, seeds, launch commands, and the expected output schema. Public claims should resolve to specific result artifacts.

For the current local project, world-model task specifications are stored under `task/`, inference adapters under `script/inference/`, and evaluation entry points under `script/evaluation/`. The existing evaluation materials describe criterion-based video assessment and separate local pipelines for camera trajectory and scene consistency. These paths identify source material for the manuscript; the paper build does not execute model inference or evaluation.

Before release, generate task counts from the frozen manifest, verify rubric and metric versions, resolve incomplete evaluations, and replace every TBD field with a verified configuration

or an explicit exclusion. Publication URLs, author contributions, funding, and availability statements remain to be finalized.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025. URL https://arxiv.org/abs/2501.03575v3.

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691.

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

[4] Zhiqi Bai, Junai Cai, Yixin Chen, Jingrun Du, et al. HappyWorld-Bench. *arXiv preprint arXiv:2609.24308*, 2026. URL https://arxiv.org/abs/2609.24308.

[5] Beijing Academy of Artificial Intelligence. BAAI REAL physical AI evaluation framework (v1). Author-supplied technical presentation, n.d. Technical roadmap: slide 3; capability taxonomy and projections: slides 6–7, 14, 20, 25–26.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

[7] Florian Bordes et al. IntPhys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025.

[8] ByteDance Seed. Seed1.8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*, 2026. URL https://arxiv.org/abs/2603.20633.

[9] François Chaumette and Seth Hutchinson. Visual servo control. I. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006. doi: 10.1109/MRA.2006.250573.

[10] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

[11] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. PhysBench: Benchmarking and enhancing vision-language models for physical world understanding. In *International Conference on Learning Representations (ICLR)*, 2025.

[12] Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees G. M. Snoek, and Yuki M. Asano. TVBench: Redesigning video-language evaluation. *arXiv preprint arXiv:2410.07752*, 2024.

[13] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. *arXiv preprint arXiv:1804.02748*, 2018.

[14] Ronghao Dang et al. RynnBrain 1.1: Towards more capable and generalizable embodied foundation model. *arXiv preprint arXiv:2607.17977*, 2026. URL https://arxiv.org/abs/2607.17977.

[15] DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026. URL https://arxiv.org/abs/2609.19969.

[16] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, et al. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. URL https://arxiv.org/abs/2303.03378.

[17] Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-Bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. *arXiv preprint arXiv:2406.05756*, 2024. URL https://arxiv.org/abs/2406.05756.

[18] Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

[19] GLM-5 Team. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

[20] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

[21] Xiaoshuai Hao et al. MiMo-Embodied: X-embodied foundation model technical report. *arXiv preprint arXiv:2511.16518*, 2025. URL https://arxiv.org/abs/2511.16518.

[22] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, et al. VBench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*, 2023. URL https://arxiv.org/abs/2311.17982.

[23] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. EgoTaskQA: Understanding human tasks in egocentric videos. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2022.

[24] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025.

[25] Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Nandiraju Gireesh, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. MomaGraph: State-aware unified scene graphs with vision–language model for embodied task planning. In *International Conference on Learning Representations (ICLR)*, 2026.

[26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

[27] Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, et al. MiniMax sparse attention. *arXiv preprint arXiv:2606.13392*, 2026. URL https://arxiv.org/abs/2606.13392.

[28] Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, et al. WorldModelBench: Judging video generation models as world models. *arXiv preprint arXiv:2502.20694*, 2025. URL https://arxiv.org/abs/2502.20694.

[29] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Fei-Fei Li, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking LLMs for embodied decision making. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

[30] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. URL https://arxiv.org/abs/2405.05941.

[31] Jieru Lin, Zhiwei Yu, and Börje F. Karlsson. SWITCH: Benchmarking modeling and handling of tangible interfaces in long-horizon embodied scenarios. *arXiv preprint arXiv:2511.17649*, 2025.

[32] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

[33] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, et al. MMBench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. URL https://arxiv.org/abs/2307.06281.

