OMNIBRAIN-0: A UNIFIED AGENTIC MODEL FOR EMBODIED INTELLIGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

General-purpose embodied intelligence requires not only individual skills but also long-horizon planning, skill coordination, and recovery toward task goals. Specialist robot policies alone do not ensure task-level reliability, while powerful multimodal agents depend on costly long-context reasoning and external experience that compact models cannot readily reproduce. We introduce **OmniBrain-0**, a lightweight unified embodied agent that brings manipulation and navigation into a shared decision framework. Rather than replacing specialist controllers, OmniBrain-0 orchestrates heterogeneous skills through a single high-level model, integrating task understanding, planning, progress tracking, and feedback-driven recovery. To transfer and integrate these capabilities, we develop a progressive training framework combining embodied pretraining, domain-specialized agent distillation, and multi-expert consolidation with MOPD. This framework first converts powerful teachers' interaction experience into specialized decision capabilities, then consolidates manipulation and navigation expertise into a compact unified agent. Crucially, unification occurs within the learned decision model rather than only at the tool interface. We evaluate OmniBrain-0 on LIBERO-Pro, RoboCasa365, RoboTwin 2.0, and R2R-CE Val-Unseen, covering manipulation and navigation. We further evaluate OmniBrain-0 on three real-world manipulation tasks, demonstrating its applicability to physical robot execution. Results show strong performance across both domains and substantial gains over the compact zero-shot backbone, supporting effective transfer and consolidation of agentic planning and recovery capabilities across task domains.

1 INTRODUCTION

General-purpose embodied intelligence requires diverse physical skills, long-horizon planning, skill composition, and closed-loop correction to fulfill task intent. Vision-language-action (VLA) models and vision-and-language navigation (VLN) policies advance manipulation and navigation Kim et al. (2025); Black et al. (2025a); Wei et al. (2026a), but reliable individual skills do not guarantee task-level success. Under complex constraints, multi-stage dependencies, or distribution shifts, local action prediction may fail to maintain goal alignment, requiring feedback-driven reassessment and replanning Zhang et al. (2026). The challenge extends beyond skill accuracy to task-level coordination and recovery. Large-model-based embodied agents offer a promising approach by exposing perception, specialist policies, and motion controllers as tools. A high-level model interprets instructions, decomposes tasks, orchestrates skills, and uses execution feedback to track progress and recover from failures, while low-level policies retain their physical interaction expertise Zhang et al. (2026); Liang et al. (2023); Huang et al. (2022). This division of labor enables goal- and feedback-driven execution, yet reliable coordination requires reasoning about skill applicability, capability boundaries, and handoff states Huang et al. (2026). For manipulation and navigation, the goal is therefore not to replace all controllers, but to establish a shared high-level decision mechanism for consistent task understanding, skill orchestration, and adaptation to execution deviations.

However, solving embodied tasks with powerful models does not explain how to learn unified agents from interaction. These models combine contextual reasoning, experience, and environmental feedback Zhang et al. (2026); Huang et al. (2026), yet learning, retaining, and extending their capabilities across domains requires dedicated training. Repeated reasoning during long-horizon interaction incurs computational overhead and decision latency, constraining deployment efficiency Chen et al.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

