Know Your Mistakes: Towards Preventing Overreliance on Task-Oriented Conversational AI Through Accountability Modeling

Suvodip Dey, Yi-Jyun Sun, Gokhan Tur, Dilek Hakkani-Tür
University of Illinois Urbana-Champaign
suvodip15@gmail.com, jennysun0830@gmail.com, {gokhan,dilek}@illinois.edu

Abstract

Recent LLMs have enabled significant advancements for conversational agents. However, they are also well known to hallucinate, producing responses that seem plausible but are factually incorrect. On the other hand, users tend to over-rely on LLM-based AI agents, accepting AI's suggestion even when it is wrong. Adding positive friction, such as explanations or getting user confirmations, has been proposed as a mitigation in AI-supported decision-making systems. In this paper, we propose an accountability model for LLM-based task-oriented dialogue agents to address user overreliance via friction turns in cases of model uncertainty and errors associated with dialogue state tracking (DST). The accountability model is an augmented LLM with an additional accountability head that functions as a binary classifier to predict the relevant slots of the dialogue state mentioned in the conversation. We perform our experiments with multiple backbone LLMs on two established benchmarks (MultiWOZ and Snips). Our empirical findings demonstrate that the proposed approach not only enables reliable estimation of AI agent errors but also guides the decoder in generating more accurate actions. We observe around 3% absolute improvement in joint goal accuracy (JGA) of DST output by incorporating accountability heads into modern LLMs. Self-correcting the detected errors further increases the JGA from 67.13 to 70.51, achieving state-of-the-art DST performance. Finally, we show that error correction through user confirmations (friction turn) achieves a similar performance gain, highlighting its potential to reduce user overreliance. [1]

1 Introduction

Conversational agents have reached remarkable advancements with the advent of large language models (LLMs). However, they are prone to hallucinations, generating information that is incorrect or not grounded in reality (Ji et al., 2023; Huang et al., 2023). At the same time, users often tend to over-rely on LLM-based AI agents, accepting AI suggestions even when they are erroneous (Passi and Vorvoreanu, 2022; Klingbeil et al., 2024). In task-oriented conversations, such overreliance causes incorrect or incomplete task execution, undermining the system's reliability. To address this issue, we employ accountability modeling, where accountability refers to the model's ability to explain or justify its actions (Doshi-Velez et al., 2019; Novelli et al., 2022). Accountability modeling of task-

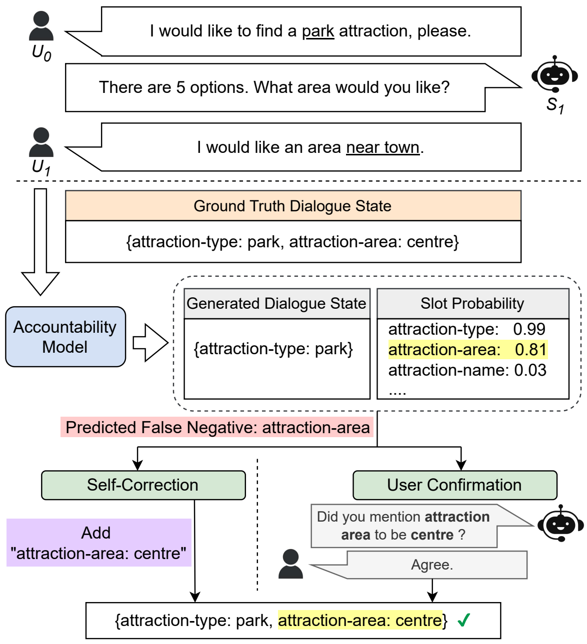

Figure 1: Overview of Accountability Modeling. The model simultaneously generates the dialogue state and estimates the probability of each slot (being included in the dialogue state). Based on these probabilities, it identifies potential errors, which can either be self-corrected using a dedicated algorithm or confirmed by the user through clarification questions.

[1]Code available at github.com/uiuc-conversational-ai-lab/Accountable-DST

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 28830–28843
July 27 - August 1, 2025

oriented dialogue systems enables the identification and resolution of errors or unintended consequences, thereby alleviating user overreliance.

Task-oriented dialogue systems (TODS) are designed to assist users in completing a task or goal through conversations. Dialogue state tracking (DST) is a crucial component of TODS that accounts for understanding the user intentions and keeping track of the dialogue state. The dialogue state contains the intents communicated by the user and is generally represented as a set of slot-value pairs. The DST task is to predict the dialogue state after each user turn, as shown in Fig. 1. It can also be seen as a function or API call in end-to-end TODS (Li et al., 2024b; Xu et al., 2024). There are three types of error associated with DST output:

- **False Positives**: Predicted slots that were not mentioned in the dialogue so far.
- **False Negatives**: Slots that were mentioned in the dialogue but are missing from the predicted dialogue state.
- **Value Errors**: The slot is relevant, but its value is wrong with respect to the dialogue context.

Task-oriented dialogues are highly sensitive to these errors, as even a single mistake can significantly alter the conversation's trajectory. LLM-based DST models generally predict the correct slot value for the relevant slots, making value errors a minor concern. However, false positive/negative errors occur more frequently and are critical to task success. For instance, in Fig. 1, the model initially predicts `{attraction-type: park}` as the dialogue state, causing `attraction-area` to be a false negative slot for this prediction. As a result, the system may recommend parks that are not near or centre of the town, potentially leading the user to book an unsuitable option due to overreliance on the system. Such issues can degrade the user experience in real-world task-oriented conversations.

In this work, we propose an accountability model for task-oriented conversations to mitigate user overreliance. Our approach aims to enhance the prediction of DST by detecting and correcting false positive and false negative errors, thereby ensuring greater accountability. To achieve this, we integrate an accountability head into the backbone LLMs, which is a binary classifier to predict the slots in the dialogue state. This augmented LLM not only generates the dialogue state but also estimates the probability of each slot being included in the dialogue state. The slot probabilities help to identify the possible false positive/negative slots. These errors are then self-corrected using a dedicated algorithm (Algorithm 1) that removes false positive slots and adds false negative slots (with the appropriate values), thereby improving the accuracy of DST prediction. For instance in Fig. 1, the ground-truth dialogue state is `{attraction-type: park, attraction-area: centre}`. However, the model initially predicted `{attraction-type: park}`. This results in a false negative, as the prediction does not contain `attraction-area: centre`. The accountability head assigns a high probability (0.81) to the slot `attraction-area`, identifying it as a potential false negative. In Fig. 1, the model self-corrects this error by generating the value for attraction-area (i.e. centre) using Algorithm 1 and adds it to the initial prediction, thereby successfully correcting the dialogue state.

