Beyond Task-Oriented and Chitchat Dialogues: Proactive and Transition-Aware Conversational Agents

Yejin Yoon† Yuri Son† Namyoung So† Minseo Kim†
Minsoo Cho‡ Chanhee Park‡ Seungshin Lee‡ Taeuk Kim†*

†Hanyang University ‡Hyundai Motor Company
{stillwithyou,yurison,thskadud,er1123090,kimtaeuk}@hanyang.ac.kr

Abstract

Conversational agents have traditionally been developed for either task-oriented dialogue (TOD) or open-ended chitchat, with limited progress in unifying the two. Yet, real-world conversations naturally involve fluid transitions between these modes. To address this gap, we introduce **TACT** (**T**OD-**A**nd-**C**hitchat **T**ransition), a dataset designed for transition-aware dialogue modeling that incorporates structurally diverse and integrated mode flows. TACT supports both user- and agent-driven mode switches, enabling robust modeling of complex conversational dynamics. To evaluate an agent's ability to initiate and recover from mode transitions, we propose two new metrics—Switch and Recovery. Models trained on TACT outperform baselines in both intent detection and mode transition handling. Moreover, applying Direct Preference Optimization (DPO) to TACT-trained models yields additional gains, achieving 75.74% joint mode-intent accuracy and a 70.1% win rate against GPT-4o in human evaluation. These results demonstrate that pairing structurally diverse data with DPO enhances response quality and transition control, paving the way for more proactive and transition-aware conversational agents.

1 Introduction

Conversational agents are generally classified into two types: task-oriented dialogue (TOD) systems and chitchat models for open-domain social interaction. TOD systems (Hosseini-Asl et al., 2020) follow predefined workflows to accomplish user goals, while chitchat models (Wu and Yan, 2018) generate contextually appropriate responses.

With the aid of large language models (LLMs), research bridging the two traditionally independent paradigms has gained momentum (Sekulic et al., 2024). However, current efforts overlook a core challenge: most dialogue systems remain predominantly *reactive*, focusing on responding to user inputs rather than *proactively* coordinating dialogue flows (Yi et al., 2024; Acikgoz et al., 2025b).

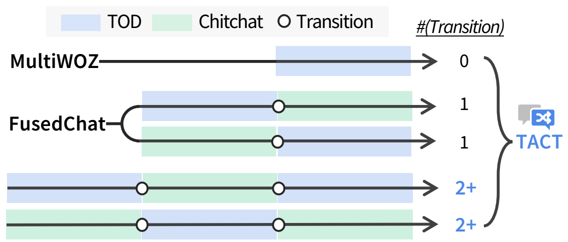


Figure 1: Comparison of dialogue flows across MultiWOZ, FusedChat, and TACT. MultiWOZ contains only task-oriented dialogues (TODs) with no mode transitions, whereas FusedChat allows a single transition between TOD and chitchat. In contrast, TACT supports multiple mode transitions (2+), enabling training on complex dialogues with diverse switching patterns.

In real-world conversations, users frequently shift between task-oriented dialogue (TOD) and chitchat within a single session, as empirically observed in deployed systems (Rim et al., 2025).[1] To manage these dynamic transitions, agents must predict mode shifts, take initiative, and maintain coherent multi-turn interactions. This requires two key abilities: (1) *transition-awareness*, for detecting and adapting to mode changes, (2) *proactivity*, to plan ahead and guide the conversation flow when appropriate—both of which are essential yet underexplored (Yi et al., 2024; Acikgoz et al., 2024).

In this work, we present a framework for building *proactive* and *transition-aware* conversational agents, grounded in our novel **TACT**[2] (**T**OD-**A**nd-**C**hitchat **T**ransition) dataset. As illustrated

*Corresponding author

[1]For example, during a conversation with a ticket-booking agent, a user might share past travel experiences. Ideally, the agent should respond appropriately to the digression, then autonomously return to the booking task when appropriate to fulfill its objective. See Figure 2 for a similar case study.

