ATOD: An Evaluation Framework and Benchmark for Agentic Task-Oriented Dialogue Systems

Yifei Zhang[1,2*] **Hooshang Nayyeri**[1] **Rinat Khaziev**[1]
Emine Yilmaz[1,3] **Gokhan Tur**[1,4] **Dilek Hakkani-Tür**[1,4] **Hari P Thadakamalla**[1]
[1]Amazon [2]Emory University [3]University College London
[4]University of Illinois Urbana–Champaign
{jimmyzyf, hooshang, rinatk, gokturt, hakkanit, thadakah}@amazon.com
eminey@amazon.co.uk

Abstract

Agentic task-oriented dialogue (TOD) requires systems to track concurrent goals, dependencies, and long-horizon state. We examine goal-lifecycle recovery from fixed dialogue trajectories. ATOD contains 1,000 synthetic dialogues annotated for six advanced-TOD properties, and ATOD-Eval defines metrics for dependency-sensitive completion, memory recall, and proactivity. We implement a symbolic–vector memory evaluator for ATOD-Eval. Among five prompt-only predictors and six matched-backbone memory baselines, our evaluator is the only configuration above 90% in both goal detection F1 and conditional status accuracy on medium dialogues. On complex dialogues, no baseline exceeds it on both metrics; it has the highest conditional status accuracy within the memory-based block and the lowest measured per-turn latency. These experiments assess lifecycle tracking rather than interactive agent task success. A cross-family judge swap and a manual audit with 94.0% agreement provide initial checks on measurement reliability. Code and data will be released at https://github.com/amazon-science/ATOD.

1 Introduction

Task-oriented dialogue (TOD) has traditionally been studied as short, sequential interactions that resolve user requests turn by turn (Rastogi et al., 2020; Budzianowski et al., 2018). LLM-based assistants such as ChatGPT (OpenAI, 2025) and Gemini (Gemini Team, 2025) have raised expectations for more complex interactions. We use **Advanced TOD** to describe settings in which agents coordinate concurrent goals, suspend and resume them, respect evolving dependencies, maintain long-horizon memory, handle asynchronous execution, and act proactively. Table 1 shows an example with interlocked flight, hotel, and restaurant bookings.

[*]Work performed during internship at Amazon.

Table 1: Illustrative interaction and the capabilities it motivates. Utterances are abbreviated for readability; an annotated ATOD excerpt appears in Appendix A.1.

User request	Agent response	Capabilities
Book a flight and hotel.	Offers coordinated flight and hotel options.	Multi-goal concurrency
Remind me to pack.	Retains a reminder for the night before travel.	Long-horizon memory
What is the weather?	Answers a weather query while bookings remain active.	Interleaving
Book dinner after my flight.	Uses arrival time to constrain the dinner reservation.	Dependencies
Arrange an airport ride.	Schedules the ride and offers to send details later.	Asynchronous execution; proactivity
Send details closer to departure.	Retains the deferred follow-up.	Long-horizon memory; asynchrony

Evaluating advanced TOD is harder than slot-filling: a system can complete individual goals yet fail to manage their dependencies, or track status for detected goals while missing others. Existing benchmarks, including MultiWOZ (Budzianowski et al., 2018), SGD (Rastogi et al., 2020), and τ-Bench (Yao et al., 2024), do not provide turn-level lifecycle annotations across this combination of behaviors; TOAD (Liu et al., 2024b) and LUCID (Stacey et al., 2024) cover only parts of it. General LLM judges (Liu et al., 2023; Zheng et al., 2023) and memory benchmarks such as LoCoMo (Maharana et al., 2024) also do not evaluate dependency-aware goal trajectories. To our knowledge, none of the surveyed benchmarks supports fine-grained evaluation of the full capability set studied here.

This paper makes three contributions. **(i) ATOD**, a benchmark of 1,000 synthetic dialogues that jointly covers all six advanced-TOD properties at turn-level granularity, together with its full gen-

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2399–2416
November 6-10, 2026 ©2026 Association for Computational Linguistics

eration pipeline (§4). **(ii) ATOD-Eval**, a multi-dimensional evaluation framework with metrics for dependency-sensitive completion (dGCR), memory recall (MRA), proactivity (PE), and response quality, providing complementary offline and online signals (§6). **(iii)** An **agentic memory evaluator** with a dual symbolic–vector store and a four-stage turn-level pipeline (extract / match / update-or-insert / audit) that balances goal detection and conditional status tracking on fixed complex dialogues at the lowest measured latency among matched-backbone baselines (§5, §8).

2 Related Work

2.1 TOD Systems Evaluation

G-Eval (Liu et al., 2023), AlpacaEval (Dubois et al., 2024), and MT-Bench (Zheng et al., 2023) assess generated text or assistant response quality rather than explicitly tracking goal lifecycles. TOD evaluation spans turn-level user satisfaction (Walker et al., 2000), dialogue-level quality estimation (Deng et al., 2022; Komma et al., 2023), and LLM-based user simulation (Kazi et al., 2024; Davidson et al., 2023). Recent benchmarks (τ-Bench (Yao et al., 2024), AutoEval-ToD (Jain et al., 2025), TD-EVAL (Acikgoz et al., 2025)) focus on inform/success rates rather than advanced-TOD capabilities. Their slot-level or rubric-based metrics are not directly comparable to our turn-level lifecycle metrics, so we discuss them as related settings rather than direct baselines. τ^2-Bench (Barres et al., 2025) evaluates conversational agents in a shared environment where both the agent and user can act through tools. ATOD instead evaluates recovery of turn-level goal lifecycles from fixed dialogues; it does not replace interactive evaluation.

2.2 TOD Datasets and Benchmarks

MultiWOZ (Budzianowski et al., 2018), SGD (Rastogi et al., 2020), RADDLE (Peng et al., 2021), τ-Bench (Yao et al., 2024), and MS-TOD (Du et al., 2025) are human-curated TOD datasets that support dialogue state tracking and task completion but are largely single-session. Synthetic datasets (TOPDIAL (Wang et al., 2023), TOAD (Liu et al., 2024b), LUCID (Stacey et al., 2024), SynthDST (Kulkarni et al., 2024)) add personalization or proactivity but lack joint coverage. In broader agent evaluation, AgentBench (Liu et al., 2024a), T1 (Chakraborty et al., 2025), and AgentChangeBench (Rana et al., 2025) target environments, tool-use planning, and goal-shift adaptation, but none provides turn-level goal lifecycle annotations with dependency-aware metrics.

2.3 Memory for Dialogue Systems

RAG (Lewis et al., 2020), MemoChat (Lu et al., 2023), and MemoryBank (Zhong et al., 2024) support session-level recall; MemGPT (Packer et al., 2023), A-Mem (Xu et al., 2025), Mem0 (Chhikara et al., 2025), and MemOS (Li et al., 2025) introduce structured long-term memory; Chronos (Sen et al., 2026) and SGMem (Wu et al., 2025) use event-based representations; HierTOD (Mo et al., 2024) and TACT (Yoon et al., 2025) target hierarchical and proactive TOD. Benchmarks like LoCoMo (Maharana et al., 2024) and THEANINE (Ong et al., 2025) study memory temporality but not dependency-aware goal evaluation, which ATOD complements.

3 Problem Formulation

3.1 Characteristics of Advanced TOD

Prior TOD benchmarks (Budzianowski et al., 2018; Rastogi et al., 2020; Yao et al., 2024) typically isolate one or two of the behaviors below, while realistic agentic interactions exhibit them jointly. We therefore characterize advanced TOD by six properties that any evaluation must support together: *multi-goal concurrency*, where users pursue multiple objectives in parallel; *interleaving*, where goals are suspended and resumed rather than confined to contiguous turn spans; *dependency management*, where goals carry prerequisite or blocking relations (e.g., PAYMENT depends on BOOKING); *long-horizon memory*, where state is preserved across many turns and goal interruptions; *asynchronous execution*, where delayed goals remain PENDING until external conditions hold, including system-initiated follow-ups; and *proactivity*, where the system offers reminders or suggestions without explicit prompting.

