Beyond Turn Limits: Training Deep Search Agents with Dynamic Context Window

Anonymous ACL submission

Abstract

While recent advances in reasoning models have demonstrated cognitive behaviors through reinforcement learning, existing approaches struggle to invoke deep reasoning capabilities in multi-turn agents with long-horizon interactions. We propose DeepMiner, a framework that elicits such abilities in deep search agents by introducing a dynamic context window and challenging training tasks. We design a dynamic context management strategy that utilizes a sliding window to selectively drop distant tool responses while preserving all assistant reasoning traces, eliminating the dependency on external summarization models. To provide training tasks that genuinely demand long-horizon reasoning, DeepMiner further employs a reverse construction method to generate complex yet verifiable question-answer pairs from authentic web sources, ensuring both difficulty and reliability. Through reinforcement learning on Qwen3-32B, we develop DeepMiner-32B, which achieves substantial performance improvements across multiple search agent benchmarks. DeepMiner attains 33.5% accuracy on BrowseComp-en, surpassing the much larger DeepSeek-V3.1-671B, and demonstrates consistent improvements on BrowseComp-zh, XBench-DS, and GAIA. Notably, our dynamic context management enables sustained interactions of nearly 100 turns within commonly used 32k context length, effectively addressing the context limitations that constrain existing multi-turn agents.

1 Introduction

Recent advances in Reinforcement Learning with Verifiable Rewards (RLVR) have enabled significant breakthroughs in mathematical reasoning and code generation (DeepSeek-AI, 2025; Yang et al., 2025; Liu et al., 2025b). These methods empower large language models to exhibit sophisticated cognitive behaviors such as self-verification, backtracking, and subgoal decomposition in single-turn settings (Gandhi et al., 2025). The release of OpenAI's Deep Research (OpenAI, 2025a) demonstrates that such cognitive capabilities can be successfully extended to long-horizon tasks, yet how to achieve this extension remains unknown.

While existing open-source efforts have explored this direction through high-quality data generation (Li et al., 2025b; Tao et al., 2025; Gao et al., 2025; Wu et al., 2025) and specialized reinforcement learning algorithms (Song et al., 2025; Jin et al., 2025; Zheng et al., 2025), they have not achieved the same level of capability enhancement observed in single-turn settings. Substantial performance gaps persist compared to proprietary systems that maintain stable performance across dozens of interactions.

In this paper, we focus on search agents and identify two fundamental challenges that prevent existing approaches from effectively leveraging long-horizon interactions:

- **Context Management Limitations.** Long-horizon interactions rapidly consume available context through accumulated tool responses, with a typical 32k context length supporting only approximately 10-20 effective interaction turns. Current solutions primarily rely on summarization compression of tool outputs (Li et al., 2025b; Gao et al., 2025), but this approach introduces several limitations. First, summarization inevitably loses fine-grained information critical for precise reasoning. Second, additional summarization models increase system complexity and computational overhead. Third, summarization components cannot be integrated into end-to-end RLVR optimization, creating optimization blind spots.

- **Insufficient Task Complexity.** Traditional multi-hop QA datasets (Joshi et al., 2017; Ho et al., 2020; Yang et al., 2018) permit success

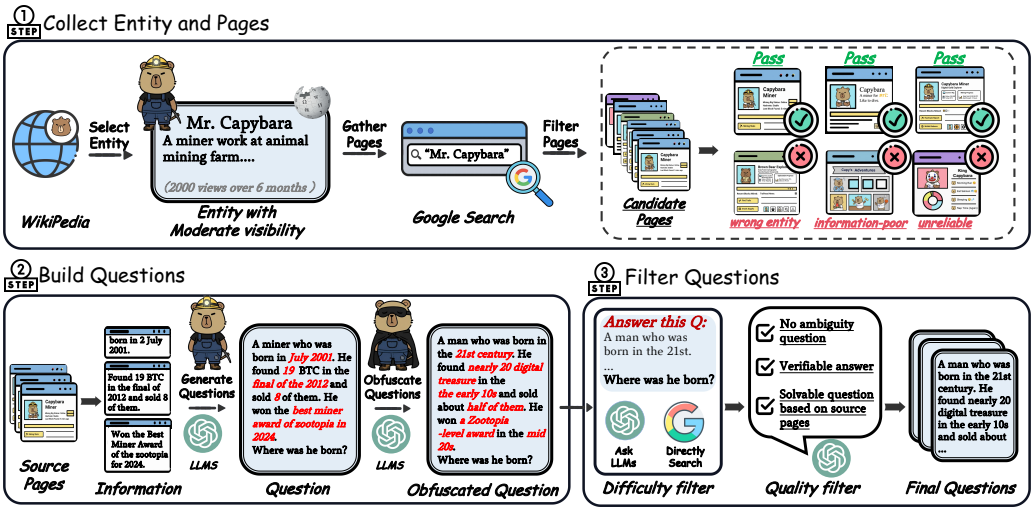

Figure 1: Pipeline for complex question construction: collecting moderate-visibility entities and related web pages, building QA pairs with obfuscation to increase difficulty, and filtering out overly simple or low-quality questions.

through shallow retrieval, and although recent efforts synthesize more challenging data (Li et al., 2025b; Tao et al., 2025; Lu et al., 2025b), producing tasks that are simultaneously difficult, verifiable, and reliable remains essential for eliciting deep reasoning.

To address these challenges, we present DeepMiner, a training framework for long-horizon multi-turn agents via managing continuously growing model contexts and constructing high-difficulty training tasks. Our primary contribution lies in addressing context management limitations through a dynamic sliding window over tool responses that selectively drops distant ones while preserving all assistant reasoning traces. Our strategy maintains access to original webpage content rather than compressed summaries, avoiding the information loss and optimization blind spots inherent in summarization-based approaches. We show that a simple recency-based dropping of distant tool responses, when combined with preserved reasoning traces and a training-inference consistent RL formulation, is sufficient to sustain long-horizon search behavior without external summarization. To provide training tasks that demand such long-horizon reasoning, we develop a reverse construction method that generates challenging QA pairs by synthesizing questions across multiple authentic web sources, with rigorous filtering to ensure genuine reasoning demands.

We implement DeepMiner on Qwen3-32B and evaluate the resulting model across multiple deep research benchmarks. DeepMiner-32B achieves 33.5% accuracy on BrowseComp-en, outperforming DeepSeek-V3.1-671B, a model over 20 times larger. Consistent improvements are observed across BrowseComp-zh, XBench-DeepSearch, and GAIA. These advantages suggest that high-quality training data and effective context management can jointly facilitate the development of deep reasoning capabilities in long-horizon, multi-turn interactions. Notably, our dynamic context management enables sustained interactions of nearly 100 tool calls within commonly used 32k context length, demonstrating the effectiveness of our approach for extending the operational horizon of long-horizon reasoning agents.

2 Complex Question Construction

Effective long-horizon training requires tasks that genuinely demand extended exploration and multi-step reasoning. To provide such a training signal, we construct complex QA pairs through a reverse construction approach. Rather than composing questions over predefined entity-relation graphs (Li et al., 2025b; Lu et al., 2025b), we synthesize them through cross-document reasoning over free-form web pages, enabling scalable construction while ensuring difficulty and verifiability. Figure 1 illustrates our three-stage pipeline: entity-driven information collection, multi-source question generation, and quality filtering.

Entity-driven Information Collection. We begin with entity selection from Wikipedia, targeting individuals with moderate visibility [1] to strike a balance between sufficient information availability

[1] 1,000∼10,000 page views over the last six months.

and excessive familiarity. For each selected entity, we conduct comprehensive information gathering through two primary search strategies: direct name queries for biographical information and news searches for recent developments, typically yielding dozens of potentially relevant web pages per entity. The collected sources are filtered to ensure they (1) refer to the correct entity rather than namesakes, (2) provide information beyond Wikipedia, which models can often recall from parametric knowledge, and (3) originate from credible sources.

Question Generation. The question generation process takes multiple selected sources as input while deliberately excluding Wikipedia pages to force synthesis across distributed sources. Each question is constrained to require information from at least four distinct sources, ensuring that no single document suffices for answering. To further increase difficulty, generated questions undergo secondary obfuscation processing. This obfuscation process increases reasoning demands by replacing specific information with more generic descriptions, e.g., "1905" $\rightarrow$ "early 20th century", "born in Manhattan" $\rightarrow$ "born in New York". This obfuscation prevents simple keyword matching while maintaining answer uniqueness, as the combination of constraints still points to exactly one correct answer. The output of this stage is a tuple of (question, answer, reference sources) for each entity, where the answer is derivable only through cross-document reasoning over the reference sources.

Multi-stage Filtering. Generated QA pairs undergo difficulty filtering and quality filtering to ensure both difficulty and reliability. *Difficulty filtering* ensures questions require extensive reasoning beyond current model capabilities through two mechanisms: direct search engine queries and zero-shot prompting of reasoning models. We eliminate questions for which either mechanism successfully finds the entity or correct answer, ensuring retained questions genuinely demand tool-assisted multi-step exploration. *Quality filtering* controls multiple aspects that could compromise reliability. We eliminate questions with the following attributes: 1) contains elements or expressions that may introduce interpretation ambiguity, 2) has evasive or ambiguous answers, and 3) the answer is not logically derivable from available evidence in the given reference documents. A human study confirms that the resulting questions are well-posed and difficult yet solvable (Appendix F.2).

This reverse construction pipeline enables scalable generation of complex QA pairs from authentic web sources. Question examples and all prompts can be found in Appendix K.

3 Reinforcement Learning with Dynamic Context Window

3.1 Dynamic Context Management Strategy

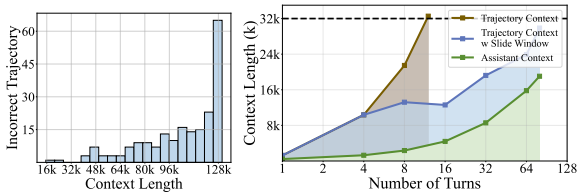

Figure 2: Preliminary experiments with GPT-OSS-120B (Agarwal et al., 2025) on the BrowseComp dataset. Left: Length distribution of incorrect trajectories. Right: Context length changes across rounds with/without sliding window. Without sliding window, tool responses grow rapidly and squeeze assistant context. With sliding window, tool length stays constant while assistant content grows normally.

Empirical Analysis of Context Challenges. Complex agent tasks typically require language models to engage in long-horizon interactions. However, in real-world scenarios, the extent of these interactions is fundamentally constrained by the model's maximum context length. To illustrate this challenge, we analyze error patterns of open-source models on the complex information-seeking task BrowseComp. As shown in left subplot of Figure 2, even when setting the max context length to 128k, the majority of model failures occur when the maximum context length is reached. Case analysis reveals that models often recognize they have not found the answer and attempt to continue searching, but are forced to terminate due to context constraints rather than task completion.

