BEYOND NEEDLE(S) IN THE EMBODIED HAYSTACK: ENVIRONMENT, ARCHITECTURE, AND TRAINING CONSIDERATIONS FOR LONG CONTEXT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce ∞-THOR, a new framework for long-horizon embodied tasks that advances long-context understanding in embodied AI. ∞-THOR provides: (1) a generation framework for synthesizing scalable, reproducible, and unlimited long-horizon trajectories; (2) a novel QA task, Needle(s) in the Embodied Haystack, where scattered clues across extended trajectories test agents' long-context reasoning ability; and (3) a long-horizon dataset and benchmark suite featuring complex tasks that span hundreds of environment steps, each paired with ground-truth action sequences. To enable this capability, we explore architectural adaptations, including interleaved Goal-State-Action and Memory-Augmented Goal-State modeling, alongside training strategies—context extension techniques and Context Parallelism—to equip VLM-based agents for extreme long-context reasoning and interaction. Experiments highlight the challenges of our benchmark and provide insights into training strategies and model behaviors under long-horizon conditions. Furthermore, we demonstrate sim-to-real transfer: our QA dataset alone improves photorealistic benchmark performance by up to +11.2%, and we successfully integrate the framework with low-level manipulation controls. Our work provides a foundation for the next generation of embodied AI systems capable of robust, long-term reasoning and planning.

1 INTRODUCTION

Real-world embodied reasoning is a sequential decision-making problem requiring long-horizon planning, where task success depends on both memorizing and reasoning over multiple events that occur far apart in time. Using vision-language models (VLM) as policies for such tasks requires surpassing the key challenge of *long-context* reasoning. We seek to answer questions pertaining to what design choices matter in terms of environments, model architectures, and training methods when using VLMs for long-horizon embodied tasks. To this end, we develop a new framework for long-horizon tasks designed to push the boundaries of long-context understanding in embodied AI.

We introduce ∞-THOR, a new framework for generation, training, and evaluation of long-horizon embodied tasks. Our benchmark uniquely features tasks with a synthetic final goal, which involves multiple objects that appear at distant time steps, requiring multi-step reasoning across over hundreds of steps. As illustrated in Figure 1, an agent might observe a tomato at $t = 16$ and a counter top much later at $t = 560$; the final task given at $t = 670$ requires the agent to retrieve the tomato and place it on the counter. This setup highlights the challenge of long-horizon dependency, where key objects and locations must be remembered and acted upon after hundreds of steps.

This long-horizon setup introduces a new challenging task, Needle(s) in the Embodied Haystack (NiEH). Unlike the standard Needle in a Haystack task (Liu et al., 2024), which focuses on recalling a single clue in text, NiEH poses two main challenges: (1) multiple scattered clues (*Needles*) and (2) multi-modal inputs that combine visual and linguistic observations from the environment (*Embodiment*). This task is designed to evaluate the agent's ability to recall and reason about previously encountered environmental details, such as identifying objects and recalling performed actions.

Going beyond static evaluations such as NiEH, ∞-THOR also provides an interactive evaluation, allowing agents to execute policies and complete long-horizon tasks within a dynamic environment.

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

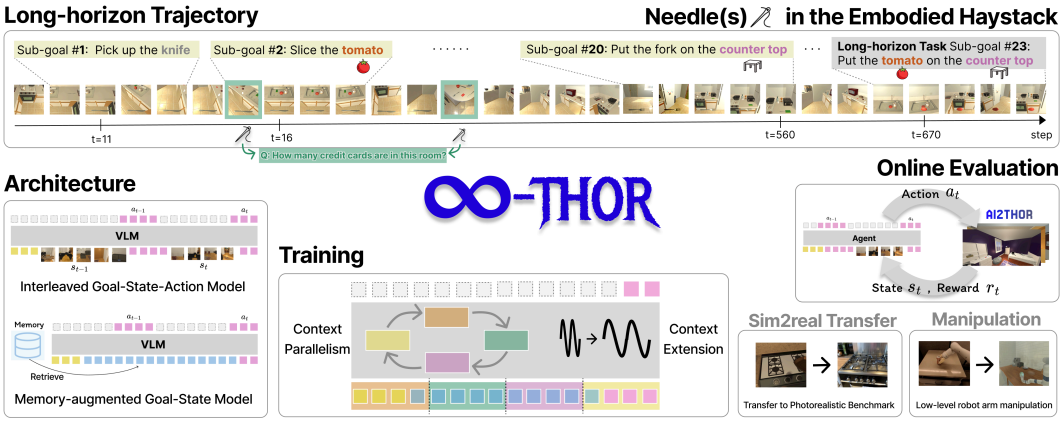


Figure 1: Overview of ∞-THOR. **Environment:** We introduce a framework for generating long-horizon trajectories, the *Needle(s) in the Embodied Haystack* benchmark to test reasoning over scattered clues, and interactive online evaluation. **Architecture:** We investigate VLM adaptations as policy models, including Interleaved Goal-State-Action and Memory-augmented models. **Training:** We incorporate Context Parallelism and Context Extension for efficient long-context learning. In experiments, we demonstrate Sim2Real transfer to photorealistic benchmark tasks and integration with low-level manipulation controls.

To support this, we release a trajectory dataset for training, with episodes over 400 steps in the training set and more than 600 steps in the dev and test sets. These trajectories can be used for imitation learning, and our experiments show that access to longer context during training leads to significant performance gains, highlighting the importance of our dataset for long-context embodied reasoning.

With the environment established, we investigate various architectural considerations for embodied agents to operate under extreme sequence lengths. We explore two complementary approaches: Interleaved Goal-State-Action model, which directly encodes the full perception–action history as a single multimodal sequence, and Memory-Augmented Goal-State model, which replaces the historical trajectory with text summaries or retrieved visual memories to provide long-term contextual grounding. Moreover, since standard VLMs are constrained by fixed context windows and cannot natively handle inputs exceeding 1M tokens, we evaluate long-context extension techniques, including linear interpolation, dynamic scaling, YaRN, and LongRoPE (Chen et al., 2023; Ding et al., 2024; Peng et al., 2024). Lastly, to support practical training and inference at these extended horizons, we incorporate Context Parallelism via Ring Attention (Liu et al., 2023a), enabling efficient fine-tuning on ultra-long sequences and further improving the model's ability to reason over extended temporal contexts.

Finally, we provide comprehensive experiments and analyses, demonstrating both the challenges posed by our benchmark and the behavior of baseline models under long-horizon settings. Furthermore, we validate the real-world applicability of our framework through *Sim2Real* transfer, showing that our QA dataset improves performance on photorealistic benchmarks. We also demonstrate the framework's versatility by integrating it with ManipulaTHOR Ehsani et al. (2021) to enable low-level robot arm manipulation.

To summarize, our contributions are: **(1)** We introduce ∞-THOR, a new framework for generating, training, and evaluating long-horizon embodied tasks, featuring synthetic goals that require multi-step reasoning across hundreds of steps. **(2)** We propose a novel embodied QA task, Needle(s) in the Embodied Haystack, requiring agents to recall and reason over multiple scattered clues across extended trajectories. **(3)** We release a large-scale trajectory dataset and an interactive evaluation environment to support both offline imitation learning and online policy execution. **(4)** We analyze architectural adaptations (Interleaved vs. Memory-Augmented) and training strategies (Context Extension, Context Parallelism) for handling long-context inputs. **(5)** We present empirical results demonstrating the benefits of our framework, including successful Sim2Real transfer and integration with low-level manipulation controls.

