ReacTOD: Bounded Neuro-Symbolic Agentic NLU for Zero-Shot Dialogue State Tracking

Yanjun Lin* **Zimo Xiao*** **Kartik Natarajan** **Mahesh Sankaranarayanan**
Niraj Nawanit **Rakshit Parashar** **Austin Zhang** **Karthik Konaraddi**
Rishita Mote **Wei Niu**

Amazon

{linyj, zimoxiao, kartikn, sankmahe, nawanit, chillorb, auszhang, kartkon, rmote, niuwei}@amazon.com

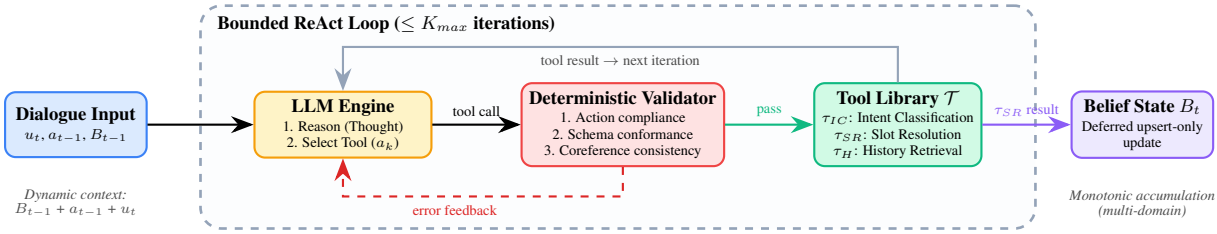

Figure 1: Overview of the ReacTOD bounded neuro-symbolic agentic architecture.

Abstract

Task-oriented dialogue systems—handling transactions, reservations, and service requests—require predictable behavior, yet the moderately-sized LLMs needed for practical latency are prone to hallucination and format errors that cascade into incorrect actions (e.g., a hotel booked for the wrong date). We propose **ReacTOD**, a bounded neuro-symbolic architecture that reformulates NLU as discrete tool calls within a self-correcting ReAct loop governed by deterministic validation. A bounded ReAct loop enables iterative self-correction, improving accuracy by up to 9.3 percentage points over single-pass inference on MultiWOZ. A symbolic validator enforces action compliance, schema conformance, and coreference consistency on every dialogue state update, achieving a 93.1% self-correction rate on intercepted errors and producing structured execution traces. Incremental state prediction and on-demand history retrieval keep prompts compact, empirically improving instruction adherence in parameter-constrained models. On MultiWOZ 2.1, ReacTOD achieves a new zero-shot state-of-the-art: gpt-oss-20B reaches 52.71% joint goal accuracy, surpassing the previous best by 14 percentage points, while Qwen3-8B achieves 47.34% with only 8B parameters. On the Schema-Guided Dialogue (SGD) benchmark, ReacTOD with Claude-Opus-4.6 achieves 80.68% JGA under fully end-to-end evaluation with predicted domains, and Qwen3-32B reaches 64.09%—demonstrating cross-benchmark generalization without task-specific training data.

1 Introduction

Task-Oriented Dialogue (TOD) systems deployed in production environments—handling hotel bookings, restaurant reservations, and transport arrangements—require predictable, verifiable NLU behavior: an incorrectly resolved slot value (e.g., a check-in date inferred from the wrong turn) propagates to downstream API calls, producing silent failures or incorrect transactions. This need for reliable outputs has historically driven the dominance of discriminative, pipelined Natural Language Understanding (NLU) architectures, where extractive models like BERT perform Intent Classification (IC) and Slot Resolution (SR) as sequential tasks over fixed label sets. While these pipelines offer sub-second latency and high predictability, they depend on large volumes of domain-specific labeled data and require retraining to accommodate new intents or language variation—limiting zero-shot generalization.

To move beyond static ontologies, recent work has shifted toward generative, LLM-driven prompting for zero-shot NLU. Frameworks such as FnC-TOD (Li et al., 2024) reframe domain logic as executable functions, leveraging in-context learning. However, single-pass generative approaches suffer from probabilistic variance and faithful hallucina-

*Equal contribution.

Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026), pages 342–352
July 4, 2026 ©2026 Association for Computational Linguistics

tions, in which the model confidently infers unstated entity values to complete a schema, posing significant risks in production dialogue pipelines where incorrect state values propagate to downstream API calls. Moreover, the linguistic complexities of real dialogue (cross-turn coreference, implicit value acceptance, non-linear domain switching) require multi-step reasoning and dynamic context retrieval. Unbounded agentic frameworks can address these phenomena in principle, but their reliance on open-ended reasoning loops and frontier-scale models introduces impractical latency and computational overhead.

We argue that improving the reliability of LLM-based DST does not primarily require larger models, but rather stronger architectural control over the reasoning process. Our key insight is that LLM errors in dialogue state tracking are predominantly local and correctable—a misformatted time value or an invalid slot name, rather than a fundamental misunderstanding of the dialogue. Based on this insight, we propose **ReacTOD**, a hybrid neuro-symbolic NLU architecture that decomposes NLU into discrete tool calls within a bounded ReAct-style reasoning loop, reducing the per-step burden on the LLM. A deterministic validator intercepts all tool outputs before any state mutation, enforcing action compliance, schema conformance, and coreference consistency—enabling the model to self-correct from structured error feedback rather than requiring re-processing of the entire dialogue. This constrained design lowers the reasoning capacity required per step, enabling parameter-efficient models (e.g., Qwen3-8B) to achieve robust agentic state tracking without frontier-scale compute. The architectural details are presented in Section 3 and illustrated in Figure 1.