[34] Yujing Lou, Pingyi Chen, Shen Cao, Jiaqi Gu, Jinhui Guo, Jintao Tong, Yunzhuo Hao, Yao Liu, Yue Wu, Lubin Fan, and Jieping Ye. SpatialAxiom: An open spatial intelligence model for general spatial reasoning, July 2026. URL https://d2i-ai.github.io/SpatialAxiom.

[35] Yulin Luo et al. Robobench: A comprehensive evaluation benchmark for multimodal large language models as embodied brain. *arXiv preprint arXiv:2510.17801*, 2025.

[36] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. OpenEQA: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16488–16498, 2024.

[37] Atsuyuki Miyai, Jingkang Yang, Jingyang Zhang, Yifei Ming, et al. Unsolvable problem detection: Robust understanding evaluation for large multimodal models. In *Findings of the Association for Computational Linguistics (ACL)*, 2025.

[38] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024. URL https://arxiv.org/abs/2406.02523.

[39] NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL https://arxiv.org/abs/2606.02800.

[40] others. Core knowledge deficits in multi-modal language models. In *International Conference on Machine Learning (ICML)*, 2025.

[41] others. Unified embodied VLM reasoning with robotic action via autoregressive discretized pre-training. *arXiv preprint arXiv:2512.24125*, 2025.

[42] others. Seeing across views: Benchmarking spatial reasoning of vision-language models in robotic scenes. *arXiv preprint arXiv:2510.19400*, 2025.

[43] others. AQuA: Toward strategic response generation for ambiguous visual questions. *arXiv preprint arXiv:2603.07394*, 2026.

[44] others. EMBER-Bench: Benchmarking cross-event causal memory in long-horizon embodied tasks. *arXiv preprint arXiv:2610.05013*, 2026.

[45] others. ROMA: LLM system for real-world object-centric multi-sensory active perception. *arXiv preprint arXiv:2610.06955*, 2026.

[46] Pelican Team. Pelican-VL 1.0: A foundation brain model for embodied intelligence. *arXiv preprint arXiv:2511.00108*, 2026. URL https://arxiv.org/abs/2511.00108.

[47] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

[48] Zongming Qiao et al. MiMo-V2.6: Scaling reinforcement learning towards self-improvement. *arXiv preprint arXiv:2610.11959*, 2026. URL https://arxiv.org/abs/2610.11959.

[49] Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026. URL https://arxiv.org/abs/2608.30320.

[50] Pierre Sermanet, Anirudha Majumdar, Alex Irpan, Dmitry Kalashnikov, and Vikas Sindhwani. Generating robot constitutions & benchmarks for semantic safety. In *Conference on Robot Learning (CoRL)*, 2025.

[51] Hui Shen et al. PhyX: Does your model have the "wits" for physical reasoning? *arXiv preprint arXiv:2505.15929*, 2025.

[52] TaichuAI. ZDTaichu5.0-9B: A multimodal foundation model for visual and spatial reasoning, 2026. URL https://huggingface.co/TaichuAI/ZDTaichu5.0-9B. Open-weight model release.

[53] Tianyi Tang et al. Causal scaffolding for physical reasoning: A benchmark for causally-informed physical world understanding in VLMs. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Dataset and Benchmark Track*, 2026.

[54] Tencent Robotics X. Hy-Embodied-VLM-1.0: Efficient physical-world agents. *arXiv preprint arXiv:2607.12894*, 2026. URL https://arxiv.org/abs/2607.12894.

[55] Unitree Robotics. UnifoLM-VLA: Vision-language-action model for humanoid robot manipulation, 2026. URL https://huggingface.co/unitreerobotics/UnifoLM-VLA-Base. Open-weight model release.

[56] Yipu Wang et al. Towards cross-view point correspondence in vision-language models. *arXiv preprint arXiv:2512.04686*, 2025.