[2]Available at https://github.com/HYU-NLP/TACT.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13280–13306
November 4-9, 2025 ©2025 Association for Computational Linguistics

Dataset	SalesBot2.0	FusedChat	InterfereChat	TACT	
Seed	SalesBot1.0	MultiWOZ2.4	FusedChat	MultiWOZ2.2	SLURP
# Intents	6	11	11	11	50*
# Dialog	5,453	10,436	4,475	7,199	9,936
# Avg. Turn	7.71	18.36	13.58	15.04	16.42
# Avg. Switch	0.96	1	0*	1.93	2.06
# Avg. Recov.	-	0	0*	0.93	1.07
# Uniq. Flow	2	2	1	11	12
Flow Types	CT	TC, CT	T*	TCT, CTC, TCTCT, etc.	

Table 1: Statistics of existing datasets and the newly proposed TACT variants. Asterisks (*) indicate factors affected by the pre-processing performed in this work. **Recov.**: recovery. **# Uniq. Flow**: the number of unique dialogue mode transition patterns.

in Figure 1, TACT features complex dialogue sessions in which task-oriented and chitchat modes frequently alternate. In contrast to prior mode-switching datasets (Chang and Chen, 2024; Young et al., 2022; Stricker and Paroubek, 2024a), which lack structural diversity (see Table 1[3]), TACT provides richer interactions that better support the training of agents to detect and manage mode shifts with both transition-awareness and proactivity. For instance, Figure 2 showcases a scenario in which only the TACT-based agent successfully returns to the original task after an interruption, whereas other models fail to recover and remain off track.

Furthermore, we define metrics to quantify how often agents attempt to switch or recover dialogue modes, and whether those transitions succeed based on user responses. Compared to previous datasets, agents trained on TACT consistently handle flow switching and recovery more effectively, while maintaining strong performance on standard TOD and chitchat tasks. Beyond structural coverage, we further adopt Direct Preference Optimization (DPO; Rafailov et al. (2023)) to align model outputs with human preferences, which significantly enhances response quality and transition naturalness. These results suggest the potential to develop more autonomous and predictive conversational agents, surpassing the current standard where mere response accuracy is considered sufficient.

2 Related Work

Datasets for dialogue mode switching Previous efforts to unify TOD and chitchat have largely relied on augmenting existing TOD corpora. FusedChat (Young et al., 2022) and InterfereChat (Stricker and Paroubek, 2024a) inject chitchat turns into TOD, typically at fixed points or as single exchanges, hereby adhering to a TOD-centric perspective. Such simple rule-based modification restricts TOD-chitchat variations, rendering these resources unsuitable for modeling dynamic dialogue transitions. To overcome this shortcoming and support rich transitions, we construct two TACT variants based on SLURP (Bastianelli et al., 2020) and MultiWOZ 2.2 (Zang et al., 2020), featuring diverse intents and interwoven mode switches.

Methods for dialogue mode switching While prior research, e.g., SimpleTOD (Hosseini-Asl et al., 2020) and SalesAgent (Chang and Chen, 2024), have made progress in unifying TOD and chitchat, these methods fall short in managing fluent mode control across multi-turn interactions. Recent work has extended these efforts by integrating TOD and chitchat systems (Accentor; Sun et al. (2020)) or employing function calling (Stricker and Paroubek, 2024b; Li et al., 2024b) for accurate intent tracking. However, a critical aspect remains underexplored: the ability to recover and re-engage with relevant dialogue history at appropriate turns.

Programmable control Prior work on controllable dialogue frameworks (Rebedea et al., 2023) enables rule-driven flow control, effective when conversation graphs are known in advance. However, while these systems achieve tight control over dialogue flows—akin to classic pipelines (Bocklisch et al., 2017; Liu et al., 2021)—they are inherently rigid and costly to adapt to new scenarios. In contrast, we address more challenging and practical settings where dialogues shift dynamically between chitchat and tasks without rigid scripts. To this end, we frame transition handling as a learning problem and construct TACT to capture such phenomena without predefined flows. Building on this, we propose a unified framework that fine-tunes LLMs with TACT for proactive transition and flow control, while incorporating response-level preferences through DPO.

3 Dataset Construction: TACT

In this section, we design the **TACT** (**T**OD-**A**nd-**C**hitchat **T**ransition) dataset to support agents that manage both TOD and chitchat, through coherent multi-turn dialogues. Unlike prior datasets, it regards mode transitions as recoverable and context-

[3]We treat InterfereChat solely as T(OD), as chitchat utterances always coexist with TOD turns. We also re-define the intent space of SLURP to eliminate overlaps between intents. We refer readers to Table 8 for more details.

