

Multi-scale Mixture of World Models for Embodied Agents in Evolving Environments

Jinwoo Jang¹ , Daniel J. Rho¹ , Sihyung Yoon¹ , Hyunsuk Cho¹ , and
Honguk Woo^{1,2} 

¹ Sungkyunkwan University

{jinustar,danielrho,godboy3752,hscho5133,hwoo}@skku.edu

² Corresponding author

Abstract. Embodied agents operating in the real world require multi-scale reasoning and knowledge adaptation as conditions change. We identify two challenges in applying Mixture of Experts (MoE) to this setting: routing lacks an explicit notion of scale, preventing targeted updates at specific scales, and a uniform update policy cannot accommodate the different rates at which knowledge at each scale becomes outdated. We present MuSix, a framework that addresses both challenges through scale-aware world model mixture and evolution. A two-stage routing mechanism first maps experiential distance, a measure of situational novelty inspired by Construal Level Theory, to a weight over continuous scale space via a meta-router, then selects world models within the identified scale. For adaptation, scale-dependent forgetting rates allow low-scale knowledge to refresh rapidly while high-scale abstractions persist, and gated inter-scale transfer maintains coherence across the hierarchy. Experiments on EmbodiedBench and HAZARD show that MuSix improves over state-of-the-art baselines.

Keywords: Embodied AI · World model · Mixture of experts · Test-time training

1 Introduction

Embodied agents powered by vision-language models (VLMs) have achieved significant progress in complex instruction following [7, 8, 11], yet real-world deployment demands multi-scale reasoning, from low-level physical dynamics to high-level abstract inference, while continuously adapting to non-stationary environments. Recent work has explored hierarchical world models that decompose predictions across scales [2, 6, 20], but these approaches do not address how world models should adapt when environmental conditions change over time.

Mixture of Experts (MoE) [9] offers a natural substrate for these demands: its modular expert selection can accommodate qualitatively different knowledge types, while its selective activation enables targeted updates without disrupting other components, effectively mitigating catastrophic forgetting [12, 18]. However, conventional MoE is not scale-aware, introducing two limitations (Figure 1). First, standard routing operates without an explicit notion of scale, so

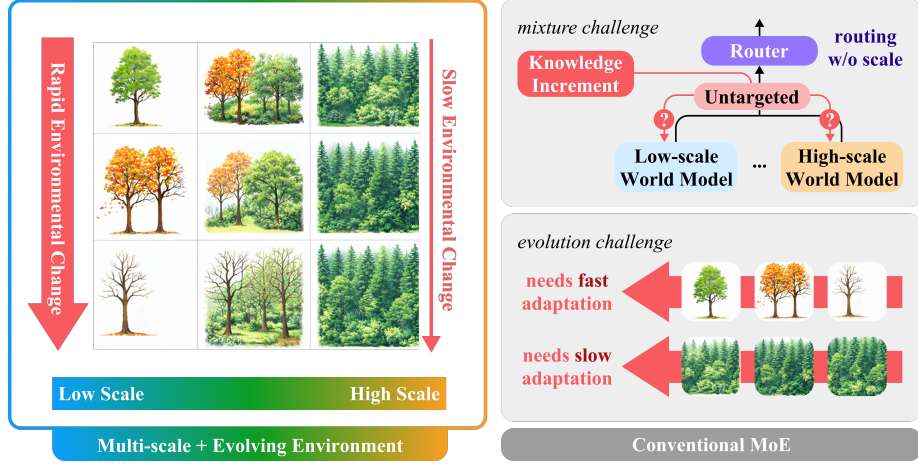


Fig. 1: Explanation of mixture challenge and evolution challenge on conventional MoE

world model selection is not tied to any identifiable scale, precluding test-time updates that target only the relevant scale (*mixture challenge*). Second, a single uniform update policy cannot respect the fact that low-level knowledge about local dynamics changes frequently while high-level abstract rules remain relatively stable, preventing each scale from evolving at its own appropriate rate (*evolution challenge*).

To address these limitations, we propose MUSIX, a multi-scale mixture of world models framework that enables embodied agents to dynamically mix and evolve world models at different scales. For the *mixture challenge*, we introduce two-stage scale-based routing that explicitly separates scale determination from world model selection, yielding transparent routing in which the identified scale directly determines how knowledge increments are distributed across groups. For the *evolution challenge*, we propose intra- and inter-scale knowledge adaptation mechanisms that allow each scale to evolve at its own characteristic rate while maintaining coherence across the hierarchy. To ground scale selection in a principled criterion, we draw on Construal Level Theory (CLT) [16], which posits that psychological distance governs the level of abstraction in human reasoning, and operationalize this principle through experiential distance, a measure of how novel the current situation is relative to the agent’s accumulated experience.

To evaluate MUSIX, we conduct experiments on EmbodiedBench [19] and HAZARD [22], covering both multi-scale reasoning and dynamic adaptation scenarios. EmbodiedBench evaluates agents across diverse capability dimensions including complex reasoning and spatial understanding, while HAZARD tests adaptation under evolving disaster conditions such as fire, flood, and wind. Our framework achieves 6.05%p improvement over SayCanPay [7] on EmbodiedBench (Habitat) and 1.49%p improvement over FLARE [11] on HAZARD (Fire), demonstrating effective multi-scale world model mixture and evolution.

Our contributions are as follows:

- We propose MUsIX, a multi-scale mixture of world models framework for embodied agents that grounds world model selection in experiential distance, allowing agents to dynamically mix and evolve scale-specific world models at test time.
- We introduce two-stage scale-based routing that decomposes routing into scale determination via a continuous meta-router and world model selection via per-scale base routers, enabling scale-aware mixture and targeted test-time updates at the appropriate scale.
- We devise intra- and inter-scale knowledge adaptation mechanisms that respect the distinct temporal characteristics at each scale, enabling rapid adjustment of transient low-level details while preserving stable high-level abstractions through gated cross-scale transfer.
- We validate MUsIX on EmbodiedBench and HAZARD benchmarks, demonstrating that scale-aware routing and knowledge adaptation yield 6.05%p and 1.49%p improvements over state-of-the-art baselines on multi-scale reasoning and dynamic adaptation scenarios, respectively.

2 Related Works

LM- and VLM-based embodied instruction following. VLM-based embodied instruction following studies how an agent grounds natural-language instructions and vision observations in a physical environment through sequential action execution. Prior work has explored different strategies, including affordance-grounded approaches that combine LM reasoning with environmental feedback to evaluate feasible actions [1, 8], and end-to-end vision-language models that directly map multimodal observations to action sequences [5, 24]. However, these approaches are often built around a fixed set of environmental assumptions or a monolithic model component, which hinders targeted adaptation to new environments and forces full system retraining even when only specific components need updating. Our MUsIX framework addresses this by adopting a Mixture-of-Experts (MoE) design that decomposes world knowledge into modular components and selects among them during inference.

Mixture-of-Experts. Mixture-of-Experts architectures have shown strong potential for domain adaptation, routing each input through a selected subset of specialized experts, allowing targeted updates to only the relevant experts during adaptation and thus mitigating catastrophic forgetting [21]. This routing paradigm has been extended to diverse settings, including vision-language adaptation [14] and multi-task learning [13]. However, conventional MoE routing is typically learned over latent representations and is often associated with specific tasks or domains, rather than an explicit notion of scale. As a result, it can be nontrivial to consistently invoke or adapt a desired knowledge granularity at test time without affecting other experts. Our approach addresses this by introducing experiential distance as an explicit, continuous scale axis that grounds routing decisions in a measurable quantity. This enables both interpretable world model selection and targeted test-time training that updates only the relevant scales.

Construal Level Theory (CLT). Construal Level Theory (CLT) [16] posits that psychological distance determines the level of abstraction at which people mentally represent objects and events. Psychologically close entities elicit concrete, detail-rich representations (low-level construal), whereas distant entities elicit abstract, schematic representations (high-level construal). This distance-abstraction pattern has been documented across temporal, spatial, social, and hypothetical dimensions of psychological distance [17]. While CLT offers a well-established account of how distance modulates abstraction in human cognition, its potential as a measurable scale axis for routing mechanisms in MoE-based systems remains underexplored. Drawing on this correspondence, we define experiential distance as a continuous scalar that captures the gap between the agent’s accumulated experience and a given observation, and use it to govern hierarchical expert selection.

