

000 BEYOND A MILLION TOKENS: BENCHMARKING AND 001 ENHANCING LONG-TERM MEMORY IN LLMs 002

003 **Anonymous authors**
004

005 Paper under double-blind review
006

007 ABSTRACT

008 Evaluating the abilities of large language models (LLMs) for tasks that require
009 long-term memory and thus long-context reasoning, for example in conversational
010 settings, is hampered by the existing benchmarks, which often lack narrative co-
011 herence, cover narrow domains, and only test simple recall-oriented tasks. This
012 paper introduces a comprehensive solution to these challenges. First, we present a
013 novel framework for automatically generating long (up to 10M tokens), coherent,
014 and topically diverse conversations, accompanied by probing questions targeting
015 a wide range of memory abilities. From this, we construct BEAM, a new bench-
016 mark comprising 100 conversations and 2,000 validated questions. Second, to
017 enhance model performance, we propose LIGHT—a framework inspired by hu-
018 man cognition that equips LLMs with three complementary memory systems: a
019 long-term episodic memory, a short-term working memory, and a scratchpad for
020 accumulating salient facts. Our experiments on BEAM reveal that even LLMs
021 with 1M token context windows (with and without retrieval-augmentation) strug-
022 gle as dialogues lengthen. In contrast, LIGHT consistently improves performance
023 across various models, achieving an average improvement of 3.5%–12.69% over
024 the strongest baselines, depending on the backbone LLM. An ablation study fur-
025 ther confirms the contribution of each memory component.

026 1 INTRODUCTION

027 Large language models (LLMs) have been deployed across diverse applications, including open-
028 domain conversational agents (Laban et al., 2025; Chen et al., 2025), retrieval-augmented generation
029 (RAG) for open-domain question answering and fact checking (Lewis et al., 2020; Salemi et al.,
030 2025; Salemi & Zamani, 2025; Kim et al., 2024b), long-document and code analysis (Li et al., 2025;
031 Jelodar et al., 2025; Fang et al., 2024), and scientific or legal research (Rueda et al., 2025; Nguyen
032 et al., 2025). Many of these tasks demand models capable of processing long inputs, motivating
033 LLMs such as Gemini (DeepMind, 2025) with input windows of up to 1M tokens. Among these
034 domains, conversational systems present an intuitive and critical need for extended context, as users
035 often engage in protracted, multi-session dialogues that require consistent memory across lengthy
036 interactions (Zhong et al., 2024; Xu et al., 2022; Du et al., 2024; Tan et al., 2025). This highlights the
037 importance of evaluating how well LLMs can reason over and utilize long conversational histories.

038 While there are many prior efforts on studying and evaluating long-term memory of LLMs (Kim
039 et al., 2024a; Xu et al., 2021; Maharana et al., 2024; Zhong et al., 2024; Xu et al., 2022; Du et al.,
040 2024; Tan et al., 2025), existing benchmarks have fundamental limitations. Most extend conversa-
041 tion length by artificially concatenating short sessions of different users, producing dialogues with
042 abrupt topic shifts and weak narrative coherence. Such a construction artificially simplifies eval-
043 uation because distinct segments are easily separable, reducing the need for true long-range rea-
044 soning. Furthermore, these datasets typically target narrow domains—often limited to personal-life
045 scenarios—leaving many real-world application areas underrepresented. Finally, they emphasize
046 simple context recall, overlooking other critical memory abilities such as contradiction resolution,
047 recognizing evolving information, and instruction following.

048 To address these limitations, this paper presents a framework for automatically generating long
049 coherent conversations between a user and an AI assistant—scaling up to 10M tokens on diverse
050 domains—with a set of probing questions designed to evaluate diverse memory abilities of any
051 LLM on the generated dialogues. An overview of the data generation framework is shown in Figure
052 1. This framework begins by defining a high-level conversation plan—a narrative for a particular do-
053 main and a simulated user with generated attributes—that outlines the overall flow of the dialogue.

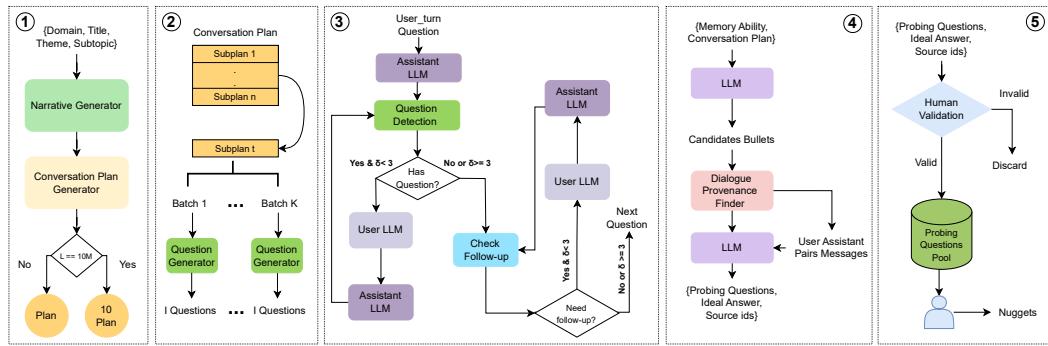


Figure 1: Overview of BEAM generation process. In the first stage, conversation plans are created for each chat seed. In the second stage, user utterances are generated from the conversation plans. In the third stage, assistant responses are produced, followed by two controller modules: one responsible for checking whether the assistant has posed a question to the user, and the other for determining whether a follow-up question from the user is needed. In the fourth stage, probing questions are generated based on the targeted memory abilities and corresponding conversation plans, and their answers are identified within the chat. In the final stage, invalid probing questions are filtered out, and nuggets are created for the validated set.

This plan is recursively decomposed into finer sub-plans that specify the storyline and its progression. From these sub-plans we generate chronologically ordered user turns, which are then expanded with corresponding assistant responses. To increase realism, the system injects follow-up questions and clarifications from both sides. Finally, we automatically create a set of probing questions that target ten distinct memory dimensions, with a focus on complicated and multi-hop reasoning, which are then validated by human annotators to ensure high quality. Using this pipeline, we construct the BEAM dataset: 100 diverse conversations ranging from 100 K to 10 M tokens each, accompanied by 2000 probing questions to evaluate the memory capabilities of LLMs.

To improve LLM performance on probing questions, we introduce the LIGHT framework (Figure 2), which is applicable to both open-source and proprietary LLMs, inspired by research in human cognitive science and human’s memorization and recall process (Sridhar et al., 2023; Binder & Desai, 2011). This framework integrates three complementary memories: (1) episodic memory, a long-term index of the full conversation used for retrieval; (2) working memory, capturing the most recent user–assistant turns; and (3) a scratchpad, where after each turn the model reasons over the dialogue and records salient facts for future use. At inference, the LLM draws jointly on retrieved episodic content, the working memory, and the accumulated scratchpad to generate accurate answers.

To evaluate LLM memory capabilities and the effectiveness of our method, we conduct experiments on the constructed dataset, BEAM, using both open-source and proprietary models. Results show that even LLMs with long context windows perform substantially worse as conversation length increases. Our method improves the LLM’s performance in answering the probing questions by 3.5%–12.69% on average over the best-performing baseline, depending on the backbone model and conversation length. An ablation study further reveals the contribution of each LIGHT component on the performance. To support future work, we release all code, data, and evaluation scripts.¹

2 BEAM: BENCHMARKING MEMORY CAPABILITIES OF LLMs

2.1 PROBLEM FORMULATION

Let $\mathcal{D} = \{T_i\}_{i=1}^{|\mathcal{D}|}$ denote a collection of $|\mathcal{D}|$ conversations between users and a conversational agent π . Each conversation is represented as $\mathcal{T} = \{t_i\}_{i=1}^{|\mathcal{T}|}$, where $t_i \in \mathcal{T}$ corresponds to the i^{th} utterance (turn) in the dialogue. The objective of this work is to systematically evaluate a predefined set of memory abilities \mathcal{M} exhibited by π across conversations. For each memory ability $m \in \mathcal{M}$, we construct a probing dataset of size N , denoted as $\mathcal{Q}_m = \{(x_i, y_i)\}_{i=1}^N$, where x_i is a probing

¹Codes and data will be released upon acceptance.

108 question and y_i is the corresponding ground-truth answer set. Each probing question $(x, y) \in \mathcal{Q}_m$
 109 is appended as the $(|\mathcal{T}| + 1)^{\text{th}}$ turn in the dialogue, and the system generates a response $\hat{y} = \pi(x; \mathcal{T})$
 110 based on the conversation. The generated response is then evaluated using an ability-specific scoring
 111 function μ_m , producing a performance score $s = \mu_m(x, y, \hat{y})$. The goal of this work is to quantify
 112 the performance of conversational systems on each memory ability in \mathcal{M} .
 113

114 2.2 BENCHMARK CREATION

115 Our goal is to evaluate how well LLMs can answer questions that depend on long-term conversational
 116 memory. We measure performance across ten complementary abilities, seven drawn from
 117 prior benchmarks and three newly introduced here—*Instruction Following*, *Event Ordering*, and
 118 *Contradiction Resolution* (see Table 2 in Appendix B.1). *Abstention* evaluates whether a model
 119 withholds answers when evidence is missing. *Contradiction Resolution* tests the capacity to detect
 120 and reconcile inconsistent statements across widely separated turns, maintaining global coherence.
 121 *Event Ordering* assesses whether a model can recognize and reconstruct the sequence of evolving
 122 information in the dialogue. *Information Extraction* measures recall of entities and factual details in
 123 long histories. *Instruction Following* examines sustained adherence to user-specified constraints over
 124 long contexts. *Information Update* evaluates revising stored facts as new ones appear. *Multi-hop*
 125 *Reasoning* probes inference that integrates evidence across multiple, non-adjacent dialogue seg-
 126 ments. *Preference Following* captures personalized responses that adapt to evolving preferences.
 127 *Summarization* assesses the ability to abstract and compress dialogue content, while *Temporal Rea-
 128 soning* tests reasoning about explicit and implicit time relations. Together, these abilities evaluate a
 129 system’s capacity to maintain, update, and manipulate information throughout extended conversa-
 130 tions (see Appendix B.6 for examples of each ability). Given these abilities and the formulation in
 131 Section 2.1, the benchmark requires three components: 1) a user–assistant conversation, 2) probing
 132 questions targeting key memory abilities, and 3) an evaluation methodology to assess the model’s
 133 responses. The overall statistics of the constructed benchmark are summarized in Table 3 in Ap-
 134 pendix B.1. The rest of this section details the process used to construct these components.
 135

136 **Overview:** The overview of our framework for creating conversations, probing questions, and the
 137 evaluation strategy is illustrated in Figure 1. The process begins by generating a simulated conver-
 138 sation between a user and an assistant. Structured conversation plans are first produced to guide
 139 the flow of the synthetic interactions. Each plan specifies sufficient information to generate both
 140 user and assistant turns, ensuring a coherent and natural conversational trajectory. While a typical
 141 exchange consists of a user question followed by an assistant response, realistic dialogues often
 142 involve follow-ups for clarification, elaboration, or related subtopics. To capture this, we incorpo-
 143 rate two interaction-control modules. The question-detection module identifies whether an assistant
 144 response includes a query that requires a user reply; if triggered, the system generates the corre-
 145 sponding user response. The follow-up detection module determines when the user would naturally
 146 pose a clarifying or elaborative question; if triggered, it produces an additional user query for the
 147 assistant. Together, these mechanisms produce conversations that exhibit interactive, bidirectional
 148 behavior beyond simple turn-taking. After the conversation is generated, an automated procedure
 149 constructs a candidate set of probing questions, each tailored to the specific memory abilities in the
 150 benchmark. These candidates are then reviewed by a human evaluator, who selects valid questions
 151 and formulates the associated evaluation rubrics used for subsequent benchmarking. A case study
 152 and an example of the different generated components of a conversation is provided in Appendix E.
 153

154 2.2.1 CONVERSATION PLAN GENERATION

155 A *conversation plan* serves as the scaffold for each dialogue, providing a coherent storyline that
 156 unfolds chronologically. Each plan is generated using an LLM conditioned on a structured seed that
 157 specifies key elements of the conversation. The seed contains:

- 158 • **Domain** of the conversation,
- 159 • **Title and theme**,
- 160 • **Subtopics** outlining specific topics,
- 161 • **Narratives** defining evolving aspects (e.g., career progression, goals),
- **User profile** (name, age, gender, location, profession, MBTI traits),

- **Relationship graph** linking user to family, friends, and acquaintances,
- **Timeline** specifying temporal scope and key chronological anchors.

To generate candidate titles and themes, human annotators specify target domains, then GPT-4.1 (OpenAI, 2025a) generates candidate titles, themes, and subtopics using Listing 22. Human reviewers refine outputs for topical diversity. For each conversation, we generate 15-20 narratives using the open-source LLaMA-3.3 70B model (AI, 2024) with the prompt in Listing 23 (Appendix H). Given the conversation seed, this model produces narrative elements capturing the evolving storyline, forming the backbone of a coherent conversation.

Conversation plans consist of N *sub-plans*, each representing a distinct stage in the conversation. Each sub-plan contains M *bullet points*, defined by a *narrative*, a descriptive statement of its role in the storyline, and a *time anchor*. For conversations of 128K, 500K, and 1M tokens, a single plan is generated (line 4 in Algorithm 1, Appendix B.3.5) by conditioning the LLM on the conversation seed, profile, relationship graph, timeline, and specified counts of sub-plans, bullet points, and narratives (prompt in Listing 24, Appendix H). The number of sub-plans varies with domain and target length to meet the token requirement; e.g., coding domains generally require fewer turns than broader domains. For 10M-token conversations, one plan cannot capture the scope, so we create ten interlocking plans forming a coherent longer narrative. The process begins with a global seed defining the overall topic and theme, but a single seed is insufficient; instead, we derive ten distinct seeds—one per plan—so the narrative can evolve across stages. We propose two strategies:

- **Sequential Expansion:** The global seed defines the initial point in the conversation’s chronology. Subsequent seeds represent successive events (e.g., a trip, job search, later milestones). Using the prompt in Listing 28 (Appendix H), each new seed is generated from the main seed, profile, and timeline. Plans are then produced sequentially (line 12 in Algorithm 1, Appendix B.3.5), with each plan conditioned on its predecessor to maintain continuity. Core relationships (e.g., parents) remain fixed, while new acquaintances are gradually introduced to reflect the evolving context.
- **Hierarchical Decomposition:** The main seed is decomposed into ten sub-seeds, each representing a distinct topical and temporal segment. Together, these sub-seeds span the full storyline (e.g., an international trip: first three for preparation, next five for trip events, final two for reflections). Similar to sequential expansion, the user’s core relationships remain constant, while new acquaintances are introduced to reflect the evolving context. These ten sub-seeds are generated using the prompt in Listing 29 (Appendix H), conditioned on the main seed, profile, and timeline.

Each conversation plan is assigned explicit topical and temporal boundaries—encoded in the seed—to avoid redundancy and ensure sub-themes appear in the right narrative stage. For coherence, the LLM conditions on summaries of prior plans and future seeds when producing a new plan, allowing anticipation of upcoming events (e.g., reserving tickets for travel dates). This procedure is implemented in line 20 of Algorithm 1 (Appendix B.3.5). Plans are generated using the prompt in Listing 31 (Appendix H), conditioned on the main seed, current sub-seed, number of sub-plans, narrative set, user profile, core and new relationships, preceding and subsequent sub-seeds, previous plan, a summary of earlier plans, current sub-seed index, and a binary flag for the first plan (triggering user introduction). Since initial plans may not sufficiently test three key memory abilities—*contradiction resolution*, *information update*, and *instruction following*—we apply a two-stage augmentation: first generate the base plan, then use GPT-4.1 (Listing 27) to augment each sub-plan with three targeted bullet points. Performing augmentation separately improves coverage and fidelity. The refinement follows the prompt in Listing 27 (Appendix H), which takes plan as input and outputs the revised version. This stage corresponds to the first module in Figure 1, which forms the first step of the overall data-generation pipeline. The detailed process for plan generation is reported in Appendix B.3.2.

2.2.2 USER UTTERANCE GENERATION

Once conversation plans are constructed, user utterances are synthesized from the sub-plans. Each sub-plan contains M bullet points, which are divided sequentially into K contiguous batches of equal size. Batching narrows the LLM’s focus, reducing repetition and low-quality outputs that can occur when conditioning on the entire sub-plan. For each batch, the LLM generates I user questions (line 6 in Algorithm 2 in Appendix B.3.5) using the prompt in Listing 32 (Appendix H),

216 conditioned on the conversation seed, the current batch, preceding batches, and context from earlier
 217 sub-plans. Each generated user question constitutes a user turn in the dialogue, ensuring coherence
 218 and continuity across extended conversations. Values of K and I are manually specified based on
 219 domain and target conversation length to meet the token budget, with configurations reported in
 220 Table 6 (Appendix B). This provides fine-grained control over user interaction density, preventing
 221 under-generation or redundancy. To balance quality and cost, question generation uses the open-
 222 source LLaMA-3.3 70B model (AI, 2024), which produces high-quality outputs efficiently as the
 223 backbone LLM. This user-utterance construction aligns with the second stage in Figure 1. The
 224 details of this procedure for user utterance generation are provided in Appendix B.3.3.

225 2.2.3 ASSISTANT UTTERANCE GENERATION

226 Assistant-side responses are generated iteratively in a role-playing setup, where one LLM assumes
 227 the *assistant role* and another the *user role*. For each sub-plan, the assistant LLM is conditioned
 228 on the conversation seed (Section 2.2.1), prior sub-plans, a summary of the last M turns, and a
 229 compressed summary of earlier ones (using the prompt in Listing 37 in Appendix B); for 10M-
 230 token conversations, additional summaries of prior plans are provided. The assistant first generates
 231 a response to the user’s most recent question (line 9 in Algorithm 3 in Appendix B.3.5), which is an-
 232 alyzed by a *question-detection module* (line 11 in Algorithm 3 in Appendix B.3.5, using the prompt
 233 in Listing 35 Appendix B) to determine the presence of a counter-question. If detected, the response
 234 is passed to the user LLM, which generates a contextually consistent reply based on the current and
 235 prior sub-plans, relevant history, and conversation summaries (using the prompt in Listing 38 in Ap-
 236 pendix B, line 14 in Algorithm 3 in Appendix B.3.5). This loop continues until no further assistant
 237 questions are detected or the threshold $\delta_1 = 2$ is reached, balancing realism and avoiding infinite
 238 cycles. In addition, a *follow-up detection module* (line 21 in Algorithm 3 in Appendix B.3.5, using
 239 the prompt in Listing 36 in Appendix B) evaluates whether a clarifying or elaborative user follow-
 240 up is warranted, based on factors such as subject complexity, ambiguity, or incomplete responses.
 241 When required, the module generates a follow-up query conditioned on the seed, current and prior
 242 sub-plans, the most recent M turns, and earlier summaries (using the prompt in Listing 39 in Ap-
 243 pendix B), which is then passed back to the assistant LLM. The number of follow-up exchanges is
 244 limited by a threshold $\delta_2 = 2$, analogous to δ_1 . Together, these modules yield dialogues with bidi-
 245 rectional dynamics, contextual referencing, and realistic clarifications, approximating human–AI
 246 interactions. This assistant-side generation maps to the third module in Figure 1. The details of this
 247 procedure are provided in Appendix B.3.4.

248 2.3 PROBING QUESTIONS GENERATION

249 After constructing conversations, we generate probing questions to evaluate memory abilities. The
 250 pipeline combines automated synthesis with human validation: an LLM first produces candidate
 251 probes, which annotators review to select valid ones. Probes are derived from both the conversa-
 252 tion plan and chat to ensure each targets a specific ability, is grounded in dialogue turns, and includes
 253 explicit provenance. The process begins by passing the plan to GPT-4.1-mini (OpenAI, 2025b),
 254 which selects candidate bullet points conditioned on the ability under evaluation. For example,
 255 knowledge-update probes require bullet pairs encoding an initial fact and its later revision, while
 256 summarization and event-ordering probes span multiple bullets. Each bullet is linked to its corre-
 257 sponding user and assistant turns through indices introduced during user-assistant turn generation,
 258 enabling retrieval of the precise dialogue segments in which the content was created. Candidate
 259 bullet selection is performed using prompts 1–9, one per memory ability. For abstention, candi-
 260 date selection is unnecessary; probes are created directly from the plan using the prompt shown in
 261 Listing 14 (Appendix H).

262 Given the selected bullet points and aligned dialogue snippets, GPT-4.1-mini generates the prob-
 263 ing question, a candidate answer, and source identifiers citing the specific messages containing the
 264 answer. For 10M-token dialogues, candidate selection and synthesis are performed with a sliding
 265 window across the ten interlocking plans, processing a limited number at a time to preserve topical
 266 locality and scalability. Probe generation uses prompts 10–19 for each memory ability, mapping
 267 candidate bullet points and contexts into fully formed questions. Finally, a human evaluator re-
 268 views the generated candidates and selects those that are valid and consistent with the conversation.
 269 Samples of probing questions are provided in Appendix D, items 1–10.

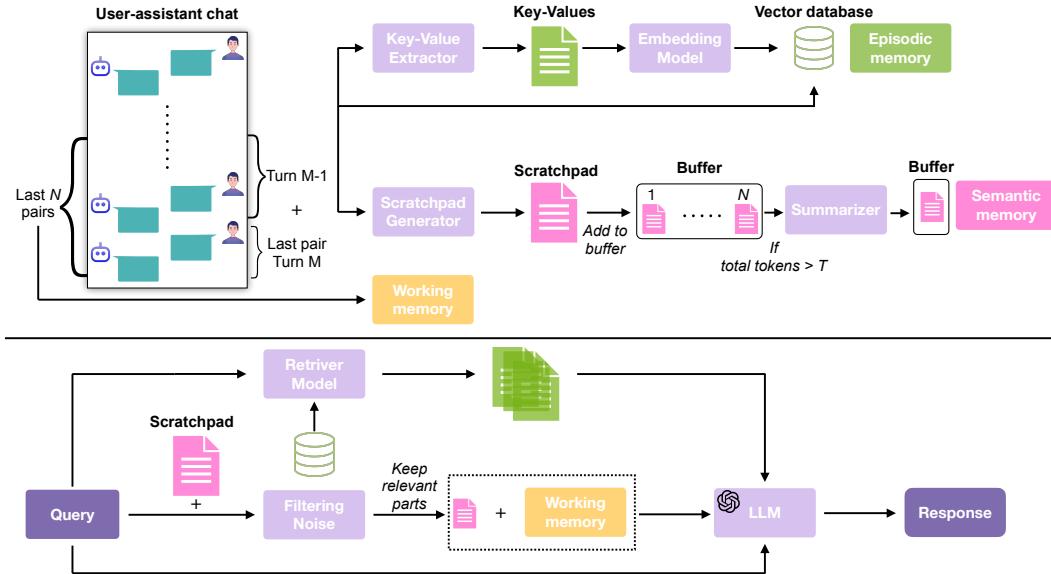


Figure 2: Overview of the LIGHT framework. The system combines (i) **episodic retrieval** from key-value indices, (ii) a **scratchpad** consolidating semantic, autobiographical, and contextual knowledge, and (iii) a **working memory buffer** of the most recent dialogue pairs. At inference, relevant items from the index and scratchpad, together with the full working memory, are integrated to generate the final response.

2.4 EVALUATION

We evaluate LLMs on the probing questions using nugget evaluation, a common approach for long-form text assessment (Pradeep et al., 2024; 2025). Each probing question is manually validated: invalid or unsupported questions are discarded, and minor inconsistencies are corrected. From the validated set, two questions per memory ability are chosen for each conversation, yielding 20 probing questions per conversation. Rubric nuggets are then derived for each question. A nugget is an atomic, self-contained criterion that a system response must satisfy. Annotators decompose the ideal reference answer into minimal semantic units, ensuring each nugget is both atomic and self-contained. System responses are scored against these nuggets by an LLM judge (Listing 20, Appendix H), which assigns 0 (unsatisfied), 0.5 (partially satisfied), or 1 (fully satisfied). Scores are averaged across nuggets to produce ability-level metrics. This nugget-based procedure applies to nine memory abilities; the exception is event ordering, where quality depends on both recall and correct sequence. We evaluate event ordering using the Kendall tau-b coefficient (Kendall, 1945), which considers both order and presence. To apply this metric, an LLM equivalence detector (using the prompt in Listing 21 in Appendix H) aligns events in system responses with nuggets, outputting `yes` if two snippets denote the same event/topic and `no` otherwise. Kendall tau-b is then computed over the aligned sequences, capturing both recall and ordering fidelity. Examples of nugget construction for each memory ability are provided in Appendix D.

3 LIGHT: IMPROVING MEMORY CAPABILITIES OF LLMs

Inspired by research in human cognitive science (Sridhar et al., 2023; Binder & Desai, 2011), humans employ two primary mechanisms for remembering and using knowledge: *episodic memory*, the ability to recall specific personal experiences along with their context, and *working memory*, the capacity to retain and manipulate information about recent events over short periods. In addition, maintaining notes on a *scratchpad* provides an external record that supports long-term recall and later retrieval. Since answering questions in long-context conversations similarly requires integrating past experiences and accumulated knowledge, we introduce a method that emulates these strategies by combining episodic recall, short-term working memory, and an external scratch-pad mechanism.

324 **Overview:** An overview of our method is shown in Figure 2. Given a question x about a con-
 325 versation $\mathcal{T} = \{t_i\}_{i=1}^{|\mathcal{T}|}$, where $|\mathcal{T}|$ is the total number of turns, the framework first queries a re-
 326 trieval model R to obtain k relevant segments from \mathcal{T} , simulating recall from episodic memory:
 327 $E = R(x, k, \mathcal{T})$. Next, the most recent z dialogue pairs of the conversation are selected to form the
 328 working memory, $W = \{t_{|\mathcal{T}|-i}\}_{i=0}^z$. In parallel, a pre-constructed scratchpad $S_{|\mathcal{T}|}$ contains up to m
 329 salient notes. A filtering function f retains only the items pertinent to x , yielding $S_x = f(S_{|\mathcal{T}|}, x)$.
 330 Finally, the LLM π generates the answer by conditioning on the question and these three mem-
 331 ory components, $y = \pi(x, E, W, S_x)$ using the prompt shown in Listing 44 in Appendix H. The
 332 remainder of this section details the construction and logic of each component in this pipeline.
 333

334 3.1 RETRIEVAL FROM THE CONVERSATION

335 **Indexing the Conversation:** After each user–assistant turn (Figure 2, top), we apply Qwen2.5-
 336 32B-AWQ (Team, 2024) with the prompt in Listing 40 (Appendix H) to extract key–value pairs and
 337 a summary of the interaction. Keys represent entities and values capture attributes or descriptive
 338 details, providing fine-grained, event-level indices analogous to hippocampal memory traces (Teyler
 339 & DiScenna, 1986). These key–value pairs and summaries are embedded using the BAAI/bge-
 340 small-en-v1.5 embedding model (of Artificial Intelligence, 2023) and stored in a vector database as
 341 keys, while the original dialogue segments are kept as values to ensure faithful grounding.
 342

343 **Retrieval from the Index:** To retrieve information from the conversation as episodic memory, we
 344 embed the question x using the same embedding model and compare it against the stored keys in the
 345 index, and the original dialogue segments corresponding to the top k nearest neighbors are returned.
 346

347 3.2 SCRATCHPAD FORMATION AND UTILIZATION

348 **Construction:** In addition to episodic memory (Figure 2, middle pathway), we build a higher-level
 349 representation that preserves information beyond individual dialogue events. It integrates semantic
 350 knowledge (facts and concepts), autobiographical details (life events), prospective memory (future
 351 intentions), and contextual metadata (time, place, acquisition context) (Binder & Desai, 2011). For
 352 each dialogue pair, we use Qwen2.5-32B-AWQ with the prompt in Listing 41 (Appendix H) to rea-
 353 son over the current and preceding turn and extract salient content. The resulting “scratchpad” is
 354 iteratively merged with earlier versions; once content exceeds a 30K-token threshold—substantially
 355 shorter than the raw conversation—it is compressed into a 15K-token summary by GPT-4.1-nano us-
 356 ing the prompt in Listing 42. This process maintains efficiency and long-term coherence, analogous
 357 to the gradual abstraction of semantic memory in humans. Unlike the episodic index, the scratchpad
 358 is not stored in a retrieval database but is provided directly as contextual input during inference.
 359

360 **Filtering Scratchpad (function f):** During inference, the scratchpad is selectively filtered with
 361 respect to the question. It is first divided into semantically coherent chunks using *semantic chunking*.
 362 ² Each chunk is evaluated by Qwen2.5-32B-AWQ with the prompt in Listing 43 (Appendix H),
 363 which assigns a binary relevance label (yes/no). Only the chunks judged relevant are retained,
 364 producing a condensed representation of scratchpad that is passed to the response generator.
 365

366 4 EXPERIMENTS

367 4.1 EXPERIMENTAL SETUP

368 **Baselines:** We evaluate our approach against two types of baselines: long-context LLMs and a
 369 RAG method. For long-context LLMs, the entire conversation history is provided, followed by
 370 the probing question. We include two proprietary LLMs (*GPT-4.1-nano*, *Gemini-2.0-flash*, both
 371 1M context), and two open-source models (*Qwen2.5-32B-AWQ*, *Llama-4-Maverick-fp8*). For long-
 372 context experiments, *Qwen2.5-32B-AWQ* is evaluated with a 128K context length, while for the
 373 RAG baseline and our proposed method a 32K context length is used. At the 10M-token, since none
 374 of the four models support this length, they are evaluated on the largest recent dialogue segment
 375

376 ²SemanticChunker in LangChain is used, which segments text into variable-length passages based on
 377 semantic rather than fixed token windows.

378
379
Table 1: Comparison of different LLMs and methods across conversation lengths and memory abilities using the created benchmark. Methods with the best performance per evaluation are bolded.
380

Length	Memory Ability	Qwen 2.5			Llama Maverick			Gemini 2 Flash			GPT-4.1-nano		
		Vanilla	RAG	Ours	Vanilla	RAG	Ours	Vanilla	RAG	Ours	Vanilla	RAG	Ours
100K	Abstention	0.300	0.650	0.475	0.200	0.800	0.600	0.800	0.800	0.675	0.475	0.800	0.575
	Contradiction Resolution	0.031	0.025	0.037	0.025	0.031	0.031	0.006	0.050	0.018	0.012	0.018	0.031
	Event Ordering	0.192	0.201	0.205	0.190	0.162	0.166	0.181	0.191	0.166	0.181	0.169	0.177
	Information Extraction	0.425	0.338	0.479	0.510	0.392	0.518	0.333	0.341	0.464	0.273	0.362	0.538
	Instruction Following	0.400	0.375	0.362	0.412	0.375	0.412	0.275	0.287	0.362	0.425	0.350	0.400
	Knowledge Update	0.437	0.275	0.362	0.300	0.350	0.450	0.125	0.325	0.300	0.275	0.375	0.375
	Multi-Hop Reasoning	0.222	0.203	0.281	0.152	0.225	0.353	0.200	0.148	0.225	0.178	0.263	0.365
	Preference Following	0.554	0.379	0.566	0.450	0.512	0.625	0.300	0.416	0.462	0.437	0.550	0.625
	Summarization	0.128	0.074	0.232	0.065	0.111	0.238	0.018	0.093	0.139	0.028	0.083	0.202
	Temporal Reasoning	0.112	0.112	0.100	0.275	0.187	0.187	0.150	0.125	0.112	0.125	0.162	
500K	Average	0.280	0.269	0.311	0.240	0.323	0.358	0.242	0.280	0.294	0.239	0.309	0.345
	Abstention	0.314	0.728	0.571	0.185	0.785	0.628	0.714	0.800	0.685	0.557	0.828	0.600
	Contradiction Resolution	0.053	0.017	0.017	0.035	0.028	0.042	0.010	0.021	0.021	0.017	0.025	0.035
	Event Ordering	0.185	0.221	0.244	0.209	0.186	0.197	0.215	0.189	0.200	0.188	0.180	0.204
	Information Extraction	0.166	0.400	0.506	0.608	0.402	0.535	0.469	0.343	0.478	0.142	0.382	0.491
	Instruction Following	0.304	0.350	0.295	0.403	0.447	0.390	0.133	0.334	0.280	0.244	0.286	0.342
	Knowledge Update	0.111	0.226	0.278	0.276	0.338	0.264	0.171	0.180	0.223	0.107	0.288	0.240
	Multi-Hop Reasoning	0.125	0.187	0.214	0.219	0.313	0.350	0.198	0.135	0.157	0.070	0.233	0.266
	Preference Following	0.567	0.477	0.571	0.560	0.525	0.623	0.379	0.427	0.532	0.450	0.577	0.684
	Summarization	0.137	0.187	0.344	0.266	0.197	0.373	0.136	0.165	0.250	0.109	0.184	0.334
1M	Temporal Reasoning	0.035	0.114	0.121	0.064	0.078	0.190	0.150	0.078	0.092	0.057	0.161	0.154
	Average	0.200	0.291	0.316	0.283	0.330	0.359	0.257	0.267	0.292	0.194	0.314	0.335
	Abstention	0.342	0.650	0.500	0.221	0.742	0.435	0.642	0.750	0.735	0.492	0.778	0.678
	Contradiction Resolution	0.035	0.035	0.021	0.046	0.028	0.042	0.010	0.028	0.007	0.050	0.028	0.021
	Event Ordering	0.183	0.195	0.200	0.214	0.179	0.193	0.190	0.198	0.185	0.191	0.179	0.211
	Information Extraction	0.138	0.407	0.366	0.489	0.431	0.474	0.374	0.380	0.341	0.153	0.399	0.410
	Instruction Following	0.383	0.300	0.419	0.440	0.338	0.433	0.120	0.290	0.380	0.226	0.271	0.394
	Knowledge Update	0.064	0.378	0.357	0.164	0.342	0.414	0.107	0.278	0.264	0.150	0.342	0.392
	Multi-Hop Reasoning	0.102	0.163	0.209	0.174	0.245	0.270	0.083	0.134	0.147	0.091	0.293	0.278
	Preference Following	0.486	0.491	0.551	0.535	0.514	0.610	0.273	0.470	0.472	0.435	0.513	0.576
10M	Summarization	0.122	0.157	0.316	0.207	0.145	0.315	0.091	0.125	0.224	0.060	0.152	0.290
	Temporal Reasoning	0.073	0.078	0.154	0.097	0.107	0.176	0.104	0.057	0.085	0.061	0.064	0.107
	Average	0.193	0.285	0.309	0.259	0.307	0.336	0.199	0.271	0.284	0.191	0.302	0.336
	Abstention	0.250	0.600	0.550	0.050	0.700	0.450	0.750	0.650	0.650	0.450	0.650	0.400
	Contradiction Resolution	0.050	0.000	0.012	0.025	0.000	0.000	0.025	0.000	0.000	0.012	0.025	
	Event Ordering	0.180	0.221	0.197	0.190	0.220	0.176	0.220	0.266	0.193	0.215	0.201	0.173
	Information Extraction	0.100	0.350	0.350	0.075	0.375	0.300	0.075	0.275	0.150	0.050	0.300	0.350
	Instruction Following	0.175	0.200	0.350	0.250	0.350	0.500	0.025	0.125	0.250	0.075	0.175	0.250
	Knowledge Update	0.100	0.300	0.275	0.100	0.375	0.325	0.050	0.325	0.200	0.050	0.325	0.300
	Multi-Hop Reasoning	0.125	0.050	0.125	0.000	0.075	0.125	0.000	0.125	0.125	0.012	0.091	0.135
401 402 403 404 405 406 407	Preference Following	0.241	0.291	0.308	0.291	0.316	0.483	0.075	0.300	0.150	0.175	0.366	0.425
	Summarization	0.114	0.106	0.220	0.065	0.053	0.277	0.000	0.045	0.136	0.020	0.063	0.179
	Temporal Reasoning	0.000	0.000	0.000	0.000	0.025	0.025	0.025	0.025	0.075	0.050	0.000	0.025
	Average	0.133	0.211	0.238	0.104	0.249	0.266	0.122	0.216	0.192	0.109	0.218	0.226

408
409 fitting their window.³ For RAG baselines, each user–assistant turn pair is treated as a document,
410 embedded and stored in a vector database. At inference, the top five most similar documents are
411 retrieved and passed to the LLM using the prompt in Listing 44 (Appendix H).

412
413 **Inference Setup:** For inference, we use Nucleus (Holtzman et al., 2020) with temperature 0, ex-
414 cept for conversation plan, user-turn, and assistant-turn generation, where temperature is 0.1 to
415 encourage diversity. All open-source LLMs are served via VLLM for efficient inference. For
416 Llama3.3-70B, we set the maximum output length to 6K tokens during user-turn generation, while
417 for other LLMs we adopt their default maximum output length. For experiments involving both
418 the RAG baseline and our proposed method, we employ FAISS as the vector database (Douze et al.,
419 2024). For dense retrieval, we use the embedding model *BAAI/bge-small-en-v1.5* (Xiao et al., 2023).
420

421 4.2 EMPIRICAL FINDINGS

422
423 **Main Results:** Across all four conversation lengths (100K–10M tokens), our method consistently
424 outperforms both long-context LLMs and RAG baselines (Table 1). At shorter contexts (100K),
425 we observe strong gains, such as +49.1% for Llama-4-Maverick and +44.3% for GPT-4.1-nano
426 over long-context baselines, showing that structured memory helps even when full history can be
427 processed. The benefits grow with context length: at 1M tokens, improvements reach +75.9% for
428 GPT-4.1-nano and +60.1% for Qwen2.5-32B. At 10M tokens—where no baseline natively supports
429 the full context—our method achieves dramatic improvements, including +155.7% for Llama-4-
430 Maverick and +107.3% for GPT-4.1-nano. The only exception is Gemini-2.0-flash at 10M, where
431 our method surpasses the long-context baseline (+57.3%) but slightly trails RAG, likely due to

³ Among available models, only *Llama-4-Scout* supports 10M-token context windows; however, due to its extreme computational requirements, we were unable to include it in our experiments.

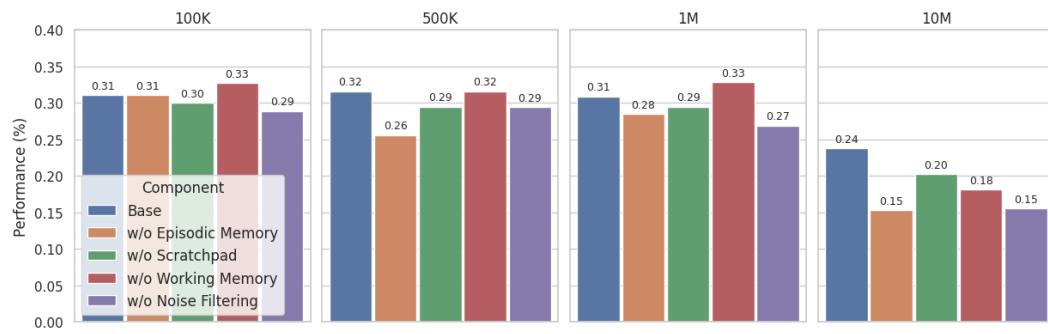


Figure 3: Ablation study illustrating the contribution of each component in LIGHT (retrieval, scratchpad, working memory, and noise filtering) across different conversation lengths.

model-specific retrieval behavior. Overall, these findings underscore the scalability and robustness of our framework across diverse architectures and extreme context lengths.

