Resilience Matters for Embodied Agents System: New Metrics, Systematic Evaluation, and Optimization

Yapeng Liu[1,2,3] **Yuanzhao Zhai**[1,2] **Xudong Gong**[1,2]
Dawei Feng[1,2] **Bo Ding**[1] **Lin Wang**[3] **Huaimin Wang**[1,2]

[1]PDL Lab, College of Computer Science and Technology, National University of Defense Technology
[2]State Key Laboratory of Complex & Critical Software Environment, Changsha 410073, Hunan, China
[3]EmPACT Lab, Nanyang Technological University, Singapore

Code: `Eas-Resilience-Evaluation` | **Website:** `Eas-Resilience-Evaluation`

Abstract

Embodied Agents System (EAS) are increasingly deployed in open-world physical domains, where reliability directly dictates deployment quality and human-agent trust. However, existing evaluations rely on outcome-centric metrics as success rate or safety scores that collapse diverse execution trajectories into coarse scores, obscuring the dynamic processes underlying agent behavior. Therefore, they ignore a critical property of EAS – which we define as the **Resilience** – that reflects how EASs *recover*, *stabilize*, and *extend* under perturbations and across iterative updates. The lack of resilience is particularly critical in open-world environments due to continuous unexpected disruptions, thus directly affecting the quality of EAS deployment. To address this problem, we gain insight from the resilience-engineering concepts to EAS groundings and propose a novel resilience evaluation framework that can be flexibly applied to any EAS. Specifically, we define the first comprehensive resilience metrics suite for EASs system that exposes *Rebound*, *Stability*, and *Graceful Extensibility* across embodied tasks execution, providing a practical grounding for EAS resilience analysis. We further implement the resilience evaluation layer that transforms execution process into assessments for diagnosis and optimization. Across 400 household tasks with 10 EAS, we reveal the process-level distinction hidden by outcome metrics, including recovery cost differences among successful episodes ($\Delta C_{\text{rec}} = 25.2$), increased instability and task-family degradation. Metrics-guided optimizations reduce recovery cost and increase stability, graceful extensibility completion, showing the diagnostic effect of resilience evaluation. Our results reveal a trade-off among resilience characteristics, suggesting that a resilient EAS construction should be configured according to deployment-specific requirements.

1 Introduction

Embodied Agents System (EAS) shows great potential in domains where physical interaction and long-term task completion are essential, especially for the collaboration domain [18, 4]. Recent advances in large language models (LLMs) further strengthen EAS capabilities of reasoning and interaction, along with increased complexity in reliability evaluation [32, 30**?**]. As illustrated in Fig. 1, two EASs receive the same household instruction and achieve the same outcome, but their thought and execution process can be fundamentally different. This difference matters because EAS operates in dynamic environments where disruptions have continuously emerged [19, 8], indicating that reproducible reliability is critical for physical tasks and collaboration trust during execution [9, 36]. However, current EAS evaluation largely remains outcome or safety metrics, such as success rate or aggregated safety triggers [**?** 35, 16, 30]. They collapse different execution processes into the same score, making brittle success indistinguishable from trustworthy adaptation success [22, 23]. This

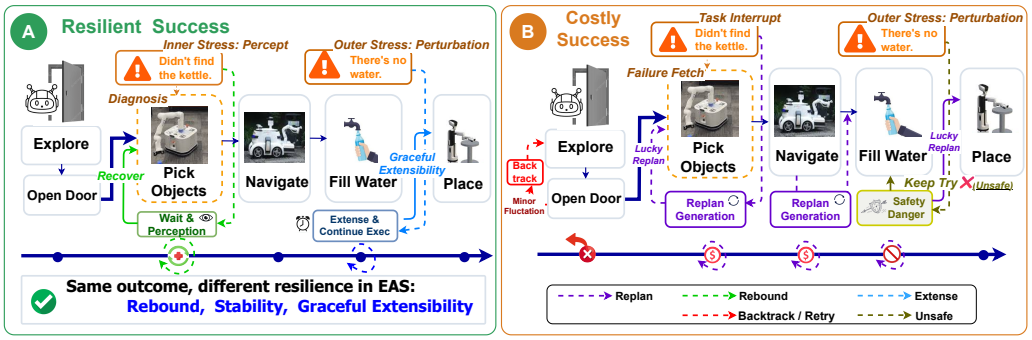

Figure 1: Outcome hide process resilience difference in EAS. (A) A resilient success trajectory handles inner and outer stress through resilient mechanisms, (B) A costly success relies on repeated backtrack, replan and unsafe behaviors.

limitation points to a missing property of EAS – which we define as the ***Resilience*** in EAS – that reflects how EASs *recover*, *stabilize*, and *extend* under stress.

To address this problem, we gain insights from the resilience-engineering concepts to EAS groundings. System resilience evaluation captures the intrinsic property of a system, which maintains its acceptable execution and effective recovery when unexpected disturbances happen [11, 3]. Woods further decomposes system resilience into four capabilities, including rebound, robustness, graceful extensibility under stress, and sustained adaptability [33]. Specifically, we define the first comprehensive **Resilience Metrics** for EAS that expose *Rebound*, *Stability*, and *Graceful Extensibility* throughout execution: recovery cost C_{rec}, stability sensitivity β, stress response mapping $M_f(\lambda)$ and stress capacityλ_f^*. We further implement the **Resilience Evaluation Layer** that extracts signals from trajectories, monitors, and LLM judge, aggregates them through stage baselines and controlled stress intensities. This layer non-intrusively enables comparison, diagnosis and optimization guidance for EAS resilience evaluation.

We conduct experiments on Habitat-Sim with 400 household tasks and 10 representative EAS methods. Focus on EAS resilience Metrics Validity, Benchmark and Practical Optimization, results show that resilience evaluation reveals process-level disparities, benchmarking EAS methods and has practical diagnosis effectiveness. Among success episodes, physical perturbations increase recovery cost from 11.6 to 36.3, semantic perturbations raise instability to $\Delta\beta = 0.32$, and stress tests expose degradation, recovery cost and perturbations intensities are strongly correlated (Spearman $\rho = 0.82$). For methods, Reflexion and CoPAL achieve comparable SR (56.3% vs. 55.8%) but differ by $4.8\times$ in recovery cost, showing that EASs have distinct resilience signatures. There is usually a trade-off in resilience aspects for EAS on our bench. Metrics-guided optimization further improve resilience performance, demonstrating that our evaluation layer is both diagnostic and actionable.