Rather than self-correcting, the conversational agent can also confirm the detected errors with the user through a conversation turn. For example in Fig. 1, the model asks a clarification question to the user to confirm the error, which helps to rectify the mistake. In recent literature, such mindful interactions or friction turns prompting analytical thinking have been explored to address overreliance on AI (Mejtoft et al., 2019; Naiseh et al., 2021; İnan et al., 2025). Therefore, slot probabilities from the accountability head can help introduce friction turns, such as confirming model uncertainty and errors, to mitigate user overreliance. The contributions of this work are as follows:

- We propose a generative LLM-based accountability model for DST capable of detecting false positives and false negatives.
- Accountability modeling improves the DST performance of backbone LLMs on two widely used corpora (MultiWOZ and Snips). For MultiWOZ, the injection of the accountability head in Llama, Mistral, and Gemma shows an absolute improvement of approximately 3% in joint goal accuracy.
- Self-correcting errors identified by the accountability head further enhance the dialogue state prediction, achieving 70.51% joint goal accuracy and surpassing the performance of state-of-the-art DST models.

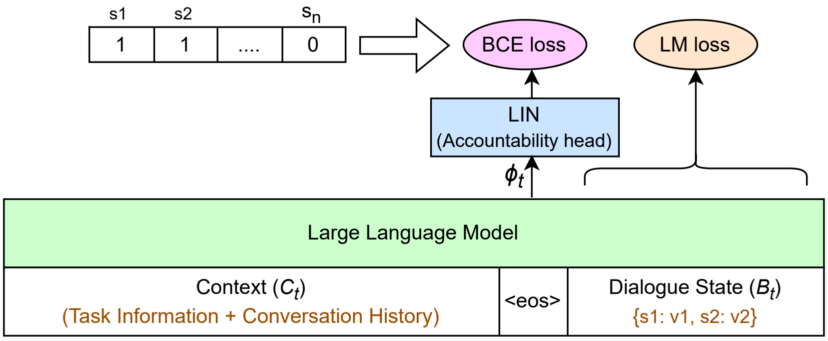

Figure 2: Model architecture of the LLM-based generative accountability modeling for DST.

- Accountability modeling enables the correction of dialogue states via user confirmations which achieves a performance gain comparable to the self-correction strategy, highlighting its potential to reduce user overreliance.

2 Related Work

DST was initially solved independently from the other TODS modules. In this approach, DST is formulated as a slot-filling task, where the task is to find the relevant slots and then fill out the slot values. Finding the relevant slots is posed as a classification task, while various strategies have been explored to find the slot values like pick-list-based (Mrkšić et al., 2017; Nouri and Hosseini-Asl, 2018; Zhong et al., 2018), generative (Wu et al., 2019; Kim et al., 2020), and reading comprehension (Gao et al., 2019; Heck et al., 2020; Dey and Desarkar, 2021). Although using models with fewer parameters compared to modern LLMs, these methods exhibit decent DST performance. With additional training with synthetic data and sophisticated training, these models have been shown to achieve state-of-the-art DST performance (Ye et al., 2022a,c). However, they are difficult to extend to an end-to-end TODS framework.

With the advancements in Transformer (Vaswani et al., 2017) based LLMs, the paradigm shifted towards fully generative and end-to-end modeling of TODS (Hosseini-Asl et al., 2020; Lin et al., 2020; Mehri et al., 2020). These methods show competitive DST performance compared to the earlier methods. Recently, prompt-based LLM methods have been studied extensively for zero-shot and few-shot DST (Hu et al., 2022; Su et al., 2022; Hudeček and Dusek, 2023; Xu et al., 2024). Despite their impressive performance, confidence estimation of dialogue states in generative LLM-based methods is challenging. LLMs are generally overconfident with the predictions, resulting in high confidence even for spurious predictions (Huang et al., 2024; Xiong et al., 2024; Li et al., 2024a). In recent work, Sun et al. (2024) addresses the problem of reliable confidence estimation for DST. However, estimating the confidence for the false negatives is not possible as they are not part of the generated dialogue state (Feng et al., 2023a; Xu et al., 2024). Note that slot-filling-based methods are not susceptible to this issue as they have dedicated classifiers to predict the slots. This work aims to combine the advantages of both approaches for accountability modeling of DST.

3 Methodology

This section presents a brief background of LLM-based generative DST, followed by our proposed accountability model and its application in correcting predicted dialogue states.

3.1 Generative DST

In this method, DST is solved like standard natural language generation tasks using LLMs. Let $D_t = \{(S_0, U_0), ...(S_n, U_n)\}$ be a task-oriented dialogue where S_t and U_t represent the system and user utterances at turn t, respectively. Let C_t represent the dialogue context, which includes D_t as well as optional task-related information and slot descriptions. Let $B_t = \{s_1 : v_1, s_2 : v_2, ...\}$ be the dialogue state at turn t where (s_i, v_i) represents the i^{th} slot-value pair. For multi-domain datasets (like MultiWOZ and Snips), s_i is expressed as domain-slot pair. The LLM is trained using the standard language model (LM) loss, defined as follows,

$$\mathcal{L}_{\text{LM}} = -\frac{1}{T}\sum_{n=1}^{T} \log p(B_{t_n} | B_{t_{<n}}, C_t; \theta) \quad (1)$$

where θ denote the parameters of the LLM and B_{t_n} is the n^{th} token of the tokenized B_t with T tokens.