When evaluated across the ten memory abilities, our method shows the largest relative gains in summarization (+160.6%), multi-hop reasoning (+27.2%), and preference following (+76.5%). Strong improvements are also observed in information extraction (+56.7%), instruction following (+39.5%), and temporal reasoning (+56.3%). These results highlight that our method is particularly effective for tasks requiring long-range recall and integration of dispersed information. In contrast, all methods—including ours—perform strongest in abstention and weakest in contradiction resolution, indicating that contradiction detection remains a challenging open problem.

Ablation: We conduct an ablation to assess the role of each component—*episodic memory*, *scratchpad*, *working memory*, and *noise filtering*—across conversation lengths (Figure 3). At 100K, removing retrieval does not change performance and it remains steady, since the scratchpad alone suffices, while removing scratchpad or noise filtering reduces performance (−1.1%, −2.2%). Working memory also degrades results here (−1.6%). At 500K, removing any component reduces performance except working memory, where removal enhances performance very slightly. At 1M, retrieval, scratchpad, and noise filtering remain beneficial, but removing working memory slightly improves performance, again reflecting its limited usefulness when few questions depend on the most recent turns. By 10M, all components are essential, with removals leading to large drops (−8.5% for retrieval, −3.7% for scratchpad, −5.7% for working memory, −8.3% for noise filtering). Overall, the ablations show that each module contributes increasingly as context length grows, and the full architecture consistently achieves the best performance. Detailed results across all memory abilities are provided in Table 8.

Effect of Retrieval Budget: We examine the effect of retrieval budget (K), testing 5, 10, 15, and 20 documents (Figure 4). Performance consistently improves when increasing K from 5 to 15, with the best results at $K=15$ (+7.39%, +10.75%, +6.79%, and +6.3% at 100K, 500K, 1M, and 10M). Increasing further to $K=20$ slightly degrades performance, likely due to noisy context. Results at $K=10$ are mixed—helpful at 100K, 500K and 1M but harmful at 10M—indicating additional documents sometimes add noisy information. Full results across memory abilities are shown in Table 9. We also conducted complementary experiments analyzing the effect of retriever choice, where we observed that at 100K, 500K, and 1M token

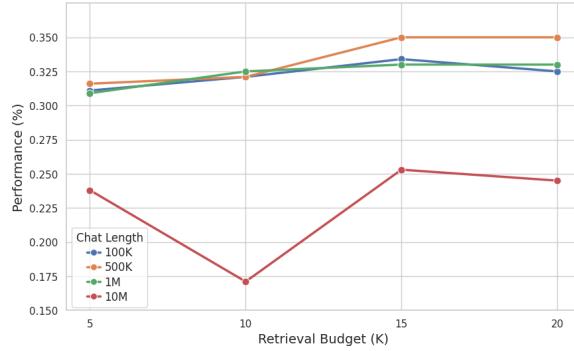


Figure 4: Effect of varying retrieval budget (K) on performance. The plot shows how the number of retrieved documents shapes the balance between recall and noise, highlighting different behaviors at short and long conversation lengths.

486 lengths, using a sparse retriever improves performance, whereas at 10M tokens, the dense retriever
 487 achieves better results. The full results and discussion are provided in Appendix C.2.
 488

489 **Case Study** A case study demonstrating the usefulness of the scratchpad is provided in Ap-
 490 pendix F.
 491

492 **Human Evaluation:** We conducted a human evaluation to assess the quality of the generated
 493 conversations. Three dimensions were considered: *Coherence and Flow*, *Realism*, and *Complexity*
 494 and *Depth*, each rated on a 5-point Likert scale (1 = lowest, 5 = highest). The average scores across
 495 all conversations were 4.53, 4.57, and 4.64, respectively, indicating consistently high quality. The
 496 evaluation rubric and detailed scores are provided in Appendix B.2.
 497

498 5 RELATED WORK

500 The detailed related work is provided in Appendix A; here we present a concise summary.

501 Context windows of LLMs have expanded dramatically, from early limits of 512–2K tokens (GPT-
 502 2/3; (Radford et al., 2019; Brown et al., 2020)) to 128K–1M (Claude-3, GPT-4-Turbo, Gemini 2.0;
 503 (DeepMind, 2025; Anthropic, 2025; OpenAI, 2025a)) and even 10M (Llama 4; (Meta-AI, 2025)).
 504 This growth is driven by advances in efficient attention (sparse, linear, memory-optimized kernels;
 505 (Beltagy et al., 2020; Wang et al., 2020; Dao et al., 2022)), improved positional encodings (relative,
 506 rotary with scaling, ALiBi; (Dai et al., 2019; Peng et al., 2023b)), long-context training strategies
 507 (continued-training, curriculum learning; (Xiong et al., 2023; Ding et al., 2024)), and inference
 508 optimizations such as paged attention, KV-cache compression, and distributed attention (Kwon et al.,
 509 2023; Zhang et al., 2023; Li et al., 2024; Liu et al., 2023). Such capabilities are especially valuable
 510 for applications involving conversational histories, the main focus of our work.

511 Beyond expanding context windows, models incorporate additional mechanisms for persistent mem-
 512 ory. These include recurrence and compression (Transformer-XL, Compressive Transformer; (Dai
 513 et al., 2019; Rae et al., 2019)), state-space architectures (RWKV, Mamba, Hyena; (Peng et al., 2023a;
 514 Gu & Dao, 2023; Poli et al., 2023)), external memory modules (Memformer, RETRO, RMT; (Wu
 515 et al., 2020; Borgeaud et al., 2022; Fan et al., 2024)), context summarization (AutoCompressor;
 516 (Chevalier et al., 2023)), and retrieval-augmented generation (REALM, RAG, HippoRAG; (Guu
 517 et al., 2020; Lewis et al., 2020; Jimenez Gutierrez et al., 2024)). These approaches complement
 518 larger windows by enabling scalable and persistent long-term reasoning.

519 Existing benchmarks such as DialSim, MSC, LoCoMo, MemoryBank, DuLeMon, PerLTQA, Long-
 520 MemEval, and MemBench (Kim et al., 2024a; Xu et al., 2021; Maharana et al., 2024; Zhong et al.,
 521 2024; Xu et al., 2022; Du et al., 2024; Tan et al., 2025) evaluate recall, temporal reasoning, and
 522 multi-session reasoning, but typically span narrow domains, exhibit shallow dependencies, and con-
 523 catenate separate user sessions to simulate long context, reducing realism. Recent work such as
 524 MemoryCode (Rakotonirina et al., 2025) generates multi-session dialogues from template-driven
 525 instruction seeds to assess long-context reasoning, but focuses on a single domain. Our benchmark
 526 instead scales to 10M tokens across diverse topics and introduces new tasks such as contradiction
 527 resolution, event ordering, and instruction following, generating coherent, single-user conversations
 528 that preserve narrative continuity for a more faithful assessment of long-term conversational mem-
 529 ory.

530 6 CONCLUSION

532 This paper addresses the shortcomings of existing benchmarks for evaluating long-term memory in
 533 conversational systems. We introduce a scalable framework to generate BEAM, a new benchmark
 534 with long, coherent dialogues (up to 10M tokens) and diverse memory probes. To improve LLMs
 535 performance, we develop LIGHT, a cognitive-inspired framework combining episodic, working,
 536 and scratchpad memories. Our experiments show that while standard LLMs’ performance degrades
 537 over long contexts, LIGHT provides substantial improvements, boosting memory performance by
 538 an average of 3.5%–12.69%. By offering a more robust evaluation and an effective memory en-
 539 hancement technique, this work helps the development of more reliable long-context conversational
 systems.

540 REFERENCES
541

542 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
543 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical
544 report. *arXiv preprint arXiv:2303.08774*, 2023.

545 Meta AI. Llama 3.3 — model cards and prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/, 2024.

546

547 Anthropic. Claude 3 model card. Technical report, Anthropic PBC, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

548

549 Anthropic. Claude 4 model card (claude opus 4 & sonnet 4). Technical report, Anthropic PBC, May
550 2025.

551

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

554

555 Jeffrey R Binder and Rutvik H Desai. The neurobiology of semantic memory. *Trends in cognitive
556 sciences*, 15(11):527–536, 2011.

557

558 Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Milli-
559 can, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al.
560 Improving language models by retrieving from trillions of tokens. In *International conference on
561 machine learning*, pp. 2206–2240. PMLR, 2022.

562

563 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
564 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
565 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

566

567 Zhihang Chen, Xinyuan Niu, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Broaden your
568 SCOPE! efficient multi-turn conversation planning for LLMs with semantic space. In *The
569 Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=3cgMU3TyyE>.

570

571 Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to
572 compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

573

574 Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas
575 Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention
576 with performers. *arXiv preprint arXiv:2009.14794*, 2020.

577

578 Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdi-
579 nov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint
580 arXiv:1901.02860*, 2019.

581

582 Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-
583 efficient exact attention with io-awareness. *Advances in neural information processing systems*,
35:16344–16359, 2022.

584

585 Google DeepMind. Gemini 2.0 flash: A multimodal model with 1 million token con-
586 text window. <https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash>, 2025.

587

588 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep
589 bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of
590 the North American chapter of the association for computational linguistics: human language
591 technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

592

593 Yiran Ding, Li Lyra Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan
Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv
594 preprint arXiv:2402.13753*, 2024.

594 Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvassy, Pierre-
 595 Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv*
 596 *preprint arXiv:2401.08281*, 2024.

597

598 Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang,
 599 and Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification,
 600 retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on*
 601 *Chinese Language Processing (SIGHAN-10)*, pp. 152–164, 2024.

602

603 Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. Rmt: Retentive networks
 604 meet vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and*
 605 *pattern recognition*, pp. 5641–5651, 2024.

606

607 Chongzhou Fang, Ning Miao, Shaurya Srivastav, Jialin Liu, Ruoyu Zhang, Ruijie Fang, Asmita,
 608 Ryan Tsang, Najmeh Nazari, Han Wang, and Houman Homayoun. Large language models for
 609 code analysis: do llms really do their job? In *Proceedings of the 33rd USENIX Conference on*
 610 *Security Symposium, SEC '24*, USA, 2024. USENIX Association. ISBN 978-1-939133-44-1.

611

612 Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2:
 613 Sparse lexical and expansion model for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.
 614 127–137, 2022.

615

616 Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*
 617 *preprint arXiv:2312.00752*, 2023.

618

619 Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented
 620 language model pre-training. In *International conference on machine learning*, pp. 3929–3938.
 621 PMLR, 2020.

622

623 Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text
 624 degeneration. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=rygGQyrFvH>.

625

626 Hamed Jelodar, Mohammad Meymani, and Roozbeh Razavi-Far. Large language models (llms) for
 627 source code analysis: applications, models and datasets, 2025. URL <https://arxiv.org/abs/2503.17502>.

628

629 Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically
 630 inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569, 2024.

631

632 Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models
 633 hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

634

635 Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

636

637 Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yohan
 638 Jo, and Edward Choi. Dialsim: A real-time simulator for evaluating long-term dialogue understanding
 639 of conversational agents. *arXiv e-prints*, pp. arXiv–2406, 2024a.

640

641 To Eun Kim, Alireza Salemi, Andrew Drozdov, Fernando Diaz, and Hamed Zamani. Retrieval-
 642 enhanced machine learning: Synthesis and opportunities, 2024b. URL <https://arxiv.org/abs/2407.12982>.

643

644 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph
 645 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 646 serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

647

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn
 648 conversation, 2025. URL <https://arxiv.org/abs/2505.06120>.

648 Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired
 649 reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024.
 650

651 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,
 652 Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented gener-
 653 ation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:
 654 9459–9474, 2020.

655 Minghan Li, Miyang Luo, Tianrui Lv, Yishuai Zhang, Siqi Zhao, Ercong Nie, and Guodong Zhou.
 656 A survey of long-document retrieval in the plm and llm era, 2025. URL <https://arxiv.org/abs/2509.07759>.
 657

658 Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle
 659 Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before
 660 generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.
 661

662 Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-
 663 infinite context. *arXiv preprint arXiv:2310.01889*, 2023.
 664

665 Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and
 666 Yuwei Fang. Evaluating very long-term conversational memory of llm agents. *arXiv preprint*
 667 *arXiv:2402.17753*, 2024.

668 Meta-AI. The llama 4 herd: The beginning of a new era of natively multimodal
 669 ai innovation. Meta AI Blog, April 2025. URL <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.
 670

671 Ha Thanh Nguyen, Wachara Fungwacharakorn, May Myo Zin, Randy Goebel, Francesca Toni,
 672 Kostas Stathis, and Ken Satoh. Llms for legal reasoning: A unified framework and future
 673 perspectives. *Computer Law Security Review*, 58:106165, 2025. ISSN 2212-473X. doi:
 674 <https://doi.org/10.1016/j.clsr.2025.106165>. URL <https://www.sciencedirect.com/science/article/pii/S2212473X25000380>.
 675

676 Beijing Academy of Artificial Intelligence. Baai/bge-small-en-v1.5. Hugging Face model, 2023.
 677 URL <https://huggingface.co/BAAI/bge-small-en-v1.5>. MIT License; embed-
 678 ding model.
 679

680 OpenAI. Introducing gpt-4.1 in the api. <https://openai.com/index/gpt-4-1/>, 2025a.
 681

682 OpenAI. Gpt-4.1-mini model card. <https://platform.openai.com/docs/models#gpt-4-1-mini>, 2025b. Accessed: 2025-09-11.
 683

684 Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman,
 685 Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for
 686 the transformer era. *arXiv preprint arXiv:2305.13048*, 2023a.
 687

688 Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window
 689 extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023b.
 690

691 Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua
 692 Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional
 693 language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR,
 694 2023.

695 Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin.
 696 Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework.
 697 *arXiv preprint arXiv:2411.09607*, 2024.
 698

699 Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff,
 700 Hoa Trang Dang, and Jimmy Lin. The great nugget recall: Automating fact extraction and rag
 701 evaluation with large language models. In *Proceedings of the 48th International ACM SIGIR
 Conference on Research and Development in Information Retrieval*, pp. 180–190, 2025.

702 Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases
 703 enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.
 704

705 Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language under-
 706 standing by generative pre-training. 2018.

707 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language
 708 models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

709

710 Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive
 711 transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

712

713 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
 714 Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text
 715 transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

716

717 Nathanaël Carraz Rakotonirina, Mohammed Hamdy, Jon Ander Campos, Lucas Weber, Alberto
 718 Testoni, Marzieh Fadaee, Sandro Pezzelle, and Marco Del Tredici. From tools to teammates:
 719 Evaluating llms in multi-session coding interactions. *arXiv preprint arXiv:2502.13791*, 2025.

720

721 Alice Rueda, Mohammed S. Hassan, Argyrios Perivolaris, Bazen G. Teferra, Reza Samavi, Sirisha
 722 Rambhatla, Yuqi Wu, Yanbo Zhang, Bo Cao, Divya Sharma, Sridhar Krishnan, and Venkat Bhat.
 723 Understanding llm scientific reasoning through promptings and model’s explanation on the an-
 724 swers, 2025. URL <https://arxiv.org/abs/2505.01482>.

725

726 Alireza Salemi and Hamed Zamani. Learning to rank for multiple retrieval-augmented mod-
 727 els through iterative utility maximization. In *Proceedings of the 2025 International ACM SI-
 728 GIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR), ICTIR
 729 '25*, pp. 183–193, New York, NY, USA, 2025. Association for Computing Machinery. ISBN
 9798400718618. doi: 10.1145/3731120.3744584. URL <https://doi.org/10.1145/3731120.3744584>.

730

731 Alireza Salemi, Chris Samarinis, and Hamed Zamani. Plan-and-refine: Diverse and comprehensive
 732 retrieval-augmented generation, 2025. URL <https://arxiv.org/abs/2504.07794>.

733

734 Sruthi Sridhar, Abdulrahman Khamaj, and Manish Kumar Asthana. Cognitive neuroscience per-
 735 spective on memory: overview and summary. *Frontiers in human neuroscience*, 17:1217093,
 2023.

736

737 Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: En-
 738 hanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

739

740 Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. Membench:
 741 Towards more comprehensive evaluation on the memory of llm-based agents. *arXiv preprint
 742 arXiv:2506.21605*, 2025.

743

744 Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer,
 745 Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal under-
 746 standing across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

747

748 Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.

749

750 Timothy J Teyler and Pascal DiScenna. The hippocampal memory indexing theory. *Behavioral
 751 neuroscience*, 100(2):147, 1986.

752

753 Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention
 754 with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

755

756 Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval:
 757 Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*,
 2024.

756 Qingyang Wu, Zhenzhong Lan, Kun Qian, Jing Gu, Alborz Geramifard, and Zhou Yu. Memformer:
757 A memory-augmented transformer for sequence modeling. *arXiv preprint arXiv:2010.06891*,
758 2020.

759 Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to
760 advance general chinese embedding, 2023.

762 Wenzhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin,
763 Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling
764 of foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

765 Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain
766 conversation. *arXiv preprint arXiv:2107.07567*, 2021.

768 Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang
769 Wang. Long time no see! open-domain conversation with long-term persona memory. *arXiv*
770 *preprint arXiv:2203.05797*, 2022.

771 Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago
772 Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for
773 longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

775 Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song,
776 Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient
777 generative inference of large language models. *Advances in Neural Information Processing Systems*,
778 36:34661–34710, 2023.

779 Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large
780 language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial*
781 *Intelligence*, volume 38, pp. 19724–19731, 2024.

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810 A DETAILED RELATED WORK
811
812
813

814 **Long-Context Large Language Models.** The context window of LLMs has expanded from
815 512–2,048 tokens in early models (GPT-1/2/3, BERT, T5; (Radford et al., 2018; 2019; Brown et al.,
816 2020; Devlin et al., 2019; Raffel et al., 2020)) to 128K–1M tokens in recent systems (Claude-3,
817 GPT-4-Turbo, Gemini 1.5 Pro, Gemini 2.0 Flash, Claude-4, GPT-4.1; (Anthropic, 2024; Achiam
818 et al., 2023; Team et al., 2024; DeepMind, 2025; Anthropic, 2025; OpenAI, 2025a)), with some
819 reaching 10M tokens (Llama 4 Scout; (Meta-AI, 2025)). This growth has been enabled by inno-
820 vations that address the quadratic cost of self-attention, including sparse mechanisms (Longformer,
821 BigBird; (Beltagy et al., 2020; Zaheer et al., 2020)), linear approximations (Linformer, Performer;
822 (Wang et al., 2020; Choromanski et al., 2020)) and memory-efficient kernels (FlashAttention; (Dao
823 et al., 2022)). Advances in positional encoding, such as relative encodings (Transformer-XL; (Dai
824 et al., 2019)), rotary embeddings (RoPE; (Su et al., 2024)) with scaling methods (YaRN, NTK;
825 (Peng et al., 2023b)), and linear biases (ALiBi; (Press et al., 2021)), have extended usable context
826 lengths. Training strategies like continued pre-training and curriculum learning (e.g., LLaMA-2-
827 Long (Xiong et al., 2023), LongRoPE (Ding et al., 2024)) further expand capabilities, while in-
828 ference optimizations such as PagedAttention (Kwon et al., 2023), KV-cache compression (H2O,
829 SnapKV; (Zhang et al., 2023; Li et al., 2024)) and distributed approaches (Ring Attention; (Liu
et al., 2023)) enable practical deployment at scale.

830 **Long-Term Memory Methods.** Researchers have developed approaches to enhance long-
831 term memory beyond simply extending context windows. Architectural modifications include
832 Transformer-XL (Dai et al., 2019), which introduced segment-level recurrence, and Compressive
833 Transformer (Rae et al., 2019), which stored both recent states and compressed older information.
834 State-space models such as RWKV (Peng et al., 2023a), Mamba (Gu & Dao, 2023), and Hyena
835 (Poli et al., 2023) replace attention with recurrent dynamics, allowing linear scaling and theoreti-
836 cally unbounded memory. Memory-augmented transformers such as Memformer (Wu et al., 2020),
837 RETRO (Borgeaud et al., 2022) and RMT (Fan et al., 2024) add external memory slots for explicit
838 storage and recall. Context compression offers an orthogonal strategy by summarizing past infor-
839 mation rather than storing it verbatim, as in AutoCompressor (Chevalier et al., 2023), which learns
840 compact, information-preserving representations to reduce token usage. Retrieval-augmented gen-
841 eration (RAG) scales further by maintaining external knowledge stores: REALM (Guu et al., 2020)
842 and RAG (Lewis et al., 2020) pioneered dense retrieval, RETRO (Borgeaud et al., 2022) integrated
843 retrieval into transformers, and HippoRAG (Jimenez Gutierrez et al., 2024) incorporated structured
844 knowledge graphs.

845 Building on these foundations, we propose a novel retrieval-augmented method that shows substan-
846 tial improvements over baselines in long-memory evaluation.

847 **Long-Term Memory Benchmarks.** Several benchmarks have emerged to evaluate long-term mem-
848 ory capabilities in LLMs. DialSim (Kim et al., 2024a) derives evaluation data from multiparty
849 television scripts, producing dialogues extending to 350K tokens with naturalistic patterns but lim-
850 ited topical diversity. MSC (Xu et al., 2021) introduces multisession human-assistant conversations
851 testing memory across session boundaries, though with brief sessions and shallow dependencies.
852 LoCoMo (Maharana et al., 2024) presents 50 conversations averaging 9K tokens in 35 sessions,
853 while MemoryBank (Zhong et al., 2024) provides 300 sessions with 194 probing questions evaluat-
854 ing recall and temporal reasoning. DuLeMon (Xu et al., 2022) focuses on dialogue-level memory
855 and forgetting curves, PerLTQA (Du et al., 2024) targets memory classification and retrieval, and
856 LongMemEval (Wu et al., 2024) constructs multisession evaluations with 500 questions testing in-
857 formation extraction and temporal reasoning. More recently, MemBench (Tan et al., 2025) evaluates
858 the memory of LLM-based agents by assessing their performance on information extraction, multi-
859 hop reasoning, knowledge updating, preference following, and temporal reasoning. Recent work
860 such as MemoryCode (Rakotonirina et al., 2025) generates multi-session dialogues from template-
driven instruction seeds to assess long-context reasoning, but focuses on a single domain.

861 As summarized in Table 2, the existing benchmarks are largely based on concatenated short sessions
862 with limited coherence, narrow personal and casual domains, and few memory abilities. They also
863 lack realistic bidirectional interactivity. In contrast, our benchmark spans diverse domains, scales up
to 10M tokens, and introduces three additional dimensions—contradiction resolution, event order-

Table 2: Comparison of our benchmark with existing long-term memory benchmarks. Memory abilities: IE = Information Extraction, MR = Multi-hop Reasoning, KU = Knowledge Update, TR = Temporal Reasoning, ABS = Abstention, CR = Contradiction Resolution, EO = Event Ordering, IF = Instruction Following, PF = Preference Following, SUM = Summarization.

ing, and instruction following—yielding a more comprehensive framework for evaluating long-term memory in conversational systems.

B BENCHMARK DESIGN

B.1 DATASET STATISTICS

Table 3 summarizes the statistics of the generated dataset, including averages of user messages, assistant messages, assistant and user follow-up questions, and dialogue turns across different chat sizes.

Table 3: Statistics of the dataset. Reported values are averages per chat in each chat size. # User Messages and # Assistant Messages denote the average number of utterances from the user and assistant, respectively. # Answer Assistant Questions is the number of times the assistant posed a question that the user answered. # Followup Questions is the number of follow-up questions asked by the user. # Turns refers to the total number of dialogue turns.

Chat Size	# User Messages	# Assistant Messages	# Answer Assistant Questions	# Followup Questions	# turns
128K	144	144	27	216	107
500K	544	544	79	51	416
1M	1067	1067	105	120	842
10M	10435	10435	1151	1528	7757

B.2 BENCHMARK QUALITY EVALUATION

To evaluate the quality of the generated conversations, we conducted human assessment across all conversations. Two annotators rated each conversation on three dimensions using a 5-point Likert scale (1 = lowest, 5 = highest): *Coherence and Flow*, *Dialogue Realism*, and *Complexity and Depth*.

- **Coherence and Flow:** Conversation continuity (each turn follows naturally from the previous one), smooth transitions across topics and responses, and thread consistency without abrupt or jarring shifts.
- **Dialogue Realism:** Naturalness of user queries (messages sound authentic), realistic progression of topics over time, human-like interactions (appropriate clarifications, follow-ups, etc.), and believability of scenarios.

918
 919 • **Complexity and Depth:** Handling of multi-layered, interconnected topics, progressive
 920 increase in difficulty, and demonstration of domain expertise when required.
 921

921 The aggregated results are reported in Table 4.
 922

923 Table 4: Conversations quality human evaluation results (1–5 scale). Higher is better.
 924

925 Chat Size	926 Coherence and Flow	927 Dialogue Realism	928 Complexity and Depth
926 128K	927 4.4	928 4.55	929 4.35
927 500K	928 4.49	929 4.4	930 4.63
928 1M	929 4.66	930 4.54	931 4.6
929 10M	930 4.6	931 4.8	932 5
930 Average	931 4.53	932 4.57	933 4.64

931
 932 **B.3 BENCHMARK CREATION DETAILS**
 933

934 **B.3.1 DOMAIN COVERAGE OF THE DATASET**
 935

936 To ensure broad coverage and realism, our dataset spans a diverse set of domains. The collection
 937 includes both technical and non-technical conversations, ranging from specialized domains such as
 938 coding, mathematics, financial investment and health to personal and social domains such as therapy,
 939 lifestyle, and trip planning. In total, we designed 100 multi-turn chats distributed across 19 domains,
 940 each represented by a set of distinct titles that capture the thematic scope of the dialogues. The full
 941 list of domains and their associated chat titles is provided in Table 5.
 942

943 Table 5: Domains and associated chat titles in our dataset (100 total chats).
 944

944 Domain	945 Chat Titles
945 Coding	946 Designing a Large-Scale Retrieval-Augmented Generation (RAG) System 947 for Enterprise Search • Creating a Self-Driving Car Simulation Environment 948 • Developing a Multi-Agent AI Research Platform • Building a Multi-Language AI Chatbot with Contextual Memory • Developing a Personalized News Aggregator with AI Summarization • Creating an Autonomous Stock Trading Bot • Implementing a Custom Image 949 Captioning Model • Building a Multiplayer Online Game with Real-Time Physics • Building a Real-Time Chat Application with Node.js 950 and Socket.io • Creating an AI-Powered Resume Analyzer with Python and NLP • Developing a Computer Vision App for Real-Time Object Detection • Creating a Restaurant Recommendation System • Automating Social Media Posts with Python • Building a Personal Budget 951 Tracker Web App in Python and Flask • Creating a Command-Line To-Do List Manager in Go • Developing a Weather Forecast App in 952 JavaScript with OpenWeather API • Training a Spam Email Classifier Using Python and Scikit-learn • Building a Portfolio Website with 953 HTML, CSS, and Bootstrap
954 Math	955 Partial Differential Equations (PDEs) in Depth • Functional Analysis and Infinite-Dimensional Spaces • Solving Ordinary Differential Equations (ODEs) • Deep Dive into Number Theory • Advanced Probability and Combinatorics • Exploring Non-Euclidean Geometry • Studying Multivariable Calculus • Diving into Analytic Geometry • Developing Skills in Mathematical Induction • Exploring Conic Sections in Depth 956 • Understanding Sequences and Series • Mastering Basic Differential Calculus • Exploring the Geometry of Triangles • Understanding the Basics of Probability • Mastering Algebraic Equations for Everyday Problem Solving • Learning the Foundations of Trigonometry • Mastering Fractions, Decimals, and Percentages

972	973	Domain	Chat Titles
974	975	Writing Assistant & Learning	Building a Portfolio-Ready Resume that Passes Any Applicant Tracking System • Mastering the Art of Persuasive Academic Essay Writing • Crafting a Standout Cover Letter for Competitive Job Markets • Designing a Multi-Purpose Personal Statement for Global Opportunities • Developing a Self-Editing System for Lifelong Writing Improvement
976	977	Therapy & Emotional Support	Recovering from Workplace Burnout and Chronic Stress • Healing After the Loss of a Loved One • Overcoming Childhood Trauma and Rebuilding Self-Trust • Coping with Post-Breakup Emotional Pain and Relationship Trauma
978	979	Career & Professional Development	Advancing from Mid-Level to Senior Leadership Roles • Building a Powerful Professional Network from Scratch • Landing Your Next Job: From Resume to Job Offer • Designing a 5-Year Career Growth Plan • Positioning Yourself for a Promotion
980	981	Financial Investment	Building a Long-Term Stock Market Investment Strategy • Getting Started in Real Estate Investing • Navigating the World of Cryptocurrency • Creating a Balanced Investment Portfolio
982	983	Health & Wellness	Creating a Personalized Nutrition and Meal Planning System • Designing a Sustainable Fitness Routine • Improving Sleep Quality for Better Health • Understanding and Managing Chronic Illness • Recognizing Symptoms and Seeking Medical Help Early
984	985	Relationship & Family	Strengthening Communication in Romantic Relationships • Parenting Through Different Life Stages • Navigating In-Law and Extended Family Relationships • Rebuilding Relationships After Trust Has Been Broken
986	987	Education & Learning	Learning to Play a Musical Instrument from Scratch • Mastering a New Language for Real-World Communication • Becoming a Skilled Photographer • Exploring Performing Arts: Acting, Theater, and Dance
988	989	Home & Real Estate	Buying Your First Home with Confidence • Renting a Home or Apartment Without Stress • Selling Your Home for Maximum Value • DIY Home Improvement and Repairs • Making Your Home More Comfortable and Functional
990	991	Lifestyle	Designing a Daily Routine That Boosts Productivity and Well-Being • Building Healthy and Sustainable Lifestyle Habits • Balancing Social Life and Personal Time
992	993	Cooking	Mastering Quick and Healthy Weeknight Dinners • Baking Like a Pro at Home • Exploring Global Cuisines from Your Kitchen • Cooking for Special Diets and Allergies • Meal Prepping for the Week Ahead
994	995	Business & Entrepreneurship	Starting a Business from Scratch • Growing and Scaling Your Small Business • Building a Successful Startup
996	997	Trip Planning	Preparing for a Week-Long Hiking and Camping Adventure in Patagonia • Organizing a Cross-Country USA Road Trip • Planning a Cultural Immersion Trip to Japan • Planning a Budget Backpacking Trip Across Southeast Asia • Arranging a Luxury Honeymoon in the Maldives
998	999	Sport	Soccer – Playing, Watching, and Supporting the World’s Most Popular Game • Basketball – From Street Courts to the NBA • Volleyball – Indoor, Beach, and Competitive Play • Hockey – Ice, Field, and Global Competitions • Tennis – From Local Courts to Grand Slams
1000	1001		
1002	1003		
1004	1005		
1006	1007		
1008	1009		
1010	1011		
1012	1013		
1014	1015		
1016	1017		
1018	1019		
1020	1021		
1022	1023		
1024	1025		

1026	Domain	Chat Titles
1027		
1028	Event Planning	Planning a Surprise 30th Birthday Party for a Close Friend • Coordinating a Destination Beach Wedding for 100 Guests • Organizing a Weekend Community Food and Music Festival • Planning a Cozy Christmas Eve Dinner for Extended Family
1029		
1030		
1031		
1032	Asking Recommendation	Finding the Perfect Smartphone for Photography and Gaming • Choosing a Lightweight Laptop for Work, Travel, and Entertainment • Selecting a Must-Read Fiction Series for Winter Evenings • Finding the Best Streaming Movies for a Family Weekend • Choosing Comfortable and Stylish Sneakers for Daily Wear
1033		
1034		
1035		
1036		
1037	Legal & Administrative	Filing for a Marriage-Based Green Card in the United States • Creating a Legally Valid Will and Estate Plan • Applying for a Patent to Protect a New Invention
1038		
1039		
1040	Philosophical & Ethical Discussion	Deciding Whether to Use AI to Automate Hiring in My Company • Considering Whether to Believe in and Live by the Idea of Free Will
1041		
1042		
1043		
1044	B.3.2 CONVERSATION PLAN GENERATION	
1045		
1046	A <i>conversation plan</i> serves as the central scaffold of each conversation, providing a coherent storyline that evolves chronologically. The process of constructing conversation plans is anchored by a <i>seed</i> that specifies the <i>domain</i> of the dialogue (e.g., sports, finance, programming, mathematics), a <i>title</i> representing the high-level topic, and a <i>theme</i> that provides a more detailed instantiation of the title. The seed also includes a set of <i>subtopics</i> , which enumerate finer-grained subtopics and details to ensure topical diversity. However, a title, theme, and subtopics alone are insufficient to support detailed and information-rich conversations. To enrich the narrative, we introduce <i>narratives set</i> that define the evolving aspects of a conversation (e.g., career progression, goals, relationships). Each narrative is paired with descriptive details that specify its scope and trajectory.	
1047		
1048		
1049		
1050		
1051		
1052		
1053		
1054	In addition to the seed and narrative set, each conversation incorporates a <i>user profile</i> , a <i>relationship graph</i> , and an explicit <i>timeline</i> . The user profile includes attributes such as name, age, gender, location, profession, and personality traits. To avoid redundancy, personality traits are grounded in the Myers–Briggs Type Indicator (MBTI). Specifically, we randomly select six MBTI types, provide their descriptions, and instruct an LLM to synthesize a composite trait profile, enabling the creation of 8,008 unique user profiles. Relationship graphs are then constructed, linking the main user to family members (parents, partner, children), friends, and acquaintances, subject to constraints (e.g., plausible age gaps) to preserve realism. The timeline specifies the temporal span of the conversation, defining the range between its beginning and end.	
1055		
1056		
1057		
1058		
1059		
1060		
1061		
1062		
1063	In order to generate titles and themes of the chats, target domains are first specified by human. Given these domains, GPT-4.1 (OpenAI, 2025a) is prompted using the prompt shown in Listing 22 in Appendix H, to produce candidate titles, themes, and subtopics. These candidates are refined by human to ensure topical diversity by removing the similar chat titles and selecting diverse chat titles. Finally, for each conversation, we generate 15–20 narratives using open-source LLaMA-3.3 70B (AI, 2024) with the prompt shown in Listing 23 to save cost. In this prompt, given the conversation seed as input, the LLM produces narratives that capture evolving aspects of the storyline, providing the backbone for constructing coherent conversation plans.	
1064		
1065		
1066		
1067		
1068		
1069		
1070		
1071	Conversation plans are structured as a sequence of N <i>sub-plans</i> , where each sub-plan corresponds to a distinct stage of the conversation. Each sub-plan contains a fixed number of M <i>bullet-points</i> , and each bullet-point is defined by a <i>narrative</i> and a descriptive statement specifying how that narrative unfolds in the storyline. To maintain temporal coherence, each sub-plan also includes a <i>time anchor</i> specifying a concrete date or period.	
1072		
1073		
1074		
1075		
1076	For conversations of sizes 128K, 500K, and 1M tokens, a single conversation plan is generated, as shown in line 4 of Algorithm 1 in Appendix B.3.5. The plan is produced by conditioning the LLM on the conversation seed, user profile, relationship graph, timeline, the number of sub-plans, the number of bullet points within each sub-plan and narrative set, using the prompt shown in Listing 24 in Appendix H. The number of sub-plans is not fixed but varies with both the domain and the target	
1077		
1078		
1079		

1080 conversation length, in order to adhere to the length budget. For instance, domains such as coding
 1081 typically require fewer dialogue turns to reach the same token budget compared to more general
 1082 domains.

1083 For 10M-token conversations, a single plan cannot adequately capture the scope and continuity
 1084 required at this scale. To address this, we construct ten distinct yet interlocking conversation plans
 1085 that together produce a coherent long-term narrative. While the process begins with a main seed that
 1086 defines the global topic and theme of the conversation, a single seed is insufficient for producing
 1087 ten plans. Instead, we generate ten distinct conversation seeds—one for each plan—so that the
 1088 narrative can unfold across multiple stages. The procedure for deriving these seeds—and the plans
 1089 that follow—differs depending on the strategy. We propose two strategies for constructing them:
 1090

- 1091 • **Sequential Expansion:** The conversation seed is used as the first seed in the sequence. The
 1092 remaining seeds are generated to represent successive stages of the user’s life, extending the sto-
 1093 ryline chronologically. For instance, if the main seed concerns an international trip, the first plan
 1094 covers the trip itself, the second covers the period after returning (e.g., job search), and subse-
 1095 quent seeds correspond to later milestones. We generate these seeds using the prompt shown in
 1096 Listing 28, which conditions on the main seed, user profile, and timeline to produce a sequence
 1097 of temporally aligned seeds. Each conversation plan is then generated sequentially, with every
 1098 plan conditioned on its predecessor to maintain continuity, as specified in line 12 of Algorithm 1
 1099 in Appendix B.3.5. The plans are generated using the prompt shown in Listing 30, yielding a
 1100 temporally ordered series of interconnected narrative arcs. To maintain realism, the user’s core
 1101 relationships (e.g., parents, children, partner) remain fixed across plans, while new acquaintances
 1102 are gradually introduced.
- 1103 • **Hierarchical Decomposition:** Instead of extending the seed chronologically, the main seed is
 1104 decomposed into ten sub-seeds, each corresponding to a distinct topical or temporal slice of the
 1105 overall storyline. Together, these seeds span the full narrative. For example, if the main seed
 1106 concerns an international trip, the first three seeds may cover preparation steps (e.g., reservations,
 1107 document gathering), the next five capture events during the trip, and the final two represent
 1108 post-trip activities (e.g., reflections, recounting experiences). Like in Sequential Expansion, the
 1109 user’s core relationships (e.g., parents, children, partner) remain fixed across plans, while new
 1110 acquaintances are gradually introduced. We generate these ten sub-seeds using the prompt shown
 1111 in Listing 29, which takes the main seed, user profile, and timeline, and outputs ten derived seeds.

1112 Each plan is assigned explicit topical and temporal boundaries to prevent redundancy or thematic
 1113 overlap, ensuring that sub-themes unfold in the correct stage of the narrative. These boundaries are
 1114 encoded in the conversation seed itself. For coherence, summaries of all prior plans are provided
 1115 to the LLM when generating a new plan, allowing contextual references to past events. Moreover,
 1116 when generating each plan, future seeds are also supplied, encoding their own topical and temporal
 1117 boundaries. This design allows earlier plans to anticipate upcoming events with consistent references
 1118 (e.g., booking tickets for the correct travel dates before the trip actually occurs). This strategy
 1119 is implemented in line 20 of Algorithm 1 in Appendix B.3.5. Conversation plans are generated
 1120 using the prompt shown in Listing 31, which takes as input the main seed, the current sub-seed, the
 1121 number of sub-plans, the narrative set, the user profile, core and newly introduced relationships, the
 1122 preceding and subsequent sub-seeds, the previous plan, the summary of all previous plans, the index
 1123 of the current sub-seed, and a binary indicator specifying whether the plan is the first in the sequence
 1124 (in which case the introduction of the user is included). The output is a fully specified conversation
 1125 plan.