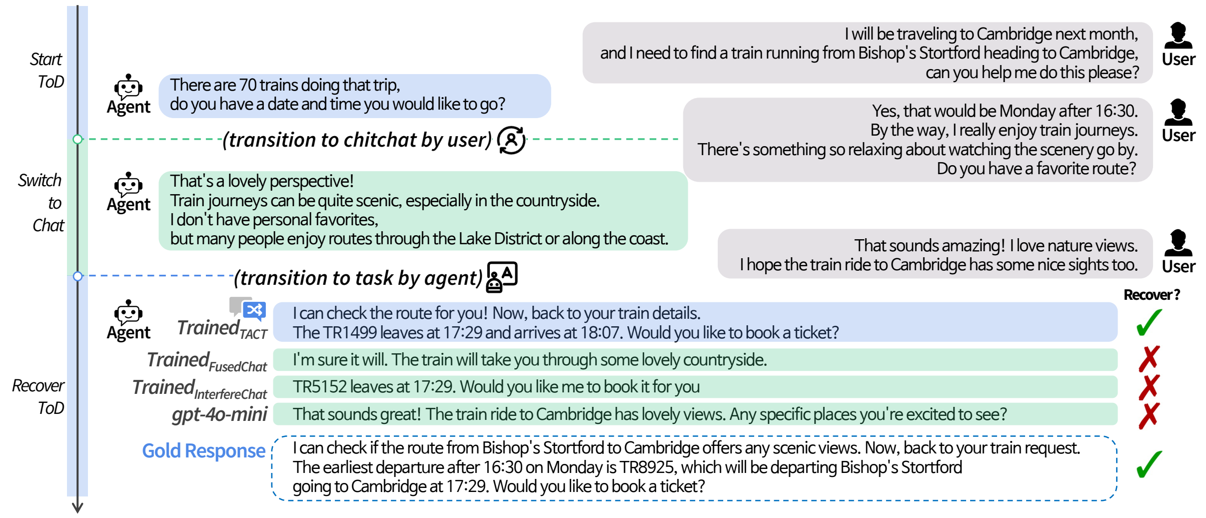


Figure 2: A scenario demonstrating that only **Agent-Trained$_{\text{TACT}}$** exhibits transition-awareness and proactivity by successfully returning to the task after a chitchat. Note that all other baselines fail to recover the original context.

sensitive, allowing agents to decide when to switch and how to resume back to previous modes. We release the dataset as open-source to support future research in multi-turn dialogue systems.[4]

3.1 Dialogue Generation

To support structured modeling of mode transitions, TACT defines two core dialogue flow types: **TCT** (TOD ↦ Chitchat ↦ TOD) and **CTC** (Chitchat ↦ TOD ↦ Chitchat). These flow types simulate patterns observed in real-world conversations, where users often deviate from a task and later return to it, or casually initiate and conclude chitchat around brief task-oriented exchanges.

We construct such flows based on existing TOD corpora—MultiWOZ2.2 and SLURP —by augmenting them with chitchat in contextually appropriate locations (see Figure 3 and Appendix A for details). This approach supports modeling of both transition awareness and recovery, which are critical for fluid, mode-integrated dialogue.

TCT flow We extract task segments of four or more turns from MultiWOZ2.2, and insert a chitchat block at a natural boundary between intents. The chitchat briefly diverges from the task, often reflecting personal curiosities or preferences, before the dialogue returns to its original goal.

CTC flow We begin with a short TOD segment (2–3 turns) and attach chitchat before and after the task, forming a wrap-around flow. This simulates cases where users casually engage in a task during a social exchange and then resume chitchat.

Validation Approaches	Human-Annotated Criteria	LLM-Generated Task Description	LLM-Generated Evaluation Steps
Active Critic	✗	✓	✗
G-Eval	✓	✗	✓
Ours	✓	✓	✓

Table 2: Comparison of data validation methods. Only our framework supports both human-aligned criteria and LLM reasoning. ✓ indicates feature presence.