We evaluate our architecture on MultiWOZ 2.1 and the Schema-Guided Dialogue (SGD) benchmark in a zero-shot setting—no labeled dialogues, no fine-tuning, no in-domain examples—using dynamic schema injection across five backbone models ranging from 8B to frontier scale. On MultiWOZ 2.1, ReacTOD with gpt-oss-20B achieves 52.71% Joint Goal Accuracy (JGA), surpassing the previous zero-shot state-of-the-art (FnCTOD with GPT-4, 38.71%) by 14 percentage points (pp). Even Qwen3-8B reaches 47.34% JGA—exceeding FnCTOD with the 4× larger Qwen3-32B (40.36%)—demonstrating that the gains stem from architectural design rather than model scale. On SGD, ReacTOD with Claude-Opus-4.6 achieves 80.68% JGA, outperforming a reproduced SRP baseline (45.20%) that uses gold domain labels, and the ReAct loop contributes up to 11.82 pp over single-pass inference, confirming cross-benchmark generalization without task-specific training data. In summary, our contributions are threefold:

1. **Bounded Agentic Reasoning:** We introduce a constrained ReAct architecture that decomposes NLU into discrete tool calls with iterative self-correction, enabling error recovery beyond what single-pass inference achieves—with ablations showing gains of up to 9.3 percentage points.
2. **Deterministic Validation:** We design a symbolic validator that gates all state mutations, enforcing action compliance, schema conformance, and coreference consistency to catch common LLM errors (e.g., invalid tool calls, hallucinated slot names) before they reach the dialogue state.
3. **Parameter-Efficient Zero-Shot DST:** We demonstrate that incremental state prediction and on-demand context retrieval keep prompts compact, enabling models as small as 8B parameters to surpass prior zero-shot baselines built on larger LLMs. ReacTOD establishes a new state-of-the-art on MultiWOZ 2.1 and strong cross-benchmark performance on SGD without task-specific training data, requiring only a machine-readable domain schema.

2 Related Work

2.1 From Pipelined NLU to Generative State Tracking

Early enterprise NLU treated IC and SR as sequential classification tasks over fixed label sets. JointBERT (Chen et al., 2019) unified both through a shared encoder, while lightweight variants targeted resource-constrained devices (Huang et al., 2022). These discriminative pipelines offer determinism but are tied to predefined vocabularies, degrading on out-of-distribution inputs and precluding zero-shot generalization. Generative sequence-to-sequence models relaxed this constraint: TRADE (Wu et al., 2019) enabled cross-domain slot transfer via pointer-generator networks, and SimpleTOD (Hosseini-Asl et al., 2022) and SOLOIST (Peng et al., 2021) consolidated the pipeline into a single autoregressive objective.

However, these approaches still required substantial in-domain fine-tuning, leaving zero-shot adaptability an open challenge.

2.2 LLM-Driven Prompting and Knowledge Distillation

Instruction-tuned LLMs enabled zero-shot DST via in-context learning. D3ST (Zhao et al., 2022) and Lu et al. (2024) replaced schema notations with natural language descriptions, SERI-DST (Lee and Lee, 2024) dynamically retrieved dialogue examples, and FnCTOD (Li et al., 2024) established the zero-shot state-of-the-art by treating domains as executable functions. However, single-pass generative approaches suffer from probabilistic variance and faithful hallucinations—confidently inferring unstated entity values to complete a schema (Ji et al., 2023)—posing reliability risks in production dialogue systems where incorrect state values propagate to downstream API calls. Knowledge distillation approaches (Xu et al., 2025b; Aguirre et al., 2024) reduce inference costs by training smaller student models on LLM-generated data, but hardcode the schema into model weights, sacrificing zero-shot flexibility.

2.3 Tool-Augmented Agents and Neuro-Symbolic Integration

ReAct (Yao et al., 2023) demonstrated that LLMs can interleave reasoning traces with task-specific actions, but deploying unbounded agents in TOD introduces reliability risks. Elizabeth et al. (2025) showed that ReAct-based agents frequently underperform structured baselines on task success metrics despite producing fluent responses, and while LLMs exhibit self-refinement capacity (Madaan et al., 2023), unconstrained self-evaluation is susceptible to confirmation bias without external grounding. These findings motivate our core design principle: confining the LLM to narrowly scoped tool-mediated subtasks and gating all state mutations through a deterministic symbolic validator.

3 Methodology

Our key insight is that LLM errors in dialogue state tracking are predominantly local and correctable—a misformatted time value or an invalid slot name, rather than a fundamental misunderstanding of the dialogue. By decomposing NLU into discrete tool calls within a bounded ReAct loop, we enable the agent to receive structured feedback from a deterministic validator and repair such mistakes iteratively, without requiring the model to re-process the entire dialogue context. This principle motivates a bounded neuro-symbolic architecture that separates the Natural Language Understanding (NLU) pipeline into isolated, verifiable tasks, orchestrates them via a constrained ReAct-style state machine, and gates all state mutations through deterministic validation.