1126 After the conversation plan is constructed, it is expanded into user-turn questions and subsequently
 1127 assistant responses, yielding complete dialogues that can be used to evaluate memory abilities. How-
 1128 ever, in its initial form, the plan may not include sufficient information to evaluate three critical mem-
 1129 ory abilities: *contradiction resolution*, *knowledge update*, and *instruction following*. To address this,
 1130 after the initial plan generation, we pass the plan to GPT-4.1 to generate high-quality plans and aug-
 1131 ment each sub-plan with additional bullet points specifically designed to enable evaluation of these
 1132 abilities. Importantly, this augmentation is performed in a second stage rather than during the initial
 1133 plan generation, since incorporating such information directly in a single-pass generation leads to
 1134 lower quality and less reliable coverage of these abilities. The augmentation is implemented using
 1135 the prompt shown in Listing 27, which takes an existing conversation plan as input and outputs a
 1136 revised version where each sub-plan includes three additional bullet points targeting these abilities.

1134 B.3.3 USER UTTERANCE GENERATION
1135

1136 Once conversation plans are constructed, user turns are synthesized directly from them. Each sub-
1137 plan within a conversation plan consists of M bullet-points, which are partitioned into K contiguous
1138 batches of equal size. Partitioning is performed sequentially, such that each batch corresponds to
1139 a consecutive segment of the sub-plan. Partitioning is necessary because conditioning the LLM
1140 on an entire sub-plan at once tends to yield repetitive or low-quality questions; batching mitigates
1141 this by narrowing the focus of generation. For each batch, the LLM produces I user questions
1142 (line 6 of Algorithm 2 in Appendix B.3.5) using the prompt presented in Listing 32. The model is
1143 conditioned on the conversation seed, the current batch specification, preceding batches within the
1144 same sub-plan, and contextual information from earlier sub-plans. This setup ensures that generated
1145 questions remain grounded in prior context, yielding conversations that are coherent and continuous
1146 over extended spans.

1147 The values of K and I vary depending on the domain and the target conversation length, in order
1148 to adhere to the overall length budget. We specify the values for K and I manually. The specific
1149 configurations of K and I across domains and conversation sizes are reported in Table 6. This
1150 provides fine-grained control over the density of user interactions and helps prevent both under-
1151 generation and excessive redundancy. Additionally, to better capture domain-specific conversational
1152 patterns, we incorporate domain-specific features during question generation:

- 1153 • **Programming:** To reflect realistic developer–assistant interactions, we incorporate questions that
1154 involve sharing code snippets. These include (i) buggy code requiring debugging assistance, (ii)
1155 correct code seeking optimization, and (iii) natural language descriptions of desired functionality
1156 for which code is requested. We use the prompt shown in Listing 33 to generate questions specific
1157 to the programming domain.
- 1158 • **Mathematics.** To capture authentic problem-solving dynamics, we incorporate questions that
1159 involve sharing mathematical work, requesting corrections, asking for the next logical step in a
1160 solution, or introducing problems to be solved. We use the prompt shown in Listing 34 to generate
1161 questions specific to the mathematics domain.

1162 To reduce computational cost while maintaining generation quality, question generation is per-
1163 formed using the open-source LLaMA-3.3 70B model (AI, 2024), which produces high-quality
1164 questions.

1165 B.3.4 ASSISTANT UTTERANCE GENERATION
1166

1167 After generating user-side questions, assistant-side responses are generated in an iterative, role-
1168 playing framework where one LLM assumes the *assistant role* and another assumes the *user role*.
1169 For each sub-plan, the assistant LLM is conditioned on the seed as explained in Section 2.2.1,
1170 prior sub-plans of the conversation plan, a summary of the most recent M dialogue turns, and a
1171 compressed summary of older turns (generated using the prompt shown in Listing 37). For 10M-
1172 token conversations, additional summaries of prior plans are also provided.

1173 The response generation process unfolds as an iterative interaction between the assistant and user
1174 roles. First, the assistant LLM produces an answer to the user’s most recent question (line 9). This
1175 output is then analyzed by a *question-detection module*, which determines whether the assistant’s
1176 response contains a counter-question directed at the user (line 11), using the prompt shown in List-
1177 ing 35 that takes the assistant response as input and outputs *yes* if a question is present and *no*
1178 otherwise. If such a counter-question is detected, the response—together with the current and pre-
1179 vious sub-plans, relevant past context, and conversation summaries—is passed to the user LLM,
1180 which generates a realistic reply that reflects the storyline and contextual details using the prompt
1181 shown in Listing 38 (line 14). This new user reply is subsequently passed back to the assistant LLM,
1182 continuing the conversation. This loop repeats until no further assistant questions are detected or the
1183 predefined threshold δ_1 (which is set to two) is reached, preventing infinite cycles. For δ_1 we tested
1184 values 2, 3 and 5 which we selected 2 as it produces more realistic dialogues.

1185 Beyond direct question–answer exchanges, a *follow-up detection module* (line 21) evaluates
1186 whether, in a realistic setting, the user would naturally ask a clarifying or elaborative follow-up.
1187 The need for a follow-up is determined using the prompt shown in Listing 36, which takes as input
1188 the seed, dialogue history, and the assistant’s most recent response, and outputs *yes* or *no*. This

1188 decision is guided by factors such as subject complexity, ambiguity in the assistant’s answer, or incompleteness of the response. When a follow-up is required, the module conditions on the seed, the
 1189 current and prior sub-plans, the most recent M turns, and summaries of earlier turns to generate the
 1190 follow-up query using the prompt shown in Listing 39. The generated query is then passed back to
 1191 the assistant LLM for resolution. As with the assistant-question loop, a strict threshold δ_2 (which is
 1192 set to two like δ_1) limits the number of follow-up exchanges, preventing unbounded cycles.
 1193

1194 Through the interaction of these two threshold-controlled modules, the system produces conversa-
 1195 tions that exhibit naturalistic bidirectional dynamics, rich contextual references, and realistic clarifi-
 1196 cation behaviors characteristic of human–AI dialogues.
 1197

1198

1199

1200

1201

1202 **B.3.5 ALGORITHMS**

1203

1204

1205

1206

1207

1208

1209

Algorithm 1 Conversation plan generation.

1210 **Input:** domain c , length budget L , title θ , theme τ , subtopics Σ , user profile u , user relationships ρ ,
 1211 timeline Γ , number of conversation sub-plans N , number of bullet-points in each conversation
 1212 sub-plan M , generator G

1213 **Output:** Conversation plan set p

```

1214 1:  $S \leftarrow (c, \theta, \tau, \Sigma)$                                      ▷ Initialize seed
1215 2: if  $L \in \{128K, 500K, 1M\}$  then
1216 3:    $\Lambda \leftarrow G(S)$                                          ▷ Generate narratives using Listing 23
1217 4:    $P \leftarrow G(S, u, \rho, \Gamma, N, M, \Lambda)$                   ▷ Generate a single conversation plan with Listing 24
1218 5: else if  $L = 10M$  then
1219 6:    $P \leftarrow \{\}$                                          ▷ Initialize set of plans
1220 7:   if  $\sigma = \text{Sequential Expansion}$  then
1221 8:      $S' \leftarrow G_{\text{seeds}}(S, \Gamma)$                          ▷ Generate sequential sub-seeds with Listing 28
1222 9:     for each  $s'_i \in S'$  do
1223 10:     $\Lambda_i \leftarrow G(s'_i)$                                          ▷ Generate narratives for sub-seed
1224 11:     $b \leftarrow \mathbf{1}[i = 0]$                                          ▷ Binary indicator: 1 if first plan, else 0
1225 12:     $P_i \leftarrow G(s'_i, \Gamma_i, N, \Lambda_i, u, \rho, P_{i-1}, i, b)$     ▷ Generate plan with Listing 30
1226 13:     $P \leftarrow P \cup \{P_i\}$ 
1227 14:   end for
1228 15: else if  $\sigma = \text{Hierarchical Decomposition}$  then
1229 16:    $S' \leftarrow G_{\text{decompose}}(S, \Gamma)$                          ▷ Decompose seed with Listing 29
1230 17:   for each  $s'_i \in S'$  do
1231 18:      $\Lambda_i \leftarrow G(s'_i)$                                          ▷ Generate narratives for sub-seed
1232 19:      $b \leftarrow \mathbf{1}[i = 0]$ 
1233 20:      $P_i \leftarrow G(S, S', s'_i, \Gamma_i, N, \Lambda_i, u, \rho, P_{i-1}, \overline{P_{0,\dots,i-1}}, i, b)$  ▷ Generate plan with
1234 21:      $P \leftarrow P \cup \{P_i\}$ 
1235 22:   end for
1236 23: end if
1237 24: end if
1238 25: return  $P$ 
  
```

1239

1240

1241

1242 **Algorithm 2** User questions generation.

1243 **Input:** seed S , conversation plan p , number of questions per iteration I , generator G

1244 **Output:** Question set Q

1245 1: $p \leftarrow \{p_1, \dots, p_N\}$ ▷ Conversation plan with N sub-plans

1246 2: $Q \leftarrow \{\}$ ▷ Initialize empty question set

1247 3: **for** each $p_i \in P$ **do**

1248 4: $p_i = \{p_{i1}, \dots, p_{iK}\}$

1249 5: **for** each $p_{ij} \in p_i$ **do**

1250 6: $Q_{ij} \leftarrow G(S, p_{ij}, \{p_{i1}, \dots, p_{i(j-1)}\}, \{p_1, \dots, p_{i-1}\}, I)$ ▷ Generate I questions using Listing 32

1251 7: $Q \leftarrow Q \cup \{Q_{ij}\}$ ▷ Append generated questions to the question set

1252 8: **end for**

1253 9: **end for**

1254 10: **return** Q

1257 **Algorithm 3** Answer generation.

1258 **Input:** question set $Q = \{Q_1, \dots, Q_N\}$, seed S , conversation plan set P , thresholds δ_1, δ_2 , assistant-question detector ϕ , follow-up detector ψ , generator G

1259 **Output:** conversation list T

1260 1: $T \leftarrow \{\}$ ▷ Initialize empty conversation list

1261 2: **for** each $Q_i \in Q$ **do**

1262 3: $Q_i = \{q_1, \dots, q_J\}$ ▷ Questions in sub-plan i

1263 4: **for** each $q_j \in Q_i$ **do**

1264 5: $t \leftarrow \{\}$ ▷ Initialize turn sequence

1265 6: $H_t^{(M)} \leftarrow$ recent- M turn window at turn t

1266 7: $\bar{H}_t \leftarrow$ summary of turns prior to $H_t^{(M)}$

1267 8: $\bar{P}^{(<p)}$ \leftarrow summaries of conversation plans preceding p

1268 9: $a_{ij} \leftarrow G_{\text{assistant}}(S, p_{1:(i-1)}, H_t^{(M)}, \bar{H}_t, \bar{P}^{(<p)})$ ▷ Generate assistant response with Listing 37

1269 10: $t \leftarrow t \cup \{a_{ij}\}$ ▷ Add assistant's response to current dialogues turn

1270 11: $isQ \leftarrow \phi(a_{ij}, H_t^{(M)}, \bar{H}_t)$ ▷ Checks if assistant response contains question from user with Listing 35

1271 12: $count \leftarrow 0$

1272 13: **while** isQ and $count < \delta_1$ **do**

1273 14: $u_{ij} \leftarrow G_{\text{user}}(S, p_i, p_{1:(i-1)}, \bar{P}^{(<p)}, H_t^{(M)}, \bar{H}_t, a_{ij})$ ▷ Generate user's response to assistant question with Listing 38

1274 15: $t \leftarrow t \cup \{u_{ij}\}$ ▷ Add user's response to current dialogues turn

1275 16: $a_{ij} \leftarrow G_{\text{assistant}}(S, p_{1:(i-1)}, H_t^{(M)}, \bar{H}_t, \bar{P}^{(<p)})$ ▷ Generate assistant's response

1276 17: $t \leftarrow t \cup \{a_{ij}\}$ ▷ Add assistant's response to current dialogues turn

1277 18: $count \leftarrow count + 1$

1278 19: $isQ \leftarrow \phi(a_{ij}, H_t^{(M)}, \bar{H}_t)$

1279 20: **end while**

1280 21: $needFU \leftarrow \psi(a_{ij}, H_t^{(M)}, \bar{H}_t, S)$ ▷ Checks if user need to ask followup question with Listing 36

1281 22: $fu_count \leftarrow 0$

1282 23: **while** $needFU$ and $fu_count < \delta_2$ **do**

1283 24: $u_{ij} \leftarrow G_{\text{user}}(S, p_i, p_{1:(i-1)}, \bar{P}^{(<p)}, H_t^{(M)}, \bar{H}_t, a_{ij})$ ▷ Generate user's followup question with Listing 39

1284 25: $t \leftarrow t \cup \{u_{ij}\}$

1285 26: $a_{ij} \leftarrow G_{\text{assistant}}(S, p_{1:(i-1)}, H_t^{(M)}, \bar{H}_t, \bar{P}^{(<p)})$ ▷ Generate assistant's response to user's followup question

1286 27: $t \leftarrow t \cup \{a_{ij}\}$

1287 28: $fu_count \leftarrow fu_count + 1$

1288 29: $needFU \leftarrow \psi(a_{ij}, H_t^{(M)}, \bar{H}_t, S)$

1289 30: **end while**

1290 31: $T \leftarrow T \cup \{t\}$

1291 32: **end for**

1292 33: **end for**

1293 34: **return** T

1296 B.4 USER UTTERANCE GENERATION HYPERPARAMETERS
12971298 Table 6: Batching configuration by chat size and domain category for user-turn question generation.
1299 NUM_SUBPLANS denotes the number of conversation sub-plans, K the number of batches per
1300 sub-plan, and I the number of questions generated per batch.
1301

Chat Size	Category	NUM_SUBPLANS	K	I
128K	General	5	10	2
	Coding	3	23	1
	Math	3	25	1
500K	General	10	10	4
	Coding	10	10	3
	Math	10	10	4
1M	General	10	10	9
	Coding	10	10	6
	Math	10	10	6
10M	General	10	10	9
	Coding	10	10	6
	Math	10	10	6

1318 B.5 CREATED PROBING QUESTIONS DISTRIBUTION
13191320 We measure which parts of the dialogue contain the information required to answer the probing
1321 questions. To this end, each conversation is divided into ten equal segments, and we record the seg-
1322 ment(s) where the supporting evidence for each probing question resides. The detailed methodology
1323 for aligning probing questions with dialogue segments is described in Section 2.3. The resulting dis-
1324 tributions across conversation lengths are reported in Table 7.
13251326 Table 7: Percentage distribution of created probing questions across ten equal chat segments
1327 (deciles) for different chat sizes. Each row corresponds to a segment of the dialogue, moving from
1328 the beginning (Segment 1) to the end (Segment 10).
1329

Chat Segment (Decile)	100K	500K	1M	10M
1	0.00%	0.65%	0.19%	0.00%
2	11.05%	23.70%	21.60%	10.24%
3	14.83%	15.91%	20.11%	16.27%
4	12.79%	14.45%	15.83%	15.06%
5	13.08%	7.95%	9.50%	14.46%
6	13.37%	9.09%	8.01%	9.64%
7	11.92%	6.33%	5.96%	10.24%
8	8.14%	5.52%	5.21%	13.25%
9	9.59%	4.55%	4.47%	8.43%
10	5.23%	11.85%	9.12%	2.41%

1343 B.6 MEMORY ABILITIES EXAMPLES
13441345 To illustrate how our benchmark evaluates different aspects of long-term conversational memory,
1346 we provide representative probing questions and their ideal answers for each of the ten memory
1347 abilities. These examples demonstrate how each ability is operationalized in practice.
1348

1349 1. Abstention (withholding answers when information is missing)

1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403

Probing Question: What specific advice did Manuel give about property management companies during the March 5 Investors Meetup?
Ideal Answer: Based on the provided chat, there is no information related to the specific advice Manuel gave about property management companies.

2. Contradiction Resolution (detecting and reconciling inconsistencies)

Probing Question: Have I ever attended any real estate webinars or investor meetups?
Ideal Answer: I notice you've mentioned contradictory information about this. You said you have never attended any real estate webinars or investor meetups, but you also mentioned attending a webinar about Turkey's rising demand for multi-family rentals. Which statement is correct?

3. Event Ordering (reasoning about chronological sequence)

Probing Question: How did my focus on different aspects of property investment and management develop throughout our conversations in order? Mention ONLY and ONLY ten items.
Ideal Answer: Your focus developed in this sequence: 1) Agent interaction and preparation, 2) Financial evaluation, 3) Financing and mortgage, 4) Contractor management, 5) Zoning and permits, 6) Renovation planning, 7) Viewing preparation, 8) Negotiation and cash flow, 9) Renovation completion, 10) Commercial property considerations.

4. Information Extraction (recalling factual details)

Probing Question: What amount did I say I had set aside as my initial capital for investing?
Ideal Answer: You said you had set aside \$50,000 as your initial capital.

5. Instruction Following (adhering to explicit user constraints)

Probing Question: How should I allocate my funds for the upcoming quarter?
Ideal Answer: The response should include an itemized list of costs, category-by-category breakdown, and detailed cost analysis, reflecting the instruction to always provide detailed financial breakdowns when budgeting.

6. Information Update (revising prior facts when updated)

Probing Question: How much initial capital have I allocated for real estate investing?
Ideal Answer: \$60,000

7. Multi-hop Reasoning (integrating evidence across turns)

Probing Question: How many different banks did I consider for my mortgage options across my conversations?
Ideal Answer: Two banks: Halkbank and Ziraat Bank.

8. Preference Following (adapting to evolving user preferences)

Probing Question: I'm looking at several properties and want to understand how to approach comparing them effectively. What steps would you suggest I take?
Ideal Answer: A structured, step-by-step process that emphasizes data collection and analysis, breaking comparisons into measurable factors, reflecting the user's preference for methodical, data-driven decisions.

1404
1405 9. Summarization (abstraction and compression of content)1406
1407 **Probing Question:** Can you give me a comprehensive summary of my journey and
1408 decision-making process around investing in rental properties, including how my budget,
1409 property choices, management considerations, and financing plans have developed over
1410 time?1411 **Ideal Answer:** Your journey began with an initial capital of \$50,000, followed by ex-
1412 ploration of market conditions, renovation planning, property type tradeoffs, financing
1413 through Halkbank and Ziraat Bank, and a structured plan for purchase and management.1414
1415 10. Temporal Reasoning (reasoning about durations and timelines)1416
1417 **Probing Question:** How many days are there between my first property viewing with
1418 Mehmet Yilmaz and the last one I scheduled?1419 **Ideal Answer:** There are 2 days between the first property viewing on March 25 and the
1420 last one on March 27.

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458 C DETAILED EXPERIMENTS
1459
1460
1461
1462
1463

C.1 ABLATION STUDY

1464 In this section, we present the complete results of our ablation experiments. We evaluate the contribution
1465 of individual components in our proposed module as shown in table 8.
1466
14671468 Table 8: Ablation study showing the impact of removing key memory components (retrieval,
1469 scratchpad, working memory, and noise filtering) on performance across various conversation
1470 lengths (100K–10M).

Length	Memory Ability	Base	w/o Retrieval from Index	w/o Scratchpad	w/o Working Memory	w/o Noise Filtering
100K	Abstention	0.475	0.725	0.600	0.575	0.700
	Contradiction Resolution	0.037	0.043	0.012	0.043	0.018
	Event Ordering	0.205	0.190	0.194	0.220	0.200
	Information Extraction	0.479	0.329	0.510	0.451	0.485
	Instruction Following	0.362	0.375	0.287	0.387	0.312
	Knowledge Update	0.362	0.237	0.350	0.362	0.312
	Multi-Hop Reasoning	0.281	0.201	0.248	0.303	0.181
	Preference Following	0.566	0.675	0.533	0.579	0.491
	Summarization	0.232	0.232	0.143	0.223	0.103
	Temporal Reasoning	0.112	0.075	0.125	0.125	0.087
Average		0.311	0.311	0.300	0.327	0.289
500K	Abstention	0.571	0.571	0.585	0.657	0.585
	Contradiction Resolution	0.017	0.007	0.014	0.017	0.014
	Event Ordering	0.244	0.222	0.266	0.262	0.229
	Information Extraction	0.506	0.254	0.466	0.485	0.464
	Instruction Following	0.295	0.307	0.316	0.334	0.286
	Knowledge Update	0.278	0.192	0.285	0.235	0.314
	Multi-Hop Reasoning	0.214	0.104	0.227	0.192	0.247
	Preference Following	0.571	0.553	0.450	0.547	0.465
	Summarization	0.344	0.312	0.225	0.353	0.203
	Temporal Reasoning	0.121	0.042	0.116	0.114	0.130
Average		0.316	0.256	0.295	0.320	0.294
1M	Abstention	0.500	0.664	0.600	0.557	0.507
	Contradiction Resolution	0.021	0.021	0.035	0.042	0.032
	Event Ordering	0.200	0.215	0.221	0.227	0.199
	Information Extraction	0.366	0.246	0.391	0.397	0.366
	Instruction Following	0.419	0.427	0.335	0.384	0.351
	Knowledge Update	0.357	0.185	0.321	0.400	0.285
	Multi-Hop Reasoning	0.209	0.129	0.227	0.221	0.169
	Preference Following	0.551	0.602	0.536	0.597	0.540
	Summarization	0.316	0.310	0.169	0.330	0.128
	Temporal Reasoning	0.154	0.050	0.111	0.121	0.111
Average		0.309	0.285	0.295	0.328	0.269
10M	Abstention	0.550	0.800	0.650	0.650	0.600
	Contradiction Resolution	0.012	0.000	0.012	0.000	0.000
	Event Ordering	0.197	0.199	0.199	0.209	0.181
	Information Extraction	0.350	0.000	0.200	0.150	0.200
	Instruction Following	0.350	0.175	0.175	0.175	0.050
	Knowledge Update	0.275	0.050	0.300	0.150	0.225
	Multi-Hop Reasoning	0.125	0.000	0.125	0.125	0.075
	Preference Following	0.308	0.191	0.241	0.200	0.175
	Summarization	0.220	0.119	0.068	0.008	0.050
	Temporal Reasoning	0.000	0.000	0.050	0.075	0.000
Average		0.238	0.153	0.202	0.181	0.155

1501
1502
1503
1504
1505 C.2 RETRIEVAL BUDGET
1506
1507
1508
1509
1510
1511We investigate the impact of the retrieval budget through two sets of experiments: (i) varying the retrieval depth by setting the number of retrieved documents $K \in \{5, 10, 15, 20\}$, and (ii) comparing a dense retriever against a sparse retriever (SPLADE).

The full results examining the effect of different retrieval depths (number of retrieved documents) are presented in Table 9.

1512 Table 9: Effect of retrieval depth on performance across conversation lengths (100K–10M)
 1513 and memory abilities. Results are shown for different numbers of retrieved documents ($K \in$
 1514 $\{5, 10, 15, 20\}$).

Length	Memory Ability	K=5	K=10	K=15	K=20
100K	Abstention	0.475	0.500	0.625	0.625
	Contradiction Resolution	0.037	0.025	0.025	0.031
	Event Ordering	0.205	0.191	0.218	0.210
	Information Extraction	0.479	0.450	0.412	0.391
	Instruction Following	0.362	0.362	0.475	0.462
	Knowledge Update	0.362	0.375	0.350	0.300
	Multi-Hop Reasoning	0.281	0.322	0.321	0.309
	Preference Following	0.566	0.591	0.562	0.575
	Summarization	0.232	0.231	0.218	0.213
	Temporal Reasoning	0.112	0.162	0.137	0.137
500K	Average	0.311	0.321	0.334	0.325
	Abstention	0.571	0.514	0.614	0.642
	Contradiction Resolution	0.017	0.021	0.071	0.071
	Event Ordering	0.244	0.229	0.238	0.247
	Information Extraction	0.506	0.531	0.503	0.507
	Instruction Following	0.295	0.341	0.390	0.373
	Knowledge Update	0.278	0.307	0.326	0.326
	Multi-Hop Reasoning	0.214	0.188	0.234	0.213
	Preference Following	0.571	0.597	0.628	0.607
	Summarization	0.344	0.354	0.375	0.376
1M	Temporal Reasoning	0.121	0.128	0.121	0.135
	Average	0.316	0.321	0.350	0.350
	Abstention	0.500	0.521	0.600	0.585
	Contradiction Resolution	0.021	0.021	0.057	0.053
	Event Ordering	0.200	0.224	0.240	0.242
	Information Extraction	0.366	0.398	0.377	0.391
	Instruction Following	0.419	0.476	0.439	0.446
	Knowledge Update	0.357	0.350	0.400	0.407
	Multi-Hop Reasoning	0.209	0.189	0.209	0.190
	Preference Following	0.551	0.596	0.535	0.514
10M	Summarization	0.316	0.317	0.325	0.351
	Temporal Reasoning	0.154	0.154	0.119	0.199
	Average	0.309	0.325	0.330	0.330
	Abstention	0.550	0.600	0.650	0.600
	Contradiction Resolution	0.012	0.012	0.025	0.025
	Event Ordering	0.197	0.210	0.213	0.236
	Information Extraction	0.350	0.150	0.300	0.300
	Instruction Following	0.350	0.150	0.450	0.400
	Knowledge Update	0.275	0.200	0.300	0.300
	Multi-Hop Reasoning	0.125	0.100	0.125	0.150

1561 In a complementary experiment, we analyzed the impact of retriever choice. Our base architecture
 1562 employs a dense retriever, which we compare against the sparse SPLADE-V2 retriever (Formal
 1563 et al., 2022). As shown in Figure 5 in Appendix C.2, SPLADE yields performance gains of 1.7%
 1564 at 100K tokens, 0.7% at 500K, and 0.8% at 1M, but results in a slight performance drop of 0.7% at
 1565 10M. On average, the sparse retriever provides a modest improvement across conversation lengths.
 Complete results comparing the dense retriever with SPLADE are presented in Table 10.

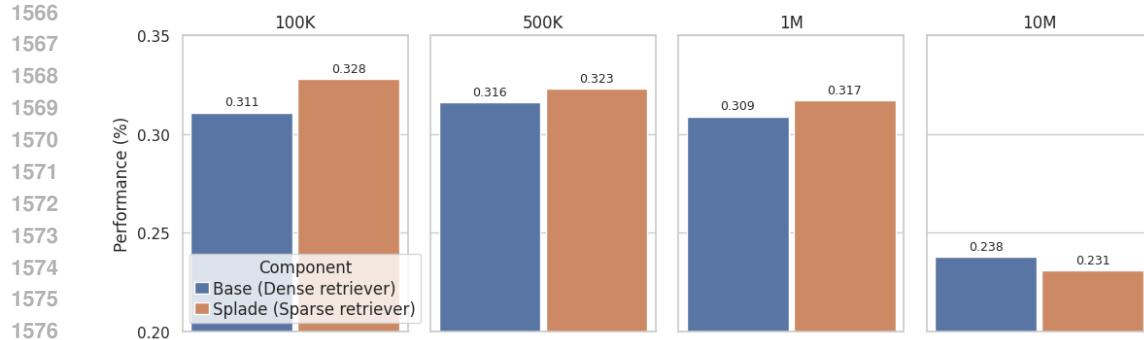


Figure 5: Performance comparison between dense retrieval and sparse retrieval (SPLADE) in LIGHT.

Table 10: Comparison of dense and sparse retrieval strategies across conversation lengths (100K–10M) and ten memory abilities. The table reports performance when using the default dense retriever versus a sparse retriever (SPLADE).

Length	Memory Ability	Base (Dense retriever)	Sparse retriever (SPLADE)
100K	Abstention	0.475	0.525
	Contradiction Resolution	0.037	0.43
	Event Ordering	0.205	0.181
	Information Extraction	0.479	0.596
	Instruction Following	0.362	0.400
	Knowledge Update	0.362	0.350
	Multi-Hop Reasoning	0.281	0.267
	Preference Following	0.566	0.562
	Summarization	0.232	0.230
	Temporal Reasoning	0.112	0.125
	Average	0.311	0.328
500K	Abstention	0.571	0.557
	Contradiction Resolution	0.017	0.025
	Event Ordering	0.244	0.226
	Information Extraction	0.506	0.559
	Instruction Following	0.295	0.345
	Knowledge Update	0.278	0.307
	Multi-Hop Reasoning	0.214	0.212
	Preference Following	0.571	0.565
	Summarization	0.344	0.330
	Temporal Reasoning	0.121	0.107
	Average	0.316	0.323
1M	Abstention	0.500	0.564
	Contradiction Resolution	0.021	0.028
	Event Ordering	0.200	0.196
	Information Extraction	0.366	0.392
	Instruction Following	0.419	0.401
	Knowledge Update	0.357	0.371
	Multi-Hop Reasoning	0.209	0.193
	Preference Following	0.551	0.595
	Summarization	0.316	0.300
	Temporal Reasoning	0.154	0.133
	Average	0.309	0.317
10M	Abstention	0.550	0.700
	Contradiction Resolution	0.012	0.000
	Event Ordering	0.197	0.202
	Information Extraction	0.350	0.350
	Instruction Following	0.350	0.250
	Knowledge Update	0.275	0.375
	Multi-Hop Reasoning	0.125	0.125
	Preference Following	0.308	0.200
	Summarization	300.220	0.090
	Temporal Reasoning	0.000	0.025
	Average	0.238	0.231

1620

C.3 EFFECT OF EMBEDDING MODEL CHOICE

1621

1622

1623

1624

We also examined how the choice of embedding model affects the performance of both the RAG baseline and the episodic memory component of LIGHT. In the primary experiments, we used the *BAAI/bge-small-en-v1.5* embedding model. To assess robustness, we re-ran all experiments using the larger *BAAI/bge-large-en-v1.5* model while keeping the LLM reader fixed to *GPT-4.1-nano*. As shown in Table 11, LIGHT consistently outperforms the RAG baseline under both embedding configurations. Moreover, LIGHT exhibits larger gains when switching to the higher-capacity embedding model, achieving an additional 2.08% improvement at the 1M-token setting and 16.37% at the 10M-token setting. These results indicate that LIGHT is robust to changes in embedding quality and can effectively leverage stronger embedding models to enhance long-term memory abilities.

1625

1626

1627

1628

1629

1630

1631

1632

1633

1634

1635

1636

Table 11: Effect of the embedding model on performance across conversation lengths (100K–10M) and ten memory abilities. Results are shown for the RAG baseline and LIGHT using two different embedding models.

Length	Memory Ability	RAG (bge-small)	Ours (bge-small)	RAG (bge-large)	Ours (bge-large)
100K	Abstention	0.800	0.575	0.825	0.600
	Contradiction Resolution	0.018	0.031	0.012	0.031
	Event Ordering	0.169	0.177	0.185	0.171
	Information Extraction	0.362	0.538	0.404	0.562
	Instruction Following	0.350	0.400	0.337	0.462
	Knowledge Update	0.375	0.375	0.325	0.375
	Multi-Hop Reasoning	0.263	0.365	0.224	0.341
	Preference Following	0.550	0.625	0.537	0.562
	Summarization	0.083	0.202	0.089	0.148
	Temporal Reasoning	0.125	0.162	0.112	0.162
	Average	0.309	0.345	0.305	0.341
500K	Abstention	0.828	0.600	0.814	0.571
	Contradiction Resolution	0.025	0.035	0.028	0.032
	Event Ordering	0.180	0.204	0.178	0.202
	Information Extraction	0.382	0.491	0.345	0.454
	Instruction Following	0.286	0.342	0.303	0.363
	Knowledge Update	0.288	0.240	0.380	0.321
	Multi-Hop Reasoning	0.233	0.266	0.272	0.282
	Preference Following	0.577	0.684	0.571	0.650
	Summarization	0.184	0.334	0.153	0.316
	Temporal Reasoning	0.161	0.154	0.126	0.126
	Average	0.314	0.335	0.317	0.331
1M	Abstention	0.778	0.678	0.771	0.657
	Contradiction Resolution	0.028	0.021	0.021	0.025
	Event Ordering	0.179	0.211	0.194	0.211
	Information Extraction	0.399	0.410	0.360	0.439
	Instruction Following	0.271	0.394	0.269	0.421
	Knowledge Update	0.342	0.392	0.371	0.378
	Multi-Hop Reasoning	0.293	0.278	0.204	0.254
	Preference Following	0.513	0.576	0.497	0.598
	Summarization	0.152	0.290	0.116	0.296
	Temporal Reasoning	0.064	0.107	0.119	0.150
	Average	0.302	0.336	0.292	0.343
10M	Abstention	0.650	0.400	0.800	0.550
	Contradiction Resolution	0.012	0.025	0.025	0.037
	Event Ordering	0.201	0.173	0.203	0.171
	Information Extraction	0.300	0.350	0.300	0.450
	Instruction Following	0.175	0.250	0.175	0.275
	Knowledge Update	0.325	0.300	0.325	0.300
	Multi-Hop Reasoning	0.091	0.135	0.066	0.075
	Preference Following	0.366	0.425	0.316	0.525
	Summarization	0.063	0.179	0.100	0.224
	Temporal Reasoning	0.000	0.025	0.000	0.025
	Average	0.218	0.226	0.231	0.263

1674 C.4 EFFECT OF INDEXING SETUP
1675
1676

1677 We also investigated the effect of the vector database indexing setup on the performance of LIGHT.
1678 In the primary experiments, we used *IndexFlatIP*, and in the experiments below, we examined the
1679 effect of switching the indexing setup to *IndexHNSWFlat*. The results are shown in Table 12.

1684 Table 12: Effect of vector database indexing setup on performance across conversation lengths
1685 (100K–10M) and ten memory abilities. Results are reported for LIGHT under two different indexing
1686 configurations.

Length	Memory Ability	Ours (IndexFlatIP)	Ours (IndexHNSWFlat)
100K	Abstention	0.575	0.600
	Contradiction Resolution	0.031	0.031
	Event Ordering	0.177	0.173
	Information Extraction	0.538	0.565
	Instruction Following	0.400	0.375
	Knowledge Update	0.375	0.400
	Multi-Hop Reasoning	0.365	0.285
	Preference Following	0.625	0.662
	Summarization	0.202	0.217
	Temporal Reasoning	0.162	0.162
500K	Average	0.345	0.347
	Abstention	0.600	0.528
	Contradiction Resolution	0.035	0.032
	Event Ordering	0.204	0.207
	Information Extraction	0.491	0.503
	Instruction Following	0.342	0.332
	Knowledge Update	0.240	0.226
	Multi-Hop Reasoning	0.266	0.269
	Preference Following	0.684	0.666
	Summarization	0.334	0.317
1M	Temporal Reasoning	0.154	0.176
	Average	0.335	0.325
	Abstention	0.678	0.578
	Contradiction Resolution	0.021	0.025
	Event Ordering	0.211	0.211
	Information Extraction	0.410	0.420
	Instruction Following	0.394	0.386
	Knowledge Update	0.392	0.385
	Multi-Hop Reasoning	0.278	0.278
	Preference Following	0.576	0.567
10M	Summarization	0.290	0.257
	Temporal Reasoning	0.107	0.128
	Average	0.336	0.324
	Abstention	0.400	0.600
	Contradiction Resolution	0.025	0.025
	Event Ordering	0.173	0.168
	Information Extraction	0.350	0.350
	Instruction Following	0.250	0.300
	Knowledge Update	0.300	0.225
	Multi-Hop Reasoning	0.135	0.075
Average	Preference Following	0.425	0.433
	Summarization	0.179	0.194
10M	Temporal Reasoning	0.025	0.000
	Average	0.226	0.237

1728 C.5 SUPPLEMENTARY BASELINE EVALUATION
1729

1730 Alongside long-context LLMs and RAG, we also evaluated ReadAgent (Lee et al., 2024), another
1731 method designed to enhance long-term memory in LLMs, on BEAM and compared it with LIGHT.
1732 The results demonstrate that LIGHT consistently outperforms ReadAgent across all four conversa-
1733 tion lengths (100K, 500K, 1M, and 10M). The full results are shown in Table 13.

1734
1735 Table 13: Comparing LIGHT with ReadAgent across conversation lengths (100K–10M) and ten
1736 memory abilities.

Length	Memory Ability	ReadAgent	Ours (LIGHT)
100K	Abstention	0.850	0.475
	Contradiction Resolution	0.000	0.037
	Event Ordering	0.200	0.205
	Information Extraction	0.066	0.479
	Instruction Following	0.237	0.362
	Knowledge Update	0.150	0.362
	Multi-Hop Reasoning	0.095	0.281
	Preference Following	0.425	0.566
	Summarization	0.045	0.232
	Temporal Reasoning	0.000	0.112
500K	Average	0.206	0.311
	Abstention	0.928	0.571
	Contradiction Resolution	0.007	0.017
	Event Ordering	0.237	0.244
	Information Extraction	0.047	0.506
	Instruction Following	0.166	0.295
	Knowledge Update	0.014	0.278
	Multi-Hop Reasoning	0.022	0.214
	Preference Following	0.386	0.571
	Summarization	0.069	0.344
1M	Temporal Reasoning	0.028	0.121
	Average	0.191	0.316
	Abstention	0.792	0.500
	Contradiction Resolution	0.003	0.021
	Event Ordering	0.211	0.200
	Information Extraction	0.106	0.366
	Instruction Following	0.166	0.419
	Knowledge Update	0.014	0.357
	Multi-Hop Reasoning	0.105	0.209
	Preference Following	0.391	0.551
10M	Summarization	0.041	0.316
	Temporal Reasoning	0.033	0.154
	Average	0.186	0.309
	Abstention	0.750	0.550
	Contradiction Resolution	0.000	0.012
	Event Ordering	0.205	0.197
	Information Extraction	0.000	0.350
	Instruction Following	0.300	0.350
	Knowledge Update	0.000	0.275
	Multi-Hop Reasoning	0.000	0.125
1728	Preference Following	0.166	0.308
	Summarization	0.061	0.220
	Temporal Reasoning	0.000	0.000
	Average	0.148	0.238

1779
1780
1781

1782 **D NUGGET DESIGN**
17831784 In this section, we provide illustrative examples for each memory ability, demonstrating how nuggets
1785 are derived from the corresponding probing questions.
17861787 **1. Abstention**
17881789 **Objective:** The correct behavior is to acknowledge that the requested information is not
1790 present in the provided conversation.
17911792 **Rubric pattern:** Each atomic unit should be in this format: *States that, based on the*
1793 *provided chat, there is no information about <target topic>*
17941795 **Example JSON:**
17961797

```
{  
  "question": "What specific advice did Manuel give about property management  
  companies during the March 5 Investors Meetup?",  
  "ideal_response": "Based on the provided chat, there is no information related  
  to the specific advice Manuel gave about property management companies.",  
  "source_chat_ids": {},  
  "rubric": [  
    "Based on the provided chat, there is no information related to the specific  
    advice Manuel gave about property management companies."  
  ]  
}
```


18001801 **2. Contradiction Resolution**
18021803 **Objective:** Correct behavior is that the LLM should detect the contradiction and state
1804 both contradictory information while requesting clarification.
18051806 **Rubric pattern:**
18071808

- States there is contradictory information.
- Mentions claim <A>
- Mentions claim
- Requests clarification about which statement is correct

18091810 **Example JSON:**
18111812

```
{  
  "question": "Have I ever attended any real estate webinars or investor meetups?",  
  "ideal_answer": "I notice you've mentioned contradictory information about this.  
  You said you have never attended any real estate webinars or investor  
  meetups, but you also mentioned attending a webinar about Turkey's rising  
  demand for multi-family rentals. Which statement is correct?",  
  "source_chat_ids": {  
    "first_statement": [  
    ],  
    "second_statement": [  
    ]  
  },  
  "rubric": [  
    "LLM response should state: there is contradictory information",  
    "LLM response should mention: You said you have never attended any real  
    estate webinars or investor meetups",  
    "LLM response should mention: you also mentioned attending a webinar about  
    Turkey\u2019s rising demand for multi-family rentals",  
    "LLM response should mention: which statement is correct?"  
  ]  
}
```


18201821 **3. Event Ordering**
18221823 **Objective:** Correct behavior is the model lists a sequence of events/topics in the correct
1824 chronological order.
18251826 **Rubric pattern:**
18271828

- LLM response should mention: <event 1>
- ...