3.2 Dataset Validation

To ensure the quality of TACT at scale, we develop an automatic validation pipeline fusing human-aligned criteria with model-based reasoning. Specifically, G-Eval (Liu et al., 2023) offers well-designed evaluation criteria aligned with human judgment, but lacks internal reasoning. In contrast, Active-Critic (Xu et al., 2024) induces task-specific criteria through reasoning over examples, but it is not designed for dialogue evaluation and does not use predefined human-authored standards. Table 2 shows that our method uniquely combines human-authored criteria with model-based reasoning.[5] Unlike G-Eval or Active-Critic, which cover only one of these aspects, our hybrid framework supports scalable and interpretable validation.

Dialogues are evaluated with a prompt that includes a full conversation, a task description, evaluation criteria, and step-by-step reasoning. The judging model, GPT-4O-MINI (Hurst et al., 2024),

[4] https://huggingface.co/datasets/HYU-NLP/TACT.

[5] Full prompt templates are provided in Figure 12.

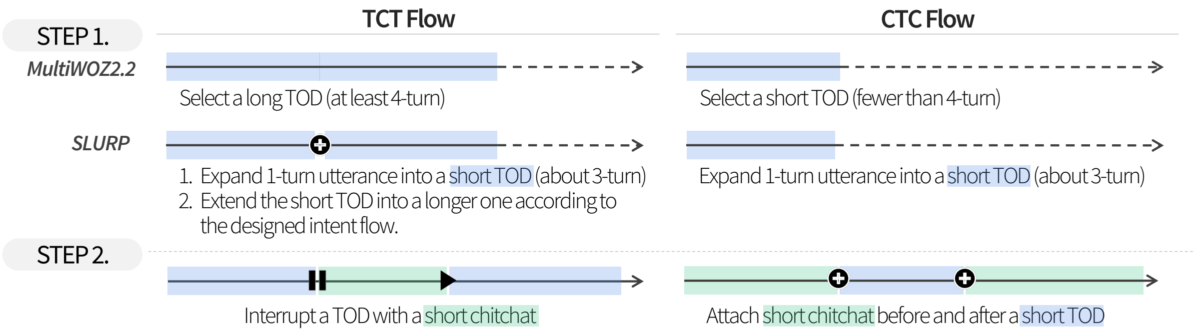


Figure 3: TACT construction steps for TCT and CTC dialogues from MultiWOZ and SLURP.

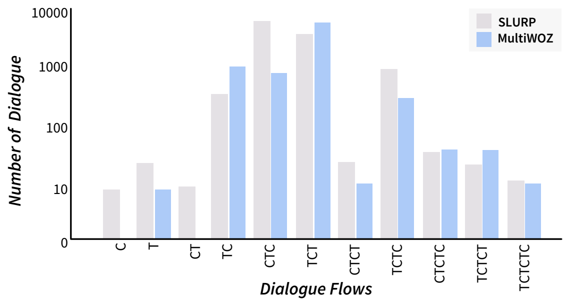


Figure 4: Dialogue flow distribution in $\text{TACT}_{\text{SLURP}}$ and $\text{TACT}_{\text{MultiWOZ}}$.

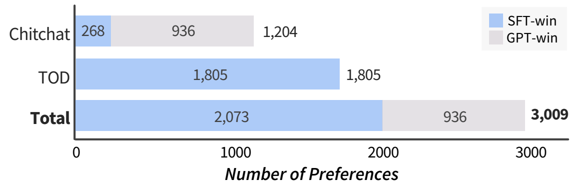


Figure 5: Distribution of response preferences between the SFT and GPT models, evaluated by GEMINI-2.5-PRO on $\text{TACT}_{\text{MultiWOZ}}$. For TOD dialogues, SFT-generated responses are consistently favored, whereas chitchat reveals a more mixed preference landscape.

scores each dialogue on three criteria: (1) **Intent Accuracy**, which checks whether the user intent is correctly conveyed and executed; (2) **Transition Quality**, which assesses whether mode switches are contextually justified; and (3) **Dialogue Naturalness**, which evaluates fluency and coherence.

3.3 Dataset Characteristics

TACT is structurally designed to support robust learning of dialogue transitions through two key properties: diverse multi-turn flow patterns and recoverable dialogue structures.