3.2 Task Formulation

We formalize advanced TOD as a corpus $\mathcal{D} = \{(\mathcal{G}_i, \mathcal{C}_i)\}_{i=1}^{N}$, where $\mathcal{C}_i = \{c_{i,t}\}_{t=1}^{T_i}$ is the turn sequence of dialogue i and $\mathcal{G}_i$ is its goal set with explicit inter-goal dependencies. In traditional TOD a goal occupies a contiguous span of turns, whereas an advanced-TOD goal $g \in \mathcal{G}_i$ may span disjoint intervals of $\mathcal{C}_i$ as it is initiated, suspended, and resumed. We therefore represent each goal by

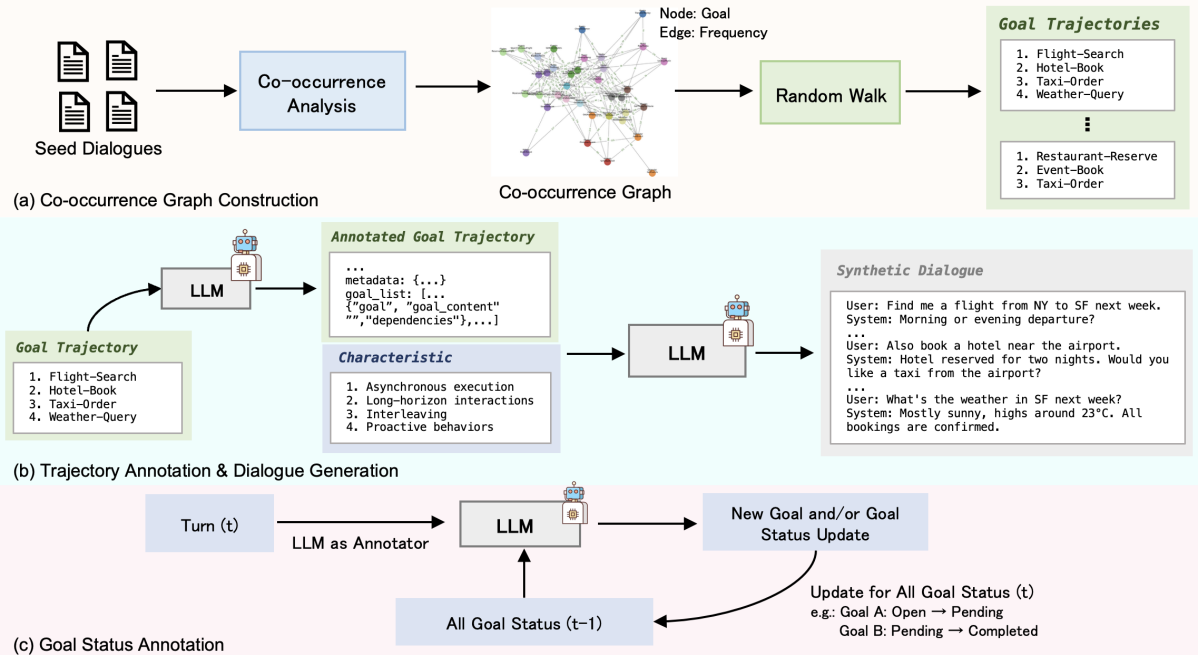


Figure 1: ATOD dataset curation pipeline. (a) **Co-occurrence Graph & Trajectory Sampling**: build a goal co-occurrence graph from seed dialogues and sample diverse multi-goal trajectories via random walks; (b) **Trajectory Annotation & Dialogue Generation**: annotate slot values, dependencies, and complexity, then generate dialogues conditioned on trajectories; (c) **Goal Status Annotation**: label active goals and lifecycle states at each turn for fine-grained tracking of dialogue progress.

its turn-aligned *status trajectory* $\{\text{Status}(g,t)\}_{t=1}^{T_i}$ (e.g., OPEN→PENDING→COMPLETED), capturing a non-contiguous lifecycle that prior single-span representations cannot. The evaluation objective is to measure how well a system maintains and updates these trajectories in both *offline* (over completed dialogues) and *online* (turn by turn) settings.

4 ATOD: A Synthetic Dialogue Dataset and Generation Pipeline

ATOD is a synthetic dataset and generation pipeline for benchmarking the advanced-TOD characteristics in §3.1. It contains 1,000 annotated multi-turn dialogues (428 medium, 572 complex) spanning 16 domains and 36 intents, with an average of 26.6 turns and 6.5 goals per dialogue. Construction includes staged quality checks (Figure 1): a verifier pass filters or regenerates inconsistent instances, and an author manually screens the generated dialogues and annotations for evident defects. Appendix A.6 documents these checks, and Appendix A.2 reports automated ratings. We plan to release ATOD as a zero-shot, test-only benchmark.

4.1 Co-occurrence Graph Construction and Goal Trajectory Sampling

Independent goal sampling yields implausible combinations, while fixed templates limit diversity. We therefore construct a goal co-occurrence graph $G = (V, E)$ from a seed dialogue dataset, where nodes are goals and weighted edges capture empirical co-occurrence frequency. Candidate goal sets $S = \{g_1, \ldots, g_k\}$ are sampled by stratified random walks of varying lengths over G (Figure 1(a)), preserving common goal combinations while adding variation. The procedure is dataset-agnostic; we instantiate it on Schema-Guided Dialogue (SGD) (Rastogi et al., 2020), producing multi-domain trajectories with interleaved, long-horizon goals.

4.2 Annotation of Goal Trajectories and Complexity Categorization

Each sampled set S is instantiated into a concrete trajectory via Claude-Sonnet-4 annotation (Figure 1(b)), yielding slot values, inter-goal dependencies D_S (e.g., prerequisites), and natural-language descriptions. Trajectories are tagged with a complexity label $c(S)$ derived from quantitative attributes (goal count, dependency density) and qualitative factors (interleaving, proactivity opportuni-

ties); see Appendix A.4. Quality control combines duplicate filtering, retries, and a dedicated verifier judgment of trajectory validity and generated-dialogue consistency (Appendix A.6).

4.3 Dialogue Generation

Claude-Sonnet-4 synthesizes each dialogue conditioned on the annotated trajectory $\tau = (S, D_S, c(S))$. Goals, dependencies, and complexity are combined into structured prompts (Appendix A.7) that produce a multi-turn conversation $\mathcal{C} = \{c_t\}_{t=1}^{T}$ with interleaving, asynchrony, proactive assistance, and dependency-aware coordination. This conditioning is central: the generator sees the global trajectory, but the evaluated system must recover its dynamics from dialogue evidence alone.

4.4 Turn-level Goal Status Annotation

A final Claude-Sonnet-4 pass labels each goal at every turn (Figure 1(c)) with an active goal set $\mathcal{A}_t \subseteq S$ and statuses $\text{Status}(g,t) \in \{\text{NOT_MENTIONED}, \text{OPEN}, \text{PENDING}, \text{COMPLETED}, \text{FAILED}, \text{ABANDONED}\}$. The resulting turn-aligned `status_history` supports evaluation of interleaved, asynchronous, and multi-goal trajectories.

4.5 Dataset Coverage

ATOD spans diverse domains and goal complexities, from two-goal cases to interdependent long-horizon workflows. Table 2 highlights the operational dimensions for which comparable annotations or behaviors are identifiable in published benchmark descriptions; ATOD combines these with concurrency and long-horizon goal trajectories. We use the full collection as a held-out benchmark and will release the dataset, generation pipeline, evaluation harness, and prompt templates upon publication. Table 3 reports the evaluation split.

5 Agentic Memory System

We instantiate ATOD-Eval with an *agentic memory system* (Figure 2) that maintains and updates goal trajectories from dialogue text alone. An LLM-only approach re-reads the full history at every turn, while flat memory does not preserve goal structure. Our system instead uses a symbolic store for lifecycle states and dependencies, together with a vector store for retrieval under lexical variation (§5.1). A turn-level pipeline updates both stores (§5.2).

5.1 Dual Memory Store

As shown in Figure 2(b), the memory system maintains (i) a *structured goal database* $\mathcal{D}_{\text{sym}}$ for state and dependency metadata, and (ii) a *semantic vector store* $\mathcal{D}_{\text{vec}}$ (e.g., FAISS (Douze et al., 2024)) for similarity-based retrieval. Each goal stores an identifier, lifecycle state, turn-level status history, textual description, dependency links, and embedding. The symbolic store preserves exact state and dependency information, while the vector store retrieves related goals expressed in different words.