3.2 Accountability Modeling for DST (AMD)

In this work, we propose to add an accountability head to make the LLM-based DST generation accountable. The accountability head functions as a classifier to determine whether a slot has been specified in the given context. Let $\phi_t \in \mathbb{R}^d$ be the encoding of the last token (separator or end-of-sentence) of the context C_t, as shown in Fig. 2. Then, the binary-cross entropy (BCE) of the accountability head for a turn t is computed as follows.

$$p = \sigma(\text{LIN}(\phi_t)) \in \mathbb{R}^{|S|} \tag{2}$$

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|S|}\sum_{s\in S}(y_s \cdot \log p_s + (1-y_s)\cdot(1-\log p_s)) \tag{3}$$

where S is the set of slots, σ is the element-wise sigmoid operation, LIN represents a linear layer, p_s denotes the probability of slot s, and $y_s \in \{0, 1\}$ indicates the label for slot classification denoting the presence/absence of slot s in the ground-truth dialogue state (B_t). The final training objective of the accountability model is defined as follows.

$$\mathcal{L}_{\text{Account}} = \mathcal{L}_{\text{LM}} + \lambda * \mathcal{L}_{\text{BCE}} \tag{4}$$

where $\lambda \in [0, 1]$ is a hyperparameter to control the weight of the BCE loss. Note that we do not generate the dialogue policy or response, as our primary objective is to study the utility of accountability modeling in preventing user overreliance. We plan to explore the complete end-to-end modeling with an accountability head as an extension of this work.

The accountability head helps estimate the slot probabilities for all the slots, which can be used to detect the false positive and negative slots in the predicted dialogue state. Furthermore, the inclusion of the accountability head acts as an auxiliary loss that helps in the learning of dialogue state generation. Any information about the correct slots can make the dialogue state generation easier. Note that ϕ_t is learned to optimize both $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{LM}}$. Therefore, ϕ_t encodes the knowledge about the relevant slots, which can improve the accuracy of the generation of the dialogue state. A similar strategy has been shown to be effective in open-domain dialogue generation, where predicting response keywords beforehand can guide and improve response generation (Dey and Desarkar, 2024).

Algorithm 1: Dialogue State Correction

Input:
B = predicted dialogue state,
S = set of slots,
p = slot probabilities,
τ_{fp} = false positive threshold,
τ_{fn} = false negative threshold,
D = dialogue history.
Output: Corrected dialogue state (B')

1 $B' = \{\}$
`/* Step 1: Filtering false positives */`
2 **for** slot, value $\in B$ **do**
3   **if** $p_{\text{slot}} \geq \tau_{\text{fp}}$ **then**
4   B'[slot] $\leftarrow$ value
`/* Step 2: Add false negatives */`
5 $S' \leftarrow$ Set of slots in B
6 **for** slot $\in S \setminus S'$ **do**
7   **if** $p_{\text{slot}} \geq \tau_{\text{fn}}$ **then**
8   B'[slot] $\leftarrow$ generateSlotValue(D, B, slot)
9 **return** B'

3.3 Dialogue State Correction using Accountability Model

The slot probabilities output by the classifier can be used to self-correct (SC) the generated dialogue state, either by removing slot-value pairs from the dialogue state or forcing the LLM to continue state generation by providing it with the missing slot names. Algo 1 shows the dialogue state correction algorithm. Let τ_{fp} and τ_{fn} be the false positive and false negative thresholds. Let B be the generated dialogue state and $p \in \mathbb{R}^{|S|}$ be the slot probabilities of the classifier.

The first step of Algo 1 attempts to filter the possible false positives, while the second step helps to include the possible false negatives. Let generateSlotValue() be the function to generate the slot value for false negative slots (Line 8, Algo 1). The function generates the slot value for a given slot by appending the slot name to the generated dialogue state (B) and runs the model decoder to complete the generation. We select the optimal τ_{fp} and τ_{fn} that maximizes the joint goal accuracy of the validation set using grid search.

Instead of self-correcting, we can correct the detected errors through user confirmation. This experimentation is discussed in Section 6.

Dataset	#Slots	Mode	#Dialogues	#Turns
MultiWOZ	30	Train	8420	56668
		Validation	1000	7374
		Test	999	7368
Snips	53	Train	13084	13084
		Validation	700	700
		Test	700	700

Table 1: Data statistics of MultiWOZ and Snips.

4 Experiment Setup

4.1 Dataset

We experiment with two widely used datasets: i) MultiWOZ (Budzianowski et al., 2018), and ii) Snips (Coucke et al., 2018). MultiWOZ is one of the largest multi-domain conversation corpora for task-oriented dialogue, containing multi-turn conversations. We use MultiWOZ 2.4 (Ye et al., 2022b), which contains fewer annotation errors and inconsistencies than the other MultiWOZ versions. On the other hand, Snips is a similar dataset for spoken language understanding, containing only single-turn conversations. The basic statistics of both datasets are shown in Table 1.

4.2 Evaluation Metric

DST is primarily evaluated using joint goal accuracy (JGA). JGA is defined as the percentage of turns where the predicted dialogue state exactly matches the ground-truth (Henderson et al., 2014; Dey et al., 2022). Additionally, we report the Slot-F1 score, which measures the slot-level F1 performance. We also analyze false positive rate (FPR) and false negative rate (FNR) for certain results. FPR and FNR are defined as the percentage of turns containing false positive and false negative slots, respectively. Let X be the number of turns containing at least one false positive slot. Let Y be the number of turns containing at least one false negative slot. Then, FPR and FNR are defined as $\text{FPR} = \frac{100X}{T}$ and $\text{FNR} = \frac{100Y}{T}$, where T is the total number of turns in the dataset.

4.3 Model Architectures and Variants

We study the utility of the proposed accountability model by applying it to three LLMs - Llama 3.1 (8B) (Dubey and et al., 2024), Mistral (7B) (Jiang et al., 2023), and Gemma (7B) (Team and et al., 2024). We use the instruction-tuned version of the three models. The model nomenclature used in our experiments is described as follows.

- $\mathcal{M}_{\text{SFT}}$: Model trained using supervised fine-tuning (SFT) for dialogue state generation with only $\mathcal{L}_{\text{LM}}$ (Eqn. 1), which is methodologically similar to LDST (Feng et al., 2023b) with minor modifications.[2]
- $\mathcal{M}_{\text{AMD}}$: Proposed **A**ccountability **M**odel for **D**ST (AMD) with accountability head, fine-tuned with $\mathcal{L}_{\text{Account}}$ (Eqn. 4).
- $\mathcal{M}_{\text{AMD+SC}}$: $\mathcal{M}_{\text{AMD}}$ after **S**elf-**C**orrecting (SC) the dialogue state using Algo 1.