3 MuSix

3.1 Overall Framework

We present MuSix, a framework for embodied agents that integrates qualitatively different types of knowledge through scale-aware world model selection driven by experiential distance, a scalar measure of how novel the current situation is relative to the agent’s accumulated experience. As illustrated in Figure 2, the framework addresses the two challenges identified in Section 1 through its core components.

Two-stage scale-based routing (Section 3.2) addresses the *mixture challenge*. In the first stage, a meta-router maps experiential distance to a weight function over a continuous scale space. In the second stage, these scale weights are integrated with per-scale base router scores to produce the final router that selects world models. This decoupled design makes the scale of each routing decision explicit, enabling targeted adaptation at the appropriate scale.

Intra- and inter-scale knowledge adaptation (Section 3.3) addresses the *evolution challenge*. Intra-scale adaptation distributes knowledge increments to the appropriate scale via the meta-router and applies scale-dependent forgetting rates so that each level evolves at its own pace. Inter-scale adaptation then extends this with gated cross-scale knowledge transfer to maintain coherence across the hierarchy.

3.2 Two-stage Scale-based Routing for World Model Mixture

World model groups and mixture of world models. Our framework organizes world models into G groups, ranging from low-scale processing to high-scale reasoning. Each group is associated with a distinct architecture type; for example, an RBF Network can serve as a low-scale world model, while a Cognitive Map can act as a high-scale world model. Here, the world model is a module $m : (o_t, a_t) \mapsto o_{t+1}$, trained via teacher-forcing on a dataset $\mathcal{D} = \{(\text{task}_i, \tau_i)\}_{i=1}^D$

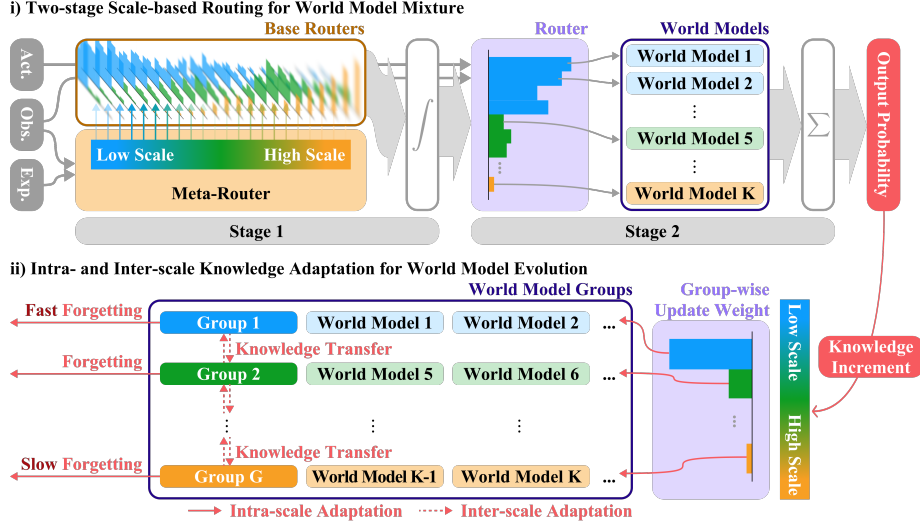


Fig. 2: Overall framework of MuSix

of D task-trajectory pairs, where $\tau_i = \{(o_t, a_t, o_{t+1})\}_{t=1}^T$. The framework comprises N world models $\{m_1, \dots, m_N\}$ distributed across G groups and selects a top- k subset for each input to form the mixture of world models M :

$$M = \sum_{i=1}^N w_{\mathcal{R},i} m_i; \quad w_{\mathcal{R},i} = \text{softmax}(\text{top}_k(\mathcal{R}(o_t, a_t)))_i, \quad (1)$$

where $o_t \in \mathcal{O}$ and $a_t \in \mathcal{A}$ denote the current observation and action, $\mathcal{O}$ and $\mathcal{A}$ are the observation space and the action space, respectively, and $\mathcal{R} : \mathcal{O} \times \mathcal{A} \rightarrow \mathbb{R}^N$ is the router that determines routing weights for each world model.

Continuous meta-router and two-stage routing. A single flat router cannot identify which scale is currently active, making it impossible to selectively adapt world model groups at the relevant scale during test time. To address this, we decompose routing into two stages: a *continuous meta-router* $\mathcal{MR}$ that first determines scale weights based on experiential distance, and a final *router* $\mathcal{R}$ that integrates these weights with per-scale *base router* scores $r(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}$, to select world models.

The meta-router produces a weight function over the scale space, and the router integrates over all scales:

$$\mathcal{R} = \int_{\mathcal{S}} w_{\mathcal{MR}}(\mathbf{s}) r(\mathbf{s}) d\mathbf{s}; \quad w_{\mathcal{MR}}(\mathbf{s}) = \text{softmax}_{\mathcal{S}}(\mathcal{MR}(o_t, \mathbf{E}_{<t}, \mathbf{s})), \quad (2)$$

where $\text{softmax}_{\mathcal{S}}$ denotes the continuous softmax over $\mathcal{S}$,

$$\text{softmax}_{\mathcal{S}}(f(\mathbf{s})) = \frac{\exp(f(\mathbf{s}))}{\int_{\mathcal{S}} \exp(f(\mathbf{s}')) d\mathbf{s}'}, \quad (3)$$

and $\mathcal{MR} : \mathcal{O} \times \mathcal{O}^* \times \mathcal{S} \rightarrow \mathbb{R}$ is the continuous meta-router. $\mathcal{S}$ is a compact subset of a normed vector space with $R_{\mathcal{S}} = \sup_{\mathbf{s} \in \mathcal{S}} \|\mathbf{s}\|$, and $\mathbf{E}_{< t} = (o_1, \dots, o_{t-1}) \in \mathcal{O}^*$ denotes the sequence of observations accumulated within the episode, encoding the agent’s past experience. In practice, we approximate the integrals via Monte Carlo sampling over $\mathcal{S}$.

For each scale $\mathbf{s} \in \mathcal{S}$, the base router $r(\mathbf{s}) : \mathcal{O} \times \mathcal{A} \rightarrow \mathbb{R}^N$ outputs unnormalized world model routing scores, with lower scales favoring low-scale world models and higher scales favoring high-scale world models.

Experiential distance and scale-aware loss. The meta-router conditions on experiential distance δ , which quantifies the deviation of the current situation from the agent’s accumulated experience. Let $\phi : \mathcal{O} \rightarrow \mathbb{R}^d$ be a frozen pretrained encoder. Recall that experience $\mathbf{E}_{< t} = (o_1, \dots, o_{t-1})$ is the observation sequence accumulated within an episode. We model its embedding distribution as a multi-variate Gaussian. To ensure that the distribution reflects the agent’s most recent experience, we assign exponentially decaying weights $w_i = \exp(-\beta(t-i))$, where $\beta > 0$ controls the decay rate. The weighted mean and covariance in the embedding space are:

$$\boldsymbol{\mu}_{\mathbf{E}} = \frac{\sum_{i=1}^{t-1} w_i \phi(o_i)}{\sum_{i=1}^{t-1} w_i}, \quad \boldsymbol{\Sigma}_{\mathbf{E}} = \frac{\sum_{i=1}^{t-1} w_i (\phi(o_i) - \boldsymbol{\mu}_{\mathbf{E}})(\phi(o_i) - \boldsymbol{\mu}_{\mathbf{E}})^\top}{\sum_{i=1}^{t-1} w_i}. \quad (4)$$

The experiential distance is then defined as the Mahalanobis distance between the current observation embedding and this distribution:

$$\delta(o_t, \mathbf{E}_{< t}) = \sqrt{(\phi(o_t) - \boldsymbol{\mu}_{\mathbf{E}})^\top \boldsymbol{\Sigma}_{\mathbf{E}}^{-1} (\phi(o_t) - \boldsymbol{\mu}_{\mathbf{E}})}. \quad (5)$$

When $t < T_{\min}$, where $T_{\min}$ is the minimum number of steps required to estimate $\boldsymbol{\Sigma}_{\mathbf{E}}$, we set $\delta = +\infty$, reflecting that all situations are novel without prior experience.