5.2 Turn-Level Processing Pipeline

At each turn t, given utterance u_t and context c_t, the system extracts candidate goals $\mathcal{G}_t = \text{extract}(u_t, c_t)$. Each candidate $g_t^{(i)}$ is then matched to an existing goal and updated, or inserted as a new goal and evolved in the dependency graph; a final auditing step keeps active states consistent:

$$(u_t, c_t) \xrightarrow{\text{extract}} \mathcal{G}_t,$$

$$g_t^{(i)} \xrightarrow{\text{match}} \begin{cases} \text{update}(g^*), & \text{Match} = 1, \\ \text{insert+evolve}(g_t^{(i)}), & \text{Match} = 0. \end{cases}$$

Stage 1: existence checking. For each candidate $g_t^{(i)}$, we retrieve top-k neighbors from $\mathcal{D}_{\text{vec}}$ above a cosine similarity threshold τ, then ask an LLM judge whether any neighbor is semantically equivalent. This two-stage design favors recall in retrieval and precision in verification: τ keeps unrelated goals out of the LLM call, while the judge filters surface-level matches that share keywords but differ in intent. Matched candidates go to Stage 2; novel ones to Stage 3.

Stage 2: update. On a match, the lifecycle is advanced (e.g., PENDING→COMPLETED), slot values are refreshed, and dependencies are preserved.

Stage 3: insert and evolve. On no match, the goal is inserted into $\mathcal{D}_{\text{sym}}$ and $\mathcal{D}_{\text{vec}}$, and an LLM classifies its relation to each top-k neighbor as a soft *link*, a hard *dependency*, or unrelated, updating the dependency graph. This is essential because new goals in advanced TOD are often conditioned on existing ones.

Stage 4: proactive auditing. At each turn, an auditing process inspects OPEN/PENDING goals against the latest context and triggers valid transitions, preventing stale states for asynchronously evolving

Table 2: Selected operational dimensions of representative TOD benchmarks; a coverage summary, not a quality comparison. "Goal Status Anno." marks explicit per-turn lifecycle labels. ✓: present, ✗: absent.

Dataset	Avg. Turns	Async.	Dependency	Interleaving	Proactivity	Goal Status Anno.
MultiWOZ (Budzianowski et al., 2018)	13	✗	✗	✗	✗	✗
SGD (Rastogi et al., 2020)	20	✗	✗	✗	✗	✗
TOPDIAL (Wang et al., 2023)	12	✗	✗	✗	✓	✗
MS-TOD (Du et al., 2025)	7	✗	✓	✓	✗	✓
TOAD (Liu et al., 2024b)	5	✗	✓	✓	✓	✗
LUCID (Stacey et al., 2024)	21	✓	✓	✗	✗	✗
ATOD (Ours)	**27**	✓	✓	✓	✓	✓

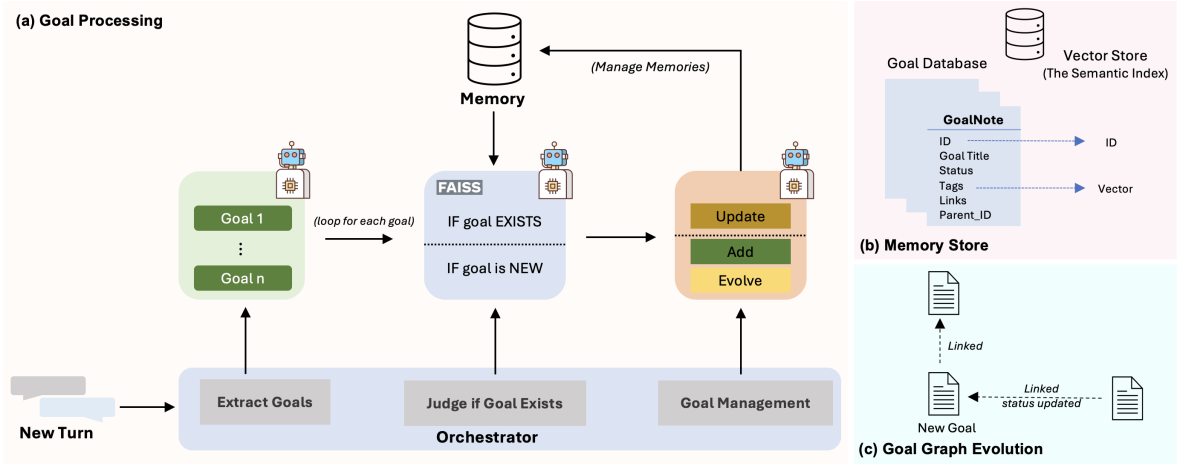


Figure 2: Architecture of the agentic memory system. (a) Turn-level goal extraction, existence checking, updating/inserting, and proactive auditing. (b) Dual memory store with symbolic metadata and semantic embeddings. (c) Dependency graph evolution for newly inserted goals, with explicit links and status transitions.

Table 3: ATOD statistics by dialogue complexity.

Statistic	Medium	Complex	Overall
Dialogues	428	572	1,000
Avg. turns	17.4	33.5	26.6
Avg. goals	3.7	8.5	6.5
With dependencies	18.7%	73.4%	50.0%

goals whose status may change without explicit user restatement.

These four stages maintain dependency-aware goal states throughout the dialogue. ATOD-Eval computes its metrics from these states (§6).

6 Evaluation Metrics and Framework

ATOD-Eval covers three dimensions: task completion and efficiency (§6.1), agentic capability (§6.2), and response quality (§6.3). Task completion alone does not show whether a system recalls earlier constraints or acts proactively, while conversational quality does not show whether dependencies were handled correctly. The framework supports both offline and online evaluation.

6.1 Task Completion and Efficiency

We evaluate whether goals are accomplished and how efficiently they progress through the dialogue. *Dependency-Aware Goal Completion Rate (dGCR).* Conventional completion metrics penalize systems for goals that remain unresolved even when blocked by unmet prerequisites. We therefore evaluate only *decided* goals (terminal status Completed or Failed). Let $\mathcal{U}_{\text{dec}}$ denote reconstructed goals whose final status in $\mathcal{D}_{\text{sym}}$ is Completed or Failed, and let $\mathcal{U}_{\text{comp}} \subseteq \mathcal{U}_{\text{dec}}$ contain those ending in Completed. Then

$$\text{dGCR} = \frac{|\mathcal{U}_{\text{comp}}|}{|\mathcal{U}_{\text{dec}}|}.$$

Dependency sensitivity enters through the lifecycle assignment: goals blocked by unmet prerequisites remain non-terminal and therefore do not enter $\mathcal{U}_{\text{dec}}$. Abandoned goals are also excluded because a cancellation is not a system failure. For example, given g_1 completed, g_2 failed, and g_3 pending because it is blocked by g_2, the denominator is $\{g_1, g_2\}$ and $\text{dGCR} = 1/2$. Because dGCR is computed over reconstructed goals, we read it

alongside Goal Detection F1 and Status Tracking Accuracy (Table 4), which compare against ground truth.

Turns to Completion (NTC). NTC is the average number of turns between goal initiation and a decided terminal status (Completed or Failed), using the same decided-goal set as dGCR; abandoned and unresolved goals are excluded. It complements dGCR with an efficiency signal and should be read relative to dialogue length. Table 6 shows a larger NTC on complex dialogues, where goals are more often suspended and resumed.

6.2 Agentic Capability Metrics

Beyond task success, we assess agentic behaviors such as memory recall and proactivity.

Memory Recall Accuracy (MRA). At each turn where a goal status changes, we construct a retrieval query q_t from the affected goal description and recent context using a fixed template (Appendix C.3). For each $q_t \in \mathcal{Q}$, an LLM judge compares the system retrieval $\hat{M}_t$ against a ground-truth memory snapshot M_t derived from the annotation and returns $r_t \in \{0, 1\}$ based on sufficiency and semantic consistency. We compute $\text{MRA} = |\mathcal{Q}|^{-1} \sum_{q_t \in \mathcal{Q}} r_t$, favoring useful memory over surface-form overlap.