In summary, our work makes three major contributions.

1. **Embodied Resilience Metrics and Evaluation Layer:** We formalize resilience metrics as three aspects: *Rebound*, *Stability*, and *Graceful Extensibility*. Then we complement EAS evaluation with the resilience evaluation layer, a practical grounding for measuring EAS resilience property quantification.
2. **Evidence and benchmarking:** We perform a experimental study on 10 representative EAS methods, demonstrating that our metrics successfully distinguish between brittle and resilient behaviors (Sec 3.2.2), we benchmark them, and show their resilience signatures under our resilience evaluation (Sec 3.2.3).
3. **Resilience evaluation provides practical optimization:** We provide quantitative evidence linking specific diagnosis to resilience improvements (Sec 4), showing that resilience-guided diagnosis can support practical optimization, improving EAS resilience while holding the resilience trade-off design principle.

2 Related Works

LLM-Enhanced EAS Evaluation: From Outcome to Resilience Metrics. LLM-enhanced EAS close the perception-execution loop by taking language-based reasoning into embodied actions generation, and they evolved from instruction grounding to feedback reasoning and self-correction [2, 17, 13, 28, 29, 10]. Recent works [14, 20, 38, 31] further explore adaptation, consistency, and open-ended skill acquisition to build trustworthy EAS. However, classical EAS evaluation remains

largely outcome-centric as PARTNR emphasizing success metrics [36, 39, 5, 24]. Under embodied scenarios – physical irreversibility, partial observability, and dynamic constraints, such outcomes cannot distinguish robust adaptation from brittle success [18**?** , 30].

Focused on EAS reliability evaluation, recent works construct perturbation test to measure safety-aware planning [35, 34, 6]. They provide reliability diagnostics, but mainly whether an agent violates constraints or approaches a hazardous state in perturbation tests rather than normal process-level execution [16, 40]. We present the resilience metrics to evaluate *EAS Resilience* as an intrinsic property of EAS, measuring how the system behaves from inner stress or perturbations. In this way, we complement EAS evaluation with our resilience evaluation layer to measure how the resilience of EAS exhibits and provide practical optimization guidance.

Resilience Concepts Practical Evaluation in EAS Execution. In systems engineering, resilience emphasizes the capability to endure, adapt to, and recover from disruptions [25, 7]. Woods organizes the technical expression of resilience into four concepts: rebound, robustness, graceful extensibility and sustained adaptability [33]. This perspective holds that handling open-world complexities relies not only on failure prevention, but on continuous, dynamic adaptation during execution [12]. Classical resilience evaluation works evaluate loss severity and recovery latency, but their reliance on exact state matching or crash detection proves insufficient for embodied scenarios [15, 27, 24]. Recently, resilience-relevant evaluating methods for EAS emerged, they evaluate on semantic safety under hazardous instructions and physical cases [41, 37, 19]. However, these methods mix EAS resilience with adversarial security or perturbation stability. They evaluate the failure under static perturbation, but rarely capture the process-aware dynamics like recovery or extensibility.

Grounding with resilient EAS construction, we implement the resilience concepts into three resilient aspects. *Rebound* captures the burden of closed-loop recovery; *Stability* captures robustness as decision consistency; and *Graceful Extensibility* captures bounded degradation and remaining operational margin under stress. Sustained adaptability is treated as a longitudinal extension because it is less directly observable within dynamic EAS execution. Our resilient evaluation realize the practical evaluation layer, providing the validated instrument that reveals the EAS resilience property.

3 The Proposed EAS Resilience Evaluation Framework

Overview. We formalize the Resilience **Metrics** and Resilience **Evaluation Layer** for EAS, which goal is to evaluate system resilience from execution dynamics. Given an episode artifact $(\tau, \mathcal{L}, \mathcal{C}, \mathcal{J})$, where $\tau = \{(o_t, a_t, r_t)\}_{t=1}^{T}$ is the execution trajectory, $\mathcal{L}$ is the runtime log, $\mathcal{C}$ is the monitor, and $\mathcal{J}$ is the LLM Judge signal, the evaluation layer maps execution dynamics to three resilience aspects:

$$\mathcal{M} : (\tau, \mathcal{L}, \mathcal{C}, \mathcal{J}) \mapsto (\mathcal{M}_{\text{rebound}}, \mathcal{M}_{\text{stability}}, \mathcal{M}_{\text{GE}}).$$

For episode i, let t denote a runtime step. At each step, we record task progress $p_t \in [0, 1]$, proposition completion $q_t \in [0, 1]$, and execution mode m_t. We define a local execution anchor as $a_t := (f, b_t^p, b_t^q, m_t)$, where f is the task family, and b_t^q is the residual proposition load. For each anchor, we estimate a local **Stage Baseline** from clean or reference episodes under the same a_t:

$$\mathcal{N}(a_t) := \big(\tau^*(a_t), W^*_{\text{rem}}(a_t), \bar{T}^{\text{cog}}(a_t), \bar{N}^{\text{phy}}(a_t), \bar{\Delta p}(a_t), \bar{\Delta q}(a_t), \bar{r}(a_t)\big).$$

Here, $\tau^*(a_t)$ is the nominal cycle time, $W^*_{\text{rem}}(a_t)$ is the estimated remaining work from the anchor, $\bar{T}^{\text{cog}}(a_t)$ and $\bar{N}^{\text{phy}}(a_t)$ summarize cognitive and physical effort, $(\bar{\Delta p}(a_t), \bar{\Delta q}(a_t))$ summarize task progress and proposition completion, $\bar{r}(a_t)$ summarizes nominal risk load. The Stage Baseline provides a common local reference for resilience metrics computation and the reliability of statistical generalization on resilience metrics. In parallel, we take LLM-as-a-Judge to densify sparse execution feedback. The LLM judges our process from instruction, state, trajectory, and env feedback, and then produces structured reward for goal progress, rational action, and efficiency; implementation details are available through our Code Repository and Project Website. Based on this shared evidence substrate, we instantiate system resilience from three complementary aspects: *Rebound*, *Stability*, and *Graceful Extensibility* as Table 1.