1836 • LLM response should mention: <event N>
 1837
 1838 **Example JSON:**
 1839 {
 1840 "question": "How did my focus on different aspects of property investment and
 1841 management develop throughout our conversations in order? Mention ONLY and
 1842 ONLY ten items.",
 1843 "answer": "Your focus on property investment and management developed in this
 1844 sequence: 1) Initial engagement with the local agent and preparation for
 1845 property viewings, 2) Evaluation of property financials including ROI and
 1846 rental income potential, 3) Exploration of financing options and mortgage
 1847 concerns, 4) Handling contractor performance and repair negotiations, 5)
 1848 Understanding zoning regulations and permit requirements for property
 1849 conversations, 6) Planning and prioritizing renovations and investment risks
 1850 for multi-family properties, 7) Detailed preparation for property viewings
 1851 involving both agent and contractor, 8) Negotiation strategies and cash
 1852 flow implications related to repair costs, 9) Final renovation project
 1853 completion steps and portfolio diversification strategies, 10)
 1854 Consideration of commercial property types and location factors for long-
 1855 term investment.",
 1856 "ordering_tested": [
 1857 "1st: Agent interaction and viewing preparation",
 1858 "2nd: Property financial evaluation",
 1859 "3rd: Financing and mortgage concerns",
 1860 "4th: Contractor management",
 1861 "5th: Zoning and permits",
 1862 "6th: Renovation planning and investment risks",
 1863 "7th: Viewing preparation with agent and contractor",
 1864 "8th: Repair cost negotiation and cash flow",
 1865 "9th: Renovation completion and portfolio diversification",
 1866 "10th: Commercial property and location considerations"
 1867],
 1868 "source_chat_ids": [],
 1869 "rubric": [
 1870 "LLM response should mention: Agent interaction and viewing preparation",
 1871 "LLM response should mention: Property financial evaluation",
 1872 "LLM response should mention: Financing and mortgage concerns",
 1873 "LLM response should mention: Contractor management",
 1874 "LLM response should mention: Zoning and permits",
 1875 "LLM response should mention: Renovation planning and investment risks",
 1876 "LLM response should mention: Viewing preparation with agent and contractor"
 1877 ",
 1878 "LLM response should mention: Repair cost negotiation and cash flow",
 1879 "LLM response should mention: Renovation completion and portfolio
 1880 diversification",
 1881 "LLM response should mention: Commercial property and location
 1882 considerations",
 1883 "Presents the events in the correct chronological order"
 1884]
 1885 }
 1886 }
 1887 }

4. Information Extraction

1875 **Objective:** LLM should answer the questioned facts correctly.
 1876 **Rubric pattern:**
 1877 • Instantiate one criterion per fact directly from the ideal answer, using the stem
 1878 "LLM response should state/mention:"
 1879 **Example JSON:**
 1880 {
 1881 "question": "What amount did I say I had set aside as my initial capital for
 1882 investing?",
 1883 "ideal_answer": "You said you had set aside \$50,000 as your initial capital.",
 1884 "source_chat_ids": [],
 1885 "rubric": [
 1886 "LLM response should state: \$50,000"
 1887]
 1888 }

5. Instruction Following

1890

1891

1892

1893

1894

1895

1896

1897

1898

1899

1900

1901

1902

1903

1904

1905

1906

1907

1908

1909

1910

1911

1912

1913

1914

1915

1916

1917

1918

1919

1920

1921

1922

1923

1924

1925

1926

1927

1928

1929

1930

1931

1932

1933

1934

1935

1936

1937

1938

1939

1940

1941

1942

1943

Objective: LLM should adhere to format and/or content priorities stated in the conversation.

Rubric pattern:

- Use `instruction_being_tested` (the explicit instruction) and decompose `expected_compliance` into atomic criteria

Example JSON:

```
{
  "question": "How should I allocate my funds for the upcoming quarter?",
  "instruction_being_tested": "Always provide detailed financial breakdowns when I ask about budgeting decisions.",
  "expected_compliance": "Response should include itemized costs, specific amounts for different categories, and detailed breakdown rather than just a total estimate",
  "source_chat_ids": [],
  "rubric": [
    "LLM response should contain: itemized list of costs",
    "LLM response should contain: category-by-category breakdown",
    "LLM response should contain: detailed cost analysis"
  ]
}
```

6. Knowledge Update

Objective: LLM must reflect updated values when prior values have changed over time.

Rubric pattern:

- Derive criteria from the ideal answer, using the stem “LLM response should state/mention:” for the updated value(s).

Example JSON:

```
{
  "question": "How much initial capital have I allocated for real estate investing ?",
  "answer": "$60,000",
  "source_chat_ids": {
    "original_info": [
      ],
      "updated_info": [
        ]
  },
  "rubric": [
    "LLM response should state: $60,000"
  ]
}
```

7. Multi-hop Reasoning

Objective: LLM must aggregate or compare information spanning multiple sessions.

Rubric pattern:

- Instantiate criteria from the ideal answer for each required intermediate or aggregated fact.

Example JSON:

```
{
  "question": "How many different banks did I consider for my mortgage options across my conversations?",
  "answer": "Two banks: Halkbank and Ziraat Bank.",
  "source_chat_ids": [],
  "rubric": [
    "LLM response should state: Two banks",
    "LLM response should state: Halkbank",
    "LLM response should state: Ziraat Bank"
  ]
}
```

8. Preference Following

1944

1945

1946

1947

1948

1949

1950

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

1962

1963

1964

1965

1966

1967

1968

1969

1970

1971

1972

1973

1974

1975

1976

1977

1978

1979

1980

1981

1982

1983

1984

1985

1986

1987

1988

1989

1990

1991

1992

1993

1994

1995

1996

1997

Objective: LLM must generate content consistent with user-specified preferences.

Rubric pattern:

- Use `preference_being_tested` (the user's stated preference) and decompose `expected_compliance` into atomic criteria.

Example JSON:

```
{
  "question": "I'm looking at several properties and want to understand how to approach comparing them effectively. What steps would you suggest I take?",  

  "preference_being_tested": "I prefer methodical, data-driven decisions over impulsive purchases, reflecting my analytical nature.",  

  "expected_compliance": "Response should outline a structured, step-by-step approach that involves gathering and analyzing relevant data before making a choice, rather than suggesting quick or impulsive actions.",  

  "source_chat_ids": [],  

  "rubric": [
    "LLM response should contain: provides a clear, logical process",
    "LLM response should contain: emphasizes data collection and analysis",
    "LLM response should contain: breaks down comparison into measurable factors"
  ]
}
```

9. Summarization

Objective: LLM must provide a comprehensive summary covering required content elements.

Rubric pattern:

- Decompose `ideal_summary` into atomic content units; use the stem "LLM response should contain".

Example JSON:

```
{
  "question": "Can you give me a comprehensive summary of my journey and decision-making process around investing in rental properties, including how my budget, property choices, management considerations, and financing plans have developed over time?",  

  "ideal_summary": "Your journey toward investing in rental properties began with an initial capital of $50,000, which you questioned as potentially insufficient for purchasing a property within 12 months. Early discussions highlighted the need to research local market conditions, down payment requirements, and additional costs like closing fees and renovations, revealing that typical investments might exceed your initial capital. You explored identifying good fixer-upper properties by learning to recognize signs such as structural issues and outdated features, emphasizing the importance of cost-benefit analysis for renovations. As your plans progressed, you weighed the pros and cons of investing close to your location versus elsewhere, balancing ease of management against market diversity and growth potential. You also considered the choice between single-family homes and multi-family units, analyzing factors like rental yield, management complexity, and investment scale, with examples showing similar yields but differing capital needs. Financing options were carefully compared, particularly between Halkbank and Ziraat Bank mortgages, focusing on interest rates, fees, and service quality to optimize costs. Throughout, you developed a step-by-step plan for purchasing your first rental property, including market research, budgeting, inspections, financing, and tenant management, with timelines to reduce anxiety and ensure readiness. This comprehensive process reflects a thoughtful evolution from initial capital concerns to detailed investment strategies, property evaluation, financing decisions, and management planning, all aimed at making informed, balanced real estate investment choices.",  

  "source_chat_ids": [],  

  "rubric": [
    "LLM response should contain: investing in rental properties began with an initial capital of $50,000",
    "LLM response should contain: Early discussions highlighted the need to research local market conditions, down payment requirements, and additional costs like closing fees",
    "LLM response should contain: You explored identifying good fixer-upper properties by learning to recognize signs such as structural issues and outdated features"
  ]
}
```

```

1998
1999 "LLM response should contain: you weighed the pros and cons of investing
2000   close to your location versus elsewhere, balancing ease of management
2001   against market diversity and growth potential",
2002 "LLM response should contain: You also considered the choice between single-
2003   family homes and multi-family units, analyzing factors like rental
2004   yield, management complexity, and investment scale",
2005 "LLM response should contain: Financing options were carefully compared,
2006   particularly between Halkbank and Ziraat Bank mortgages, focusing on
2007   interest rates, fees, and service quality to optimize costs",
2008 "LLM response should contain: you developed a step-by-step plan for
2009   purchasing your first rental property, including market research,
2010   budgeting, inspections, financing, and tenant management"
2011   ]
2012   }
2013

```

10. Temporal Reasoning

Objective: LLM must compute or restate durations and timeline relations correctly.

Rubric pattern:

- Derive criteria from the ideal answer, using the stem “LLM response should state:”.

Example JSON:

```

2014 {
2015   "question": "How many days are there between my first property viewing with
2016   Mehmet Yilmaz and the last one I scheduled?",
2017   "answer": "There are 2 days between the first property viewing on March 25 and
2018   the last one on March 27.",
2019   "calculation_required": "March 27 - March 25 = 2 days",
2020   "source_chat_ids": {
2021     "first_event": [],
2022     "second_event": []
2023   },
2024   "rubric": [
2025     "LLM response should state: 2 days",
2026     "LLM response should state: from March 25, 2024 till March 27, 2024"
2027   ]
2028 }
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051

```

2052 **E EXAMPLES FROM DIFFERENT COMPONENTS OF BEAM**

2053

2054 In this section, we provide illustrative examples of generating a chat in the *coding* domain. Speci-
 2055 cally, we include a representative *chat seed* with its domain, title, theme, and subtopics, followed
 2056 by the corresponding *narratives*, where only a truncated set is shown for brevity. We then present
 2057 the *user profile* and the user’s social *relationships*. Next, we provide excerpts from the *conversation*
 2058 *plans*, showing only a subset of bullet points from each sub-plan while preserving their full descrip-
 2059 tions to maintain clarity. Finally, we provide samples of the *generated chat*, highlighting exchanges
 2060 where the user shares or requests code, and including follow-up turns to demonstrate the natural-
 2061 istic back-and-forth flow. Together, these examples illustrate how different components of BEAM
 2062 interact to form coherent, long-context dialogues.

2063 Chat Seed

2064

2065 **Domain:** Coding

2066 **Title:** Automating Social Media Posts with Python

2067 **Theme:** Scheduling and posting content across multiple platforms

2068 **Subtopics:**

- 2069 • Twitter API integration • Facebook Graph API usage • Instagram automation tools
- 2070 • Scheduling with cron jobs / APScheduler
- 2071 • Image and caption management; hashtag generation
- 2072 • Error handling for failed posts; tracking engagement metrics

2073 Narratives (Truncated)

2074

2075 **Technical Problem-Solving:** Debugging Twitter OAuth/403/429; fixing hashtag validation;
 2076 profiling scheduler bottlenecks.

2077 **Learning & Knowledge:** API docs comprehension (Twitter v2, Facebook Graph v12–15);
 2078 best practices for Instagram automation; mastering cron/APScheduler.

2079 **Progress & Development:** Setting up Twitter/Facebook integrations; building Instagram
 2080 tools; designing scheduling algorithms.

2081 **Implementation:** Feature implementation and refactoring for efficiency; async migration;
 2082 retry and backoff strategies.

2083 **Framework & Technology:** Python libraries (Tweepy, facebook-sdk, requests); APSched-
 2084 uler/cron; Redis; asyncio.

2085 **Testing & QA:** Unit/integration/E2E tests (pytest, Selenium); TDD for schedulers and hash-
 2086 tag rules.

2087 **DevOps & Deployment:** CI/CD (GitHub Actions), containerization (Docker), EC2 deploy-
 2088 ment, blue–green releases.

2089 **Data:** PostgreSQL schemas, indices, ETL for engagement metrics, Redis caching.

2090 **Integration & APIs:** Webhooks, message queues (RabbitMQ), API Gateway, SNS/Lambda.

2091 **Performance:** Caching, load balancing (HAProxy), CPU/memory targets, throughput goals.

2092 **Security/Compliance:** OAuth, token rotation, TLS, GDPR.

2093 **PM & Workflow:** Sprints, reviews, documentation standards.

2094 User Profile

2095

2096 **Name:** John Brooks **Age:** 52 **Gender:** Male

2097 **Location:** Port Charles, Luxembourg **Profession:** Secretary/Administrator

2098 **Personality:** He is a pillar of his community, always ready to lend a helping hand and offer
 2099 guidance when needed. With a strong sense of tradition and order, he values honesty and
 2100 dedication, often taking on a mentorship role to help others. His diligent and efficient ap-
 2101 proach to planning and organization makes him a reliable asset to those around him. He has a
 2102 warm and welcoming demeanor, always willing to open his heart and home to friends, loved
 2103 ones, and neighbors. Despite his strong convictions, he believes in the power of hospitality
 2104 and good manners, often going out of his way to make others feel supported and cared for.

2106
2107
2108
2109
2110

With a dry sense of humor and a quick wit, he can be entertaining to be around, but he's not afraid to speak his mind and challenge the status quo when necessary. His practical and responsible nature makes him a respected member of his community, and his ability to stay grounded and logical in stressful situations is a valuable asset to those around him.

2111
2112
2113
2114
2115
2116
2117

Relationships

Parents: Elizabeth (74), Robert (76)

Partner: Shannon (48)

Close Friends: Taylor (51), Teresa (62), Thomas (44), Charles (56), Patricia (46)

Acquaintances/Colleagues: Wesley (26), Jason (59), Claudia (15), Janice (13), Dana (55)

2118
2119

Conversation Plan (Only a few representative bullets from each sub-plan)

2120
2121
2122
2123
2124
2125
2126

Subplan 1 — March 1, 2024

- **Project Initialization:** I'm setting up a Python 3.10 environment with Tweepy v4.10.1 and Facebook SDK v3.1.0 for API integrations.
- **Security & Compliance Labels: Authentication for Twitter API Integration:** Implemented OAuth 1.0a with environment variables TWITTER_API_KEY and TWITTER_API_SECRET securely stored.
- **Database & Data Management Labels: Database Design for Social Media Posting:** Designed PostgreSQL 14 schema with tables for posts, platforms, and scheduling metadata.
- **User Instruction:** Always include exact API version numbers when I ask about integration details.
- **Logical Contradiction:** I have never registered a Twitter Developer account or created any Twitter app.

2130
2131
2132
2133

Subplan 2 — March 20, 2024

- **Technical Problem-Solving Labels: Debugging Twitter API Integration:** Fixed “403 Forbidden” error caused by missing media upload step before tweet creation.
- **Implementation & Development Labels: Code Refactoring for Performance:** Refactored `twitter_post.py` to `async` functions using `asyncio`, improved throughput by 30%.
- **Security & Compliance Labels: Authorization for Facebook Graph API:** Implemented OAuth 2.0 flow with refresh tokens stored encrypted using Fernet symmetric encryption.
- **Information Update:** The Instagram automation prototype sprint deadline was adjusted to April 5, 2024, to allow additional testing of media upload features.

2140
2141
2142
2143
2144

Subplan 3 — April 5, 2024

- **Implementation & Development Labels: Implementing Error Handling:** Added retry logic with exponential backoff for Instagram API 429 Too Many Requests errors.
- **Performance & Optimization Labels: Caching Strategies for Image and Caption Management:** Implemented Redis caching for resized images, reducing image processing time from 800ms to 200ms.
- **Debugging & Troubleshooting Labels: Incident Response for Social Media Automation:** Responded to March 30, 2024, outage caused by expired Instagram tokens, implemented alerting via Slack webhook.

2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159

Subplan 4 — April 20, 2024

- **Implementation & Development Labels: Algorithm Optimization for Scheduling:** Rewrote scheduling algorithm to use `async` priority queues, reducing average job dispatch latency from 500ms to 150ms.

2160
2161
2162
2163
2164
2165

- **Framework & Technology Labels: Integrating Twitter API with Python:** Upgraded Tweepy from v4.10.1 to v4.12.1 to leverage new media upload endpoints.
- **Security & Compliance Labels: Authentication for Twitter API Integration:** Rotated Twitter API keys on April 15, 2024, updated environment variables `TWITTER_API_KEY` and `TWITTER_API_SECRET`.

2166 Subplan 5 — May 5, 2024

- **Progress & Development Labels: Building Hashtag Generation Tools:** Developed hashtag generator supporting dynamic keyword extraction using spaCy v3.5.0 NLP library.
- **Database & Data Management Labels: Data Warehousing for Engagement Metrics:** Designed PostgreSQL 14 schema for engagement_metrics with partitioning by month for scalability.
- **Debugging & Troubleshooting Labels: Log Analysis for Facebook Graph API:** Detected “OAuthException: Error validating access token” on May 1, 2024, resolved by token refresh automation.

2176 Subplan 6 — May 20, 2024

- **Implementation & Development Labels: Implementing Error Handling:** Added centralized error handler middleware in posting API, logging errors with Sentry v1.12.0.
- **Debugging & Troubleshooting Labels: Error Diagnosis for Twitter API Integration:** Fixed intermittent “ConnectionResetError” during media upload by adding retry with jitter.
- **DevOps & Deployment Labels: Containerization for Instagram Automation:** Updated Dockerfile to use multi-stage builds, reduced image size from 120MB to 85MB.

2185 Subplan 7 — June 5, 2024

- **DevOps & Deployment Labels: Deploying Social Media Automation Tools:** Deployed v1.0.0 release on AWS EC2 t3.medium with 99.9% uptime SLA.
- **Integration & API Labels: Event-Driven Architecture for Social Media Automation:** Implemented AWS SNS topics for post status updates, integrated with Lambda v3.2.1 functions.
- **User Experience & Interface Labels: Mobile App Design for Social Media Automation:** Released beta version of React Native app on Android with basic scheduling and metrics display.

2195 Subplan 8 — June 20, 2024

- **Progress & Development Labels: Developing Instagram Automation Tools:** Implemented batch media uploads for Instagram, supporting up to 10 images per carousel post.
- **User Experience & Interface Labels: Responsive Design for Scheduling:** Enhanced React 18.2 dashboard for scheduling with drag-and-drop post reordering, tested on Chrome and Safari.
- **Security & Compliance Labels: Encryption for Instagram Automation:** Migrated token storage to AWS KMS encrypted S3 bucket, access controlled via IAM roles.

2204 Subplan 9 — July 5, 2024

- **Security & Compliance Labels: Authentication for Twitter API Integration:** Implemented OAuth 2.0 PKCE flow for enhanced security, tested with Postman v10.15.0.
- **Performance & Optimization Labels: Code Optimization for Twitter API Integration:** Reduced memory usage by 20% by optimizing data serialization and connection pooling.
- **Debugging & Troubleshooting Labels: Error Diagnosis for Twitter API Integration:** Fixed “TimeoutError” during high load by increasing HTTP client timeout from 10s to 30s.

2214

2215

2216

2217

2218

2219

2220

2221

2222

2223

2224

2225

2226

Subplan 10 — July 20, 2024

- **DevOps & Deployment Labels: Deploying Social Media Automation Tools:** Completed production rollout on July 18, 2024, with zero downtime and 99.95% uptime in first 48 hours.
- **Integration & API Labels: Event-Driven Architecture for Social Media Automation:** Fully implemented AWS SNS event bus with Lambda v3.2.1 for asynchronous post status updates.
- **Testing & Quality Assurance Labels: End-to-End Testing for Instagram Automation:** Verified Instagram automation stability with 1000+ posts over 7 days, no failures reported.

Generated Dialogues (Exemplars, Truncated)

Turn 1 (Mar 1, 2024)

User: I'm trying to set up a Python 3.10 environment for automating social media posts, and I've chosen Tweepy v4.10.1 and Facebook SDK v3.1.0 for API integrations, but I'm not sure how to structure my project for better outreach. Can you help me plan this out in terms of scheduling and deadlines?

Assistant:

Certainly! Setting up an automated posting system with Tweepy and Facebook SDK is a great start.

Project Structure:

1. Project Setup (Mar 1–7) [Truncated ...]
2. Content Creation (Mar 8–14) [Truncated ...]
3. Automation Development (Mar 15–21) [Truncated ...]
4. Deployment and Monitoring (Mar 22–28) [Truncated ...]

User (Follow-up)

That sounds like a great plan! One thing though—could we add a step for testing posts on a smaller scale before full deployment? Just to make sure everything looks good and functions as expected.

Assistant

Absolutely, adding a step for testing the posts on a smaller scale is a great idea to ensure everything looks good and functions as expected before full deployment. Here's how you can incorporate that into the existing plan:

Updated Project Structure

1. **Project Setup [Truncated ...]
4. **Testing Phase (March 22 - March 24, 2024)**- **Days 22-23:** Conduct a small-scale test.- Post a few test messages to a limited audience or a test account.- Verify that the posts look good and function as expected.- **Day 24:** Review the test results.- Address any issues found during the test phase.- Make final adjustments to the automation script and content.[Truncated ...]

User (Mid-turn) -> Code Sharing

I'm trying to implement a feature to collect engagement metrics for Instagram posts, but I'm not sure how to integrate it with the existing PostgreSQL schema. I've designed a table for `instagram_posts` with foreign key to `posts` table, but I'm not sure how to handle the data migration. Can you help me with that? Here's my code:

```
import psycopg2
conn = psycopg2.connect(dbname="engagement_metrics", user="username",
password="password",
host="localhost",
port="5432")
cur = conn.cursor()
cur.execute("""
CREATE TABLE instagram_posts (id SERIAL PRIMARY KEY, post_id INTEGER REFERENCES
posts(id), engagement_metrics JSONB);
```

```

2268
2269     \\"\\\")\\n\\nconn.commit()
2270     cur.close()
2271     conn.close()
2272
Assistant -> Code Review
2273 Your approach is good! Consider adding NOT NULL, timestamps, and indexes for performance ...
2274
2275     CREATE TABLE IF NOT EXISTS instagram_posts (
2276         id SERIAL PRIMARY KEY,
2277         post_id INTEGER NOT NULL REFERENCES posts(id),
2278         engagement_metrics JSONB,
2279         created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
2280     );
2281     CREATE INDEX IF NOT EXISTS idx_instagram_posts_post_id
2282         ON instagram_posts(post_id);
2283
2284     [Truncated ...] plus safe migration with INSERT SELECT to backfill existing posts.
2285
User -> Deployment
2286 I'm trying to deploy my updated posting services to an AWS EC2 t3.medium instance with 2
2287 vCPUs and 4GB RAM, but I'm running into some issues with the deployment process. Can
2288 you help me figure out how to properly deploy my application to this instance, considering
2289 the DevOps and deployment aspects? I've been using Docker to containerize my application,
2290 and I'm not sure what changes I need to make to ensure a smooth deployment. Here's a
2291 snippet of my current Dockerfile:
2292
2293     FROM python:3.10-slim
2294     WORKDIR /app
2295     COPY requirements.txt .
2296     RUN pip install -r requirements.txt
2297     COPY . .
2298     CMD ["python", "app.py"]
2299
Assistant -> Deployment Guidance
2300 Install Docker on EC2, push image to registry, pull with pinned tag, and run with
2301 --restart=always [Truncated ...] add healthchecks and a minimal systemd unit
2302 [Truncated ...]
2303
User (Follow-up)
2304 hmm, what's the best way to automate the pulling and running of the Docker container?
2305
Assistant -> CI/CD Automation
2306 Use GitHub Actions with SSH deploy step [Truncated ...]
2307
2308     name: Deploy
2309     on: { push: { branches: [main] } }
2310     jobs:
2311         deploy:
2312             runs-on: ubuntu-latest
2313             steps:
2314                 - uses: actions/checkout@v4
2315                 - run: |
2316                     ssh -i id_rsa ec2-user@EC2_DNS \
2317                     "docker stop socauto || true && docker rm socauto || true &&
2318                     docker pull username/your-app:sha-$(GITHUB_SHA) &&
2319                     docker run -d --restart=always -p 80:80 username/your-app:sha-$(GITHUB_SHA)"
2320
User (Follow-up)
2321 hmm, what's the best way to handle environment variables for different stages like dev and
2322 prod?
2323
Assistant -> Env Vars
2324 Use separate .env.dev / .env.prod files locally. In production, prefer AWS SSM Parameter
2325 Store or Secrets Manager [Truncated ...] inject via --env-file or systemd
2326 environment exports.

```

2322 **F CASE STUDY**

2323

2324 The scratchpad serves as a persistent, iteratively compressed semantic layer that (i) consolidates
 2325 scattered entities and facts (improving *Information Extraction*), (ii) preserves user-level meta-
 2326 instructions alongside task context (improving *Instruction Following*), (iii) records state changes and
 2327 overrides (improving *Information Update*), and (iv) normalizes timeline cues into explicit anchors
 2328 (improving *Temporal Reasoning*). Out of the ten memory abilities in our benchmark, we present
 2329 case studies from these four; however, the scratchpad provides similar benefits for the remaining six
 2330 abilities as well. Below we contrast “LIGHT with all components” versus “LIGHT w/o scratchpad”
 2331 on representative probes, and show the exact scratchpad cue that enabled the correct answer.

2332 **Information Extraction**

2333

2334 **(1) Question:** Which versions of the tools did I say I was using to start my project?

2335 **Correct Answer:** Python 3.10, spaCy v3.5, Flask 2.2, PyMuPDF 1.22.0

2336 **LIGHT with all components:** Python 3.10, spaCy v3.5, Flask 2.2, PyMuPDF 1.22.0

2337 **LIGHT w/o scratchpad:** Based solely on the provided context, there is no direct answer to this question

2338 **Scratchpad cue (excerpt):**

2339 ****KEY ENTITIES & RELATIONSHIPS:****

2340 [truncated]

2341 - Technologies: Python 3.10, spaCy v3.5, Flask 2.2/2.2.3, PyMuPDF 1.22.0, scikit-learn v1.2.2, Redis,
 RabbitMQ, Docker

2342 [truncated]

2343 **(2) Question:** Can you remind me of the city and country I said I live in, along with my age?

2344 **Correct Answer:** New Gary, Saint Helena; 26

2345 **LIGHT with all components:** New Gary, Saint Helena; 26

2346 **LIGHT w/o scratchpad:** No information provided about your city, country, or age in the given context

2347 **Scratchpad cue (excerpt):**

2348 ****KEY ENTITIES & RELATIONSHIPS:****

2349 - ****Primary individuals:**** User and partner Wendy (both turning 26 on March 3, 2024)

2350 - ****Friends and collaborators:**** Joe (writer, 35), Stacy [truncated]

2351 - ****Locations:**** New Gary, Saint Helena; Island Art Gallery; Saint Helena Book Fair; Saint Helena
 Community Center; GreenLeaf Nursery; Island Finds thrift store; Saint Helena Charity; The Light-
 house Bistro

2353 **Instruction Following**

2354

2355 **Question:** Can you tell me about the mentoring activities I've been involved in recently?

2356 **User instruction:** Always provide the number of team members involved when I ask about mentoring
 2357 activities.

2358 **Correct Answer:** Mentions number of team members

2359 **LIGHT with all components:** Mentoring 3 junior editors

2360 **LIGHT w/o scratchpad:** No recent mentoring activities noted in the provided context

2361 **Scratchpad cue (excerpt):**

2362 [truncated]

2363 The user requested to always include the number of team members when discussing mentoring activities.

2364 [truncated]

2365 ***Mentoring & Crisis Management:***

2366 - Mentoring 3 junior editors; balancing mentoring with crisis management.

2367 [truncated]

2368 **Knowledge Update**

2369

2370 **(1) Question:** What is the typical memory usage per session for batch resume processing?

2371 **Correct Answer:** Around 140MB per session

2372 **LIGHT with all components:** ~140MB

2373 **LIGHT w/o scratchpad:** The typical memory usage per session for batch resume processing can vary
 2374 widely depending on the complexity of the resumes and the system’s available memory, but optimizing
 2375 with batch sizes and efficient data handling can keep it under 100MB per resume.

2376 **Scratchpad cue (excerpt):**

2376 [truncated]
 2377 - Memory optimization:
 2378 - Reduce peak memory usage from 250MB to 140MB per batch.
 2379 [truncated]
 2380

2381 (2) **Question:** What is the deadline for completing the Instagram automation prototype sprint?
 2382 **Correct Answer:** April 5, 2024

2383 **LIGHT with all components:** April 5, 2024

2384 **LIGHT wo scratchpad:** April 1, 2024

2385 **Scratchpad cue (excerpt):**

2386 ****PROCESSES & WORKFLOWS:****

2387 1. ****Project Planning & Sprint Management:****

2388 [truncated]

2389 - Instagram automation sprint extended to April 5, 2024.

Temporal Reasoning

2392 (1) **Question:** How many days passed between when I met my mom at home and when I completed
 2393 my first successful batch of Miche's sourdough?

2394 **Correct Answer:** Jan 6 – Jan 1 = 5 days

2395 **LIGHT with all components:** 5 days

2396 **LIGHT wo scratchpad:** 37 days

2397 **Scratchpad cue (excerpt):**

2398 ****IMPORTANT DATES:****

2399 - ****January 1, 2024:**** Met mother Brittney at home.

2400 - ****January 5-6, 2024:**** Completed first batch of Michele's sourdough.

2400 [truncated]

2401 (2) **Question:** How many days do I have between finalizing my survey plans at Café Soleil and attending
 2402 the Raymondburgh Startup Meetup to prepare effectively?

2403 **Correct Answer:** Mar 28 – Mar 10 = 18 days

2404 **LIGHT with all components:** 18 days

2405 **LIGHT wo scratchpad:** 28 days

2406 **Scratchpad cue (excerpt):**

2407 ****IMPORTANT DATES:****

2408 - ****March 10, 2024:**** Paper-based customer survey at Cafe Soleil.

2409 - ****March 28, 2024:**** Raymondburgh Startup Meetup.

2409 [truncated]

2411 **Takeaways.** Across abilities, removing the scratchpad consistently causes failures that the full
 2412 model avoids. In *Information Extraction*, the scratchpad aggregates dispersed entity/version men-
 2413 tions so the model can recover exact tool versions and bios (city/age). For *Instruction Following*, it
 2414 retains user meta-preferences (e.g., “always include team count”), ensuring style/format compliance
 2415 even many turns later. For *Knowledge Update*, it encodes overrides (e.g., extended deadline; reduced
 2416 memory), preventing stale answers. For *Temporal Reasoning*, it surfaces normalized date anchors,
 2417 enabling simple, correct day-difference calculations. These examples show that the scratchpad pro-
 2418 vides a high-utility semantic scaffold that complements working (recency) and episodic (retrieval)
 2419 memory, yielding robust long-context behavior.

2420

2421

2422

2423

2424

2425

2426

2427

2428

2429

2430 **G QUALITATIVE ERROR ANALYSIS**
2431

2432 We conduct a qualitative analysis of failure cases across the ten memory abilities in our benchmark
2433 to better characterize the limitations of LIGHT and identify systematic patterns. For each ability, we
2434 manually inspected probing questions that LIGHT answered incorrectly and analyzed the underlying
2435 reasons. Below, we summarize the dominant error modes observed for each ability.
2436

2437 **Abstention** In this ability, the LLM should abstain from answering because the answer to the
2438 probing question is not present in the conversation. Therefore, the context that LIGHT provides
2439 to the LLM does not contain the required information. One failure mode occurs when the context
2440 contains nothing relevant to the question, yet the LLM hallucinates and generates an answer. This is
2441 because these LLMs are usually trained to always provide an answer, regardless of actually having
2442 this knowledge (Kalai et al., 2025). Another hallucination pattern occurs when the LLM produces
2443 an answer entirely unrelated to the question, which stems from the long-context nature of the task
2444 and the inability of the LLM to understand the context correctly. The main failure mode, however,
2445 arises when the context contains information about entities, dates, or concepts that are similar to,
2446 but not the same as, the information requested by the question. In these cases, the LLM uses these
2447 similar details and generates an answer instead of abstaining. This pattern is the primary failure
2448 mode for abstention.
2449

2450 **Contradiction Resolution** For this ability, the LLM should identify the contradiction, state both
2451 sides, and request clarification. One common failure mode occurs when the context contains only
2452 one side of the contradiction, leading the LLM to answer based solely on that information. Since
2453 the model does not have access to the other side of the story, it cannot detect the conflict. Another
2454 common failure occurs when both sides of the contradiction are present in the context, but the LLM
2455 still overweights one side of the contradiction due to position and frequency bias for that side in the
2456 context.
2457

2458 **Event Ordering** In this ability, the LLM should recognize and reconstruct the sequence of evolving
2459 information in the conversation. A common failure occurs when the context contains items from
2460 the sequence but the LLM does not include them in the response. Another failure mode occurs
2461 when the model includes the items but presents them in the wrong order. This happens because
2462 the retrieval model retrieves based on similarity, which does not necessarily preserve temporal order,
2463 leaving the LLM without clues about the correct sequence of events. Also, in many cases, the
2464 retriever does not retrieve all the events related to the question.
2465

2466 **Information Extraction** In this ability, one failure mode arises when the context does not contain
2467 the answer to the question; thus, the LLM cannot extract the answer from the retrieved context.
2468 Another occurs when the answer is present, but the LLM produces an incorrect answer because it
2469 becomes confused by details in the context that are similar to the answer. A third failure happens
2470 when the answer is present but the LLM provides an incomplete answer.
2471

2472 **Instruction Following** For this ability, the LLM should adhere to user-specified instructions. Fail-
2473 ures occur either when the user instruction is present in the context, but the LLM does not follow
2474 it, or when the instruction is missing from the context, and the LLM cannot answer the question
2475 without it.
2476

2477 **Knowledge Update** Here, the LLM should answer the question using the updated version of the
2478 facts. One failure mode occurs when the context contains only the old value and does not include the
2479 updated value, causing the LLM to respond using outdated information, because the retrieval model
2480 retrieves based on similarity, which does not necessarily preserve temporal order. A more common
2481 failure mode occurs when the context contains both the old and updated values, but the LLM still
2482 bases its answer on the old value, because the retrieved documents are not necessarily presented in
2483 the correct temporal order, again due to the retriever.
2484

2485 **Multi-Hop Reasoning** For this ability, the LLM often fails when the context contains the neces-
2486 sary pieces of information but the model does not use them to answer the question. Another failure
2487

2484 occurs when the context is missing some components required for the multi-hop reasoning chain,
2485 making it impossible for the LLM to answer correctly.
2486

2487 **Preference Following** In this ability, the LLM should incorporate user-stated preferences into
2488 its answer. Failures occur when the context does not contain the user’s preference and the model
2489 therefore answers without considering it, or when the preference is present but the LLM does not
2490 use it when generating the response.
2491

2492 **Summarization** For this ability, one failure mode occurs when the context contains some compo-
2493 nents of the correct answer but the LLM fails to include them in the summary. Another failure arises
2494 when the context is missing some parts of the answer, which leads the LLM to omit those details as
2495 well.
2496

2497 **Temporal Reasoning** In this ability, the LLM should reason about explicit and implicit temporal
2498 relations. A common failure mode occurs when the context contains the required dates, but the
2499 LLM becomes confused by another date (or dates) in the context and answers incorrectly. Another
2500 failure occurs when the context contains both dates but the LLM performs the arithmetic incorrectly,
2501 producing an answer with a numerical error. A third failure mode occurs when the context does not
2502 contain one of the necessary dates, causing the LLM to incorrectly substitute another date when
2503 answering.
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537

2538
2539

H PROMPTS

2540
2541

Here we provide the prompts used in different stages of our framework.

2542
2543

This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic chat conversations between a user and an AI assistant, which are then used to evaluate the long-term memory capabilities of LLMs.

2544
2545

Your task is to analyze this plan and select bullet points that would be most effective for testing information extraction abilities when incorporated into chat conversations.

2546
2547

Analyze this plan and identify bullet points that contain specific factual information ideal for testing precise recall and information extraction capabilities.