2 Related Work

Table 1: Comparison of benchmarks. We use Short (< 50 steps), Medium (50-300 steps), and Long (> 300 steps) to describe task horizon, reflecting the approximate number of environment steps required to complete a task in each benchmark. (Inter. w/ env: Interaction with the environment, Mod: Modality, GT: GT actions; single/multi in the QA set column denotes single- and multi-evidence question type. * indicates the number of annotations newly collected in that work.)

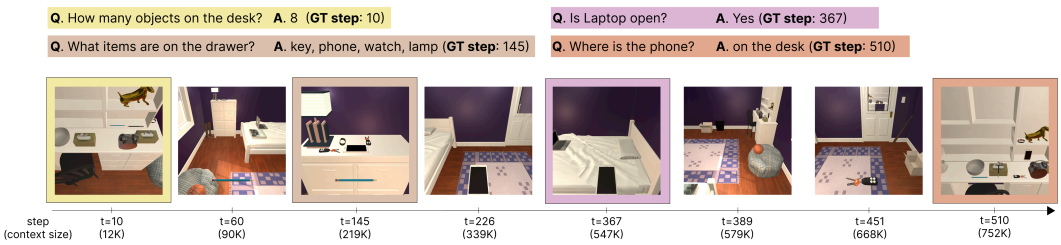
Benchmark / Platform	Task Horizon	Inter. w/ env	Dataset			QA set	
			Mod	Avg steps	GT	single	multi
ProcTHOR (Deitke et al., 2022)	✗	✓	✗	✗	✗	✗	✗
MineDojo (Fan et al., 2022)	Long	✓	✗	✗	✗	✗	✗
Habitat 3.0 (Puig et al., 2023)	Long	✓	✗	✗	✗	✗	✗
VirtualHome (Puig et al., 2018)	Short	✓	multi	11.6	✓	✗	✗
EQA (Das et al., 2018)	✗	✗	✗	✗	✗	✓	✗
ALFRED (Shridhar et al., 2020)	Medium	✓	multi	50	✓	✗	✗
ALFWorld (Shridhar et al., 2021)	Medium	✓	text	50	✓	✗	✗
BEHAVIOR-100 (Srivastava et al., 2021)	Med/Long	✓	✗	✗	✗	✗	✗
EAI (Li et al., 2024b)	Med/Long	✓	✓	14.6*	✓	✗	✗
BALROG (Paglieri et al., 2025)	Long	✗	✗	✗	✗	✗	✗
MM-EGO (Ye et al., 2025)	✗	✗	✗	✗	✗	✓	✗
∞-THOR	∞	✓	multi	627	✓	✓	✓

Long-horizon Planning in Virtual Environments. AI2THOR (Kolve et al., 2017) provides interactive indoor environments widely used for embodied reasoning research, while ProcTHOR (Deitke et al., 2022) extends these capabilities by procedurally generating scalable environments, potentially facilitating longer trajectories. MineDojo (Fan et al., 2022) offers an open-ended platform within Minecraft, explicitly geared toward tasks requiring extensive long-term planning. Additionally, platforms such as VirtualHome (Puig et al., 2018) and Habitat 3.0 (Puig et al., 2023) have demonstrated suitability for tasks involving long-term interactions and complex activity sequences. However, all of these platforms only provide environments and do not include standardized datasets or benchmark suites to support training and evaluation for long-horizon embodied tasks.

Embodied QA and Multimodal Needle in the Haystack Tasks. Embodied QA tasks, such as EmbodiedQA (Das et al., 2018) and MM-EGO (Ye et al., 2025), require agents to answer questions grounded in visual observations with spatial and temporal reasoning, but without active environment interaction during evaluation. Our NiEH task is also related to multimodal Needle in a Haystack (NiH) problems. While early NiH focused on textual recall in long contexts (Liu et al., 2024), recent multimodal extensions add visual inputs (Wang et al., 2024; 2025), though they remain limited to shorter contexts (up to 72K tokens) and lack embodied reasoning or temporal dependencies.

Datasets and Benchmarks for Long-horizon Embodied Tasks. Recent efforts have pushed toward long-horizon embodied tasks, where agents must complete multi-step goals with extended temporal dependencies. While ALFRED (Shridhar et al., 2020) and ALFWorld (Shridhar et al., 2021) introduced multi-step instruction-following tasks with action annotations and textual grounding, their task horizons are relatively short, typically under 50 steps. BEHAVIOR-100 (Srivastava et al., 2021) evaluates agent generalization on household activities, some of which require prolonged engagement, but mainly focus on single task. BALROG (Paglieri et al., 2025) is a benchmark for testing the agentic capabilities of long-context LLMs, but its scope is limited to games. Recently, EAI (Li et al., 2024b) proposed a generalized interface to evaluate LLMs for embodied decision making, while our framework focuses on multimodal online evaluation with real-time rewards and a new long-horizon reasoning task (NiEH).

Long-context Benchmarks. Outside embodied AI, general benchmarks have addressed challenges in long-context reasoning. Benchmarks, such as LongBench (Bai et al., 2024) and RULER (Hsieh et al., 2024), focus on retrieval or summarization tasks. GSM-∞ (Zhou et al., 2025) extends GSM-8K (Cobbe et al., 2021) to assess mathematical reasoning over extremely long textual inputs. More recently, LMAct (Ruoss et al., 2025) proposed a benchmark for evaluating frontier models' long-context multimodal decision-making on interactive game-based tasks, with up to 1M context lengths.



(a) Needle in the Embodied Haystack: Single-evidence question types.



(b) Needles in the Embodied Haystack: Multi-evidence question types.

Figure 2: Example of N(s)iEH task and Ground-truth steps.

3 ∞-THOR: AN ENVIRONMENT FOR GENERATING, TRAINING, AND EVALUATING LONG-HORIZON EMBODIED TASKS

∞-THOR features with a generation framework for synthesizing long trajectories to train and evaluate AI agents in long-horizon embodied tasks. We build ∞-THOR upon AI2-THOR (Kolve et al., 2017) simulator, an interactive 3D environment for embodied AI research that supports diverse scenes, objects, and agent actions. ∞-THOR enables the creation of unlimited trajectories with arbitrary length, and provides an evaluation setup where agents can interact dynamically with the environment during both training and testing. This supports both offline learning by producing large-scale datasets, and online learning through direct agent-environment interaction.

Each trajectory generated by ∞-THOR consists of multiple task goals, such as "Put a clean sponge on a metal rack" and "Pick up the apple and place it on the microwave", requiring grounded understanding and action to achieve the goal. At the end of each trajectory, the agent is assigned a synthetic long-horizon task that requires reasoning over entities encountered at distant time steps. For the example in Figure 1, the long-horizon task (Sub-goal #23) at step $t = 670$, "Put the tomato on the counter top", depends on observations made far earlier: the tomato at $t = 16$ and the counter top at $t = 560$.

Our generation framework can generate unlimited tasks, the trajectories can be exceptionally long, exceeding 1M context tokens or beyond when the trajectory is processed with LLMs. Successfully completing this task requires the agent to (1) memorize and integrate key environmental information over hundreds of steps, and (2) plan actions based on dependencies that are separated in time, demonstrating the need for long-context reasoning and robust spatio-temporal memory.

3.1 STATIC EVALUATION: NEEDLE(S) IN THE EMBODIED HAYSTACK

We first introduce a novel task in the form of a static evaluation: Needle(s) in the Embodied Haystack (NiEH). NiEH is designed to evaluate an agent's ability to recall and reason about environmental states encountered throughout a trajectory. Unlike traditional embodied QA tasks that focus primarily on visual understanding of a single image, NiEH emphasizes reasoning about environmental changes over time, requiring agents to interpret and integrate sequences of multimodal observations.