To align the meta-router with experiential distance, we first normalize δ to a bounded range via $\bar{\delta} = 1 - \exp(-\tau\delta)$, mapping $\delta \in [0, \infty)$ to $[0, 1)$, where $\tau > 0$ controls the transition sharpness. The *scale-aware loss* is then:

$$\mathcal{L}_{\mathcal{S}} = \underbrace{(R_{\mathcal{S}} \bar{\delta} - \|\boldsymbol{\mu}_{\mathcal{S}}\|)^2}_{\mathcal{L}_{\mathcal{S}}^{(1)}} + \underbrace{\lambda_1 \text{tr}(\boldsymbol{\Sigma}_{\mathcal{S}})}_{\mathcal{L}_{\mathcal{S}}^{(2)}} + \underbrace{\lambda_2 Z^2}_{\mathcal{L}_{\mathcal{S}}^{(3)}}, \quad (6)$$

where $\boldsymbol{\mu}_{\mathcal{S}}$ and $\boldsymbol{\Sigma}_{\mathcal{S}}$ are the weighted mean and covariance of $\mathbf{s}$ under $w_{\mathcal{MR}}$, and $\lambda_1, \lambda_2 > 0$ are balancing coefficients. $Z = \int_{\mathcal{S}} \mathcal{MR}(o, \mathbf{E}, \mathbf{s}) d\mathbf{s}$ is the integral of the meta-router output. The first term ($\mathcal{L}_{\mathcal{S}}^{(1)}$) encourages the expected scale magnitude to match the experiential distance, routing familiar situations to lower scales and novel situations to higher scales. The second term ($\mathcal{L}_{\mathcal{S}}^{(2)}$) encourages the scale distribution to concentrate rather than spread diffusely across the scale space. The third term ($\mathcal{L}_{\mathcal{S}}^{(3)}$) prevents the meta-router outputs from diverging.

The overall training objective combines world model prediction loss with scale alignment:

$$\mathcal{L} = \mathcal{L}_{\text{TF}} + \lambda_{\mathcal{S}} \mathcal{L}_{\mathcal{S}}, \quad (7)$$

where $\mathcal{L}_{\text{TF}}$ is the teacher-forcing loss that trains the model to predict next action or observation from the trajectory [10, 23] and $\lambda_{\mathcal{S}} > 0$ controls the strength of scale alignment.

The design of experiential distance and the scale-aware loss draws on Construal Level Theory (CLT) in cognitive science [16], which posits that psychological distance governs the level of abstraction in human reasoning. CLT further posits that this abstraction arises from multiple dimensions, including spatial, temporal, social, and hypothetical distance; our multi-dimensional scale space reflects this structure. Applying this principle, the scale-aware loss routes low- δ situations to low-scale world models that operate on concrete, well-established patterns, and high- δ situations to high-scale world models capable of abstract generalization.

3.3 Intra- and Inter-scale Knowledge Adaptation for World Model Evolution

To enable selective adaptation at each scale, we introduce the intra- and inter-scale knowledge adaptation mechanism that updates world models at test time. We define knowledge states $\{K^{(1)}, \dots, K^{(G)}\}$, where $K^{(g)} \in \mathcal{K}^*$ represents the accumulated knowledge for group g (e.g. neural memory), and $\mathcal{K}$ is a knowledge space. Each group- g world model exclusively references its corresponding knowledge state $K^{(g)}$, maintaining clear separation of scale-specific knowledge.

During inference, the prediction error yields a knowledge increment ΔK_{t+1} at each step $t \rightarrow t+1$ (in the spirit of the surprise metric in [4]). This increment is distributed across scales via the meta-router and integrated with scale-dependent forgetting (*intra-scale adaptation*), and then propagated through gated cross-scale transfer (*inter-scale adaptation*).

Intra-scale adaptation. We distribute each knowledge increment to the appropriate scale via the meta-router and apply scale-dependent forgetting, so that each level adapts at a rate matching the volatility of its content.

Given a knowledge increment ΔK_{t+1} , the agent must first determine which scale requires updating. Our two-stage routing enables this: given the next observation o_{t+1} , the meta-router identifies the current scale as

$$\mathbf{s}^* = \arg \max_{\mathbf{s} \in \mathcal{S}} \mathcal{MR}(o_{t+1}, \mathbf{E}_{<t+1}, \mathbf{s}). \quad (8)$$

The knowledge increment is then distributed across groups according to the group-wise update weight $\eta^{(g)}$:

$$\Delta K_{t+1}^{(g)} = \eta^{(g)} \Delta K_{t+1}; \quad \eta^{(g)} = \sum_{i=1}^N \mathbf{1}[m_i \in \text{group } g] \cdot \text{softmax}(r(\mathbf{s}^*)(o_t, a_t))_i, \quad (9)$$

ensuring that only the relevant scale receives substantial updates.

Each group then integrates its increment with scale-dependent forgetting:

$$K_{t+1}^{(g)} = (1 - \alpha^{(g)})K_t^{(g)} + \Delta K_{t+1}^{(g)}, \quad (10)$$

where $\alpha^{(g)} = \alpha_{\max} \exp(-\gamma(g-1))$, with $\alpha_{\max}$ controlling the maximum forgetting rate and $\gamma > 0$ determining the decay across groups. Low-scale groups exhibit rapid forgetting ($\alpha^{(g)}$ large) while high-scale groups retain information over extended periods ($\alpha^{(g)}$ small).

Inter-scale adaptation. We extend the isolated per-group update (Equation 10) with gated bidirectional knowledge transfer across neighboring scales, so that knowledge gained at one level can inform adjacent levels. Gating parameters $\mathbf{W}_+^{(g)}, \mathbf{W}_-^{(g)}$ modulate information flow between neighboring groups:

$$g_+^{(g)} = \sigma(\phi(o_{t+1})\mathbf{W}_+^{(g)}), \quad g_-^{(g)} = \sigma(\phi(o_{t+1})\mathbf{W}_-^{(g)}), \quad (11)$$

where $\mathbf{W}_+^{(g)}$ and $\mathbf{W}_-^{(g)}$ are learned during training and held fixed at test time, while only the knowledge states $K^{(g)}$ are updated. With cross-scale transfer, the update rule in Equation 10 extends to:

$$K_{t+1}^{(g)} = (1 - \alpha^{(g)})K_t^{(g)} + \Delta K_{t+1}^{(g)} + \Delta K_{\text{in},t+1}^{(g)} - \Delta K_{\text{out},t+1}^{(g)}, \quad (12)$$

where

$$\Delta K_{\text{in},t+1}^{(g)} = g_+^{(g)} \odot K_t^{(g-1)} + g_-^{(g)} \odot K_t^{(g+1)}, \quad (13)$$

$$\Delta K_{\text{out},t+1}^{(g)} = g_+^{(g+1)} \odot K_t^{(g)} + g_-^{(g-1)} \odot K_t^{(g)}. \quad (14)$$

Terms involving $K^{(0)}, K^{(G+1)}, g_+^{(G+1)}$, or $g_-^{(0)}$ are set to zero at boundaries. The incoming term $\Delta K_{\text{in},t+1}^{(g)}$ aggregates knowledge from neighboring groups, while the outgoing term $\Delta K_{\text{out},t+1}^{(g)}$ accounts for knowledge transferred away, ensuring consistency across the hierarchy.

4 Experiments

We evaluate MUsIX on two complementary benchmarks: EmbodiedBench [19] for multi-scale reasoning capabilities and HAZARD [22] for adaptation in dynamic environments.

4.1 Experimental Setup

Benchmarks. EmbodiedBench evaluates embodied agents across six capability dimensions in two simulation environments. EB-Habitat tests household task completion requiring object manipulation and navigation, while EB-Navigation focuses on indoor navigation with realistic visual observations. HAZARD evaluates agents in dynamic disaster scenarios (fire, flood) where environmental conditions evolve over time, requiring continuous adaptation of both low-level reactive behaviors and high-level rescue strategies.

Table 1: Evaluation result on EB-Habitat and EB-Navigation. We report success rate (%) for each capability subset within EB-Habitat and EB-Navigation environments.

Methods	EB-Habitat							EB-Navigation						
	Base	Cmn	Cpx	Vis	Spt	Lng	Avg ($\uparrow$)	Base	Cmn	Cpx	Vis	Lng	Avg ($\uparrow$)	
LLM-Planner	53.33	2.67	18.67	22.67	30.67	10.67	23.11 $\pm$ 1.45	36.80	30.40	27.20	28.80	1.60	24.96 $\pm$ 0.67	
SayCanPay	72.00	11.67	30.67	37.33	37.33	17.33	34.39 $\pm$ 2.46	54.40	47.20	57.60	48.00	28.80	47.20 $\pm$ 0.98	
FLARE	72.00	4.00	18.67	29.33	28.00	10.67	27.11 $\pm$ 3.48	45.60	35.20	44.00	32.00	14.40	34.24 $\pm$ 3.64	
Conventional MoE	60.00	10.67	8.00	32.00	30.67	12.00	25.56 $\pm$ 1.36	36.00	28.80	39.20	32.80	9.60	29.28 $\pm$ 2.50	
MuSix	73.33	26.67	37.33	42.67	36.00	26.67	40.44$\pm$1.27	63.20	63.20	57.60	62.40	43.20	57.92$\pm$1.75	

Baselines. We compare against representative methods spanning different paradigms: **LLM-Planner** [15] generates plans through few-shot prompting without explicit world modeling. **SayCanPay** [7] combines language model planning with learned affordance functions. **FLARE** [11] employs retrieval-augmented generation for grounded action prediction. For ablation, we include a **Conventional MoE** variant that uses standard top- k routing without scale-aware mechanisms.