3.1 Problem Formulation and Architecture Overview

Given a dialogue turn consisting of user utterance u_t, prior system action a_{t-1}, persistent belief state B_{t-1}, and intents i_{t-1}, our architecture formulates NLU as a sequential, tool-augmented policy $\pi(a \mid s)$ whose action space $\mathcal{A}$ is restricted to tool library $\mathcal{T}$:

$$a_k \sim \pi(\cdot \mid u_t, a_{t-1}, B_{t-1}, i_{t-1}, H_{<k}), \quad a_k \in \mathcal{T} \tag{1}$$

where $H_{<k}$ denotes the agent's action–observation trace up to reasoning step k. Unlike single-pass approaches that jointly predict intent, slots, and schema formatting, this decomposition isolates each sub-task into a separate, verifiable tool invocation. As illustrated in Figure 2, tool calls are first validated by a deterministic validator V (§3.4) before execution, and only validated τ_{SR} results are permitted to update B_t, transforming state tracking from unbounded sequence generation into a bounded neuro-symbolic verification loop.

Incremental Belief State Prediction. To reduce per-turn complexity, the model predicts only incremental updates ΔB_t (newly mentioned or changed slots), with the full state recovered as $B_t = B_{t-1} \cup_{upsert} \Delta B_t$.

Dynamic Context Construction. Prior work has shown that shorter, focused prompts improve instruction adherence in smaller LLMs (Xu et al., 2025a). Motivated by this finding, we construct each tool's context window dynamically rather than providing the full schema and dialogue history upfront. Intent definitions are included in the system prompt, but slot descriptions are injected only for the active intent at the time of slot resolution, avoiding irrelevant schema noise. Conversation history is not included by default; instead, the agent retrieves it on demand via a dedicated history tool (τ_H) only when coreference resolution requires prior context. This lazy loading strategy keeps prompts minimal—typically containing only the active schema, current belief state B_{t-1}, previous system utterance a_{t-1},

and current user utterance u_t—reducing cognitive load for parameter-efficient models.

3.2 Neuro-Symbolic Task Subdivision

Following the empirical finding of FnCTOD (Li et al., 2024) that decomposing NLU into separate function calls improves LLM accuracy over joint prediction, we functionally separate Intent Classification (IC) and Slot Resolution (SR) into distinct generative invocations (see Figure 2). This subdivision constrains the LLM's search space per inference step, reducing cognitive load and allowing parameter-efficient models to achieve reasoning parity with monolithic frontier models.

3.2.1 Schema-Driven Intent Classification (IC)

The IC module maps the user utterance u_t to a target intent $i_t \in \mathcal{I}$, where $\mathcal{I}$ represents the dynamic ontology of the enterprise system. To achieve zero-shot generalization—requiring no labeled dialogues or fine-tuning—we include the domain schema in the LLM's context window at inference time, yielding the IC policy:

$$i_t \sim \pi_{IC}(\cdot \mid u_t, B_{t-1}, a_{t-1}) \tag{2}$$

To ensure deterministic out-of-domain (OOD) routing, $\mathcal{I}$ mandates the inclusion of non-transactional classes, such as $i_{fallback}$.

The Short-Circuit Efficiency Logic: If the IC policy predicts a non-transactional intent, the system executes a programmatic short-circuit, bypassing the SR module entirely. This prevents the computational inefficiency of forcing expensive entity extraction on conversational acknowledgments.

3.2.2 Slot Resolution (SR)

Conditioned on a transactional intent i_t, the SR module extracts the relevant entity set E_t. For each extracted slot, the model produces a tuple $e = \langle v_{raw}, v_{norm} \rangle$, where v_{raw} is the surface form as it appears in the dialogue and v_{norm} is the canonical, system-compliant normalization (e.g., mapping "tmrw" to an ISO-8601 date). To keep the generative task focused, only the slot definitions associated with the active intent i_t are injected into the prompt, excluding irrelevant schema from other domains.

To handle **Implicit Acceptance**—where a user confirms a system proposal without restating the entity (e.g., the system asks "How about the Hilton?" and the user responds "Yes")—the previous system utterance a_{t-1} is included in the SR context window, allowing the model to ground extractions in the system's prior turn.

3.3 Bounded Agentic Control Flow

The agent operates within a single bounded ReAct loop with a maximum of K_{max} iterations. The system prompt instructs the agent to follow a prescribed tool-calling sequence: first invoke τ_{IC} to classify the user's intent, then invoke τ_{SR} to extract and resolve slots for the predicted intent. While this ordering is enforced via prompt instruction rather than programmatic constraint, the deterministic validator (§3.4) provides a hard guarantee: τ_{SR} cannot successfully terminate the loop unless the extracted slots pass all validation checks for the predicted intent. This creates an effective control flow where the agent self-organizes into an IC-first, SR-second pattern, with the validator acting as the structural enforcement mechanism.