[57] Haotian Xue, Yunhao Ge, Yu Zeng, Zhaoshuo Li, Ming-Yu Liu, Yongxin Chen, and Jiaojiao Fan. Point-it-out: Benchmarking embodied reasoning for vision language models in precise visual grounding. *arXiv preprint arXiv:2509.25794*, 2025.

[58] Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, et al. RoboChallenge: Large-scale real-robot evaluation of embodied policies. *arXiv preprint arXiv:2510.17950*, 2025. URL https://arxiv.org/abs/2510.17950.

[59] Jinhui Yang et al. VSI-Bench: Visual spatial intelligence benchmark. *arXiv preprint arXiv:2412.14171*, 2024.

[60] Yifu Yuan et al. Embodied-R1.5: Evolving physical intelligence via embodied foundation models. *arXiv preprint arXiv:2606.11324*, 2026. URL https://arxiv.org/abs/2606.11324.

[61] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. *arXiv preprint arXiv:2311.16502*, 2023. URL https://arxiv.org/abs/2311.16502.

[62] Yuhui Zhang, Yuchang Su, Yiming Liu, and Serena Yeung-Levy. NegVQA: Can vision language models understand negation? In *Findings of the Association for Computational Linguistics (ACL)*, 2025.

[63] Enshen Zhou et al. RoboRefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.

[64] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. VLM4D: Towards spatiotemporal awareness in vision language models. In *International Conference on Computer Vision (ICCV)*, 2025.

Contributions and Author List

FlagEval Evaluation Team. The table below records the confirmed leaderboard leads and writing responsibilities. Each lead completes every subsection of the corresponding chapter and supplies that board's settings, data, results, and analysis to shared chapters. Existing prose is a working basis; subsection notes specify the required inputs. 刘璐 and 侯祥锐 jointly own the simulation chapter. The final author list, ordering, equal contributions, corresponding authors, and other contribution records remain to be confirmed.

Leaderboard leads and writing assignments

Introduction writing. 高达辉 and 史可为 jointly write Sections 1.1–1.5. **Related-work writing (Section 1.6).** 史可为 reviews world models; 刘璐 and 侯祥锐 and 郭美衫 jointly review embodied manipulation models; and 武若楠 reviews VLMs. The benchmark subsection has separate world-model, simulation, physical manipulation, and reasoning assignments for the corresponding leaderboard leads. 高达辉 and 史可为 coordinate the section and synthesize unified Physical AI evaluation, coverage gaps, and the positioning of REAL, using evidence from all four teams.

Table 20. Confirmed leaderboard leads and chapter-writing responsibilities.

Leaderboard	Lead(s)	Writing responsibility
Embodied reasoning	武若楠	All subsections of Chapter 3, plus reasoning inputs to shared chapters
Simulated manipulation	刘璐 and 侯祥锐	Jointly complete all subsections of Chapter 4 and simulation inputs to shared studies
Real-world manipulation	郭美衫	All subsections of Chapter 5, plus physical execution and alignment evidence
World models	史可为	All subsections of Chapter 6, world-model scoring details, and shared prediction evidence

Each subsection identifies its writing owner and required inputs. Existing prose is a working basis; owners supply verified settings, data, results, and analysis. This table records responsibilities rather than final author ordering.

Core contributors

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide actual contributors and verified contributions for each board.

- **Benchmark design and methodology:** TBD.
- **Task construction and validation:** TBD.
- **Model integration and experimental execution:** TBD.
- **Evaluation, statistics, and analysis:** TBD.
- **Report writing and release:** TBD.

Contributors

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Provide other contributors and their specific work.

TBD: contributors to data collection, annotation, engineering, infrastructure, physical and simulated evaluation, artifact review, and project coordination.

Author designations and contact

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Confirm author designations and contact details with the team.

If applicable, the final list will identify equal contribution, project leads, and corresponding authors using explicitly defined symbols. No designation is assigned in this draft. Team contact and acknowledgments remain TBD.

Appendix

A Capability System Index and Version Notes

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Review all supplied capability paths and complete definitions, level-5 observations, and executable method/metric/benchmark fields for each assigned board.