Implementation details. We use Qwen3-VL-4B-Instruct [3] as the vision-language backbone. Our framework consists of $G = 3$ world model groups with $K = 15$ world models distributed across scales, and operates in a 3-dimensional scale space. We select top-4 world models per input. For knowledge adaptation, we set $\alpha_{\max} = 0.3$ and $\gamma = 1.0$. Hyperparameters are selected via grid search.

4.2 Main Results

Multi-scale reasoning on EmbodiedBench. Table 1 presents results across six capability dimensions. MuSix achieves consistent improvements over baselines in both EB-Habitat and EB-Navigation environments. Notably, our method shows substantial gains on tasks requiring complex reasoning (Cpx) and visual understanding (Vis), which demand effective integration of abstract planning with concrete physical inference. The improvement over Conventional MoE demonstrates that scale-aware routing is essential for leveraging the multi-scale architecture effectively.

Adaptation in dynamic environments on HAZARD. Table 2 shows results on disaster scenarios with evolving environmental conditions. In the fire scenario, MuSix achieves the highest rescue value among all methods. In the flood scenario, rescue values are comparable across methods within error margins, but MuSix notably reduces the damage ratio (4.65 vs. 6.60–7.49 for baselines), indicating more efficient rescue behavior that avoids unnecessary environmental exposure. These results suggest that multi-scale routing and knowledge adaptation enable the agent to balance quick low-level reactions with stable high-level strategies as conditions evolve.

Table 2: Evaluation result on HAZARD benchmark. We report value of rescued objects (Val), number of steps (Step), and damage ratio (Dmg) across three dynamic disaster scenarios. We use 3 random seeds.

Methods	Fire			Flood		
	Val ($\uparrow$)	Step ($\downarrow$)	Dmg ($\downarrow$)	Val ($\uparrow$)	Step ($\downarrow$)	Dmg ($\downarrow$)
LLM-Planner	38.48 $\pm$ 7.91	1285.42 $\pm$ 272.72	2.53 $\pm$ 0.13	49.78 $\pm$ 0.07	1145.20 $\pm$ 7.46	6.60 $\pm$ 0.35
SayCanPay	41.07 $\pm$ 1.43	1467.72 $\pm$ 12.50	2.55 $\pm$ 0.11	49.83 $\pm$ 0.00	1148.17 $\pm$ 2.56	6.80 $\pm$ 0.50
FLARE	42.22 $\pm$ 0.75	1480.48 $\pm$ 7.14	2.88 $\pm$ 0.32	49.00 $\pm$ 1.18	1133.90 $\pm$ 23.10	6.71 $\pm$ 0.46
Conventional MoE	41.53 $\pm$ 0.83	1500.77 $\pm$ 7.73	2.89 $\pm$ 0.12	49.83 $\pm$ 0.00	1123.30 $\pm$ 18.03	7.49 $\pm$ 0.68
MuSix	43.71 $\pm$ 0.82	1466.2 $\pm$ 17.90	1.95 $\pm$ 0.45	49.83 $\pm$ 0.00	1130.7 $\pm$ 12.70	4.65 $\pm$ 0.60

Table 3: Real-world robotic manipulation with Franka Research 3. We report success rate (%) across 8 manipulation tasks.

Method	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Avg ($\uparrow$)
LLM-Planner	66.7	16.7	33.3	7.6	50.0	0.0	0.0	0.0	21.8 $\pm$ 2.0
SayCanPay	66.7	66.7	83.3	33.3	83.3	66.7	17.5	50.0	58.4 $\pm$ 0.9
FLARE	66.7	16.7	33.3	28.5	50.0	0.0	0.0	0.0	24.4 $\pm$ 2.8
Conventional MoE	100.0	100.0	0.0	33.3	73.3	70.0	30.0	23.3	53.7 $\pm$ 2.7
MuSix	66.7	50.0	83.3	50.0	76.7	66.7	30.0	26.7	56.2 $\pm$ 1.9



Fig. 3: Real-world robotic manipulation examples

Real-world robotic manipulation. To validate the practical applicability of MuSix, we conduct real-world experiments using a Franka Research 3 robot arm (Figure 3), evaluating across 8 manipulation tasks (Table 3). MuSix outperforms most baselines and performs comparably to the strongest one, SayCanPay, in average success rate. Moreover, MuSix attains the highest worst-case performance across tasks, indicating more consistent behavior under diverse real-world conditions.

4.3 Ablation Study

We conduct ablation studies to validate the contribution of each component. Table 4 shows all ablations evaluated on both EmbodiedBench and HAZARD.

Table 4: Ablation study. We report success rate (%) on EmbodiedBench and rescue value on HAZARD.

Method	EmbodiedBench (SR $\uparrow$)	HAZARD (Value $\uparrow$)	
	EB-Habitat	Fire	Flood
MuSix (full)	40.44	43.71	49.83
<i>Two-stage Scale-based Routing</i>			
w/o meta-router	32.22	42.22	49.83
w/o $\mathcal{L}_S^{(1)}$ (alignment)	37.78	39.85	45.38
w/o $\mathcal{L}_S^{(2)}$ (variance)	35.56	44.89	49.83
w/o $\mathcal{L}_S^{(3)}$ (magnitude)	38.89	44.52	49.83
<i>Intra- and Inter-scale Knowledge Adaptation</i>			
w/o intra-scale adaptation	38.89	39.84	45.38
w/o inter-scale adaptation	37.78	41.67	45.38

Ablations of two-stage scale-based routing. Removing the meta-router (w/o meta-router) causes the largest overall degradation, with EB-Habitat dropping from 40.44 to 32.22, confirming that the two-stage hierarchy is essential for effective scale-aware routing. Without the meta-router, the current scale $\mathbf{s}^*$ cannot be identified, so the scale-group weight for knowledge adaptation falls back to $\eta^{(g)} = \sum_{i \in \text{group } g} w_{R,i}$, which simply aggregates routing weights without any explicit scale information. This validates the core motivation of two-stage routing, where identifying the current scale enables selective adaptation of world model groups at the relevant scale, a capability that scale-agnostic approaches cannot achieve.

We further ablate each term of the scale-aware loss $\mathcal{L}_S$ (Equation 6) to assess its individual contribution. Removing the alignment term (w/o $\mathcal{L}_S^{(1)}$) leads to degradation across all benchmarks, confirming that explicit alignment between experiential distance and scale is necessary for the meta-router to learn scale-aware routing. The variance and magnitude regularizers (w/o $\mathcal{L}_S^{(2)}$, $\mathcal{L}_S^{(3)}$), in contrast, show environment-dependent effects. They improve stability on EB-Habitat but offer only marginal or mixed impact on HAZARD, suggesting that their inclusion should be tuned per deployment environment.

Ablations of intra- and inter-scale knowledge adaptation. Disabling scale-dependent forgetting (w/o intra-scale) and applying uniform forgetting rates across all scales is harmful, as this variant either forgets stable high-scale knowledge too quickly or retains outdated low-scale details too long. Removing cross-scale knowledge transfer (w/o inter-scale) also results in consistent degradation. Without bidirectional information flow, knowledge remains siloed within each scale, preventing coordinated adaptation across the hierarchy.

Table 5: World model group compositions. Each variant assigns a specific architecture type to each group, ordered from low-scale (Group 1) to high-scale (Group G).

Variant	G	Group 1	Group 2	Group 3	Group 4
I (default)	3	PINN	Cognitive Map	Relational Network	–
II	3	RBF Network	Concept Bottleneck	Schema Network	–
III	4	Sensory	PINN	Relational Network	Concept Bottleneck
IV	4	Sensory	RBF Network	Cognitive Map	Schema Network
V	2	PINN	Schema Network	–	–

Table 6: World model group variants on EmbodiedBench. We report success rate (%) for each capability subset. Variant I is our default configuration. We use 3 random seeds.