3.1 Metrics Design

1) Rebound Metrics: Recovery Cost to Availability. Rebound measures how much extra work the agent must expend to return the acceptable execution after disruption. We use the **Recovery Cost** as the primary indicator, revealing the efficient rebound and brute-force rebound. Detailed implementation is available through our Code Repository and Project Website. For a rebound window

Table 1: Overview of resilience metrics with aspect, type (**D**: deterministic outcome, **S**: statistical, **A**: analysis, **T**: time-series), and data source

Sym.	Metric Name	Calculation	Aspect	Type	Data Source
C_{rec}	Recovery Cost	$C_{\text{rec}} = \frac{1}{W^*_{\text{rem}}(a_{t_d})} \sum_{t_d}^{t_r} \frac{\Delta\tau_t}{\tau^*(a_t)} \cdot g_t^{\text{rec}}$	Rebound	D, S	System Logs
β	Policy Stability	$\beta = \sup_{z,\rho} \frac{\lvert\ell(\pi,\rho(z)) - \ell(\pi,z)\rvert}{\lVert\rho\rVert}$	Stability	S	Sensitivity Test
$M_f(\lambda)$	Stress Curve Mapping	$M_f(\lambda) = \frac{1}{K} \sum_{j=1}^{K} \tilde{r}_{(j),f}(\lambda) - 1$	Extensibility	D, S	Stress Test
λ_f^*	Stress Capacity	$\lambda_f^* = \sup\{\lambda \in [0,1] \mid M_f(\lambda) \geq 0\}$	Extensibility	S, A	Stress Grid

$[t_d, t_r]$, t_d is the first step at which execution departs from the nominal continuation under the local anchor a_t, and t_r is the first step at which the task re-enters the acceptable execution. Execution *StageBaseline* $\mathcal{N}(a_t)$ (execution local reference, refer to Sec 3.2) provides the nominal local reference, including the remaining nominal work $W^*_{\text{rem}}(a_t)$, and the expected local execution statistics.

We decompose recovery cost into three aligned parts: **Cognitive recovery**, **Physical recovery**, and **State debt** in Eq. 1. We aggregate their excess variables as

$$\mathbf{z}_t^{\text{cog}} = \frac{(T_t^{\text{cog}} - \bar{T}^{\text{cog}}(a_t))_+}{\bar{T}^{\text{cog}}(a_t) + \epsilon}, \quad \mathbf{z}_t^{\text{phy}} = \frac{(N_t^{\text{phy}} - \bar{N}^{\text{phy}}(a_t))_+}{\bar{N}^{\text{phy}}(a_t) + \epsilon}, \quad \mathbf{z}_t^{\text{deb}} = \frac{(r_t - \bar{r}(a_t))_+}{\bar{r}(a_t) + \epsilon}. \tag{1}$$

where $\mathbf{z}_t^{\text{cog}}$ measures excess planning, perception effort; $\mathbf{z}_t^{\text{phy}}$ measures excess navigation, retry effort; and $\mathbf{z}_t^{\text{deb}}$ measures lagged progress leakage, risk load. $(\cdot)_+$ indicates the ReLU function, we record the excess costs while discarding pre-complete [1]. Implementation details and illustrative calculations are available through our Code Repository and Project Website. To make the three channels comparable under the same anchor-conditioned regime, we convert each local excess vector into a covariance-adjusted deviation as Mahalanobis Distance[21]:

$$g_t^u = \sqrt{\mathbf{z}_t^{u\top} \big(\Sigma_u(a_t) + \eta I\big)^{-1} \mathbf{z}_t^u}, \qquad u \in \{\text{cog}, \text{phy}, \text{deb}\}, \tag{2}$$

where $\Sigma_u(a_t)$ is the covariance estimated from standard stage baselines, ηI is the regularizer. This step removes scale inconsistency and accounts for correlation within each recovery channel. We then aggregate the three channel deviations into recovery intensity $g_t^{\text{rec}} = \sqrt{\frac{1}{U} \sum_u^U (g_t^u)^2}$. Finally, for episode i, the *Rebound Cost* is defined as

$$C_{\text{rec}}^{(i)} = \frac{1}{W^*_{\text{rem}}(a_{t_d}) + \epsilon} \sum_{t=t_d}^{t_r} \frac{\Delta\tau_t}{\tau^*(a_t) + \epsilon} \cdot g_t^{\text{rec}}. \tag{3}$$

This quantity measures the total extra work required to restore acceptable execution after disruption. Normalization by $W^*_{\text{rem}}(a_{t_d})$ ensures that recovery is judged against the remaining nominal work at the disturbance point. We aggregate rebound signatures using the statistical procedures documented in our Code Repository and Project Website.

2) Stability Metrics: Execution Fluctuations. Stability quantifies the system's ability to remain *consistent* under stresses, whether arising from stochasticity (Execution), perturbations (Sensitivity), or self-evolution. It is reflected in execution as frequent replanning, unstable progress, and abrupt shifts in the progress. We take *Policy Stability*(β) as the primary indicator, because it directly measures whether small changes in task instruction or execution condition amplify into process fluctuations. We describe how β is computed from execution representations; implementation details, including the value-network design, are available through our Code Repository and Project Website.

We use the value function $V(\cdot)$ for each EAS execution representation, which is used to diagnose execution progress changes trend. Given the encoded state s_t, the value function is

$$V(s_t) = \mathbb{E}_\pi \left[\sum_{u=t}^{T} \gamma^{u-t} \tilde{r}_u \mid s_t \right], \tag{4}$$

where π is the agent policy and $\tilde{r}_u$ is the shaped reward by LLM judge and W^*_{rem}. For each state transition, we compute the one-step temporal-difference residual $\delta_t = \tilde{r}_t + \gamma(1-d_t)V(s_{t+1}) - V(s_t)$, where d_t is the termination signal. We further compute the generalized advantage estimate A_t as an exponentially decayed accumulation of TD residuals: $A_t = \sum_{\ell=0}^{T-t} (\gamma\lambda)^\ell \delta_{t+\ell}$. Then our resilience

loss for shape reward update is

$$\ell(\pi, z_i) = \frac{1}{T_i} \sum_{t=1}^{T_i} \sqrt{\mathbf{x}_t^\top \left(\Sigma(a_t) + \eta I\right)^{-1} \mathbf{x}_t}, \tag{5}$$

where z_i denotes an episode, $\Sigma(a_t)$ is the standard stage baselines covariance, and ηI is regularizer. This loss increases when the trajectory $\mathbf{x}_t$ contains large value variance, TD spikes, and advantages. We define β-*Stability* as this perturbations sensitivity to execution progress:

$$\beta_f = \sup_{z \in \mathcal{Z}_f, \rho \in \mathcal{P}_f} \frac{|\ell(\pi, \rho(z)) - \ell(\pi, z)|}{\|\rho\| + \epsilon}. \tag{6}$$

Here, $\mathcal{Z}_f$ is the episode set from task family f, $\mathcal{P}_f$ is the controlled perturbations set. A low β_f means that similar conditions lead to similar results, while high β_f means more replanning, value fluctuation, and progress instability.