The agent may re-invoke τ_{IC} if extracted slots are inconsistent with the predicted intent, re-invoke τ_{SR} to correct slot values after validation feedback, or invoke τ_H to retrieve conversation history for coreference resolution. If the iteration limit K_{max} is reached without successful validation, the system forces graceful degradation by returning a fallback response.

3.4 Deterministic Validation and Self-Correction

A central design principle of our architecture is that the LLM proposes actions but never directly modifies system state. We introduce a Deterministic Validator V, a deterministic gatekeeper that evaluates any proposed tool call a_k and the current state s_k.

$$V(a_k, s_k) \rightarrow \begin{cases} (True, \emptyset) & \text{if safe} \\ (False, \varepsilon_{feedback}) & \text{if violated} \end{cases} \tag{3}$$

The validator operates via deterministic symbolic checks to avoid the infinite regress problem commonly associated with LLM-as-a-judge frameworks.

The Symbolic Validator: This layer executes computationally cheap, $O(1)$ algorithmic checks organized into three categories: ***action compliance*** (rejecting calls to undefined tools, enforcing prerequisite ordering such as requiring τ_{IC} before τ_{SR},

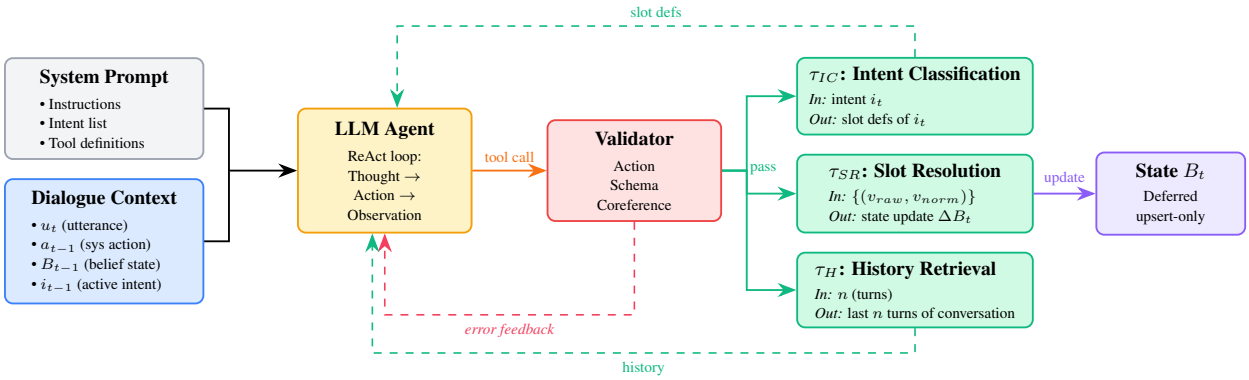

Figure 2: Tool definitions and data flow in ReacTOD. The validator checks all tool calls before execution; τ_{IC} and τ_H results feed back to the agent for the next step; only validated τ_{SR} output updates the belief state B_t. Invalid calls trigger error feedback to the LLM for self-correction.

and suppressing duplicate tool calls), ***schema conformance*** (validating intent and slot names against the domain ontology, and enforcing value constraints including regex matching for dates, times, and numbers against canonical formats and enumeration membership checks), and ***coreference consistency*** (flagging generic references such as "restaurant" that indicate unresolved entities requiring history retrieval). When a check fails, the validator generates a structured error message describing the violation and injects it back into the agent's context, prompting the model to self-correct on the next iteration. For example, given the utterance "I need to be there on time for my reservation," if the model extracts a non-temporal value for slot `taxi-arriveby`, the validator rejects the output with feedback such as "invalid format for slot `taxi-arriveby`: expected HH:MM," steering the model to re-examine the utterance and extract the correct time reference. To prevent infinite looping, the system enforces a maximum iteration threshold K_{max}. If this limit is reached without successful validation, the system forces graceful degradation by returning a fallback response, safely terminating the loop without further LLM inference.

Deferred State Updates. The belief state B_t is a persistent, multi-domain table that supports non-linear context switching across domains. Crucially, B_t is never mutated during the agent's reasoning iterations—updates are deferred until the validator confirms the extracted slots pass all checks. This isolation ensures that rejected intermediate outputs from self-correction attempts cannot corrupt the persistent state, preserving a consistent view of the dialogue for subsequent turns. Updates follow a strict upsert-only protocol: new slot-value pairs are inserted, existing slots are overwritten on user revision (e.g., "Actually, make it for 3 people"), but values are never deleted (explicit removal uses a designated null value). This monotonic accumulation ensures that cross-domain state is preserved even as the conversation switches between domains.

4 Experiment

4.1 Evaluation Datasets

To evaluate the DST performance of our proposed methodology, we use two widely adopted multi-domain task-oriented dialogue benchmarks.

MultiWOZ 2.1. The Multi-Domain Wizard-of-Oz 2.1 dataset (Budzianowski et al., 2018; Eric et al., 2020) contains human-to-human task-oriented conversations spanning five domains with cross-domain coreference. We evaluate on the 1,000-dialogue test split, using version 2.1 for comparability with prior work despite known annotation issues in later versions (Zang et al., 2020; Ye et al., 2022).