2548
2549

```
## INPUT DATA
- **PLAN**: <plan>

## CRITICAL REQUIREMENT: EARLY BATCH PRIORITIZATION
**SELECTION PRIORITY ORDER:**
```

1. **Batch 1-3 (HIGHEST PRIORITY)**: Select 70-80% of your choices from these early batches
2. **Batch 4-6 (MEDIUM PRIORITY)**: Select 10-20% of your choices from these middle batches
3. **Batch 7+ (LOW PRIORITY)**: Select only 5-10% of your choices from later batches

2550
2551

Focus on bullet points with:

- **Specific numbers, quantities, measurements, prices, percentages**
- **Proper names of people, organizations, brands, locations**
- **Exact dates, times, schedules, or deadlines**
- **Contact details such as addresses, phone numbers, email IDs**
- **Technical or detailed descriptions (model names, product codes, ratings, specifications)**
- **Distinctive events, awards, or milestones**
- **Direct quotes, messages, or instructions with exact wording**
- **Precise parameters, formulas, or datasets in technical and academic contexts**
- **Mathematical expressions, theorems, proofs**

2552
2553

Prioritize information that:

- Appears early in the timeline
- Contains multiple distinct factual details in one bullet point
- Includes uncommon names, technical terms, or culturally specific references
- Has precise numerical values or measurements that could be easily confused
- Could be misremembered if details are swapped, rounded, or reworded
- Requires high accuracy to preserve meaning (e.g., formulas, addresses, step-by-step processes)

2554
2555

Return your analysis in this exact JSON format:

```
[{"capability": "information_extraction", "batch_numbers": 1, "bullet_numbers": 3,
    "bullet_points": "    **Personal Introduction:** I am Sherry Rodriguez, 34, licensed conveyancer in
                    Hollyborough, Bahrain, earning approximately $68,000 annually."}]
```

2556
2557

Important formatting notes:

- The "batch_numbers" and "bullet_numbers" correspond to each other positionally
- "1" and "3" means: Batch 1 Bullet 3

2558
2559

Select ONLY <bullet_number> bullet points total that would generate the highest quality information extraction questions.

2560
2561

NOTE: Only output the list without any explanation before or after the list.

2562
2563

Listing 1: Candidate selection information extraction prompt

2564
2565

This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic chat conversations between a user and an AI assistant, which are then used to evaluate the long-term memory capabilities of LLMs.

2566
2567

Your task is to analyze this plan and select GROUPS of related bullet points that would be most effective for testing multi-session reasoning abilities when incorporated into chat conversations.

2568
2569

Analyze this project plan and identify GROUPS of bullet points that enable testing of aggregation, comparison, and synthesis across multiple batches/sessions. Each group should contain 2-6 related bullet points that together enable complex multi-hop reasoning questions.

2570
2571

```

2592
2593     - Enable mathematical aggregation across multiple entries (total costs, time durations, quantity counting)
2594     - Allow before/after comparisons of the same entities across different batches
2595     - Require synthesis of information from 3+ different batches
2596     - Create opportunities for complex multi-hop reasoning questions
2597
2598     Return your analysis in this exact JSON format where each object contains multiple related bullet points:
2599     [{"capability": "multi_session_reasoning", "batch_numbers": "1, 2, 3, 5", "bullet_numbers": "10, 4, 7, 7",
2600         "bullet_points": "Financial & Budget:Cost Estimation: Initial budget set at $12,500, including
2601             materials and labor for decoupled framing and MLV. | Financial & Budget:Expense Tracking: Paid
2602             $2,200 deposit to QuietFlow Bahrain for HVAC silencing on March 14 via bank transfer. | Financial & Budget:Expense Tracking: Total spent $9,300 by April 1 on materials, labor, and
2603             consultant fees. | Financial & Budget:Expense Tracking: Total project spending $12,000 as of May
2604             1, within original $12,500 budget."}
2605
2606     Important formatting notes:
2607     - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2608     - "1, 2, 3, 5" and "10, 4, 7, 7" means: Batch 1 Bullet 10, Batch 2 Bullet 4, Batch 3 Bullet 7, Batch 5 Bullet
2609         7
2610     - Use comma-separated values for batch_number and bullet_number
2611     - Separate multiple bullet_point entries with " | "
2612     - Each group should contain 2-6 related bullet points
2613     - Focus on groups that enable the most sophisticated multi-hop aggregation and comparison questions
2614
2615     Select 8-12 groups of bullet points that would enable the most sophisticated multi-session reasoning questions
2616
2617     .
2618
2619     NOTE: Only output the list without any explanation before or after the list.

```

Listing 2: Candidate selection multi-hop reasoning prompt

```

2611 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
2612     chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
2613     memory capabilities of LLMs.
2614 Your task is to analyze this plan and select PAIRS of related bullet points that would be most effective for
2615     testing knowledge update abilities when incorporated into chat conversations.
2616 Analyze this plan and identify bullet points labeled as "Information Update" and match them with their
2617     corresponding original facts from earlier in the plan.
2618
2619     ## INPUT DATA
2620     - **PLAN**: <plan>
2621
2622     ## CRITICAL REQUIREMENT: SPECIAL UPDATE BULLETS WITH ORIGINAL FACTS
2623     Focus on bullet points that:
2624     - **Are labeled "Information Update"**: Look specifically for bullet points with this exact label
2625     - **Have corresponding original facts**: Find the earlier bullet point that contains the original information
2626         being updated
2627     - **Show clear before/after relationships**: Original information paired with its explicit update or
2628         correction
2629
2630     For each "Information Update" bullet point you find:
2631     1. **Locate the original fact** in an earlier bullet point that this update refers to
2632     2. **Create a pair** with the original bullet point first, then the "Information Update" bullet point second
2633     3. **Ensure clear connection** between the original fact and its update
2634
2635     Look specifically for "Information Update" bullet points that contain:
2636     - Clear update language ("updated," "changed," "revised," "rescheduled," "increased," "decreased")
2637     - References to modifications of previously mentioned information
2638     - Corrections or adjustments to earlier facts
2639     - Timeline or specification changes
2640     Then match each update with its original fact from earlier bullet points.
2641
2642     Return your analysis in this exact JSON format where each object contains exactly TWO related bullet points (one
2643         original + Information Update):
2644     [{"capability": "knowledge_update", "batch_numbers": "1, 3", "bullet_numbers": "9, 31",
2645         "bullet_points": "Financial & Budget:Cost Estimation: Initial budget set at $12,500, including
2646             materials and labor for decoupled framing and MLV. | Information Update: The initial framing
2647             materials purchase included an additional 10% surplus to accommodate unexpected cuts and errors
2648             ."}
2649
2650     Important formatting notes:
2651     - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2652     - "1, 3" and "9, 31" means: Batch 1 Bullet 9 (original), Batch 3 Bullet 31 (Information Update)
2653     - Each object must contain exactly 2 bullet points separated by " | "
2654     - Use comma-separated values for batch_number and bullet_number
2655     - First bullet point should represent the original information
2656     - Second bullet point should be the "Information Update" labeled bullet
2657     - Focus on pairs that enable questions like "How did the original plan change when you got the update?"
2658
2659     Select all "Information Update" bullet points and pair them with their corresponding original facts (approximately 10 pairs total).
2660
2661     NOTE: Only output the list without any explanation before or after the list.

```

Listing 3: Candidate selection knowledge update prompt

```

2662
2663
2664
2665     This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
2666     chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
2667     memory capabilities of LLMs.

```

```

2646 Your task is to analyze this plan and select PAIRS of related bullet points that would be most effective for
2647 testing temporal reasoning abilities when incorporated into chat conversations.
2648 Analyze this project plan and identify PAIRS of bullet points that enable testing duration calculations and
2649 sequence understanding between two events. Each pair should enable questions about time duration,
2650 sequence, or temporal relationships between two events.
2651
2652 ## INPUT DATA
2653 - **PLAN**: <plan>
2654
2655 ## CRITICAL REQUIREMENT: BALANCED BATCH DISTRIBUTION
2656 **SELECTION PRIORITY ORDER:***
2657 1. **Pairs starting in Batches 1-3 (MEDIUM-HIGH PRIORITY)**: Select 40-50% of your pairs with at least one
2658 bullet from early batches
2659 2. **Far-distance pairs (HIGH PRIORITY)**: Select 30-40% of pairs that span large batch distances (e.g., Batch
2660 1 & Batch 6, Batch 2 & Batch 8, Batch 1 & Batch 7, etc.) to test long-term temporal reasoning
2661 3. **Pairs spanning early to middle batches (MEDIUM PRIORITY)**: Select 10-15% of pairs that bridge early-to-
2662 middle timeline
2663 4. **Pairs from later batches only (LOW PRIORITY)**: Select only 5-10% from purely later batches
2664
2665 Focus on bullet point pairs that: - Enable duration calculations between two time points - Show sequence
2666 relationships between events
2667 - Allow comparison of timing across different batches - Demonstrate temporal progression or changes over time
2668 - Include scheduling, deadlines, or milestone comparisons
2669
2670 ## EXPLICIT TIME MENTION REQUIREMENTS
2671 **ONLY** absolute dates count as explicit time mentions:***
2672 [Examples]
2673
2674 **THESE DO NOT COUNT as explicit time mentions:***
2675 - Specific times - Calendar references - Specific weekdays - Relative durations - Time periods - Vague
2676 references - Duration spans
2677
2678 ## IMPORTANT TIME ANCHOR RULES:
2679 1. If BOTH bullet points contain explicit absolute dates, use them as-is
2680 2. If ONE bullet point lacks explicit absolute dates, prepend that bullet point with its batch's Time Anchor
2681 3. If BOTH bullet points lack explicit absolute dates, prepend both with their respective Time Anchors
2682
2683 FORMAT EXAMPLES:
2684 Case 1 - Both have time mentions (no Time Anchor needed):
2685 [Example]
2686 Case 2 - Second bullet point lacks explicit absolute dates (add Time Anchor to second):
2687 [Example]
2688 Case 3 - Both bullet points lack explicit absolute dates (add Time Anchors to both):
2689 [Example]
2690
2691 Return your analysis in this exact JSON format where each object contains exactly TWO related bullet points:
2692 [{"capability": "temporal_reasoning", "batch_numbers": "1, 2", "bullet_numbers": "17, 9",
2693 "bullet_points": "Bullet Description: ... | Bullet Description: ..."}]
2694
2695 Important formatting notes:
2696 - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2697 - "1, 2" and "17, 9" means: Batch 1 Bullet 17, Batch 2 Bullet 9
2698 - Each object must contain exactly 2 bullet points separated by " | "
2699 - Use comma-separated values for batch_number and bullet_number
- Add Time Anchors before bullet points that lack explicit time mentions
- Focus on pairs that enable duration calculation questions like "How many days between X and Y?"
2700
2701 Select 8-10 pairs of bullet points that would enable the most sophisticated temporal reasoning and duration
2702 calculation questions.
2703
2704 NOTE: Only output the list without any explanation before or after the list.

```

Listing 4: Candidate selection temporal reasoning prompt

```

2705 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
2706 chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
2707 memory capabilities of LLMs.
2708 Your task is to analyze this plan and select bullet points that would be most effective for testing preference
2709 following abilities when incorporated into chat conversations.
2710 Analyze this plan and identify bullet points labeled as "Preference Statement" and select all.
2711
2712 ## INPUT DATA
2713 - **PLAN**: <plan>
2714
2715 Focus on bullet points with:
2716 - **Explicit preference statements**: "I prefer", "I like", "I choose", "I favor"
2717 - **Decision choices**: Selections between options with stated reasoning
2718 - **Personal preferences**: Style, approach, method, or format preferences
2719 - **Avoidance statements**: "I don't like", "I avoid", "I prefer not to"
2720 - **Priority preferences**: What user values most or considers important
2721
2722 Prioritize preferences that:
2723 - Are clearly stated with specific reasoning
2724 - Involve choices between multiple options
2725 - Contain detailed preference explanations
2726 - Include comparative preferences (X over Y)
2727 - Express strong preferences or dislikes
2728 - Relate to recurring decisions or situations
2729
2730 NOTE: ONLY CONSIDER "PREFERENCE" NOT INSTRUCTION.

```

```

2700
2701     Return your analysis in this exact JSON format:
2702     [{"capability": "preference_following", "batch_numbers": 1, "bullet_numbers": 17,
2703      "bullet_points": "    **Preference Statement:** I prefer materials that balance cost and performance;
2704      choose 3.5 lb/ft  MLV despite 20% higher price."}
2705
2706     Important formatting notes:
2707     - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2708     - "1" and "17" means: Batch 1 Bullet 17
2709
2710     Select ONLY <bullet_number> bullet points total that would generate the highest quality preference following
2711     questions.
2712
2713     NOTE: Only output the list without any explanation before or after the list.

```

Listing 5: Candidate selection preference following prompt

```

2714
2715     This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
2716     chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
2717     memory capabilities of LLMs.
2718     Your task is to analyze this plan and select GROUPS of related bullet points that would be most effective for
2719     testing event ordering abilities when incorporated into chat conversations.
2720     Event ordering tests whether the LLM can recall the chronological order in which events or topics were
2721     MENTIONED in the conversation, regardless of when the actual events occurred in real life.
2722     Analyze this plan and identify GROUPS of 8-12 or more related bullet points that represent the same topic/
2723     theme mentioned across different batches, enabling testing of mention-order recall and conversation
2724     sequence understanding.
2725
2726     ## INPUT DATA
2727     - **PLAN**: <plan>
2728
2729     ## CRITICAL REQUIREMENT: EARLY BATCH PRIORITIZATION
2730     **SELECTION PRIORITY ORDER:***
2731     1. **Groups starting in Batches 1-3 (HIGHEST PRIORITY)**: Select 70-80% of your groups with first mention in
2732        early batches
2733     2. **Groups spanning early to middle batches (MEDIUM PRIORITY)**: Select 10-20% of groups that bridge early-to-
2734        middle timeline
2735     3. **Groups from later batches only (LOW PRIORITY)**: Select only 5-10% from purely later batches
2736
2737     Focus on bullet point groups that show:
2738     - **Same person mentioned multiple times**: Different interactions or mentions of the same person across
2739        batches
2740     - **Same component/process discussed repeatedly**: Multiple mentions of the same equipment, material, or
2741        process
2742     - **Same location/venue referenced**: Multiple mentions of the same place or address
2743     - **Same decision/topic revisited**: The same subject brought up in different conversation sessions
2744     - **Same problem/solution mentioned**: Multiple references to the same issue across different times
2745     - **Same financial item tracked**: Multiple mentions of the same cost, budget item, or expense
2746
2747     Prioritize bullet point groups that:
2748     - Contain 8-12 mentions of the same topic across different batches
2749     - Enable questions about "In what order events X,Y,Z,... happen?" Or ...
2750     - Allow testing of conversation chronology rather than real-world event chronology
2751     - Test recall of mention sequence: "Which did I talk about first, second, third?"
2752     - Focus on the order topics appeared in conversation, not when events actually happened
2753     - Create opportunities to test conversational memory rather than factual timeline memory
2754
2755     Return your analysis in this exact JSON format where each object contains 3+ bullet points about the same
2756     topic across different batches:
2757     [{"capability": "event_ordering", "batch_numbers": "1, 3, 5, 7", "bullet_numbers": "22, 18, 11, 27",
2758      "bullet_points": "    **Character & Relationship:Acoustic Consultant:** Met Rami Al-Hassan at Bahrain
2759      Acoustic Expo on Feb 20; he recommended HVAC silencing at $2,200. |    **Character &
2760      Relationship:Acoustic Consultant:** Rami conducted mid-project site visit April 3, advised on
2761      bass trap repositioning to improve 5 dB absorption. |    **Character & Relationship:Acoustic
2762      Consultant:** Rami praised progress in May 1 email; suggested minor EQ tweaks. |    **Character
2763      & Relationship:Acoustic Consultant:** Rami praised final results August 20; recommended ongoing
2764      maintenance and periodic EQ checks."}
2765
2766     Important formatting notes:
2767     - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2768     - "1, 2, 3, 5, 7" and "22, 18, 11, 27" means: Batch 1 Bullet 22, Batch 3 Bullet 18, Batch 5 Bullet 11, Batch 7
2769     Bullet 27
2770     - Each object must contain 8-12 related bullet points separated by " | "
2771     - Use comma-separated values for batch_number and bullet_number
2772     - All bullet points must reference the same topic/person/component/theme
2773     - Focus on groups that enable mention-order questions
2774     - Test conversational chronology, not real-world event chronology
2775
2776     Select 8-10 groups of bullet points that would enable the most sophisticated mention-order and conversation
2777     sequence questions.
2778
2779     CRITICAL NOTE: DO NOT consider bulletpoint names for selecting the bullets. ONLY consider the bullets contents
2780
2781     NOTE: Only output the list without any explanation before or after the list.

```

Listing 6: Candidate selection event ordering prompt

2754 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
 2755 chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
 2756 memory capabilities of LLMs.
 2757 Your task is to analyze this plan and select PAIRS of bullet points that would be most effective for testing
 contradiction resolution abilities when incorporated into chat conversations.
 2758 Contradiction resolution tests whether the LLM can detect and appropriately handle impossible contradictions -
 statements that logically cannot both be true simultaneously.
 2759 Analyze this project plan and identify PAIRS of bullet points where one completely contradicts the other with
 impossible contradictions. Each pair should contain statements that are logically incompatible and
 cannot both be true.
 2760

```
## INPUT DATA
- **PLAN**: <plan>
```

2761 Focus on bullet point pairs that show:
 2762
 - **Never-Statement Violations**: One bullet says "never" did something, another shows they did it
 - **Always-Statement Violations**: One bullet claims "always" pattern, another breaks that pattern
 - **Only-Statement Conflicts**: One bullet claims exclusivity ("only"), another contradicts it
 - **Impossible Reversals**: Age going backward, timeline impossibilities, logical reversals
 - **Dead-Alive Contradictions**: References to deceased people being active
 - **Mutually Exclusive States**: Being in two places simultaneously, having contradictory capabilities
 - **Absolute Negations**: Claiming something is impossible then showing it happened

2763 **Types of Impossible Contradictions to look for:**
 2764 1. **Never-Statement Violations**: "Never attended X" vs "Attended X event"
 2. **Always-Statement Violations**: "Always lived in Y" vs "Moved from Z to Y"
 2765 3. **Only-Statement Conflicts**: "Only child" vs "Has siblings"
 4. **Timeline Impossibilities**: Events happening in wrong chronological order
 2766 5. **Capability Contradictions**: "Cannot do X" vs "Successfully did X"
 6. **Location Impossibilities**: Being in two places at once
 2767 7. **Relationship Contradictions**: "Never met person" vs "Long friendship with person"

2768 Prioritize bullet point pairs that:
 2769 - Contain completely impossible contradictions that cannot be resolved or explained
 2770 - Use absolute language ("never," "always," "only," "impossible," "cannot")
 2771 - Create clear logical impossibilities rather than simple inconsistencies
 2772 - Enable questions about detecting fundamental contradictions
 2773 - Test whether the AI can identify when statements are mutually exclusive
 2774 - Focus on contradictions that are objectively impossible, not subjective differences

2775 Return your analysis in this exact JSON format where each object contains exactly TWO contradicting bullet
 points:
 2776 [{"capability": "contradiction_resolution", "batch_numbers": "1, 8", "bullet_numbers": "30, 29",
 2777 "bullet_points": " **Logical Contradiction:** Jeremiah has never attended any Bahrain Jazz Festival
 2778 events. | **Character & Relationship:** Close Friend:** Jeremiah, 37, met at Bahrain Jazz
 2779 Festival 2015, recommended an acoustic consultant."}
 2780]

2781 Important formatting notes:
 2782 - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
 2783 - "1, 8" and "30, 29" means: Batch 1 Bullet 30, Batch 8 Bullet 29
 2784 - Each object must contain exactly 2 bullet points separated by " | "
 2785 - Use comma-separated values for batch_number and bullet_number
 2786 - First bullet point can be the contradiction marker or the contradicted statement
 2787 - Second bullet point should directly contradict the first with impossible logic
 2788 - Focus on pairs that test detection of fundamental logical impossibilities

2789 Select <bullet_number> pairs of bullet points that demonstrate the clearest impossible contradictions for
 testing contradiction resolution abilities.

2790 NOTE: Only output the list without any explanation before or after the list.

Listing 7: Candidate selection contradiction resolution prompt

2791
 2792 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
 2793 chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
 2794 memory capabilities of LLMs.
 2795 Your task is to analyze this plan and select GROUPS of bullet points that would be most effective for testing
 summarization abilities when incorporated into chat conversations.
 2796 Summarization tests whether the LLM can synthesize and condense information from across multiple conversation
 sessions into coherent, comprehensive summaries.
 2797 Analyze this plan and identify GROUPS of 8-12 related bullet points that represent topics suitable for
 summarization testing. Groups can vary in size depending on the richness and complexity of the topic.
 2798

```
## INPUT DATA
- **PLAN**: <plan>
```

2799 ## CRITICAL REQUIREMENT: EARLY BATCH PRIORITIZATION
 2800 **SELECTION PRIORITY ORDER:**
 2801 1. **Groups starting in Batches 1-3 (HIGHEST PRIORITY)**: Select 60-70% of your groups with foundational
 2802 content from early batches
 2803 2. **Groups spanning early to middle batches (MEDIUM PRIORITY)**: Select 20-30% of groups that bridge early-to
 2804 -middle timeline
 2805 3. **Groups from later batches only (LOW PRIORITY)**: Select only 10-20% from purely later batches

2806 ## CRITICAL REQUIREMENT: CONTENT-BASED ANALYSIS
 2807 **ANALYZE BULLET CONTENT, NOT CATEGORY NAMES:**
 - Read the actual bullet point text to identify mentions of entities (people, places, items, topics, amounts,
 processes)
 - The same entity might appear in different category types - include mentions across all categories
 - The same process might appear in different category types - include mentions across all categories

2808 - The same project might appear in different category types - include mentions across all categories
 2809
 2810 ****SEARCH METHODOLOGY:****
 2811 1. ****Identify key entities**** in bullet content: names, places, amounts, equipment, topics, processes
 2811 2. ****Search ALL batches**** for any mention of these entities in ANY category
 2811 3. ****Group by content similarity****, not category similarity
 2811 4. ****Include 8-12 mentions**** regardless of how they're categorized
 2812
 2813 Focus on complete topic clusters that enable summarization of:
 2814 - ****Entity or Relationship Histories****: interactions, developments, or changes related to a specific person, organization, group, or other identifiable entity across the entire plan (complete relationship or entity arc)
 2815 - ****End-to-End Processes****: steps, stages, or phases of a specific process, workflow, or methodology from initiation to conclusion across the entire plan (no missing steps)
 2816 - ****Resource or Asset Lifecycles****: mentions of acquisition, allocation, usage, modification, and outcomes for a specific resource, asset, or material across the entire plan
 2817 - ****Decision and Strategy Journeys****: details of decision-making, planning, and strategy development from problem identification through implementation across the entire plan
 2818 - ****Problem/Challenge Resolution Narratives****: instances of identifying, analyzing, addressing, and resolving a particular issue or challenge across the entire plan
 2819 - ****Timeline-Driven Developments****: events and updates showing chronological evolution of a specific project, initiative, or topic across the entire plan
 2820 - ****Knowledge or Skill Development Sequences****: progress updates, milestones, and learning activities related to acquiring or improving a specific skill or knowledge area across the entire plan
 2821 - ****Discussion and Agreement Processes****: discussions, debates, negotiations, and agreements related to a specific matter across the entire plan
 2822
 2823 Prioritize bullet point groups that:
 2824 - Contain rich, interconnected information suitable for synthesis
 2824 - Enable questions like "Can you summarize my interactions with X?" or "Summarize the [X] process"
 2825 - Include both factual/quantitative details and qualitative/narrative elements for well-rounded summaries
 2826 - Have varying complexity levels (simple single-topic vs. complex multi-faceted stories)
 2827 - Allow testing of information condensation across multiple conversation sessions
 2828 - Include both factual details and narrative elements for comprehensive summarization
 2829 - Create opportunities to test synthesis of scattered information into coherent narratives
 2830
 2831 Return your analysis in this exact JSON format where each object contains 8-12 related bullet points:
 2831 [{"capability": "summarization", "batch_numbers": "1, 1, 2, 3, 4, 5", "bullet_numbers": "5, 22, 18, 19, 23,
 31",
 2832 "bullet_points": " ****Character & Relationship:Close Friend:**** Jeremiah, 37, met at festival,
 2832 recommended consultant. | ****Conflict & Resolution:Relationship Boundaries:**** Jeremiah
 2833 requested exclusive studio access; agreed to 3 hours only. | ****Character & Relationship:**
 2834 **Close Friend:**** Jeremiah helped install MLV, bringing snacks from bakery. | ****Character &**
 2834 **Relationship:Close Friend:**** Jeremiah invited me to music club to test prototype. | ******
 2835 **Character & Relationship:Close Friend:**** Jeremiah brought dinner during late work session. |
 2835 ****Goals & Progress:Milestone Celebration:**** Hosted listening party with Jeremiah, Jon, and
 2836 Tonya."}
 2837]
 2838
 2839 Important formatting notes:
 2839 - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
 2839 - "1, 1, 2, 3, 4, 5" and "5, 22, 18, 19, 23, 31" means: Batch 1 Bullet 5, Batch 1 Bullet 22, Batch 2 Bullet
 2840 18, Batch 3 Bullet 19, Batch 4 Bullet 23, Batch 5 Bullet 31
 2840 - Each bullet point separated by " | "
 2841 - Use comma-separated values for batch_number and bullet_number
 2841 - Vary group sizes based on topic complexity and richness
 2842 - Focus on groups that enable comprehensive summarization questions
 2842 - Include both simple single-topic and complex multi-topic groups
 2842 - ****NO LIMIT** on number of bullet points* - include as many as needed for complete coverage
 2843
 2844 **CRITICAL NOTES:**
 2844 - ****ANALYZE BULLET CONTENT, NOT CATEGORY NAMES****: Search for mentions of entities in the actual text
 2845 - ****IGNORE CATEGORY LABELS****: The same entity mentioned in different category types should all be grouped together
 2846 - ****COMPREHENSIVE ENTITY SEARCH****: For each entity/topic/process, scan ALL batches and ALL categories for any mention
 2847 - Include 8-12 mentions regardless of bullet point category if they reference the same entity/topic in the content
 2848
 2849 Select 7-9 groups of bullet points with COMPLETE mention coverage that would enable the most sophisticated and comprehensive summarization questions across different complexity levels.
 2850
 2851 **NOTE:** Only output the list without any explanation before or after the list.

Listing 8: Candidate selection summarization prompt

2852
 2853 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
 2853 chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
 2854 memory capabilities of LLMs.
 2855 Your task is to analyze this plan and select bullet points that would be most effective for testing
 2855 instruction following abilities when incorporated into chat conversations.
 2856 Analyze this plan and identify bullet points that contain user instructions ideal for testing whether the LLM
 2856 remembers and follows user-given instructions.
 2857
 2858 **## INPUT DATA**
 2859 - ****PLAN****: <plan>
 2860
 2861 Focus on bullet points with:
 2861 - ****User Instruction**** category/label
 2861 - ****Explicit instruction statements****: "Always", "Never", "When I ask about X, do Y"
 2861 - ****Behavioral directives****: How the AI should respond or behave

```

2862
2863 - Format instructions: Specific response formats or structures requested
2864 - Content instructions: What to include or exclude in responses
2865 - Process instructions: How to handle specific types of requests
2866
2867 Look specifically for bullet points labeled as "User Instruction" that contain:
2868 - Clear directive language ("Always provide", "Never include", "When I ask")
2869 - Specific behavioral expectations for the AI assistant
2870 - Conditional instructions ("When I ask about X, do Y")
2871 - Response formatting requirements
2872 - Content inclusion/exclusion rules
2873
2874 Return your analysis in this exact JSON format:
2875 [ {"capability": "instruction_following", "batch_numbers": 1, "bullet_numbers": 32,
2876     "bullet_points": "User Instruction: Always provide detailed cost breakdowns when I ask about budget
2877     estimates."}
2878 ]
2879
2880 Important formatting notes:
2881 - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
2882 - "1" and "32" means: Batch 1 Bullet 32
2883 - Include the full bullet point text as it appears in the plan
2884 - Focus specifically on "User Instruction" labeled bullet points
2885
2886 Select all bullet points labeled as "User Instruction" from each batch (approximately 10 total).
2887
2888 NOTE: Only output the list without any explanation before or after the list.

```

Listing 9: Candidate selection instruction following prompt

```

2889
2890 You are tasked with generating a probing question to test information extraction capabilities of LLMs. You
2891 will be given a bullet point and the corresponding multi-turn dialog between a user and assistant that
2892 incorporates this bullet point information.
2893
2894 Your task is to create ONE question that tests whether an LLM can precisely extract and recall specific
2895 factual details from the conversation through indirect questioning that requires synthesizing multiple
2896 details from different parts of the conversation.
2897
2898 ## INPUT DATA
2899 - **BULLET POINT**: <bullet_point>
2900 - **CONVERSATION TURNS**: <conversation_turns>
2901
2902 ## CRITICAL REQUIREMENT: INFORMATION EXTRACTION
2903 - The question MUST NOT directly ask for the information being tested
2904 - Ask about related topics/contexts that require the LLM to synthesize multiple details from different
2905 conversation parts
2906 - Force the LLM to extract and combine information scattered across different conversation turns
2907 - Make the LLM demonstrate knowledge of facts without being directly asked for them
2908 - Require connecting and integrating information from multiple different conversation elements
2909 - Remove ALL specific details from the question that would give away the answer
2910
2911 ## FORBIDDEN QUESTION ELEMENTS
2912 - Do NOT repeat specific names, numbers, or details being tested
2913 - Do NOT mention key characteristics or attributes being extracted
2914 - Do NOT include descriptive words that hint at the answer
2915 - Do NOT reference specific categories or types being tested
2916 - Do NOT use qualifying details that narrow down the answer
2917
2918 ## QUESTION LANGUAGE REQUIREMENTS
2919 - Write questions as if the USER is asking them naturally
2920 - Questions MUST ONLY BE from USER language not ASSISTANT
2921 - **If testing information from USER messages**: Use first person ("I", "my", "me") in question Answer
2922     uses ("you", "your")
2923 - Example: "How did I decide on the location?" "You decided on the location because..."
2924 - **If testing information from ASSISTANT messages**: Use second person ("you", "your") in question Answer
2925     uses ("I", "my")
2926 - Example: "What steps did you suggest for handling this?" "I suggested doing..."
2927 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
2928 - Make questions sound conversational and natural
2929 - Questions should flow naturally as if continuing the conversation
2930 - Ask about context, or relationships rather than direct facts
2931
2932 ## INDIRECT QUESTIONING STRATEGIES
2933 ## 1. **Context-Based Recall** Ask about the surrounding circumstances instead of the exact fact
2934 [Example]
2935 ## 2. **Comparison Questions** Encourage differentiation between similar elements
2936 [Example]
2937 ## 3. **Timeline Integration** Link facts to their sequence in time
2938 [Example]
2939 ## 4. **Problem-Solution Context** Frame questions around issues and how they were addressed:
2940 [Example]
2941 ## 5. **Discovery and Learning Process** Focus on the origin of knowledge or awareness
2942 [Example]
2943 ## 6. **Relationship and Connection Context** Test understanding of associations
2944 [Example]
2945
2946 ## FORBIDDEN DIRECT QUESTIONS
2947 [Examples]
2948
2949 ## CHAT ID TRACKING REQUIREMENT
2950 - You MUST identify which specific chat_id(s) contain the information being tested - List ALL chat_ids where
2951     the answer appears

```

```

2916
2917     - NOTE: If the answer is spread out between multiple chat_ids, group them in one list - NOTE: DO NOT INCLUDE
2918         chat_ids in the answer
2919     - If answer spans multiple chats, include all relevant chat_ids - Use the exact chat_id numbers from the
2920         conversation turns
2921
2922     ## DIFFICULTY LEVEL: HARD
2923     - **Hard**: Requires synthesizing multiple details from different parts of conversation
2924     - Force integration of information scattered across multiple conversation turns
2925     - Test ability to connect related facts from different conversation contexts
2926     - Require deep understanding and synthesis rather than simple recall
2927
2928     ## OUTPUT FORMAT
2929     Return your analysis in this exact JSON format:
2930     {
2931         "question": [], "answer": [], "difficulty": "hard", "question_type": # one of: [] "conversation_reference
2932             ": "", "key_facts_tested": "", "extraction_challenge": "", "source_chat_ids": [x, y, ...]
2933     }
2934
2935     ## IMPORTANT REQUIREMENTS
2936     1. **Indirect questioning**: Ask about context rather than direct facts
2937     2. **Question source flexibility**: Questions can be based on information from EITHER user messages OR
2938         assistant messages
2939     3. **Perspective matching**: Question perspective must match the source of information:
2940         - **User info** "I/my/me" question "you/your" answer
2941         - **Assistant info** "you/your" question "I/my" answer
2942     4. **Assistant information questions**: When testing assistant advice/suggestions, use "What did you suggest/
2943         recommend/advise..." format
2944     5. **Multi-detail synthesis**: Question should require combining information from different conversation parts
2945     6. **Cross-turn integration**: Force LLM to connect scattered information across multiple turns
2946     7. **Complex reasoning**: Require understanding of relationships and synthesis of multiple elements
2947     8. **Challenging extraction**: Force LLM to demonstrate knowledge through indirect demonstration
2948
2949     Generate ONE high-quality indirect information extraction question that tests recall of specific factual
2950         details through contextual questioning requiring synthesis of multiple details from different parts of
2951         the conversation.
2952
2953     NOTE: Only output the JSON object without any explanation before or after.

```

Listing 10: Information extraction probing question generation prompt

```

2954
2955     You are tasked with generating a probing question to test multi-session reasoning capabilities of LLMs. You
2956         will be given multiple related bullet points and the corresponding multi-turn dialogs between a user and
2957         assistant that incorporate this information across different conversation sessions.
2958
2959     Your task is to create ONE question that tests whether an LLM can perform complex multi-hop reasoning,
2960         synthesis, and analysis across 4+ conversation sessions.
2961
2962     ## INPUT DATA
2963     - **BULLET POINTS**: <bullet_points>
2964     - **CONVERSATION TURNS**: <conversation_turns>
2965
2966     ## CRITICAL REQUIREMENT: HARD MULTI-SESSION REASONING
2967     - The question MUST NOT include any explicit numbers, dates, times, duration, or temporal references
2968     - Focus on complex synthesis requiring multi-hop reasoning across 4+ sessions
2969     - Test sophisticated analysis that requires connecting multiple data points
2970     - Ask for complex calculations, patterns, or insights that need advanced reasoning
2971
2972     ## QUESTION GENERATION GUIDELINES
2973     Focus on creating questions that require: - **Complex Aggregation** - **Advanced Synthesis** - **Multi-hop
2974         Reasoning** - **Pattern Recognition** - **Performance Evaluation** - **Comparative Analysis** - **Predictive Reasoning**
2975
2976     ## QUESTION TYPES TO GENERATE (HARD LEVEL)
2977     1. **Complex Multi-hop Calculation** 2. **Performance Evaluation** 3. **Multi-variable Comparison** 4. **
2978         Complex Evolution Analysis**
2979
2980     ## REASONING COMPLEXITY LEVEL: HARD
2981     - **Hard**: Requires complex multi-hop reasoning, synthesis, and analysis across 4+ sessions
2982     - Focus on sophisticated calculations or insights requiring advanced reasoning
2983     - Test ability to identify complex patterns, correlations, or relationships
2984     - Include deep analytical thinking and synthesis of multiple data points
2985
2986     ## QUESTION LANGUAGE REQUIREMENTS
2987     - Write questions as if the USER is asking them naturally
2988     - **If testing information from USER messages**: Use first person ("I", "my", "me") in question Answer
2989         uses ("you", "your")
2990         - Example: "How did I decide on the location?" "You decided on the location because..."
2991     - **If testing information from ASSISTANT messages**: Use second person ("you", "your") in question Answer
2992         uses ("I", "my")
2993         - Example: "What steps did you suggest for handling this?" "I suggested doing..."
2994     - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
2995     - Make questions sound conversational and natural
2996     - Questions should flow naturally as if continuing the conversation
2997
2998     ## CHAT ID TRACKING REQUIREMENT
2999     - You MUST identify which specific chat_id(s) contain the information needed for reasoning
3000     - List ALL chat_ids where relevant information appears across the reasoning chain
3001     - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
3002     - NOTE: DO NOT INCLUDE chat_ids in the answer
3003     - If reasoning spans multiple chats, include all relevant chat_ids

```

```

2970 - Use the exact chat_id numbers from the conversation turns
2971
2972 ## OUTPUT FORMAT
2973 You will output exactly ONE JSON object matching this schema:
2974 {
2975     "question": string, "answer": string, "difficulty": "hard", "reasoning_type": [Some categories]
2976     "sessions_required": integer, "conversation_references": [string,...], "reasoning_steps": [string,...], "
2977     source_chat_ids": [integer,...]
2978 }
2979
2980 ## IMPORTANT REQUIREMENTS
2981 1. **Complex multi-session dependency**: Question must require sophisticated information synthesis from 4+
2982     conversation sessions
2983 2. **Question source**: The question and answer to it MUST BE based on information from USER messages in
2984     CONVERSATION TURNS. You MUST NOT generate questions about assistant responses or suggestions.
2985 3. **User information only**: Only create questions that test details the user provided, not assistant advice
2986     or recommendations.
2987 5. **Advanced reasoning path**: Provide complex reasoning steps that require multi-hop thinking
2988 6. **Precise answer**: Give exact answer that demonstrates complex analysis
2989 7. **Session references**: Note which sessions contain relevant information
2990 8. **High complexity**: Ensure question requires advanced multi-session reasoning and sophisticated synthesis
2991
2992 Generate ONE high-quality hard multi-session reasoning question.
2993
2994 NOTE: Only output the JSON object without any explanation before or after.