Variant	G	EB-Habitat							EB-Navigation					
		Base	Cmn	Cpx	Vis	Spt	Lng	Avg (↑)	Base	Cmn	Cpx	Vis	Lng	Avg (↑)
I (default)	3	73.33	26.67	37.33	42.67	36.00	26.67	40.44±1.27	63.20	63.20	57.60	62.40	43.20	57.92±1.75
II	3	75.56	26.67	44.45	37.78	35.56	33.34	42.22±2.22	65.33	57.33	61.33	54.67	32.00	54.13±3.95
III	4	71.11	22.22	37.78	35.56	35.56	31.11	38.89±5.88	62.67	62.67	60.00	57.33	44.00	57.33±5.33
IV	4	77.78	24.44	24.45	40.00	37.78	31.11	39.26±2.80	58.67	65.33	57.33	56.00	44.00	56.27±1.67
V	2	73.33	28.89	35.56	28.89	37.78	28.89	38.89±2.94	66.67	62.67	61.33	53.33	44.00	57.60±1.39

4.4 Analysis

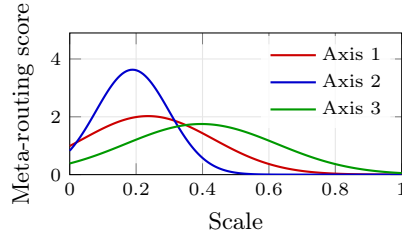
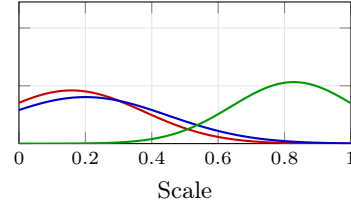
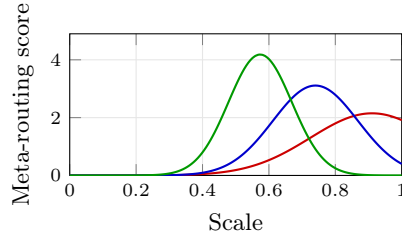
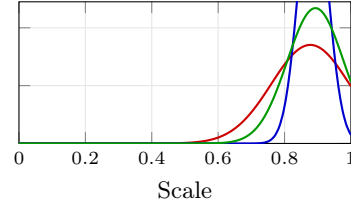
World model group variants. To verify that MUsix is not tied to a specific set of world model architectures, we evaluate five group configurations (Variants I–V) on EmbodiedBench, varying both the number of groups and architecture types. Detailed compositions are provided in Table 5. Low-scale groups employ architectures suited to modeling physical dynamics and local patterns, mid-scale groups capture relational and spatial structure, and high-scale groups handle abstract concept-level or rule-based reasoning.

Table 6 shows that all five variants fall within a narrow range (38.89–42.22% on EB-Habitat, 54.13–57.92% on EB-Navigation), and even the weakest surpasses the strongest baseline (SayCanPay; 34.39% and 47.20%; Table 1) by a clear margin. This confirms that the framework’s effectiveness stems from the scale-aware routing mechanism rather than from a particular choice of world model architectures.

Role of scale space axes. Since the scale-aware loss aligns the norm of the expected scale vector with experiential distance, the scale space may capture multiple aspects of situational novelty rather than a single one. We compare performance across 1D, 2D, and 3D scale spaces on EmbodiedBench using the L1 norm (Table 7). The 3D scale space yields the best average performance in both environments among the dimensionalities we evaluate, leading us to adopt it.

Table 7: Effect of scale space dimensionality on EmbodiedBench. We report success rate (%) for each capability subset.

Dim	EB-Habitat							EB-Navigation					
	Base	Cmn	Cpx	Vis	Spt	Lng	Avg ($\uparrow$)	Base	Cmn	Cpx	Vis	Lng	Avg ($\uparrow$)
1D	86.67	13.33	20.00	40.00	40.00	33.33	38.89	68.00	48.00	56.00	60.00	36.00	53.60
2D	80.00	13.33	20.00	26.67	40.00	20.00	33.33	60.00	64.00	60.00	64.00	36.00	56.80
3D (default)	73.33	26.67	37.33	42.67	36.00	26.67	40.44	63.20	63.20	57.60	62.40	43.20	57.92

**(a)** Routine object manipulation in a familiar room. Axes concentrate toward low scales.**(b)** Navigation to an unseen room with unfamiliar layout. Axis 3 moves to high scales while the others stay low.**(c)** Multi-step task requiring sequential subgoal planning. Axes 1 and 2 move to high scales while Axis 3 stays lower.**(d)** First encounter with a completely novel environment. All axes shift to high scales.**Fig. 4: Per-axis meta-routing score distributions for four input types on EmbodiedBench.** Each curve shows $w_{\mathcal{MR}}(\mathbf{s})$ projected onto one axis of the 3D scale space.

To understand why additional dimensions help, we inspect how individual axes respond to different inputs (Figure 4). We find that the axes move toward low or high scales depending on the novelty of the current observation, sometimes converging across the three axes and sometimes separating. Notably, the scale-aware loss supervises only $\|\mu_{\mathcal{S}}\|$, yet the axes respond differently across situations rather than in unison, indicating that each axis comes to encode a distinct aspect of novelty without any axis-level supervision. This emergent specialization suggests that a higher-dimensional scale space offers the capacity for such structure to arise, accounting for the consistent gains we observe from higher dimensionality.

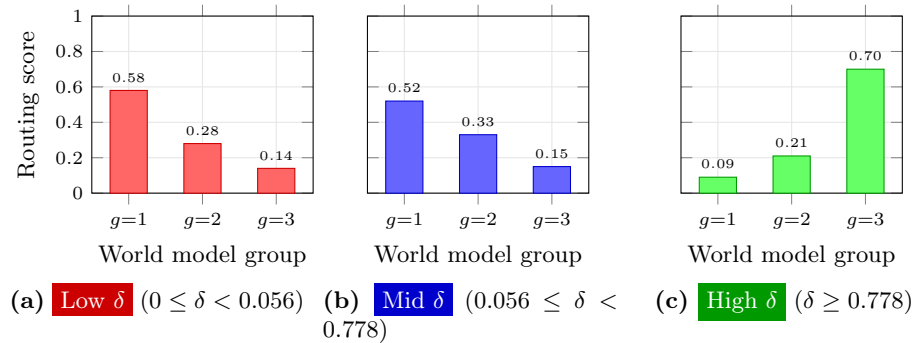


Fig. 5: World model group activation at different experiential distances on EmbodiedBench. Each bar shows the aggregate routing score for group g ($g = 1$: low-scale, $g = 2$: mid-scale, $g = 3$: high-scale). Low, Mid, and High denote equal-frequency tertiles of experiential distance δ (each $n \approx 790$).

Experiential distance and world model group activation. We examine how experiential distance δ translates into world model group activation on EmbodiedBench (Figure 5). We group samples into equal-frequency tertiles of δ , so the bin boundaries follow the empirical distribution of δ . Routing concentrates on the low-scale group across the **lower** and **middle** tertiles ($\delta < 0.778$), with a slight shift of mass toward the mid-scale group in the middle tertile, and moves to the high-scale group once δ reaches the **top tertile** ($\delta \geq 0.778$). This shows that δ aligns world model activation with the appropriate scale, with familiar situations handled by low-scale models and novel ones routed to high-scale models.

5 Conclusion

We presented MUSIX, a multi-scale mixture of world models framework that enables embodied agents to dynamically select and adapt world models across scales. By grounding routing in experiential distance, a measure of situational familiarity inspired by Construal Level Theory, our two-stage scale-based routing decomposes world model selection into scale determination and per-scale expert selection, making the scale of each decision explicit and amenable to targeted test-time updates. Our intra- and inter-scale knowledge adaptation mechanisms respect the distinct temporal characteristics at each scale, allowing low-level knowledge to update rapidly while preserving stable high-level abstractions, with gated cross-scale transfer maintaining coherence across the hierarchy. Experiments on EmbodiedBench and HAZARD demonstrate consistent improvements over state-of-the-art baselines in both multi-scale reasoning and dynamic adaptation scenarios, and analyses confirm that the framework generalizes across different world model architectures.