3) Graceful Extensibility Metrics: Stress Response. Graceful Extensibility measures how an EAS performance extends as stress increases. It ensures that when stressors exceed the system's adaptive capacity, performance declines *gracefully* and predictably rather than collapsing catastrophically. This is a dynamic stress-response, revealing the resilience property of EAS. We therefore model Graceful Extensibility as a stress-response mapping $\lambda \mapsto M_f(\lambda)$, where λ is the stress severity and $M_f(\lambda)$ is the operational margin.

Let $\mathcal{T}_\lambda$ denote a stress operator with severity $\lambda \in [0, 1]$, which is related to perturbations and EAS self-evolvement. In our resilience evaluation, we unify external perturbation and internal evolution through the same stress-response formulation, as documented in our Code Repository and Project Website. For an episode i from task family f under stress $\mathcal{T}_\lambda$, we first define a ***hard*** boundary monitor:

$$m_i(\lambda) = \min \left\{ \frac{p_{\mathcal{T},i}(\lambda)}{\tau_p}, \frac{q_{\mathcal{T},i}(\lambda)}{\tau_q}, \frac{\tau_{\text{rec}}}{C_{\text{rec},i}(\lambda) + \epsilon}, \frac{\tau_{\text{stab}}}{\beta_{\text{stab},i}(\lambda) + \epsilon}, \frac{\tau_{\text{safe}}}{U_i(\lambda) + \epsilon} \right\} - 1. \tag{7}$$

where U_i is the constraints violation load. The thresholds define the minimal acceptable task contract, $m_i(\lambda) < 0$ means that the contract is violated.

After accumulation of stage baselines from the task family f, we aggregate the statistical extensibility of the operation. For compact notation, let $Q_\alpha^f[X](\lambda) \triangleq Q_\alpha(X(\lambda) \mid f)$, where $Q_\alpha^f[X](\lambda)$ denotes the α-quantile of variable X over episodes from family f under stress λ. We construct the statistical extensibility vector

$$\mathbf{r}_f(\lambda) = \left[\frac{Q_\alpha^f[p_T](\lambda)}{\tau_p}, \frac{Q_\alpha^f[q_T](\lambda)}{\tau_q}, \frac{\tau_{\text{rec}}}{Q_{1-\alpha}^f[C_{\text{rec}}](\lambda) + \epsilon}, \frac{\tau_{\text{stab}}}{Q_{1-\alpha}^f[S_{\text{stab}}](\lambda) + \epsilon}, \frac{\tau_{\text{safe}}}{Q_{1-\alpha}^f[U](\lambda) + \epsilon} \right]^\top. \tag{8}$$

We take lower and upper quantiles for benefit variables and cost variables, which makes our metrics sensitive to graceful extensibility. Let $r_{(1),f}(\lambda) \le \cdots \le r_{(K_{\max}),f}(\lambda)$ be the sorted entries of $\mathbf{r}_f(\lambda)$. The statistical operational margin is defined as

$$M_f(\lambda) = \frac{1}{K} \sum_{j=1}^{K} r_{(j),f}(\lambda) - 1, \qquad 1 < K \le K_{\max}. \tag{9}$$

This bottom-K aggregation preserves the semantics of extensibility, but avoids the rigid behavior of a minimum optimization. For practical diagnosis, dynamic ***mapping*** between $M_f(\lambda)$ extensibility and execution performance is an important indicator to analyze EAS resilience. Further, the quantitative Graceful Extensibility indicator is the largest stress level where the EAS remains acceptable execution:

$$\lambda_f^* = \sup \{\lambda \in [0, 1] \mid M_f(\lambda) \ge 0\}. \tag{10}$$

We call λ_f^* the *stress capacity*. It gives the operational boundary of task family f: the maximum stress severity at which the system can still preserve its positive task progress in task family f.

3.2 Resilience Evaluation Benchmark

The Resilience Evaluation Layer Construction We construct the Resilience Evaluation Layer as a non-intrusive evaluation instrument that complements the EAS reliability evaluation process. We present our resilience evaluation pipeline in Alg 1, it attaches probes during the EAS execution, then computes collected signals to resilience metrics. This design allows resilience to be evaluated as an execution property rather than an additional task objective.

Algorithm 1 Resilience Metrics Evaluation Pipeline

Require: Task family set $\mathcal{F}$; episode artifacts $\{(\tau_i, \mathcal{L}_i, \mathcal{C}_i, \mathcal{J}_i)\}_{i=1}^{N}$; StageBaseline estimation; perturbation set $\mathcal{P}_f$; stress grid $\Lambda = \{\lambda_j\}_{j=1}^{m}$; fixed thresholds and windows.
Ensure: $\mathcal{M}_{\text{rebound}}, \mathcal{M}_{\text{stability}}, \mathcal{M}_{\text{GE}}$