```

Listing 11: Multi-hop reasoning probing question generation prompt

```

2995 You are tasked with generating a probing question to test knowledge update capabilities of LLMs. You will be
2996     given two related bullet points (original information and updated information) and the corresponding
2997     multi-turn dialogs between a user and assistant that incorporate both pieces of information across
2998     different conversation sessions.
2999
3000 Your task is to create ONE question that asks about the current/updated state of information, testing whether
3001     the LLM correctly recalls the most recent version rather than outdated information.
3002
3003 ## INPUT DATA
3004 - **BULLET POINTS**: <bullet_points>
3005 - **CONVERSATION TURNS**: <conversation_turns>
3006
3007 ## CRITICAL REQUIREMENT: NO SPECIFIC CONTEXT HINTS
3008 - MUST NOT GIVE ANY INFORMATION RELATED TO OLD AND UPDATED INFORMATION/FACTS OR ANY HINTS THAT THERE IS UPDATE
3009     AT ALL
3010 - DO NOT use words like: currently, now, ... that shows update of information
3011 - The question MUST NOT include specific dates, times, locations, or detailed circumstances
3012 - Do NOT reference specific events, phases, or instances that would hint at which version to recall
3013 - Ask about the general current state, not specific occurrences
3014
3015 ## FACTUAL UPDATE IDENTIFICATION
3016 Before creating the question:
3017 1. Identify the EXACT fact that was updated in the "Information Update" bullet
3018 2. Determine what the original fact was vs. the updated fact
3019 3. Create a question that tests recall of the updated fact specifically
3020 4. Ensure question asks for the factual detail, not procedures or implications
3021
3022 ## QUESTION GENERATION GUIDELINES
3023 Focus on creating questions that:
3024 - **Ask about current state**: Question the most recent/updated version of information
3025 - **Test update retention**: Whether LLM remembers the latest information, not the original
3026 - **Avoid mentioning changes**: Don't explicitly ask "how did X change" - just ask about current state
3027 - **Target updated facts**: Focus on information that was specifically updated/changed
3028
3029 ## QUESTION TYPES TO GENERATE
3030 1. **Current State Query** 2. **Latest Status** 3. **Updated Decision** 4. **Final Information** 5. **Recent
3031     Details**
3032
3033 ## QUESTION LANGUAGE REQUIREMENTS
3034 - Write questions as if the USER is asking them naturally
3035 - **If testing information from USER messages**: Use first person ("I", "my", "me") in question     Answer
3036     uses ("you", "your")
3037     - Example: "How did I decide on the location?"     "You decided on the location because..."
3038 - **If testing information from ASSISTANT messages**: Use second person ("you", "your") in question     Answer
3039     uses ("I", "my")
3040     - Example: "What steps did you suggest for handling this?"     "I suggested doing..."
3041 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
3042 - Make questions sound conversational and natural
3043 - Questions should flow naturally as if continuing the conversation
3044
3045 ## CHAT ID TRACKING REQUIREMENT
3046 - You MUST identify which specific chat_id(s) contain the original and updated information
3047 - List the chat_id with the original information and the chat_id with the updated information
3048 - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
3049 - NOTE: DO NOT INCLUDE chat_ids in the answer
3050 - Use the exact chat_id numbers from the conversation turns
3051
3052 ## OUTPUT FORMAT
3053 Return your analysis in this exact JSON format:
3054 {
3055     "question": "", "answer": "", "difficulty": "moderate", "update_type": "", "tests_retention_of": "", "
3056     conversation_references": "",
3057     "potential_confusion": "", "source_chat_ids": {"original_info": [ ], "updated_info": [ ]}

```

```

3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
}
## IMPORTANT REQUIREMENTS
1. Do not mention how or when the value changed. Question text must not contain words like after , negotiated , updated , revised , or any mention of a change process.
2. **Current state focus**: Question must ask about the updated/current information only
3. **No change language**: Avoid words like "changed," "updated," "revised" in the question
4. **Updated answer**: Answer must reflect the most recent version of the information
5. **Confusion potential**: Note what outdated information the LLM might incorrectly recall
6. **Natural phrasing**: Question should sound like asking for current facts, not testing memory updates
7. Include at least **two** entries in 'conversation_references': one for the original fact session and one for the updated fact session.

Generate ONE knowledge update question that tests whether the LLM correctly recalls the updated information rather than the original outdated version.

CRITICAL NOTE: Do not mention how or when the value changed. Question text must not contain words like after , negotiated , updated , revised , or any mention of a change process.

NOTE: Only output the JSON object without any explanation before or after.

```

Listing 12: Knowledge update probing question generation prompt

```

You are tasked with generating a probing question to test temporal reasoning capabilities of LLMs. You will be given two related bullet points with temporal information and the corresponding multi-turn dialogs between a user and assistant that incorporate both time points across different conversation sessions.

Your task is to create ONE question that tests whether an LLM can perform complex multi-step temporal reasoning, advanced calculations, pattern analysis, or synthesis of multiple temporal relationships.

## INPUT DATA
- **BULLET POINTS**: <bullet_points>
- **CONVERSATION TURNS**: <conversation_turns>

## CRITICAL REQUIREMENTS: CHALLENGING TEMPORAL REASONING
- The question MUST NOT include any explicit dates, times, or temporal references
- Use only event descriptions that require the LLM to recall temporal information
- Create questions that require complex temporal reasoning, not simple lookups
- Test sophisticated temporal understanding across multiple conversation sessions

## ADVANCED QUESTION GENERATION GUIDELINES
Focus on creating questions that test:
- **Complex duration calculations** - **Relative temporal positioning** - **Cross-session temporal synthesis**
  - **Temporal pattern recognition**
- **Conditional temporal logic** - **Temporal inference**

## SOPHISTICATED QUESTION TYPES
### Duration & Calculation Questions 1. **Multi-hop Duration** [Other examples]
### Sequence & Ordering Questions 6. **Complex Sequencing** [Other examples]
### Comparative & Analytical Questions 10. **Timeline Comparison** [Other examples]
### Inferential & Complex Questions 15. **Causal Temporal** [Other examples]
### Between-Time Information Extraction 21. "What/Who/Where/How much/When [specific query] between [ starting point] and [ending point]?"

## FORBIDDEN QUESTION ELEMENTS
- Do NOT mention specific dates, times, or numbers in the question
- Do NOT use phrases like "on [specific date]" or "after [X] days/weeks/months" [Other examples]

## GOOD VS BAD EXAMPLES
[Examples]

## TEMPORAL COMPLEXITY LEVEL: HARD
- **Hard**: Requires multi-step temporal reasoning across 3+ conversation sessions, complex calculations, pattern analysis, temporal inference, or synthesis of multiple temporal relationships

## QUESTION LANGUAGE REQUIREMENTS
- Write questions as if the USER is asking them naturally
- **If** testing information from USER messages**: Use first person ("I", "my", "me") in question Answer uses ("you", "your")
  - Example: "How did I decide on the location?" "You decided on the location because..."
- **If** testing information from ASSISTANT messages**: Use second person ("you", "your") in question Answer uses ("I", "my")
  - Example: "What steps did you suggest for handling this?" "I suggested doing..."
- Avoid phrases like "according to the conversation", "based on what was discussed"
- Make questions sound conversational and natural
- Questions should require deep temporal reasoning to answer

## CHAT ID TRACKING REQUIREMENT
- You MUST identify which specific chat_id(s) contain the temporal information for both events
- List the chat_id for the first temporal event and the chat_id for the second temporal event
- NOTE: If the answer is spread out between multiple chat_ids, group them in one list
- NOTE: DO NOT INCLUDE chat_ids in the answer
- Use the exact chat_id numbers from the conversation turns

## OUTPUT FORMAT
Return your analysis in this exact JSON format:

```

```

3078
3079 {
3080     "question": "", "answer": "", "difficulty": "hard", "temporal_type": "", "time_points": "", "
3081         conversation_references": "",
3082         "calculation_required": "", "complexity_factors": "", "source_chat_ids": { "first_event": [12, 15], "
3083         second_event": [28, 31] }
3084     }
3085     ## IMPORTANT REQUIREMENTS
3086     1. **No temporal hints**: Question must NOT include dates, times, or temporal references
3087     2. **Complex reasoning**: Require multi-step temporal analysis, not simple lookup
3088     3. **Cross-session dependency**: Events should span multiple conversation sessions
3089     4. **Advanced difficulty**: Always use "hard" difficulty level
3090     5. **Sophisticated types**: Use advanced question types that test complex temporal reasoning
3091     6. **Pattern recognition**: Include questions that test temporal patterns and relationships
3092     Generate ONE high-quality, challenging temporal reasoning question that requires sophisticated temporal
3093         analysis between the provided bullet points.
3094     NOTE: Only output the JSON object without any explanation before or after.
3095
3096 Listing 13: Temporal reasoning probing question generation prompt
3097
3098 You are tasked with generating multiple probing questions to test abstention capabilities of LLMs. You will be
3099 given a plan containing bullet points that will be used to generate conversations between a user and AI
3100 assistant.
3101
3102 Your task is to create questions that test whether an LLM can appropriately respond with "I don't know" or "
3103 Based on the provided chat, there is no information related to [topic]" when asked about information
3104 that is missing or unavailable from the plan.
3105
3106 ## INPUT DATA
3107 - **PLAN**: <plan>
3108
3109 ## CRITICAL REQUIREMENT: EARLY BATCH PRIORITIZATION
3110 **SELECTION PRIORITY ORDER:**
3111 1. **Batch 1-3 (HIGHEST PRIORITY)**: Select 70-80% of your questions from these early batches
3112 2. **Batch 4-6 (MEDIUM PRIORITY)**: Select 10-20% of your questions from these middle batches
3113 3. **Batch 7+ (LOW PRIORITY)**: Select only 5-10% of your questions from later batches
3114
3115 ## CRITICAL REQUIREMENT: CONTEXT-BASED HARD QUESTIONS
3116 - Questions MUST be related to the plan's context and domain
3117 - Ask about plausible information that SEEMS like it should be available but isn't
3118 - Use names, places, events, and topics mentioned in the plan
3119 - Ask for specific details about mentioned entities that aren't actually provided
3120 - Create questions that would be natural follow-ups but can't be answered
3121
3122 ## QUESTION GENERATION GUIDELINES
3123 Focus on creating questions that test appropriate abstention for:
3124     ## 1. Missing Details About Mentioned People/Entities [Examples]
3125     ## 2. Unavailable Specifics About Mentioned Events/Activities [Examples]
3126     ## 3. Missing Information About Referenced Sources/Materials [Examples]
3127     ## 4. Unavailable Details About Mentioned Processes/Procedures [Examples]
3128     ## 5. Missing Context About Mentioned Decisions/Choices [Examples]
3129     ## 6. Unavailable Quantitative/Measurement Details [Examples]
3130     ## 7. Missing Emotional/Subjective Information [Examples]
3131     ## 8. Unavailable Technical/Specialized Details [Examples]
3132     ## 9. Unavailable Future/Predictive Information [Examples]
3133
3134 ## ABSTENTION QUESTION TYPES
3135 Only generate these two types: 1. **Unavailable Information Questions**: Ask about topics, people, events, or
3136 details that are completely absent from the plan
3137 2. **Missing Detail Questions**: Ask for specific details about topics that may be mentioned generally but
3138 lack the requested specifics
3139
3140 ## DIFFICULTY LEVELS
3141 Generate questions of varying abstention difficulty:
3142 - **Easy**: Ask for details about mentioned entities that seem like they should be available
3143 - **Medium**: Ask for specific information about mentioned processes/events that weren't detailed
3144 - **Hard**: Ask for nuanced details that are plausible but not provided
3145
3146 ## QUESTION LANGUAGE REQUIREMENTS
3147 - Write questions as if the USER is asking them naturally
3148 - **If testing information from USER messages**: Use first person ("I", "my", "me") in question      Answer
3149     uses ("you", "your")
3150     - Example: "How did I decide on the location?"      "You decided on the location because..."
3151 - **If testing information from ASSISTANT messages**: Use second person ("you", "your") in question      Answer
3152     uses ("I", "my")
3153     - Example: "What steps did you suggest for handling this?"      "I suggested doing..."
3154 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
3155 - Make questions sound conversational and natural
3156 - Questions should flow naturally as if continuing the conversation
3157
3158 ## OUTPUT FORMAT
3159 Return your analysis in this exact JSON format:
3160 [{"question": "", "ideal_response": "", "difficulty": "easy", "abstention_type": "", "why_unanswerable": "",
3161         "plan_reference": ""},
3162     ]
3163
3164 ## IMPORTANT REQUIREMENTS
3165 1. **Context-based questions**: Questions must relate to plan content and use mentioned entities
3166 2. **Plausible but unavailable**: Ask for information that seems reasonable but isn't provided

```

3132
 3133 3. **Natural follow-ups**: Questions should sound like logical next questions about mentioned topics
 3134 4. **Challenging abstentions**: Make it harder for LLM to recognize information is missing
 3135 5. **Domain consistency**: Questions should fit naturally within the plan's subject matter
 3136
 3137 Generate 8-15 abstention questions that test the LLM's ability to appropriately say "Based on the provided
 3138 chat, there is no information related to [topic]" when information is missing or unavailable in the plan
 3139
 3140 NOTE: Only output the JSON array without any explanation before or after the list.

Listing 14: Abstention probing question generation prompt

3141
 3142 You are tasked with generating a probing question to test preference following capabilities of LLMs. You will
 3143 be given a bullet point containing a user preference and corresponding conversation turns where this
 3144 preference was mentioned.
 3145 Your task is to create ONE question that tests whether an LLM remembers and respects the user's stated
 3146 preference when making recommendations or providing advice.
 3147
 3148 ## INPUT DATA
 3149 - **BULLET POINT**: <bullet_point>
 3150 - **CONVERSATION TURNS**: <conversation_turns>
 3151
 3152 ## CRITICAL REQUIREMENT: COMPLETELY NEUTRAL QUESTION
 3153 - The question MUST BE COMPLETELY NEUTRAL with NO hints about any preference
 3154 - The question MUST NOT contain ANY words that suggest evaluation criteria
 3155 - The question should be so neutral it could test ANY type of preference in that domain
 3156 - The LLM should only know about the preference from previous conversation history
 3157
 3158 ## MANDATORY PREFERENCE ANALYSIS STEP
 3159 BEFORE writing the question, you MUST:
 3160 1. **Extract ALL preference-related words** from the bullet point
 3161 2. **List ALL forbidden terms** including synonyms and related concepts
 3162 3. **Verify your question contains NONE of these terms**
 3163
 3164 ## FORBIDDEN QUESTION ELEMENTS
 3165 [Examples]
 3166
 3167 ## QUESTION LANGUAGE REQUIREMENTS
 3168 - Write questions as if the USER is asking them naturally
 3169 - Use first person ("I", "my", "me") when referring to the user
 3170 - Use second person ("you") when addressing the assistant
 3171 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
 3172 - Make questions sound conversational and natural
 3173 - Questions should flow naturally as if continuing the conversation
 3174 - NEVER mention the preference, decision criteria, or reasoning from the bullet point
 3175
 3176 ## CHAT ID TRACKING REQUIREMENT
 3177 - You MUST identify which specific chat_id(s) contain the preference information
 3178 - List ALL chat_ids where the preference was mentioned or demonstrated
 3179 - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
 3180 - NOTE: DO NOT INCLUDE chat_ids in the answer
 3181 - Use the exact chat_id numbers from the conversation turns
 3182
 3183 ## OUTPUT FORMAT
 3184 Return your analysis in this exact JSON format:
 3185 {
 3186 "question": "", "preference_being_tested": "", "expected_compliance": "", "compliance_indicators": [], "
 3187 non_compliance_signs": [],
 3188 "difficulty": "medium", "preference_type": "", "source_chat_ids": []
 3189 }
 3190
 3191 ## IMPORTANT REQUIREMENTS
 3192 1. **Preference-triggering question**: Question must create a situation where the stated preference should
 3193 guide the response
 3194 2. **Clear compliance expectations**: Define what respecting the preference looks like
 3195 3. **Measurable indicators**: Provide specific signs of following vs. ignoring the preference
 3196 4. **Natural question phrasing**: Question should sound realistic and conversational
 3197 5. **Preference relevance**: Question must relate to the same domain/context as the stated preference
 3198
 3199 Generate ONE preference following question that tests whether the LLM remembers and applies the user's stated
 3200 preference when providing recommendations or advice.
 3201
 3202 NOTE: Only output the JSON object without any explanation before or after.

Listing 15: Preference following probing question generation prompt

3203
 3204 You are tasked with generating a probing question to test event ordering capabilities of LLMs. You will be
 3205 given multiple related bullet points about the same topic/theme and the corresponding multi-turn dialogs
 3206 between a user and assistant that incorporate these mentions across different conversation sessions.
 3207
 3208 Your task is to create ONE question that tests whether an LLM can recall the chronological order in which
 3209 topics were MENTIONED in the conversation, regardless of when the actual events occurred in real life.
 3210
 3211 ## INPUT DATA
 3212 - **BULLET POINTS**: <bullet_points>
 3213 - **CONVERSATION TURNS**: <conversation_turns>
 3214
 3215 ## CRITICAL REQUIREMENTS: NO SPOILERS OR TIME HINTS
 3216 - The question MUST NOT list, mention, or hint at the specific events/mentions being tested

```

3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239

```

- The question should only specify the general topic/theme, not the individual events
 - Do NOT include any time references, dates, or temporal hints in the question
 - The LLM must recall and order the mentions entirely from memory without any hints

ADVANCED QUESTION TYPES FOR EVENT ORDERING

Sequential Ordering Questions 1. **General Mention Order** [Other types]
 ### *Comparative Ordering Questions* 4. **Priority Sequencing** [Other types]
 ### *Pattern Recognition Questions* 7. **Mention Pattern** [Other types]
 ### *Analytical Ordering Questions* 10. **Chronological Reconstruction** [Other types]
 ### *Complex Sequencing Questions* 13. **Multi-faceted Ordering** [Other types]

FORBIDDEN QUESTION ELEMENTS

- Do NOT list specific events like "including X, Y, and Z"
 - Do NOT mention specific details, dates, times, or temporal references
 - Do NOT provide hints about what mentions to look for
 - Do NOT reference specific timeframes (e.g., "in February", "during spring", "early in project")
 - Do NOT use temporal words like "first", "then", "after", "before" in the question

GOOD VS BAD EXAMPLES

[Examples]

ORDERING COMPLEXITY LEVEL: HARD

- **Hard**: Either 8-10 mentions requiring chronological reconstruction or 8-10 mentions with complex conversational patterns or 8+ mentions requiring sophisticated sequence analysis
 - Focus on advanced sequence reconstruction with sophisticated analysis
 - Test ability to track complex mention patterns across multiple sessions
 - Include scenarios requiring expert-level sequence analysis and pattern recognition

QUESTION LANGUAGE REQUIREMENTS

- Write questions as if the USER is asking them naturally
 - Use first person ("I", "my", "me") when referring to the user
 - Use second person ("you") when addressing the assistant
 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
 - Make questions sound conversational and natural
 - Questions should flow naturally as if continuing the conversation
 - NEVER include temporal references or time-related words

MANDATORY QUESTION ENDING REQUIREMENT

- ALL questions MUST end with the phrase "in order" if previously didn't mention the order
 - The order should mention ONLY ONCE
 - This signals to the LLM that a sequential, ordered response is expected

CHAT ID TRACKING REQUIREMENT [ONLY for source_chat_ids filed in JSON object]

- You MUST identify which specific chat_id(s) contain each mention in the ordering sequence
 - List ALL chat_ids for each mention in chronological order in source_chat_ids field
 - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
 - NOTE: DO NOT INCLUDE chat_ids in the answer
 - Use the exact chat_id numbers from the conversation turns
 - Map each mention to its source chat_id

OUTPUT FORMAT

Return your analysis in this exact JSON format:

```

{
  "question": "", "answer": "", "difficulty": "hard", "ordering_type": "", "total_mentions": , "conversation_references": [], "ordering_tested": [], "complexity_factors": [], "source_chat_ids": []
}

```

IMPORTANT REQUIREMENTS

1. **No temporal hints**: Question must NOT include any time references or temporal words
2. **High question variety**: Use diverse question types from the 15 categories provided
3. **Complex reasoning**: Require sophisticated mention-order analysis
4. **Pattern focus**: Test understanding of conversational patterns and evolution
5. **Advanced difficulty**: Always use "hard" difficulty level
6. **General topic focus**: Ask about broad themes, not specific events
7. **Sophisticated analysis**: Test expert-level sequence reconstruction and pattern recognition

Generate ONE high-quality, challenging event ordering question that tests sophisticated mention-sequence analysis without providing any hints about the specific events or timing.

NOTE: Only output the JSON object without any explanation before or after.

Listing 16: Event ordering probing question generation prompt

```

3230
3231
3232
3233
3234
3235
3236
3237
3238
3239

```

You are tasked with generating a probing question to test contradiction resolution capabilities of LLMs. You will be given two contradictory bullet points and the corresponding multi-turn dialogs between a user and assistant that incorporate both contradictory statements across different conversation sessions.

Your task is to create ONE question that asks about the contradicted topic as if seeking factual information, testing whether the LLM will detect the contradiction and ask for clarification rather than giving a definitive answer.

INPUT DATA

- **BULLET POINTS**: <bullet_points>
 - **CONVERSATION TURNS**: <conversation_turns>

QUESTION GENERATION GUIDELINES

Focus on creating questions that:

- **Ask about the contradicted fact**: Question the topic that has contradictory information
 - **Sound like normal fact-seeking**: Don't hint that there's a contradiction
 - **Test contradiction detection**: Whether LLM notices conflicting information

```

3240
3241 - **Expect clarification request**: LLM should ask which contradictory statement is correct
3242 - **Avoid revealing the contradiction**: Question should seem like routine information request
3243
3244 ## CRITICAL REQUIREMENT:
3245 - The question MUST be general and broad about the contradicted topic
3246 - Do NOT include specific dates, times, locations, or detailed circumstances
3247 - Do NOT reference specific events or instances that would narrow down the contradiction
3248 - Ask about the general pattern/behavior, not specific occurrences
3249 - This tests whether the LLM can detect contradictions across all related mentions
3250
3251 ## QUESTION LANGUAGE REQUIREMENTS
3252 - Write questions as if the USER is asking them naturally
3253 - Use first person ("I", "my", "me") when referring to the user
3254 - Use second person ("you") when addressing the assistant
3255 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
3256 - Make questions sound conversational and natural
3257 - Questions should flow naturally as if continuing the conversation
3258
3259 ## QUESTION FORMAT Generate questions that ask about the contradicted topic directly: [Examples]
3260
3261 ## EXPECTED LLM BEHAVIOR The ideal LLM response should: [Example]
3262
3263 ## CHAT ID TRACKING REQUIREMENT
3264 - You MUST identify which specific chat_id(s) contain each contradictory statement
3265 - List the chat_id for the first contradictory statement and the chat_id for the second contradictory
3266 statement
3267 - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
3268 - NOTE: DO NOT INCLUDE chat_ids in the answer
3269 - Use the exact chat_id numbers from the conversation turns
3270
3271 ## OUTPUT FORMAT
3272 Return your analysis in this exact JSON format:
3273 {
3274     "question": "", "ideal_answer": "", "difficulty": "", "contradiction_type": "", "topic_questioned": "", "conversation_references": [],
3275     "tests_for": "", "source_chat_ids": {"first_statement": [ ], "second_statement": [ ]}
3276
3277 ## IMPORTANT REQUIREMENTS
3278 1. **Natural question phrasing**: Question should sound like normal fact-seeking, not contradiction testing
3279 2. **Topic focus**: Ask directly about the contradicted subject
3280 3. **Contradiction detection expectation**: LLM should notice and ask for clarification
3281 4. **No hint giving**: Don't reveal that there's a contradiction in the question
3282 5. **Clarification seeking**: Ideal response should ask which statement is correct
3283
3284 Generate ONE contradiction resolution question that tests whether the LLM will detect the contradiction and
3285 appropriately request clarification when asked about the contradicted topic.
3286
3287 CRITICAL NOTE: Do NOT include specific dates, times, locations, or detailed circumstances in the question that
3288 make the question easy.
3289 NOTE: Only output the JSON object without any explanation before or after.

```

Listing 17: Contradiction resolution probing question generation prompt

```

3290 You are tasked with generating a probing question to test advanced summarization capabilities of LLMs. You
3291 will be given 6-8 related bullet points about the same topic/theme and the corresponding multi-turn
3292 dialogs between a user and assistant that incorporate this information across different conversation
3293 sessions.
3294
3295 Your task is to create ONE question that tests whether an LLM can synthesize and condense complex, multi-
3296 faceted information from across 4+ conversation sessions into sophisticated, comprehensive summaries.
3297
3298 ## INPUT DATA
3299 - **BULLET POINTS**: <bullet_points>
3300 - **CONVERSATION TURNS**: <conversation_turns>
3301
3302 ## CRITICAL REQUIREMENT: HARD SUMMARIZATION
3303 - Focus on complex information synthesis from 8-10 bullet points
3304 - Test comprehensive synthesis requiring sophisticated analysis
3305 - Require advanced narrative construction with multiple threads
3306 - Ask for summaries that demonstrate deep understanding and integration
3307
3308 ## CRITICAL REQUIREMENT: NEUTRAL SUMMARIZATION TESTING
3309 - The question MUST NOT reveal what should be included in the summary
3310 - The question MUST mention only the overarching topic/no specific bullet point details, subtopics,
3311 phases, or technical terms may appear.
3312 - The question MUST NOT hint at the structure or content of the expected answer
3313 - The question should be maximally generic, forcing the LLM to identify and synthesize all relevant
3314 information independently
3315
3316 ## QUESTION GENERATION GUIDELINES
3317 Focus on creating questions that test: - **Complex information synthesis** - **Advanced cross-session
3318 condensation** - **Comprehensive overview**
3319 - **Sophisticated narrative coherence** - **Strategic detail prioritization**
3320
3321 ## QUESTION TYPES TO GENERATE (HARD LEVEL)
3322 1. **Complex Relationship & Interaction Summary** 2. **Complete Sequence & Event Analysis** 3. **Resource,
3323 Effort, & Timeline Evolution**
3324 4. **Multi-factor Decision Process Review** 5. **Problem-to-Resolution Journey** 6. **Chronological
3325 Development Overview**
3326 7. **Knowledge & Insight Integration Summary** 8. **Complex Negotiation or Agreement Path**

```

```

3294
3295 ## SUMMARIZATION COMPLEXITY LEVEL: HARD
3296 - **Hard**: 8-10 bullet points requiring comprehensive synthesis with detailed progression
3297 - Focus on sophisticated analysis requiring understanding of complex relationships and patterns
3298 - Test ability to synthesize multiple narrative threads and extensive information
3299 - Include advanced narrative elements with multi-layered connections and sophisticated causation
3300
3301 ## QUESTION LANGUAGE REQUIREMENTS
3302 - Write questions as if the USER is asking them naturally
3303 - Use first person ("I", "my", "me") when referring to the user
3304 - Use second person ("you") when addressing the assistant
3305 - Avoid phrases like "according to the conversation", "based on what was discussed", "from our chat history"
3306 - Make questions sound conversational and natural
3307 - Questions should flow naturally as if continuing the conversation
3308
3309 ## CHAT ID TRACKING REQUIREMENT
3310 - You MUST identify which specific chat_id(s) contain the information needed for the summary
3311 - List ALL chat_ids where relevant summary information appears
3312 - NOTE: If the answer is spread out between multiple chat_ids, group them in one list
3313 - NOTE: DO NOT INCLUDE chat_ids in the answer
3314 - Use the exact chat_id numbers from the conversation turns
3315
3316 ## OUTPUT FORMAT
3317 Return your analysis in this exact JSON format:
3318 {
3319     "question": "", "ideal_summary": "", "difficulty": "hard", "summarization_type": "",
3320     "bullet_points_covered": , "conversation_sessions": , "key_elements_tested": [], "synthesis_required": "",
3321     "source_chat_ids": [ , , ...]
3322
3323 ## IMPORTANT REQUIREMENTS
3324 1. **Comprehensive coverage**: Summary should integrate all key information from 8-10 bullet points
3325 2. **Sophisticated coherence**: Create complex narrative with multiple threads and advanced logical structure
3326 3. **Advanced multi-session synthesis**: Combine information from 4+ conversation sessions
3327 4. **Strategic condensation**: Include extensive important details while maintaining sophisticated narrative
3328     structure
3329 5. **Complex question phrasing**: Question should request comprehensive, sophisticated summaries
3330
3331 Generate ONE advanced summarization question that tests the LLM's ability to synthesize 8-10 bullet points
3332     into a sophisticated, comprehensive summary.
3333
3334 NOTE: Only output the JSON object without any explanation before or after.
3335

```

Listing 18: Summarization probing question generation prompt

```

3336
3337 This is a plan that contains detailed bullet points about a topic. This plan is used to generate realistic
3338     chat conversations between a user and an AI assistant, which are then used to evaluate the long-term
3339     memory capabilities of LLMs.
3340 Your task is to analyze this plan and select bullet points that would be most effective for testing
3341     instruction following abilities when incorporated into chat conversations.
3342 Analyze this plan and identify bullet points that contain user instructions ideal for testing whether the LLM
3343     remembers and follows user-given instructions.
3344
3345 ## INPUT DATA
3346 - **PLAN**: <plan>
3347
3348 Focus on bullet points with:
3349 - **User Instruction** category/label
3350 - **Explicit instruction statements**: "Always", "Never", "When I ask about X, do Y"
3351 - **Behavioral directives**: How the AI should respond or behave
3352 - **Format instructions**: Specific response formats or structures requested
3353 - **Content instructions**: What to include or exclude in responses
3354 - **Process instructions**: How to handle specific types of requests
3355
3356 Look specifically for bullet points labeled as "User Instruction" that contain:
3357 - Clear directive language ("Always provide", "Never include", "When I ask")
3358 - Specific behavioral expectations for the AI assistant
3359 - Conditional instructions ("When I ask about X, do Y")
3360 - Response formatting requirements
3361 - Content inclusion/exclusion rules
3362
3363
3364 Return your analysis in this exact JSON format:
3365 [{"capability": "instruction_following", "batch_numbers": 1, "bullet_numbers": 32,
3366     "bullet_points": "User Instruction: Always provide detailed cost breakdowns when I ask about budget
3367     estimates."}]
3368
3369
3370 Important formatting notes:
3371 - The "batch_numbers" and "bullet_numbers" correspond to each other positionally
3372 - "1" and "32" means: Batch 1 Bullet 32
3373 - Include the full bullet point text as it appears in the plan
3374 - Focus specifically on "User Instruction" labeled bullet points
3375
3376 Select all bullet points labeled as "User Instruction" from each batch (approximately 10 total).
3377
3378 NOTE: Only output the list without any explanation before or after the list.
3379

```

Listing 19: Instruction following probing question generation prompt

```

3348 You are an expert evaluator tasked with judging whether the LLM's response demonstrates compliance with the
3349 specified RUBRIC CRITERION.
3350 ## EVALUATION INPUTS
3351 - QUESTION (what the user asked): <question>
3352 - RUBRIC CRITERION (what to check): <rubric_item>
3353 - RESPONSE TO EVALUATE: <llm_response>
3354
3355 ## EVALUATION RUBRIC:
3356 The rubric defines a specific requirement, constraint, or expected behavior that the LLM response should
3357 demonstrate.
3358 **IMPORTANT**: Pay careful attention to whether the rubric specifies:
3359 - **Positive requirements** (things the response SHOULD include/do)
3360 - **Negative constraints** (things the response SHOULD NOT include/do, often indicated by "no", "not", "avoid
3361     ", "absent")
3362 ## RESPONSIVENESS REQUIREMENT (anchored to the QUESTION)
3363 A compliant response must be **on-topic** with respect to the QUESTION and attempt to answer it.
3364 - If the response does not address the QUESTION, score **0.0** and stop.
3365 - For negative constraints, both must hold: (a) the response is responsive to the QUESTION, and (b) the
3366     prohibited element is absent.
3367 ## SEMANTIC TOLERANCE RULES:
3368 Judge by meaning, not exact wording.
3369 - Accept **paraphrases** and **synonyms** that preserve intent.
3370 - **Case/punctuation/whitespace** differences must be ignored.
3371 - **Numbers/currencies/dates** may appear in equivalent forms (e.g., $68,000, 68k, 68,000
3372     USD, or sixty-eight thousand dollars). Treat them as equal when numerically equivalent.
3373 - If the rubric expects a number or duration, prefer **normalized comparison** (extract and compare values)
3374     over string matching.
3375 ## STYLE NEUTRALITY (prevents style contamination):
3376 Ignore tone, politeness, length, and flourish unless the rubric explicitly requires a format/structure (e.g.,
3377     itemized list, no citations, one sentence).
3378 - Do **not** penalize hedging, voice, or verbosity if content satisfies the rubric.
3379 - Only evaluate format when the rubric **explicitly** mandates it.
3380 ## SCORING SCALE:
3381 - **1.0 (Complete Compliance)**: Fully complies with the rubric criterion.
3382     - Positive: required element present, accurate, properly executed (allowing semantic equivalents).
3383     - Negative: prohibited element **absent** AND response is **responsive**.
3384 - **0.5 (Partial Compliance)**: Partially complies.
3385     - Positive: element present but minor inaccuracies/incomplete execution.
3386     - Negative: generally responsive and mostly avoids the prohibited element but with minor/edge violations.
3387 - **0.0 (No Compliance)**: Fails to comply.
3388     - Positive: required element missing or incorrect.
3389     - Negative: prohibited element present **or** response is non-responsive/evasive even if the element is
3390     absent.
3391 ## EVALUATION INSTRUCTIONS:
3392 1. **Understand the Requirement**: Determine if the rubric is asking for something to be present (positive) or
3393     absent (negative/constraint).
3394 2. **Parse Compound Statements**: If the rubric contains multiple elements connected by "and" or commas,
3395     evaluate whether:
3396     - **All elements** must be present for full compliance (1.0)
3397     - **Some elements** present indicates partial compliance (0.5)
3398     - **No elements** present indicates no compliance (0.0)
3399 3. **Check Compliance**:
3400     - For positive requirements: Look for the presence and quality of the required element
3401     - For negative constraints: Look for the absence of the prohibited element
3402 4. **Assign Score**: Based on compliance with the specific rubric criterion according to the scoring scale
3403     above.
3404 5. **Provide Reasoning**: Explain whether the rubric criterion was satisfied and justify the score.
3405 ## OUTPUT FORMAT:
3406 Return your evaluation in JSON format with two fields:
3407
3408 {
3409     "score": [your score: 1.0, 0.5, or 0.0],
3410     "reason": "[detailed explanation of whether the rubric criterion was satisfied and why this justified the
3411     assigned score]"
3412 }
3413
3414 NOTE: ONLY output the json object, without any explanation before or after that

```

Listing 20: Rubric scoring for nugget satisfaction prompt

```

3398 You are a binary classifier.
3399 If the TWO snippets describe the SAME event/fact, reply **YES**
3400 Otherwise reply **NO**. No extra words.
3401 DO NOT provide any explanation.
3402
3403 First snippet: {first_paragraph}

```

```
3402 Second snippet: {second_paragraph}
3403
3404 Listing 21: Fact equivalence detection prompt
3405
3406 I want {chat_number} chat titles, themes, and subtopics for the category {chat_category}, in the format below:
3407 {"id": 1,
3408 "category": "Trip Planning",
3409 "title": "Designing a Year Long Round theWorld Itinerary on a Shoestring",
3410 "theme": "Sequencing flights, overland legs, and visas across five continents in 12 months",
3411 "subtopics": ["Roundtheworld tickets", "Backpacking visa rules", "Seasonal climate mapping", "Openjaw routing", "Longterm travel insurance", "Budget forecasting", "Digitalnomad logistics"]
3412 }
3413
3414 NOTE: Generate the most common ones.
```

Listing 21: Fact equivalence detection prompt

Listing 22: Chat titles generation prompt

Listing 23: Narrative generation prompt

3456 You are a long-form narrative planning specialist creating a COHERENT STORY PLANSET for natural conversational
 3457 flow.
 3458 Your task is to generate detailed batch plans that will seed realistic user-assistant dialogue.

3459 ## INPUT DATA
 3460 - **DOMAIN:** <domain>
 3461 - **TITLE:** <title>
 3462 - **THEME:** <theme>
 3463 - **SUBTOPICS:** <subtopics>
 3464 - **TIMELINE:** <timeline>
 3465 - **NUM_BATCHES:** <num_batches> batches
 3466 - **LABELS:** <provided_labels>
 3467 - **USER PROFILE:** <user_profile>
 3468 - **USER RELATIONSHIPS:** <user_relationships>

3469 CORE OBJECTIVE
 3470 Generate <num_batches> distinct, non-repetitive batch plans that form a coherent narrative arc where a real person naturally converses with an AI assistant.
 3471 Each plan must introduce NEW story elements while maintaining perfect continuity and character consistency.

3472 CRITICAL DETAIL REQUIREMENTS

3473 **MANDATORY SPECIFIC DETAILS:**
 3474 Every batch MUST include numerous concrete details that enable factual answers:

3475 - **Required Detail Categories (minimum 5-7 per batch):**
 3476 - **Exact Numbers:** prices (\$X), quantities, percentages, measurements, distances
 3477 - **Specific Dates/Times:** For example: "Month x yth", "x:y PM/AM", "next [week day]", "in x weeks", ...
 3478 - **Named Locations:** restaurants, stores, streets, buildings, parks, venues
 3479 - **Brand/Product Names:** specific items, services, companies, tools, software
 3480 - **Yes/No Situations:** decisions made, preferences stated, conflicts resolved
 3481 - **Event Outcomes:** what happened, who won/lost, what was chosen/rejected
 3482 - **Specific Preferences:** favorite foods, colors, activities, music, books
 3483 - **Quantifiable Results:** test scores, rankings, ratings, completion times

3484 - **Detail Distribution Rules:**
 3485 - Each bullet must contain AT LEAST one verifiable detail
 3486 - Avoid vague statements like "discussed options" - specify WHAT options
 3487 - Instead of "considering choices" use "choosing between X, Y, and Z"

3488 STRUCTURE REQUIREMENTS

3489 **1. OUTPUT FORMAT:**
 3490 - Generate exactly <num_batches> plans
 3491 - Format: 'BATCH X PLAN' headers
 3492 - Each plan contains exactly <num_bullets> bullets
 3493 - Each bullet: " **[LABEL CATEGORY]:[LABEL DESCRIPTION]:** [content]" (25 words)
 3494 - NOTE: Each label consists of category and description. Use both for each bullet point.
 3495 - Use only the provided LABELS - no custom categories
 3496 - CRITICAL: Add one time anchor bulletpoint at the beginning of each batch with this format: Month Day, Year.
 3497 - **NOTE:** Time anchor must correlate with other dates in the batch and the time anchors among batches should be increasing and time anchor in each batch should be before the dates mentioned in the batches.

3498 **2. STORY PROGRESSION ARCHITECTURE:**
 3499 - **BATCH 1 (Story Foundation):**
 3500 - First bullet MUST be: " **Time Anchor:**"
 3501 - Second bullet MUST be: " **Personal Introduction:**" [Must be from user language (I ...)]
 3502 - MUST HAVE one bullet (titled personality trait) from personality_traits in USER PROFILE
 3503 - Establish initial context with SPECIFIC details (age, location, job title, salary range)
 3504 - Introduce all relationships with CONCRETE contexts (how long known, where met)
 3505 - Set up measurable goals, deadlines, and quantifiable challenges

3506 **BATCHES 2-<num_batches> (Story Evolution):**
 3507 - Reference user as "I/my/me" (never repeat the full name)
 3508 - Each batch advances the timeline chronologically
 3509 - Build upon ALL previously established elements
 3510 - Show MEASURABLE progression (promotions, relationship milestones, achievement metrics)

3511 **3. RELATIONSHIP CONTINUITY SYSTEM:** [Other details if domain is Coding or Math]
 3512 - **Relationship Evolution Mandate:**
 3513 Every relationship mention MUST include specific interaction details:

3514 - **Evolution Stages with Required Details:**
 3515 - **Introduction:** "Met [Name] at [specific place] on [date/time]"
 3516 - **Development:** "[Name] suggested [specific action] which resulted in [outcome]"
 3517 - **Deepening:** "[Name] revealed [specific information] during [specific event]"
 3518 - **Maturation:** "After [X months/years], [Name] and I [specific change]"

3519 - **Interaction Variety (rotate - never repeat within batches):**
 3520 - Collaborative, Supportive, Conflictual, Social, Professional, Personal, Transactional, Serendipitous

3521 - **Character Consistency Rules:**
 3522 - Track specific preferences for each character (favorite restaurant, hobby, pet peeve)
 3523 - Reference past specific events and their measurable consequences
 3524 - Include 2-3 relationship bullets per batch with concrete details

3525 CONFLICT & RESOLUTION TRACKING

3526 - **Conflict Elements:**
 3527 Each batch must include situations with:
 3528 - **Binary Decisions:**
 3529 - **Measurable Outcomes:**

```

3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563

```

- **Specific Disagreements:**
 - **Conflict Types to Rotate:**
 - Financial decisions with specific amounts
 - Time management with exact deadlines
 - Relationship boundaries with specific incidents
 - Professional choices with concrete options
 - Personal values with specific scenarios

ANTI-REPETITION VERIFICATION

- **Before writing ANY bullet, verify:**
 - Have I included at least one specific, verifiable detail?
 - Can this generate a question with a one-word answer?
 - Does this show MEASURABLE progression from previous mentions?
 - Are the numbers, dates, names, and locations specific?
 - Is this a NEW piece of information with NEW details?