4.4 Training Details

All the models are implemented using the PyTorch and Huggingface libraries in Python 3.12. We used LoRA (Hu et al., 2021) finetuning with rank (r) 8, $\alpha = 32$, and dropout 0.1. We used AdamW optimizer with learning rate 5e-5 to fine-tune both $\mathcal{M}_{\text{SFT}}$ and $\mathcal{M}_{\text{AMD}}$. We trained all the models for 4 epochs and chose the final model with the minimum validation loss. The prompts used for model finetuning are provided in Appendix A.2. In Eqn. 4, the optimal λ is selected based on the JGA score of the validation set. The best λ was found to be 0.25 except for Mistral-Snips and Gemma-Snips, where $\lambda = 0.1$ resulted in the best validation performance. For $\mathcal{M}_{\text{AMD+SC}}$, the false positive ($\tau_{\text{fp}}$) and the false negative (τ_{fn}) threshold is selected similarly based on the validation performance. We observed that (0.1, 0.5) and (0.05, 0.9) are the best (τ_{fp}, τ_{fn}) for MultiWOZ and Snips, respectively. These optimal values of λ, τ_{fp}, and τ_{fn} are used to show the test performance in the rest of the article.

Further details on prompts, training, and inference are provided in Section A.2.

5 Results and Analysis

5.1 DST Performance

Table 2 compares the DST performance of the different model variants described in Section 4.3. Table 3 shows the comparison with the previous baselines. The observations are summarized as follows.

Impact of Accountability Head: In Table 2, we observe that adding the accountability head in $\mathcal{M}_{\text{AMD}}$ improves the performance of $\mathcal{M}_{\text{SFT}}$. Both joint goal accuracy (JGA) and Slot-F1 are improved for all the backbone models in both datasets. Overall, in MultiWOZ, we can observe an absolute improvement of around 3%. Similarly, we observe

[2] LDST (Feng et al., 2023b) is trained to generate the value for a single slot where all possible slot-values from the database is provided in the context as meta-data, limiting its scalability. In contrast, $\mathcal{M}_{\text{SFT}}$ is trained to generate the full dialogue state without using any such meta-data.

Model	Type	MultiWOZ				Snips			
		JGA ↑	Slot-F1 ↑	FPR ↓	FNR ↓	JGA ↑	Slot-F1 ↑	FPR ↓	FNR ↓
Llama	$\mathcal{M}_{\text{SFT}}$	64.34	95.23	12.17	23.72	92.43	97.76	5.14	4.71
	$\mathcal{M}_{\text{AMD}}$	67.13 ↑4.3	95.90	13.17	18.28	93.57 ↑1.2	98.00	4.71	4.43
	$\mathcal{M}_{\text{AMD+SC}}$	**70.51** ↑9.6	**96.51**	12.83	14.44	93.71 ↑1.4	98.17	4.00	4.00
Mistral	$\mathcal{M}_{\text{SFT}}$	65.86	95.68	11.90	20.41	92.57	97.57	6.28	5.28
	$\mathcal{M}_{\text{AMD}}$	68.58 ↑4.1	96.19	11.35	16.94	93.71 ↑1.2	98.06	4.85	4.43
	$\mathcal{M}_{\text{AMD+SC}}$	69.84 ↑6.0	96.37	12.74	14.19	**94.00** ↑1.5	**98.17**	4.57	4.14
Gemma	$\mathcal{M}_{\text{SFT}}$	62.12	95.05	6.35	28.84	91.43	97.37	5.71	5.29
	$\mathcal{M}_{\text{AMD}}$	65.05 ↑4.7	95.68	12.47	20.15	91.86 ↑0.5	97.86	5.43	5.29
	$\mathcal{M}_{\text{AMD+SC}}$	66.27 ↑6.7	96.03	16.08	15.08	92.00 ↑0.6	98.00	5.14	5.14

Table 2: Comparison of the DST performance on the MultiWOZ 2.4 and Snips test datasets with different LLM backbones. The relative JGA improvement of proposed $\mathcal{M}_{\text{AMD}}$ and $\mathcal{M}_{\text{AMD+SC}}$, compared to the respective $\mathcal{M}_{\text{SFT}}$ baseline, is highlighted in blue. The best results are indicated in bold font.

Type	Model	JGA
Zero-Shot	GPT-4o (Sun et al., 2024)	36.10
Few-Shot	IC-DST (Hu et al., 2022)	62.43
	OrchestraLLM (Lee et al., 2024b)	52.68
	CorrectionLM (Lee et al., 2024a)	57.35
SFT	TRADE (Wu et al., 2019)	55.05
	SUMBT (Lee et al., 2019)	61.86
	SimpleTOD (Hosseini-Asl et al., 2020)	57.18
	TripPy (Heck et al., 2020)	64.75
	SOM-DST (Kim et al., 2020)	66.78
	Seq2Seq (Zhao et al., 2021)	67.10
	TripPy-R (Heck et al., 2022)	69.87
	LDST (Feng et al., 2023b) (10% data)	62.45
	$\mathcal{M}_{\text{AMD}}$ (Ours)	67.13
	$\mathcal{M}_{\text{AMD+SC}}$ (Ours)	**70.51**

Table 3: JGA comparison between various DST models on the MultiWOZ 2.4 test data.

Type	τ_{fp}	τ_{fn}	Cost	JGA ↑	FPR ↓	FNR ↓
$\mathcal{M}_{\text{AMD}}$	0	1	0	67.13	13.17	18.28
Filter False Positives	0.1	1	0	68.15	11.16	18.92
	0.2	1	0	68.01	9.54	20.63
	0.3	1	0	67.24	8.24	22.86
Add False Negatives	0	0.9	1.5	67.55	13.59	17.19
	0	0.7	4.7	68.59	14.68	14.60
	0	0.5	7.5	69.31	16.39	12.11
	0	0.4	8.9	68.97	18.28	10.74

Table 4: Impact of false positive and negative threshold on dialogue state correction in MultiWOZ 2.4 test set with Llama. "Cost" denotes the %turns where the model generated values for false negative correction.

approximately 1% improvement for Snips. The margin of improvement in Snips is lower as it is relatively easier than MultiWOZ. The improvement in JGA with the accountability head can be attributed to the significant reduction in FNR for MultiWOZ and the reduction in both FPR and FNR for Snips.