Figure 2 presents examples of the two NiEH task types. In the single-evidence setting, a question is answerable based on a single observation step; in the multi-evidence setting, multiple temporally distant steps must be combined to answer the question. The NiEH testset includes diverse question types, such as binary ("yes" or "no"), "what"-, "where"-, and "how many"-style questions. These questions span a broad range of difficulty, from simple memory recall (similar to the Needle in a Haystack paradigm (Liu et al., 2024)) to complex queries that requiring multi-step reasoning across temporally and spatially distributed evidence.

Testset Construction. We first replay the generated trajectories to collect egocentric views and record object interactions directly from the simulator's metadata. Using rule-based templates, we generate QA pairs (e.g., "Q. What object did you slice? A. `{object}`") and sample them based on frequency to ensure diversity. To ensure valid answerability, we cross-validate questions against ground-truth images using an ensemble of four multimodal LLMs: LLaVA-OneVision, Qwen2.5-VL, Deepseek-VL, and Pixtral (Li et al., 2024a; Bai et al., 2025; Lu et al., 2024; Agrawal et al., 2024). We filter out questions that none of these models can successfully answer. Full details on generation rules and validation are provided in Appendix A.1.

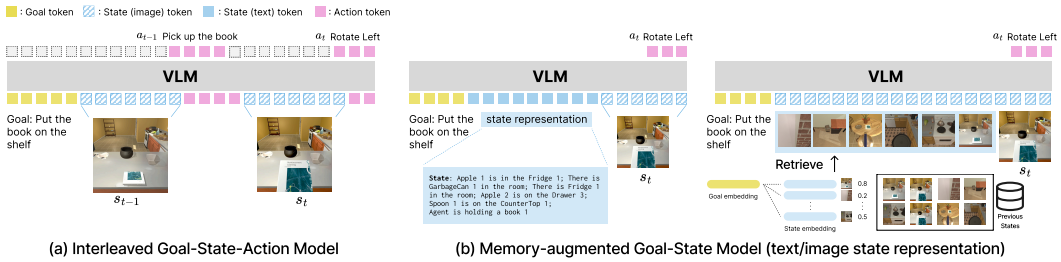
Challenges in NiEH. NiEH introduces two key challenges for current models. First, questions require reasoning over multiple temporally distant events. As shown in Figure 2(b), the agent moves a `dish sponge` from the garbage bin at $t = 24$, then to the counter top, and later places it into a drawer at $t = 751$. A question such as *"Where was the dish sponge before you put it on the counter top?"* requires the model to recall and chain together multiple actions and locations across hundreds of steps. Second, some questions demand aggregating sparse and temporally scattered evidence from long trajectories. In the second example in Figure 2(b), answering *"How many times did you move the credit card?"* requires the model to track and count all relevant actions occurring from the beginning to the end of the episode. These tasks necessitate robust long-horizon reasoning across both time and modalities in complex embodied environments.

Table 2: Dataset statistics. Token lengths are calculated using the LLaVA-OneVision tokenizer. See Appendix A.3 for detailed statistics.

NiEH testset		Single-clue	Multi-clue
# of question-answer pair		829	474
Trajectory	**Train**	**Dev**	**Test**
# trajectory	2,456	125	225
# avg/max subgoals	14/30	16/24	18/33
# avg/max steps	405/654	613/890	627/952
# avg token length	602K	880K	912K
# max token length	954K	1.2M	1.3M

3.2 CONSTRUCTING LONG-HORIZON TRAJECTORIES FOR INTERACTIVE EVALUATIONS

With ∞-THOR, we synthesize long-horizon datasets by sequentially concatenating multiple single-task demonstrations using a PDDL-based planner (Kolve et al., 2017; McDermott et al., 1998). For each trajectory, we sample predefined task templates and objects, generating ground-truth action sequences that are retained only if successfully simulated to ensure reproducibility. Crucially, to enforce long-term temporal dependencies, the final task exclusively involves objects seen during the early 20% and final 20% of the episode. This requires the agent to jointly reference temporally distant objects to complete the task. As a result, we obtain a comprehensive dataset of long-horizon trajectories (Table 2). Further details and a generation pseudo-algorithm are provided in Appendix A.2.



(a) Interleaved Goal-State-Action Model (b) Memory-augmented Goal-State Model (text/image state representation)

Figure 3: **Overview of Policy Architectures.** (a) Interleaved Goal-State-Action Model: Encodes the full history as a single continuous sequence of goal, state, and action tokens. (b) Memory-augmented Goal-State Model: Retrieves relevant past information to provide long-term context without processing the entire trajectory.

4 Architectures and Training Considerations for Long-Horizon Embodied Reasoning

Long-horizon embodied tasks pose significant challenges due to the need to interpret multimodal inputs (vision, language) and produce coherent sequences of actions over hundreds of interactive steps. However, existing approaches either use separate encoders for vision, language, and action modules (Shridhar et al., 2020; Hafner et al., 2025), or focus on short-horizon in constrained environments (Brohan et al., 2023), where decisions depend only on the most recent observation and a single instruction. While recent VLMs Liu et al. (2023b); Lu et al. (2024); Bai et al. (2025) show strong multimodal reasoning abilities, these models are not directly suited to long-horizon embodied tasks, which are interaction-dominant, involving hundreds of egocentric frames and continuous vision-language-action sequences. Moreover, many state-of-the-art proprietary models are only accessible via paid, rate-limited APIs, which makes large-scale, real-time experimentation in embodied settings more difficult and costly in practice Li et al. (2024b). In this work, we explore two distinct policy architectures for leveraging a VLM as a unified embodied agent: (1) Interleaved Goal-State-Action Model and (2) Memory-Augmented Goal-State Model.

Interleaved Goal-State-Action Model treats the entire perception–action stream as a single unified sequence. At each timestep, the input consists of the task goal followed by all past visual observations and actions in an interleaved order $(g_0, s_0, a_0, \ldots, s_t, a_t)$. This design enables the VLM backbone to jointly reason over multimodal information and autoregressively predict the next action conditioned on the full trajectory history. By leveraging interleaved modeling, our architecture supports coherent decision-making over temporally distant information while maintaining grounded behavior in dynamic settings.

Memory-Augmented Goal-State Model incorporates compact history representations into the input context. Instead of supplying the full trajectory, the model conditions on either (1) textual summaries of past states and environment information (Figure 3-(b)-left), or (2) retrieved relevant historical images selected via embedding similarity (Figure 3-(b)-right). These memory tokens provide distilled representations of long-term context, allowing the model to focus on the current observation while retaining access to key historical information. By combining memory tokens with the current state, the model can maintain semantic grounding over long horizons without exceeding context-length constraints.