The right subplot of Figure 2 further examines context composition under the commonly used 32k limit. Models typically reach context limits after only 10-20 turns, which proves insufficient for complex deep research tasks. Detailed analysis of context composition reveals that tool responses are typically 5-10 times longer than assistant responses, creating rapid context consumption that severely limits interaction depth. Further investigation into tool response shows that in most cases, current tool responses primarily influence the model's immediate next few decisions, with minimal impact on

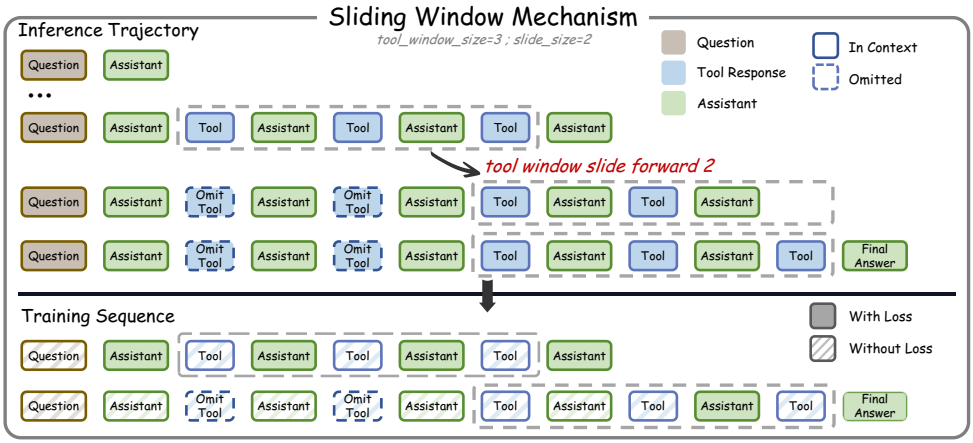

Figure 3: Sliding window mechanism for dynamic context management. Upper: inference trajectory with sliding window (illustrated with W=3, S=2 for simplicity; experiments use W=5, S=3) where older tool responses are replaced by omission tokens while assistant outputs are preserved. Lower: corresponding training sequences where only newly generated responses receive gradient updates while previous outputs serve as fixed context.

outputs at distant interaction positions. This observation suggests that while tool responses are essential for local decision-making, their long-term retention may not be necessary for maintaining coherent reasoning strategies across extended interactions. We provide detailed analysis of this assumption in Appendix H.

Sliding Window Mechanism. Motivated by these findings, we design a block-level sliding window strategy: the window slides over complete tool responses rather than individual tokens, retaining recent tool responses while preserving all assistant reasoning traces. Let $\tau = \{q, a_1, t_1, \ldots, a_{T-1}, t_{T-1}, a_T\}$ represent a complete interaction trajectory, where q denotes the user query, a_i represents the i-th assistant response, and t_i represents the corresponding tool response. We implement sliding window management with two parameters: window size $\mathcal{W}$ controls how many recent tool responses remain visible, and slide size $\mathcal{S}$ controls how many tool responses are dropped in each sliding operation.

As shown in Figure 3, when generating a_i, if the number of prior visible tool responses reaches $\mathcal{W}$, the sliding operation is triggered; otherwise, the agent continues with standard ReAct-style reasoning using the prior context. When the sliding operation is triggered, we identify the sliding boundary $b = \max(1, i - \mathcal{W} + \mathcal{S})$ and replace tool responses $\{t_1, \ldots, t_{b-1}\}$ with a placeholder sentence ϕ ="[Previous tool output skipped. Re-run tool if needed.]", while $\{t_b, \ldots, t_{i-1}\}$ remain visible. Consequently, the prompt trajectory used to generate a_i is $\tilde{\tau}^{(i)} = \{q, a_1, \phi, \ldots, a_{b-1}, \phi, a_b, t_b, \ldots, a_{i-1}, t_{i-1}\}$. This yields a modified trajectory where early tool responses are dropped while all assistant reasoning remains intact. As shown in Figure 2, the sliding window lets the agent sustain nearly 100 turns within a commonly used 32k context.

Training-Testing Consistency. The sliding window mechanism introduces a new challenge: training models to operate effectively under dynamic context conditions. Direct training on complete trajectories would create a mismatch with inference behavior, where models encounter varying context states as sliding operations occur.

To address this, we decompose each training trajectory into multiple subsequences, where each subsequence replicates the exact context the model will encounter at inference. Specifically, a trajectory τ with T assistant responses is segmented into K training subsequences. Each time the sliding operation is triggered, a new subsequence begins. If the sliding operation is triggered before generating a_i at step i, we create subsequence k as: $\tau^{(k)} = \{q, a_1, \phi, \ldots, a_{b-1}, \phi, a_b, t_b, \ldots, a_{i-1}\}$ where the context reflects the dropped tool responses followed by the visible sliding window, exactly as the model would see during inference. The total number of subsequences equals the number of sliding operations plus one: $K = \lfloor \frac{T-\mathcal{W}}{\mathcal{S}} \rfloor + 1$.

At this point, early assistant responses appear in multiple subsequences. To ensure each response is trained exactly once, we design a masking mechanism that selectively enables gradients. Specifically, for the i-th assistant response in the k-th

subsequence, we define the loss mask as:

$$M_i^{(k)} = \begin{cases} 0 & \text{if } i < \mathcal{W} + (k-2) \cdot \mathcal{S} + 2 \text{ and } k \neq 1 \\ 1 & \text{otherwise} \end{cases}$$

This ensures models train under the same dynamic context conditions they encounter at inference, while enabling scalable optimization on arbitrarily long trajectories.

3.2 Cold Start

To establish foundational tool-calling and long-horizon reasoning capabilities, we employ supervised fine-tuning as a cold-start phase. We design a pipeline to generate action trajectories through powerful open-source language models, where the sliding window mechanism is also applied during trajectory generation to eliminate turn limitations. We filter trajectories with incorrect answers or excessive length, retaining high-quality examples that demonstrate effective tool usage and multi-step reasoning. During training, we employ the multi-sequence construction method described in Section 3.1 to ensure training-testing consistency.

3.3 Reinforcement Learning Training

Trajectory-level Advantage with Subsequence-level Training. We employ Group Relative Policy Optimization (Shao et al., 2024) as our reinforcement learning algorithm, and adapt the advantage computation to work with our sliding window context management. The key challenge lies in reconciling trajectory-level reward signals with subsequence-level training requirements imposed by sliding window processing.

For each question q, we generate G complete trajectory rollouts using the sliding window mechanism during inference. Rewards are computed at the trajectory level based on the final answer, and advantages are calculated through group-relative comparison among these G trajectories: $\hat{A} = \left(R_i - \text{mean}(\{R_j\}_{j=1}^G)\right) / \text{std}(\{R_j\}_{j=1}^G)$.

Our modification lies in advantage propagation: each trajectory τ_i is decomposed into multiple training sequences $\{\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(K)}\}$ following the sliding window protocol. The trajectory-level advantage $\hat{A}$ is then propagated to all training subsequences derived from that trajectory, ensuring each sequence receives the same advantage signal regardless of its specific context state. This preserves the group-relative advantage computation while enabling training under dynamic context conditions.

Reward Design. We employ a straightforward reward design where an LLM judge evaluates whether the final answer matches the ground truth, assigning a binary reward of 1 for correct answers and 0 for incorrect ones. This binary signal provides a clear optimization target while avoiding the biases of more complex reward engineering.

4 Experiments

4.1 Experimental Setup

Tool Configuration. Our agent operates with two primary tools: *web_search* (backed by the Serper API) returns the top-10 search results (titles, URLs, and snippets), and *fetch* (built on Jina Reader) retrieves webpage content in Markdown with pagination for navigating long documents. The full tool details are provided in Appendix C.

Training Details. We adopt Qwen3-32B (Yang et al., 2025) as our base model with thinking mode enabled, and employ a two-stage SFT-then-RL strategy: SFT on approximately 3,000 high-quality trajectories (generated by GPT-OSS-120B with rejection sampling; see Appendix D), followed by RL on around 4,000 questions. We use a maximum trajectory length of 40K tokens and a turn limit of 60, with sliding window size 5 and slide size 3. Full hyperparameters and training details are provided in Appendix E.

Evaluation. We evaluate DeepMiner on four challenging deep research benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-zh (Zhou et al., 2025a), XBench-DeepSearch (Chen et al., 2025), and GAIA (Mialon et al., 2023). We use the complete test sets for the main experiments and a BrowseComp subset for detailed analysis. We use temperature 0.6, top-p 0.9, and up to 100 interaction turns. All results are judged by ChatGPT-4o-Latest (prompts in Appendix J), following standard practice for short-answer browsing benchmarks (Wei et al., 2025; Phan et al., 2025).

Baselines. We compare against three categories: commercial deep research systems (OpenAI, 2025a; Metaso, 2025; Team et al., 2025), general LLMs with tools (OpenAI, 2025b; Anthropic, 2025; DeepSeek-AI, 2025, 2024; Zeng et al., 2025), and open-source search agents (Tao et al., 2025; Gao et al., 2025; Wu et al., 2025; Li et al., 2025b; Liu et al., 2025a; Lu et al., 2025b; Li et al., 2025a).

Model	BC	BCZH	XBench	GAIA
Commercial Agents				
OpenAI Deep Research	51.5	42.9	-	67.4
Metaso Deep Research	12.0	45.4	64.0	-
Kimi Researcher	-	-	-	69.0
General LLMs with tools				
OpenAI-o3	49.7	58.1	66.7	70.5
Claude-4-Sonnet	12.2	29.1	64.6	68.3
DeepSeek-R1	8.9	35.7	55.0	-
Kimi-K2-Instruct-1T	14.1	28.8	50.0	57.7
GLM-4.5-355B	26.4	37.5	70.0	66.0
DeepSeek-V3.1-671B	30.0	49.2	71.2	63.1
Open-source Agents				
WebShaper-72B	-	-	-	60.0
SearchR1-32B	-	-	19.5	28.6
ASearcher-32B	-	-	42.1	52.8
WebDancer-32B	3.8	18.0	39.0	51.5
WebSailor-72B	12.0	30.1	55.0	55.4
DeepDive-32B	14.8	25.6	50.5	-
WebExplorer-8B†	15.6	32.0	53.7	50.0
WebSailor-V2†‡	35.3	44.1	73.7	74.1
Our Agents				
DeepMiner-32B-SFT	21.2	28.0	53.0	54.4
DeepMiner-32B-RL	33.5	40.1	62.0	58.7

Table 1: Performance comparison on four deep research benchmarks. Most open-source agents, including DeepMiner, use a 30k to 40k context; † denotes the open-source agents that instead use 128k, and ‡ indicates that WebSailor-V2 additionally uses Google Scholar and a Python interpreter beyond web search and fetch. Baseline numbers are quoted from their respective papers.

4.2 Main Results

We evaluate DeepMiner across four challenging deep research benchmarks. As shown in Table 1, DeepMiner achieves substantial improvements over existing open-source agents, with particularly impressive results on the challenging BrowseComp benchmarks. On BrowseComp-en, DeepMiner-32B reaches 33.5% accuracy using only a 40k context, surpassing the much larger DeepSeek-V3.1-671B and approaching the recent WebSailor-V2 (35.3% on BrowseComp-en) despite the latter's 128k context and additional tools (Google Scholar and a Python interpreter). Notably, even our SFT model (21.2%) already outperforms many open-source agents, highlighting the effectiveness of our reverse-constructed training data before RL optimization. The performance gains extend consistently across BrowseComp-zh, XBench-DeepSearch, and GAIA, validating both the generalizability of our approach and the diversity of our automatically constructed training data. While gaps remain compared to leading proprietary systems, DeepMiner substantially narrows the divide between open-source and commercial deep research agents.