Impact of Self-Correction: $\mathcal{M}_{\text{AMD+SC}}$ self-corrects the predicted dialogue states of $\mathcal{M}_{\text{AMD}}$ using Algo 1. We use the optimal τ_{fp} and τ_{fn} thresholds that maximize the JGA of the validation set (as described in Section 4.4). We can observe that self-correcting the dialogue state improves DST performance significantly. Considering the best results in Table 2, $\mathcal{M}_{\text{AMD+SC}}$ improves the performance of the baseline $\mathcal{M}_{\text{SFT}}$ model by 9.6% and 1.5% for MultiWOZ and Snips, respectively. We also observe that correcting false negatives significantly reduces the FNR in both MultiWOZ and Snips. However, false negative correction can sometimes introduce spurious slots (discussed in Section 5.2), leading to a higher FPR in $\mathcal{M}_{\text{AMD+SC}}$ compared to $\mathcal{M}_{\text{AMD}}$ in certain MultiWOZ instances where false negatives are more prevalent than false positives. This happens because we optimized τ_{fp} and τ_{fn} to maximize JGA rather than FPR or FNR. In Snips, self-correction reduces both FPR and FNR because of the more balanced occurrence of false positives and false negatives.

Comparison with previous baselines: Table 3 compares our proposed $\mathcal{M}_{\text{AMD}}$ and $\mathcal{M}_{\text{AMD+SC}}$ with various zero-shot, few-shot, and supervised fine-tuning (SFT) baselines on the MultiWOZ 2.4 dataset. The results demonstrate that $\mathcal{M}_{\text{AMD+SC}}$ outperforms both zero-shot and few-shot baselines and achieves state-of-the-art performance compared to SFT baselines. For fair comparison, we do not consider models such as STAR (Ye et al., 2021) and ASSIST (Ye et al., 2022a) that use additional synthetic data for training and generally achieve higher JGA (around 80%).

5.2 Impact of Varying False Positive and False Negative Threshold

In this section, we study the impact of varying the false positive threshold (τ_{fp}) and false negative threshold (τ_{fn}) while correcting the dialogue states using Algo 1. Table 4 shows the change in DST performance by varying τ_{fp} and τ_{fn}. In our setup, $\tau_{\text{fp}} = 0$ and $\tau_{\text{fn}} = 1$ indicate no correction for false positives and false negatives, respectively.

Firstly, we study the impact of filtering only pos-

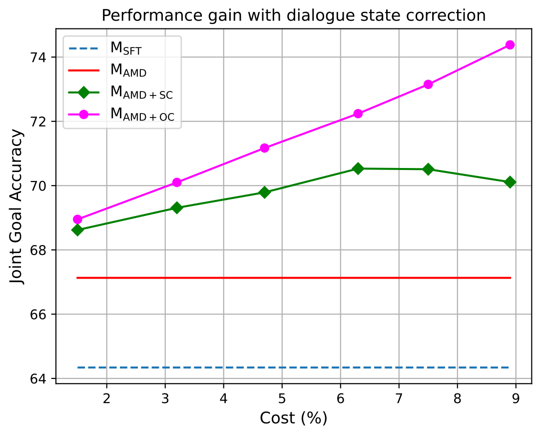

Figure 3: Performance gain (JGA) with varying τ_{fp} and τ_{fn} on MultiWOZ 2.4 test set with Llama. The x-axis shows the %turns with false negative correction. On average, 1.1 slots have been involved in the false negative corrections consistently over the range of this plot. Hence, we did not include it in the depiction.

sible false positive slots whose probabilities are less than τ_{fp} (Line 3-4, Algo 1). We can observe that increasing τ_{fp} reduces FPR. However, this process also filters correct slots with $p_{\text{slot}} < \tau_{\text{fp}}$, thereby increasing the FNR. As a result, we can observe that the JGA starts degrading when $\tau_{\text{fp}} > 0.1$ due to increasing FNR. Since we are discarding slots based on a threshold, no additional cost is involved in generating slot values.

Next, we analyze the impact of correcting only possible false negative slots with different τ_{fn} (Line 5-8, Algo 1). We can observe that decreasing τ_{fn} reduces FNR. However, this process also tries to include false positive slots with $p_{\text{slot}} \geq \tau_{\text{fn}}$, thereby increasing the FPR. Consequently, JGA drops when $\tau_{\text{fn}} < 0.5$. Note that correcting false negatives involves generating the slot values. In Table 4, we denote this generation cost by the percentage of turns that require slot-value generation to rectify false negatives. This generation cost is inversely proportional to τ_{fn}. On average, slot values are generated for 1.1 slots per updated turn.

Fig. 3 shows the summary of the performance gain achieved with dialogue state correction. We observe that $\mathcal{M}_{\text{AMD+SC}}$ significantly enhances the performance of $\mathcal{M}_{\text{SFT}}$ (9.6% relative) through self-corrections. The best performance in MultiWOZ is achieved with $\tau_{\text{fp}} = 0.1$ and $\tau_{\text{fn}} = 0.5$. However, beyond a certain point, the JGA begins to decline as false negative corrections induce false positives, as previously discussed. Figure 3 also depicts the maximum JGA achievable through dialogue state

λ	MultiWOZ		Snips	
	JGA	**Slot-F1**	**JGA**	**Slot-F1**
0	64.34	95.23	92.43	97.76
0.1	65.93	95.64	92.71	97.82
0.25	67.13	95.90	93.57	98.01
0.5	66.27	95.80	93.14	97.87
0.75	65.40	95.61	93.00	97.84
1.0	65.25	95.57	92.86	97.72

Table 5: Impact of varying λ in $\mathcal{M}_{\text{AMD}}$ with Llama backbone on test data.

correction using $\mathcal{M}_{\text{AMD+OC}}$ where OC stands for **O**racle-**C**orrection. Instead of self-correcting, $\mathcal{M}_{\text{AMD+OC}}$ corrects the output of $\mathcal{M}_{\text{AMD}}$ using the oracle or ground-truth slot values in Step 2 of Algo 1. $\mathcal{M}_{\text{AMD+OC}}$ does not introduce false positives while addressing false negatives, as only the slots that correspond to the ground truth are updated. The plot suggests that there is still room to achieve higher JGA through self-correction.