Diverse transition flows As illustrated in Table 1, TACT supports a wide variety of dialogue flows, including TCT, CTC, TCTCT, and others. TACT is the first dataset to combine interwoven transition patterns, a balanced distribution of transition initiators, and recoverable dialogue structures. Figure 4 reveals that while $\text{TACT}_{\text{MultiWOZ}}$ examples concentrate more on TCT and TC flows, $\text{TACT}_{\text{SLURP}}$ covers a broader spectrum, enabling agents to generalize across diverse flow structures.

Recoverable structures Differing from existing datasets, TACT includes dialogues that explicitly return to a previously suspended mode, enabling agents to learn dialogue-level consistency. For example, Figure 2 shows a TCT dialogue where only the TACT-trained model resumes the original task after a chitchat interruption. We provide a detailed evaluation of recovery performance in § 6.

4 Methodology

We evaluate 4 methods for unified TOD-chitchat response generation: (1) **in-context learning (ICL)**, (2) **supervised fine-tuning (SFT)** (3) **Direct Preference Optimization (DPO)**, our main approach, which aligns model outputs with response preferences, and (4) **pipeline-based methods** that separate mode prediction from response generation.

4.1 In-Context Learning

We use GPT-4o (Hurst et al., 2024) to explore the potential of ICL for unified TOD-chitchat dialogue modeling, on both zero-shot and few-shot prompting setups. Prompt formats and example inputs are explained in Appendix C.1. In the **zero-shot** setting, the model receives a task instruction and dialogue history as input, and is asked to predict both the current mode and corresponding responses. In the **few-shot** setting, we provide four annotated exemplars: two for TCT flows and two for CTC.

Training Set	Test Set	TOD						Flow			
		Mode Selection		Intent Detection		Joint Accuracy		Switch		Recovery	
		Acc.	F1	Acc./turn	Acc./dialogue	Acc./turn	Acc./dialogue	Attempt	Success	Attempt	Success
FusedChat	MultiWOZ	98.44	76.71	93.79	69.80	93.57	68.90	0.000	0.000	-	-
	FusedChat	97.00	96.70	92.85	66.90	94.20	60.50	0.000	0.000	-	-
	InterfereChat	97.04	94.68	93.21	67.62	93.09	61.89	0.000	0.000	-	-
	TACT$_{\text{MultiWOZ}}$	91.79	87.27	94.46	72.65	88.13	33.24	0.000	0.000	-	-
	Average	96.07	88.84	93.58	69.24	**92.25**	56.13				
InterfereChat	MultiWOZ	98.27	75.97	93.92	70.90	93.74	70.10	0.000	0.000	-	-
	FusedChat	79.92	73.43	92.85	67.10	76.19	7.10	0.000	0.000	-	-
	InterfereChat	97.63	95.72	93.28	68.44	93.34	64.34	0.000	0.000	-	-
	TACT$_{\text{MultiWOZ}}$	79.26	58.41	93.95	71.34	84.33	34.89	0.000	0.000	-	-
	Average	88.77	75.88	**93.84**	**70.62**	84.82	35.39				
TACT$_{\text{MultiWOZ}}$	MultiWOZ	98.06	74.91	92.70	66.20	92.57	65.50	0.000	0.000	1.000	< 0.001
	FusedChat	90.63	89.05	92.57	65.70	87.08	34.10	0.160	0.008	1.000	< 0.001
	InterfereChat	97.32	95.20	92.13	64.14	92.38	59.22	0.619	0.309	0.013	0.104
	TACT$_{\text{MultiWOZ}}$	98.95	98.50	96.35	80.94	96.41	75.59	1.322	1.300	0.977	0.856
	Average	**96.24**	**89.42**	93.44	69.25	92.11	**58.60**				

Table 3: Comparison of SFT variants trained on different datasets. The TACT-based agent performs competitively across diverse metrics. The best average score in each column is in **bold**; '< 0.001' indicates a value less than 0.001.

4.2 End-to-End Generation with Finetuning

As our SFT baseline, we adopt **FnCTOD** (Li et al., 2024a), originally developed for zero-shot dialogue state tracking in unified TOD-chitchat scenarios using structured function calls.[6] We reinterpret the function-calling mechanism as a structured intent representation: at each dialogue turn, the model first predicts an intent based on the user input, then produces a response conditioned on that intent. The model is trained on TACT using system-provided function schemas in the prompt, enabling unified intent prediction and response generation within a single auto-regressive decoding process.