Estimate Stage Baselines $\mathcal{N}(a)$ from executions.
for each task family $f \in \mathcal{F}$ **do**
  for each stress level $\lambda \in \Lambda$ **do**
    for each episode i, artifact $(\tau_i, \mathcal{L}_i, \mathcal{C}_i, \mathcal{J}_i)$ **do**
      Extract stepwise signals: progress p_t, q_t, execution mode m_t, execution $\mathcal{L}_i$, critics $\mathcal{C}_i, \mathcal{J}_i$.
      Assign local execution anchors $a_t = (f, b_t^p, b_t^q, m_t)$.
      if Detect **Rebound** window $[t_d, t_r]$ **then**
        Compute $\mathbf{z}_t^{\text{cog}}$, $\mathbf{z}_t^{\text{phy}}$ and $\mathbf{z}_t^{\text{deb}}$.
        Compute rebound cost $C_{\text{rec}}^{(i)}$ from g_t^{rec}.
        Return to acceptable $\mathcal{N}(a_t)$.
      end if
      Collect **Stability** values $V(s_t)$, TD residuals δ_t, and GAE advantages A_t.
      Aggregate to vector $\mathbf{x}_t$ and β_i.
      Backpropagation Critic $\ell_{\text{stab}}(\pi, z_i)$ and Stage Baselines $\mathcal{N}_f(a)$.
    end for
    Retrieve stressed executions under $\mathcal{T}_\lambda$.
    Compute hard boundary monitors $m_i(\lambda)$ in family f.
    Aggregate episode statistics $\mathbf{r}_f(\lambda)$.
    Compute stress mapping: $M_f(\lambda) \leftarrow \frac{1}{K}\sum_{j=1}^{K} r_{(j),f}(\lambda) - 1$.
  end for
  Compute Graceful Extensibility stress capacity: $\lambda_f^* \leftarrow \sup\{\lambda \in \Lambda \mid M_f(\lambda) \geq 0\}$.
  Update Stage Baseline $\mathcal{N}$ and Statistical Aggregation Metrics
end for
return C_{rec}, β, M_f, λ_f^*, trajectory.

3.2.1 Resilience EAS Testbed Setup

We conduct experiments on *Habitat-Sim 3.0* [26] with PARTNR-style [**?**] collaborative household tasks, including navigation, object manipulation, and object transportation scenarios. To expose different EAS resilience, we introduce controlled perturbations over physical states, semantic instructions, object states, and stress severity. The full perturbation specification, evaluated EAS methods, and dataset setting are documented in our Code Repository and Project Website.

3.2.2 Resilience Metrics Validation: Execution Differences Hidden by Outcomes

We first validate whether resilience metrics ($\mathcal{M}_{\text{rebound}}, \mathcal{M}_{\text{stability}}, \mathcal{M}_{\text{GE}}$) capture distinct resilience property hidden by outcome-centric evaluation, and verify them as the measurement instruments.

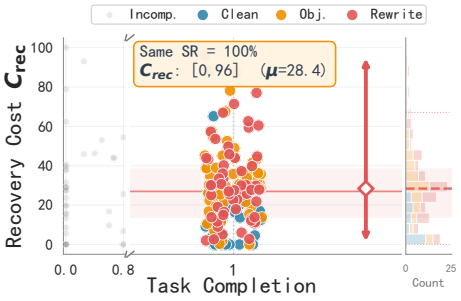

Figure 2: Different C_{rec} under outcomes.

Resilience metrics disperse differences in same outcome. Resilience metrics reveal significant performance differences that both SR and safety scores fail to capture, demonstrating discrimination capability for EAS with similar success rates and safety scores. We take C_{rec} as an example, Fig. 2 illustrates "the same outcome, different resilience" effect.

Resilience Metrics Construct Validity and Sensitivity. Fig. 3 further confirms the construct validity of the three metric families. Rebound C_{rec} separates executions by recovery burden: perturbations increase recovery cost from 11.6 in clean episodes to 36.3 in average, with a significant distributional shift (Mann–Whitney $p < 0.001$). Stability β captures sensitivity to semantic-preserving instruction changes: semantic perturbations inflate instability to $\beta = 0.32$, mainly driven by divergent action generation instability ($\beta_{\text{out}} = 0.42$). Graceful Extensibility characterizes degradation as a stress-response curve rather than a binary failure event: as stress severity increases over $\lambda \in [0.2, 1.0]$, the relative margin ΔM decreases while recovery cost rises, with strong correlation (Spearman $\rho = 0.82$). These results show that the metrics measure complementary resilience properties and are not reducible to SR, safety, or completion alone.

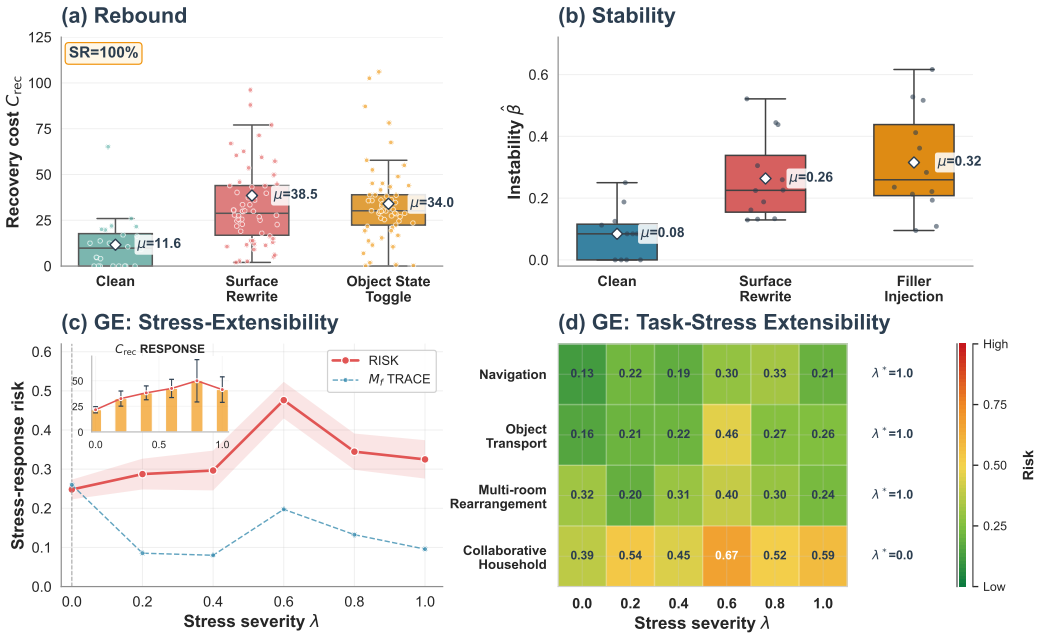

Figure 3: Resilience metrics construct validity results (*zoom in for better view*).

In addition, our Code Repository and Project Website provide analyses of the statistical reliability and sensitivity of the resilience metrics. These analyses indicate that resilience metrics are related to EAS inherent property, which maintains the consistency across embodied tasks execution.

Also, a scalability sensitivity analysis across Qwen3-8B-Instruct, Qwen2.5-7B-Instruct, and Llama3.1-8B-Instruct shows that stronger backbones improve baseline task performance while preserving similar resilience profile patterns, suggesting that resilience is a system-level property rather than only a backbone-size effect (see our Code Repository and Project Website).