Limitations and future work. Our framework inherits the capabilities and limitations of the underlying language model; reasoning quality is bounded by the VLM backbone, and improvements in base model capacity would directly benefit our approach. While our real-world experiments demonstrate initial transfer, broader validation across diverse robotic platforms and longer-horizon tasks would further establish the practical applicability of the approach. Promising future directions include automatically discovering the appropriate number and composition of world model groups from data, and extending real-world evaluation to a wider range of embodied platforms and task domains.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT), (RS-2022-II220043, Adaptive Personality for Intelligent Agents, RS-2022-II221045, Self-directed multi-modal Intelligence for solving unknown, open domain problems, RS-2025-02218768, Accelerated Insight Reasoning via Continual Learning, RS-2025-25442569, AI Star Fellowship Support Program (Sungkyunkwan Univ.), RS-2026-25543726, Development of Leading Talent in Medical Domain-Specific Generative AI, RS-2026-25528384, Resource-Intensive AI Technologies Based on Sustainable GPU Integrated Platforms, RS-2019-III190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25474409), IITP-ITRC (Information Technology Research Center) grant funded by the Korea government (MSIT) (IITP-2025-RS-2024-00437633), and by Samsung Electronics Co., Ltd.

References

1. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R.M.J., Jeffrey, K., Jesmonth, S., Joshi, N.J., Julian, R.C., Kalashnikov, D., Kuang, Y., Lee, K.H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D.M., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M.: Do as i can, not as i say: Grounding language in robotic affordances. In: Conference on Robot Learning (2022)
2. Azzolini, A., Bai, J., Brandon, H., Cao, J., Chattopadhyay, P., Chen, H., Chu, J., Cui, Y., Diamond, J., Ding, Y., et al.: Cosmos-reason1: From physical common sense to embodied reasoning. arXiv preprint arXiv:2503.15558 (2025)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang,

- F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Behrouz, A., Zhong, P., Mirrokni, V.: Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems* **38**, 113506–113543 (2026)
5. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q.H., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P.R.: Palm-e: An embodied multimodal language model. In: *International Conference on Machine Learning* (2023)
6. Gumbsch, C., Sajid, N., Martius, G., Butz, M.V.: Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In: *The Twelfth International Conference on Learning Representations* (2023)
7. Hazra, R., Dos Martires, P.Z., De Raedt, L.: Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 20123–20133 (2024)
8. Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P.R., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., Ichter, B.: Inner monologue: Embodied reasoning through planning with language models. *ArXiv* (2022)
9. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural computation* **3**(1), 79–87 (1991)
10. Janner, M., Li, Q., Levine, S.: Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems* **34**, 1273–1286 (2021)
11. Kim, T., Kim, B., Choi, J.: Multi-modal grounded planning and efficient replanning for learning embodied agents with a few examples. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 39, pp. 4329–4337 (2025)
12. Li, H., Lin, S., Duan, L., Liang, Y., Shroff, N.B.: Theory on mixture-of-experts in continual learning. arXiv preprint arXiv:2406.16437 (2024)
13. Ma, J.W., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018)
14. Shen, S., Yao, Z., Li, C., Darrell, T., Keutzer, K., He, Y.: Scaling vision-language models with sparse mixture of experts. In: *Conference on Empirical Methods in Natural Language Processing* (2023)
15. Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.L., Su, Y.: Llm-planner: Few-shot grounded planning for embodied agents with large language models. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 2998–3009 (2023)
16. Trope, Y., Liberman, N.: Construal-level theory of psychological distance. *Psychological review* **117**(2), 440 (2010)
17. Trope, Y., Liberman, N., Wakslak, C.J.: Construal levels and psychological distance: Effects on representation, prediction, evaluation, and behavior. *Journal of consumer psychology : the official journal of the Society for Consumer Psychology* **17**, 83–95 (2007)
18. Willi, T., Obando-Ceron, J., Foerster, J., Dziugaite, K., Castro, P.S.: Mixture of experts in a mixture of rl settings. arXiv preprint arXiv:2406.18420 (2024)
19. Yang, R., Chen, H., Zhang, J., Zhao, M., Qian, C., Wang, K., Wang, Q., Koripella, T.V., Movahedi, M., Li, M., et al.: Embodiedbench: Comprehensive benchmark-

- ing multi-modal large language models for vision-driven embodied agents. arXiv preprint arXiv:2502.09560 (2025)
20. Zadem, M., Mover, S., Nguyen, S.M.: Reconciling spatial and temporal abstractions for goal representation. arXiv preprint arXiv:2401.09870 (2024)
 21. Zhong, T., Chi, Z., Gu, L., Wang, Y., Yu, Y., Tang, J.: Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. ArXiv (2022)
 22. Zhou, Q., Chen, S., Wang, Y., Xu, H., Du, W., Zhang, H., Du, Y., Tenenbaum, J.B., Gan, C.: Hazard challenge: Embodied decision making in dynamically changing environments. arXiv preprint arXiv:2401.12975 (2024)
 23. Zhu, C., Yu, R., Feng, S., Burchfiel, B., Shah, P., Gupta, A.: Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. arXiv preprint arXiv:2504.02792 (2025)
 24. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)



Fig. 6: The top-view example of the Habitat environment.

A Benchmarks

A.1 EmbodiedBench

We conduct our experiments using EmbodiedBench [19], a comprehensive benchmark designed to evaluate vision-driven embodied agents powered by multi-modal large language models. It comprises 1,128 evaluation instances across four environments, covering tasks from high-level instruction following to low-level navigation and manipulation. In our experiments, we focus on the EB-Habitat and EB-Navigation environments.

Environment structure. EmbodiedBench follows an episodic interaction protocol in which each instance provides a natural-language instruction and an initial environment configuration. At each step, the agent receives egocentric visual observations (along with textual feedback on action validity and optional auxiliary signals depending on the environment) and outputs an executable action plan.

Action space. EmbodiedBench provides a hierarchical action space that bridges high-level semantic planning and low-level atomic control. In EB-Habitat, the action space consists of high-level skills (e.g., `navigate`, `pick`, `place`, `open`, `close`) parameterized by specific objects or receptacles. This formulation requires agents to focus on task decomposition and logical sequencing. In EB-Navigation, the action space is defined by discrete atomic commands directly executable by physical robots. It employs 8 discrete actions for translational movements (`forward/backward/left/right` by Δx), yaw rotations ($\Delta\theta$), and camera tilt adjustments ($\Delta\varphi$).

Task specifications. To ensure rigorous assessment, we partition the dataset of the selected environments. In our experimental setup, the first 35 episodes

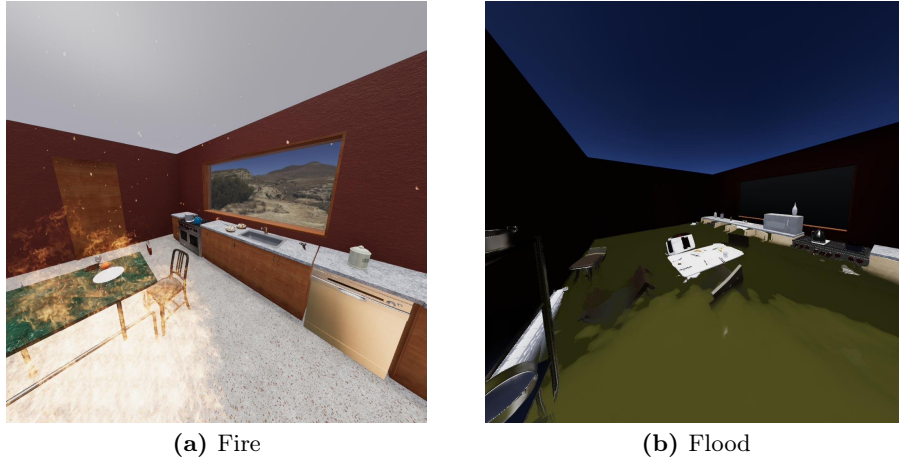


Fig. 7: The example scene of the HAZARD environment.

of each subset are exclusively reserved as training data and a knowledge pool for retrieval-augmented in-context demonstrations, while the remaining unseen episodes (36 to 50 for EB-Habitat, and 36 to 60 for EB-Navigation) are used for evaluation. Task success is defined via PDDL-based goal conditions in EB-Habitat, and by a predefined distance threshold to the target in EB-Navigation. To evaluate diverse reasoning capabilities, tasks are structured into fine-grained subsets. EB-Habitat utilizes all six subsets (Base, Common Sense, Complex Instruction, Visual Appearance, Spatial Awareness, and Long Horizon), while EB-Navigation employs five, omitting spatial awareness.