Context Extension and Context Parallelism. Given the limitations in context length of most VLMs, using off-the-shelf models is insufficient for processing long inputs such as those exceeding 1M tokens, particularly in our interleaved Goal-State-Action setting. We explore various long-context extension techniques that allows the model to generalize to longer input sequences without retraining from scratch. Specifically, we consider: **Linear Interpolation** Chen et al. (2023): Rescales input positions to fit within the pretrained RoPE range by linearly interpolating positional indices; **Dynamic Scaling** Chen et al. (2023): Adapts RoPE frequencies at runtime based on the input sequence length, using a linear rescaling to maintain consistent positional encoding behavior across varying lengths; **YaRN** Peng et al. (2024): Dynamically interpolates attention frequencies during inference, balancing between pretrained and extrapolated positional regimes;

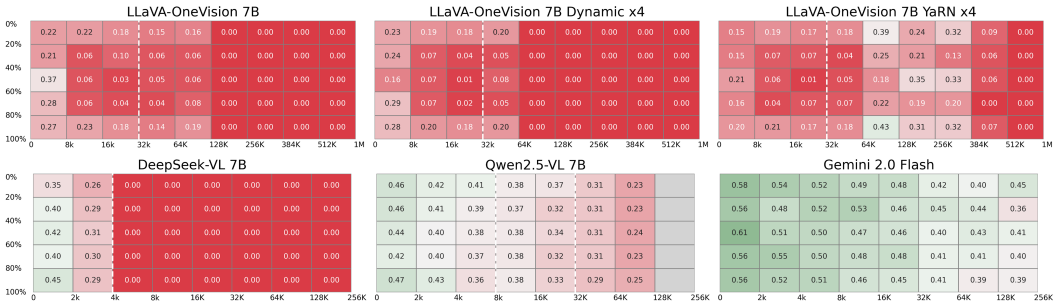


Figure 4: **Results of Needle(s) in the Embodied Haystack.** The x-axis denotes the input context length (tokens), and the y-axis denotes the needle depth (percentage position of the evidence within the trajectory). The white dashed line denotes the maximum pre-training context length of each model. Qwen2.5-VL was initially pre-trained with an 8K token context window and incrementally scaled up to 32K tokens in subsequent stages (Bai et al., 2025). The gray area indicates contexts not applicable (N/A) due to the model's smaller image token size which limits sequences to under 128K tokens. Context Parallelism is applied to all experiments with the context sizes over 384K. Full results are provided in Appendix C.1.

Table 3: **Architectural Comparison.** QA performance on N(s)iEH across different architectures in a setting where the *full trajectory context is available*. LLaVA-OV model fails when provided with the full trajectory as input (>1M tokens). GPT-4o is used as an LLM judge for evaluation. For Memory-Augmented approaches, we use a CLIP encoder (Radford et al., 2021) to compute embedding similarity between goal descriptions and historical state observations, selecting the Top-10 and Top-20 most relevant images.

	NiEH (Single-evidence)			**NsiEH** (Multi-evidence)		
	LLaVA-OV	Qwen2.5-VL	Gemini 2.0 Flash	LLaVA-OV	Qwen2.5-VL	Gemini 2.0 Flash
(a) Full trajectory (Image-only)	0.0	44.67	35.44	0.0	19.34	37.08
(b) Interleaved Goal-State-Action	0.0	51.64	**83.04**	0.0	43.24	**62.72**
(c) Memory-Augmented (Text)	51.33	54.38	56.21	53.83	51.33	40.99
(d) Memory-Augmented (Image, Top-10)	35.80	33.99	25.18	22.15	18.63	21.18
(e) Memory-Augmented (Image, Top-20)	37.36	42.68	28.16	23.80	21.46	30.25
(f) Memory-Augmented (Text + Image, Top-20)	35.13	55.73	32.05	40.76	53.98	34.24

LongRoPE Ding et al. (2024): Augments RoPE with specially designed extrapolation functions, enabling robust generalization to long sequences without degrading attention quality. We apply these techniques during fine-tuning, at inference time, or both.

We also incorporate Context Parallelism built upon Ring Attention (Liu et al., 2023a), which cyclically exchanges key-value shards across devices to compute full attention with significantly reduced memory overhead. We apply these techniques during fine-tuning, inference, or both as essential tools for handling long-context inputs and evaluate how effectively they improve practical performance when interleaved trajectories become substantially long.

5 Experiments

5.1 Static Evaluation: Needle(s) in the Embodied Haystack

We first evaluate model performance on the Needle in the Embodied Haystack (NiEH) and Needle**s** in the Embodied Haystack (NsiEH) tasks, which test an agent's ability to retrieve and reason over sparse evidence scattered throughout long embodied trajectories.

Building a Embodied Haystack. Unlike the traditional Needle in the Haystack setup, which inserts a target sentence into a long text corpus like a book, we use the entire embodied trajectory as the input context. To simulate different reasoning depths, we crop the input sequence either from the beginning or the end based on the GT image's position. In the NsiEH task, where multiple evidences are scattered throughout the trajectory, we fill the context with intermediate steps in between the GT steps keeping their temporal order.

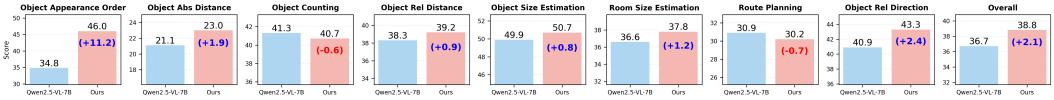


Figure 5: **Sim2Real Transfer on VSI-Bench.** We compare Qwen2.5-VL 7B (Base) against the same model fine-tuned on 226K synthetic NiEH QA pairs. Our model achieves a +11.2% in *Object Appearance Order* and a +2.1% improvement overall, demonstrating effective transfer to photorealistic environments.

Results. Figure 4 (top row) shows the performance of LLaVA-OneVision (OV) 7B across different context extension methods. While Dynamic scaling struggles with contexts beyond 64K tokens, YaRN x4 improves the baseline, handling sequences up to 384K tokens. The second row compares different VLMs: DeepSeek-VL, Qwen2.5-VL, and Gemini 2.0 Flash. Each model processes images into tokens differently–DeepSeek (576 tokens/image), Qwen2.5-VL (121 tokens/image), and Gemini 2.0 Flash (258 tokens/image)–which consequently impacts maximum context lengths when transforming N(s)iEH sequences into tokenized inputs (DeepSeek-VL: 512K, Qwen2.5-VL: 128K, Gemini 2.0 Flash: 256K). DeepSeek-VL fails immediately beyond its 4K pre-training limit. Qwen2.5-VL outperforms DeepSeek but shows clear degradation beyond its 32K pre-training limit. Gemini 2.0 Flash performs robustly up to 32K tokens but performance drops below 40% at 256K, indicating that complex long-range multimodal reasoning remains a challenge even for state-of-the-art models. See Appendix C.1 for more results.

Architectural Comparison. Table 3 presents N(s)iEH task performance across different architectures in a setting where *full trajectory context is available*. LLaVA-OV fails to process these full trajectories (>1M tokens), exposing an inherent limitation in its ability to handle long contexts. In contrast, Gemini 2.0 Flash effectively leverages this temporal context. For Memory-Augmented approaches, textual memory demonstrates significantly higher performance than image-based memory, as textual state representations provide denser information within a shorter context size. Notably, we observe that Gemini 2.0 Flash tends to hallucinate object locations in image-only settings when language grounding is not provided (Table 3-(a), (d), and (e)).

Single vs Multi-evidence Reasoning. Comparing NiEH to the more challenging NsiEH task, we observe a significant performance drop in the multi-evidence setting. This is especially pronounced for questions involving sparse or distant evidence (e.g., *"Where was the Mug before you put it on the CounterTop?"*) or questions requiring the aggregation of multiple evidences (e.g., *"How many times did you move the Apple?"*), as shown in Figure 2(b). These results demonstrate that our NiEH and NsiEH tasks pose a substantial challenge to current long-context models and success requires both fine-grained temporal memory and multi-evidence reasoning across extended interactions.

5.2 Online Evaluation: Long-horizon Tasks

We conduct an *online* evaluation using the AI2THOR simulator. We evaluate models on the final long-horizon task of each test trajectory (e.g., Sub-goal #23 in Figure 1), which requires the agent to utilize history from the preceding hundreds of steps. We fine-tune both LLaVA-OV and Qwen2.5-VL on our training set. Detailed training specifics are available in Appendix B.