Schema-Guided Dialogue (SGD). The SGD dataset (Rastogi et al., 2020) spans 26 services across 16 domains. Its schema-driven design—where each service defines its own intents, required/optional slots, and result slots—closely mirrors real-world API-driven systems. We evaluate on the 4,201-dialogue test split.

4.2 Metrics

We adopt Joint Goal Accuracy (JGA) as our primary metric. For MultiWOZ 2.1, following the TRADE protocol (Wu et al., 2019), we report *overall JGA* (exact match across all active domains

simultaneously) and *domain-specific JGA* (per-domain exact match). For SGD, we follow the official evaluation protocol (Rastogi et al., 2020) with per-service JGA averaged across services. Non-categorical slot values are compared using fuzzy token-sort matching.

4.3 Experiment Setups

4.3.1 Model Configuration

To ensure our methodology is deployable in real-world applications with practical latency and cost constraints, we primarily evaluate on open-source LLMs with fewer than 32 billion parameters. We additionally require models to possess sufficient reasoning capability to operate within the ReAct loop and make appropriate action decisions. Based on these criteria, we evaluate the following models: Qwen3-8B, Qwen3-32B (Team, 2025), gpt-oss-20B (OpenAI, 2025), and Gemma3-12B (Team et al., 2025). To further assess the upper bound of our system's performance, we also include Claude-Opus-4.6 as a high-capacity reference model.

All models are evaluated with a temperature of 0.0 to encourage deterministic and reproducible outputs and share a uniform maximum ReAct iteration cap of $K_{\max} = 6$. Thinking mode is disabled for Qwen3 models and Gemma3-12B (text-based ReAct prompting); gpt-oss-20B uses native thinking with low effort; Claude-Opus-4.6 interleaves free-form reasoning with native tool calls. Qwen3-32B and Claude-Opus-4.6 are served via Amazon Bedrock; remaining models are hosted locally on A100 GPUs with vLLM. All five backbone models are evaluated on both MultiWOZ 2.1 and SGD.

4.3.2 Schema Configuration

MultiWOZ. We source schema information from MultiWOZ 2.2, which provides formal intent definitions and slot metadata absent from 2.1. We merge intents within each domain into a single intent and augment each slot with a type annotation (*Categorical*, *Time*, *Number*, or *Freeform Text*) for the Deterministic Validator. Slots such as service name and food type are treated as freeform text to reflect real-world conditions where exhaustive enumeration is impractical.

SGD. We derive the schema programmatically from the official test set schema definitions (Rastogi et al., 2020). We retain separate intents per service and promote result-only slots to sibling search intents when needed. Each slot is annotated with a role—*Required* or *Filter*—derived from the schema's `required_slots`, `optional_slots`, and default values. Purely informational slots (appearing only in `result_slots`) are excluded from the model's slot list to reduce hallucination. Date and time slots are normalized to canonical formats; other values are treated as freeform text.

4.4 Baseline

We compare our method **ReacTOD** against representative methods in the LLM-based dialogue state tracking paradigm. We include **SERI-DST** (Lee and Lee, 2024), which dynamically retrieves in-context dialogue examples to guide the LLM at inference time. Most critically, we compare against **FnCTOD** (Li et al., 2024), which represents the previous zero-shot state-of-the-art by treating domain logic as executable functions and constraining the LLM to produce structured JSON arguments. To ensure a fair and up-to-date comparison, we additionally re-evaluate FnCTOD using its publicly available implementation with the same backbone LLMs used in our experiments (Qwen3-32B, gpt-oss-20B), isolating the contribution of our architectural design from differences in underlying model capability. We further include **DistDST** (Xu et al., 2025b), a distillation-based method requiring offline fine-tuning (included for reference only). For SGD, we compare against **SRP** (Safa and Şahin, 2025), which employs self-refined prompts with per-domain isolated chat sessions and gold domain labels. The original paper reports 88.70% JGA with GPT-4-Turbo; our reproduction using the published SRP codebase with Claude-Opus-4.6 yields 45.20% JGA after removing result-only slot predictions that the SRP prompt incorrectly extracts from system utterances. The dominant error is hallucination of informational attributes (e.g., `car_name`, `venue`, `price`) caused by the prompt's instruction to track system-mentioned values. We report our reproduced result in Table 1.

5 Results

We first report the overall zero-shot DST performance of **ReacTOD** against LLM-based baselines (§5.1), then conduct an ablation study isolating the ReAct loop's contribution (§5.2), followed by an efficiency and validator activation analysis (§5.3).

5.1 Zero-Shot DST Performance

We first compare ReacTOD against baselines across models of varying capacity to assess whether

		MultiWOZ 2.1		SGD
Approach	**Model**	**Overall JGA**	**Domain Avg. JGA**	**Avg. Svc. JGA**
SERI-DST	GPT-3.5	N/A	60.58%	—
FnCTOD	GPT-4	38.71%	62.59%	—
FnCTOD	Llama2-13B*	37.67%	59.54%	—
FnCTOD	Qwen3-32B**	40.36%	63.10%	—
FnCTOD	gpt-oss-20B**	34.03%	58.56%	—
DistDST	Llama-3.1-8B*	45.20%	—	—
SRP†	Claude-Opus-4.6††	—	—	45.20%
ReacTOD	Qwen3-8B	47.34%	68.11%	57.31%
ReacTOD	Qwen3-32B	51.53%	**71.83%**	**64.09%**
ReacTOD	gpt-oss-20B	**52.71%**	71.77%	62.92%
ReacTOD	Gemma3-12B	45.11%	66.35%	55.58%
ReacTOD	Claude-Opus-4.6***	61.29%	78.34%	80.68%

* Models are fine-tuned in the original work.
** Re-evaluated with the public FnCTOD code using 5-shot prompts.
*** Reference only; impractical for production latency.
† Uses gold domain labels and per-domain isolated sessions; ReacTOD predicts domains end-to-end.
†† Reproduced with the SRP codebase and prompts; the original paper reports 88.70% with GPT-4-Turbo, which we were unable to reproduce (see §4.4).