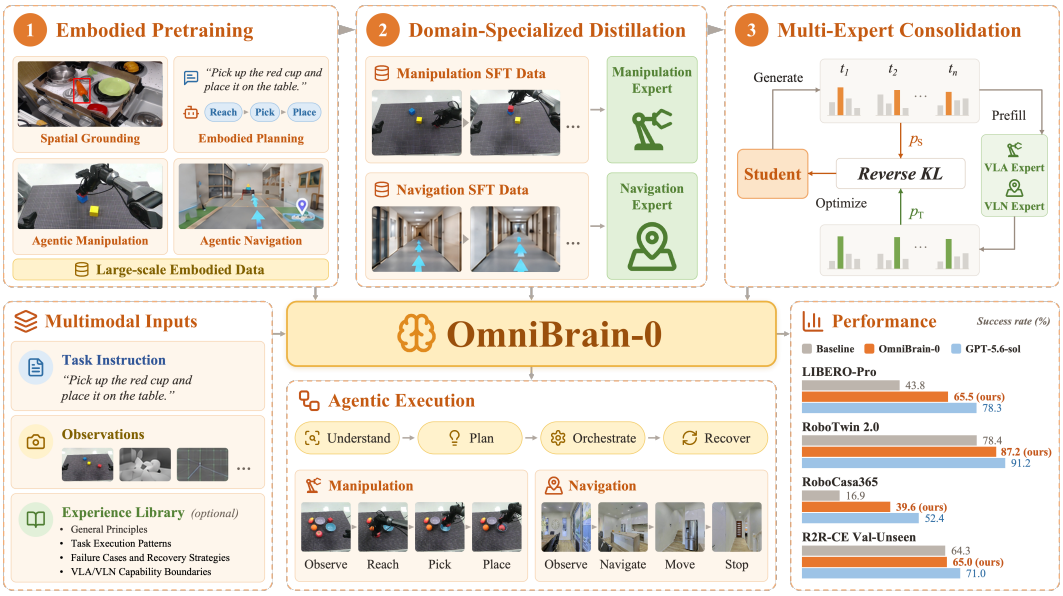

Figure 1: **Overview of OmniBrain-0.** *Top:* Three-stage training: embodied pretraining, domain-specialized distillation into manipulation and navigation experts, and multi-expert consolidation (MOPD) into a single compact agent. *Bottom:* Given a task instruction, observations, and an optional experience library, OmniBrain-0 understands, plans, orchestrates skills, and recovers from failures across manipulation and navigation. The bar chart reports success rates (%) on four benchmarks, comparing OmniBrain-0 with the baseline policy ($\pi_{0.5}$ for LIBERO-Pro, RoboTwin 2.0, and RoboCasa365; InternVLA-N1 for R2R-CE Val-Unseen) and the zero-shot teacher GPT-5.6-Sol.

(2025b). We therefore ask: *How can powerful models' embodied interaction experience be transformed into learnable planning and recovery capabilities, and how can domain-specific expertise be integrated into a unified, deployable embodied decision model?*

We view this problem as the *transfer and integration of embodied agent capabilities*. First, the goal is to transfer effective decision behaviors from teacher interactions, rather than reproduce teachers' language outputs or reasoning processes Kang et al. (2025). Because embodied decisions alter the environment and subsequent observations, students must learn to interpret execution feedback, track progress, and adapt when plans become invalid. Fixed demonstrations alone cannot ensure appropriate decisions during autonomous execution Huang et al. (2022); Agarwal et al. (2024). Second, specialized capability acquisition and unified learning are distinct problems. Manipulation and navigation differ in scene distributions, skill interfaces, and task structures, so expertise learned by specialized planners must be further integrated for joint deployment Ma et al. (2026). Accordingly, capability learning involves two transformations: *from powerful models' interaction experience to specialized agent capabilities, and from multiple specialized planners to a shared decision model*.

Therefore, we introduce **OmniBrain-0**, a **lightweight, unified embodied agent for manipulation and navigation** (Figure 1). OmniBrain-0 integrates specialist policies and tools through a shared skill-invocation framework. A compact high-level planner performs task understanding, long-horizon planning, skill orchestration, state tracking, and failure recovery. Conditioned on observations, instructions, and interaction history, it selects skills and adapts to execution feedback, preserving low-level specialization while learning task decomposition and adaptive correction.

To realize these capability transformations, we develop a progressive training framework comprising *shared embodied initialization, specialized capability acquisition, and cross-domain capability integration*. First, we combine embodied, question-answering, and general language data with limited long-horizon interaction data to establish a shared initialization for a compact model. We then train manipulation and navigation planners from this initialization using quality-filtered interaction trajectories generated by multiple powerful models. By learning observation–decision–feedback relationships from standard successful execution and successful recovery, these planners acquire the teachers' planning and correction behaviors. Finally, we adopt multi-teacher on-policy distilla-

tion (MOPD) Ma et al. (2026), using corresponding experts to supervise student-generated decision sequences and integrate specialized capabilities into a unified student initialized from the shared pretrained model. This process consolidates teacher interaction experience and domain-specific decision-making expertise into the lightweight OmniBrain-0, enabling a single compact planner to orchestrate long-horizon tasks across manipulation and navigation.

Our main contributions are summarized as follows:

- **A unified agentic model for manipulation and navigation.** We introduce OmniBrain-0, a compact high-level agent that coordinates heterogeneous perception, manipulation, navigation, and motion skills through a shared decision model for planning, orchestration, and recovery.
- **Closed-loop embodied capability transfer from powerful agents.** We construct structured interaction trajectories that preserve observation–decision–feedback dependencies and recovery, transferring teachers' planning and corrective behaviors into compact domain-specialized agents.
- **Progressive specialization and multi-expert consolidation.** We combine embodied pretraining, domain-specialized distillation, and MOPD to integrate manipulation and navigation expertise into a single lightweight model while retaining specialized low-level skills.

2 Related Work

Specialist policies for manipulation and navigation. VLA models such as π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), and LingBot-VLA (Wu et al., 2026a;b) learn multitask control from large-scale robot data and vision-language pretraining. Yet even $\pi_{0.5}$ and LingBot-VLA 2.0 degrade substantially on long-horizon tasks under object and initial-position shifts (Wu et al., 2026b). For navigation, NavGPT-2 (Zhou et al., 2024) couples language reasoning with a navigation policy, and DualVLN (InternVLA-N1) (Wei et al., 2026a) connects high-level planning to low-level execution in a dual-system design. Rather than replacing these policies, OmniBrain-0 invokes them as skills and learns a shared high-level decision model across both domains.