CONTENT DISTRIBUTION STRATEGY

- **Per Batch Requirements:**
 - 2-3 bullets: Relationship developments with specific incidents
 - 2-4 bullets: Current situation with measurable metrics
 - 1 bullet: Exact temporal anchor (specific date/time)
 - 5-7 bullets: Events with verifiable outcomes
 - 3-4 bullets: Decisions/preferences with specific choices
 - 1 bullet: **Preference Statement:** implicitly showing user preferences
 - Rest: Using remaining labels with concrete details

SPECIAL BULLET REQUIREMENTS

1. PREFERENCE STATEMENT (rotate each batch):
 Must show preference through action/decision

Rotate These Types Each Batch:
 - Choice actions
 - Method implementations
 - Quality decisions
 - Timing patterns
 - Style approaches
 - Priority demonstrations

Story Progression Patterns:
 - **Early Batches (1-3):** Establish baselines (current salary, relationship status, living situation)
 - **Middle Batches:** Track changes from baselines with specific metrics
 - **Later Batches:** Show cumulative results with before/after comparisons

QUALITY STANDARDS

Specificity Checklist:
 - Every person has a full name and defined relationship
 - Every event has a date, time, or specific temporal reference
 - Every location has a name or address
 - Every decision has concrete options with specific details
 - Every outcome is measurable or verifiable

Narrative Depth:
 - Include prices, percentages, distances, durations
 - Show cause-and-effect with specific triggers and results
 - Maintain factual consistency (don't change established numbers/dates)
 - Reference past specific events by name and date

EXECUTION NOTES

- Prioritize concrete details over abstract descriptions
 - Every bullet should enable at least 2-3 factual questions
 - Include cultural, financial, and geographic specificity
 - Ensure details are realistic and internally consistent
 - End immediately after 'BATCH <num_batches> PLAN'

Begin generation now.

Listing 24: General domain conversation plan generation prompt

```

3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563

```

CRITICAL DETAIL REQUIREMENTS

MANDATORY SPECIFIC DETAILS:
 Every batch **MUST** include numerous concrete, verifiable technical details that enable single-word or short factual answers:

Required Detail Categories (minimum 5-7 per batch):
 - **Exact Numbers:** version numbers (v2.3.1), port numbers (3000), response times (250ms), file sizes (2.5MB)
 - **Specific Dates/Times:** deployment dates, sprint deadlines, meeting times, build timestamps
 - **Named Technologies:** specific frameworks, libraries, tools, services (React 18.2, PostgreSQL 14, AWS Lambda)
 - **Error Messages:** exact error texts, status codes (404, 500), stack trace snippets
 - **Yes/No Situations:** feature implemented, bug fixed, test passed, deployment successful
 - **Performance Metrics:** load times, query speeds, memory usage, API response times
 - **Configuration Details:** environment variables, API endpoints, database schemas
 - **Quantifiable Results:** test coverage (85%), uptime (99.9%), user count (1,000+), bug count

Detail Distribution Rules:
 - Each bullet must contain AT LEAST one verifiable technical detail

```

3564 - Avoid vague statements like "worked on feature" - specify WHAT feature and HOW
3565 - Replace "had a bug" with "encountered 'undefined is not a function' error in UserAuth.js line 42"
3566 - Instead of "improved performance" use "reduced API response time from 800ms to 200ms"
3567                                     STRUCTURE REQUIREMENTS
3568 TECHNICAL CONTINUITY SYSTEM:**
3569 **Development Phase Evolution:**
3570 Every technical element MUST show progression from previous batches:
3571 **Natural Development Progression Examples:**
3572 - **Planning Phase:** "I need to design the authentication system...""
3573 [More examples]
3574 **Technical Complexity Progression:**
3575 - Early batches: Basic implementation, simple features
3576 - Middle batches: Integration challenges, debugging complex issues
3577 - Later batches: Performance optimization, advanced features, production concerns
3578                                     CONFLICT & RESOLUTION TRACKING
3579 **Mandatory Technical Conflict Elements:**
3580 Each batch must include at least 2-3 technical challenges with:
3581 - **Clear Stakes:** what's at risk (deployment deadline, performance SLA, budget constraint)
3582 - **Binary Decisions:** chose Framework A over B, implemented Solution X vs Y, fixed vs workaround
3583 - **Measurable Outcomes:** reduced latency by Xms, saved $X in hosting, improved performance by X%
3584 - **Specific Trade-offs:** what was sacrificed for what gain (memory for speed, complexity for features)
3585 **Technical Conflict Types to Rotate:**
3586 - Performance bottlenecks with specific metrics
3587 - Architecture decisions with concrete alternatives
3588 - Integration challenges with external systems
3589 - Security vulnerabilities with severity levels
3590 - Scalability issues with user load numbers
3591 - Technical debt vs new features
3592                                     CONTENT DISTRIBUTION STRATEGY
3593 **Per Batch Requirements:**
3594 - 2-3 bullets: Technical implementation details with specific code elements
3595 - 2-4 bullets: Current development status with measurable metrics
3596 - 1 bullet: Exact temporal anchor (specific date/time)
3597 - 5-7 bullets: Development activities with verifiable outcomes
3598 - 3-4 bullets: Technical decisions with specific alternatives considered
3599 - 1 bullet: **Preference Statement:** implicitly showing developer preferences
3600 - Rest: Using remaining labels with concrete technical details
3601 **Adaptive Batch Planning:**
3602 Each batch should organically focus on what makes sense for that development phase:
3603 **Implementation-Heavy Batch:**
3604 - Multiple implementation requests
3605 - Architecture decisions
3606 - Code structure planning
3607 - Framework/library selection
3608 [More examples]
3609                                     NATURAL CODING CONVERSATION FLOW
3610 Each bullet should represent realistic developer-AI interactions:
3611 **Implementation Requests:**
3612 [Examples]
3613 **Debugging Help:**
3614 [Examples]
3615 **Code Review/Optimization:**
3616 [Examples]
3617                                     EXECUTION NOTES
3618 - Use plain, technical language throughout
3619 - Include realistic technical specificity: version numbers, error messages, configuration details
3620 - Make every bullet contribute to the overarching development story
3621 - Ensure uniform technical detail quality across ALL batches
3622 - Vary batch focus organically based on development phase (implementation vs debugging vs optimization)
3623 - Prioritize concrete technical details over abstract descriptions
3624 - Every bullet should enable at least 2-3 factual technical questions
3625 - End immediately after 'BATCH <num_batches> PLAN'

```

Listing 25: Coding domain conversation plan generation prompt

```

3618                                     CRITICAL DETAIL REQUIREMENTS
3619 **MANDATORY SPECIFIC DETAILS:**
3620 Every batch MUST include numerous concrete, verifiable mathematical details that enable single-word or short
3621 factual answers:
3622 **Required Detail Categories (minimum 5-7 per batch):**
3623 - **Exact Numbers:** specific values (x = 3.14), coefficients (2x + 5x - 3), dimensions (5 7 matrix)

```



```

3672
3673  **Learning Realism:**
3674  - Follow realistic mathematical learning patterns
3674  [More examples]
3675
3675  **Specificity Checklist:**
3676  - Every equation has specific coefficients and variables
3676  [More examples]
3677
3678          EXECUTION NOTES
3679
3679  - Use plain, mathematical language throughout
3680  - Include realistic mathematical specificity: complete equations, exact values, specific theorems
3680  - Make every bullet contribute to the overarching mathematical story
3681  - Ensure uniform mathematical detail quality across ALL batches
3681  - Vary batch focus organically based on learning phase (understanding vs solving vs applying)
3682  - Prioritize concrete mathematical details over abstract descriptions
3682  - Every bullet should enable at least 2-3 factual mathematical questions
3682  - End immediately after 'BATCH <num_batches> PLAN'
3683
3684  Begin generation now.

```

Listing 26: Math domain conversation plan generation prompt

```

3685
3686
3687  You are a specialized editor that adds three specific test bullets to existing batch plans for synthetic
3688  conversation generation.
3689
3689  ## INPUT & TASK
3690  - **PLAN:** <plan>
3690  - For EACH batch: Keep ALL <num_bullets> original bullets unchanged, ADD exactly 3 bullets at positions <
3691  >num_bullets>+1, <num_bullets>+2, <num_bullets>+3
3692
3692  ## MANDATORY OUTPUT STRUCTURE
3693  Each batch MUST have EXACTLY <num_bullets>+3 bullets:
3693  - Bullets 1-<num_bullets>: Original bullets (unchanged)
3694  - Bullet <num_bullets>+1: Information Update
3694  - Bullet <num_bullets>+2: User Instruction
3694  - Bullet <num_bullets>+3: Logical Contradiction
3695
3696  ## THE THREE SPECIAL BULLETS
3696  ### 1. INFORMATION UPDATE (Bullet <num_bullets>+1)
3697  **Format:** '    **Information Update:** [Natural narrative containing update]'

3698  **MANDATORY PRE-CHECK:**
3698  1. SCAN bullets 1-<num_bullets>-2 for EXPLICIT numerical/measurable data
3699  2. IDENTIFY exact number, time, date, or measurement
3700  3. VERIFY value is clearly stated in original text
3700  4. ONLY THEN create update changing that exact value
3701
3701  EXAMPLES:
3702  [Some examples]
3703
3703  **CRITICAL:** Make update IMPLICIT - embed new value in natural narrative, don't state "X is now Y"
3704
3704  **Update Categories (rotate through all):**
3705  1. Numerical shifts (prices, quantities, measurements)
3706  2. Status changes (employment, relationships, health)
3706  3. Location changes (addresses, venues, destinations)
3707  4. Relationship progressions (social connections)
3707  [Some other categories]
3708
3708  **VERIFICATION:** STOP if no matching fact exists in bullets 1-<num_bullets>-2.
3709
3709  ### 2. USER INSTRUCTION (Bullet <num_bullets>+2)
3710  **Format:** '    **User Instruction:** Always [action] when I ask about [condition]'

3711  **MANDATORY FORMAT:** Must include "when I ask about" - this makes it testable.
3712
3712  **Instruction Types (rotate through all):**
3713  1. Output formatting rules
3714  2. Content restrictions
3714  3. Personal preferences
3715  4. Conditional responses
3715  5. Time-based rules
3716  [Some other categories]
3717
3717  **EXAMPLES:**
3718  [Some examples]
3719
3719  ### 3. LOGICAL CONTRADICTION (Bullet <num_bullets>+3)
3720  **Format:** '    **Logical Contradiction:** [Contradicting fact only]'

3721  **CRITICAL PRE-CHECK:**
3721  1. SCAN bullets 1-<num_bullets>-2 for COMPLETED ACTIONS or PERMANENT STATES:
3722  - Past tense actions: "visited", "ate", "traveled", "lived"
3722  - Permanent conditions: "born in", "raised as", "died in"
3722  - Absolute statements: "never did X", "always was Y"
3723  2. FIND exactly ONE target fact to contradict
3724  3. IF NO COMPLETED ACTIONS EXIST: CREATE setup bullet with completed action, insert between bullets 5-<
3724  >num_bullets>-2, THEN contradict in bullet <num_bullets>+4
3725

```

FORBIDDEN WORDS/PHRASES (NEVER USE):

```

3726 "Before this batch", "In this batch", [Other examples]
3727
3728 **RULE:** Only contradict COMPLETED ACTIONS, never plans/intentions.
3729 TEMPORAL QUALIFIER PROBLEM:
3730 [Some examples]
3731
3732 FORBIDDEN WORDS/PHRASES (NEVER USE):
3733 "Before this batch", "In this batch", "Previously",
3734 [Other examples]
3735 **CRITICAL:** Contradiction must make the **same original event/fact IMPOSSIBLE**, not describe a different
3736 event with different outcomes.
3737 **Contradiction Types (use variety):**
3738 1. Age/Time Reversal: Age going backward
3739 2. Death Resurrection: Dead people doing activities
3740 3. Never-Statement Violations: Contradicting "never" claims
3741 4. Location Impossibilities: Being in two places simultaneously
3742 5. Only-Statement Conflicts: Contradicting exclusivity
3743 **VERIFICATION:** Original is COMPLETED ACTION (past tense)?
3744 Contradiction makes original IMPOSSIBLE, not just different?
3745 Reads like normal, natural statement with NO hint words?
3746 Avoided ALL forbidden words that suggest conflict?
3747 ## CRITICAL VERIFICATION STEPS
3748 **Information Update:** - Can you point to EXACT number/time/date from bullets 1-<num_bullets>-2?
3749 - Changing ONLY that specific value?
3750 **Logical Contradiction:** - Can you point to EXACT bullet (1-<num_bullets>-2) being contradicted?
3751 - Original fact is COMPLETED ACTION, not plan?
3752 - Contradiction is IMPOSSIBLE, not just different?
3753 - NOT using FORBIDDEN WORDS/PHRASES mentioned in LOGICAL CONTRADICTION section
3754 ## COMMON ERRORS TO AVOID
3755 Don't place special bullets anywhere except <num_bullets>+1, <num_bullets>+2, <num_bullets>+3
3756 Don't modify original <num_bullets> bullets
3757 Don't skip any of the three special bullets
3758 Don't use other labels for special bullets
3759 ## OUTPUT FORMAT
3760 Return COMPLETE plan where each batch has ALL original bullets unchanged + exactly 3 additional bullets at the
3761 end. When setup fact is created, insert between bullets 5-<num_bullets>-2, renumber, and add special
3762 bullets as <num_bullets>+2, <num_bullets>+3, <num_bullets>+4.
3763
3764 Begin processing the plan now.

```

Listing 27: Adding special bulletpoints to conversation plan prompt

```

3759 You are a narrative coherence specialist creating CHRONOLOGICALLY SEQUENCED topic clusters for realistic
3760 conversational AI dataset generation. Your task is to generate 10 interconnected topics that form a
3761 natural life progression.
3762 ## INPUT DATA
3763 - **SEED TOPIC**: <seed_topic>
3764 - **SEED THEME**: <seed_theme>
3765 - **SEED SUBTOPICS**: <seed_subtopics>
3766 - **USER PROFILE**: <user_profile>
3767 - **TIMELINE**: <timeline>
3768 ## CORE OBJECTIVE
3769 Generate a JSON object containing 10 topics (including the provided seed topic as Topic 0) that form a
3770 CHRONOLOGICALLY COHERENT narrative where each topic naturally follows the previous one in realistic time
3771 progression.
3772 ## CRITICAL REQUIREMENTS
3773 ### 1. CHRONOLOGICAL COHERENCE & TOPIC INDEPENDENCE
3774 **TOPIC PROGRESSION RULES:** - **Topic 0**: Use the provided seed topic EXACTLY as given
3775 - **Topics 1-9**: Each must be a COMPLETELY DIFFERENT life domain/category
3776 - **NO EXTENDED NARRATIVES**: Topics 1-9 should NOT continue the seed topic's story
3777 - **LIFE PROGRESSION**: Each topic represents what naturally happens AFTER completing the previous life
3778 experience
3779 ## TOPIC INDEPENDENCE MANDATE:
3780 - Each topic must address a DIFFERENT life area (career, relationships, health, education, finances, etc.)
3781 - Topics should show how one life experience leads to growth in OTHER areas
3782 - NO topic should be "Part 2" of a previous topic
3783 ## 2. NATURAL LIFE FLOW REQUIREMENTS
3784 **CAUSAL RELATIONSHIPS WITHOUT CONTINUATION:** - Topic N+1 is INFLUENCED BY Topic N but addresses a DIFFERENT life domain
3785 - Show how growth in one area catalyzes change in another area
3786 - Example: Travel experience (Topic 0) Career reassessment (Topic 1) Relationship priorities (Topic 2)
3787 [Examples]

```

```

3780
3781     ### 3. USER PROFILE ALIGNMENT
3782     - **Demographic Consistency**: All topics must align with user's age, education, career level, and life stage
3783     - **Financial Realism**: Topics must reflect user's actual financial capacity and constraints
3784     - **Geographic Logic**: Topics must consider user's location and mobility constraints
3785     - **Value Alignment**: Topics must reflect user's stated priorities, interests, and life goals
3786
3787     ### 4. TOPIC BREADTH REQUIREMENTS
3788     Each topic must include a realistic timeline that:
3789     - **Sequential Timing**: Topics must not overlap and should follow logical temporal progression
3790     - **Duration Realism**: Each topic should span 1-2 months for authentic decision-making and implementation
3791     - **Natural Gaps**: Include realistic time gaps between major life transitions
3792     - **Seasonal Considerations**: Account for natural timing (job searches, moving seasons, academic calendars)
3793     - **Timeline Format**: Use "Month X, Year Y - Month X', Year Y'" format
3794     Each topic must be sufficiently BROAD to generate 2000+ authentic conversations by including:
3795     - **Multiple Decision Points**: 15-20 major decisions per topic
3796     - **Complex Subtopics**: 9-10 substantial subtopics that each require extensive discussion
3797     - **Ongoing Processes**: Topics involving multi-month planning, execution, and adjustment phases
3798     - **Cross-Domain Impact**: Topics affecting multiple life areas
3799
3800     ### 5. NARRATIVE REALISM
3801     - **Natural Timing**: Realistic time gaps between major life decisions
3802     - **Emotional Progression**: Topics should reflect natural emotional and psychological development
3803     - **Practical Constraints**: Topics must acknowledge real-world limitations (money, time, responsibilities)
3804
3805     ## OUTPUT FORMAT REQUIREMENTS
3806     Generate a single JSON object with this EXACT structure:
3807     ````json
3808     {"topics": [{"id": 0, "category": "[Provided Category]", "title": "[Provided Title]", "theme": "[Provided Theme]",
3809         "subtopics": [/* Provided Subtopics Array */], "timeline": "[Start Month, Year - End Month, Year]"}, {
3810         "id": 1, "category": "[New Category]", "title": "[Descriptive Title]", "theme": "[Character-focused theme
3811             describing the challenge/opportunity]", "subtopics": ["[Subtopic 1]", "[Subtopic 2]", "[Subtopic
3812             3]", "[Subtopic 4]", "[Subtopic 5]", "[Subtopic 6]", "[Subtopic 7]", "[Subtopic 8]", "[Subtopic 9]", "[Subtopic 10]"], "timeline": "[Start Month, Year - End Month, Year]"}, {
3813         // ... topics 2-9 following same structure]
3814     }]}
3815
3816     ## CRITICAL TIMELINE REQUIREMENTS
3817     ### MANDATORY NON-OVERLAPPING TIMELINE RULES
3818     **ABSOLUTE RULE**: Each topic's timeline MUST start AT LEAST one month AFTER the previous topic ends.
3819     **TIMELINE CALCULATION PROTOCOL**:
3820     1. Topic 0: Uses provided timeline exactly
3821     2. Topic N+1 start = Topic N end + AT LEAST 1 month gap
3822     3. NO overlapping months between any topics
3823     4. Each topic duration: 1-2 months
3824     [Examples]
3825
3826     **TIMELINE VERIFICATION STEPS**:
3827     Before finalizing each topic:
3828     1. Identify previous topic's END month
3829     2. Add AT LEAST 1 month to get earliest possible START
3830     3. Verify NO month appears in multiple topics
3831     4. Confirm realistic gaps for life transitions
3832
3833     **GAP JUSTIFICATION**:
3834     The 1+ month gaps represent:
3835     - Processing and integration time after major experiences
3836     - Natural life rhythms and decision-making periods
3837     - Realistic pacing of significant life changes
3838     - Time for consequences of previous decisions to manifest
3839
3840     ## TOPIC PROGRESSION GUIDELINES
3841     ### Phase 1: Post-Seed Topic Reality (Topics 1-2)
3842     - **Topic 1**: How the seed topic experience changes perspective on ANOTHER life area
3843     - **Topic 2**: Ripple effects creating needs in YET ANOTHER domain
3844
3845     ### Phase 2: Multi-Domain Growth (Topics 3-5)
3846     - **Topics 3-5**: Leveraging cumulative growth to address diverse life challenges
3847
3848     ### Phase 3: Integration Across Life (Topics 6-8)
3849     - **Topics 6-8**: Synthesizing learnings to optimize different life areas
3850
3851     ### Phase 4: Holistic Vision (Topic 9)
3852     - **Topic 9**: Long-term life design incorporating all previous growth
3853
3854     **CRITICAL QUESTION FOR EACH TOPIC**:
3855     "After completing [previous topic], what DIFFERENT area of life would this person naturally need to address
3856         next?"
3857
3858     ## QUALITY VALIDATION CHECKLIST
3859     [Examples]
3860
3861     ## EXAMPLE PROGRESSION LOGIC
3862     [Example]
3863
3864     ## FORBIDDEN ELEMENTS
3865     - **Non-sequential topics**: Topics that could happen in any order
3866     - **Profile contradictions**: Topics that contradict user's established circumstances
3867     - **Unrealistic jumps**: Major life changes without proper foundation/motivation
3868     - **Narrow topics**: Topics that couldn't generate extensive conversation
3869     - **Template responses**: Generic topics that don't reflect unique user circumstances
3870
3871     ## EXECUTION NOTES

```

3834 - Generate all 10 topics in a single coherent response
3835 - Ensure seamless narrative flow from Topic 0 through Topic 9
3836 - Prioritize realism and character consistency over dramatic storylines
3837 - Focus on authentic life progressions that real people experience
3838 - End output immediately after closing the JSON structure
3839
3840 ***CRITICAL**:** Output your response in JSON format only. Do not include any explanatory text, markdown
formatting, or additional commentary. Provide only the raw JSON object.
3841
3842 Generate the complete topic cluster now.

Listing 28: Ten million sequential seed generation prompt

```

3888
3889     "core_action_phase": {"start": "[Date]", "end": "[Date]", "duration": "[Duration]", "topics": [3, 4, 5, 6]},
3890     "integration_phase": {"start": "[Date]", "end": "[Date]", "topics": [7, 8, 9]},
3891     "subtopics": [/* 10 UNIQUE subtopic objects */]}

```

```

3891 CRITICAL:
3892 Each subtopic explores DIFFERENT aspects
3893 NO thematic repetition across topics
3894 Progressive narrative arc
3895 Diverse conversation opportunities
3896
3897 Output ONLY the JSON. No explanations

```

Listing 29: Ten million hierarchical seed generation prompt

```

3898 You are a long-form narrative planning specialist creating a COHERENT STORY PLANSET for natural conversational
3899     flow. Your task is to generate detailed batch plans that will seed realistic user-assistant dialogue.
3900
3901     ## INPUT DATA
3902     - **DOMAIN:** <domain>
3903     - **TITLE:** <title>
3904     - **THEME:** <theme>
3905     - **SUBTOPICS:** <subtopics>
3906     - **TIMELINE:** <timeline>
3907     - **NUM_BATCHES:** <num_batches> batches
3908     - **LABELS:** <provided_labels>
3909     - **USER PROFILE:** <user_profile>
3910     - **CORE RELATIONSHIPS:** <core_relationships>
3911     - **NEW RELATIONSHIPS:** <new_relationships>
3912     - **PREVIOUS PLAN:** <previous_plan>
3913     - **INCLUDE_INTRODUCTION:** <YES/NO>
3914
3915             CORE OBJECTIVE
3916             Generate <num_batches> distinct, non-repetitive batch plans that form a coherent narrative arc where a real
3917             person naturally converses with an AI assistant. Each plan must introduce NEW story elements while
3918             maintaining perfect continuity and character consistency.
3919
3920             CRITICAL NARRATIVE PERSPECTIVE
3921
3922             **MANDATORY FIRST-PERSON PERSPECTIVE:** - ALL content must be written from the USER's perspective (first-
3923             person)
3924             - Use first-person perspective throughout but VARY sentence structures - Natural narrative flow - avoid
3925             starting every bullet with "I"
3926             - Mix active and passive voice while maintaining first-person perspective
3927
3928             CONTINUITY REQUIREMENTS
3929
3930             **CRITICAL:** If PREVIOUS PLAN is provided, you MUST:
3931             - **Reference Previous Events** **Maintain Core Character Consistency** **Integrate New Relationships** **Show
3932                 Temporal Progression**
3933             **Build Upon Previous Decisions** **Preserve Established Facts** **Continue Relationship Arcs**
3934
3935             STRICT TIMELINE ENFORCEMENT
3936
3937             **CRITICAL TIMELINE PARSING (MANDATORY FIRST STEP):**
3938             Before generating ANY content, you MUST internally calculate timeline boundaries.
3939
3940             **STEP 1: Extract and Write Timeline Boundaries**
3941             **CALCULATE YOUR PARSED DATES:** 
3942
3943             **STEP 2: Create Batch Date Assignments**
3944             Divide timeline into <num_batches> segments: Days per batch = TOTAL DAYS      <num_batches>
3945
3946             **ABSOLUTE TIMELINE RULES:** 1. **EVERY date mentioned MUST be between START and END dates** 2. **NO future
3947             references beyond TIMELINE END** (no "next month" if timeline ends this month)
3948             3. **NO past references before TIMELINE START** 4. **Temporal anchors MUST progress chronologically within
3949             boundaries** 5. **Final batch MUST conclude naturally before or on END DATE**
3950
3951             **TEMPORAL ANCHOR REQUIREMENTS:**
3952             - First bullet of EACH batch MUST be temporal anchor
3953             - Format: "    **Temporal Anchor:** [Month] [Day], [year], [event description]"
3954             - Each temporal anchor date MUST be within that batch's assigned date range
3955             - Dates must progress: Batch 2's date > Batch 1's date, etc.
3956
3957             **TIMELINE VIOLATION EXAMPLES (FORBIDDEN):**
3958             [Examples]
3959             **PRE-GENERATION CHECKLIST:**
3960             [Examples]
3961
3962             CRITICAL DETAIL REQUIREMENTS
3963
3964             **MANDATORY SPECIFIC DETAILS:**
3965             Every batch MUST include numerous concrete, verifiable details that enable single-word or short factual
3966             answers:
3967
3968             **Required Detail Categories (minimum 5-7 per batch):**
3969             - **Exact Numbers:** prices ($X), quantities, percentages, measurements, distances
3970             [Some categories examples]
3971
3972             **Detail Distribution Rules:**
3973             - Each bullet must contain AT LEAST one verifiable detail - Avoid vague statements

```

STRUCTURE REQUIREMENTS

3944 ****1. OUTPUT FORMAT:****
3945 - Generate exactly <num_batches> plans
3946 Format: 'BATCH X PLAN' headers
3947 Each plan contains exactly 30 bullets
3948 Each bullet: " *[LABEL CATEGORY]:[LABEL DESCRIPTION]:** [content]" (25 words)
3949 NOTE: Each label consists of category and description. Use both for each bullet point.
3950 - Use only the provided LABELS - no custom categories
3951 - ****MANDATORY**:** First bullet MUST be Temporal Anchor with the ONLY date reference in the batch

3952 ****2. STORY PROGRESSION ARCHITECTURE:****
3953 ****IF INCLUDE_INTRODUCTION = YES:****
3954 ****BATCH 1 (Story Foundation):****
3955 - First bullet MUST be: " **Personal Introduction:** Establish initial context with SPECIFIC details
3956 Introduce all relationships with CONCRETE contexts Set up measurable goals, deadlines, and quantifiable challenges

3957 ****IF INCLUDE_INTRODUCTION = NO:****
3958 ****BATCH 1 (Continuation):**** NO personal introduction bullets Begin directly with current topic-related content
3959 Reference established character details from PREVIOUS PLAN

3960 ****BATCHES 2-<num_batches> (Story Evolution):****
3961 - Reference user as "I/my/me" (never repeat the full name) Each batch advances the timeline chronologically
3962 Build upon ALL previously established elements
3963 Show MEASURABLE progression (promotions, relationship milestones, achievement metrics)

3964 ****3. RELATIONSHIP CONTINUITY SYSTEM:****
3965 ****Core vs. New Relationship Management:****
3966 - ****CORE RELATIONSHIPS**:** Must remain consistent across all plans - same names, established details, ongoing dynamics
3967 - ****NEW RELATIONSHIPS**:** Introduce naturally based on current topic and life phase ****Relationship Integration****
3968 ****Relationship Evolution Mandate:****
3969 Every relationship mention MUST include specific interaction details:
3970 ****Evolution Stages with Required Details:****
3971 [Examples]

3972 **CONFLICT & RESOLUTION TRACKING**

3973 ****Mandatory Conflict Elements:****
3974 Each batch must include at least 2-3 situations with:
3975 - ****Clear Stakes:**** what's at risk (money amount, deadline, relationship status)
3976 [Examples]

3977 ****Conflict Types to Rotate:**** Financial decisions with specific amounts Time management with exact deadlines
3978 Relationship boundaries with specific incidents
3979 Professional choices with concrete options Personal values with specific scenarios

3980 **ANTI-REPETITION VERIFICATION**

3981 [Examples]

3982 **CONTENT DISTRIBUTION STRATEGY**

3983 ****Per Batch Requirements:****
3984 - 2-3 bullets: Relationship developments with specific incidents (mix of core and new relationships)
3985 - 2-4 bullets: Current situation with measurable metrics
3986 - 1 bullet: Exact temporal anchor (specific date/time)
3987 - 5-7 bullets: Events with verifiable outcomes
3988 - 3-4 bullets: Decisions/preferences with specific choices
3989 - Rest: Using remaining labels with concrete details

3990 ****Story Progression Patterns:****
3991 - ****Early Batches (1-3):**** Establish baselines (current salary, relationship status, living situation) **
3992 Middle Batches:** Track changes from baselines with specific metrics
3993 - ****Later Batches:**** Show cumulative results with before/after comparisons

3994 **NATURAL CONVERSATION FLOW**

3995 These plans generate conversations where users seek AI assistance for SPECIFIC situations:
3996 [Example]

3997 **QUALITY STANDARDS**

3998 ****Specificity Checklist:****
3999 - Every person has a full name and defined relationship
4000 [Other examples]

4001 ****Narrative Depth:****
4002 - Include prices, percentages, distances, durations
4003 - Show cause-and-effect with specific triggers and results
4004 - Maintain factual consistency (don't change established numbers/dates)
4005 - Reference past specific events by name and date

4006 **EXECUTION NOTES**

4007 - Prioritize concrete details over abstract descriptions
4008 - Every bullet should enable at least 2-3 factual questions
4009 - Include cultural, financial, and geographic specificity
4010 - Ensure details are realistic and internally consistent
4011 - If PREVIOUS PLAN provided, include 3-5 specific references to previous events per batch

```

3996
3997     - End immediately after 'BATCH <num_batches> PLAN'
3998
3999     **FINAL TIMELINE REMINDER:**
4000     - Parse TIMELINE boundaries FIRST
4001     - EVERY date must fall within those boundaries
4002     - NO exceptions to timeline limits
4003     - Verify each batch respects the timeline
4004
4005     Output ONLY the batch plans. No explanations or additional text.
4006
4007     Begin generation now.

```

Listing 30: Ten million sequential conversation plan generation prompt

```

4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049

You are a precision narrative architect generating TEMPORALLY COHERENT BATCH PLANS with absolute timeline
integrity and phase-appropriate content.

## INPUTS
MAIN_TITLE:<main_title> | MAIN_THEME:<main_theme> | TITLE:<title> | THEME:<theme> | TIMELINE:<timeline> |
NUM_BATCHES:<num_batches> | LABELS:<provided_labels> | USER_PROFILE:<user_profile> | CORE_RELATIONSHIPS
:<core_relationships> | NEW_RELATIONSHIPS:<new_relationships> | ALL_SUBTOPIC_PLANS:<all_subtopic_plans>
| PREVIOUS_PLANS_SUMMARY:<previous_plans_summary> | PREVIOUS_PLAN:<previous_plan> |
CURRENT_SUBTOPIC_DATA:<current_subtopic_data> | CURRENT_SUBTOPIC_ID:<current_subtopic_id> |
INCLUDE_INTRODUCTION:<YES/NO>

## OBJECTIVE
Generate <num_batches> distinct, non-repetitive batch plans forming coherent narrative where user naturally
converses with AI. Each plan introduces NEW elements while maintaining continuity.

## STRUCTURE [MANDATORY]
- Format: 'BATCH X PLAN' headers
- Exactly 30 bullets per batch
- Bullet format: " **[LABEL CATEGORY]:[LABEL DESCRIPTION]:** [content]" ( 30  words)
- First bullet ALWAYS: " **Temporal Anchor:** [Date], [context]"
- NO other dates in batch except temporal anchor
- ALL content in FIRST-PERSON ("I/my/me")
- Vary sentence structures, avoid starting every bullet with "I"

## DETAIL REQUIREMENTS [8-10 per batch minimum]
- Exact Numbers: prices($X), quantities, percentages, measurements
- Specific Dates/Times: "Month x yth", "x:y PM/AM"
[Some other examples]

Replace vague with specific:
[Examples]

## FACT TRACKING SYSTEM [MAINTAIN THROUGHOUT]
Track per batch:
1. Purchases: [Item, Price, Store, Date]
[Examples]

Before EVERY bullet:
- Check if fact exists in registry If similar exists, ADD NEW DIMENSION (consequence/complication/perspective/
progression)

## PROGRESSION PATTERNS
**Batches 1-3:** Establish baselines, initial decisions, relationship intros **Batches 4-6:** Show
consequences, complications, deepen relationships
**Batches 7-8:** Unexpected developments, secondary effects, evolution **Batches 9-10:** Long-term impacts,
synthesis, maturity, future implications

Recurring element progression: 1. First: Basic establishment 2. Second: Add complication 3. Third: Show
resolution 4. Fourth: Reveal impact 5. Fifth+: FORBIDDEN unless dramatic change

## LABEL ROTATION RULES
Track usage: Label+Focus combination FORBIDDEN across all batches
[Examples]

## RELATIONSHIP RULES
**IF INCLUDE_INTRODUCTION=YES:** Batch 1 first bullet: " **Personal Introduction:**" Establish context with
SPECIFICS (age, location, job, salary) Introduce relationships with context (how long known, where met)
**NEW_RELATIONSHIPS first appearance:** Include relationship to user + age + context After introduction, refer
naturally

Every relationship mention needs specific interaction: Introduction Development Deepening Maturation

## BATCH REQUIREMENTS
- 1 temporal anchor (specific date)
- 2-3 relationship developments
- 3-4 current situation with metrics
- 5-6 events with outcomes
- 4-5 decisions with choices
- Rest: remaining labels with details

## ANTI-REPETITION PROTOCOL
**THREE-PASS REVIEW:** 1. **Fact Uniqueness:** 2. **Information Advancement:** 3. **Cross-Batch:**

[Examples]

## DATE EXTRACTION

```

```

4050 Extract from CURRENT_SUBTOPIC_DATA
4051
4052 ## TEMPORAL BOUNDARIES
4053 **preparation phase:** [Example]
4054 **core_action phase:** [Example]
4055 **integration phase:** [Example]
4056 **ABSOLUTE:** No dates outside [START_DATE, END_DATE] from TIMELINE
4057
4058 ## CONTENT BOUNDARIES
4059 Extract from CURRENT_SUBTOPIC_DATA
4060
4061 **Rules:** ONLY generate from can_mention NEVER generate from cannot_mention ONLY reference current subtopic
4062 activities Use specified tense for main action
4063
4064 **Phase Content:** **Preparation:** [Example] **Core Action:** [Example] **Integration:** [Example]
4065
4066 ## TIMELINE DISTRIBUTION
4067 1. Calculate: Total Days = END - START + 1 2. IF Days < < num_batches >: Sequential dates 3. IF Days < <
4068 num_batches >: Group batches per day
4069
4070 **Same-day differentiation:** Time progression (morning/evening) Activity focus shifts Perspective changes
4071 Depth layers
4072
4073 ## CONTEXT INTEGRATION
4074 1. Review previous plans summary for established facts 2. Continue from previous plan if exists 3. IF
4075 INCLUDE_INTRODUCTION=YES: Introduce naturally
4076 4. IF NO: Continue without re-introduction 5. Reference prior facts consistently 6. Show progression from
4077 previous ending
4078
4079 ## VALIDATION GATES
4080 [Examples]
4081
4082 ## EXECUTION 1. Extract/verify dates 2. Write FIRST-PERSON 3. Date ONLY in anchor 4. Maximum detail density 5.
4083 Exactly 30 bullets 6. Validate boundaries
4084
4085 Output ONLY batch plans. End after 'BATCH <num_batches> PLAN'

```

Listing 31: Ten million hierarchical conversation plan generation prompt

```

4074 You are generating realistic questions that a USER would ask an AI ASSISTANT. Create questions based ONLY on
4075 the specific details in the current bullet points.
4076
4077 ## DOMAIN: <domain>
4078 ## TITLE: <title>
4079 ## CURRENT FOCUS AREAS (ONLY SOURCE FOR QUESTIONS): <FOCUSED_BULLETS>
4080 ## AVOID (ALREADY COVERED): <BATCH_HISTORY>
4081 ## CONTEXT REFERENCE (FOR UNDERSTANDING ONLY): <PREVIOUS_SUB_BATCH_PLANS> <PREVIOUS_BATCH_PLANS>
4082
4083 ## CRITICAL RULES:
4084 ### 1. MANDATORY DETAIL COVERAGE & TRACKING
4085 **BEFORE GENERATING:** List every detail from CURRENT FOCUS AREAS:
4086 - Names: [extract all names]
4087 - Ages/Numbers: [extract all numbers]
4088 - Locations: [extract all places]
4089 - Facts/Situations: [extract all specific facts]
4090
4091 **USAGE TRACKING:** Mark each detail as used to prevent repetition within current questions.
4092
4093 ### 2. ABSOLUTE SOURCE RESTRICTION
4094 **ONLY ALLOWED SOURCE:** Details explicitly written in CURRENT FOCUS AREAS bullet points
4095 **COMPLETELY FORBIDDEN:***
4096 - ANY names, places, facts, or details from CONTEXT REFERENCE sections
4097 - ANY topics or content from BATCH_HISTORY
4098 **CONTEXT REFERENCE RULE:** Use CONTEXT REFERENCE only to understand WHO people are or WHAT things mean when
4099 they appear in CURRENT FOCUS AREAS. NEVER generate questions about CONTEXT REFERENCE content.
4100
4101 ### 3. ZERO REPETITION ENFORCEMENT
4102 **ABSOLUTE REQUIREMENT:** Each specific detail can ONLY be mentioned ONCE across all questions.
4103 **ABSOLUTE PROHIBITIONS:***
4104 - Using ANY detail more than once in current questions
4105 - Mentioning ANY topic/detail from BATCH_HISTORY
4106 - Referencing ANY content from CONTEXT REFERENCE sections
4107 - Asking about broader topics not in current bullets
4108 **VERIFICATION:** Before each question, confirm it doesn't repeat previous content.
4109
4110 ### 4. ANTI-REPETITION SYSTEM
4111 **DETAIL USAGE PATTERN:***
4112 - First mention: Use full specific detail from bullet point
4113 - Subsequent references: Use pronouns ("he", "she", "it", "that", "my choice")
4114 **VERIFICATION:** Check each question doesn't repeat:
4115 - Specific names/numbers already used
4116 - Topics from BATCH_HISTORY
4117 - Any details or content from reference sections (CONTEXT REFERENCE)
4118
4119 ### 5. REALISTIC CONVERSATION STYLE
4120 **NATURAL LANGUAGE:***
4121 - Contractions: "I'm", "don't", "can't"
4122 - Casual words: "kinda", "sorta", "gonna"
4123 - Fillers: "like", "um", "you know"
4124 - Informal: "...", "?", "!"
```

```

4104
4105     ### 6. QUESTION VARIETY
4106     **AVOID REPETITIVE PATTERNS:**
4107     - Don't start multiple questions the same way
4108     - Vary question length and complexity
4109
4110     ### 7. QUESTION GENERATION STRATEGY
4111     - **Normal question**
4112     - **Seek advice**
4113     - **Ask for help**
4114     - **Request clarification**
4115     - **Get guidance**
4116     - **Express emotions**
4117     - **Validate decisions**
4118     - **Process thoughts**
4119     - **Explore options**
4120
4121     **QUESTION CLUSTERING:**
4122     - Some bullets get 1 question, others get 2-3
4123     - Deep dive into complex situations
4124     - Quick questions for simple details
4125     - User introducing himself/herself should be first question
4126
4127     ## OUTPUT REQUIREMENTS:
4128     Generate exactly <SUB_BATCH_SIZE> questions that:
4129     1. **USE EVERY DETAIL** from current bullet points exactly once
4130     2. Sound like genuine human requests for AI help
4131     3. Focus on specific personal situations mentioned
4132     4. Avoid all repetition from previous batches
4133     5. Show realistic emotional responses to bullet situations
4134     6. Follow natural conversation flow
4135     7. **NEVER** repeat specific details within current questions**
4136     8. **NO** repetitive "and" chains** in any message
4137
4138     **SUCCESS CRITERIA:**
4139     - Every name, age, location, fact from bullets MUST appear and appear ONLY ONCE
4140     - No repetition of BATCH_HISTORY topics
4141     - Questions sound like real people texting for advice
4142     - All questions trace back to specific bullet details
4143     - Subsequent references use pronouns/generic terms only
4144     - ZERO content from CONTEXT REFERENCE sections
4145
4146     **OUTPUT FORMAT:**
4147     For each question, use this exact format:
4148     [question text] ->-> [bullet_number]
4149
4150     **CRITICAL:**
4151     - Each question MUST end with "->-> [number]" where [number] is the bullet point it's based on
4152     - Use bullet numbers 1, 2, 3, etc. as they appear in CURRENT FOCUS AREAS
4153     - If a question combines details from multiple bullets, use the primary bullet number
4154     - If the question is not generated from any bulletpoints, put N/A
4155     - Generate exactly <SUB_BATCH_SIZE> questions
4156
4157     **Format:** One question per line, natural length, no numbering or extra text.