3.2.3 Benchmarking EAS Resilience Results

We apply the validated resilience metrics to benchmark representative EAS methods under identical environments and perturbation settings. Table 2 reports the resilience benchmark scores, and Fig. 4 summarizes the normalized resilience profiles. For example, Reflexion and CoPAL achieve comparable SR (56.3% vs. 55.8%), but Reflexion incurs $4.8\times$ higher recovery cost than CoPAL (57.3 vs. 11.9). This indicates that EAS take different recovery processes, and resilience evaluation reveals these features.

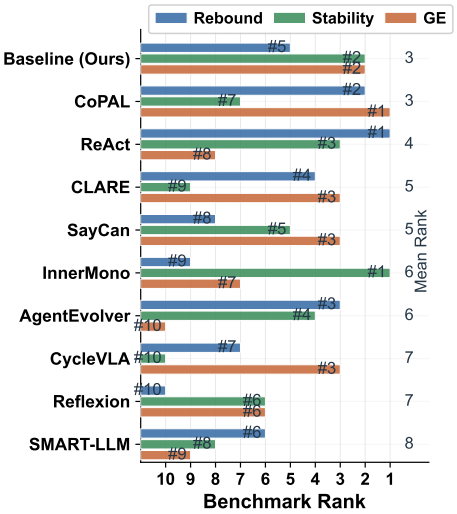

Figure 4: Resilience benchmark landscape.

This benchmark also shows that no current method dominates all resilience aspects. CoPAL achieves a favorable rebound–extensibility trade-off, with low recovery cost ($C_{\text{rec}} = 11.9$) and the highest stress capacity ($\lambda^* = 0.85$), but it requires long simulated execution steps. InnerMono has the best stability profile ($\beta = 0.149$), yet its recovery cost remains high ($C_{\text{rec}} = 43.5$). AgentEvolver has relatively low recovery cost ($C_{\text{rec}} = 19.2$), yet its stress capacity is the weakest among all methods ($\lambda^* = 0.21$), showing limited graceful extensibility. These results demonstrate that resilience should be interpreted as a multi-dimensional execution profile rather than a single scalar ranking.

EAS Methods Resilience Signatures Under the same benchmark protocol, we analyze distinct behaviors (performance, simulation steps, rebound, stability, and graceful extensibility) among the methods in Fig. 5. The shape of the radar plot serves as a visual fingerprint for the EAS resilience profile, and exhibits different resilience profiles, strength, and weakness. Some methods continue

Table 2: Comparative EAS resilience benchmark results in resilience aspects.

Method	Resilience Metrics			Outcome Ref.		Sim.
	C_{rec} (↓)	β (↓)	Stress Cap. λ^* (↑)	SR	Comp.	Steps
ReAct	6.5 ± 0.46	0.239 ± 0.011	0.50 ± 0.042	$38.5\% \pm 3.1\%$	$57.8\% \pm 4.6\%$	3374 ± 285
Reflexion	57.3 ± 5.07	0.299 ± 0.021	0.66 ± 0.035	$56.3\% \pm 4.2\%$	$75.0\% \pm 3.8\%$	4390 ± 312
CycleVLA	35.5 ± 2.91	0.400 ± 0.026	0.72 ± 0.047	$22.2\% \pm 2.8\%$	$47.5\% \pm 4.1\%$	4688 ± 420
CLARE	21.6 ± 1.68	0.344 ± 0.016	0.72 ± 0.052	$28.4\% \pm 3.5\%$	$46.0\% \pm 4.5\%$	4249 ± 360
SayCan	38.5 ± 2.49	0.283 ± 0.013	0.78 ± 0.046	$57.0\% \pm 4.1\%$	$72.6\% \pm 3.7\%$	4682 ± 380
AgentEvolver	19.2 ± 1.24	0.264 ± 0.012	0.21 ± 0.015	$48.8\% \pm 3.9\%$	$66.8\% \pm 4.2\%$	3084 ± 250
InnerMono	43.5 ± 2.69	0.149 ± 0.007	0.53 ± 0.026	$49.5\% \pm 4.0\%$	$67.5\% \pm 4.3\%$	3670 ± 290
CoPAL	11.9 ± 1.02	0.302 ± 0.017	0.85 ± 0.078	$55.8\% \pm 4.5\%$	$72.6\% \pm 3.9\%$	4846 ± 410
SMART-LLM	28.2 ± 2.20	0.339 ± 0.018	0.40 ± 0.036	$55.7\% \pm 3.8\%$	$74.3\% \pm 4.1\%$	5085 ± 395
Baseline	25.0 ± 2.03	0.200 ± 0.009	0.79 ± 0.042	$53.0\% \pm 3.6\%$	$69.7\% \pm 3.5\%$	3671 ± 275

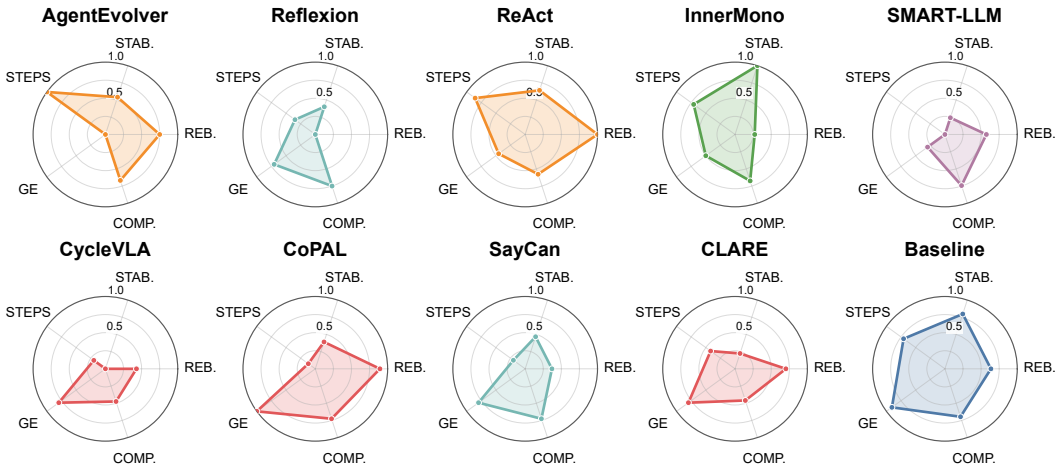

Figure 5: The Resilience Diagnosis. Visualizing the distinct behavioral archetypes of EASs.