Proactivity Effectiveness (PE). A response is flagged as proactive if it references a goal not in the current user turn but present in the tracked state (e.g., a reminder about a pending booking). For each such response $p_t \in \mathcal{P}$, an LLM judge returns separate binary decisions g_t and h_t for whether the action is *grounded* (supported by context) and *beneficial* (advances an unresolved goal without redundancy). We set $b_t = g_t h_t$, so a response receives credit only when both conditions hold, and compute $\text{PE} = |\mathcal{P}|^{-1} \sum_{p_t \in \mathcal{P}} b_t$. Judge templates are in Appendix C.3.

6.3 Response Quality Metrics

Following Liu et al. (2023); Zheng et al. (2023), *turn-level relevance* scores each response against the local context, current request, and tracked goal state (averaged over turns), while *dialogue-level coherence* scores the full conversation for consistency and natural progression across interleaved goals. The judge returns both scores on a 1–5 scale. We report turn-level relevance as a fraction of the maximum score ($s/5$) and retain the native 1–5 scale for dialogue-level coherence. These measures show whether improved goal management comes at a cost to interaction quality. Templates appear in Appendix C.3.

6.4 Evaluation Framework

ATOD-Eval is evaluator-agnostic: any system producing per-turn goal states can be scored. In our experiments, **offline** evaluation uses completed synthetic dialogues, while **online** evaluation processes those same fixed dialogues turn by turn. Online denotes incremental lifecycle tracking, not interaction with a user or tool environment. The framework could support monitoring of deployed agents, but end-to-end agent rollouts and real-world task success are not evaluated here.

7 Experimental Setup

Evaluation covers capability, metric validity, and efficiency. We report goal detection F1 and status tracking accuracy at the final state and across normalized dialogue progress, Pearson's r and Spearman's ρ between dGCR and the other metrics, and per-turn latency and token usage. Implementation details appear in Appendix C.1.

Baselines. All baselines are evaluated zero-shot on the same 1,000-dialogue set. The prompt-only judges are DeepSeek-V3.2 (DeepSeek-AI, 2025), Kimi-K2.5 (Kimi Team, 2026), MiniMax-M2.5, Mistral-Large-3, and GLM-5 (GLM-5 Team, 2026); each infers goal status from the full dialogue context without external memory. We also adapt Mem0 (Chhikara et al., 2025), MemoChat (Lu et al., 2023), MemoryBank (Zhong et al., 2024), RAG (Lewis et al., 2020), A-Mem (Xu et al., 2025), and LLM-Rsum (Wang et al., 2025) to our goal-status schema by mapping their memory units to our goal representation.

Architecture-isolated comparison. Every memory-based method, including ours, uses the same backbone LLM and the same `all-MiniLM-L6-v2` embedding model; only the memory architecture varies. We use Claude-Sonnet-4 as the backbone for all memory-based evaluators (details in Appendix C.1). The prompt-only judges are the LLMs under test and indicate what prompting alone achieves across model scales. No shared external judge scores these rows. The existence threshold and top-k for our evaluator were selected on a disjoint 50-dialogue development set and frozen before test evaluation; we do not claim equal hyperparameter tuning across methods.

Run variability. The reported memory experiments

Table 4: Goal detection F1 and conditional status accuracy (%) by dialogue complexity. Status accuracy is measured over detected goals; interpret jointly with detection quality. Shaded rows indicate memory-based methods sharing the same backbone LLM. **Bold**: our method; underline: best baseline in each column.

Category	Method	Medium		Complex	
		Goal Det. F1	Cond. Status Acc.	Goal Det. F1	Cond. Status Acc.
LLM-based	DeepSeek-V3.2	52.12	81.72	45.96	65.16
	Kimi-K2.5	66.70	92.90	66.90	82.80
	MiniMax-M2.5	72.20	90.20	66.60	83.90
	Mistral-Large-3	60.51	89.85	55.73	85.98
	GLM-5	71.42	93.52	64.84	83.42
Memory-based	Mem0 (Chhikara et al., 2025)	76.80	75.60	61.10	71.20
	MemoChat (Lu et al., 2023)	80.38	73.88	58.07	66.83
	MemoryBank (Zhong et al., 2024)	82.56	94.23	76.86	78.50
	RAG (Lewis et al., 2020)	85.59	94.85	87.13	77.83
	A-Mem (Xu et al., 2025)	87.30	89.50	62.00	80.60
	LLM-Rsum (Wang et al., 2025)	93.83	89.47	89.13	69.95
	Ours	**91.92**	**92.31**	**86.49**	**84.28**

use temperature 0 and seed 42. These settings do not guarantee identical outputs from API-served models or quantify uncertainty over sampled dialogues. We treat the reported margins as point estimates, not tests of statistical significance.

Mitigating LLM circularity. Because the evaluator and the metric judges may share a model family, we provide two checks: a cross-family judge swap with Qwen3-32B (Yang et al., 2025) and a manual audit of judge decisions on a stratified subset (§8.4).

8 Results

8.1 Evaluation of the Memory System

Table 4 reports goal detection F1 and status tracking accuracy by complexity. The two columns must be read jointly: status accuracy is computed only over detected goals, so a system can miss many goals yet still look competitive on the second column. DeepSeek-V3.2 (52.12/81.72 on medium) is the clearest case. Mistral-Large-3 (55.73/85.98 on complex) and RAG (85.59/94.85 on medium) show the same pattern more subtly. Among the evaluated configurations, only our evaluator exceeds 90% on both axes for medium dialogues. On complex dialogues, no baseline exceeds ours (86.49/84.28) on both metrics. Mistral-Large-3 has higher conditional status accuracy (85.98), while LLM-Rsum and RAG have higher detection F1. Our complex conditional status accuracy is highest within the matched-backbone memory block, exceeding A-Mem by 3.68 points. Our claim concerns the joint result; several baselines lead in individual columns.

Architecture-isolated block. All memory-based methods share the same backbone and embedding (§7), so this block compares memory architectures. LLM-Rsum obtains higher complex detection F1 than ours (89.13 vs. 86.49) but lower complex status accuracy (69.95 vs. 84.28); A-Mem and Mem0 also drop sharply in complex detection (87.30→62.00 and 76.80→61.10). This result is compatible with our design rationale, but component-level causal attribution requires further ablations.

Capacity reference: prompt-only judges. Among the five prompt-only models in Table 4, the strongest F1 trails the best memory-based F1 by more than 20 points in each split. Table 5 separately controls for backbone capacity by comparing our evaluator with its prompt-only backbone.

Table 5: Same-backbone comparison between prompt-only judging and our agentic memory evaluator.

Method	Medium		Complex	
	F1	Acc.	F1	Acc.
Prompt-only	78.95	93.26	75.58	84.26
Ours	**91.92**	**92.31**	**86.49**	**84.28**
Δ	+12.97	−0.95	+10.91	+0.02

Online behavior. Figure 3 shows incremental tracking performance as the fixed dialogue is revealed. This evaluates the online replay setting (§6.4); it does not measure adaptation during interactive agent rollouts.

No baseline dominates both axes. LLM-Rsum detects more goals but trails our complex status

tracking and incurs more than 7× the per-turn latency (§8.2), while RAG's higher medium status accuracy comes with lower detection F1. Our evaluator balances detection and conditional status tracking at the lowest measured per-turn latency among these memory methods.

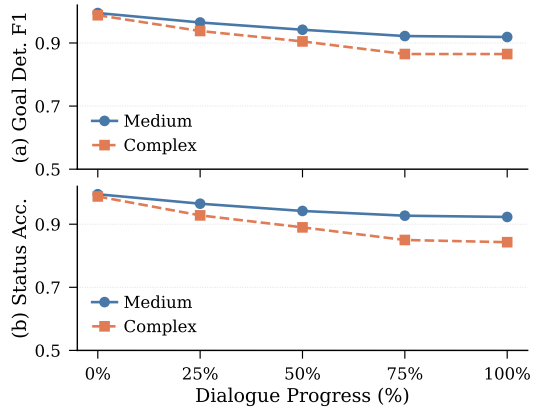


Figure 3: Goal detection F1 (top) and status tracking accuracy (bottom) versus normalized dialogue progress on medium and complex dialogues. The leftmost point is the first available turn-level checkpoint after progress normalization, not an empty-context evaluation.