4.3 DPO for Unified TOD-Chitchat Modeling

To improve both response quality and transition handling in hybrid dialogues, we further apply **DPO** (Rafailov et al., 2023) to the FnCTOD model trained on TACT. While SFT enables basic task completion, we observed its clear limitations in sensibleness, tone, and flow continuity, as reflected in the win rate outcomes of Figure 7. Such shortcomings are consistent with recent findings (Chu et al., 2025) that SFT tends to emphasize memorization over generalization, whereas RL-based methods encourage more robust adaptation.

To this end, we apply DPO as a way to steer the model toward more desirable behaviors. By distinguishing between preferred and non-preferred responses, DPO guides the model toward accurate and complete outputs in TOD, while encouraging natural tone, continuity, and fluency in chitchat. In this setting, TACT exposes the model to dialogues where transitional contexts are clearly present and preferences can be readily determined.

Each training instance consists of an input prompt and two candidate responses: one *preferred* and one *rejected*. We generate these pairs by comparing outputs from the FnCTOD and GPT-4O-MINI models, with preferences determined by the GEMINI-2.5-PRO (DeepMind, 2023) judge according to the criteria in §5.3—i.e., sensibleness, specificity, interestingness, and transition naturalness.

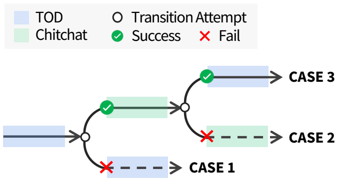
The final dataset consists of 3,009 preference-labeled pair instances, as summarized in Figure 5. Training on this dataset with DPO equips the model to balance precise task intent prediction with fluid, engaging conversational behavior. We refer to the resulting model as **(SFT-)DPO**. To the best of our knowledge, this is the first application of DPO in a unified dialogue generation setting that combines structured function-calling with TOD and chitchat.

4.4 Pipeline Approach

Previous work on unified TOD–chitchat modeling typically adopts a modular pipeline: a classifier selects the dialogue mode, followed by a dedicated TOD or chitchat module for response generation (Young et al., 2022). To facilitate comparison with this paradigm, we construct a **generative-classifier-based pipeline**.[7] The FnCTOD model trained on

[6] Among several candidate architectures—including SimpleToD (Hosseini-Asl et al., 2020) and SalesAgent (Chang and Chen, 2024)—FnCTOD demonstrated the best overall performance in our preliminary experiments (see Appendix E.1) and was thus selected as the main SFT approach.

[7] We also attempted to implement a variant with a BERT-based mode detector. However, this approach performed poorly in mode selection and exhibited low transition fluency. We refer readers to Appendix E.1 for more details.



CASE 3	Attempt	Success
Switch	+2	+2
Recovery	+1	+1
CASE 2	Attempt	Success
Switch	+2	+1
Recovery	+1	X
CASE 1	Attempt	Success
Switch	+1	X
Recovery	-	-

Figure 6: Visualization of **Switch** and **Recovery** metrics. **Case 1**: Attempted Switch, but unsuccessful; **Case 2**: Successful switch but failed recovery; **Case 3**: Both switch and recovery succeed.

TACT predicts a mode label and an intent. If the mode is TOD, it generates the response; otherwise, GPT-4O-MINI is used with in-context prompts to generate a chitchat response.

5 Experimental Setup

5.1 Datasets and Tasks

We first investigate the performance of SFT-based model variants trained separately on four different datasets—**MultiWOZ2.2** (Zang et al., 2020), **FusedChat** (Young et al., 2022), **InterfereChat** (Stricker and Paroubek, 2024a), and our proposed **TACT**$_{\textbf{MultiWOZ}}$[8]—for TOD–chitchat unification modeling. The objective is to evaluate the impact of training data, demonstrating TACT's effectiveness in activating transition-awareness and proactivity, while maintaining overall performance.

In the second part, we compare four modeling strategies to measure their relative effectiveness in terms of leveraging TACT: **ICL** using GPT-4O (zero- and few-shot), **SFT** with FnCTOD, **DPO**, and a **generative-classifier-based pipeline**.