5.3 Impact λ on DST performance

In this ablation study, we vary the value of λ in Eqn. 4. In Table 5, for both datasets, we observe that the JGA and Slot-F1 increase with λ initially and then start dropping. Note that the primary task here is to generate the dialogue state for which the JGA or Slot-F1 is estimated. Now, increasing the λ increases the importance of the accountability head, which starts affecting the generation head after a certain point. Therefore, although any $\lambda \in [0, 1]$ can improve the base model, a thorough hyperparameter optimization of λ is required to achieve the best possible DST performance. The validation performance used to select the optimal value of λ is discussed in Appendix A.1.

5.4 Qualitative Analysis

In this section, we provide a qualitative analysis of the $\mathcal{M}_{\text{AMD+SC}}$ model. Table 6 shows illustrative examples of the model prediction from the MultiWOZ datasets. In the first example, the model detected a false positive slot (*restaurant-pricerange*) and filtered it to rectify the prediction. The second example contains a false negative slot (*attraction-type*), which is corrected successfully. In the third example, the model detects both false positive and false negative slots and successfully rectifies them. In the fourth example, the original prediction of the model is correct. However, self-correction adds an extra slot (*train-departure*), which makes the prediction wrong. The fifth one shows an instance where the algorithm partially corrects an

ID	Dialogue History	Generated dialogue state	Predicted Errors	Post-Correct Status
Example #1 PMUL3514 (MultiWOZ)	U_0: I am looking for information in Cambridge. S_1: I need more specifics to help you. what type of information do you need? U_1: I would like a moderately priced place to stay. but only if it is a 0 star. I love a little adventure!	{restaurant-pricerange: expensive, hotel-pricerange: moderate, hotel-stars: 0} ✗	False Positive:- restaurant-pricerange: expensive (0.02)	✓
Example #2 PMUL3919 (MultiWOZ)	U_0: Can you find a theater to go to in town? S_1: Sure, do you have an area of town you would like to visit? U_1: No, I am not concerned about that.	{attraction-type: theatre} ✗	False Negative:- attraction-area: dontcare (0.6)	✓
Example #3 MUL1289 (MultiWOZ)	U_0: I am trying to plan a trip there but need a cheap place to stay. S_1: Alexander bed and breakfast is located in the centre. They are located at 56 saint barnabas road. U_1: Hmm, i am looking for a place in the west. It does not need to include internet. S_2: finches bed and breakfast is cheap. U_2: okay do they have free wifi?	{hotel-area: west, hotel-pricerange: cheap, hotel-internet: yes, hotel-type: hotel} ✗	False Positive:- hotel-type: hotel (0.06) False Negative:- hotel-name: finches bed and breakfast (0.72)	✓
Example #4 PMUL1091 (MultiWOZ)	U_0: Can you locate for me a train that leaves on tuesday after 3:15 pm? thanks. S_1: There is a train that leaves cambridge at 15:00 and arrives at london kings cross at 15:51 on tuesday. U_1: That's too early. I need to leave after 15:15.	{train-leaveat: 15:15, train-day: tuesday} ✓	False Negative:- train-departure: cambridge (0.75)	✗
Example #5 MUL2053 (MultiWOZ)	U_0: Hi there. Can you help me find a 2 star rated hotel or guest house? S_1: Ashley hotel is a 2 star hotel in 74 chesterton road. U_1: Does that include wifi?	{hotel-stars: 2, hotel-internet: yes} ✗	False Negative:- hotel-name: ashley hotel (0.81), hotel-type: hotel (0.52)	✗

Table 6: Illustrative example of dialogue state corrections using the proposed $\mathcal{M}_{\text{AMD+SC}}$ model.

error. Here, the model included *hotel-name* to its prediction but added a false positive (*hotel-type*) in the process because τ_{fn} was set to 0.5. This example demonstrates how addressing false negatives can sometimes introduce false positives.

6 Preventing User Overeliance with Accountability Model

So far, we have focused on analyzing the capability of the proposed accountability model in detecting errors and self-correcting them to improve DST performance. In this section, we discuss its application to prevent user overreliance on task-oriented conversational AI. The approach is based on introducing positive friction (İnan et al., 2025) like user confirmations that can eventually lead to successful task completion. Let E_t be the set of erroneous slot-value pairs predicted by $\mathcal{M}_{\text{AMD}}$. Given E_t, a task-oriented conversational agent can introduce friction turns to get clarification on the unconfident slots. Introducing such friction turns have been shown to be helpful in preventing user overreliance on AI (Naiseh et al., 2021; İnan et al., 2025).

To evaluate this method, we conduct an experiment using a user simulator. We employ "GPT-4o mini" (OpenAI and et al., 2024) as our user simulator, assuming cooperative user behavior. Given the dialogue history and a slot-value pair (detected as false negative or false positive), the user simulator is asked to confirm if it mentioned the slot-value pair by responding "Agree" (confirms interest) or "Disagree" (rejects). We correct the dialogue state by removing the false positive slots and adding the false negative slots confirmed by the user. A detailed description of the experimental setup is provided in Appendix A.2.

Table 7 shows the results of this experiment where $\mathcal{M}_{\text{AMD+UC}}$ denotes the **U**ser **C**onfirmation (UC) provided by the simulator. We can observe that $\mathcal{M}_{\text{AMD+UC}}$ achieves comparable performance to $\mathcal{M}_{\text{AMD+SC}}$. Although the user simulator may not fully reflect actual user behavior, assuming a

Model	Type	JGA	Slot-F1
Llama	$\mathcal{M}_{\text{AMD}}$	67.13	95.90
	$\mathcal{M}_{\text{AMD+SC}}$	70.49	96.51
	$\mathcal{M}_{\text{AMD+UC}}$	**70.78**	**96.55**
Mistral	$\mathcal{M}_{\text{AMD}}$	68.58	96.19
	$\mathcal{M}_{\text{AMD+SC}}$	69.84	96.37
	$\mathcal{M}_{\text{AMD+UC}}$	70.21	96.41
Gemma	$\mathcal{M}_{\text{AMD}}$	65.05	95.68
	$\mathcal{M}_{\text{AMD+SC}}$	66.27	96.03
	$\mathcal{M}_{\text{AMD+UC}}$	66.32	96.05

Table 7: Impact of correcting dialogue state using user simulator on MultiWOZ test data. SC and UC denote self-correction and user confirmation, respectively.

cooperative user allows us to interpret the results in Table 7 as a lower bound on achievable DST performance. A real cooperative user would likely be even more effective in identifying and confirming errors, potentially leading to further performance improvements. Additional improvements may also be possible through the use of more advanced simulators and refined prompt engineering. Nonetheless, this experiment demonstrates the practical value of our accountability modeling framework in reducing user overreliance in real-world task-oriented dialogues.