Results. Table 4 presents the success rates from our interactive online evaluation. LLaVA-OV failed to complete any long-horizon tasks in this setting (>1M tokens); therefore, we report results only for the Qwen2.5-VL model in Table 4. We observe that interleaved models consistently outperform Memory-Augmented approaches, suggesting that processing the continuous history is more effective than retrieval-based methods for grounded reasoning. Training with longer contexts leads to significant gains: the 128K context model achieves a 9.2% success rate, confirming that access to extended history is crucial for long-horizon performance. Incorporating long-context extension techniques further boosts performance. YaRN (x4) achieves the highest total success rate of 11.5%, providing a +2.3% gain over the base interleaved model. However, the `Put` task remains unsolved across all methods, as `Put` task comes after `Go to` and `Pick up`, and the agent rarely reaches this final stage due to failures in the earlier steps. Evaluation results for full-episode execution are available in Appendix C.2. Overall, these results demonstrate that online evaluation for long-horizon tasks remains a significant open challenge.

5.3 DEMONSTRATIONS: SIM2REAL TRANSFER AND LOW-LEVEL CONTROL

Finally, we demonstrate the versatility of ∞-THOR through Sim2Real transfer and integration with low-level control policies. 1) **Sim2Real Transfer.** We show the transferability of our framework. We used the NiEH pipeline to generate 226K synthetic QA pairs from the training set and fine-tuned the Qwen2.5-VL model. We evaluated the model on VSI-Bench Yang et al. (2024), a *photorealistic* benchmark for visual-spatial reasoning. As shown in Figure 5, our model achieves +11.2% improvement on the *Object Appearance Order* task, confirming that temporal reasoning skills learned in simulation transfer to real-world images. Overall, we observe a +2.1% performance gain with negligible degradation in other spatial tasks. 2) **Low-level Manipulation.** We integrated ∞-THOR with ManipulaTHOR Ehsani et al. (2021) to enable fine-grained, low-level robot arm control. Table 9 reports the baseline success rates on these tasks using OpenVLA-7B (Kim et al., 2024) and SpatialVLA-4B (Qu et al., 2025). We provide further results and analyses in Appendix C.3.

Table 4: **Online Evaluation.** Success Rates (%) for the long-horizon tasks. 32K and 128K denote the context size used during fine-tuning.

Method	`Go to`	`Pick up`	`Put`	**Total**
Memory-Augmented (Text)	10.8	1.5	0.0	6.1
Memory-Augmented (Image, Top-20)	6.2	0.0	0.0	3.1
Interleaved Goal-State-Action (32K)	11.3	1.5	0.0	6.4
Interleaved Goal-State-Action (128K)	18.5	2.3	0.0	9.2
+ Dynamic Scaling	20.0	2.3	0.0	9.9
+ YaRN (x4)	**23.1**	**2.6**	0.0	**11.5**

6 CONCLUSION

We presented ∞-THOR, a framework for long-horizon embodied tasks designed to advance long-context understanding in embodied AI. Our framework enables scalable synthesis of complex trajectories paired with ground-truth action sequences, supporting both offline training and online interaction. Additionally, we introduced a novel embodied QA benchmark, Needle(s) in the Embodied Haystack, that challenges agents to reason over sparse, temporally distant visual evidence. To equip models for this setting, we explored policy architectures including interleaved Goal-State-Action and Memory-Augmented modelings, along with context extension techniques and Context Parallelism. Our experiments demonstrate that while inference-time extension provides gains, access to longer training contexts is critical for robust performance. Furthermore, we validated the versatility of ∞-THOR through successful Sim2Real transfer to photorealistic benchmarks and integration with low-level manipulation controls. Despite these advances, state-of-the-art VLMs still struggle over extreme context lengths, highlighting that long-horizon reasoning remains an open challenge. We hope our framework encourages further research into the next generation of embodied agents capable of robust long-term planning.

AI USE STATEMENT

We used generative AI models for dataset screening and automated evaluation. Long-horizon trajectories were generated with a classical PDDL-based planner and validated through simulator execution; question–answer pairs and their supporting time steps were derived from simulator metadata and replay logs using handcrafted rules (Appendix A). For QA screening, LLaVA-OneVision 7B, Qwen2.5-VL 7B, DeepSeek-VL 7B, and Pixtral 12B were prompted with the annotated ground-truth images to assess whether the questions could be answered from the visual evidence. Their responses informed dataset filtering, and multi-evidence questions were also inspected manually for answerability, as described in Appendix A.1. GPT-4o was used as an automated judge for the architectural comparison in Table 3. The generative models evaluated as agents for question answering and action prediction are described in the experimental sections.

We also used AI assistants to assist with literature searches, LaTeX editing, figure-preparation code, and drafting manuscript.

Reproducibility Statement

To ensure the full reproducibility of our study, we commit to the comprehensive release of our dataset and experimental artifacts. Upon publication, we will provide public access to the dataset, including simulated episodes, trajectories, and QA pairs. We will also provide the full source code and model weights used to generate our results. All resources will be hosted in a public repository to ensure the community can verify and reproduce our results.

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024. URL https://arxiv.org/abs/2410.07073.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023. URL https://arxiv.org/abs/2306.15595.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation, 2022. URL https://arxiv.org/abs/2206.06994.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024. URL https://arxiv.org/abs/2402.13753.

Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4495–4504, 2021. URL https://api.semanticscholar.org/CorpusID:233346916.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge, 2022. URL https://arxiv.org/abs/2206.08853.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. URL https://arxiv.org/abs/2509.24527.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024. URL https://arxiv.org/abs/2308.16137.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism, 2019. URL https://arxiv.org/abs/1811.06965.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. In *NeurIPS 2024*, 2024b.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023b.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024. URL https://arxiv.org/abs/2403.05525.

Drew McDermott, Malik Ghallab, Adele E. Howe, Craig A. Knoblock, Ashwin Ram, Manuela M. Veloso, Daniel S. Weld, and David E. Wilkins. Pddl-the planning domain definition language. 1998. URL https://api.semanticscholar.org/CorpusID:59656859.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. Balrog: Benchmarking agentic llm and vlm reasoning on games, 2025. URL https://arxiv.org/abs/2411.13543.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs, 2018. URL https://arxiv.org/abs/1806.07011.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023. URL https://arxiv.org/abs/2310.13724.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Anian Ruoss, Fabio Pardo, Harris Chan, Bonnie Li, Volodymyr Mnih, and Tim Genewein. Lmact: A benchmark for in-context imitation learning with long multimodal demonstrations, 2025. URL https://arxiv.org/abs/2412.01441.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Xinyi Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10740–10749, 2020.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.03768.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments, 2021. URL https://arxiv.org/abs/2108.03332.

Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3221–3241, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.166/.

Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, Xizhou Zhu, Ping Luo, Yu Qiao, Jifeng Dai, Wenqi Shao, and Wenhai Wang. Needle in a multimodal haystack, 2024. URL https://arxiv.org/abs/2406.07230.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Fei-Fei Li, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, 2024. URL https://api.semanticscholar.org/CorpusID:274822996.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. MMEgo: Towards building egocentric multimodal LLMs for video QA. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=67sSPPAZiG.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity?, 2025. URL https://arxiv.org/abs/2502.05252.

A Dataset Construction

A.1 Building the Needle(s) in the Embodied Haystack Benchmark

We construct the Needle(s) in the Embodied Haystack benchmark in three stages: 1) Trajectory Replay and Metadata Collection; 2) Rule-Based QA Generation; and 3) Cross-validation with Multimodal LLMs. The following sections provide detailed descriptions of each step.