Tool-using embodied agents. SayCan (Ahn et al., 2022) and Code as Policies (Liang et al., 2023) ground task decomposition in executable skills, and Inner Monologue (Huang et al., 2022) feeds environmental feedback back into planning. Agent harnesses, including Harness VLA (Zhang et al., 2026), RoboHarness (Huang et al., 2026), and Pigey (Galanti et al., 2026), add state tracking, outcome verification, and recovery through agent loops. However, their orchestration relies on repeated reasoning by large models at deployment time. OmniBrain-0 instead internalizes planning, progress tracking, and recovery into a compact decision model learned from interaction experience.

Agent distillation and expert consolidation. AgentTuning (Zeng et al., 2024) and STEVE-2 (Zhao et al., 2024) transfer agent capabilities via trajectory tuning and hierarchical distillation, but fixed teacher trajectories may not cover states encountered by students. On-policy distillation mitigates this mismatch by supervising student-generated sequences (Agarwal et al., 2024); TurnOPD (Zhou et al., 2026) adds turn-aware supervision for agents, and MOPD (Ma et al., 2026) merges domain-specialized teachers. In embodied settings, each decision alters subsequent observations and skill applicability; OmniBrain-0 applies multi-teacher distillation to consolidate manipulation and navigation expertise, including planning and recovery, into a single model.

3 Method

3.1 Problem Formulation and Overview

We consider language-conditioned manipulation and navigation tasks, with tool invocation serving as a unified high-level action representation. Given a task instruction g, OmniBrain-0 selects tools based on current observations and interaction history, continually updating its decisions through environmental feedback. At decision step t, the observation and interaction context are defined as

$$o_t = (I_t, q_t, p_t), \qquad \mathcal{H}_t = (h_t, \mathcal{M}), \tag{1}$$

where I_t denotes visual observations, q_t the robot's proprioceptive state, and p_t task-status information exposed by the environment. The within-episode interaction history is denoted by h_t, while $\mathcal{M}$

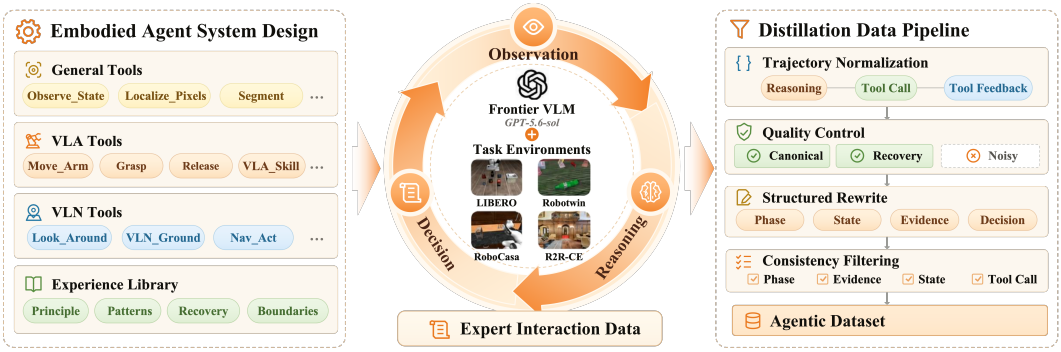

Figure 2: **Agentic execution framework and distillation data pipeline.** *Left:* General, VLA, and VLN tools with an experience library. *Center:* Expert models collect trajectories through closed-loop interaction with task environments. *Right:* Trajectory normalization, quality control, structured rewriting, and consistency filtering convert expert interaction records into an agentic training dataset.

represents optional experience library. Decisions are conditioned only on information available at the current or previous steps, with experience library providing priors for task organization and skill use.

Let $\mathcal{P}_e$ denote the available tool set, the high-level policy π_θ generates a tool identifier $k_t \in \mathcal{P}_e$ and invocation parameters ξ_t:

$$\begin{aligned} x_t &= (g, o_t, \mathcal{H}_t), \\ u_t &= (k_t, \xi_t) \sim \pi_\theta(\cdot \mid x_t), \qquad k_t \in \mathcal{P}_e. \end{aligned} \tag{2}$$

The selected tool performs the corresponding perception or control operation and returns a new observation o_{t+1} and execution feedback f_{t+1}, which condition subsequent decisions. The objective is to fulfill the instruction within a given execution budget, with completion determined by the environment's task criteria. This formulation provides a unified high-level tool interface for manipulation and navigation, while preserving their distinct low-level action spaces and execution durations.

3.2 Serial Closed-Loop Architecture of OmniBrain-0

The agentic execution framework of OmniBrain-0 follows the system design of Harness VLA (Zhang et al., 2026), with its tools and experience library summarized in Figure 2 (left). OmniBrain-0 uses the general-purpose multimodal model Qwen3.5-9B Qwen Team (2026) as a shared agent, decoupling task-level decision-making from specialized tool execution. The agent performs instruction understanding, long-horizon task decomposition, tool orchestration, and failure recovery, while execution modules provide perception, manipulation, navigation, and motion-control capabilities. A unified invocation-and-feedback interface connects the agent and execution modules in a closed loop, enabling the same decision model to coordinate heterogeneous tools with different inputs and control mechanisms.