7 Conclusion

In conclusion, we present an LLM-based generative accountability modeling for task-oriented dialogue systems. The core idea of our approach involves incorporating an accountability head into backbone LLMs, which functions as a binary classifier to predict the slots in the dialogue state. Doing so not only enables the detection of both false positives and false negatives but also guides the generation of accurate dialogue states. We empirically show that accountability modeling improves the DST performance of backbone LLMs (Llama, Mistral, and Gemma) on two widely used task-oriented corpora (MultiWOZ and Snips). Identifying the errors also enables self-correction of the dialogue state, which helps to achieve state-of-the-art performance. Finally, we demonstrate the utility of the proposed accountability modeling to correct the DST errors in an interactive setup via user confirmations, thereby preventing user overreliance. In the future, we want to extend accountability modeling for end-to-end task-oriented conversations.

8 Limitations

We recognize the following limitations regarding our work.

- The accountability model assumes the data to be annotated along with a fixed number of domains and slots. This is why the proposed model cannot be directly used for new or unseen domains/slots. However, in task-oriented conversations, domains and slots are typically well-defined, making the model effectively trainable. Furthermore, the accountability model outperforms few-shot and zero-shot methods, as shown in Table 3. While general LLMs (such as GPT-4o) can operate with limited data, their performance remains suboptimal. In task-oriented conversations, accurately understanding intent is essential for effective dialogue management. Moreover, disregarding the training overhead, a small LLM with accountability modeling is more cost-effective than a large general-purpose LLM for dialogue state tracking in real applications.

- In this work, the notion of accountability is added by estimating the errors in the DST task. Hence, the proposed accountability modeling approach is specific to sequence tagging tasks like DST, information extraction, entity extraction, etc.

9 Ethics Statement

This work proposes accountability modeling for task-oriented conversational agents. We use the publicly available MultiWOZ and Snips datasets in full compliance with their terms of use. Our experiments do not use any private, confidential, or real personal data. Our model's use of personal information is limited to the task-oriented conversation of MultiWOZ and Snips. Since we use LLMs to generate dialogue states, there is minimal risk of generating harmful, biased, or discriminatory statements. We acknowledge such potential ethical concerns associated with this work.

10 Acknowledgments

This work was supported in part by Other Transaction award HR0011249XXX from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program and has benefited from the Microsoft Accelerate Foundation Models Research (AFMR) grant program, through which leading foundation models hosted by Mi-

crosoft Azure and access to Azure credits were provided to conduct the research.

References

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, Brussels, Belgium. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *ArXiv*, abs/1805.10190.

Suvodip Dey and Maunendra Sankar Desarkar. 2021. Hi-DST: A hierarchical approach for scalable and extensible dialogue state tracking. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 218–227, Singapore and Online. Association for Computational Linguistics.

Suvodip Dey and Maunendra Sankar Desarkar. 2024. BoK: Introducing bag-of-keywords loss for interpretable dialogue response generation. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 566–578, Kyoto, Japan. Association for Computational Linguistics.

Suvodip Dey, Ramamohan Kummara, and Maunendra Desarkar. 2022. Towards fair evaluation of dialogue state tracking by flexible incorporation of turn-level performances. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 318–324, Dublin, Ireland. Association for Computational Linguistics.

Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Kate Scott, Stuart Schieber, James Waldo, David Weinberger, Adrian Weller, and Alexandra Wood. 2019. Accountability of ai under the law: The role of explanation. *Preprint*, arXiv:1711.01134.

Abhimanyu Dubey and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023a. Towards LLM-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755, Singapore. Association for Computational Linguistics.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023b. Towards LLM-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755, Singapore. Association for Computational Linguistics.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Michael Heck, Nurul Lubis, Carel van Niekerk, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, and Milica Gašić. 2022. Robust dialogue state tracking with weak supervision and sparse data. *Transactions of the Association for Computational Linguistics*, 10:1175–1192.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

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).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Artur Klingbeil, Cassandra Grützner, and Philipp Schreck. 2024. Trust and reliance on ai — an experimental study on the extent and costs of overreliance on ai. *Computers in Human Behavior*, 160:108352.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2024a. Correctionlm: Self-corrections with slm for dialogue state tracking. *Preprint*, arXiv:2410.18209.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2024b. OrchestraLLM: Efficient orchestration of language models for dialogue state tracking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1434–1445, Mexico City, Mexico. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024a. Think twice before trusting: Self-detection for large language models through comprehensive answer reflection. *Preprint*, arXiv:2403.09972.

Zekun Li, Zhiyu Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Dong, Adithya Sagar, Xifeng Yan, and Paul Crook. 2024b. 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 (Volume 1: Long Papers)*, pages 8688–8704, Bangkok, Thailand. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *Preprint*, arXiv:2009.13570.

Thomas Mejtoft, Sarah Hale, and Ulrik Söderström. 2019. Design friction. ECCE ’19, page 41–44, New York, NY, USA. Association for Computing Machinery.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Mohammad Naiseh, Reem S. Al-Mansoori, Dena Al-Thani, Nan Jiang, and Raian Ali. 2021. Nudging through friction: An approach for calibrating trust in explainable ai. In *2021 8th International Conference on Behavioral and Social Computing (BESC)*, pages 1–5.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking. In *NeurIPS 2018, 2nd Conversational AI workshop*.

Claudio Novelli, Mariarosaria Taddeo, and Luciano Floridi. 2022. Accountability in artificial intelligence: What it is and how it works. *AI & Society: Journal of Knowledge, Culture and Communication - Springer*.

OpenAI and et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Samir Passi and Mihaela Vorvoreanu. 2022. Overreliance on ai: Literature review. Technical report, Microsoft Technical Report MSR-TR-2022-12. Microsoft Corporation.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Yi-Jyun Sun, Suvodip Dey, Dilek Hakkani-Tur, and Gokhan Tur. 2024. Confidence estimation for llm-based dialogue state tracking. *Preprint*, arXiv:2409.09629.