Furthermore, the progression from supervised fine-tuning to reinforcement learning demonstrates substantial and consistent improvements: +12.3 points on BrowseComp-en, +12.1 points on BrowseComp-zh, +9.0 points on XBench-DeepSearch, and +4.3 points on GAIA. The improvement magnitude correlates with task complexity—larger gains on benchmarks requiring extended multi-step reasoning (BrowseComp series) versus more modest gains on shorter-horizon tasks. This pattern confirms that our RL training effectively develops sophisticated search strategies for long-horizon interactions.

Moreover, we explore DeepMiner's potential under a longer context. We extend DeepMiner to 128k at inference using YaRN (Peng et al., 2023) with a scaling factor of 3.2.[2] Although the model is never trained at this length, this training-free extrapolation already lifts BrowseComp-en accuracy from 33.5% to 39.2%. This indicates that DeepMiner's context-management strategy scales gracefully with a larger context budget.

4.3 Data Construction Effectiveness

A useful data construction method should produce questions that are both genuinely difficult and effective for training. We therefore evaluate DeepMiner data against a broad set of existing construction methods along two axes: the *intrinsic difficulty* of the questions, and the *downstream performance* of agents trained on them. For difficulty, we measure the solve rate of GPT-OSS-120B and the average tool-calling turns it requires on each source's questions. For downstream effectiveness, we train Qwen3-32B under identical distillation and SFT pipelines on 500 questions[3] from each source with sufficient released data. We compare against HotpotQA (Yang et al., 2018) and recent web-research synthesis methods (Tao et al., 2025; Lu et al., 2025b; Liu et al., 2025a; Li et al., 2025b).

Difficulty. As shown in Table 2, DeepMiner data is the most challenging among all compared methods: GPT-OSS-120B solves only 37.2% of its questions and requires the most tool turns (26.0 on average), versus solve rates between 47% and 72% and

[2]The intrinsic context length of our base model Qwen3-32B is 40k, which makes it infeasible to train a 128k variant directly; we therefore apply YaRN only at inference.

[3]Limited by the size of WebShaper's publicly released dataset.

	Difficulty		Downstream Accuracy			
Data Source	**ACC**	**Turns**	**BC**	**BCZH**	**XBench**	**GAIA**
HotpotQA	72.3	8.7	7.0	8.7	49.0	39.8
WebShaper	61.4	20.9	13.7	22.1	**50.0**	48.5
DeepDive	55.3	16.5	14.7	18.0	49.0	49.5
WebExplorer	47.0	23.3	–	–	–	–
WebSailor	50.0	20.8	–	–	–	–
DeepMiner	37.2	26.0	**16.3**	**23.9**	**50.0**	**55.3**

Table 2: Data construction comparison. *Difficulty*: GPT-OSS-120B accuracy (lower is harder) and average tool turns (higher is harder, max=40) on each source's questions. *Downstream*: accuracy after training Qwen3-32B on 500 questions per source under identical distillation and SFT pipelines. WebExplorer and WebSailor release too few questions for the controlled training comparison, so we report only their difficulty.

fewer turns elsewhere. This difficulty is genuine rather than artificial, as the much stronger GPT-5.4 (OpenAI, 2026) (82.7% on BrowseComp) still reaches 90% on DeepMiner questions, indicating they are hard yet solvable.

Downstream effectiveness. Training on DeepMiner data yields the strongest agent on every benchmark, beating HotpotQA by wide margins (+9.3 on BrowseComp, +15.5 on GAIA) and the recent synthesis methods under identical conditions (+2.6 over WebShaper and +1.6 over DeepDive on BrowseComp). Across methods, downstream performance closely tracks intrinsic difficulty: the easiest data produces the weakest agent and the hardest the strongest.

Together, these two perspectives show that DeepMiner data is both substantially harder and more effective than existing construction methods.

4.4 Context Management Efficiency

We evaluate the effectiveness of our sliding window mechanism against alternative context management strategies. We consider three strategies: (1) *Truncation*: removing the oldest turns when approaching the context limit; (2) *Summary*: compressing the entire context via an external summarization model when approaching the limit; and (3) *Sliding Window*: our proposed method that selectively compresses distant tool responses while preserving complete assistant reasoning traces. We evaluate under two settings: training-free, where GPT-OSS-120B is directly applied with each strategy without any fine-tuning; and training, where Qwen3-32B is separately trained with each strategy using identical DeepMiner data and hyperparameters.

Table 3 presents the results. The training-free setting provides a controlled comparison that isolates

Setting	Strategy	BC	BCZH	XBench	GAIA
Training-free GPT-OSS-120B	Truncation	30.7	36.3	67.0	70.9
	Summary	31.7	39.8	67.0	**71.8**
	Sliding Window	**33.3**	**40.5**	**68.0**	**71.8**
Training Qwen3-32B	Truncation	19.7	25.6	47.0	44.7
	Summary	20.3	27.0	**53.0**	52.4
	Sliding Window	**21.2**	**28.0**	**53.0**	**54.4**

Table 3: Context management strategy comparison.

the effect of context management strategies from training dynamics. Here, sliding window achieves 33.3% on BrowseComp compared to 30.7% for truncation and 31.7% summary, demonstrating the effectiveness of our strategy. The training setting further examines whether each strategy can be improved through explicit optimization. Despite giving each baseline the opportunity to adapt to its respective context management scheme, sliding window still outperforms truncation by 1.5 points on BrowseComp and 9.7 points on GAIA, confirming that the advantage of sliding window is fundamental rather than superficial.

4.5 Detailed Analysis

We analyze training dynamics, context scaling, and failure modes below. Further studies, including model-size scaling and sliding-window hyperparameter sensitivity, are deferred to Appendix G.

Training Dynamics. Figure 4 illustrates the training dynamics of our reinforcement learning process. Our trajectory length steadily increases with training steps while remaining within the 40k trajectory length limit. This improvement is reflected in the training rewards, which gradually increase from 0.45 to 0.60 and converging after 90 steps, indicating that our tasks are of sufficient difficulty for effective policy learning. We select the checkpoint at step 90 as the model for the main experiments. The in-domain rewards translate to advantages on evaluation benchmarks, where BrowseComp performance improves from 21.2% to 33.5%, confirming that our reverse-constructed tasks can stimulate the model to perform deep cognitive behaviors in long-horizon multi-turn scenarios.

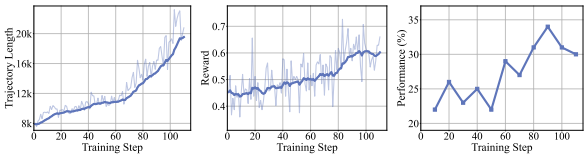

Figure 4: Training dynamics of DeepMiner during RL.

Context Scaling. We analyze how DeepMiner scales with tool-call budget and context length on

BrowseComp (Figure 5). Performance increases steadily with larger tool-call budgets, surpassing DeepSeek-V3.1-671B at around 60 calls and reaching 33.5 at 100 calls. With a standard 32k context length, DeepMiner sustains close to 100 rounds of tool interactions. This demonstrates that careful context management can be more effective than simply expanding capacity.

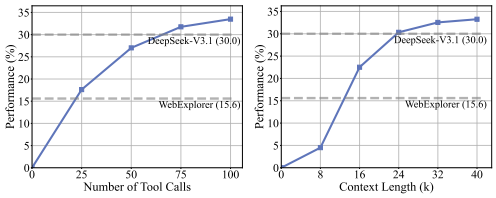

Figure 5: Scaling on tool call budget and context length.

Failure Modes. A natural concern with our design is whether dropping distant tool responses causes the agent to lose information it later needs. To examine this, we manually inspect 267 failed RL rollouts. Failures are overwhelmingly about search and reasoning rather than lost context: 40% never retrieve the target entity at all, and 32% retrieve it but fail in verification or answer formulation (e.g., granularity mismatches such as "Centre-Val de Loire" vs. "Loiret"). The remaining 28% of cases, where the target entity appears in a tool response but never enters the reasoning chain, are the ones most easily attributed to dropping; on inspection, however, they are recognition failures (the entity is present among many candidates but not prioritized) rather than consequences of dropping. Across all inspected trajectories we observe no failure clearly attributable to the discarding of a useful tool response, suggesting that our context management incurs little information loss in practice and supporting the locality assumption behind it (further evidence in Appendix H).

5 Related Work

Deep Search Agents. Deep search agents autonomously use search and browsing tools to solve complex information-seeking tasks (Xi et al., 2025). Proprietary systems (OpenAI, 2025a; Google, 2025; Claude Team, 2025) match or exceed human experts, but their closed architectures and data hinder reproducibility. Open-source efforts have concentrated on synthesizing sufficiently difficult training data: WebSailor (Li et al., 2025b) and DeepDive (Lu et al., 2025b) build questions over entity-relation knowledge graphs, WebShaper (Tao et al., 2025) formalizes construction via set theory, and ASearcher (Gao et al., 2025) and WebExplorer (Liu et al., 2025a) iteratively evolve queries to raise difficulty. DeepMiner instead synthesizes questions directly from open web pages through cross-document reasoning, without predefined graphs, and we show this data is both harder and more effective than existing synthetic datasets.

Context Management for Long-Horizon Agents. As interaction history accumulates, long-horizon agents quickly exhaust their context window. The most common remedy is summarization with an external model: WebSailor and ASearcher (Li et al., 2025b; Gao et al., 2025) compress tool outputs, while ReSum (Wu et al., 2026) periodically condenses the full trajectory with a trained summary model. Other approaches learn to fold salient content proactively (Li et al., 2025c; Sun et al., 2025) or maintain a compact memory through learnable memory tokens trained end-to-end (Zhou et al., 2025b; Yu et al., 2025a). In contrast, DeepMiner drops entire distant tool responses while preserving the complete reasoning trace. This simple strategy requires no external summarization model, is applicable both off-the-shelf and under end-to-end RL, and yet achieves strong performance.

We further discuss token-level context compression and RLVR in Appendix I.

6 Conclusion

We present DeepMiner, a framework for training long-horizon deep search agents. We design an efficient context-management strategy that drops distant tool responses while preserving the complete reasoning trace. Grounded in the observation that tool responses dominate context consumption yet mainly affect the immediate next few decisions, this strategy sustains nearly 100 interaction turns within a 32k context, needs no external summarization model, and integrates into end-to-end RL. To provide training signal that genuinely demands such long-horizon reasoning, we further construct high-difficulty, verifiable data through reverse construction from open web sources. Built on Qwen3-32B, DeepMiner-32B reaches 33.5% on BrowseComp with only a 40k context, surpassing the much larger DeepSeek-V3.1-671B. These results show that careful context management, paired with sufficiently difficult training data, is an effective and efficient path toward stronger open-source deep search agents.