execution through costly recovery, some maintain stable execution, some fluctuate during planning, and some show graceful extensibility under stress. These profiles provide a resilience overview of current EAS methods and serve as a diagnostic handbook for optimization experiments. We conclude resilience benchmark findings as below:

Resilience Evaluation Benchmark Findings

1) Outcome-equivalent executions can be resilience-inequivalent. Success rate and completion describe whether an EAS finishes the task, but they do not reveal how much recovery, replanning instability, or stress margin is consumed during execution. Resilience metrics expose these hidden process differences.
2) There is no current EAS method dominates all resilience aspects. Some methods recover efficiently, some remain stable under semantic variation, and some tolerate stress, but no method dominates all dimensions. This indicates that EAS deployment should be matched to the target failure regime: difficult tasks require stronger Graceful Extensibility, while sensitive tasks require stronger Stability.
3) Resilience metrics provide actionable diagnosis. Abnormal metric patterns can be traced back to concrete execution events, such as repeated recovery loops, action oscillation, or stress-boundary collapse. This makes the evaluation layer not only a benchmark tool, but also a diagnostic basis for the metric-guided optimizations in Sec. 4.

4 EAS Optimization: Metric-Guided Resilience Diagnostics

Based on the resilience metrics and the EAS resilience profiles, we introduce a unified *metric-guided resilience optimization* to the EAS execution. The goal is not to replace the original planner, but to use our resilience metrics as diagnostic signals that indicate where the execution loop should be minimally repaired. We implement three targeted optimizations for Rebound, Stability, and Graceful Extensibility, with implementation details provided in our Code Repository and Project Website. We compare the *Original* and *Optimized* variants under identical evaluation conditions. Table 3 shows that the resilience metrics are not only descriptive, but can also guide practical resilience improvements.

Table 3: Comparative results of metric-guided targeted improvements.

Target	Methods	Rebound $C_{\text{rec}} \downarrow$	Rec. Win. $\downarrow$	$\beta_{\text{step}} \downarrow$	$\beta \downarrow$	GE. Comp. $\uparrow$	Formal GE
Rebound	Baseline	59.11	3.13	0.2881	0.6834	0.887	–
	Optimized	**33.73** ↓42.94%	**2.38** ↓23.96%	0.2079 ↓27.84%	0.6870 ↑0.53%	0.819 ↓7.67%	–
Stability	Baseline	51.01	2.63	0.3185	0.6788	0.903	–
	Optimized	43.01 ↓15.68%	3.00 ↑14.07%	**0.2552** ↓19.87%	**0.6772** ↓0.24%	0.750 ↓16.94%	–
GE	Baseline	36.70	1.33	1.4594	0.6917	0.833	Incomplete
	Optimized	99.59 ↑171.36%	2.25 ↑69.17%	0.7215 ↓50.56%	0.6371 ↓7.89%	**0.917** ↑10.08%	Complete†

† Formal GE is reported only for GE stress-tests, since it requires a stress-response curve over λ grid.

What's more, targeted optimization of a single resilient aspect often leads to compromises in other resilience aspects (optimizing GE results in high recovery cost C_{rec}). This confirms our findings: resilience is a multi-dimensional structural balance, and our resilient evaluation layer could capture this trade-off.

Opt1 Recovery: Feedback-Guided Recovery Implementation High recovery cost C_{rec} and long recovery windows indicate that the agent often spends extra reasoning or physical actions after local failures. To address this failure mode, we add a feedback-guided recovery layer on top of the centralized LLM planner. The layer monitors World-Graph evidence, recent execution feedback, and task-progress stagnation, and diagnoses recoverable faults such as missing graph nodes, stale object locations, repeated tool failures, or invalid object references. Instead of overriding the planner with a separate recovery policy, we get *Rebound Guidance* Π_{t+1} into the next planning context, including perception refresh, reflection, state summarization, or belief rollback. Then we formalize it as Habitat skill tools for better use. This keeps the original action space unchanged while converting low-level execution feedback $\mathcal{F}_t$ into planner recovery hints. As shown in Table 3, this optimization reduces C_{rec} from 59.11 to 33.73 and the recovery window from 3.13 to 2.38, showing that the metric-guided feedback loop directly reduces redundant recovery effort.

Opt2 Stability: Consistency State Record High β values reveal unstable replanning behavior, where the agent adapts to new feedback but may oscillate between similar actions, repeat completed subtasks, or drift away from the current execution c_t. We therefore introduce a consistency record for replanning. At each step, it constructs a read-only phase guidance from the task goal, current phase, World-Graph state, current room and holding state, completed and pending subtasks, and recent action-response history. This design improves consistency without making the policy brittle. The optimized variant reduces β_{step} from 0.3185 to 0.2552 and improves β from 0.6788 to 0.6772, indicating that the stability metric captures and guides reductions in local replanning variance.

Opt3 GE: Boundary Long-tail Control for Difficulties Graceful Extensibility focuses on whether performance degrades smoothly as stress increases, rather than collapsing or entering unbounded long-tail behavior. In high-stress cases, we observed that the planner may repeatedly wait or receive near-duplicate execution feedback without making progress. Such behavior inflates execution cost and makes the stress-response curve insensitive to true adaptive capacity. To control this boundary effect, we add a long-tail monitor that detects dual-wait patterns, semantically repeated feedback, and stagnant task progress. When a boundary stall B_t is detected, the system injects a bounded cognitive reset that discourages further non-progressive waiting and requests either a concrete productive action or termination. This converts uncontrolled long-tail execution into measurable boundary execution.

5 Conclusion and Future Work

We presented the first Resilience Evaluation Framework for EASs. By designing novel metrics: *Rebound*, *Stability*, and *Graceful Extensibility*, our framework can expose resilience hidden pathologies, such as high-cost recovery and catastrophic forgetting, which remain invisible to outcome-centric metrics. Empirical analysis across representative baselines identified distinct Resilience Archetypes, confirming that architectural choices involve quantifiable trade-offs between execution, stability, and adaptability. The proposed optimization framework can help EASs developers to diagnose root causes and enforce non-degradation contracts to build a resilient and trustworthy EAS.