8.2 Efficiency Analysis

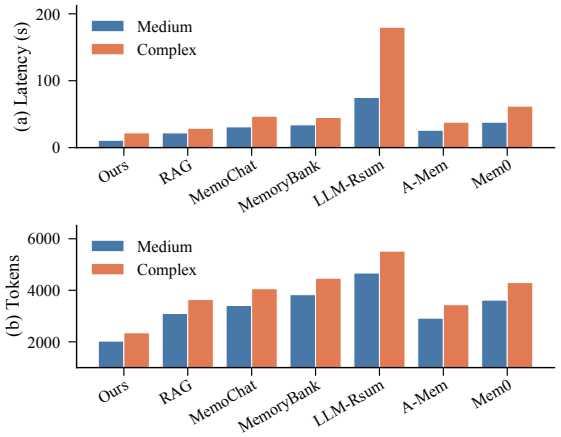


Figure 4: Per-turn update latency (top) and token usage (bottom) across methods.

Figure 4 reports per-turn latency and token usage. Our evaluator is most efficient on both axes (<25s per turn on complex dialogues; 2,030/2,350 tokens medium/complex). RAG and A-Mem are next (22–38s), Mem0 incurs moderate overhead (38–62s), and LLM-Rsum reaches 180s on complex dialogues with more than twice the tokens because it re-summarizes the full history each turn. Among the evaluated methods, our evaluator lies on the accuracy–latency Pareto frontier.

8.3 Metric Validity Analysis

Table 6: Average diagnostic scores on fixed synthetic dialogues, not success rates from interactive agent rollouts. Turn-level Relevance is the native 1–5 judge score divided by 5. All other bounded metrics are on a 0–1 scale; Dialogue-level Coherence remains on its native 1–5 scale, and NTC is a turn count.

Metric	Medium	Complex
dGCR	0.967	0.930
NTC	7.04	10.50
MRA	0.913	0.743
PE	0.619	0.586
Turn-level Relevance	0.752	0.766
Dialogue-level Coherence	4.40	4.45

Table 6 summarizes the fixed dialogues and their reconstructed states. Complex dialogues have lower MRA and larger NTC, while relevance and coherence are slightly higher. We report these metrics separately because they capture different aspects of the dialogues, although these aggregate values alone do not show that the metrics are non-redundant.

Table 7: Correlation of proposed metrics with dGCR, reported as Pearson's r and Spearman's ρ under Medium and Complex settings.

Metric	Medium		Complex	
	r	ρ	r	ρ
NTC	+0.08	+0.16	+0.20	+0.05
MRA	+0.75	+0.60	+0.44	+0.43
PE	−0.05	−0.03	+0.16	+0.12
Turn-level Relevance	+0.22	+0.29	+0.08	+0.09
Dialogue-level Coherence	−0.11	−0.08	+0.13	+0.25

The metrics distinguish different failure modes by construction: two dialogues can end with the same completed and failed goals, and thus the same dGCR, while taking different numbers of turns (NTC) or differing in recall of earlier constraints (MRA). Successful completion also does not establish that unsolicited actions were useful (PE). These distinctions motivate the metric set but are not empirical evidence of independence.

Table 7 reports correlations with dGCR as a task-completion anchor. Because dGCR is computed from reconstructed states (§6.1), these values measure internal alignment rather than independent

ground truth. MRA correlates most strongly with dGCR ($r = 0.75$ on medium and 0.44 on complex dialogues), while PE has weak correlations ($r = -0.05$ and 0.16). Low correlation with dGCR alone does not establish independent information: it may also reflect measurement noise or restricted variation. In particular, the 81.6% audit agreement for beneficial proactivity limits the interpretation of PE. We use PE as a diagnostic, not a task-success measure or a basis for the main method-ranking claim; we do not claim probabilistic calibration. A full pairwise analysis and incremental prediction of independently measured user success are needed to test redundancy more directly.

8.4 Validity of LLM Judges

Manual audit. We manually reviewed 2,880 LLM judge decisions from 100 stratified dialogues. Agreement with this audit is 94.0% overall (Table 8), but falls to 81.6% for beneficial proactivity, indicating that this subjective metric should be interpreted cautiously. This audit is an initial check on metric judgments, not an independent multi-annotator validation or manual screening of generated dialogues. The protocol is detailed in Appendix B.1.

Table 8: Agreement between LLM judge decisions and a single-author manual audit on 100 stratified dialogues (2,880 items). Overall agreement is weighted by item count.

Metric Dimension	# Items	Audit Agr.
Goal Status Transition	1,850	94.2%
Dependency Relation	650	96.8%
Proactivity (grounded)	190	95.3%
Proactivity (beneficial)	190	81.6%
Overall	2,880	**94.0%**

Cross-family judge swap. Replacing the default backbone judge with Qwen3-32B (Yang et al., 2025) shifts our scores by $-1.82/-1.71$ on medium F1/Acc and $-3.39/+1.02$ on complex. The swap checks whether scores are sensitive to the judge family; it does not rerank all baselines.

9 Conclusions

We introduced **ATOD** and **ATOD-Eval** for evaluating interleaved, dependency-aware TOD with turn-level lifecycle annotations, together with a symbolic–vector memory evaluator. The evaluator balances detection and tracking on complex dialogues at the lowest measured per-turn latency among matched-backbone baselines. The judge swap and manual audit support the metric judgments, although the audit also shows that beneficial proactivity remains subjective. Future work will extend the benchmark to real interactions and isolate individual memory components.

Table 9: Judge swap for our evaluator (F1 / status accuracy).

Judge	Medium		Complex	
	F1	Acc.	F1	Acc.
Default LLM Judge	91.92	92.31	86.49	84.28
Qwen3-32B	90.10	90.60	83.10	85.30
Δ	-1.82	-1.71	-3.39	$+1.02$

Limitations

Construction and validation. LLM-based construction and evaluation risk circularity. A verifier pass and author screening check dataset quality; a cross-family judge swap and separate manual audit check metric judgments. Neither provides independent multi-annotator validation.

Uncertainty and attribution. Point estimates do not quantify dialogue-sampling or API-call variability. Correlations with dGCR do not establish metric independence, and co-occurring properties prevent attribution to individual behaviors. A top-k sweep and same-backbone comparison assess the memory layer, but proactive auditing and the dependency graph lack separate ablations.

Scope. ATOD comprises held-out synthetic English, text-only dialogues derived from SGD domains. It measures goal-trajectory recovery rather than live user success, without personalization or multimodal inputs.

Ethical Considerations

ATOD uses synthetic dialogues generated from public SGD goal structures, not deployed-system conversations. Despite staged quality control, synthetic generation does not guarantee privacy or eliminate model biases. Proactivity can become intrusive or manipulative without user controls, and judgments of benefit vary across cultures and user populations. Downstream evaluation should therefore account for user preferences and these limitations. The dataset is intended solely for research and will be released under the CC BY-NC 4.0 license.

References

Emre Can Acikgoz, Carl Guo, Suvodip Dey, Akul Datta, Takyoung Kim, Gokhan Tur, and Dilek Hakkani-Tür. 2025. TD-EVAL: Revisiting task-oriented dialogue evaluation by combining turn-level precision with dialogue-level comparisons. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 113–132. Association for Computational Linguistics.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026. Association for Computational Linguistics.

Amartya Chakraborty, Paresh Dashore, Nadia Bathaee, Anmol Jain, Anirban Das, Shi-Xiong Zhang, Sambit Sahu, Milind Naphade, and Genta Indra Winata. 2025. T1: A tool-oriented conversational dataset for multi-turn agentic planning. In *Advances in Neural Information Processing Systems*, volume 38, pages 115282–115327.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour, and Yi Zhang. 2023. User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*.

DeepSeek-AI. 2025. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Yang Deng, Wenxuan Zhang, Wai Lam, Hong Cheng, and Helen Meng. 2022. User satisfaction estimation with sequential dialogue act modeling in goal-oriented conversational systems. In *Proceedings of the ACM Web Conference 2022*, pages 2998–3008.

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

Yiming Du, Bingbing Wang, Yang He, Bin Liang, Baojun Wang, Zhongyang Li, Lin Gui, Jeff Z. Pan, Ruifeng Xu, and Kam-Fai Wong. 2025. Bridging the long-term gap: A memory-active policy for multi-session task-oriented dialogue. *arXiv preprint arXiv:2505.20231*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

GLM-5 Team. 2026. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*.