A.1.1 Trajectory Replay and Metadata Collection

We first replay 225 test trajectories generated by ∞-THOR, logging both visual observations (agent's egocentric views) and structured metadata at each timestep. For every step, we store the list of visible objects, agent-inventory items, openable containers, and their contents from the simulator. This produces a fine-grained interaction log that captures grounded scene dynamics over time.

An example of the collected metadata at a single timestep is shown below:

Example of metadata entry


```
{
    "img_idx": 2,
    "img_filename": "000000002.png",
    "step": 1,
    "object_log": {
        "visible": ["Shelf", "Vase", "Book"],
        "pickupable": ["Vase", "Book"],
        "isOpen": [],
        "inven_obj": [],
        "receptacles": ["Shelf"],
        "recep_objs": {
            "Shelf": ["Vase", "Book"]
        }
    }
}
```


Each metadata entry corresponds to a low-level action step and provides the semantic state of the scene, enabling the construction of temporally grounded QA instances in later stages.

A.1.2 Rule-Based QA Generation

To construct the QA set, we apply rule-based generation templates to each trajectory using its sequence of low-level actions and associated metadata. The QA generation process involves parsing the agent's interactions with objects, containers, and the environment, and applying a set of handcrafted rules to synthesize grounded questions.

Our QA generation logic covers a diverse range of question types, including object presence, object state, location tracking, slicing actions, container content reasoning, and action counting. For instance, if an object is seen for the first time at a particular step, a presence question such as "Is there any `apple` in this room?" is generated. Similarly, after a `PutObject` action, location-based questions like "Where was the `apple` before you put it to the `microwave`?" are produced. When slicing actions happen, we create questions about the object being sliced and other nearby items (e.g., "What objects were in the `Fridge` when you sliced the `apple`?"). Then, we sample questions based on the frequency to ensure diversity across object types, and annotate the GT answer steps using the replay logs. Table 5 summarizes the types of questions generated, and corresponding trigger conditions and example templates.

A.1.3 Cross-validation with Multimodal LLMs

To ensure the answerability and clarity of the generated QA pairs, we perform cross-validation using four powerful multimodal LLMs: LLaVA-OneVision 7B (Li et al., 2024a), Qwen2.5-VL 7B (Bai

Table 5: QA types, trigger conditions, and corresponding question templates used in rule-based generation.

QA Type	Trigger Condition	Example Template(s)
object presence (Yes/No)	object appears visibly in the trajectory	`Is there any {obj} in this room?` `Have you seen a/an {obj}?`
open state questions	container marked as open in metadata	`Was {container} open?`

object location tracing	sequences of `Pickup` and `PutObject` actions	`Where was {obj} before you put it to {container}?`
		`Where did you move the {obj} from the {container}?`
		`Where is {obj} now?`

slicing-based questions	`SliceObject` action detected in trajectory	`What did you slice?` `What objects were in/on the {container} when you slice the {obj}?`
container content	container visibility with non-empty contents	`What objects were in/on the {container}?` `What object did you put in/on the {container}?`
put action questions	unique `PutObject` action for a container	`What object did you put in/on the {container}?`

final object state	final location of an object at episode end	`Is {obj} in/on the {container}?`
		`What objects are in/on the {container}?`
		`How many objects were in/on the {container}?`

movement counting	object picked up more than once	`How many times did you move {obj}?`

et al., 2025), Deepseek-VL 7B (Lu et al., 2024), and Pixtral 12B (Agrawal et al., 2024). Each model is prompted with the GT images corresponding to the annotated QA steps and asked to answer the associated questions. Given their strong performance on standard visual QA tasks, we use these models to assess whether a question can be correctly answered or not. We keep only the QA pairs that are correctly answered by at least one of the four models, and discard those that fail across all models. This helps improve dataset quality and filtering out ambiguous or visually ungroundable questions. Table 6 shows the accuracy of each model on the finalized QA set when evaluated with GT images. Notably, even with access to GT images, all models struggle with questions requiring reasoning over three or more evidence steps. To maintain the benchmark's difficulty and support evaluation of more capable models in future, we manually inspect the multi-clue questions and include those that are answerable.

Table 6: QA accuracy (%) of multimodal LLMs on ground-truth images.

Model	Size	# of clues (GT steps)			Total
		1	2	≥3	
LLaVA-OneVision	7B	86.61	68.55	23.74	71.15
Qwen2.5-VL	7B	85.94	89.83	64.40	82.20
Deepseek-VL	7B	81.56	39.14	22.57	62.88
Pixtral	12B	91.34	39.60	58.56	76.25

A.2 Constructing Long-Horizon Trajectories

Figure 6 shows an example of the trajectory and a long-horizon embodied task generated from ∞-THOR. To synthesize long-horizon trajectories, we construct each trajectory by sequentially chaining successful sub-tasks sampled from a predefined set of task templates. This process is illustrated in Algorithm 1. We begin by sampling a task template from a fixed task pool, which includes goal types such as `pick and place simple`, `pick two obj and place`, and `pick and`

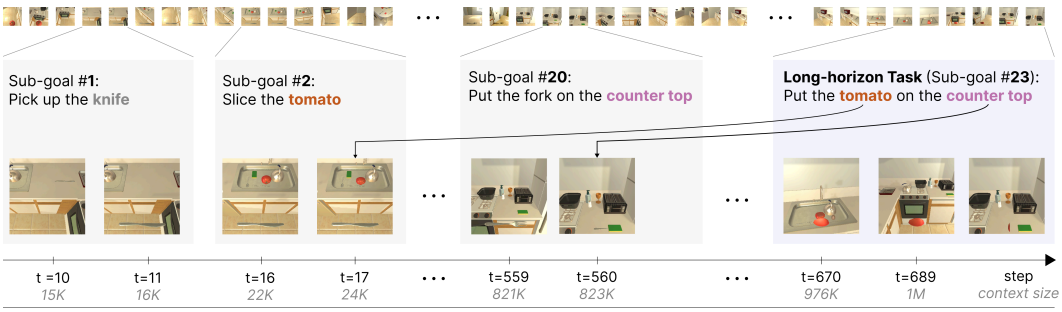


Figure 6: Example of the trajectory and a long-horizon embodied task generated from ∞-THOR. The final goal ("Put the tomato on the counter top" at t=670) requires recalling both the tomato (seen at t=17) and the counter (seen at t=560) to solved the long-horizon task. Context size refers to the input token length when converting the trajectory into the VLM input space.

Algorithm 1 Construct Long-horizon Trajectory

Input: Task Pool $\mathcal{T}$, max sub goals N
Output: Long-horizon trajectory τ
Initialize empty trajectory $\tau \leftarrow []$
while len(τ) $< N$ **do**
    Sample task template $g \sim \mathcal{T}$ and objects
    Plan action sequence π_g by planner
    if Simulate(π_g) is successful **then**
        Append π_g to trajectory: $\tau \leftarrow \tau \parallel \pi_g$
    else
        Discard and re-sample
    end if
end while
// Final sub-task with long-term object dependency
Sample $g_{\text{final}} \sim \mathcal{T}$ and objects in early 20% and late 20%
Plan and simulate π_{final} using restricted objects
if Simulate(π_{final}) is successful **then**
    Append π_{final} to trajectory: $\tau \leftarrow \tau \parallel \pi_{\text{final}}$
else
    Repeat sampling until success
end if
return τ

`place with movable recep`. Each sampled template requires relevant objects in the scene (e.g., pickupable items, target receptacles), which are then used to define the goal for that task.