Future Work Future work can extend resilience evaluation along three directions: *Sim2Real* validation of whether simulation-derived resilience profiles transfer to physical robots, *theoretical analysis of trade-offs* among resilience aspects, and *deployment-aware optimization* that uses benchmarked resilience signatures to improve EASs for target application conditions. These directions can move resilience evaluation toward more principled, deployable, and balanced EAS design.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[2] 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.

[3] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33, 2004.

[4] Rajkumar Buyya et al. Agentic artificial intelligence (ai): Architectures, taxonomies, and evaluation of large language model agents. *arXiv preprint arXiv:2601.12560*, 2026.

[5] Matthew Cavorsi, Lorenzo Sabattini, and Stephanie Gil. Multirobot adversarial resilience using control barrier functions. *IEEE Transactions on Robotics*, 40:797–815, 2023.

[6] Zixing Chen, Yifeng Gao, Li Wang, Yunhan Zhao, Yi Liu, Jiayu Li, Xiang Zheng, Zuxuan Wu, Cong Wang, Xingjun Ma, et al. Hazardarena: Evaluating semantic safety in vision-language-action models. *arXiv preprint arXiv:2604.12447*, 2026.

[7] Richard I Cook and Beth Adele Long. Building and revising adaptive capacity sharing for technical incident response: A case of resilience engineering. *Applied ergonomics*, 90:103240, 2021.

[8] Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, et al. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*, 2025.

[9] Cuiyun Gao, Xing Hu, Shan Gao, Xin Xia, and Zhi Jin. The current challenges of software engineering in the era of large language models. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–30, 2025.

[10] Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12124–12131. IEEE, 2024.

[11] Crawford S Holling et al. Resilience and stability of ecological systems, 1973.

[12] Erik Hollnagel. *Safety-I and safety-II: the past and future of safety management*. CRC press, 2018.

[13] 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.

[14] Frank Joublin, Antonello Ceravola, Pavel Smirnov, Felix Ocker, Joerg Deigmoeller, Anna Belardinelli, Chao Wang, Stephan Hasler, Daniel Tanneberg, and Michael Gienger. Copal: corrective planning of robot actions with large language models. In *2024 ieee international conference on robotics and automation (ICRA)*, pages 8664–8670. IEEE, 2024.

[15] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[16] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.

[17] 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. *arXiv preprint arXiv:2209.07753*, 2022.

[18] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 2025.

[19] Xiaoya Lu, Zeren Chen, Xuhao Hu, Yijin Zhou, Weichen Zhang, Dongrui Liu, Lu Sheng, and Jing Shao. Is-bench: Evaluating interactive safety of vlm-driven embodied agents in daily household tasks. *arXiv preprint arXiv:2506.16402*, 2025.

[20] Chenyang Ma, Guangyu Yang, Kai Lu, Shitong Xu, Bill Byrne, Niki Trigoni, and Andrew Markham. Cyclevla: Proactive self-correcting vision-language-action models via subtask backtracking and minimum bayes risk decoding. *arXiv preprint arXiv:2601.02295*, 2026.

[21] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018.

[22] Viacheslav Moskalenko, Vyacheslav Kharchenko, Alona Moskalenko, and Borys Kuzikov. Resilience and resilient systems of artificial intelligence: taxonomy, models and methods. *Algorithms*, 16(3):165, 2023.

[23] Fei Ni, Min Zhang, Pengyi Li, Yifu Yuan, Lingfeng Zhang, Yuecheng Liu, Peilong Han, Longxin Kou, Shaojin Ma, Jinbin Qiao, David Gamaliel Arcos Bravo, Yuening Wang, Xiao Hu, Zhanguang Zhang, Xianze Yao, Yutong Li, Zhao Zhang, Ying Wen, Ying-Cong Chen, Xiaodan Liang, Liang Lin, Bin He, Haitham Bou-Ammar, He Wang, Huazhe Xu, Jiankang Deng, Shan Luo, Shuqiang Jiang, Wei Pan, Yang Gao, Stefanos Zafeiriou, Jan Peters, Yuzheng Zhuang, Yingxue Zhang, Yan Zheng, Hongyao Tang, and Jianye Hao. Embodied arena: A comprehensive, unified, and evolving evaluation platform for embodied ai, 2025.

[24] Barak Or. Mttr-a: Measuring cognitive recovery latency in multi-agent systems. *arXiv preprint arXiv:2511.20663*, 2025.

[25] Amanda Prorok, Matthew Malencia, Luca Carlone, Gaurav S Sukhatme, Brian M Sadler, and Vijay Kumar. Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. *arXiv preprint arXiv:2109.12343*, 2021.

[26] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, volume 2024, pages 15306–15336, 2024.

[27] Neetesh Sharma, Armin Tabandeh, and Paolo Gardoni. Resilience analysis: A mathematical formulation to model resilience of engineering systems. *Sustainable and Resilient Infrastructure*, 3(2):49–67, 2018.

[28] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[29] Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. Replan: Robotic replanning with perception and language models. *arXiv preprint arXiv:2401.04157*, 2024.

[30] Xin Tan, Bangwei Liu, Yicheng Bao, Qij // i jian Tian, Zhenkun Gao, Xiongbin Wu, Zhihao Luo, Sen Wang, Yuqi Zhang, Xuhong Wang, et al. Towards safe and trustworthy embodied ai: foundations, status, and prospects. *Shanghai AI Lab., Shanghai, China, Tech. Rep.*, 2025.

[31] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[32] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[33] David D Woods. Four concepts for resilience and the implications for the future of resilience engineering. *Reliability engineering & system safety*, 141:5–9, 2015.

[34] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

[35] Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*, 2024.

[36] Naoki Yokoyama, Sehoon Ha, and Dhruv Batra. Success weighted by completion time: A dynamics-aware evaluation criteria for embodied navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1562–1569. IEEE, 2021.

[37] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1467–1490, 2024.

[38] Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, et al. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025.

[39] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11142–11152, 2025.

[40] Tao Zhang, Kaixian Qu, Zhibin Li, Jiajun Wu, Marco Hutter, Manling Li, and Fan Shi. Using large language models for embodied planning introduces systematic safety risks. *arXiv preprint arXiv:2604.18463*, 2026.

[41] Zihao Zhu, Bingzhe Wu, Zhengyou Zhang, Lei Han, Qingshan Liu, and Baoyuan Wu. Earbench: Towards evaluating physical risk awareness for task planning of foundation model-based embodied ai agents. *arXiv preprint arXiv:2408.04449*, 2024.