Limitations

While DeepMiner demonstrates strong empirical results, our approach has several limitations that point to directions for future work. First, our sliding window drops tool responses by recency rather than semantic importance, so it relies on the model potentially having distilled decision-critical evidence into its preserved reasoning trace; when this fails, information in a dropped response cannot be recovered without re-invoking the tool. Content-aware, adaptive retention of salient tool outputs is a natural direction for future work. Second, our reinforcement learning adaptation assigns a single trajectory-level advantage to all subsequences, giving every context state the same learning signal regardless of its contribution; finer-grained credit assignment under dynamic context windows remains an open problem. Finally, both our data construction and evaluation rely on live web search, whose results evolve over time and pose challenges for exact reproducibility; we partially mitigate this by releasing a subset of our constructed data along with the evaluation prompts.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Anthropic. 2025. Introducing claude 4.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, and 1 others. 2025. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*.

Claude Team. 2025. Claude research.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl.

Google. 2025. Gemini Deep Research: your personal research assistant. Accessed: 2025-08-05.

Yiduo Guo, Zhen Guo, Chuanwei Huang, Zi-Ang Wang, Zekai Zhang, Haofei Yu, Huishuai Zhang, and Yikang Shen. 2025. Synthetic data rl: Task definition is all you need. *Preprint*, arXiv:2505.17063.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025a. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *Preprint*, arXiv:2509.13305.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025b. Websailor: Navigating super-human reasoning for web agent. *Preprint*, arXiv:2507.02592.

Xiaoxi Li, Wenxiang Zhang, Jiajie Jin, Yuanhao Zhu, Hongjin Wang, Jin Zhang, Zhicheng Zhu, Guanting Dong, Yujia Chen, Hongbo Sun, Yan Wang, Yutao Wang, Yong Chen, and Wayne Xin Zhao. 2025c. Deepagent: A general reasoning agent with scalable toolsets. *Preprint*, arXiv:2510.21618.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, and 1 others. 2025a. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. 2025a. Arpo:end-to-end policy optimization for gui agents with experience replay. *Preprint*, arXiv:2505.16282.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. 2025b. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *Preprint*, arXiv:2509.10446.

Metaso. 2025. Metaso. Accessed: 2025-08-05.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2025a. Introducing deep research. https://openai.com/index/introducing-deep-research/. Accessed: 2025-06-26.

OpenAI. 2025b. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-06-24.

OpenAI. 2026. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *Preprint*, arXiv:2309.00071.

Long Phan, Alice Gatti, Ziwen Han, and 1 others. 2025. Humanity's last exam. *Preprint*, arXiv:2501.14249.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. 2025. Scaling long-horizon llm agent via context-folding. *Preprint*, arXiv:2510.11967.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, and 1 others. 2025. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Preprint*, arXiv:2504.12516.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, and 1 others. 2025. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. 2026. Resum: Unlocking long-horizon search intelligence via context summarization. *Preprint*, arXiv:2509.13313.

Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. 2025. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges. *Preprint*, arXiv:2508.05668.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025a. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *Preprint*, arXiv:2507.02259.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and

16 others. 2025b. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Preprint*, arXiv:2306.14048.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *Preprint*, arXiv:2504.03160.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025a. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025b. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *Preprint*, arXiv:2506.15841.

A The Use of Large Language Models

We utilized LLMs to aid and polish writing.

B Ethical Considerations

Potential Risks. Our work releases a stronger open-source deep-search agent, which is net beneficial for the research community but could in principle be misused to aggregate public information about individuals at scale. To mitigate this, we restrict access to released artifacts to research use only through a formal request process.

Licenses. We use Qwen3-32B, GPT-OSS-120B, VERL, BrowseComp, BrowseComp-zh, XBench-DeepSearch, and GAIA in accordance with their respective licenses for research purposes. Our released data and model weights are distributed under access-controlled terms for academic research use only, consistent with the original access conditions of the underlying sources.

Privacy. Our training data is constructed from public web pages and Wikipedia entries about individuals with moderate visibility. To mitigate privacy risks, we restrict sources to Wikipedia and credible public web pages, filtering out social media and informal sources. Before public release, we perform anonymization on the released subset, replacing names and identifying information so that no real individual can be re-identified.

C Enhanced Tool Suite

Our agent operates with two primary tools, **Web Search** and **Fetch**, and a lightweight auxiliary **Find** tool, all designed for fine-grained web exploration:

1. **Web Search** takes text queries as input and returns document titles, URLs, and snippets from search results.

2. **Fetch** retrieves webpage content given a URL. Unlike existing approaches that truncate content or apply external summarization, our fetch tool implements paginated browsing that mimics human web navigation, allowing the model to assess initial content and decide whether to continue reading or exit.

3. **Find** enables in-page keyword search for lengthy webpages. The model can locate relevant information sections and the surrounding context before deciding which portions warrant detailed examination.

This design addresses a fundamental limitation where existing web agents suffer information loss through hard truncation or external summarization. By preserving complete information access while providing navigation flexibility, our tool suite enables fine-grained control over information gathering across extended interaction sequences.

D SFT Trajectory Construction

We generate cold-start trajectories with GPT-OSS-120B (Agarwal et al., 2025), allowing up to 40 turns per trajectory. We apply rejection sampling, keeping only trajectories whose final answer is correct, yielding approximately 3,000 trajectories with an average of 16 turns. Following the training-inference consistency principle described in Section 3.1, each trajectory is then segmented into multiple subsequences for SFT.

E Training Hyperparameters

For SFT, we use a batch size of 256 and a learning rate of 1×10^{-5}. The RL phase uses a batch size of 32, a learning rate of 2×10^{-6}, and 8 rollouts per question, and is implemented with the VERL framework. DeepMiner undergoes 110 RL steps in total, and we select the checkpoint at step 90 as the final model based on validation performance.

F Data Quality Analysis

F.1 Data Decontamination

Since our training data is constructed from web entities and pages, we examine potential overlap with the evaluation benchmarks. BrowseComp releases only question-answer pairs without exposing the underlying entities, so standard entity-level decontamination cannot be applied directly. Nonetheless, the risk of overlap is inherently low: the moderate-visibility criterion spans millions of Wikipedia entities across diverse domains, whereas our training set samples only a few thousand from this space. To verify this empirically, we use GPT-5.4 (OpenAI, 2026) (which attains 82.7% on BrowseComp, indicating strong entity-identification ability) to infer the underlying entities from 100 randomly sampled BrowseComp questions, and match them against our entire training entity pool. We find zero overlap.

F.2 Data Quality Verification

Because both question generation and filtering rely on LLMs, we additionally validate data quality with human annotators. First, we manually inspect 30 randomly sampled questions for ambiguity, answer verifiability, and solvability. Only one question exhibits a potential issue: a constraint phrased as a subjective expression ("disappointingly low tourism employment figures") that lacks an objectively verifiable threshold. Second, to confirm that the questions are genuinely difficult yet solvable by humans, three annotators each attempt 10 randomly sampled questions under a 30-minute budget per question, correctly answering 3, 3, and 4 questions respectively. Among the 20 failed attempts, the errors stem from ineffective retrieval queries (10), identifying an incorrect entity (6), and exceeding the time limit (4); none are attributable to flawed or under-constrained questions. These results indicate that our pipeline yields questions that are challenging for humans while remaining well-posed and answerable.

G Additional Experiments

G.1 Evaluation Judge Reliability

In our setting, the LLM judge is used in a narrow, well-defined way: it does not assess the quality of an entire reasoning trajectory, but only checks whether the agent's final answer is equivalent to the ground-truth answer. Because benchmark answers are short, unambiguous spans (e.g., "April 1" or "Fall River"), this reduces to a simple semantic match that is highly reliable for a strong model such as ChatGPT-4o-Latest. The same LLM-as-judge protocol is standard practice in recent benchmarks (Wei et al., 2025; Phan et al., 2025). To quantify reliability, we manually inspect 100 predicted-correct and 100 predicted-incorrect cases. The judge matches human verdicts on 100% of the predicted-incorrect cases and 98% of the predicted-correct cases. The few disagreements are conservative mismatches caused by overly strict string matching (e.g., "Compassionate Pugilist" vs. "The Compassionate Pugilist", or "Glass Furnace Way" vs. "Glass Furnace Road"), which introduce negligible variance and do not bias RL training.

G.2 Sliding-Window Hyperparameter Sensitivity

We study sensitivity to the window size $\mathcal{W}$ and slide size $\mathcal{S}$ from two angles.

Inference-time generalization. Although DeepMiner is optimized with fixed $\mathcal{W} = 5$ and $\mathcal{S} = 3$, we evaluate its zero-shot generalization across inference-time settings on a 100-sample BrowseComp subset. Table 4 reports DeepMiner-32B with $\mathcal{W}$ from 3 to 6 and $\mathcal{S}$ from 1 to 5; performance stays between 30% and 38% across settings, indicating the model does not overfit to specific window parameters and that $\mathcal{W}$ and $\mathcal{S}$ can be adjusted at deployment without retraining.

$\mathcal{W} \setminus \mathcal{S}$	1	2	3	4	5
3	32	37	-	-	-
4	33	30	38	-	-
5	33	33	34	31	-
6	31	32	34	34	35

Table 4: The results of DeepMiner under different window sizes and slide sizes. "-" indicates the setting is invalid.

Training-time configuration. We also train separate models with different window configurations

and evaluate with matching inference settings (Table 5). Performance variations are modest, confirming the approach is not sensitive to specific hyperparameter choices.

Window	Slide	BC	BCZH	XBench	GAIA
4	2	21.0	30.8	49.0	51.5
5	2	22.3	27.3	49.0	52.4
5	3	21.2	28.0	53.0	54.4
6	3	22.7	29.8	54.0	52.4

Table 5: Performance across different training-time sliding window configurations.

G.3 Model Size Scaling

To assess how model capacity affects performance, we evaluate DeepMiner across the Qwen3 family (4B, 8B, 14B, and 32B). As shown in Table 6, performance scales consistently with model size on all benchmarks; even the smallest Qwen3-4B achieves reasonably strong results on BrowseComp, indicating that DeepMiner data provides a robust training signal across a wide range of model capacities.

Model Size	BC	BCZH	XBench	GAIA
4B	14.3	13.8	35.0	38.8
8B	15.7	18.3	35.0	43.7
14B	20.0	27.0	47.0	53.4
32B	21.2	28.0	53.0	54.4

Table 6: Scaling on model size with the Qwen3 family.

G.4 Tool Set Ablation

To assess whether our results depend on a richer tool set than prior agents, we ablate the auxiliary *find* tool, a lightweight in-page keyword locator analogous to Ctrl+F. Removing it changes BrowseComp accuracy only marginally (33.5 $\rightarrow$ 32.8), confirming its negligible impact. We further note that our *fetch* tool returns raw page content, whereas some agents (e.g., WebSailor's *visit*) employ an LLM to summarize page content—an implicit form of context compression. DeepMiner therefore operates on uncompressed tool responses, making its evaluation setting, if anything, less favorable than these baselines.