We use a classical task planner, which operates over PDDL-defined domains (Shridhar et al., 2020), to generate a low-level action sequence for the sampled goal, and simulate this plan in an interactive environment. If the rollout fails (e.g., due to collisions, object occlusions, or unreachable conditions), we discard the sequence and re-sample from the task pool. Otherwise, the successful rollout is retained and appended to the ongoing trajectory.

This sampling-execution loop is repeated until a long trajectory with a desired number of sub-goals is formed. The resulting synthetic long-horizon trajectory consists of multiple sub-goals concatenated into a continuous sequence. To induce long-term temporal dependencies, the final sub-task is constrained to involve only objects that appear in the early 20% and late 20% of the overall trajectory, requiring the agent to integrate temporally distant evidence to answer associated questions.

A.3 Dataset Statistics

Table 7 summarizes the statistics for the NiEH test set and the generated long-horizon trajectories. The training set consists of 2,456 trajectories with an average of 14 sub-goals and 405 steps per

Table 7: Dataset statistics.

NiEH testset		Single-clue	Multi-clue
# of question-answer pair		829	474
Trajectory	**Train**	**Dev**	**Test**
# trajectory	2,456	125	225
# avg subgoals	14	16	18
# max subgoals	30	24	33
# avg steps	405	613	627
# max steps	654	890	952
# avg token length	602K	880K	912K
# max token length	954K	1.2M	1.3M
# avg interaction with objects per episode	31	36	37
# max interaction with objects per episode	82	62	74
# scene	82	30	39

episode. Token lengths are calculated using the LLaVA-OneVision tokenizer. With an average input length of 602K tokens, the entire training dataset amounts to approximately 1.48B tokens (2,456 $\times$ 602K), providing a substantial corpus for learning long-horizon temporal dependencies.

B Training Details

We fine-tune the LLaVA-OneVision 7B and Qwen2.5-VL 7B model on our training set while freezing the vision encoder. Training utilized 8 H100 GPUs with tensor Shoeybi et al. (2020) and pipeline parallelism Huang et al. (2019) for 32K context, and Context Parallelism for larger contexts (64K, 130K). The model is trained using a next-action prediction objective, where only the action tokens are optimized, conditioned on the goal and state tokens. Table 8 summarizes the training specifications for different context lengths. For 32K training, we apply tensor parallelism with a degree of 4 and pipeline parallelism with a degree of 2, utilizing 8 H100 GPUs in total. Since pipeline parallelism requires the batch size to match the pipeline degree, we set the batch size to 2. For longer context lengths, we use context parallelism: 8-way for 64K (on 8 GPUs) and 16-way for 130K (on 16 GPUs). All models are fine-tuned for approximately 3 epochs with a learning rate of 1e-5, using the AdamW optimizer and a linear learning rate schedule with a 0.03 warmup ratio.

Table 8: Training specifications for different context lengths.

Context Length	**Parallelism**	**# GPUs**	**Training Time**
32K	Tensor (4) + Pipeline (2)	8	160 hrs
64K	Context (8)	8	120 hrs
130K	Context (16)	16	134 hrs

C Additional Results

C.1 Static Evaluation: Needle(s) in the Emboided Haystack

Figure 7 presents the performance of LLaVA-OneVision (OV) 7B (Li et al., 2024a) model across various context extension methods. Linear Interpolation, Dynamic Scaling, and LongRoPE scaling all struggled with very long contexts beyond 128K tokens (the results of Linear Interpolation are excluded from Figure since it fails at all examples). YaRN consistently outperformed other methods, effectively handling contexts above 384K tokens, likely due to its architectural alignment with LLaVA-OV's Qwen2 LM backbone, which employs RoPE and YaRN scaling during pretraining. YaRN performed best at moderate scaling factors (e.g., x4), however, further scaling to x8 and x16 did not yield additional gains. In particular, x16 slightly improved performance in the 256K–384K token range but led to degradation notably at shorter context sizes ($<$64K), suggesting that excessive

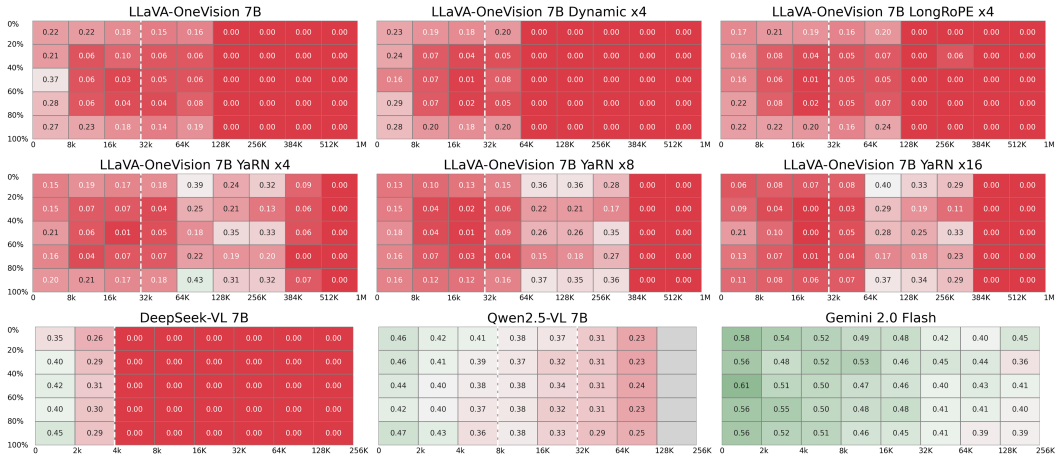


Figure 7: **Full results of Needle(s) in the Embodied Haystack.** The x-axis denotes the input context length (tokens), and the y-axis denotes the needle depth. The white dashed line denotes the maximum pre-training context length of each model.

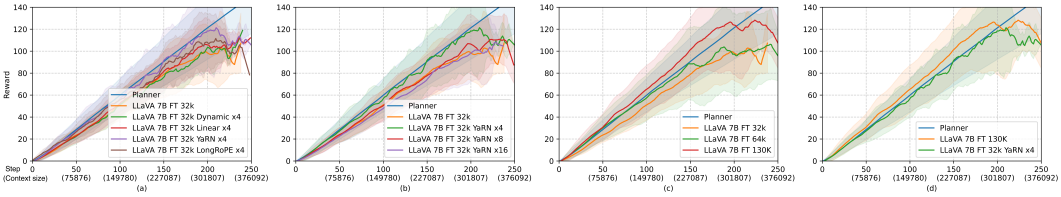


Figure 8: Agent's reward across different experimental configurations for high-level planning tasks. We compare (a) context extension methods at fixed scaling (x4), (b) varying YaRN scaling factors, and (c) fine-tuning with different context lengths using Context Parallelism. (d) summarizes the most effective strategies, highlighting that exposure to longer contexts during training significantly improves performance. Non-planner models cannot generate valid actions after around 250 steps (≈376K in context size).

scaling may introduce instability and negatively impact performance. Overall, all methods fail beyond 512K tokens, highlighting a need for improved long-context methods. The third row includes additional results from DeepSeek-VL 7B (Lu et al., 2024), Qwen2.5-VL 7B (Bai et al., 2025), and Gemini 2.0 Flash[1]. Although Gemini is renowned for strong long-context multimodal reasoning (up to 3,000 image input), our results (<40% with full trajectories at 256K) indicate substantial room for improvement.