```

Listing 32: Question generation general domain prompt

```

4138 You are generating realistic coding questions that a DEVELOPER would ask an AI ASSISTANT. Create questions
4139 based ONLY on the specific details in CURRENT FOCUS AREAS.
4140
4141     ## CURRENT FOCUS AREAS (STRICT SCOPE - ONLY SOURCE FOR QUESTIONS):
4142     <FOCUSED_BULLETS>
4143
4144     ## QUESTIONS ALREADY COVERED IN THIS BATCH (AVOID THESE):
4145     <BATCH_HISTORY>
4146
4147     ## CONTEXT REFERENCE (FOR UNDERSTANDING ONLY - DO NOT GENERATE QUESTIONS ABOUT THIS):
4148     <PREVIOUS_SUB_BATCH_PLANS>
4149     <PREVIOUS_BATCH_PLANS>
4150
4151     ## BULLET TYPE DETECTION
4152     **MANDATORY FIRST STEP - CHECK EACH BULLET:**
4153     - If bullet contains "**Time Anchor:**"      ABSOLUTELY NO CODE, ONLY project/scheduling questions
4154     - If bullet contains "**Personal Introduction:**"  ABSOLUTELY NO CODE, ONLY career/personal questions
4155     - Otherwise      Technical bullet, GENERATE CODE
4156
4157     ### 1. CURRENT FOCUS AREAS BULLET TYPE IDENTIFICATION - CHECK FIRST
4158     **BEFORE DOING ANYTHING:** Identify the bullet type in CURRENT FOCUS AREAS:
4159     - **Time Anchor:** bullets (contain "Time Anchor:" in title)      NO CODE GENERATION
4160     - **Personal Introduction:** bullets (contain "Personal Introduction:" in title)      NO CODE GENERATION
4161     - **Technical:** (all others)      CODE GENERATION REQUIRED
4162
4163     ### 2. MANDATORY DETAIL COVERAGE & TRACKING
4164     **STEP 1 - MANDATORY EXTRACTION:** Before writing ANY questions, you MUST extract and list EVERY SINGLE detail
4165         from CURRENT FOCUS AREAS:
4166         **Extract ALL of these categories:**
4167         - **Names:** [list EVERY name - developer names, company names, project names, client names]
4168         - **Numbers/Versions:** [list EVERY version number, date, time, quantity, port, ID, measurement]
4169         [Other categories examples]
4170
4171         **STEP 2 - VERIFICATION:** Count total extracted details. You MUST use 100% of them.
4172         **STEP 3 - TRACKING:** As you write each question, mark which specific details it uses.

```

```

4158
4159    **STEP 4 - FINAL CHECK:** Before submitting, verify EVERY extracted detail appears in at least one question.
4160    **ABSOLUTE REQUIREMENT:** Every single extracted detail MUST appear in at least one question across the batch.
4161    NO EXCEPTIONS.
4162
4163    #### 3. ABSOLUTE SOURCE RESTRICTION
4164    **ONLY ALLOWED:** Details explicitly written in CURRENT FOCUS AREAS bullet points
4165    **COMPLETELY FORBIDDEN:** 
4166    - ANY content from BATCH_HISTORY
4167    - ANY content from CONTEXT_REFERENCE sections
4168    - Generic programming questions
4169    - Details not explicitly mentioned in current bullets
4170
4171    **CONTEXT REFERENCE RULE:** Use only to understand WHAT technologies/components mean when they appear in
4172    CURRENT FOCUS AREAS.
4173
4174    #### 4. DETAIL USAGE PATTERN (ANTI-REPETITION)
4175    - **First mention:** Use full specific detail from bullet point (exact names, versions, error messages)
4176    - **Subsequent references:** Use pronouns ("it", "that", "my React app", "the API")
4177    - **VERIFICATION:** Each specific detail appears ONLY ONCE across all questions
4178
4179    #### 5. MANDATORY COMPLEX CODE GENERATION
4180    **CRITICAL:** 85% of questions MUST include substantial code snippets (20-60+ lines)**
4181    ONLY IF bullet's title in CURRENT FOCUS AREAS is not time anchor or personal introduction
4182
4183    **ABSOLUTE EXCEPTIONS - NO CODE GENERATION:** 
4184    - **Time Anchor:** bullets - NEVER EVER generate any code, programming solutions, or technical implementations
4185    - **Personal Introduction:** bullets - NEVER EVER generate any code, programming solutions, or technical
4186    implementations
4187    - **FOR PERSONAL INTRODUCTION BULLETS:** Generate questions FROM the perspective of the person introducing
4188    themselves
4189    The person in the bullet is the USER asking the questions
4190    These are contextual/personal details, NOT technical coding scenarios
4191
4192    **STOP AND VERIFY:** If the bullet contains "Time Anchor:" or "Personal Introduction:" in the title, you MUST
4193    NOT generate ANY code blocks, programming solutions, scripts, or technical implementations. Period.**
4194
4195    **FOR TIME ANCHOR/PERSONAL INTRODUCTION BULLETS:** 
4196    - Focus on project management, scheduling, personal goals
4197    - Ask about deadlines, meeting coordination, project planning
4198    - NO code blocks, NO programming solutions, NO technical implementations
4199    - Use natural conversation about timing, goals, and context
4200
4201    **CODE COMPLEXITY REQUIREMENTS (for all other bullets):** 
4202    - **Minimum 20-60+ lines per code block**
4203    - **Multiple functions/methods/classes (4-6 minimum)**
4204    - **Realistic imports and dependencies (3-5 minimum)**
4205    - **Proper error handling, validation, edge cases**
4206    - **Complex business logic, database operations, API calls**
4207    - **Production-level structure and realistic variable names**
4208
4209    **REQUIRED PATTERNS (for coding bullets only):** 
4210    - **Debugging (40%):** Generate buggy code with realistic, hard-to-spot errors
4211    - **Code Review (25%):** Generate working but suboptimal code needing improvements
4212    - **Implementation (20%):** Generate partial implementations with detailed TODOS
4213    - **Optimization (15%):** Generate slow/inefficient but functional code
4214
4215    **NEVER use simple examples or tutorial-style code - always production-level complexity**
4216
4217    #### 6. AUTHENTIC DEVELOPER STYLE
4218    **Language:** Use contractions ("I'm", "don't"), dev slang ("lol", "btw"), fillers ("like", "um"), informal
4219    punctuation ("...", "?!", "!!")
4220    **Emotion:** Show genuine feelings - frustration with bugs, excitement about features
4221    **Natural Flow:** Mix question lengths, include rambling, thinking out loud
4222    **Technical Authenticity:** Include actual error messages, file names, version numbers from bullets
4223
4224    #### 7. CHRONOLOGICAL ORDER
4225    Process bullet points in exact order provided. Earlier bullet details appear in earlier questions.
4226
4227    #### 8. CODING QUESTION STRATEGY
4228    - **Implementation:** "Help me build [specific feature from bullet]"
4229    - **Debugging:** "I'm getting this error: [specific error]. How do I fix it?"
4230    - **Code Review:** "Can you review this [specific code] and suggest improvements?"
4231    - **Optimization:** "How can I make [specific implementation] faster?"
4232
4233    ## OUTPUT REQUIREMENTS:
4234    Generate exactly <SUB_BATCH_SIZE> questions that:
4235
4236    1. **MANDATORY:** Use EVERY SINGLE detail from FOCUSED_BULLETS at least once (names, versions, errors, files,
4237        specs, etc.)
4238    2. **85% MUST** include substantial code snippets (20-60+ lines) with production-level complexity**
4239    3. Sound like genuine developer requests with realistic technical scenarios
4240    4. Follow chronological order of bullet points
4241    5. **ALWAYS** include complete, complex code - NEVER use simple examples**
4242    6. Match one of the four coding patterns with appropriate complexity
4243    7. Stay strictly within bullet point scope
4244    8. **MANDATORY VERIFICATION:** Before submitting, confirm every extracted detail appears in the questions
4245
4246    **OUTPUT FORMAT:** 
4247    - **CRITICAL:** Generate ONLY the developer messages, nothing else**
4248    - Do NOT include question numbers, headers, or organizational text
4249    - **MANDATORY:** Separate each complete message with "---MESSAGE_SEPARATOR---"**
4250    - Each message can span multiple lines and include code blocks

```

```

4212
4213 - **CRITICAL: Each message MUST end with ">>> [number]" where [number] is the bullet point it's based on**
4214 - **MANDATORY: End with "### COMPLETE ###"**
4215
4216 **REQUIRED FORMAT PATTERN:** [Output format example]
4217
4218 **CRITICAL FORMATTING RULES:** [Output formatting rules]
4219
4220 **CRITICAL VERIFICATION:** Before each question:
4221 1. **MANDATORY FIRST CHECK: What type of bullet is this?**
4222 - Time Anchor bullet Generate project management/scheduling questions, ABSOLUTELY NO CODE
4223 - Personal Introduction bullet Generate career/personal questions FROM their perspective, ABSOLUTELY NO
4224 CODE
4225 - Technical bullet Generate coding questions WITH substantial code
4226 2. Does this use a specific detail from current bullets?
4227 3. Have I included substantial, complex code (not simple examples) for technical bullets?
4228 4. Does this sound like a real developer asking for help?
4229 5. Am I following one of the four coding patterns correctly for technical bullets?
4230
4231 **FINAL VERIFICATION BEFORE SUBMITTING:** Count how many extracted details appear in your questions. It MUST be 100% of all details from CURRENT FOCUS
4232 AREAS.
4233
4234 **Generate exactly <SUB_BATCH_SIZE> questions in the format above.**

```

Listing 33: Question generation coding domain prompt

```

4235 You are generating realistic math questions that a USER would ask an AI ASSISTANT. Create questions based ONLY
4236 on the specific details in CURRENT FOCUS AREAS.
4237
4238 ## CURRENT FOCUS AREAS (STRICT SCOPE - ONLY SOURCE FOR QUESTIONS):
4239 <FOCUSED_BULLETS>
4240
4241 ## QUESTIONS ALREADY COVERED IN THIS BATCH (AVOID THESE):
4242 <BATCH_HISTORY>
4243
4244 ## CONTEXT REFERENCE (FOR UNDERSTANDING ONLY - DO NOT GENERATE QUESTIONS ABOUT THIS):
4245 <PREVIOUS_SUB_BATCH_PLANS>
4246 <PREVIOUS_BATCH_PLANS>
4247
4248 ## CRITICAL RULES:
4249
4250 1. CURRENT FOCUS AREAS BULLET TYPE IDENTIFICATION - CHECK FIRST
4251 **BEFORE DOING ANYTHING:** Identify the bullet type in CURRENT FOCUS AREAS:
4252 - **Time Anchor:** bullets (contain "Time Anchor:" in title) NO MATHEMATICAL WORK GENERATION
4253 - **Personal Introduction:** bullets (contain "Personal Introduction:" in title) NO MATHEMATICAL WORK
4254 GENERATION
4255 - **Mathematical bullets:** (all others) MATHEMATICAL WORK GENERATION REQUIRED
4256
4257 2. MANDATORY DETAIL COVERAGE & TRACKING
4258 **STEP 1 - MANDATORY EXTRACTION:** Before writing ANY questions, you MUST extract and list EVERY SINGLE detail
4259 from CURRENT FOCUS AREAS:
4260
4261 **Extract ALL of these categories:** [list EVERY name mentioned - people, places, institutions, etc.]
4262 - **Names:** [list EVERY number, age, percentage, score, quantity, measurement]
4263 [Other categories example]
4264
4265 **STEP 2 - VERIFICATION:** Count total extracted details. You MUST use 100% of them.
4266 **STEP 3 - TRACKING:** As you write each question, mark which specific details it uses.
4267 **STEP 4 - FINAL CHECK:** Before submitting, verify EVERY extracted detail appears in at least one question.
4268
4269 **ABSOLUTE REQUIREMENT:** Every single extracted detail MUST appear in at least one question across the batch.
4270 NO EXCEPTIONS.
4271
4272 3. ABSOLUTE SOURCE RESTRICTION
4273 **ONLY ALLOWED:** Details explicitly written in CURRENT FOCUS AREAS bullet points
4274 **COMPLETELY FORBIDDEN:** [list]
4275 - ANY content from BATCH_HISTORY
4276 - ANY content from CONTEXT REFERENCE sections
4277 - Generic questions about mathematical fields
4278 - Details not explicitly mentioned in current bullets
4279
4280 **CONTEXT REFERENCE RULE:** Use only to understand WHO/WHAT things mean when they appear in CURRENT FOCUS
4281 AREAS.
4282
4283 4. DETAIL USAGE PATTERN (ANTI-REPETITION)
4284 - **First mention:** Use full specific detail from bullet point
4285 - **Subsequent references:** Use pronouns ("it", "that", "my homework")
4286 - **VERIFICATION:** Each specific detail appears ONLY ONCE across all questions
4287
4288 5. MANDATORY MATHEMATICAL WORK INCLUSION
4289 **CRITICAL:** When referencing problems, equations, or mathematical work that isn't explicitly provided in
4290 bullets, you MUST generate and include the complete mathematical content.
4291
4292 **ABSOLUTE EXCEPTIONS - NO MATHEMATICAL WORK GENERATION:** [list]
4293 - **Time Anchor:** bullets - Generate questions about scheduling, deadlines, timing without any MATHEMATICAL
4294 WORK
4295 - **Personal Introduction:** bullets - Generate questions FROM the person introducing themselves about their
4296 background, career, goals without any MATHEMATICAL WORK
4297 - **FOR PERSONAL INTRODUCTION BULLETS:** Generate questions FROM the perspective of the person introducing
4298 themselves

```

```

4266 The person in the bullet is the USER asking the questions
4267 These are contextual/personal details, NOT MATHEMATICAL WORK scenarios
4268 **FOR TIME ANCHOR/PERSONAL INTRODUCTION BULLETS:**
4269 - Focus on study scheduling, academic deadlines, learning goals
4270 - Ask about exam preparation, study coordination, academic planning
4271 - NO mathematical equations, NO problem-solving, NO calculations
4272 - Use natural conversation about timing, goals, and academic context
4273 **REQUIRED PATTERNS:**
4274 - **Completely Stuck (40%):** NO work shown, just describe what you're trying to solve
4275 - **Partially Stuck (30%):** Show ONLY initial 2-4 steps where you got stuck
4276 - **Need Verification (20%):** Show ONLY final answer/result
4277 - **Conceptual Confusion (10%):** NO calculations, concept-focused questions
4278 **WORK GENERATION REQUIREMENTS:**
4279 - Match mathematical level mentioned in bullet points
4280 - Include specific numbers, variables, expressions
4281 - Create realistic problems users would encounter
4282 - **NEVER use placeholders like "... (insert work)" - ALWAYS generate actual mathematical work**
4283 #### 6. AUTHENTIC USER STYLE
4284 **Language:** Use contractions ("I'm", "don't"), casual slang ("lol", "btw"), fillers ("like", "um"), informal
4285 punctuation ("...", "?", "!")
4286 **Emotion:** Show genuine feelings - confusion, frustration, excitement
4287 **Natural Flow:** Mix question lengths, include rambling, thinking out loud
4288 #### 7. CHRONOLOGICAL ORDER
4289 Process bullet points in exact order provided. Earlier bullet details appear in earlier questions.
4290 #### 8. QUESTION GENERATION STRATEGY
4291 - **Problem-Solving:** "Help me solve [specific problem]"
4292 - **Concept Clarification:** "I don't understand [specific concept]"
4293 - **Solution Verification:** "Can you check if my solution is correct?"
4294 - **Method Explanation:** "Why does [specific method] work?"
4295 ## OUTPUT REQUIREMENTS:
4296 Generate exactly <SUB_BATCH_SIZE> questions that:
4297
4298 1. **MANDATORY:** Use EVERY SINGLE detail from FOCUSED_BULLETS at least once (names, ages, dates, traits,
4299 goals, timeframes, etc.)
4300 2. **80% MUST include substantial mathematical work** (equations, calculations, solution attempts)
4301 3. Sound like genuine user requests with realistic mathematical content
4302 4. Follow chronological order of bullet points
4303 5. **ALWAYS include complete mathematical problems/equations - NEVER use placeholders**
4304 6. Match one of the four behavioral patterns with appropriate work shown
4305 7. Stay strictly within bullet point scope
4306 8. **MANDATORY VERIFICATION:** Before submitting, confirm every extracted detail appears in the questions
4307
4308 **OUTPUT FORMAT:**
4309 - **CRITICAL:** Generate ONLY the user messages, nothing else**
4310 - Do NOT include question numbers, headers, or organizational text
4311 - **MANDATORY:** Separate each complete message with "---MESSAGE_SEPARATOR---"**
4312 - Each message can span multiple lines and include mathematical expressions
4313 - **CRITICAL:** Each message MUST end with ">-> [number]" where [number] is the bullet point it's based on**
4314
4315 **REQUIRED FORMAT PATTERN:**
4316 [Output format example]
4317
4318 **CRITICAL FORMATTING RULES:**
4319 [Output formatting rules]
4320
4321 **CRITICAL VERIFICATION:** Before each question:
4322 1. Does this use a specific detail from current bullets?
4323 2. **Is this a Time Anchor or Personal Introduction bullet? If YES, do NOT include any mathematical work**
4324 3. **For Personal Introduction: Am I generating questions FROM the person's perspective (they are the user)?**
4325 4. Have I included actual mathematical work (not placeholders) for mathematical bullets?
4326 5. Does this sound like a real user asking for help?
4327 6. Am I following one of the four behavioral patterns correctly for mathematical bullets?
4328
4329 **FINAL VERIFICATION BEFORE SUBMITTING:**
4330 Count how many extracted details appear in your questions. It MUST be 100% of all details from CURRENT FOCUS
4331 AREAS.
4332
4333 **Generate exactly <SUB_BATCH_SIZE> questions in the format above.**

```

Listing 34: Question generation math domain prompt

```

4313 You will receive an AI assistants reply. Determine whether it contains a **direct, specific question** that
4314 the user must answer next by providing new information, preferences, or a decision.
4315 - **YES** if, and only if, the assistants reply asks for a concrete user response (e.g. Whats your
4316 budget for this trip? , Which option would you prefer? ).
4317 - **NO** for generic or rhetorical prompts (e.g. Any questions? , Would you like to dive deeper? ,
4318 Consider your budget) that do not demand an immediate, specific answer.
4319
4320 AI ASSISTANT RESPONSE:
4321 <assistant_response>
4322
4323 CRITICAL NOTE: Respond only in English. Do not include any Chinese.

```

```

4320
4321     Output exactly **YES** or **NO**, nothing else.
4322
4323
4324     You are simulating a typical user in conversation. Decide if you would ask a follow-up question after the AI's
4325     response.
4326
4327     **CONVERSATION CONTEXT:**  

4328     - DOMAIN: <domain>  

4329     - TITLE: <title>  

4330     - THEME: <theme>  

4331     - SUBTOPICS: <subtopics>  

4332     - Recent History: <formatted_history>  

4333     - AI's Last Response: <assistant_response>
4334
4335     **ASK FOLLOW-UP ("yes") WHEN:**  

4336     1. **Missing Info**: The response lacks details you genuinely need to proceed  

4337         - Specific steps for a process you're trying to follow  

4338         - Key parameters (dates, amounts, requirements) for a decision you're making  

4339         - Clarification on which option applies to your specific situation
4340
4341     2. **Genuine Confusion**: Something is unclear or contradictory  

4342         - Technical terms used without explanation that block understanding  

4343         - Conflicting information that affects your next action  

4344         - Ambiguous instructions where the wrong interpretation has consequences
4345
4346     3. **Incomplete Practical Guidance**: You asked "how to" but can't actually do it yet  

4347         - Missing steps in a procedure  

4348         - Lacks specifics needed for implementation  

4349         - Assumes knowledge you don't have
4350
4351     **NO FOLLOW-UP ("no") WHEN:**  

4352     1. **Good Enough to Proceed**: You have what you need
4353
4354     **OUTPUT:** Only "yes" or "no"

```

Listing 35: Check assistant's response include question prompt

```

4355
4356
4357     You must respond only in English. Never switch to Chinese or any other language mid-sentence. All responses
4358     should be entirely in English.
4359     Respond to the user's message by either:  

4360         Fully answering their question  

4361         . Provide a comprehensive answer  

4362         Answering plus asking at most ONE follow-up question if you need more detail
4363
4364     Always honor these rules:  

4365     1. Do NOT ask questions the user already answered.  

4366     2. Only ask a question if you genuinely need context to provide a complete, actionable answer.  

4367     3. Keep your main answer clear and comprehensive before you ask.  

4368     4. Use the following system inputs:  

4369
4370     DOMAIN: <domain>  

4371     TITLE: <title>  

4372     THEME: <theme>  

4373     SUBTOPICS: <subtopics>  

4374     PREVIOUS PLANS SUMMARY: <previous_plans_summary>  

4375     PREVIOUS BATCHES OF THIS PLAN: <previous_batches>  

4376     CURRENT HISTORY: <current_batch_messages>
4377
4378     CRITICAL NOTE: Respond only in English. Do not include any Chinese.
4379
4380     **Output**  

4381     Return exactly what you'd say to the user no tags, no internal notes.

```

Listing 36: Check need for followup prompt

```

4382
4383
4384     You are role-playing as a real user having an authentic conversation with an AI chat assistant.
4385     The AI assistant has just asked you a question, and you need to provide a natural, human-like response.
4386
4387     #### Input Context You'll Receive:  

4388     - **Current Batch Message History**: The conversation flow leading to the AI's question  

4389     - **Domain, Title, Theme & Subtopics**: The main subject of this conversation  

4390     - **Previous Plans**: Summary of earlier conversation contexts for continuity  

4391     - **Current Plan**: The overarching narrative direction for this conversation batch  

4392     - **AI's Question**: The specific question the assistant asked that requires your response
4393
4394     #### INPUTS:  

4395     **Current Batch Message History**: <current_batch_messages>  

4396     **DOMAIN**: <domain>  

4397     **TITLE**: <title>  

4398     **THEME**: <theme>  

4399     **SUBTOPICS**: <subtopics>  

4400     **Previous Plans Summary**: <previous_plans_summary>  

4401     **Previous Batches of This Plan**: <previous_batches>  

4402     **Current Plan**: <current_plan>  

4403     **AI's Question**: <ai_last_message>
4404
4405     #### CRITICAL: Keep Responses SHORT and Natural

```

```

4374 **Real users give brief, to-the-point answers to AI questions**
4375
4376 ##### Your Role & Behavior:
4377 You are a real human user with:
4378 - Personal experiences, opinions, and emotions
4379 - Natural speech patterns and conversational habits
4380 - Realistic knowledge limitations and curiosity
4381 - Consistent personality traits across the conversation
4382
4383 ##### Language Authenticity:
4384 - Use lots of contractions: "I'm", "don't", "can't", "it's", "that's"
4385 - Include casual slang: "lol", "btw", "thh", "kinda", "sorta", "gonna", "wanna"
4386 - Add filler words: "like", "um", "you know", "I mean", "so", "well"
4387 - Use informal punctuation: multiple periods "...", question marks "??", exclamation points "!!"
4388
4389 ##### Imperfect Natural Speech:
4390 - Include minor typos and informal grammar
4391 - Add rambling elements: "I mean, ..."
4392 - Include thinking out loud: "hmm", "actually", "oh", "maybe"
4393
4394 ##### Emotional Authenticity:
4395 - Show genuine feelings: excitement, frustration, uncertainty, hope
4396 - Use emotional language
4397 - Add personal reactions: "ugh", "omg", "yay", "oof", "blah"
4398
4399 ##### Response Guidelines:
4400 **STEP 1 - Check Plans for Existing Information**:
4401 - **First**: carefully review the Current Plan and Previous Plans for any information that answers the AI's
4402 question
4403 - **If found in plans**: Base your response on that established information to maintain story continuity
4404 - **If not found in plans**: Create a new answer that aligns with the topic, theme, and existing storyline
4405
4406 **STEP 2 - Answer the AI's Question Directly**:
4407 - Keep your reply focused on answering; do **not** introduce new questions.
4408 - Give a direct answer to what the AI asked
4409 - Don't over-explain or provide unnecessary details
4410 - Answer like you would in a real text conversation
4411 - Include personal context or examples when natural
4412 - **CRITICAL**: Ensure your answer doesn't contradict anything established in previous or current plans
4413
4414 ##### Stay Consistent with Context:
4415 - Maintain the same personality and circumstances throughout
4416 - Keep your responses aligned with the current topic and theme
4417
4418 ##### Response Characteristics:
4419 - **Tone**: Match the conversation's emotional tone and your established personality
4420 - **Authenticity**: Sound like a real person, not an AI trying to sound human
4421
4422 ##### Critical Instructions:
4423 - **Be concise** - Real people don't write long in chat
4424 - Keep your reply focused on answering; do **not** introduce new questions.
4425 - **ALWAYS check plans first** - Look for any information that answers the AI's question before creating new
4426 details
4427 - **Maintain consistency** - Never contradict information established in current or previous plans
4428 - **Fill gaps naturally** - If plans don't have the answer, create responses that fit the established
4429 storyline
4430 - ONLY provide your response as the user - no meta-commentary
4431 - Stay in character as a human user throughout
4432 - Answer the question but don't feel obligated to ask a question back (the AI asked YOU)
4433

```

Listing 38: User LLM answer generation prompt

```

4441 You must respond **only in English**. Do not include any Chinese characters or phrases in your response.
4442 You are a real person having a conversation with an AI assistant. Based on the conversation history and the AI
4443 's last response, ask ONE natural follow-up question.
4444
4445 ## CONTEXT:
4446 **Current Batch Message History**: <current_batch_messages>
4447 **DOMAIN**: <domain>
4448 **TITLE**: <title>
4449 **THEME**: <theme>
4450 **SUBTOPICS**: <subtopics>
4451 **Previous Plans Summary**: <previous_plans_summary>
4452 **Previous Batches of This Plan**: <previous_batches>
4453 **Current Plan**: <current_plan>
4454 **AI's Response**: <ai_last_message>
4455
4456 ## YOUR TASK
4457 Ask a follow-up question (10-20 words) that a real person would naturally ask after receiving the AI's
4458 response.
4459
4460 ## CRITICAL RULES TO PREVENT REPETITION
4461 Before generating your question:
4462 1. **Scan the Current Batch Message History** for all topics already discussed
4463 2. **Check Previous Batches** for questions already asked
4464 3. **Never ask about something already covered**
4465
4466 If you notice your question seeks information already provided in the conversation history, STOP and generate
4467 a completely different question.
4468
4469 ## HOW REAL PEOPLE ASK FOLLOW-UPS

```

```

4428
4429     ### Natural conversation starters:
4430     - "oh wait..." / "hmm..." / "actually..." / "btw..." / "ok but..."
4431     - "that's cool but..." / "makes sense, though..." / "yeah but what about...""
4432
4433     ### Authentic reaction patterns:
4434     **Building on the last ai response:***
4435     - Ask about a specific topic not yet covered
4436     - Connect it to your personal situation from the Current Plan
4437
4438     **Showing genuine reactions:***
4439     - If AI gave good news      "nice! but does that mean...""
4440     [Other examples]
4441
4442     ### Question types that feel natural:
4443     - **Practical concerns**: "how long does that usually take?""
4444     [Other types]
4445
4446     ## NATURAL SPEECH PATTERNS
4447     Include these elements to sound human:
4448     - Contractions: "don't", "can't", "won't", "that's"
4449     - Casual words: "kinda", "sorta", "gonna", "like"
4450     - Emotional reactions: "ugh", "hmm", "oh", "yikes"
4451     - Informal punctuation: "..." or "???" or "!"
4452
4453     ## CONVERSATION FLOW AWARENESS
4454     Based on where you are in the Current Batch Message History:
4455     Ask broader exploratory questions
4456     Ask for specific details or comparisons
4457     Ask about implementation or next steps
4458
4459     ## AUTHENTICITY CHECKLIST
4460     [Some examples]
4461
4462     CRITICAL NOTE: Respond only in English. Do not include any Chinese.
4463
4464     ## YOUR RESPONSE:
4465     [Generate only the follow-up question, nothing else]

```

Listing 39: User LLM ask followup question prompt

```

4453 I provide you with a text. Your task is to identify all the details stated in the text,
4454 and output that in key: value format.
4455 E.g.:
4456 Key 1: Value 1,
4457 Key 2: Value 2,
4458 Key 3: Value 3,
4459 ....
4460 Also at the end, I want you to provide a brief summary of what this text was about in this format: Summary: '
4461 summarized text'
4462 Note: only output key-values and the summary. DO NOT provide any explanation before or after that.
4463 Note: Do not output Key 1, Key 2, ...
4464
4465     **Previous Context:***
4466     {history}
4467
4468     text: {text}

```

Listing 40: Key-value extraction prompt

```

4464
4465
4466 You are a highly analytical AI assistant. Your task is to analyze the latest conversation exchange and produce
4467 a structured summary of key information and insights.
4468
4469     **Your Internal Process:***
4470     To ensure maximum accuracy, you must first think step-by-step.
4471     1. **Analyze:** Break down the user's latest message.
4472     2. **Identify:** Pinpoint all facts, instructions, and updates.
4473     3. **Deduce:** Reason about the implications of the new information in the context of the conversation
4474     history. What is the user's underlying goal or state?
4475     4. **Format:** After completing your internal analysis, format the conclusions into the 'Extracted Facts'
4476     structure.
4477
4478     **Crucial Instruction:** Your final output must **ONLY** be the 'Extracted Facts' block. **DO NOT** include
4479     your step-by-step reasoning or any other text in your response. Strictly follow the format shown in the
4480     example's output.
4481
4482     ---
4483     **EXAMPLE**
4484
4485     **Conversation Context:***
4486     * **Recent Conversation History:***
4487     USER: Hey, I need some help with the "Project Phoenix" launch plan.
4488     ASSISTANT: Of course. What do you need?
4489     USER: The launch date is set for September 15th, 2025. I'm responsible for the marketing materials.
4490     * **Latest Exchange to Analyze:***
4491     USER: Okay, the final budget for the social media campaign is $7,500. The client, Innovate Corp, just
4492     approved it. Please find me three case studies of successful B2B SaaS launches by tomorrow, August
4493     28th. And don't include any of our direct competitors in the examples.

```

```

4482
4483     ASSISTANT: Understood. I will find three case studies of successful B2B SaaS launches, excluding
4484         competitors, and have them for you by tomorrow, August 28th. The approved budget of $7,500 for the
4485         social media campaign has been noted.
4486
4487     **Example of Correct Final Output:**
4488         * The client's name is "Innovate Corp".
4489         * The project is related to a "B2B SaaS launch".
4490         * The final budget for the social media campaign is \$7,500.
4491         * A deadline is set for "tomorrow, August 28th".
4492         * User intends to review three case studies for the project.
4493         * Instruction: Find three case studies.
4494         * Constraint: Do not include direct competitors in the examples.
4495         * The budget for the social media campaign has been approved by the client.
4496         * The user is under a deadline and needs the case studies urgently to inform their work on the marketing
4497             materials.
4498
4499     ---
4500     **ACTUAL TASK**
4501
4502     **Recent Conversation History:**  
(history)
4503
4504     **Latest Exchange to Analyze:**  
USER: {latest_user_message}  
ASSISTANT: {latest_assistant_message}
4505
4506     **Extracted Facts:**
```

Listing 41: Scratchpad creation prompt

```

4507 You are tasked with summarizing and compressing scratch pad content to fit within a specific token limit.
4508 **Input Content:**  
{content}
4509
4510 **Target Length:** {tokens_limit} tokens
4511
4512 **Your Task:**  
Compress this content by clustering related information, removing redundancy, and prioritizing the most
4513     important details.
4514
4515 **Process:**  
1. **Cluster:** Group related information by topic, entity, or theme
4516     2. **Deduplicate:** Remove redundant or repetitive information
4517     3. **Prioritize:** Keep the most important and contextually relevant details
4518     4. **Compress:** Condense while maintaining essential meaning and context
4519
4520 **Output Format:**  
Return ONLY the compressed content organized as:
4521
4522 **KEY ENTITIES & RELATIONSHIPS:**  
- [Most important people, organizations, systems mentioned]
4523
4524 **CORE DECISIONS & PREFERENCES:**  
- [Critical decision points, requirements, constraints]
4525
4526 **PROCESSES & WORKFLOWS:**  
- [Essential procedural information and methodologies]
4527
4528 **USER PREFERENCES:**  
- [User's stated likes, dislikes, preferred methods, settings, choices]
4529
4530 **USER INSTRUCTIONS:**  
- [Specific directions, commands, or guidance provided by the user]
4531
4532 **IMPORTANT DATES:**  
- [Deadlines, milestones, scheduled events, time-sensitive information]
4533
4534 **CRITICAL CONTEXT:**  
- [Background information necessary for understanding]
4535
4536 **ACTIONABLE ITEMS:**  
- [Next steps, pending actions, deadlines]
4537
4538 **IMPORTANT DEVELOPMENTS:**  
- [Significant events, changes, milestones]
4539
4540 **Requirements:**  
- Stay within {tokens_limit} tokens
4541     - Eliminate redundancy while preserving essential information
4542     - Eliminate older values when there is newer and updated value for a thing
4543     - Maintain chronological context where important
4544     - Prioritize information with ongoing relevance"
4545
4546 **CRITICAL LENGTH REQUIREMENT:**  
- Your response should be approximately {tokens_limit} tokens
4547     - If your draft is significantly shorter than {tokens_limit} tokens, ADD MORE DETAIL
```

Listing 42: Scratchpad summarization prompt

```
I provide you with a user query and a text chunk.
```

```

4536 You need to decide if the text chunk is necessary for answering user question.
4537 If we need the text chunk to answer the user question, or if the text chunk is part of the answer to user
4538 question return 'yes'.
4539 If the text chunk is noise and not relevant to user question, return 'no'.
4540 Output format: Return only 'yes' or 'no', without any explanation before or after that.
4541
4542 User query: {query} \n\n
4543 Text chunk: {doc_text}

```

Listing 43: Scratchpad noise filtering prompt

```

4544 You are an assistant that MUST answer questions using ONLY the information provided in the context below.
4545 STRICT INSTRUCTIONS:
4546 1. Answer ONLY based on the provided context
4547 2. Do NOT use your internal knowledge
4548 CONTEXT:
4549 <context>
4550 QUESTION:
4551 <question>
4552 ANSWER REQUIREMENTS:
4553 - Be direct and concise
4554 - Only output the answer to the question without any explanation
4555
4556
4557 I LLM USAGE
4558
4559 We used ChatGPT4 as a writing assistant. Specifically, we first drafted the paper and then employed
4560 ChatGPT to refine the text/assist with rephrasing and grammar. The suggestions were manually
4561 reviewed and edited before inclusion in the final version.
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589

```

Listing 44: Answer generation with RAG prompt

I LLM USAGE

We used ChatGPT⁴ as a writing assistant. Specifically, we first drafted the paper and then employed ChatGPT to refine the text/assist with rephrasing and grammar. The suggestions were manually reviewed and edited before inclusion in the final version.

⁴<https://chatgpt.com>