Table 1: Zero-shot DST on MultiWOZ 2.1 and SGD. Best results in **bold**; — = not evaluated.

the architectural gains hold independently of model scale. Table 1 presents the zero-shot DST results on MultiWOZ 2.1 and SGD. ReacTOD consistently outperforms all prompting baselines across both benchmarks. Even with Qwen3-8B, ReacTOD achieves 47.34% Overall JGA and 68.11% Domain Average JGA on MultiWOZ, surpassing FnCTOD with GPT-4 (38.71% / 62.59%) despite using a far smaller backbone. Our fair re-evaluation of FnCTOD with Qwen3-32B (40.36%) further confirms that the gains stem from architectural design, not model capacity—ReacTOD with the smaller Qwen3-8B exceeds it by nearly 7 percentage points. The strongest MultiWOZ result comes from gpt-oss-20B at 52.71% Overall JGA and 71.77% Domain Average JGA, surpassing even the fine-tuned DistDST baseline (45.2%). On SGD, which presents greater complexity (26 services across 16 domains, fine-grained schema distinctions), ReacTOD with Claude-Opus-4.6 achieves **80.68%** average service JGA, outperforming the reproduced SRP baseline (45.20%) despite operating under harder conditions—predicted domains and a single session, versus SRP's gold domain labels and per-domain isolation. All production-viable models (8B–32B) also surpass the SRP baseline.

Two cross-model comparisons reveal that reasoning capability matters more than parameter count within our framework. First, gpt-oss-20B matches the larger Qwen3-32B on MultiWOZ (52.71% vs. 51.53%) despite Qwen3-32B having 60% more parameters, by leveraging native thinking mode and built-in tool calling for deeper per-step reasoning. On SGD, however, Qwen3-32B pulls ahead (64.09% vs. 62.92%), suggesting that the advantage of native tool calling diminishes when schema complexity increases and text-based reasoning suffices. Table 3 corroborates the cost of native thinking: gpt-oss-20B consumes 448 avg / 1611 P99 output tokens per turn versus 150 / 366 for Qwen3-32B. Second, Qwen3-8B overtakes the larger Gemma3-12B on both benchmarks (MultiWOZ: 47.34% vs. 45.11%; SGD: 57.31% vs. 55.58%) despite trailing without the ReAct loop (39.29% vs. 42.73%, Table 2). This crossover aligns with the Qwen3 technical report (Team, 2025), which shows Qwen3 outperforming Gemma-3 at comparable sizes on reasoning and agent benchmarks. These findings suggest that a model's capacity for structured multi-step reasoning is a stronger predictor of agentic DST performance than raw parameter count.

5.2 Ablation Study

We next isolate *bounded agentic reasoning* by ablating the bounded ReAct loop. We conduct experiments under two methodological conditions. In addition to the full **ReacTOD** pipeline, we evaluate a decomposed variant in which Intent Classification (IC) and Slot Resolution (SR) are issued as two independent LLM calls, without the iterative ReAct loop or thinking capability, serving as a direct ablation of the agentic reasoning component.

Table 2 shows that the ReAct reasoning loop consistently yields substantial improvements across

Model	Variant	MultiWOZ 2.1		SGD
		Overall JGA	Domain Avg. JGA	Avg. Svc. JGA
Qwen3-8B	w/o ReAct Loop	39.29%	61.67%	45.49%
	ReacTOD	47.34% (+8.05)	68.11% (**+6.44**)	57.31% (**+11.82**)
Qwen3-32B	w/o ReAct Loop	46.35%	68.24%	56.36%
	ReacTOD	51.53% (+5.18)	**71.83%** (+3.59)	**64.09%** (+7.73)
gpt-oss-20B	w/o ReAct Loop	43.39%	66.44%	56.42%
	ReacTOD	**52.71%** (**+9.32**)	71.77% (+5.33)	62.92% (+6.50)
Gemma3-12B	w/o ReAct Loop	42.73%	64.10%	53.34%
	ReacTOD	45.11% (+2.38)	66.35% (+2.25)	55.58% (+2.24)
Claude-Opus-4.6	w/o ReAct Loop	59.69%	77.89%	73.49%
	ReacTOD	61.29% (+1.60)	78.34% (+0.45)	80.68% (+7.19)

Table 2: Ablation study on MultiWOZ and SGD. "w/o ReAct Loop" = IC + SR as independent calls; (+) = gain in pp.

all backbone models. For Qwen3-8B, the full ReacTOD pipeline achieves 47.34% Overall JGA and 68.11% Domain Average JGA, compared to 39.29% and 61.67% for the decomposed variant—a gain of **8.05 pp** in Overall JGA. A similar pattern holds for gpt-oss-20B, where the full pipeline achieves 52.71% / 71.77% versus 43.39% / 66.44% without the ReAct loop, a gain of **9.32 pp**. The largest absolute improvement on MultiWOZ is observed with gpt-oss-20B, suggesting that models with native reasoning and tool-calling capabilities benefit most from the iterative self-correction mechanism. These results confirm that the iterative agentic reasoning loop is a critical architectural component of ReacTOD, enabling the model to self-correct slot resolution errors that single-pass inference cannot recover from.