C.2 Online Evaluation: High-level Planning

Plan-level Evaluation. We evaluate agent performance using a plan-level framework, where each plan corresponds to a short sequence of actions aimed at achieving a specific intermediate subgoal (e.g., navigating to an object, placing an item). A trajectory is composed of multiple such plans, executed sequentially. For the interactive evaluation, the agent is presented with the current plan's goal along with the history of previous GT states and actions. Using this context, the agent predicts the next action and interacts step-by-step with the environment. The interaction continues until the current plan is either successfully completed or terminated due to failure (e.g., collisions or deadlocks). After each plan, the context is reset to include the GT actions and states from the completed portion of the trajectory, and the agent proceeds to the next plan. This ensures that each plan is evaluated independently, conditioned only on the correct prior history. The agent's performance is measured via cumulative reward across all plans in the trajectory. Pseudocode for this evaluation procedure is provided in Algorithm 2.

[1]We used the `gemini-2.0-flash-001` version for all experiments.

Algorithm 2 Plan-Level Evaluation

Input: Trajectory $T = \{P_1, P_2, \ldots, P_N\}$, Agent policy π, Environment $\mathcal{E}$
Initialize: Reward $R \leftarrow 0$
Initialize state and history with initial observation
for each plan P_i in T **do**
    Initialize context with GT actions up to P_{i-1}
    while not *done* and not *failure* **do**
        $a_t \leftarrow \pi(\text{context})$
        $s_{t+1}, r_t, \textit{done}, \textit{failure} \leftarrow \mathcal{E}.\text{step}(a_t)$
        Append (a_t, s_{t+1}) to context
        $R \leftarrow R + r_t$
    end while
    if *failure* **then**
        Break evaluation
    end if
end for
return Total accumulated reward R

Results and Discussion. Figure 8 presents the accumulated rewards over time across four experimental configurations. The Planner trajectory represents the performance upper bound. Our analysis focuses on addressing the following key questions:

Q. Which context extension methods perform best? Figure 8(a) compares different context extension methods at a fixed scaling factor of x4. Similar to the NiEH results, YaRN consistently achieves the highest performance showing very close performance to Planner.

Q. Does further scaling enhance performance? Figure 8(b) explores YaRN scaling at different scaling factors (x4, x8, and x16). Interestingly, increasing the scaling factor beyond x4 does not significantly improve performance, indicating a diminishing return for larger scaling factors.

Q. Is fine-tuning on a dataset with long trajectories effective? Figure 8(c) demonstrates the effectiveness of fine-tuning with Context Parallelism, enabling scaling of context lengths up to 64K and 130K tokens. At a 130K context size, the model can learn sequences comprising approximately 86 steps, substantially longer compared to only 22 steps with a 32K context size. This shows that exposure to longer context during training significantly enhances model performance, suggesting that incorporating more long-horizon data by ∞-THOR could further improve model capabilities. We note that context extension methods were not applied in this experiment.

Q. Does combining context extension methods during both training and inference provide additional benefits? Results of experiments with scaling at both training and evaluation (Figures 8(d)) indicate that additional scaling at evaluation after fine-tuning with scaled RoPE provides no further performance improvement and may degrade performance at shorter context lengths ($\leq$300K tokens).

Based on these observations, we can conclude that fine-tuning strategies are most effective when long-trajectory datasets are available. In the absence of extensive training data, employing YaRN scaling at x4 yields performance comparable to the Planner upper-bound, particularly within context lengths under 200K tokens.

C.3 ONLINE EVALUATION: LOW-LEVEL MANIPULATION

Table 9: Success rates of OpenVLA-7B and SpatialVLA-4B on low-level Pick-up and Put tasks in ManipulaTHOR.

Task	OpenVLA-7B	SpatialVLA-4B
Pick-up	16.52%	18.70%
Put	2.08%	1.19%

We evaluate the ability of existing VLA models, OpenVLA-7B (Kim et al., 2024) and SpatialVLA-4B (Qu et al., 2025), to control robot arms on low-level manipulation tasks. The evaluation protocol follows the procedure described in Appendix B.1, with the only difference being that low-level VLA models are used to directly execute Pick-up and Put actions through arm control. For the text input, we provide task-specific instructions ("Pick-up" or "Put" object), since the existing models are only trained on single-task formulations. For the image input, we use an ego-centric camera view. Since these VLA models were originally trained on datasets with different viewing angles, there remains considerable room for improvement by incorporating additional camera perspectives during training and evaluation.

For the evaluation criterion, ManipulaTHOR implements a "magnet sphere" hand mechanism: if a pickupable object is within a specified radius of the agent's hand when the `PickupObject` action is called, the object is successfully picked up. Since neither OpenVLA nor SpatialVLA was trained in the AI2-THOR environment, we relax this success threshold by setting the radius to 0.4 to account for discrepancies between training and evaluation environments. This loosened criterion ensures that small deviations in arm trajectories do not result in an immediate failure, thereby providing a fairer comparison of the models.

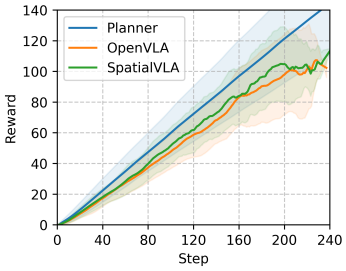


Figure 9: Agent's reward with low-level manipulation tasks. Low-level VLA models are used to control the robot arm for Pick-up and Put actions.

Figure 9 presents accumulated rewards on low-level manipulation tasks. Both models underperform relative to the Planner baseline, largely due to differences in robot arm configuration and the out-of-distribution nature of the visual inputs. The results show that SpatialVLA achieves slightly higher and more stable rewards than OpenVLA across long sequences.

Table 9 reports task success rates. Overall performance remains low, with both models struggling on manipulation tasks. We attribute these results primarily to two factors: (1) ManipulaTHOR's egocentric view, which provides nearly identical images for fine-grained arm movements $(\Delta x, \Delta y, \Delta z)$ smaller than 0.01, making it difficult for VLA models to perceive subtle adjustments; and (2) discrepancies between the robot arm in ManipulaTHOR and those assumed by the VLA models, causing many predicted actions to fail in the physical simulation. We expect that fine-tuning on our dataset could provide an effective solution to bridge these gaps.

D LIMITATIONS

While ∞-THOR enables the generation of arbitrarily long trajectories, the diversity of environment layouts in AI2-THOR is inherently limited. This can lead to repetitive agent behaviors within certain scenes. For instance, compared to kitchen and living room scenes, bedroom scenes tend to involve fewer action types, mostly constrained to simple pick-and-place tasks within small spatial areas. Due to the limited scene size and low task diversity, we excluded bathroom scenes from our dev and test sets. In future work, we plan to integrate ∞-THOR with ProcTHOR (Deitke et al., 2022), which supports procedurally generated environments, enabling a broader range of dynamic and diverse scene configurations.

Additionally, although the GT action sequences generated by the PDDL-based planner are sufficient for task completion, they are not guaranteed to be optimal. This may lead to agents learning suboptimal behaviors when trained solely on these demonstrations. Incorporating learning from exploration or reinforcement-based optimization could improve policy quality.

Finally, inference with LLMs in long-context settings remains a computational bottleneck, particularly as context lengths approach 1M tokens. Since our long-horizon tasks require access to information spread throughout the entire trajectory, full-context inference becomes increasingly expensive, even with Context Parallelism. One promising direction is to equip agents with memory systems that selectively keep relevant information from prior steps, allowing the model to reason without reprocessing the full context at each step. Other architectural solutions, such as sparse attention

mechanisms (Han et al., 2024; Jiang et al., 2024) or state-space models with linear-time inference (Gu & Dao, 2024), also hold potential for scalable long-context reasoning.