H Analysis of Tool Response Influence

A key assumption underlying our sliding window mechanism is that tool responses primarily influence the model's immediate next few decisions, with diminishing impact on distant future outputs. This section provides empirical evidence supporting this assumption through multiple complementary analyses.

Perturbing Distant Tool Responses. To test this assumption directly, we run a controlled inference-time perturbation. Each time the window slides, in addition to the usual dropping, we substitute a single out-of-window tool response with an unrelated one sampled from a different trajectory. We do not enforce tool-type consistency, so a *web_search* result may be replaced by *fetch* output for an unrelated query; the injection is thus mismatched in both content and structure and, unlike our placeholder, carries no signal that it should be ignored, making it a strictly harder perturbation than the dropping our mechanism performs. We perturb a single response rather than all of them by design, as a controlled, minimal intervention that isolates the causal effect of one distant response's content; replacing all out-of-window responses at once would instead reintroduce context overflow and confound length with content. As shown in Table 7, performance changes only marginally. This indicates that the deployed agent's decisions are largely invariant to the content of distant tool responses, relying on recent responses together with its preserved reasoning trace rather than the raw text of distant ones. Since dropping with a placeholder is a weaker perturbation than the misleading, type-mismatched injection tested here, replacing distant responses with placeholders is correspondingly safe.

Window Size Insensitivity. Further evidence comes from our hyperparameter analysis in Section G.2. Reducing the window size from 6 to 3 causes minimal performance variation. If early tool responses were critical for current decisions, smaller windows should cause significant degradation. This insensitivity suggests that the model primarily relies on information from the recent few tool responses.

Error Analysis. Finally, our manual inspection of the 267 failed trajectories finds almost no evidence that errors were primarily caused by the omission of early tool responses.

These three lines of evidence—selective summarization, window size insensitivity, and error analysis—collectively confirm that the sliding window mechanism does not introduce systematic information loss. Tool responses primarily serve as im-

Method	BC	BC-zh	XBench	GAIA
Standard Inference	33.5	40.1	62.0	58.7
Corrupted Tool Responses	33.0	38.8	60.0	58.2

Table 7: Test-time perturbation. Substituting a distant (out-of-window) tool response with unrelated, type-mismatched content causes only marginal degradation, indicating that the deployed agent is largely invariant to the content of distant tool responses.

mediate context for the next reasoning step. Their long-term utility is superseded by DeepMiner's implicit summarization, further justifying our design choice of discarding distant tool responses while preserving all assistant responses. Finally, we acknowledge that no context management strategy is completely lossless. But we must make a trade-off between interaction rounds and information completeness. The experiments in Section 4.4 confirm DeepMiner's trade-off to be effective—sliding window consistently outperforms both truncation and summarization approaches.

I Extended Related Work

Token-level Context Compression. A separate line of work reduces context cost at the token or attention level within a single forward pass. Sparse-attention models such as Longformer (Beltagy et al., 2020) restrict attention to local windows to extend the tractable sequence length, while KV-cache eviction methods (Xiao et al., 2023; Zhang et al., 2023) drop or retain key-value entries to bound memory during generation. These methods operate on the model's internal token representations and are largely agnostic to the semantics of what is removed. DeepMiner's sliding window is fundamentally different and complementary: it operates at the level of complete tool responses in the agent's prompt across interaction turns, drops them by a deterministic rule while preserving the full assistant reasoning trace, and is kept consistent between training and inference so that it can be optimized end-to-end under reinforcement learning.

Reinforcement Learning with Verifiable Rewards. Reinforcement learning with verifiable rewards has become the dominant paradigm for eliciting long-horizon reasoning. Most approaches build on GRPO (Shao et al., 2024) and its variants such as DAPO (Yu et al., 2025b) and ARPO (Lu et al., 2025a), and apply outcome-based rewards to train search agents (Song et al., 2025; Jin et al., 2025; Guo et al., 2025). DeepMiner follows this paradigm but adapts the advantage computation to dynamic context: each trajectory is decomposed into sliding-window subsequences, and the trajectory-level advantage is propagated to all of them (Section 3.1), which keeps the group-relative estimate intact while enabling training under continually changing context states.

J Judgment Templates

Below is the judgment template we used.

Judgment Template

Judge whether the following [response] to [question] is correct or not based on the precise and unambiguous [correct_answer] below.

[question]: question

[response]: response

Your judgment must be in the format and criteria specified below: extracted_final_answer: The final exact answer extracted from the [response]. Put the extracted answer as 'None' if there is no exact, final answer to extract from the response.

[correct_answer]: answer

reasoning: Explain why the extracted_final_answer is correct or incorrect based on [correct_answer], focusing only on if there are meaningful differences between [correct_answer] and the extracted_final_answer. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than [correct_answer], focus only on whether the answers match.

correct: Answer 'yes' if extracted_final_answer matches the [correct_answer] given above, or is within a small margin of error for numerical problems. Answer 'no' otherwise, i.e. if there if there is any inconsistency, ambiguity, non-equivalency, or if the extracted answer is incorrect.

K Details of Question Construction Pipeline

K.1 Models Used to Synthesize Data

We primarily use Qwen3-235B-A22B-Thinking-2507 and DeepSeek-R1-0528 to construct questions. In the document filtering stage, for cost and performance considerations, we only use Qwen3-235B-A22B-Thinking-2507 to filter web documents. When generating questions, we utilize Qwen3-235B-A22B-Thinking-2507 and DeepSeek-R1-0528 to enhance question diversity. For question filtering, we prompt both models with each question separately, and if either mentions the answer, we drop that question.

K.2 Document Filtering

After collecting documents, each of them undergoes a three-stage filtering to ensure quality, including entity correspondence verification, information complementarity assessment, and credibility validation. Detailed templates are as follows:

Document Filter Template (Entity Correspondence Verification)

[Task]
Evaluate the quality and suitability of the provided webpage for further processing. Determine whether it is directly related to the specific entity described by the provided Wikidata.

[Wikidata]
{wikidata}

[Web Content]
Link: {link}
Content: {content}

[Evaluation Criteria]
Assess the entity relevance of this page:
- Verify that the webpage explicitly and accurately focuses on the exact entity described by Wikidata.
- Confirm it's not about another individual with the same or similar name.
- The main content should be text and accessible by request, avoiding web pages that are mainly about videos, images, pdf, files, etc.

[Output Format]
Provide a detailed analysis and output your final assessment as follows:
Acceptable: {{yes or no}}

Document Filter Template (Information Complementarity Assessment)

[Task]
Evaluate the quality and suitability of the provided webpage for further processing. Determine whether it contains specific, valuable, and credible information about the entity.

[Wikidata]
{wikidata}

[Web Content]
Link: {link}
Content: {content}

[Evaluation Criteria]
Assess the information complementarity of this page: - Contains substantial new information not present in Wikidata.
- Confirm the webpage provides specific details, insights, or data that enrich or supplement the information present in the Wikidata entry.
- Check for unique or exclusive information, direct quotes, primary source material, or additional context not covered by Wikidata.
- The main content should be text and accessible by request, avoiding web pages that are mainly about videos, images, pdf, files, etc.

[Output Format]
Provide a detailed analysis and output your final assessment as follows:
Acceptable: {{yes or no}}

Document Filter Template (Credibility Validation)

[Task]
Evaluate the quality and suitability of the provided webpage for further processing. Determine whether it is formal, authoritative, believable, and trustworthy.

[Wikidata]
{wikidata}

[Web Content]
Link: {link}
Content: {content}

[Evaluation Criteria]
Assess the credibility of this page: - Evaluate the formality, reliability, and trustworthiness of the content.
- Consider the source's authority (e.g., official organizations, reputable media outlets, recognized academic institutions, professional profiles).
- Avoid informal sources such as social media platforms, forums, or websites with questionable authenticity.
- The main content should be text and accessible by request, avoiding web pages that are mainly about videos, images, pdf, files, etc.

[Output Format]
Provide a detailed analysis and output your final assessment as follows:
Acceptable: {{yes or no}}

K.3 Question Generation

We present the core prompt for question generation. Some question examples and context cases have been omitted.

Question Generation Template

You will be provided with multiple documents from different sources about an entity. You need to generate extremely high-difficulty retrieval questions to test users' search abilities using search engines only. The purpose is to test users' search engine retrieval skills. Users will only see the question text and must find answers through web searches - they have no access to the provided documents.

Core Requirements:

1. Cross-source integration: Questions must combine information from at least 4 different source documents, and this question cannot be answered solely through the content of any single document.

2. Difficulty through hard-to-retrieve features: The questions should use characteristics that are harder to search for directly, avoiding direct keyword approaches.
- Each clue should be necessary and sufficient, avoid redundant descriptive details
- Heavily reduce use of easily searchable proper nouns, job titles, and location names
- Use numerical ranges, distances, time periods, and other measurable constraints rather than descriptive narratives

3. Feature obfuscation: Obfuscate features appearing in questions to increase difficulty:
- "1905" → "early 20th century"
- "born in Manhattan" → "born in New York"

4. Answer specificity: The answer cannot be the entity itself, but should be related attributes, dates, people, etc. The answer must maintain uniqueness and specificity - avoid questions like "Who are this person's colleagues?" because there may be multiple valid answers.

5. Unique, verifiable answers: Each question must have one clear, precise answer that can be definitively verified. Ensure the combination of features is unique

enough that only one entity can satisfy all the specified characteristics, minimizing the possibility of multiple valid answers.

Example Questions:
(omitted example questions)

Ket Entity: {entity name}

Documents: {documents}

Now, you need to generate at least 2 questions.

K.4 Question Filtering

We present the core prompt used by the most effective filter during question filtering, which identifies solvable questions based on source pages and ensures the answer is derivable from the given reference documents. Some question examples and context cases have been omitted.

Question Filter Template

You need to determine whether the given question-answer pair can be verified based on the provided document content. This is a retrieval question verification task, focused on ensuring the question does not mislead searchers to find incorrect answers.

[Question]
{question}

[Provided Answer]
{answer}

[Document Content]
{document}

Task Background

These are complex retrieval questions with the following characteristics:
- Use multiple constraints combined to lock onto a unique answer
- Deliberately obscure certain information to increase retrieval difficulty
- Searchers need to find relevant documents through search engines, then reason to get the answer
- Your task is to ensure the question does not mislead searchers to find wrong answers

Core Verification Principles

1. Multi-Condition Combination Principle
Individual constraints can be vague, but all constraints combined must point to a unique answer. Don't reject questions just because a single condition isn't precise—evaluate the overall combined effect.

2. Factual Accuracy Principle
Every description in the question must be objectively factually correct. Allow reasonable transitions from specific to abstract, but absolutely prohibit any factual errors.

3. Temporal Context Principle
Documents often contain time-sensitive statements from their writing period. Questions must account for the document's temporal context, not assume current validity.