The SGD ablation results confirm cross-benchmark generalization, with even larger gains than on MultiWOZ. Qwen3-8B gains 11.82 pp—the largest absolute improvement on either benchmark—suggesting that smaller models benefit disproportionately when the schema is more complex and the validator has more opportunities to catch errors. Gemma3-12B shows the smallest loop gain on both benchmarks, consistent with its weaker performance on reasoning and agentic benchmarks (Team et al., 2025; Team, 2025): the self-correction mechanism requires the model to interpret error feedback and revise its output, a capability that scales with reasoning proficiency.

5.3 Efficiency Analysis

Finally, we analyze computational overhead to confirm the bounded loop remains practical. We report two hardware-agnostic proxy metrics—**LLM calls per turn** and **output tokens per turn**—rather than wall-clock latency, which varies with deployment configuration. Table 3 summarizes both metrics. Across all models, the median (P50) turn completes in exactly 2 LLM calls (the mandatory IC and SR invocations), with averages ranging from 1.83 to 2.19. P99 values of 3–6 calls confirm the loop remains bounded even in tail cases; Gemma3-12B exhibits the highest P99 (6.00), consistent with its weaker single-pass accuracy requiring more correction attempts.

Token consumption reflects differences in reasoning strategy: Qwen3 and Gemma3 models produce compact outputs (150–166 avg tokens) via text-based ReAct prompting, while gpt-oss-20B consumes roughly 3× more (448 avg, 1611 P99) due to native chain-of-thought tokens before each tool call. Claude-Opus-4.6 falls between these groups (207 avg), interleaving free-form reasoning with tool calls. The token overhead is thus determined by the backbone's reasoning strategy, not the agentic architecture.

Validator Activation Analysis. We next assess *deterministic validation* by quantifying the validator's active role. We analyze Qwen3-8B, the smallest backbone where validation is most critical. Of 7,372 MultiWOZ turns, 683 (9.3%) triggered at least one validator correction, producing 1,606 total feedback messages across three categories. *Action compliance* dominates with 498 turns (6.8%): the most frequent violation is attempting to submit slot values without first calling τ_{IC} (771 messages), followed by calls to undefined tools (222) and redundant duplicate invocations of the same tool within a turn (77). *Schema conformance* accounts for 177 turns (2.4%): invalid enumeration values such as

Model	LLM Calls / Turn			Output Tokens / Turn		
	Avg	P50	P99	Avg	P50	P99
Qwen3-8B	1.86	2.00	4.00	165.25	161.00	424.00
Qwen3-32B	1.83	2.00	4.00	150.40	151.00	365.58
gpt-oss-20B	1.85	2.00	3.00	448.09	386.00	1611.29
Gemma3-12B	2.19	2.00	6.00	161.69	147.00	557.90
Claude-Opus-4.6	1.89	2.00	3.00	207.45	199.00	629.58

Table 3: Efficiency of **ReacTOD** on MultiWOZ 2.1: LLM calls and output tokens per turn under full ReAct loop.

"Cambridge" for an area slot constrained to {centre, east, north, south, west} (274 messages), hallucinated slot names like `attraction-postcode` or `train-departure-time` instead of the valid `train-leaveat` (58), and unrecognized intent names (47). *Coreference consistency* triggers least frequently at 44 turns (0.6%), flagging 157 generic entity references where the model outputs "restaurant" or "hotel" instead of the actual entity name, steering it to invoke τ_H for history retrieval. Of the 683 impacted turns, 636 self-corrected after structured feedback, with only 47 exhausting the $K_{max} = 6$ ceiling—a **93.1%** overall recovery rate (action 91.6%, schema 91.5%, coreference 95.5%). To isolate the validator's contribution from the loop's iteration opportunity, we additionally evaluate a variant with the ReAct loop active but the validator disabled: Qwen3-8B drops from 47.34% to 43.00% JGA (-4.34 pp), confirming that structured error feedback—not merely additional inference attempts—drives the self-correction gains.

6 Limitations

While **ReacTOD** demonstrates strong zero-shot performance, several limitations warrant acknowledgment.

LLM Dependency and Cost. ReacTOD introduces additional LLM calls per turn compared to single-pass approaches. While the ReAct loop is bounded in practice, each turn still incurs multiple inference requests, which may be a constraint in cost-sensitive or high-throughput production deployments.