This index compiles 14 dimensions and 55 abilities under five domains from the REAL presentation. The current hierarchy, four leaderboard projections, and source pages are maintained in `shared/capability_system.json`. The presentation declares 158 indicators without enumerating all names and formulas; the index therefore ends at level 3 and does not substitute legacy entries for the new indicator inventory.

Source reconciliation. The overview lists cross-task and training stability; the physical and world-model slides also explicitly list long-term stability. Including it yields the stated total of 55 abilities. The world-model slide title and row totals give 52 abilities, so its header's "33" is treated as a typographical error. The overview's "skill adaptation" and "skill and knowledge coverage" are used as shared names for the board slides' "skill learning" and "knowledge and skill coverage."

Legacy inventory. The October 22, 2026 table and November schedule are preserved in `shared/capability_system_20261022.json` for tracing existing tasks and scoring settings. Their mapping to the new indicator inventory still requires verification. Level-5 examples illustrate task design rather than constituting a complete new observation inventory. Owners must supply definitions, indicator details and formulas, task mappings, and execution specifications.

Table 21. PPT-based hierarchy: 14 dimensions and 55 abilities.

Level 2	Level-3 abilities
Basic intelligence	
World understanding	Instruction understanding; Object understanding; Spatiotemporal understanding; Scientific understanding; Causal understanding
World prediction	Kinematic consistency; Dynamic consistency; Scene consistency; Spatial geometric consistency
Autonomous decision-making	Task planning; Task execution; Evaluation and decision-making
Embodied motion	Embodied locomotion; Embodied manipulation; Whole-body coordination; Tool use
Adaptive learning	
Generalization and transfer	Instruction generalization; Object generalization; Scene generalization; Task generalization; Embodiment generalization; Skill-composition generalization
Learning and evolution	Memory; Reflection; Learning by analogy; Lifelong learning; Autonomous exploration

Level 2	Level-3 abilities
Human–machine coexistence	
Multi-robot collaboration	Task collaboration; Motion collaboration; One controller for multiple robots
Human–robot collaboration	Task collaboration; Motion collaboration; Behavior understanding; Social norm understanding; Preference alignment
Human–robot interaction	Natural language interaction; Nonverbal interaction
Safety and reliability	
Safety	Physical safety; Cognitive safety; Proactive safety; Attack resistance
Robustness	Hardware robustness; Task robustness; Temporal robustness
Application practice	
Skill and knowledge coverage	Knowledge coverage; Skill coverage
Application efficiency	Out-of-the-box usability; Skill adaptation; Training and execution efficiency; Model compression; Cross-platform adaptation; Resource efficiency
Application stability	Cross-task stability; Training stability; Long-term stability

B Qualitative Evaluation Examples

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply verified qualitative artifacts and criterion evidence.

This section reserves the layout for evidence-based cases from all four leaderboards. Each case pairs its input, model output, and criterion evidence. The panels below are layout placeholders, not measured successes or failures.

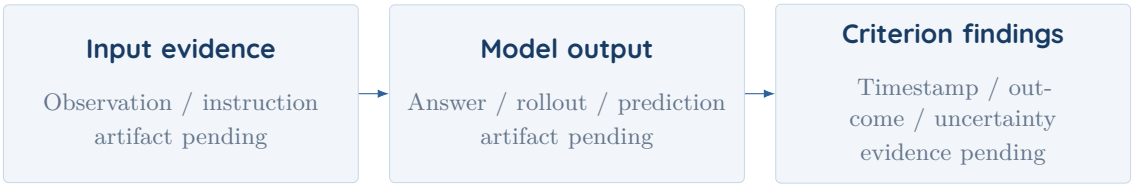


Figure 2. Layout for qualitative evaluation cases. All panels are placeholders awaiting measured artifacts.

B.1 Embodied reasoning cases

Writing owner: 武若楠 **To complete:** Supply observations, questions, model answers, and grading evidence.