Gemma Team and et al. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Fanghua Ye, Yue Feng, and Emine Yilmaz. 2022a. ASSIST: Towards label noise-robust dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2719–2731, Dublin, Ireland. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022b. 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, Edinburgh, UK. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. *Preprint*, arXiv:2101.09374.

Fanghua Ye, Xi Wang, Jie Huang, Shenghui Li, Samuel Stern, and Emine Yilmaz. 2022c. MetaASSIST: Robust dialogue state tracking with meta learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1169, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective sequence-to-sequence dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Mert İnan, Anthony Sicilia, Suvodip Dey, Vardhan Dongre, Tejas Srinivasan, Jesse Thomason, Gökhan Tür, Dilek Hakkani-Tür, and Malihe Alikhani. 2025. Better slow than sorry: Introducing positive friction for reliable dialogue systems. *Preprint*, arXiv:2501.17348.

A Appendix

A.1 Selection of λ

Table 8 shows the joint goal accuracy for different λ (in Eqn. 4) on both the MultiWOZ and Snips validation sets. Table 8 shows the results with the Llama 3.1 model. We can observe that $\lambda = 0.25$ results in the best validation performance in our setup. It also results in the best validation performance for Mistral and Gemma on the MultiWOZ dataset. However, Mistral and Gemma show the best validation performance for the Snips dataset with $\lambda = 0.1$.

A.2 Additional Details

This section provides additional details related to model training, inference, and prompts. The experiments are performed on Nvidia A100 machines. The hidden size of Llama and Mistral is 4096 and

λ	JGA	
	MultiWOZ	**Snips**
0	64.24	91.14
0.1	66.13	92.00
0.25	**66.57**	**92.86**
0.5	66.25	92.43
0.75	65.34	92.04
1.0	65.15	91.76

Table 8: Impact of varying λ in $\mathcal{M}_{\text{account}}$ with Llama backbone on validation data.

3072 for Gemma. The number of trainable parameters in our LoRA fine-tuned models is 3.6M for Llama and Mistral. On the other hand, the trainable parameters of the Gemma model is 3.4M. We found that 8B models provided an optimal balance for training on our GPU servers. Smaller models resulted in lower performance, while larger models posed computational challenges due to resource constraints. This is why we conducted our experiments using LLaMA 3.1 (8B), Mistral (7B), and Gemma (7B). It also helped to balance between performance and computational efficiency, making our approach both scalable and practical for real-world applications. Due to memory limitations in GPU, we use a batch size of 1. We increase the effective batch size to 8 by using gradient accumulation. The training time for MultiWOZ and Snips is approximately 20 and 6 GPU hours, respectively.

During inference, the model takes only the dialogue context as input and generates the corresponding dialogue state, without relying on any oracle belief state. We do not apply any additional post-processing to enforce JSON formatting. In our experiments, all outputs adhered to the expected JSON format, and no formatting issues were observed. However, in the event of a failure, the output will default to an empty dictionary (i.e., {}). All the belief states are generated using a single run.

Table 9 shows the prompt format used to formulate the dialogue context (C_t) to generate the belief states. It contains the task information, slot descriptions, dialogue history, and output format. Note that Table 9 shows the prompt specific to Llama. The prompt remains exactly the same for Mistral and Gemma, except for the special tokens. For the Snips dataset, we use the same template with the Snips-specific slot description.

In Section 6, we used two kinds of prompts to confirm the predicted errors from the user-simulator. The prompt for confirming false positives is shown in Table 10, while the prompt for confirming false negatives is shown in Table 11.

<|begin_of_text|>You are a helpful assistant who can perform dialogue-state tracking. The user interacts with the system to book entities from multiple domains (hotel, restaurant, attraction, taxi, and train) in Cambridge. Your goal is to find all the intents shown by the user in the conversation.

The user can ask for a hotel by slots - hotel-name, hotel-type, hotel-parking, hotel-area, hotel-bookday, hotel-bookstay, hotel-internet, hotel-bookpeople, hotel-stars, hotel-pricerange. The user can ask for an attraction by slots - attraction-name, attraction-type, attraction-area. The user can ask for a restaurant by slots - restaurant-name, restaurant-food, restaurant-area, restaurant-bookday, restaurant-booktime, restaurant-bookpeople, restaurant-pricerange. The user can ask for a taxi by slots - taxi-arriveby, taxi-departure, taxi-leaveat, taxi-destination. The user can ask for a train by slots - train-arriveby, train-day, train-leaveat, train-destination, train-departure, train-bookpeople. Do not capture any other slots!

Task
You will be provided with a chronological dialogue history between the system and the user. You must find all the user intents and output them in JSON format.

Sample Output
{"restaurant-name": "abc", "restaurant-food": "xyz"}
Conversation History
<|start_header_id|>system<|end_header_id|>
S_0<|eot_id|>
<|start_header_id|>user<|end_header_id|>
U_0<|eot_id|>
...
<|start_header_id|>assistant<|end_header_id|>

Table 9: Prompt template for Llama model. S_0 and U_0 denote the system and user utterance for the Turn 0.

Dialogue history:
{Current Dialogue History}

It seems that some information might have been incorrectly predicted. Based on the current user utterance, please help us clarify the following:

1. If the slot name is correct but the value is wrong, you can provide the correct value.
2. If the slot name should not appear at all, let us know to delete it.

Statement: Slot was mentioned in the dialogue history and the value is Value.
- Please respond with one of the following options:
- "Agree" (if both the slot name and value are correct)
- "Not Agree: Update to [new value]" (if the slot name is correct but the value is wrong)
- "Not Agree: Delete" (if the slot name should not appear in the belief state)

Table 10: Prompt template for clarifying false positive with user simulator.

Dialogue history:
Current Dialogue History

It seems that some important information might be missing.
Based on the latest user utterance, could you please confirm the following:
Should the Slot be "Value"?

- Please respond with "Agree" if this information is correct.
- Respond with "Not Agree" if this information is incorrect.

Your confirmation will help us improve the accuracy of the prediction.

Table 11: Prompt template for clarifying false negative with user simulator.