Schema Dependency. ReacTOD is zero-shot with respect to training data—no labeled dialogues, fine-tuning, or in-domain examples are required—but it does require a configured domain schema specifying intent definitions, slot names with descriptions, and type annotations. For schema-rich benchmarks like SGD, this configuration is largely derivable from existing service definitions. For MultiWOZ, where structured schema metadata is more limited, manual annotation of slot types was necessary. This schema engineering effort, while modest compared to collecting labeled training data, represents a setup cost that should be factored into deployment planning. Performance may further degrade in settings where the schema is incomplete, noisy, or absent, limiting applicability to truly open-domain dialogue.

7 Conclusion

We presented ReacTOD, a bounded neuro-symbolic architecture that decomposes NLU into discrete, validator-gated tool calls within a constrained ReAct loop. On MultiWOZ 2.1, ReacTOD establishes a new zero-shot state-of-the-art: gpt-oss-20B reaches 52.71% Overall JGA, surpassing FnCTOD with GPT-4 (38.71%) by 14 percentage points, while the 8B-parameter Qwen3-8B achieves 47.34%—exceeding FnCTOD with the 4$\times$ larger Qwen3-32B. On SGD, ReacTOD with Claude-Opus-4.6 achieves 80.68% per-service JGA under fully end-to-end evaluation (predicted domains), and Qwen3-32B reaches 64.09%, confirming cross-benchmark generalization to SGD's 26-service, 16-domain schema space without task-specific training data. The agentic loop is the critical differentiator, contributing up to 9.3 pp on MultiWOZ and 11.82 pp on SGD over single-pass inference, yet remains computationally bounded: the median turn requires just two LLM calls with 150–207 avg output tokens. The deterministic symbolic validator catches and corrects common LLM errors—malformed values, invalid slot names, hallucinated entities—before they reach the dialogue state, achieving a 93.1% self-correction rate on intercepted errors and producing a fully inspectable execution trace at every turn. By confining generative flexibility to narrowly scoped tasks and delegating control flow to symbolic logic, ReacTOD demonstrates that reliable zero-shot agentic NLU does not require frontier-scale models—structured reasoning capacity and deterministic safeguards matter more than parameter count.

Future Work

Component-Isolation Ablations. While we have isolated the validator's contribution via a controlled experiment (loop active, validator disabled; §5.3), controlled comparisons of lazy vs. full schema injection across model sizes—particularly on SGD where the 26-service schema amplifies context load—and always-on history inclusion vs. on-demand retrieval via τ_H remain as future work.

Extended Dialogue Management. A natural extension of **ReacTOD** is to enrich the agent's tool repertoire beyond Intent Classification and Slot Resolution to support broader dialogue management capabilities. In particular, we plan to introduce tools for handling conversational control flows that arise frequently in real-world deployments: for instance, a *wait* or *clarification-pending* tool to gracefully manage turns where the system must defer resolution until additional user information is collected, and a *repeat* or *confirmation* tool to allow the agent to re-surface prior questions or confirm ambiguous slot values with the user. These additions would move **ReacTOD** closer to a fully agentic dialogue manager, capable of handling the full spectrum of conversational acts beyond state tracking.

References

Maia Aguirre, Ariane Méndez, Arantza del Pozo, María Inés Torres, and Manuel Torralbo. 2024. Fine-tuning medium-scale llms for joint intent classification and slot filling: A data-efficient and cost-effective solution for smes. In *Vicomtech Foundation*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *Preprint*, arXiv:1902.10909.

Michelle Elizabeth, Morgan Veyret, Miguel Couceiro, Ondrej Dusek, and Lina M. Rojas Barahona. 2025. Exploring ReAct prompting for task-oriented dialogue: Insights and shortcomings. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 143–153, Bilbao, Spain. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the twelfth language resources and evaluation conference*, pages 422–428.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2022. A simple language model for task-oriented dialogue. *Preprint*, arXiv:2005.00796.

Liang Huang, Senjie Liang, Feiyang Ye, and Nan Gao. 2022. A fast attention network for joint intent detection and slot filling on edge devices. *Preprint*, arXiv:2205.07646.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Jihyun Lee and Gary Geunbae Lee. 2024. Inference is all you need: Self example retriever for cross-domain dialogue state tracking with chatgpt. *Preprint*, arXiv:2409.06243.

Zekun Li, Zhiyu Zoey Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Luna Dong, Adithya Sagar, Xifeng Yan, and Paul A. Crook. 2024. Large language models as zero-shot dialogue state tracker through function calling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

H. Lu, L. Zhong, H. Jiang, W. Chen, C. Yuan, and X. Wang. 2024. Prompt-based end-to-end cross-domain dialogue state tracking. *Electronics*, 13:3587.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

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.

Abdulfattah Safa and Gözde Gül Şahin. 2025. A zero-shot open-vocabulary pipeline for dialogue understanding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Borui Xu, Yao Chen, Zeyi Wen, Weiguo Liu, and Bingsheng He. 2025a. Evaluating small language models for news summarization: Implications and factors influencing performance. 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 4909–4922.

Huan Xu, Zequn Li, Wen Tang, and Jian Jun Zhang. 2025b. From schema to state: Zero-shot scheme-only dialogue state tracking via diverse synthetic dialogue and step-by-step distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1640–1652.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd workshop on natural language processing for conversational AI*, pages 109–117.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *Preprint*, arXiv:2201.08904.