Record the original observation, question, allowed history, model answer, and grading evidence. Distinguish a correct answer, unsupported reasoning, and an unresolved judgment under the frozen rubric.

B.2 Real-world manipulation cases

Writing owner: 郭美衫 **To complete:** Supply initial/execution/final frames, state logs, interventions, and outcomes.

Present initial, key execution, and final observations with action or state logs. Mark interventions, stops, and recovery attempts. Verify the success predicate against the physical outcome.

B.3 Simulated manipulation cases

Writing owner: 刘璐 and 侯祥锐 **To complete:** Jointly supply environments, seeds, trajectories, predicates, and variations.

Fix environment, seed, and initialization, then show the policy trajectory and success-predicate output. For generalization cases, declare the factor changed relative to the baseline condition.

B.4 World-model cases

Writing owner: 史可为 **To complete:** Supply conditioning, control timelines, predictions, and timestamped criteria.

Align conditioning observations, the control timeline, and predicted segments. Attach timestamps to criterion findings. Preserve the complete five-level capability path and link judgments through the artifact index.

B.5 Embodied reasoning — level-3 ability breakdown

Tables 22 and 23 provide a finer-grained breakdown of the reasoning leaderboard results at the level-3 ability granularity, supplementing the level-2 dimension view in Table 10.

Table 22. Embodied reasoning — level-3 scores for world understanding and decision-making. **1st**[1], 2nd[2], *3rd*[3] in each column. A dash indicates missing data.

Model	World understanding					Decision-making	
	Causal	*Object*	*Instruction*	*Spatiotemporal*	*Scientific*	*Execution*	*Planning*
Qwen3.8-Max	**72.2**[1]	**81.1**[1]	**71.1**[1]	**66.2**[1]	89.7	**96.0**[1]	**76.8**[1]
Doubao-Seed	71.7[2]	*76.0*[3]	*69.5*[3]	64.0[2]	**100.0**[1]	90.0[2]	76.4[2]
Qwen3.8-Flash	62.4	76.1[2]	69.7[2]	*63.7*[3]	*97.4*[3]	80.0	70.8
Qwen3.8-27B	58.7	74.8	68.3	61.6	94.9	82.0	*71.6*[3]
GLM-5.3-Flash	*68.4*[3]	69.6	65.5	53.5	94.9	88.0	66.6
MiMo-v2.6-Pro	57.5	65.4	66.4	49.0	84.6	78.0	63.8
DeepSeek-Flash	60.2	65.4	66.9	46.0	94.9	64.0	65.0
MiniMax-M3	61.9	61.9	62.9	44.6	87.2	*88.0*[3]	67.0
MiMo-v2.6-Flash	57.5	60.7	60.8	49.5	89.7	78.0	63.9
Gemma-4-26B	58.9	59.8	59.9	37.5	87.2	70.0	65.0
Open-source embodied reasoning models							
SpatialAxiom	60.2	55.7	55.5	58.0	92.3	39.3	49.9
Pelican-72B	63.6	57.0	53.8	50.8	82.0	32.1	59.5
RynnBrain-122B	64.5	54.0	63.7	59.5	74.4	22.0	52.3
ZDTaichu	60.1	57.2	60.5	47.1	92.3	49.3	60.3
RynnBrain-9B	65.0	49.4	56.3	53.4	100.0[2]	46.5	63.8
CosmosR2-32B	57.4	55.3	55.2	46.6	56.4	41.1	65.7
HY-A3B	62.4	58.1	63.2	56.6	–	32.5	55.8
MiMo-ER-7B	56.7	56.2	50.7	43.3	94.9	49.4	52.4
EmbodiedR-15	58.0	50.9	50.5	47.0	48.7	36.9	46.3
UniFoLM-ER-1	52.3	53.3	48.2	47.7	41.0	25.0	24.2
CosmosR2-8B	58.3	49.0	45.4	41.6	38.5	34.5	41.1

Model names are abbreviated for space. Models follow the order in Table 10; its aggregates remain provisional where ability scores are missing.