Unified tool abstraction. We encapsulate heterogeneous execution capabilities as parameterized semantic tools. Perception tools provide scene understanding and target localization; VLA tools handle manipulation and contact interactions; navigation tools support goal reaching through VLN policies or geometric navigators; and motion-control tools provide explicit pose and actuator adjustments. Each tool retains its native conditioning and low-level implementation while exposing a unified invocation-and-feedback interface. Thus, the agent shares a common high-level decision mechanism and interaction protocol rather than a single low-level control policy.

Context-driven tool orchestration. The agent jointly processes task instructions, current observations, and historical feedback to select subsequent tools and determine their invocation parameters. Interaction history supports task progress tracking, while optional experience library provides priors on task organization, tool composition, and recovery strategies. Experience guides task-level organization, whereas online perception grounds tool invocation in the current execution state.

Serial closed-loop execution and recovery. At each turn, the agent selects one tool and makes the next decision only after receiving its execution feedback.High-level tool invocations are executed sequentially and connected through execution outcomes. The agent continuously updates its decisions based on observations and feedback, adapting target localization, invocation parameters, or tool selection when execution deviates from expectations. Detailed tool definitions and implementation specifications are provided in the Appendix C.

3.3 Embodied Agent Capability Transfer and Consolidation

We develop a progressive capability transfer framework that first collects successful interaction experience from powerful expert models, then distills domain-specific decision-making capabilities into specialized teachers, and finally consolidates these teachers into a unified shared model through multi-teacher on-policy distillation. This process separates capability acquisition from cross-domain integration, allowing the shared agent to inherit complementary planning, tool orchestration, and recovery capabilities while preserving the specialization of low-level execution tools.

3.3.1 Expert Interaction Trajectories

As illustrated in Figure 2 (center), we collect closed-loop interaction trajectories from multiple powerful models on manipulation and navigation tasks. Experts invoke tools within an execution framework that a Meta-Agent iteratively refines, continually adjusting decisions based on task objectives, online observations, and execution feedback. Each trajectory is organized as a temporally ordered, multi-turn multimodal sequence:

$$\tau = (g, o_0, y_0, u_0, f_1, o_1, \ldots, y_{T-1}, u_{T-1}, f_T, o_T)\,, \tag{3}$$

where y_t denotes the expert's decision text, u_t the executed skill invocation, and f_{t+1} the corresponding execution feedback; o_t follows the definition in Section 3.1. We preserve the temporal relationships among observations, decisions, and feedback, ensuring that each decision is grounded in information available at that step.

Figure 2 (right) summarizes the data pipeline for converting expert interaction records into training trajectories. Training trajectories are first selected based on task success and interaction completeness, retaining effective decision behaviors from both standard execution and successful recovery. Through semantic normalization and interface alignment, expert experience is converted into supervision compatible with the observations and tool interfaces available to the student. We then remove redundant behaviors and expression variations while preserving the temporal structure of tool invocations, focusing supervision on the relationships among task states, observations, and tool selection. The resulting dataset captures task progression and adaptive decisions under execution deviations, supporting the transfer of both planning and recovery capabilities. Dataset composition, splits, and processing details are provided in the experimental setup and appendix.

3.3.2 Progressive Specialization and Consolidation

As shown in Figure 1 (top), we adopt a three-stage progressive training framework that establishes a shared initialization through embodied pretraining, acquires specialized decision-making capabilities for manipulation and navigation from expert interaction trajectories, and integrates planning, skill orchestration, and recovery expertise into a unified agent through multi-teacher distillation. By decoupling specialized capability acquisition from cross-domain integration, this framework enables a single lightweight model to jointly retain decision-making expertise from different domains.

Stage I: Embodied Pretraining. We pretrain on a mixture of embodied question-answering data and long-horizon agent interaction data to establish a shared basis for embodied capabilities in manipulation and navigation. Embodied question answering provides supervision for scene understanding and task-related knowledge, while long-horizon interactions guide the model to organize decisions and use feedback in multi-turn tasks through temporal relationships among instructions, observations, tool invocations, and execution feedback. These complementary sources connect static embodied understanding with dynamic interactive decision-making, providing shared knowledge and interaction priors for subsequent specialization. The resulting model, denoted as **OmniBrain-Base** with parameters θ_{base}, initializes both the domain teachers and the unified student.

Stage II: Domain-Specialized Agent Distillation. We initialize domain-specialized teachers for manipulation and navigation from OmniBrain-Base and train them through supervised hard distillation on expert interaction trajectories. Each teacher learns goal understanding, skill orchestration, and recovery decisions appropriate to its domain's task structure. These teachers are high-level agent models: they learn to organize and use skills rather than generate low-level actions.