Verification Process

Step 1: Temporal Context Analysis
- Identify when the document was written
- Note any time-sensitive claims ("currently", "now", "recently", "latest", "this year")
- Check if the question inappropriately transfers temporal language

Step 2: Individual Constraint Verification
- Verify each constraint type following the guidelines above
- Look for factual errors, unreasonable abstractions, or category confusion

Step 3: Answer Verification
- Confirm the document contains information needed to answer the question
- Verify the provided answer is consistent with document content

Step 4: Risk Assessment
- Assess if the question could mislead searchers to find wrong answers
- Identify potential ambiguities that could cause confusion

Final Verification Checklist

- [ ] Are there any factual errors in the question?
- [ ] Does the document contain the core information needed to answer the question?
- [ ] Are there any temporal context mismatches?
- [ ] Do time-sensitive claims in the question match the document's temporal context?

Output Format

Provide your analysis covering:
1. **Temporal Context Analysis**: Document timeframe and time-sensitive claims
2. **Individual Constraint Verification**: Check each constraint for accuracy
3. **Answer Verification**: Confirm the document supports the answer
4. **Risk Assessment**: Identify potential issues (factual errors, ambiguity, temporal mismatches)

Final Judgment: Acceptable / Unacceptable

Reasoning: [2-3 sentences explaining your decision, highlighting key factors and any temporal context issues]

K.5 Generated Questions

We show several representative questions below. We present the question, the core entity, and the answer. We show a subset of DeepMiner data in the supplementary material.

Question Examples

Question Example 1

Question:
A journalist who graduated from an Ivy League institution in the 1990s and later worked as a White House correspondent covering national security topics joined a major American newspaper between 2018 and 2020. This journalist previously worked at a political news organization where they served as an editor focusing on both domestic executive branch coverage and international affairs. During their tenure at this political publication, they wrote a piece with a provocative headline suggesting a connection between a major world leader and a specific presidential candidate. In a different role at a weekly news magazine between 2009 and 2015, this same journalist served in a senior position covering international relations while also holding a deputy leadership role in that publication's Washington office. What is the birth month and day of this journalist?
Entity: Michael Crowley (journalist)
Answer: April 1

Question Example 2

Question:
An academic who transitioned from directing a public affairs institute in the Midwest to a Pennsylvania university position co-authored a book about religious themes in the late 2000s with a Pulitzer Prize-winning author. This academic appeared on a PBS program in early 2014 and published a memoir in the early 2020s that was reviewed by someone surnamed Begiebing. The memoir explores his grandfather's gambling problems. What New England city did this academic leave for college?
Entity: J. Michael Lennon
Answer: Fall River

Question Example 3

Question:
A basketball player who missed an entire season in the late 2010s due to a leg injury later earned a major academic honor at the Division I level between 2020 and 2023. This player's father competed at a university whose basketball team reached the NCAA tournament in the early 1990s. The player transferred to an Atlantic Coast

Conference school for their final collegiate season. What month and day was this player born?
Entity: Ben Vander Plas
Answer: September 19

L Case Study

The following is a case trajectory of DeepMiner on BrowseComp.

A Case Trajectory on BrowseComp

Question

I am looking for the name of a historical place that meets the following criteria: 1. As of 2023, the place is located in the capital city of a country. 2. It is situated beside a river as of 2023. 3. Its construction began between 1830 and 1860 (inclusive). 4. The construction was completed between 1870 and 1880 (inclusive). 5. The thickness of its walls ranges from 0.5 to 0.9 meters (inclusive). 6. It was acquired by the government of the country between 1980 and 1990(inclusive). 7. This place was once damaged by a tornado between 1880 and 1890(inclusive). 8. It also suffered damage from an earthquake between 1890 and 1900(inclusive). 9. The president of the country at the time of its acquisition was born between 1920 and 1935(inclusive).

Step 1

Analysis:

Search results not obvious. Perhaps we need to think of specific known historical places that were damaged by tornado and earthquake.
Tornado damage to a historical site in late 1800s: maybe the Alamo? No tornado.
Earthquake early 1900s: maybe the 1906 San Francisco earthquake. Tornado damage earlier? Not likely.
Capital city beside a river: many.
Acquired by government between 1980-1990: maybe nationalization of a historic house.
President at acquisition time born 1920-1935: which president? For example, US President Ronald Reagan (born 1911) no. US President George H.W. Bush (born 1924). If acquisition happened during his presidency (1989-1993). If acquisition was 1989, president George H.W. Bush (born 1924). So US capital Washington D.C. Beside Potomac River. Historical place built 1830-1860 start, completed 1870-1880, walls 0.5-0.9m thick. Damaged by tornado 1880-1890 in Washington D.C.? There was a tornado in Washington D.C. in 1884? I recall a tornado in 1884 that damaged the Smithsonian? Let's search.

Tool Call:

"name": "web_search", "arguments": "queries": ["Washington D.C. tornado 1884 historical building damaged", "tornado 1885 Washington DC historic site", "earthquake 1897 Washington DC historic building damaged"]

Tool Response:

Tool responses are omitted

Step 2

Analysis:

Seems not obvious.
Perhaps it's not US. Another capital city with tornado and earthquake: maybe Islamabad? Tornado unlikely.
Maybe it's a European capital with tornado? Tornadoes rare in Europe but can occur. Earthquake early 1900s.
Capital city beside a river: many.
Let's think of historical place like a palace or fortress.
Construction started 1830-1860, completed 1870-1880, walls 0.5-0.9m thick.
Acquired by government 1980-1990.
Damaged by tornado 1880-1890, earthquake 1890-1900.
Which country's president at acquisition time born 1920-1935? If it's a president (i.e., head of state is president), not necessarily US.
Could be a country where the capital is also named after the country? Eg. Mexico City (capital of

Mexico). Tornado damage in Mexico City? Unlikely.
Maybe it's a capital in South America: e.g., Lima (Peru) beside Rímac River. Tornado damage? Unlikely.
Buenos Aires (Argentina) beside Riachuelo. Tornado? Not typical.
Brasília (Brazil) no river.
Ottawa (Canada) beside Ottawa River. Tornado in Ottawa? There was a tornado in Ottawa in 1888? Not sure.
President of Canada? Canada is a constitutional monarchy, no president.
So it's a presidential republic.
Maybe it's a small country.
Tornado damage in late 1800s could be in Bangladesh (Dhaka) but tornado rare.
Perhaps it's in Africa: capital city beside a river, tornado? Unlikely.
Could be in Central America: e.g., Guatemala City beside Paz River? Tornado?
Maybe it's in the Caribbean: e.g., Port-au-Prince (Haiti) beside Artibonite? Tornado?
Let's search for historical buildings damaged by tornado and earthquake.

Tool Call:
"name": "web_search", "arguments": "queries": ["'tornado' 1880' earthquake' 1890' historic building' capital'", "'tornado' 1885' earthquake' 1895' historical site' capital'", "'tornado' 1890' earthquake' 1897' historical building' capital'"]

Tool Response:
Tool responses are omitted

Step 3

Analysis:
Maybe the tornado and earthquake are not in the same location? The place was damaged by a tornado between 1880-1890, and also suffered damage from an earthquake between 1890-1900.
So two separate events damaging the same place.
Thus the place survived both.
We need to find such a place.
Let's think of famous historical places that have been damaged by tornado and earthquake.
Possibly the "Kashmir Fort" ?
Search for historical site tornado damage 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["historical site tornado damage 1880s", "tornado damage to historic building 1885", "tornado 1887 historic building damage"]

Tool Response:
Tool responses are omitted

Step 4

Analysis:
Search not effective.
Maybe the place is not well-known.
We might need to think of a specific country where tornadoes are common: USA, Bangladesh, Canada (some), Argentina (rare), etc.
If it's USA, capital Washington D.C. Historical place beside a river (Potomac). Construction started 1830-1860, completed 1870-1880. Wall thickness 0.5-0.9m.
Damaged by tornado 1880-1890 in D.C. Which building? Possibly the United States Capitol? It was damaged by tornado? Not sure.
Check if US Capitol was damaged by tornado. Search.

Tool Call:
"name": "web_search", "arguments": "queries": ["Washington DC historic building tornado 1884 damage", "tornado damage to Smithsonian 1885", "tornado damage to National Gallery of Art 1880s"]

Tool Response:
Tool responses are omitted

Step 5

Analysis:
Seems not.
Maybe the historical place is not in US. Let's think of other countries where tornadoes occur: Bangladesh (tornadoes called "sudden wind"), Pakistan, India (tornadoes in some regions), Argentina (southeast), Brazil (rare), etc.
Capital city beside a river: e.g., Dhaka (Bangladesh) beside Buriganga River. Historical place built 1830-1860 start, completed 1870-1880, walls 0.5-0.9m thick. Damaged by tornado 1880-1890 in Dhaka? Not sure about tornado.
Earthquake 1890-1900 in Dhaka? There was an earthquake in 1897 (Rangpur earthquake) affecting Dhaka? Not sure.
Acquired by government 1980-1990, president at that time born 1920-1935. Bangladesh president? Bangladesh is a parliamentary republic, head of state is President (since 1971). Presidents: Sheikh Mujibur Rahman (1971-75, born 1920). He was president at acquisition? If acquisition happened 1980s, president would be not Mujib (he died 1975). So not Bangladesh.
Maybe it's Pakistan: capital Islamabad beside River? Islamabad is not on a river. Karachi is not capital. Islamabad near River Soan? Actually Islamabad is near River Soan? Not sure.
Tornado in Islamabad? Unlikely.
Earthquake in Pakistan 1890s? The 1897 Quetta earthquake (1897). Not capital.
President of Pakistan in 1980s: Zia-ul-Haq (military ruler, not president). President Ghulam Ishaq Khan (1988-1993, born 1915). Not within 1920-1935.
India: capital New Delhi beside Yamuna River. Historical place built 1830-1860 start, completed 1870-1880, walls thickness 0.5-0.9m. Damaged by tornado 1880-1890 in Delhi? Unlikely tornado in Delhi.
Earthquake 1890-1900 in Delhi? There was an earthquake in 1897? Not sure.
Acquired by government 1980-1990, president at acquisition time (President of India) born 1920-1935. Presidents of India: Zail Singh (1982-1987, born 1916). R. Venkataraman (1987-1992, born 1918). Not within 1920-1935.
So not India.
Argentina: capital Buenos Aires beside Riachuelo. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado 1880-1890? Buenos Aires tornado rare.
Earthquake 1890-1900? Not.
President at acquisition (1980s) born 1920-1935: President Raúl Alfonsín (1983-1989, born 1927). Could be.
Historical place in Buenos Aires damaged by tornado and earthquake? Not likely.
Maybe it's a smaller country.
Consider Caribbean: Haiti capital Port-au-Prince beside Artibonite? Tornado rare.
Jamaica capital Kingston beside Kingston Harbour (not river). Tornado?
Let's think of tornadoes in Europe: tornadoes do occur, e.g., in Germany, Poland, UK.
Capital city beside a river: e.g., Warsaw (Poland) on Vistula River. Historical place built 1830-1860 start, completed 1870-1880, walls thickness 0.5-0.9m. Damaged by tornado 1880-1890 in Warsaw? Possibly a tornado hit Warsaw in 1886? Not sure.