Table 23. Embodied reasoning — level-3 scores for world prediction and learning. **1st**[1], 2nd[2], *3rd*[3] in each column. A dash indicates missing data.

Model	World prediction					Learning and evolution		
	Dyn. consist.	Reflection	Scene consist.	Spatial geom.	Kinem. consist.	Reflection	Exploration	Memory
Qwen3.8-Max	67.4[2]	84.5[2]	74.6[2]	**69.9**[1]	74.6[2]	**78.7**[1]	**77.1**[1]	–
Doubao-Seed	**67.4**[1]	**84.5**[1]	*70.5*[3]	63.4[2]	**75.3**[1]	*72.4*[3]	67.0	–
Qwen3.8-Flash	57.6	*81.8*[3]	**75.3**[1]	*62.2*[3]	67.6	71.3	71.1[2]	–
Qwen3.8-27B	*58.3*[3]	79.6	70.2	59.1	*68.5*[3]	74.7[2]	*70.1*[3]	–
GLM-5.3-Flash	55.8	79.8	66.9	51.7	67.4	71.8	62.9	–
MiMo-v2.6-Pro	42.8	78.6	61.3	41.4	59.7	60.6	59.6	–
DeepSeek-Flash	48.7	71.1	61.0	38.4	58.6	59.5	53.6	–
MiniMax-M3	45.3	71.6	62.8	42.2	64.5	62.1	53.9	–
MiMo-v2.6-Flash	47.6	71.6	55.8	39.9	54.6	61.5	52.8	–
Gemma-4-26B	41.5	64.3	48.3	23.8	52.8	55.2	55.4	–
Open-source embodied reasoning models								
SpatialAxiom	49.6	65.5	56.5	48.0	61.9	40.7	38.8	*30.9*[3]
Pelican-72B	52.0	78.2	54.3	49.5	57.6	36.0	31.9	29.0
RynnBrain-122B	37.3	66.7	50.1	53.8	60.3	20.1	27.3	**37.2**[1]
ZDTaichu	43.1	64.1	56.0	45.3	55.8	40.1	33.1	27.9
RynnBrain-9B	39.9	71.8	53.5	44.2	61.0	40.0	33.0	28.2
CosmosR2-32B	45.8	66.0	51.1	47.8	56.7	40.1	25.6	31.5[2]
HY-A3B	52.2	41.7	50.5	50.1	59.7	22.7	23.4	0.0
MiMo-ER-7B	45.6	52.9	55.8	42.1	53.0	34.5	41.4	27.2
EmbodiedR-15	48.5	51.7	48.1	48.4	52.9	34.8	21.0	24.3
UniFoLM-ER-1	47.6	50.0	46.5	45.3	50.6	35.1	37.4	23.5
CosmosR2-8B	47.1	55.3	51.9	46.6	50.9	36.9	21.2	26.2

Model names are abbreviated for space. Models follow the order in Table 10; its aggregates remain provisional where ability scores are missing.

C Task and Run Record Templates

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Map actual task/run fields to the shared record template.

The following is a proposed interchange record for the report, rather than an existing executable `uniwmbench` task schema. An adapter should map actual task fields into this record while preserving their meaning.

```
{
  "task_id": "<stable identifier>",
  "release_id": "<frozen release>",
  "leaderboard": "<reasoning|real_manip|sim_manip|world_model>",
  "capability_primary": "<unique scored level-4 point ID>",
  "capability_path": {
    "level_1": "<capability domain>",
    "level_2": "<dimension>",
    "level_3": "<ability>",
    "level_4": "<assessment point>",
    "level_5": ["<relevant observational aspects>"]
  },
  "ability_definition": "<meaning of the level-3 construct>",
  "matched_concept_id": "<optional cross-board identifier>",
  "input": {
    "instruction": "<model-visible instruction>",
    "observations": ["<artifact references and fingerprints>"],
    "allowed_state": "<explicit information permissions>",
    "control_interface": "<type, frame, units, frequency>"
  },
  "budget": "<time, attempts, steps, inference budget>",
  "split": "<development|test|held_out_condition>",
  "provenance": "<source, transformations, license>",
  "evaluation": {
    "task_format": "<question, prediction, or execution>",
    "scoring_method": "<tool, judge, human, or combination>",
    "metric_version": "<definition, denominator, aggregation>",
    "benchmark": "<adopted source or newly constructed suite>",
    "private_rubric": "<hidden criteria and evidence rules>"
  }
}
```