Let ϕ_k denote the parameters of teacher k and $\mathcal{D}_k$ its training dataset. Given a context x comprising task instructions, observations, interaction history, and skill definitions, we serialize the teacher's decision text and executed invocation into a target sequence z and minimize the autoregressive negative log-likelihood:

$$\mathcal{L}_{\text{expert}}^{(k)} = -\mathbb{E}_{(x,z)\sim\mathcal{D}_k}\left[\frac{1}{|z|}\sum_{j=1}^{|z|}\log \pi_{\phi_k}(z_j \mid x, z_{<j})\right], \qquad \phi_k^{(0)} = \theta_{\text{base}}. \tag{4}$$

Supervision is applied only to the agent's decision text and tool invocations, while environmental observations and tool feedback serve as conditioning information. By jointly learning from standard execution and recovery trajectories, specialization transforms expert interaction experience into observation-conditioned, task-level decision-making capabilities, yielding domain teachers for subsequent cross-domain integration.

Stage III: Multi-Teacher Consolidation with MOPD. We adopt multi-teacher on-policy distillation (MOPD) Ma et al. (2026) to integrate domain teachers' distributional supervision over student-generated decision prefixes. The unified student is initialized from OmniBrain-Base, and the domain-specialized agents obtained in Stage II form a fixed teacher set:

$$\theta^{(0)} = \theta_{\text{base}}, \qquad \mathcal{T} = \{\pi_{\phi_k}\}_{k=1}^{K}, \tag{5}$$

where K is the number of teachers. The student and teachers share the same model architecture and vocabulary. Each teacher provides domain-specific supervision, jointly optimizing the unified student.

Teacher Supervision on Student Prefixes. We sample domain k with weight ρ_k, where $\rho_k \geq 0$ and $\sum_k \rho_k = 1$. For a domain context $x \sim \mu_k$, a student decision sequence is sampled as $\hat{z} \sim \pi_{\bar{\theta}}(\cdot \mid x)$, where $\bar{\theta}$ denotes the student parameters used for sampling. The teacher provides token-level distributional supervision conditioned on the same context and the student-generated prefix:

$$\begin{aligned} p_{\theta,j}(v) &= \pi_\theta(v \mid x, \hat{z}_{<j}), \\ q_{k,j}(v) &= \pi_{\phi_k}(v \mid x, \hat{z}_{<j}), \qquad v \in \mathcal{V}, \end{aligned} \tag{6}$$

where $\mathcal{V}$ is the shared vocabulary. At this stage, distribution alignment is conditioned on student-generated prefixes, allowing teacher supervision to adapt to the student's generation behavior.

Joint multi-domain distillation objective. We minimize the token-level reverse Kullback–Leibler (KL) divergence from the student to the corresponding expert:

$$\begin{aligned} \mathcal{L}_{\text{MOPD}}(\theta) &= \sum_{k=1}^{K}\rho_k \mathbb{E}_{\substack{x\sim\mu_k \\ \hat{z}\sim\pi_{\bar{\theta}}(\cdot|x)}}\left[\frac{1}{M}\sum_{j=1}^{|\hat{z}|} m_j\, D_{\text{KL}}(p_{\theta,j} \,\|\, q_{k,j})\right], \\ M &= \sum_{j=1}^{|\hat{z}|} m_j > 0, \end{aligned} \tag{7}$$

where $m_j \in \{0, 1\}$ marks only valid agent decisions and tool-invocation outputs. The divergence is defined as

$$D_{\text{KL}}(p_{\theta,j} \,\|\, q_{k,j}) = \sum_{v\in\mathcal{V}} p_{\theta,j}(v)\log\frac{p_{\theta,j}(v)}{q_{k,j}(v)}. \tag{8}$$

This objective weights distributional discrepancies by student prediction probabilities, using the decision preferences of different teachers to constrain the student's current output distribution. Teacher parameters and sampled prefixes remain fixed during optimization, and only the unified student parameters θ are updated. By decoupling specialization from cross-domain integration, we consolidate manipulation and navigation expertise into the lightweight **OmniBrain-0**, which orchestrates cross-domain skills with a single set of high-level agent parameters.

4 EXPERIMENT

In this section, we first introduce the experimental setup, including the evaluation benchmarks and implementation details. We then analyze the performance of OmniBrain-0 and visualize its task execution. More ablation studies are provided in Appendix G.

4.1 EXPERIMENTAL SETUP

Evaluation Benchmarks We evaluate OmniBrain-0 on three manipulation benchmarks, one navigation benchmark, and a real-world suite: (1) **LIBERO-Pro** Zhou et al. (2025): VLA evaluation under systematic perturbations of objects, states, instructions, and environments; (2) **RoboCasa365** Nasiriany et al. (2026): Household manipulation with 365 tasks for multitask and long-horizon evaluation; (3) **RoboTwin 2.0** Chen et al. (2025a): Bimanual manipulation with randomized tasks and environments; (4) **R2R-CE** Krantz et al. (2020): Continuous-environment vision-and-language navigation for indoor instruction following; and (5) **Real-world**: Manipulation tasks include block stacking, apple placement, and drawer opening. More details and evaluation protocols are provided in Appendix E.