Earthquake 1890-1900 in Warsaw? Earthquake in 1898? Not typical.
Acquired by government 1980-1990, president of Poland at that time (Poland is a republic with president) born 1920-1935. Presidents: Wojciech Jaruzelski (1989-1990, born 1923). Could be.
Historical place in Warsaw damaged by tornado and earthquake? Not aware.
Let's search for tornado in Warsaw 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["Warsaw tornado 1885", "Warsaw tornado 1890", "Warsaw historical building tornado damage 1880s"]

Tool Response:
Tool responses are omitted

Step 6

Analysis:
Seems not helpful.
Maybe the place is not well-known, need to think of specific historical place: a palace, fortress, monastery, etc.
Let's consider countries where president (head of state) is relevant: e.g., Mexico (president), Brazil (president), etc.
Capital city beside a river: Mexico City beside Rio Grande? Actually Mexico City is near Rio Pánuco? Not a river.
Brazil capital Brasília not on a river.
Argentina capital Buenos Aires beside Riachuelo.
Historical place built 1830-1860 start, completed 1870-1880, walls thickness 0.5-0.9m.
Damaged by tornado 1880-1890 and earthquake 1890-1900.
Acquired by government 1980-1990.
President at acquisition time born 1920-1935.
If it's Argentina, president in 1980s: Raúl Alfonsín (born 1927). So acquisition during his presidency (1983-1989). Historical place in Buenos Aires damaged by tornado and earthquake? Unlikely.
But maybe the place is in another Argentine city that is capital of a province, not national capital? The question says capital city of a country.
Let's think of African countries with tornadoes: e.g., Zimbabwe, Mozambique, South Africa (tornadoes rare). Earthquake early 1900s.
President at acquisition time born 1920-1935.
Historical place in capital city beside a river.
Maybe it's a castle in Kenya? Nairobi beside Nairobi River. Historical place built 1830s? Not.
Tornado in Nairobi? Unlikely.
Earthquake in Kenya early 1900s? Not.
Let's consider Caribbean: Haiti capital Port-au-Prince beside Artibonite? Actually Port-au-Prince is near the Léogâne River? Not sure.
Historical place built 1830-1860 start, completed 1870-1880. Could be a fortress like Citadelle Laferrière (Haiti) built 1805-1840, not.
Tornado damage? Not.
President of Haiti in 1980s: Jean-Claude Duvalier (born 1951). Not.
Let's think of Middle East: capital city beside a river, tornado? Tornadoes in Middle East rare.
Maybe it's a place in Pakistan: capital Islamabad not on river, but Karachi not capital.
Historical place in Islamabad? Not.
Maybe it's a place in Sri Lanka: capital Colombo beside Kelani River. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado? Unlikely.
President of Sri Lanka (head of state) in 1980s: J.R. Jayewardene (president 1978-1989, born

1906). Not.
Let's consider European countries with tornadoes: tornadoes occur in UK (e.g., Birmingham). Capital London beside Thames.
Historical place in London built 1830-1860 start, completed 1870-1880, walls thickness 0.5-0.9m. Damaged by tornado 1880-1890 in London? There was a tornado in London in 1884? I recall a tornado in 1885 near London? Not sure.
Earthquake 1890-1900 in London? Earthquake in 1896? Not typical.
Acquired by government 1980-1990, president of UK? UK is constitutional monarchy, not president.
Thus not UK.
Maybe it's a country with a president and capital city with tornado and earthquake: e.g., Philippines capital Manila beside Pasig River. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado 1880-1890 in Manila? Tornadoes rare in Philippines (tropical cyclones). Earthquake 1897? The 1897 earthquake in Manila? Not sure.
President of Philippines in 1980s: Ferdinand Marcos (born 1917). Not within 1920-1935.
Corazon Aquino (president 1986-1992, born 1933). Yes, born 1933. Acquisition could be during her presidency (1986-1992). Historical place in Manila damaged by tornado and earthquake? Tornado in Manila? There was a tornado in Manila in 1880? Not aware.
Earthquake in Manila 1897? The 1897 earthquake in Manila? There was a major earthquake in 1897 in Philippines (Mindanao?). Not sure.
Let's search for historical building in Manila damaged by tornado 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["Manila tornado 1885 historical building damage", "Manila earthquake 1890 historical building damage", "Manila historical site tornado 1880s earthquake 1890s"]

Tool Response:
Tool responses are omitted

Step 7

Analysis:
Result 1: Fort Santiago front structure destroyed by earthquake of 1880. Fort Santiago is a historical fortress in Intramuros, Manila, beside Pasig River? Intramuros is within Manila, beside Pasig River. Fort Santiago construction started? Let's check.

Tool Call:
"name": "fetch", "arguments": "url": "https://en.wikipedia.org/wiki/Fort_Santiago", "page": 1

Tool Response:
Tool responses are omitted

Step 8

Analysis:
Fort Santiago construction started 1590, not within 1830-1860. So not.
We need construction start 1830-1860 inclusive. So maybe a more recent building.
Historical place in capital city beside a river, damaged by tornado and earthquake in late 19th century, acquired by government 1980-1990, walls thickness 0.5-0.9m.
Let's think of a specific known building that matches tornado and earthquake damage.
Tornado damage to a historic house in Washington D.C. maybe the "Old Post Office Pavilion"? Not.
Earthquake damage in late 19th century in D.C.? The 1897 earthquake? Not.

Maybe it's not US.
Consider Canada: capital Ottawa beside Ottawa River. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado 1880-1890 in Ottawa? There was a tornado in Ottawa in 1888? Not sure.
Earthquake in Ottawa 1890-1900? The 1896 earthquake in Ottawa? Not typical.
President of Canada? No president.
Let's think of a country with president: e.g., Indonesia (president), capital Jakarta beside Ciliwung River. Historical place built 1830-1860 start, completed 1870-1880, damaged by tornado? Tornado in Jakarta? Unlikely.
Earthquake 1890-1900 in Jakarta? The 1899 Banten earthquake? Not Jakarta.
President at acquisition time (1980s) born 1920-1935: Indonesian president Suharto (born 1921, president 1967-1998). Acquisition could be during his presidency (1980s). Historical place in Jakarta damaged by tornado? Not.
Maybe it's a palace in Thailand: capital Bangkok beside Chao Phraya River. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado? Thailand tornado rare.
Earthquake 1890-1900 in Bangkok? Not.
President? Thailand is monarchy.
Thus not.
Maybe it's a country in South America with president: e.g., Brazil (president), capital Brasília (no river?), but Brasília not on river. Previous capital Rio de Janeiro (beside Guanabara Bay). Historical place built 1830-1860 start, completed 1870-1880, damaged by tornado? Tornado in Rio? Unlikely.
President of Brazil in 1980s: José Sarney (born 1930). He was president 1985-1990. Born 1930 within 1920-1935. So acquisition during his presidency. Historical place in Rio de Janeiro damaged by tornado and earthquake? Not.
But maybe it's a historical place in Brasília? No river.
Historical place in Rio de Janeiro beside a river (e.g., Rodrigo de Freitas Lagoon? Not river). Maybe the "Palácio do Catete" (Presidential palace) built 1894? Not.
Let's search for historical building in Rio damaged by tornado 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["Rio de Janeiro historical building tornado 1880s", "Rio de Janeiro earthquake 1890s historical building damage", "Rio de Janeiro palace tornado 1885"]

Tool Response:
Tool responses are omitted

Step 9

Analysis:
Result 3: NYTimes article about tornado at Rio in 1864. That's a tornado in 1864, outside our range (we need tornado 1880-1890). But maybe there was a tornado in Rio in 1885? Not sure. Nevertheless, the Paço Imperial built in 18th century, not.
Let's open result 3.

Tool Call:
"name": "fetch", "arguments": "url": "https://www.nytimes.com
/1864/12/02/
archives/tornado-at-rio-janeiro-terrible-hail-storm-great-damage-to-the-city.html", "page": 1

Tool Response:
Tool responses are omitted

1216

Step 10

Analysis:

Probably blocked.
But the tornado in Rio in 1864 is outside our window.
Thus not.
Maybe it's not Brazil.
Let's think of tornado in Europe: e.g., in Italy, tornado in 1884 in Milan? There was a tornado in Milan in 1884? Not sure.
Earthquake in Italy early 1900s: 1908 Messina earthquake. Not.
Capital city beside a river: e.g., Rome beside Tiber River. Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado? Unlikely.
President of Italy? Italy is republic with president (head of state). President in 1980s: Sandro Pertini (born 1919). Not within 1920-1935? Actually 1919 outside.
President of Italy in 1985-1992: Francesco Cossiga (born 1924). Born 1924 within range. Acquisition during his presidency (1985-1992). Historical place in Rome damaged by tornado and earthquake? Tornado in Rome? Unlikely.
Earthquake in Rome 1890s? The 1897 earthquake in Italy? Not Rome.
Maybe it's a historical place in another Italian city that is capital of region? No, capital city of country.
Let's search for historical building in Rome damaged by tornado 1880s.

Tool Call:

"name": "web_search", "arguments": "queries": ["Rome tornado 1880s historical building damage", "Rome earthquake 1890s historical building damage tornado", "Rome historical site tornado 1885"]

Tool Response:

Tool responses are omitted

Step 11

Analysis:

Search not helpful.
Maybe the tornado and earthquake are in the same country but different years.
We need to find a specific historical place, perhaps a castle or palace.
Let's think of countries where tornadoes are common and earthquakes also: e.g., USA (tornadoes common, earthquakes in some regions). So USA seems plausible.
Capital Washington D.C. Beside Potomac River.
Historical place built 1830-1860 start, completed 1870-1880. Wall thickness 0.5-0.9m (2-3 ft). Damaged by tornado 1880-1890 in D.C. Which tornado? There was a tornado in Washington D.C. on May 31, 1884 (the Great D.C. tornado). It damaged many buildings.
Historical place damaged by that tornado and also by an earthquake 1890-1900 (maybe the 1897 earthquake in D.C.? Actually there was an earthquake in 1897 in D.C.? Not sure.
Earthquake in 1897 in D.C.? There was an earthquake in 1897 in D.C. area? I recall the 1897 New Madrid earthquake was in 1897? Actually New Madrid earthquakes were 1811-1812. Not.
But there was an earthquake in 1898 near D.C.? Not typical.
However, there was an earthquake in 1897 in D.C. area? Let's search.