A run record should preserve its scheduled identity even when execution or evaluation fails. Criterion evidence includes timestamps or state references sufficient for independent review.

```
{
  "run_id": "<unique attempt identifier>",
  "task_id": "<task identifier>",
  "model_revision": "<checkpoint hash or service revision>",
  "adapter_revision": "<source revision>",
  "input_fingerprints": "<task and observation hashes>",
  "seed": "<recorded seed, or unavailable>",
  "setting_id": "<robot/simulator/control comparison group>",
  "status": "<resolved outcome or explicit error category>",
  "artifacts": ["<video, trajectory, answer, logs>"],
  "outcome": "<success predicate or criterion judgments>",
  "evaluator_revision": "<metric/judge/prompt revision>",
  "elapsed_seconds": "<measured duration>"
}
```

D Publication Completion Checklist

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Complete each board's publication evidence and outstanding items.

Table 24. Evidence required before publication.

Item	Evidence to supply
Authors and affiliations	Confirmed names, ordering, contact, contributions, acknowledgments
Task inventory	Release ID, generated counts, splits, provenance
Leaderboard protocols	Interfaces, groups, metrics, weights, budgets
Real-world setup	Robot, cameras, controllers, calibration, resets, verification
Simulation setup	Simulator and assets, seeds, observations, success predicates
World-model setup	Controls, horizons, preprocessing, evaluator routing, validity
Submissions	Revisions, adaptation, training exposure, resource budgets
Judge and statistics	Calibration, uncertainty, ranking policy
Framework plan	Full taxonomy, board specifications, validated coverage, PSR/TEQ definitions
Alignment study	Matched tasks, paired budgets, coordinates, timing, transfer evidence
GPT6 VLM study	VLM revision, sim/real tasks, skills, executors, call budgets and evidence
Results and release	Artifacts, failure accounting, supported findings, reproduction, licenses

E Complete Evaluation Data and Artifact Index

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Supply per-board data, results, status, review, and artifact indexes.

Generate the final attachment from the frozen manifest, indexing inputs, outputs, criterion evidence, statuses, and review records by leaderboard, comparison group, task, model revision, and run identity. Include splits, provenance and licenses, configuration fingerprints, invalidated trials, and missing-result disclosures. Restricted material receives public metadata and declared access rules. This draft defines the index requirements without supplying unverified result data.

F Repository Sources for This Draft

Writing owner: 武若楠; 刘璐 and 侯祥锐; 郭美衫; 史可为 **To complete:** Verify local sources, versions, provenance, and references.

The manuscript structure is grounded in the existing local materials below. Some source documents are historical snapshots; their numerical counts have not been treated as a frozen benchmark release.

- The supplied Physical AI report screenshots and their Markdown transcription: sources for the nine-chapter organization, framework planning targets, alignment study, and model-role study. Planning counts do not replace the scoring schedule.
- The November 2026 world-model capability evaluation table supplied by the project owner: authoritative for the four displayed second-level dimensions, the 33-item scheduled scope, and the 3,300-point additive scoring rule.
- `document/CAPABILITY_TAXONOMY.md`: capability vocabulary and task design.
- `task/README.md`: task organization, revision notices, and readiness notes.
- `script/evaluation/README.md`: current evaluation entry points and artifacts.
- The leaderboard overview dated 2026-10-13 under `document/`: world-model grouping and evidence-based scoring design.
- The evaluation integration and result-display specification under `document/`: inference inputs, output artifacts, status handling, and provenance.