Implementation Details To collect training data for OmniBrain-0, we use multiple expert models, including GPT-5.6-Sol and GLM-5.3-Flash, to perform closed-loop embodied interactions across four simulated benchmarks and collect diverse trajectories. We then filter and process these trajectories with our data pipeline, yielding 1,902, 427, 614, and 903 agentic training samples from LIBERO-Pro, RoboTwin 2.0, RoboCasa365, and R2R-CE, respectively. For embodied pretraining, we collect approximately 500K question-answering samples from ShareRobot Ji et al. (2025) and RoboInter-VQA Li et al. (2025), covering spatial grounding, affordance understanding, and task planning. Further details and training hyperparameters are provided in Appendix F.

4.2 MAIN RESULT

LIBERO-Pro. Table 1 summarizes results on LIBERO-Pro across SPATIAL, OBJECT, GOAL, and LIBERO-10 under instruction-redirection (T) and position-swap (S) perturbations, together with average task execution time. The results show that strong expert models achieve high success rates but incur substantial execution overhead. In contrast, OmniBrain-0 reaches 65.5% overall success with an average execution time of only 1.6 minutes, substantially improving over the zero-shot backbone while remaining far more efficient than large expert models. Its gains are consistent across multiple perturbation types, indicating improved robustness to task and spatial changes. It also outperforms larger zero-shot models, demonstrating that distilling agentic planning and recovery capabilities provides a favorable balance between task performance and execution efficiency.

Table 1: Success rates (%) on LIBERO-Pro across the SPATIAL, OBJECT, GOAL, and L10 suites under Task (**T**) and Swap (**S**) perturbations. Time denotes the average task completion time in minutes; "/" indicates unavailable runtime. † denotes zero-shot evaluation.

Method	Success Rate (%) ↑									Time ↓
	Spat-T	Spat-S	Obj-T	Obj-S	Goal-T	Goal-S	L10-T	L10-S	Overall	
OpenVLA Kim et al. (2025)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	/
π_0 Black et al. (2025b)	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.3	/
X-VLA Zheng et al. (2026)	0.0	0.0	8.0	2.0	9.0	1.0	10.0	0.0	3.8	/
CaP-Agent0	14.0	12.0	18.0	22.0	17.0	26.0	2.4	5.2	14.6	/
$\pi_{0.5}$ Black et al. (2025a)	41.0	52.0	56.0	69.0	42.0	36.0	46.0	8.0	43.8	**0.6**
Qwen3.5-9B†	6.0	11.0	28.0	13.0	13.0	20.0	22.0	12.0	15.6	6.2
Qwen3.8-27B†	13.0	18.0	55.0	48.0	24.0	17.0	24.0	9.0	26.0	9.2
GLM-5.3-Flash†	68.0	67.0	76.0	80.0	69.0	71.0	52.0	36.0	64.9	8.5
GPT-5.6-Sol†	**92.0**	**81.0**	**95.0**	**97.0**	**84.0**	**83.0**	**57.0**	**37.0**	**78.3**	13.8
OmniBrain-0 (Ours)	78.0	75.0	93.0	96.0	64.0	52.0	44.0	22.0	65.5	1.6

RoboCasa365. Table 2 summarizes the evaluation results on the RoboCasa365 benchmark across Atomic-Seen, Composite-Seen, and Composite-Unseen scenarios. GPT-5.6-Sol still demonstrates a high performance ceiling and exceptional task generalization, achieving top success rates across all task splits. Meanwhile, Our OmniBrain-0 exhibits an impressive and well-balanced overall capability: it achieves an overall success rate of 39.6%, outperforming prior end-to-end VLA baselines such as RLDX-1, achieving 3× the success rate of zero-shot Qwen3.5-9B, and approaching the performance of GLM.

Table 2: Success rates (%) on RoboCasa365 over the Atomic-Seen, Composite-Seen, and Composite-Unseen task splits. † denotes zero-shot evaluation.

Method	Success Rate (%) ↑			
	Atomic-Seen	Composite-Seen	Composite-Unseen	Overall
$\pi_{0.5}$ Black et al. (2025a)	39.6	7.1	1.2	16.9
GR00T N1.5 NVIDIA et al. (2025)	50.7	14.8	2.7	23.9
RLDX-1 Kim et al. (2026)	67.6	27.9	8.5	36.0
Qwen3.5-9B†	28.3	6.3	0.0	12.2
Qwen3.8-27B†	37.2	11.3	2.5	17.8
GLM-5.3-Flash†	74.4	28.8	13.8	40.4
GPT-5.6-Sol†	**81.1**	**45.0**	**27.5**	**52.4**
OmniBrain-0 (Ours)	71.1	32.5	11.3	39.6

Table 3: Success rates (%) on RoboTwin 2.0. `Clean` denotes the standard setting with fixed scene configurations, while `Randomized` applies domain randomization. † denotes zero-shot evaluation.