A.2 HAZARD

HAZARD [22] is a simulated embodied benchmark built on the ThreeDWorld platform, where agents must rescue valuable objects from dynamically evolving disaster scenarios. It features three disaster types—fire, flood, and wind—in which the environment spontaneously changes over time through spreading flames, rising water, or turbulent winds, demanding rapid perception, reasoning, and action from the agent.

Environment structure. At each step, the agent receives RGB-D observations, environment-specific signals such as temperature or water level, target object information (e.g., object names and associated values) along with an optional semantic segmentation mask depending on the evaluation setting. Each disaster scenario introduces distinct environmental dynamics. In fire, a temperature-based propagation system causes flames to spread and progressively destroy flammable objects. In flood, buoyancy and drag forces cause rising water to submerge and displace objects. In wind, aerodynamic forces scatter lightweight

objects across the scene, complicating retrieval. We evaluate the performance of the baselines and MUSIX in fire and flood scenario.

Action space. HAZARD provides four high-level compressed actions derived from low-level primitives: **Walk To**, **Pick Up**, **Drop**, and **Explore**. The **Walk To** action leverages A*-based path planning to compress multiple navigation primitives into a single command, significantly reducing the frequency of LLM queries. As a result, the LLM-based decision maker only needs to select which object to walk to, while **Pick Up** and **Drop** handle object manipulation, and **Explore** enables the agent to survey its surroundings.

Task specifications. The objective for the agent is to rescue a predetermined set of target objects by transporting them to designated safe locations, such as a bag or a shopping cart. Performance is evaluated using three metrics. The rescued value rate (Value) captures the ratio of rescued object value to total target value. The averaged rescue step (Step) measures the efficiency of the rescue process. The averaged damaged rate (Damage) tracks the proportion of rescued objects that sustained damage from environmental hazards.

B Implementation Details

B.1 MuSix

Tables 8 and 9 summarize the training and inference hyperparameters for MUSIX. The knowledge states described in Section 3.3 are implemented as a test-time training (TTT) memory module that maintains per-group knowledge states $K^{(g)}$ as updatable parameter vectors. At every inference step, this module realizes both intra-scale adaptation (scale-dependent forgetting) and inter-scale adaptation (gated bidirectional transfer across neighboring groups). After each action execution, the agent computes a knowledge increment from the prediction error and distributes it across groups according to the meta-router weights, while the gated transfer matrices propagate relevant information between adjacent scales to maintain hierarchical coherence.

World Model Architectures. Our framework employs seven types of world model architectures, each designed to capture a distinct aspect of environment dynamics. All world models share the same LoRA adapter configuration for the base VLM backbone but differ in their auxiliary prediction heads and loss functions. Table 10 summarizes the auxiliary loss weight for each architecture.

Sensory. The Sensory world model encodes low-level affordance signals by predicting three binary properties from the VLM’s hidden state $\mathbf{h}_{\text{state}}$: whether the agent is near an interactable object, whether the target is pickable, and

Table 8: Training hyperparameters for MuSIX.

Hyperparameter	Value
<i>Architecture</i>	
Base Model	Qwen3-VL-4B-Instruct
World Model Types	PINN, Cognitive Map, Relational
Num World Models	15 (3 types $\times$ 5)
Routing Top- k	4
LoRA Rank / α / Dropout	16 / 32 / 0.05
Embedding Dim	384
Scale Dim	3
MC Samples	16
Target Encoder	CLIP-ViT-Large-Patch14
<i>Optimization</i>	
Learning Rate	1×10^{-4}
Optimizer	AdamW
Weight Decay	0.01
Batch Size	1
Num Epochs	1
LR Schedule	Cosine
Warmup Steps	200
Max Grad Norm	1.0
<i>Loss Weights</i>	
λ_{action} / λ_{aux} / $\lambda_{\text{transition}}$	0.1 / 0.1 / 1.0
Scale-Aware Loss (λ_S)	0.1
Variance reg. (λ_1) / Z reg. (λ_2)	0.01 / 0.01
Diversity Loss (λ)	0.01
Dormant Boost / Threshold / EMA	2.0 / 0.1 / 0.99
<i>Curriculum Phases</i>	
Phase A (router only)	steps 0–500
Phase B (+world models)	steps 500–1,500
Phase C (+diversity)	steps 1,500+

Table 9: Inference hyperparameters for MuSIX.

Hyperparameter Value	
Max New Tokens	50
Temperature	1.0
<i>Knowledge Adaptation</i>	
α_{max}	0.3
γ	1.0

whether a placement location is available. Each property is predicted by an independent two-layer MLP ($d_{\text{model}} \rightarrow d_{\text{model}}/4 \rightarrow 1$, GELU, dropout). The aux-

iliary loss sums three binary cross-entropy terms: $\mathcal{L}_{\text{Sensory}} = \text{BCE}(\hat{y}_{\text{near}}, y_{\text{near}}) + \text{BCE}(\hat{y}_{\text{pick}}, y_{\text{pick}}) + \text{BCE}(\hat{y}_{\text{place}}, y_{\text{place}})$.

Schema. The Schema world model captures action precondition rules by predicting whether an action is valid and, if not, the reason for failure. It consists of two heads: a validity head ($d_{\text{model}} \rightarrow d_{\text{model}}/4 \rightarrow 1$, GELU, dropout) producing a binary logit, and a failure-reason head ($d_{\text{model}} \rightarrow d_{\text{model}}/4 \rightarrow 5$) classifying among five failure categories. The auxiliary loss is $\mathcal{L}_{\text{Schema}} = \text{BCE}(\hat{v}, v) + \text{CE}(\hat{f}, f)$, where the failure-reason term is computed only over invalid-action samples.

Concept Bottleneck. The Concept Bottleneck world model predicts interpretable state concepts from $\mathbf{h}_{\text{state}}$. Three parallel two-layer MLPs predict the currently held object (34 classes), the open/closed state of receptacles (5-dim binary), and the current location (16 classes), respectively. The auxiliary loss is $\mathcal{L}_{\text{Concept}} = \text{CE}(\hat{h}, h) + \text{BCE}(\hat{o}, o) + \text{CE}(\hat{l}, l)$.

PINN. The Physics-Informed Neural Network (PINN) world model learns an explicit state transition function $g_{\theta}(\hat{s}_t, a_t) \rightarrow \hat{s}_{t+1}$. The environment state is decomposed into the same three components as the Concept Bottleneck (held object, open state, location), but instead of predicting them from $\mathbf{h}_{\text{state}}$, PINN concatenates one-hot encodings of the current state and action to form a 60-dimensional input. This input is passed through a two-layer MLP ($60 \rightarrow d_h/4 \rightarrow d_h/4$, GELU, dropout) followed by three parallel linear heads. The auxiliary loss has the same form: $\mathcal{L}_{\text{PINN}} = \text{CE}(\hat{h}, h) + \text{BCE}(\hat{o}, o) + \text{CE}(\hat{l}, l)$.

Cognitive Map. The Cognitive Map world model focuses on spatial structure by predicting location-level outcomes from $\mathbf{h}_{\text{state}}$. It consists of two parallel two-layer MLPs ($d_{\text{model}} \rightarrow d_{\text{model}}/4 \rightarrow 16$, GELU, dropout), one predicting the next location and the other the target location. The auxiliary loss is $\mathcal{L}_{\text{CM}} = \text{CE}(\hat{l}_{\text{next}}, l_{\text{next}}) + \text{CE}(\hat{l}_{\text{target}}, l_{\text{target}})$.

RBF Network. The Radial Basis Function (RBF) world model learns a contrastive embedding space that distinguishes successful from unsuccessful action executions. A two-layer MLP ($d_{\text{model}} \rightarrow d_{\text{model}}/4 \rightarrow 256$, GELU, dropout) projects $\mathbf{h}_{\text{state}}$ into an ℓ_2 -normalized 256-dimensional embedding. The auxiliary loss is an InfoNCE objective with temperature $\tau=0.07$, where positive pairs share a successful action execution and negative pairs share the same action type but differ in success: $\mathcal{L}_{\text{RBF}} = -\log \frac{\exp(\text{sim}(\mathbf{z}, \mathbf{z}^+)/\tau)}{\sum_j \exp(\text{sim}(\mathbf{z}, \mathbf{z}_j)/\tau)}$.