Arihant Jain, Purav Aggarwal, Rishav Sahay, Chaosheng Dong, and Anoop Saladi. 2025. AutoEval-ToD: Automated evaluation of task-oriented dialog systems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10133–10148. Association for Computational Linguistics.

Taaha Kazi, Ruiliang Lyu, Sizhe Zhou, Dilek Hakkani-Tür, and Gokhan Tur. 2024. Large language models as user-agents for evaluating task-oriented-dialogue systems. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 913–920. IEEE.

Kimi Team. 2026. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Abishek Komma, Nagesh Panyam Chandrasekarasastry, Timothy Leffel, Anuj Goyal, Angeliki Metallinou, Spyros Matsoukas, and Aram Galstyan. 2023. Toward more accurate and generalizable evaluation metrics for task-oriented dialogs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 186–195. Association for Computational Linguistics.

Atharva Kulkarni, Bo-Hsiang Tseng, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Hong Yu, and Shruti Bhargava. 2024. SynthDST: Synthetic data is all you need for few-shot dialog state tracking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1988–2001. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang,

Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. 2025. MemOS: An operating system for memory-augmented generation (MAG) in large language models. *arXiv preprint arXiv:2505.22101*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024a. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. Association for Computational Linguistics.

Yinhong Liu, Yimai Fang, David Vandyke, and Nigel Collier. 2024b. TOAD: Task-oriented automatic dialogs with diverse response styles. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8341–8356. Association for Computational Linguistics.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. MemoChat: Tuning LLMs to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870. Association for Computational Linguistics.

Lingbo Mo, Shun Jiang, Akash Maharaj, Bernard Hishamunda, and Yunyao Li. 2024. HierTOD: A task-oriented dialogue system driven by hierarchical goals. *arXiv preprint arXiv:2411.07152*.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seungwon Hwang, Dongha Lee, and Jinyoung Yeo. 2025. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8631–8661. Association for Computational Linguistics.

OpenAI. 2025. Introducing ChatGPT agent: bridging research and action. `https://openai.com/index/introducing-chatgpt-agent/`. Accessed September 11, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2021. RADDLE: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4418–4429. Association for Computational Linguistics.

Manik Rana, Calissa Man, Anotida Expected Msiiwa, Jeffrey Paine, Kevin Zhu, Sunishchal Dev, Vasu Sharma, and Ahan M R. 2025. AgentChangeBench: A multi-dimensional evaluation framework for goal-shift robustness in conversational AI. *arXiv preprint arXiv:2510.18170*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Sahil Sen, Elias Lumer, Anmol Gulati, and Vamse Kumar Subbiah. 2026. Chronos: Temporal-aware conversational agents with structured event retrieval for long-term memory. *arXiv preprint arXiv:2603.16862*.

Joe Stacey, Jianpeng Cheng, John Torr, Tristan Guigue, Joris Driesen, Alexandru Coca, Mark Gaynor, and Anders Johannsen. 2024. LUCID: LLM-generated utterances for complex and interesting dialogues. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 56–74. Association for Computational Linguistics.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3-4):363–377.

Jian Wang, Yi Cheng, Dongding Lin, Chak Tou Leong, and Wenjie Li. 2023. Target-oriented proactive dialogue systems with personalization: Problem formulation and dataset curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1143. Association for Computational Linguistics.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193.

Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. 2025. SGMem: Sentence graph memory for

long-term conversational agents. *arXiv preprint arXiv:2509.21212*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-MEM: Agentic memory for LLM agents. *arXiv preprint arXiv:2502.12110*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Yejin Yoon, Yuri Son, Namyoung So, Minseo Kim, Minsoo Cho, Chanhee Park, Seungshin Lee, and Taeuk Kim. 2025. Beyond task-oriented and chitchat dialogues: Proactive and transition-aware conversational agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13280–13306. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

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

A ATOD Dataset: Statistics, Examples, and Construction Details

A.1 Annotated Dialogue Example

Table 10 presents a representative medium-complexity dialogue from ATOD, illustrating interleaving, dependency management, and goal abandonment. The dialogue involves four goals across three domains (Calendar, Services, Flights). Goal 2 (salon appointment) depends on Goal 1 (availability check); Goal 3 (SF flight) is abandoned in turn 7 when the user changes plans to Chicago (Goal 4), which is booked while the salon appointment is confirmed. The status column reproduces the released turn-level annotations.

A.2 Synthetic Dataset Quality Analysis

As aggregate evidence for the construction-time verifier, we assess synthetic dialogues along five dimensions using an LLM quality rater following Liu et al. (2023). Table 11 provides automated quality-control scores, complementing the author-performed manual screening described in Appendix A.6.

A.3 Goal Extraction, Co-occurrence Graph Statistics, and Sampling Strategy

We first extract goal sequences from the SGD dataset, where each sequence is an ordered list of user goals (domain–intent pairs) within a dialogue. Table 12 summarizes the extraction results. All 10,739 sequences are multi-domain, with an average length of 3.90 goals and a range of 2–8 goals. These seed sequences span 16 unique domains and 37 unique intents; the retained ATOD evaluation set covers 36 intents (§4).

We then construct a goal co-occurrence graph where each node is a unique goal and edges represent co-occurrence within the same sequence. Table 13 shows its statistics. The graph contains 52 nodes and 396 edges, forming a single connected component with relatively high density (0.2986) and average degree (15.23), indicating frequent goal co-occurrence across dialogues. This structure supports a diverse sampling of multi-goal trajectories, including high-degree hubs (up to 29) and rare goals (degree as low as 2).

We sample goal trajectories using the stratified random walks described in §4.1. Each sampled walk induces a connected subgraph that is then screened against the desired complexity criteria (§A.4), ensuring diversity in goal count, domain coverage, and dependency patterns.

A.4 Complexity Criteria

Our pipeline (§4) uses a two-category complexity system (*medium* vs. *complex*), combining quantitative thresholds with qualitative LLM analysis for balanced distribution. Table 14 shows the criteria based on goals, turns, domains, and advanced agentic behaviors.

For the categorization process, we follow a three-step procedure. First, *goal sampling* draws trajectories under a two-category distribution (default: 65% medium, 35% complex; the retained evaluation set contains 428 medium and 572 complex dialogues after classification and quality control). Second, *annotation* enriches sampled goals with slots, dependencies, and realistic characteristics. Third, *hybrid classification* assigns complexity using pre-defined rules combined with LLM anal-

Table 10: Annotated ATOD dialogue excerpt (medium complexity, 4 goals, 3 domains). Status changes shown per turn; only selected turns are displayed, and omitted turns include intermediate system responses. Goals: G1=check availability, G2=book salon (depends on G1), G3=book SF flight, G4=book Chicago flight.

Turn	Speaker	Utterance	Status Changes
1	User	I need to check Sarah Johnson's availability on October 12th, between 2 and 5 PM.	G1→OPEN
2	System	Let me look that up for you right away.	G1→PENDING
3	User	While you're checking that, I also need a one-way flight from Boston to San Francisco on June 12th.	G3→OPEN
4	System	I'll work on both. Sarah has availability from 2–4 PM. I'm looking for morning Delta flights for you.	G3→PENDING
5	User	Great—I'd like to book a haircut with her at Bella Salon for 3:30 PM on October 12th.	G2→OPEN
6	System	I'll book that appointment. Meanwhile, I found several Boston–SF flights.	G2→PENDING
7	User	Actually, change the flight to Chicago O'Hare instead, same date. Is there a United flight?	G3→ABAND., G4→OPEN
8	System	Your salon appointment is confirmed for Oct 12 at 3:30 PM. Searching Chicago flights now.	G2→COMPL., G4→PENDING
11	User	Yes, please book that United flight for me. And could you still check on those Delta flights to San Francisco?	G4→COMPL.
15	User	No need to book the San Francisco flights, I just needed the information. So I have the salon on Oct 12 and the Chicago flight on June 12?	—

Table 11: LLM-based quality ratings on a 1–5 Likert scale.

Dimension	Medium	Complex
Coherence	4.04	3.92
Fluency	5.00	5.00
Consistency	4.42	4.04
Relevance	4.58	4.62
Naturalness	4.04	4.04

Table 12: Summary statistics for extracted goal sequences.