Method	Success Rate (%) ↑		
	Clean	Randomized	Overall
X-VLA Zheng et al. (2026)	72.8	41.2	57.0
$\pi_{0.5}$ Black et al. (2025a)	81.2	75.6	78.4
LingBot-VLA Wu et al. (2026a)	86.0	83.6	84.8
Qwen3.5-9B†	62.8	60.4	61.6
Qwen3.8-27B†	68.4	67.6	68.0
GLM-5.3-Flash†	87.6	86.4	87.0
GPT-5.6-Sol†	**92.0**	**90.4**	**91.2**
OmniBrain-0 (Ours)	88.8	85.6	87.2

Table 4: Results on R2R-CE Val-Unseen. Success Rate (SR), Success weighted by Path Length (SPL), and Oracle Success Rate (OSR) are reported as percentages; Navigation Error (NE) is reported in meters. † denotes zero-shot evaluation.

Method	SR ↑	SPL ↑	OSR ↑	NE ↓
Uni-NaVid Zhang et al. (2025)	47.0	42.7	53.3	5.58
StreamVLN Wei et al. (2026b)	56.9	51.9	64.2	4.98
DualVLN Wei et al. (2026a)	64.3	**58.5**	70.7	4.05
Qwen3.5-9B†	34.0	26.9	48.0	6.57
Qwen3.8-27B†	41.7	31.6	58.3	6.12
GLM-5.3-Flash†	58.2	35.5	73.6	4.31
GPT-5.6-Sol†	**71.0**	57.5	**75.0**	**3.11**
OmniBrain-0 (Ours)	65.0	54.8	71.9	3.78

RoboTwin 2.0. Table 3 summarizes the RoboTwin evaluation results across `Clean` and `Randomized` task configurations. Our OmniBrain-0 achieves a success rate of 88.8% under the Clean setup and maintains 85.6% under environmental randomization, reaching an overall score of 87.2%. This performance surpasses strong baselines such as LingBot-VLA and GLM-5.3-Flash, while maintaining a compact model size and high execution efficiency.

R2R-CE. Table 4 summarizes the vision-and-language navigation (VLN) performance on the R2R-CE Val-Unseen benchmark, reporting Success Rate (SR), Success weighted by Path Length (SPL), Oracle Success Rate (OSR), and Navigation Error (NE). Among zero-shot planners, GPT-5.6-Sol achieves the highest SR (71.0%) and OSR (75.0%) and the lowest NE (3.11 m), reflecting the performance ceiling of strong general-purpose models on this task. Compared with the zero-shot Qwen3.5-9B sharing the same backbone, OmniBrain-0 improves SR from 34.0% to 65.0% and SPL from 26.9% to 54.8%, while reducing NE from 6.57 m to 3.78 m. It also outperforms zero-shot Qwen3.8-27B, a larger model, and zero-shot GLM-5.3-Flash on SR, SPL, and NE. Compared with specialist VLN models, OmniBrain-0 achieves a higher SR than all of them, slightly exceeding the strongest one, DualVLN (65.0% vs. 64.3%), and attains a lower NE (3.78 m vs. 4.05 m), although its

SPL remains lower (54.8% vs. 58.5%). These results indicate that distillation effectively transfers the navigation decision-making capability of strong models into a compact model.

Real-world manipulation. Figure 3 summarizes results on three real-world manipulation tasks: block stacking, apple placement, and drawer opening. OmniBrain-0 achieves a macro-average success rate of 73.3%, outperforming $\pi_{0.5}$, which achieves 58.3% across the three tasks. The improvement is primarily driven by block stacking, where success increases from 50% to 75%, followed by apple placement (40% to 60%), while performance on drawer opening is matched at 85%. GPT-5.6-Sol (zero-shot) achieves a higher macro-average of 86.7%, with the largest gaps to OmniBrain-0 on block stacking (90% vs. 75%) and apple placement (75% vs. 60%). These results support the applicability of the compact unified agent to the evaluated physical manipulation tasks, although further improvements are needed to match the teacher's task-level performance.

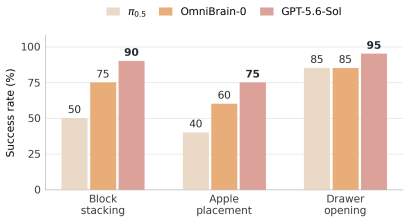

Figure 3: **Real-world manipulation performance.** Comparison of the standalone baseline $\pi_{0.5}$, the compact unified agent OmniBrain-0, and GPT-5.6-Sol (zero-shot). Success rates are reported as percentages.

4.3 Visualization

Figure 4 visualizes OmniBrain-0 on a manipulation task and a navigation task. In the manipulation task, the first grasp fails and the robot reaches the stove without the moka pot (stages 2–3). Instead of blindly retrying, the agent resets, turns on the stove first, and then re-grasps the pot and places it on the stove (stages 4–8), demonstrating feedback-driven failure recovery. In the navigation task, OmniBrain-0 decomposes a multi-sentence instruction into successive sub-goals, from climbing the stairs to stopping in front of the bathroom mirror. More detailed visualizations are provided in Appendix H.