Relational Network. The Relational world model reasons over pairwise object interactions using a relation network architecture. Unlike the other world models, it operates on frozen CLIP embeddings rather than the VLM’s hidden state. For each pair of objects (i, j) in an episode (up to 49 objects), a pairwise relation function $g_{\theta}([\mathbf{e}_i; \mathbf{e}_j; \mathbf{a}_t; \mathbf{u}_t])$ produces a relation vector $\mathbf{r}_{ij} \in \mathbb{R}^{512}$, where $\mathbf{e}$ are CLIP object embeddings and $\mathbf{a}_t, \mathbf{u}_t$ are CLIP-encoded action and instruction

Table 10: Auxiliary loss weights (λ) for each world model architecture. All scheduled losses use linear warmup over the first 10% of training steps. The Relational Network loss is used directly as the transition prediction objective without additional weighting.

World Model	λ	Scheduling
Sensory	0.10	Linear warmup
Schema	0.20	Linear warmup
Concept Bottleneck	0.20	Linear warmup
PINN	0.05	Linear warmup
Cognitive Map	0.20	Linear warmup
RBF Network	0.10	Linear warmup
Relational Network	–	–

Table 11: Hyperparameters for LLM-Planner (inference only).

Hyperparameter	Value
Base Model	Qwen3-VL-4B-Instruct
Max New Tokens	1,024
Temperature	1.0
Top- p	0.95
RAG Top- k	5
Retrieval	BERT emb. + obs. rerank
Max Plan Length	No limit
Precision	bfloat16
Special Mechanism	Failed action filtering

strings. An attention pooling layer, with query derived from $[\mathbf{z}_t; \mathbf{a}_t; \mathbf{u}_t]$, aggregates over all valid pairs to produce a context vector $\mathbf{h}_t$. A prediction head then maps $[\mathbf{z}_t; \mathbf{a}_t; \mathbf{h}_t]$ to a predicted next-observation embedding $\hat{\mathbf{z}}_{t+1}$. The loss combines cosine and ℓ_2 objectives: $\mathcal{L}_{\text{RN}} = \alpha(1 - \cos(\hat{\mathbf{z}}, \mathbf{z}_{t+1})) + (1 - \alpha)\|\hat{\mathbf{z}} - \mathbf{z}_{t+1}\|_2^2$ with $\alpha = 0.5$. Since the Relational loss directly serves as the transition prediction objective, no additional auxiliary loss weighting is applied.

B.2 Baselines

LLM-Planner. LLM-Planner retrieves relevant demonstration snippets based on the current observation and instruction, and concatenates them with the planner prompt as few-shot examples for the model to propose the next action. We include this baseline to compare retrieval-augmented in-context learning directly with MUSIX. In our implementation, we replace the text-only LLM with a vision-language model to enable direct visual grounding from egocentric observations. Table 11 summarizes the inference hyperparameters.

SayCanPay. SayCanPay combines LLM-based task planning with a learned cost model to select grounded and cost-aware actions. It consists of three com-

Table 12: Training hyperparameters for SayCanPay.

Hyperparameter	Value
Base Model	Qwen3-VL-4B-Instruct
Model Type	Both (Say + Pay)
LoRA Rank / α	16 / 32
Learning Rate	2×10^{-4}
Batch Size	4
Num Epochs	1
Max Sequence Length	1,024
Discount Factor (Pay)	0.9
Quantization	Off

Table 13: Inference hyperparameters for SayCanPay.

Hyperparameter	Value
Temperature	1.0 (nucleus)
Top- p	0.6
Pay Top- k	30
Scoring	$\log P_{\text{say}} + \log P_{\text{pay}}$ (token-level)
Max Plan Length	Auto-regressive
Precision	bfloat16
Special Mechanism	Action blocking

ponents, where the Say model proposes candidate actions, the Can model provides environmental affordances, and the Pay model estimates execution costs via reinforcement learning. We include this baseline to compare learning-based action scoring with MUsix. In our implementation, we fine-tune Qwen3-VL-4B-Instruct with LoRA adapters for both the Say and Pay models jointly. Tables 12 and 13 summarize the training and inference hyperparameters, respectively.

FLARE. FLARE enhances standard retrieval-based planning by incorporating a dynamic replanning mechanism. By monitoring the current scene, it adjusts action sequences on the fly. Specifically, if an action fails due to a mismatched object name, the agent identifies and substitutes alternative targets using semantic similarity. We include this baseline to compare MUsix against a representative retrieval-based method with adaptive replanning capabilities. Table 14 summarizes the inference hyperparameters.

Conventional MoE. To isolate the contribution of our proposed expert routing strategy, we implement a conventional Mixture-of-Experts baseline that employs the identical backbone architecture as MUsix. This baseline attaches seven LoRA adapters as experts to Qwen3-VL-4B-Instruct, with a learned router that assigns input tokens to experts based on standard top- k gating. By comparing

Table 14: Hyperparameters for FLARE (inference only).

Hyperparameter	Value
Base Model	Qwen3-VL-4B-Instruct
Max New Tokens	1,024
Temperature	1.0
Top- p	0.95
RAG Top- k	5
Retrieval	Multi-Modal (BERT + CLIP, $w_l=0.5$, $w_e=0.5$)
Max Plan Length	10
Precision	bfloat16
Special Mechanism	Env. Adaptive Replanning (EAR)

Table 15: Training hyperparameters for Conventional MoE.

Hyperparameter	Value
<i>Architecture</i>	
Base Model	Qwen3-VL-4B-Instruct
Num Experts	7
Top- k	3
LoRA Rank / α	16 / 32
Expert Dropout	0.2
Router d_{model}	512
Router Num Heads / Aspects	8 / 8
Router FFN Dim	1,024
Router Dropout	0.1
<i>Optimization</i>	
Learning Rate (LoRA)	2×10^{-4}
Learning Rate (Router)	1×10^{-3}
Batch Size	1
Gradient Accum. Steps	32 (eff. batch = 32)
Num Epochs	1
Warmup Ratio	0.05
Max Grad Norm	1.0
Max Seq Length	4,096
Image Size	256×256
Quantization	4-bit
<i>Loss Weights</i>	
LM Loss	1.0
Router LM Loss	1.0
Balance Loss	$0.1 \rightarrow 0.05$ (annealing)

against this baseline, we evaluate whether the structured expert decomposition in MUSIX provided advantages over a generic MoE formulation. Tables 15 and 16 summarize the training and inference hyperparameters, respectively.

Table 16: Inference hyperparameters for Conventional MoE.

Hyperparameter	Value
Max New Tokens	1,024
Temperature	1.0
Top- p	0.95
Max Plan Length	No limit
Image Resize	256×256
Precision	bfloat16
Scoring	Router $\rightarrow$ LoRA expert $\rightarrow$ VLM gen.

Table 17: Experiential distance analysis.

Distance metric	SR ($\uparrow$)	PS ($\downarrow$)
Cosine similarity	35.11 ± 3.00	10.30 ± 1.12
KL divergence	38.22 ± 5.70	7.44 ± 0.60
Mahalanobis (ours)	40.44 ± 1.27	7.25 ± 0.50

Table 18: Compute cost comparison.

	LLM-Planner	SayCanPay	Conv. MoE	MuSix
Lat. (s)	2.00	1.86	2.52	3.52
Mem. (GB)	8.30	8.43	8.43	9.60

C Additional Experiments

Experiential distance analysis. We replace the Mahalanobis distance with cosine similarity and KL divergence. For KL, we fit the embedding distribution of experience $\mathbf{E}_{<t}$ as $\mathcal{N}(\boldsymbol{\mu}_E, \boldsymbol{\Sigma}_E)$ and treat the current observation as $\mathcal{N}(\boldsymbol{\phi}(o_t), \mathbf{I})$, using $\text{KL}(\mathcal{N}(\boldsymbol{\phi}(o_t), \mathbf{I}) \parallel \mathcal{N}(\boldsymbol{\mu}_E, \boldsymbol{\Sigma}_E))$.

As shown in Table 17, Mahalanobis is **+2.22%p** ahead of the next best. Cosine is unfavorable for $\mathcal{L}_S^{(1)}$ alignment in three ways: it yields a non-convex objective in $\boldsymbol{\phi}(o_t)$, ignores embedding magnitude, and cannot exploit the experience covariance $\boldsymbol{\Sigma}_E$ (all directions are treated equally). Under this direction, KL contains the (squared) Mahalanobis distance as its mean-difference term, which explains why it tracks Mahalanobis more closely than cosine.

Compute cost. We profile MuSix against baselines on EB-Habitat (RTX Pro 6000 Blackwell, batch=1). The two-stage routing and 16-sample MC integration add **39.7%** latency over conventional MoE (Table 18), in exchange for the reported accuracy gains.