Statistic	Value
Total Goal Sequences	10,739
Avg. Sequence Length	3.90
Length Range	2–8
Unique Domains	16
Unique Intents	37

ysis, considering quantitative factors (goal count, domain diversity, dependency structures), qualitative factors (goal interdependence and coordination complexity), and realistic dialogue requirements such as interleaving and proactivity needs. These labels are assigned during construction, before evaluator testing, and are shared by every method. The overlapping goal/turn ranges are therefore not sufficient to reconstruct the labels alone; reproducibility requires the construction attributes and saved final label for each dialogue.

Table 13: Summary statistics for the co-occurrence graph.

Statistic	Value
Total Nodes (Unique Goals)	52
Total Edges (Co-occurrences)	396
Graph Density	0.2986
Average Degree	15.23
Max Degree	29
Min Degree	2

A.5 Annotated Trajectories and Metadata Specification

To represent complex goal structures and agentic behaviors in ATOD, we define a formal schema for dialogue trajectories and metadata (§4.2), shown in Listing 1. The schema captures interleaved goals, slot-filling states, and explicit inter-goal dependencies that arise in advanced TOD settings. Metadata fields encode global dialogue attributes and execution characteristics (e.g., interleaving, proactivity, and asynchronous actions), which are later used to condition dependency-aware and turn-level evaluation signals. Each goal entry specifies its intent, slots, and dependency relations, enabling systematic analysis of goal initiation, suspension, and resumption across multi-goal interactions.

A.6 ATOD: Quality Control

We apply layered screening during construction. Claude-Sonnet-4 performs trajectory annotation,

```
{
  "dialogue_id": "string",
  "complexity_class": "medium | complex",
  "classification_method": "pre_defined |
    model_based",
  "dependency_label": "boolean",
  "defectiveness_label": "boolean",
  "metadata": {
    "num_goals": "integer",
    "estimated_turns": "integer",
    "async_execution": "boolean",
    "interleaving": "boolean",
    "proactivity": "boolean"
    },
  "goal_list": [
    {
      "id": "string",
      "domain": "string",
      "intent": "string",
      "slots": ["string", ...],
      "slot_values": {
        "slot_name_1": "value1",
        "slot_name_2": "value2"
      },
      "dependencies": ["goal_id", ...],
      "content": "string",
      "core_content": "string"
    }
    // ...more goals
  ]
}
```

Listing 1: Annotation schema for dialogue trajectories and goal-level metadata.

Table 14: Construction-time complexity criteria. Boolean fields use ✓ for true/present and ✗ for false/absent. "Failure / obstacle scenario" corresponds to the `defectiveness_label` field in Listing 1; it denotes an intentional task obstacle or failure outcome, not an annotation defect. Overlapping ranges are resolved by the hybrid classification procedure described below.

Attribute	**Medium**	**Complex**
Goals	2–8	7+
Turns	8–35	30+
Asynchronous execution	✓	✓
Interleaving	✓	✓
Dependencies	≤ 2	≥ 2
Proactivity	✗	✓
Failure / obstacle scenario	✗	✓

dialogue generation, turn-level status annotation, and both verification stages. The stages use separate calls, but all calls use the same model family and are not independent. Before dialogue generation, a verifier judge checks each annotated trajectory for clear goals, plausible slots, complete fields, and consistent dependencies. After dialogue generation and status annotation, another verifier checks whether the dialogue follows its trajectory, whether lifecycle annotations agree with the utterances, and whether the interaction is coherent and plausible. Flagged instances are retried or filtered. An author then screens the generated dialogues and status annotations for the same failure types. Because this screening was not independently replicated, we treat it as quality control rather than an inter-annotator validation study. The pre-generation verifier prompt is shown below.

Table 15: Prompt template for quality assessment of annotated trajectories.

Quality Assessment Prompt
`You are a quality judge for annotated goal trajectories.`
Input: TRAJECTORY ({num_goals} goals, {complexity} complexity): {goals_text}
Task: Assess whether this trajectory is ready for dialogue generation. Check:
- Goal descriptions are clear and specific - Slot values are realistic (no placeholders)
- All required fields are present - Annotations are logically consistent
Output format: Respond with exactly one word: `PASS` or `FAIL`

A.7 Dialogue Generation Prompt

Table 16 shows the prompt used for LLM-based dialogue generation in §4.3. Placeholders are filled from the annotated trajectory metadata (§A.5).

Table 16: Prompt template for dialogue generation.

Dialogue Generation Prompt Template

`Generate a realistic task-oriented dialogue between USER and SYSTEM.`

Requirements: Complexity: {complexity} Length: {estimated_turns} turns Goals: {goal_descriptions} Attributes: {agentic_attrs}
{combined_guidance} {outcome_guidance}

Dialogue Structure: (1) User introduces goals naturally; (2) System works under realistic constraints; (3) Natural obstacles and preference changes may occur; (4) Dialogue ends at a natural stopping point; (5) Goal completion may be partial or conditional.

Natural Conversation Patterns: Users express needs as they arise; system handles practical constraints; users may add, revise, or abandon goals; availability or technical limitations may surface.

Format: Alternating USER/SYSTEM turns, starting with USER.

A.8 Goal Status Annotation Prompt

Table 17 shows the prompt used for turn-level goal status annotation in §4.4. Listing 2 provides a sample annotated dialogue.

Table 17: Prompt template for turn-level goal status annotation.

Goal Status Annotation Prompt Template

`You are tracking goal status in a task-oriented dialogue. Analyze only the current turn and update statuses based on what actually happens.`

Current Turn: {last_turn} **Goals and Current Statuses:** {goal_descriptions}

Status Meanings:

- `NOT_MENTIONED`: not yet appeared - `OPEN`: mentioned, no action - `PENDING`: system working

- `COMPLETED`: successfully finished - `FAILED`: failed - `ABANDONED`: user cancelled

Critical Rules: (1) Change status only if something definitive occurs; (2) `PENDING` → only `COMPLETED`/`FAILED`/`ABANDONED`; (3) Preserve status if no clear change.

Terminal States: {goal_id: current_status, . . . } **JSON Template:** {json_template}

Instruction: Respond with only the JSON, updating only goals whose status clearly changes.

```
{
  "dialogue_id": "...",
  "complexity_class": "complex",
  "metadata": {
    "num_goals": ...,
    "num_turns": ...,
    "async_execution": true,
    "interleaving": true,
    "proactivity": true
  },
  "goal_list": [...],
  "turns": [
    {
      "turn_id": 1,
      "speaker": "USER",
      "utterance": "I need to book a hotel in
    Chicago.",
      "goal_status_changes": [
        {"goal_id": "g1", "new_status": "open"}
      ],
      "all_goals": {
        "g1": "open",
        "g2": "not_mentioned",
        "g3": "not_mentioned"
      }
    }
    // ... remaining turns omitted
  ]
}
```

Listing 2: Sample annotated ATOD dialogue with turn-level status tracking.

B Robustness and Sensitivity Analyses

This appendix details the single-author manual audit of metric judgments and top-k retrieval sensitivity. The cross-family judge-swap results are reported in §8.4. All analyses use the same 1,000-dialogue test set unless otherwise noted.

B.1 Single-Author Manual Audit of Metric Judgments

Protocol. We sample a stratified subset of 100 dialogues (50 medium, 50 complex) and extract 2,880 individual judge decisions across four dimensions: goal status transitions, dependency relations, proactivity grounding, and proactivity benefit. Under resource constraints, one author manually inspects each decision against the dialogue context, reference annotation, and metric definition, marking whether the LLM judgment is acceptable. Table 18 reports agreement between the LLM decisions and this manual audit. Since no independent second annotator or adjudication was available, we do not report inter-annotator agreement and interpret the audit as a preliminary validity check.

Table 18: Agreement between LLM judge decisions and a single-author manual audit on 100 stratified dialogues (2,880 items). Overall agreement is weighted by item count.

Metric	# Items	Audit Agr.
Goal Status Transition	1,850	94.2%
Dependency Relation	650	96.8%
Proactivity (grounded)	190	95.3%
Proactivity (beneficial)	190	81.6%
Overall	2,880	94.0%

Findings. The single-author audit agrees with the LLM judge on 94.0% of items overall. Agreement is highest for dependency relations (96.8%), while proactivity *benefit* is lowest (81.6%), consistent with the subjectivity of that dimension. These results provide an initial error check for the metric judges; they are distinct from the construction-time screening in Appendix A.6 and are not independent multi-annotator validation.