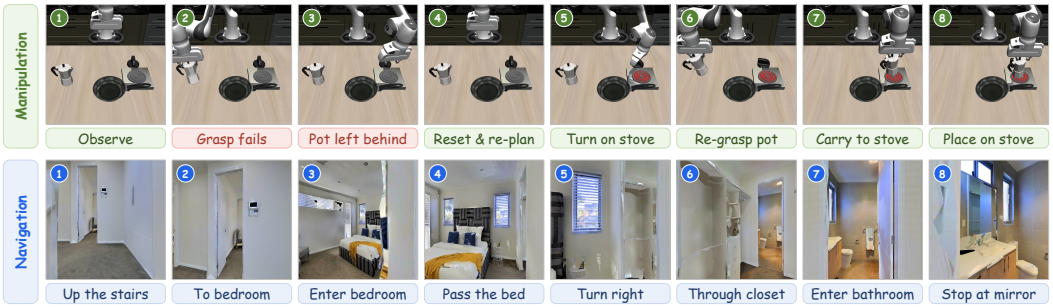

Figure 4: **Examples of OmniBrain-0 on Manipulation and Navigation.** Top: LIBERO-Pro L10-S; red labels mark failed attempts. Bottom: R2R-CE Val-Unseen.

5 Conclusion

We presented **OmniBrain-0**, a lightweight unified embodied agent that coordinates manipulation and navigation through a shared high-level decision model. By combining embodied pretraining, domain-specialized agent distillation, and MOPD-based expert consolidation, OmniBrain-0 transfers interaction experience from powerful agents into a single compact planner while preserving the specialization of low-level skills. Experiments across manipulation and navigation benchmarks demonstrate consistent improvements over the compact zero-shot backbone and strong cross-domain task performance. These results suggest a practical path toward learning unified agentic capabilities for long-horizon embodied decision-making. Future work will further scale task diversity, improve cross-domain generalization, and extend evaluation to broader real-world settings.

AI USE STATEMENT

Generative AI is part of the proposed method: GPT-5.6-sol serves as the teacher agent that generates the interaction trajectories used for training (Section 3.3.1). In addition, we used generative AI tools to assist with code implementation and translation between Chinese and English. Formulating mathematical claims or writing proofs is not applicable to this work. We also used generative AI tools to assist with paper writing, including polishing the language, editing the text, and formatting tables and the manuscript layout. We have reviewed all AI-assisted work: AI-assisted code was tested and verified by the authors, and all AI-edited text was checked by the authors for accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025a.

William Chen, Suneel Belkhale, Suvir Mirchandani, Karl Pertsch, Danny Driess, Oier Mees, and Sergey Levine. Training strategies for efficient embodied reasoning. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 365–391. PMLR, 27–30 Sep 2025b. URL https://proceedings.mlr.press/v305/chen25a.html.

Liane Galanti, Dhruv Shah, and Tri Dao. Addressing the orchestration gap in generalist robots via physical agency. *arXiv preprint arXiv:2607.21725*, 2026.

Jinbang Huang, Yuanzhao Hu, Zhiyuan Li, Ran Qi, Yixin Xiao, Zhanguang Zhang, Mark Coates, Tongtong Cao, and Yingxue Zhang. Roboharness: Memory-driven orchestration of heterogeneous robot policies for long-horizon planning. *arXiv preprint arXiv:2607.18060*, 2026.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. Distilling llm agent into small models with retrieval and code tools, 2025. URL https://arxiv.org/abs/2505.17612.

Dongyoung Kim, Huiwon Jang, Myungkyu Koo, Suhyeok Jang, Taeyoung Kim, et al. Rldx-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the navgraph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Hao Li, Ziqin Wang, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, and Jiangmiao Pang. Robointer: A holistic intermediate representation suite towards robotic manipulation. *arXiv preprint arXiv:2602.09973*, 2025.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations*, volume 2026, pp. 98643–98667, 2026.

NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Meng Wei, Chenyang Wan, Peng Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 12380–12396, 2026a.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. Streamvln: Streaming vision-and-language navigation via slowfast context modeling, 2026b. URL https://arxiv.org/abs/2507.05240.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026a.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026b.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.013.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, et al. Harness vla: Steering frozen vlas into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026.

Zhonghan Zhao, Ke Ma, Wenhao Chai, Xuan Wang, Kewei Chen, Dongxu Guo, Yanting Zhang, Hongwei Wang, and Gaoang Wang. Do we really need a complex agent system? distill embodied agent into a single model. *arXiv preprint arXiv:2404.04619*, 2024.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pp. 260–278. Springer, 2024.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. Turnopd: Making on-policy distillation turn-aware for efficient long-horizon agent training. *arXiv preprint arXiv:2607.05804*, 2026.