Tool Call:

"name": "web_search", "arguments": "queries": ["Washington DC earthquake 1890s tornado 1884 historical building damaged", "Washington DC historical site tornado 1884 earthquake 1890s", "Great tornado of 1884 Washington DC damage to historical building earthquake later"]

Tool Response:
Tool responses are omitted

Step 12

Analysis:
Seems not straightforward.
Maybe the historical place is not a famous monument but a lesser-known house or building.
Let's consider the acquisition by government between 1980-1990. In US, many historic sites were acquired by National Park Service during that period. Example: The Old Post Office Pavilion? No.
Historical place damaged by tornado 1884 and earthquake 1897? Which building in D.C. experienced both? The United States Capitol? Not.
The White House? No tornado damage.
Perhaps it's a building in another US capital: e.g., Sacramento (California) beside American River. Historical place built 1830-1860 start, completed 1870-1880, damaged by tornado 1880-1890 in Sacramento? Unlikely.
Or state capital not national capital? The question says capital city of a country.
Maybe it's not US but another country with tornadoes: e.g., Bangladesh (tornadoes called "sudden wind") but capital Dhaka beside Buriganga River. Historical place built 1830-1860 start, completed 1870-1880, damaged by tornado 1880-1890 in Dhaka? Not sure about tornado.
Earthquake 1890-1900 in Dhaka? The 1897 Great Assam earthquake (1897) felt in Dhaka? Possibly.
Acquisition by government 1980-1990: perhaps the government of Bangladesh acquired a historical house.
President of Bangladesh at acquisition time (1980s) born 1920-1935: President Ziaur Rahman (born 1937) no. President Hussain Muhammad Ershad (born 1930). He was president 1982-1990 (military). Born 1930 within range. So acquisition during his presidency.
Historical place in Dhaka damaged by tornado and earthquake? Not sure.
Let's search for historical building in Dhaka damaged by tornado 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["Dhaka tornado 1885 historical building damage", "Dhaka earthquake 1890 historical building damage", "Dhak historical site tornado 1880s earthquake 1890s"]

Tool Response:
Tool responses are omitted

Step 13

Analysis:
Seems not.
Maybe the country is not US or Bangladesh.
Let's think of tornadoes in Europe: e.g., in Italy, tornado in 1884 in Milan? There was a tornado in 1890 in Florence? Not sure.
Earthquake in early 1900s in same city.
Acquisition by government 1980-1990, president at acquisition born 1920-1935.
Which European country's president during 1980s had birth year within that range? For example, France: President François Mitterrand (born 1916) no. President Jacques Chirac (born 1932) president 1995-2007, not 1980s.
Germany: President (federal) Karl Carstens (born 1914) no. President Richard von Weizsäcker (born 1920). He was president 1984-1994. So acquisition could be during his presidency (1984-1994). Historical place in capital Berlin (beside Spree River). Construction started 1830-1860,

completed 1870-1880, walls thickness 0.5-0.9m, damaged by tornado 1880-1890 in Berlin? Unlikely tornado.
Earthquake 1890-1900 in Berlin? Not typical.
Maybe it's a country with president and capital city with tornado and earthquake: e.g., Mexico: President Carlos Salinas de Gortari (born 1942) no. President Miguel de la Madrid (born 1934) president 1982-1988. Born 1934 within range. Capital Mexico City beside Rio Grande? Actually Mexico City is near the River Mexico (Río Panuco? Not). Mexico City is on Valley of Mexico, no major river.
Historical place in Mexico City damaged by tornado 1880s? Unlikely.
Earthquake 1890-1900 in Mexico City: The 1896 Mexico City earthquake? Not sure.
But there was a major earthquake in 1985, not relevant.
Let's search for historical building in Mexico City damaged by tornado 1880s.

Tool Call:
"name": "web_search", "arguments": "queries": ["Mexico City tornado 1880s historical building damage", "Mexico City earthquake 1890s historical building damage tornado", "Mexico City historical site tornado 1885 earthquake 1890"]

Tool Response:
Tool responses are omitted

Step 14

Analysis:
No.
Let's reconsider US.
Historical place in Washington D.C. that was damaged by tornado 1884 and earthquake 1897? Which building? The Washington Monument? It was damaged by earthquake? The Washington Monument suffered damage in 1897? I recall the 1897 earthquake in Virginia? Actually the Washington Monument was damaged in 1942? Not.
Let's search for D.C. historical site damaged by tornado 1884.

Tool Call:
"name": "web_search", "arguments": "queries": ["'tornado' 1884' Washington' historical building damaged", "'tornado' 1884' Washington DC' building damage", "'Great tornado' 1884 Washington DC historical site"]

Tool Response:
Tool responses are omitted

Step 15

Analysis:
Search not picking up.
Maybe the tornado damage is not in D.C. but in another capital city where tornadoes are common: e.g., Tallahassee (Florida capital) beside a river? Tallahassee beside the Apalachicola River? Actually Tallahassee is near the Little Tallahassee River? Not major.
Historical place built 1830-1860 start, completed 1870-1880. Damaged by tornado 1880-1890 in Tallahassee? President of US at acquisition (1980-1990) born 1920-1935: President Ronald Reagan (born 1911) no, George H.W. Bush (born 1924). Acquisition of a historical place in Tallahassee by US government? Not.
But the question says "the president of the country at the time of its acquisition was born between 1920 and 1935 (inclusive)." So the country's president, not necessarily US.
If it's a US state capital, the country is the US, president is US president.

Thus it's a US capital city (Washington D.C.) historical place.
Thus we need a historical place in Washington D.C. that was damaged by tornado 1884 and earthquake 1897 (maybe the 1897 earthquake in D.C. area? Actually there was an earthquake in 1897 in D.C.? I recall the 1897 "New Madrid" earthquake was 1811-1812. There was a minor earthquake in 1897 in D.C.? Not sure.
Let's search for earthquake damage to Washington D.C. historical sites in late 19th century.

Tool Call:
"name": "web_search", "arguments": "queries": ["Washington DC earthquake 1890s historical building damage", "Washington DC earthquake 1897 building damage", "Washington DC earthquake 1895 historical site"]

Tool Response:
Tool responses are omitted

Step 16 to Step 49

(The medium steps have been omitted.)

Step 50

Analysis:
Scrolling for info about tornado damage.

Tool Call:
"name": "find", "arguments": "url": "https://en.wikipedia.org/wiki/Ruplal_House", "query": "tornado"

Tool Response:
Tool responses are omitted

Step 51

Analysis:
The Wikipedia page doesn't mention tornado. But earlier search result indicated that Ahsan Manzil had tornado damage. Let's open Ahsan Manzil page.

Tool Call:
"name": "fetch", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "page": 1

Tool Response:
Tool responses are omitted

Step 52

Analysis:
Great! Ahsan Manzil matches many criteria:
- Capital city of a country: Dhaka, capital of Bangladesh.
- Beside a river: Buriganga River.
- Construction started 1859 (within 1830-1860), completed 1872 (within 1870-1880).
- Wall thickness: Need to verify 0.5-0.9 meters. Need evidence. Let's find wall thickness info.

Tool Call:
"name": "find", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "query": "wall thickness"

Tool Response:
Tool responses are omitted

Step 53

Analysis:
Search for wall thickness of Ahsan Manzil.

Tool Call:
"name": "web_search", "arguments": "queries": ["Ahsan Manzil wall thickness", "Ahsan Manzil walls thickness meters", "Ahsan Manzil construction walls thickness"]

Tool Response:
Tool responses are omitted

Step 54

Analysis:
Thus wall thickness 0.78 m fits criteria 5.
Now we have evidence for tornado damage (April 7 1888 tornado) and earthquake repair after 1897 Assam earthquake (criterion 7 and 8). Also acquisition by government in 1985 (criterion 6). President of Bangladesh at acquisition time (1985) was Hussain Muhammad Ershad (born 1930, within 1920-1935). So all criteria satisfied.
Thus answer: Ahsan Manzil.
Provide answer with citations.
We need to cite sources for each claim.
We have Wikipedia page for Ahsan Manzil with tornado and earthquake info. Let's capture relevant lines.
Find tornado mention.

Tool Call:
"name": "find", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "query": "tornado"

Tool Response:
Tool responses are omitted

Step 55

Analysis:
Find earthquake mention.

Tool Call:
"name": "find", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "query": "earthquake"

Tool Response:
Tool responses are omitted

Step 56

Analysis:
Scrolling near that area for earthquake mention.

Tool Call:
"name": "fetch", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "page": 1

Tool Response:
Tool responses are omitted

Step 57

Analysis:
We have evidence for tornado and earthquake.
Need evidence for wall thickness 0.5-0.9 m. Search within page for "thickness".

Tool Call:
"name": "find", "arguments": "url": "https://en.wikipedia.org/wiki/Ahsan_Manzil", "query": "thickness"

Tool Response:
Tool responses are omitted

Step 58

Analysis:
Maybe wall thickness info not in Wikipedia. Need another source.
Search for Ahsan Manzil wall thickness.

Tool Call:
"name": "web_search", "arguments": "queries": ["'Ahsan Manzil' wall thickness", "'Ahsan Manzil' walls thickness", "'Ahsan Manzil' brick thickness"]

Tool Response:
Tool responses are omitted

Step 59

Analysis:
Thus we have multiple sources for wall thickness.
We can use Banglapedia entry.
Let's open Banglapedia page.

Tool Call:
"name": "fetch", "arguments": "url": "https://en.banglapedia.org/index.php/Ahsan_Manzil", "page": 1

Tool Response:
Tool responses are omitted

Step 60

Analysis:
We have evidence for:
- Construction start 1859, completed 1872.
- Tornado damage on April 7, 1888.
- Earthquake damage on 12 June 1897.
- Wall thickness 0.78 m.
- Acquired by government in 1985.
- President at acquisition time: Hussain Muhammad Ershad (born 1930).

Now we need to cite president info.
Search for president of Bangladesh in 1985.

Tool Call:
"name": "web_search", "arguments": "queries": ["Bangladesh president 1985 Ershad born 1930", "Hussain Muhammad Ershad president 1985 birth 1930", "Bangladesh president 1985 born 1930"]

Tool Response:
Tool responses are omitted

Step 61

Analysis:
Thus we have sources.
Now we need to answer: name of historical place: Ahsan Manzil.
Provide answer with citations.
We'll cite Banglapedia for construction dates, tornado, earthquake, wall thickness, acquisition.
Also cite Wikipedia for president birth and presidency.
Let's open Wikipedia page for Hussain Muhammad Ershad.

Tool Call:
"name": "fetch", "arguments": "url": "https://en.wikipedia.org/wiki/
Hussain_Muhammad
_Ershad", "page": 1

Tool Response:
Tool responses are omitted

Final Response

We have evidence for president birth 1930 and presidency 1983-1990.
Now we have all needed evidence.
Now craft final answer.
<|answer_start|>
The historical place is **Ahsan Manzil**, the former palace of the Nawabs of Dhaka in Bangladesh.
Why Ahsan Manzil fits all the criteria
Criterion	Evidence
———-	———-
Capital city of a country	Ahsan Manzil is located in Dhaka, the capital of Bangladesh.
Situated beside a river	It stands on the bank of the Buriganga River.
Construction started between 1830-1860	Construction of the palace began in **1859**.
Construction completed between 1870-1880	The palace was completed in **1872**.
Wall thickness 0.5-0.9 m	The walls of the palace are **about 0.78 m thick**.
Acquired by the government between 1980-1990	The government of Bangladesh **acquired Ahsan Manzil and its surroundings in 1985**.
Damaged by a tornado between 1880-1890	The palace was **partly damaged by a tornado on 7 April 1888**.
Damaged by an earthquake between 1890-1900	It was **badly damaged by the earthquake of 12 June 1897**.
President of the country at the time of acquisition was born between 1920-1935	The president of Bangladesh in 1985 was **Hussain Muhammad Ershad**, who was **born in 1930** (within the required range).
All of these details are documented in the Banglapedia entry for Ahsan Manzil and the Wikipedia page for Hussain Muhammad Ershad. Hence, the answer is **Ahsan Manzil**.
<|answer_end|>