B.2 Top-k Retrieval Sensitivity

The top-k retrieval hyperparameter controls how many neighbors are considered during existence checking (Stage 1 in §5.2). We sweep $k \in \{1, 3, 5, 7, 9\}$ to assess sensitivity. All other settings remain at their default values (Table 19).

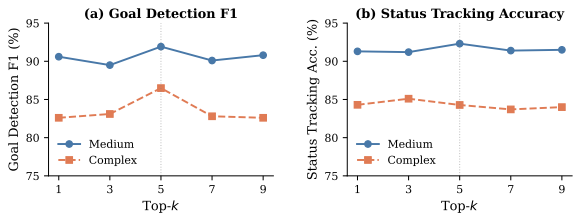


Figure 5: Top-k retrieval sensitivity. Goal detection F1 (left) and status tracking accuracy (right) as functions of top-k on the 1,000-dialogue test set. The default k=5 is marked by the dotted vertical line.

On medium dialogues, goal detection F1 varies by less than 2.5 percentage points across the tested values of k; conditional status accuracy varies by less than 1.5 points in each split. The default k=5 achieves the best overall performance, particularly on complex dialogues where it outperforms other settings by 3–4 points in goal F1. Smaller values (k=1, 3) risk missing relevant neighbors during existence checking, while larger values (k=7, 9) introduce noise from less relevant candidates. The larger variation in complex-dialogue F1 indicates greater sensitivity to retrieval breadth in that split.

C Evaluator Implementation and Prompt Templates

C.1 Implementation Details

The memory system uses Claude-Sonnet-4 (accessed via the Amazon Bedrock API) for goal extraction, existence checking, relationship analysis, and proactive status auditing. Retrieval uses `all-MiniLM-L6-v2` sentence embeddings (384 dimensions) indexed with FAISS. An LLM classifies links among the top-k retrieved goals rather than relying on a fixed similarity cutoff. Table 19 lists the settings used for all reported results.

The default memory-based runs use Claude-Sonnet-4 via the Amazon Bedrock API (temperature 0, seed 42). Every memory-based baseline (Mem0, MemoChat, MemoryBank, RAG, A-Mem, LLM-Rsum) uses the same backbone LLM and `all-MiniLM-L6-v2` embedding model as our evaluator. The prompt-only baselines in Table 4 are the LLMs under test. This setup controls for model capacity within the memory-based block. We examine potential circularity between the evaluator and metric judges through a Qwen3-32B judge swap and the manual audit in §8.4; Appendix B.1 gives the audit protocol. FAISS retrieval runs on a single NVIDIA A10 GPU, while LLM calls account for most of the end-to-end latency.

Both hyperparameters were selected on a small held-out development set of 50 dialogues drawn from the ATOD generation pipeline using the same procedure as the test split, then frozen before the 1,000-dialogue evaluation. The existence-match threshold τ=0.40 was tuned to favor recall over precision during goal matching: a lower threshold ensures semantically related but differently phrased goals are linked, with the subsequent LLM verification step filtering false positives. Top-k=5 retrieval balances coverage against prompt length; most dialogues in the development set contained fewer than 10 concurrent goals, so k=5 retrieves the majority of relevant candidates without exceeding context limits. Sensitivity to k on the full test set is reported in Appendix B.2.

C.2 Agentic Memory System Templates

The memory system uses three LLM prompt templates: goal extraction from individual turns, turn-level status classification, and goal-graph updates for inter-goal links and dependencies. The prompts list five active labels. The sixth state, NOT_MENTIONED, is assigned implicitly to refer-

Table 19: Implementation settings for the agentic memory system.

Setting	Value
Primary LLM Judge	Backbone LLM (see text)
Embedding Model	`all-MiniLM-L6-v2` (384-d)
Vector Store / Index Type	FAISS `IndexFlatIP`
Similarity Metric	Cosine (IP on L2-normalized)
Top-k Retrieval	5
Existence-Match Threshold τ	0.40
Proactive Auditing Frequency	Every turn

ence goals that have not yet appeared and does not require an LLM call.

Table 20: Prompt template for goal extraction in the agentic memory system.

Goal Extraction Prompt

`Extract user goals from this conversation turn. Use standardized core_content patterns.`

Conversation Turn:

User: {user_utterance}

System: {system_response}

Core Content Patterns (examples):

- "book hotel" — hotels, rentals - "book flight" — flights - "book ticket" — bus, concert, train

- "check account" — balance, account info - "search restaurant" — find restaurants

Status Labels: `OPEN`, `PENDING`, `COMPLETED`, `FAILED`, `ABANDONED`

Output format (JSON array):

```
[{"goal_content": "...", "core_content": "...", "status": "OPEN"}, ...]
```

C.3 Evaluation Prompt Templates

To make the evaluation protocol explicit, we also provide the prompt templates used for LLM-based judgment of memory recall, proactive behavior, and response quality. In all cases, the judge is given the dialogue context, the system output under evaluation, and the corresponding benchmark reference derived from ATOD annotations.

Table 21: Prompt template for per-turn goal status classification.

Goal Status Classification Prompt

`Analyze this conversation turn and classify the status of the specific goal below.`

Goal to classify: "{goal_content}"

Conversation Turn:

User: {user_utterance} System: {system_response}

Status Definitions:

- `OPEN`: mentioned, no action yet - `PENDING`: system processing

- `COMPLETED`: successfully achieved - `FAILED`: explicitly failed - `ABANDONED`: cancelled by user

Transition Examples:

"book a flight" → OPEN "which dates?" → PENDING "flight booked" → COMPLETED

"no flights available" → FAILED "never mind" → ABANDONED

Output format (JSON): `{"status": "STATUS"}`

Table 22: Prompt template for goal graph evolution.

Goal Evolution Prompt

`Analyze relationships between a new goal and existing related goals.`

New Goal: Content: {new_goal.content} Core Content: {new_goal.core_content}

Related Goals (top-k by semantic similarity): {related_goals_context}

Task: For each related goal, determine the relationship type:

- `"link"`: semantically related but independent

- `"dependency"`: new goal depends on the related goal

- `"none"`: no significant relationship

Output format (JSON):

```
{"goal_id_1": "relationship_type",
 "goal_id_2": "relationship_type"}
```

Table 23: Prompt template for Memory Recall Accuracy evaluation.

Memory Recall Accuracy Prompt

`You are evaluating whether a retrieved memory output correctly matches the benchmark memory state.`

Dialogue Context: {dialogue_context}

Retrieval Query: {memory_query}

Ground-Truth Memory State: {gold_memory_json}

Retrieved Memory Output: {predicted_memory}

Judging Criteria:

- Retrieved output contains information required to answer the query

- Goal statuses, slot values, and historical facts are semantically consistent with gold state

- No contradiction is introduced

Output format (JSON): `{"match": 0 or 1}`

Table 24: Prompt template for Proactivity Effectiveness evaluation.

Proactivity Effectiveness Prompt

`You are evaluating whether a system's proactive action is contextually grounded and genuinely helpful.`

Dialogue Context: {dialogue_context}

Current System Action: {system_action}

Relevant Goal State: {goal_state_json}

Judging Criteria:

- The action is not explicitly requested by the user in the current turn
- The action is grounded in prior dialogue and memory state
- The action advances an unresolved or dependency-unlocked goal, or provides a timely reminder
- The action is not irrelevant, redundant, or distracting

Output format (JSON):

```
{"grounded": 0 or 1, "beneficial": 0 or 1}
```

Table 25: Prompt template for Response Quality evaluation.

Response Quality Prompt

`You are evaluating the response quality of a task-oriented dialogue system.`

Dialogue Context: {dialogue_context}

Response Under Evaluation: {system_response}

Goal State / Dialogue Reference: {goal_state_or_full_dialogue}

Turn-Level Relevance: addresses current request; consistent with tracked goals; useful and natural

Dialogue-Level Coherence: globally consistent; interleaved goals progress coherently; no contradictions

Output format (JSON):

```
{"turn_quality": 1-5, "dialogue_quality": 1-5}
```