5.2 Training Configuration

All SFT models, including DPO and pipeline variants, are initialized with LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024) and fine-tuned using DeepSpeed ZeRO-3 (DeepSpeed, 2021) with bf16 precision for efficiency and stability. We train each model for 3 epochs with a fixed learning rate of 1×10^{-5} and a batch size of 256. These hyper-parameters follow prior work, ensuring convergence and robust performance. Implementation details are provided in Appendix D.1.

5.3 Evaluation Metrics

Existing evaluation metrics for dialogue systems, e.g., slot accuracy, goal completion rate, and BLEU (Papineni et al., 2002), are mostly focused on TOD (Wen et al., 2017; Rastogi et al., 2020). However, they are insufficient for considering scenarios involving frequent mode transitions between TOD and chitchat. A contemporary study (Acikgoz et al., 2025a) also highlights their inability to capture mid-dialogue failures and long-range inconsistencies.

To tackle this, we adopt a transition-aware evaluation framework with three categories: TOD-centric measures, chitchat response quality (i.e., win rate), and transition-aware flow metrics.[9]

TOD-centric metrics We use three metrics to assess a model's task-handling abilities. **(1) Mode Selection Accuracy and F1:** Each turn is labeled as TOD or chitchat, and the model is evaluated for both overall accuracy and F1 scores. **(2) Intent Detection Accuracy:** For TOD-predicted turns, we evaluate whether the model accurately identifies the user's intent. Accuracy is reported at both the turn and dialogue level, where the latter requires perfect intent prediction across all TOD turns. **(3) Mode+Intent Joint Accuracy:** This metric checks if the dialogue mode and, when in TOD, the intent are correctly predicted at each turn, yielding end-to-end accuracy in integrated-mode settings.

Chat response quality metrics We assess chat responses on four criteria: *Sensibleness*, *Specificity*, *Interestingness* (Thoppilan et al., 2022), and *Transition Naturalness*, which evaluates the contextual appropriateness of mode switches. For each criterion, we conduct pairwise comparisons between the target model and GPT-4O (few-shot), with judgments provided either by an LLM evaluator[10] or by human annotators. The pairwise win/lose outcomes are aggregated into win rates for each criterion. To mitigate verbosity bias (Dubois et al., 2024; Hu et al., 2024), we apply length-controlled prompting for GPT-4O (see Appendix D.2.3).

Transition-aware metrics To evaluate an agent's ability to manage mode transitions, we propose two transition-aware metrics: **Switch**—when the agent shifts from one mode to another (e.g., TC, CT)—and **Recovery**—when the agent returns to a previously suspended mode (e.g., TCT, CTC). For each metric, we report two statistics: (1) **At-**

[8]Results on **TACT**$_{\textbf{SLURP}}$ are presented in Appendix E.3, showing similar trends with results discussed in §6.

[9]Note that TOD-centric and chitchat response metrics are reported as percentages (w/o the % symbol), while transition-aware measures are based on average attempts and successes.

[10]To mitigate bias in model-based evaluation (Li et al., 2025; Wataoka et al., 2024), we employ GEMINI-2.5-PRO as the judge, chosen for its independence from all evaluated models.

	TOD						Flow				Chitchat
	Mode Selection		Intent Detection		Joint Accuracy		Switch		Recovery		Overall
Method	Acc.	F1-score	Acc./turn	Acc./dialogue	Acc./turn	Acc./dialogue	Attempt	Success	Attempt	Success	Win-Rate
ICL-ZS	90.46	86.21	87.57	50.44	85.01	30.00	0.879	0.374	0.880	0.099	-
ICL-FS	91.45	88.98	84.09	40.00	86.89	36.76	**1.577**	0.865	**1.571**	0.652	-
SFT	**98.95**	**98.50**	**96.35**	**80.94**	**96.41**	75.59	1.322	1.300	0.977	0.856	23.16
SFT-DPO	98.82	98.32	96.03	80.00	96.21	**75.74**	1.343	**1.322**	0.977	**0.859**	**40.86**
Pipeline	**98.95**	**98.50**	**96.35**	**80.94**	**96.41**	75.59	1.322	1.300	0.977	0.856	24.32

Table 4: Method comparison across TOD, chitchat, and transition-aware metrics: ICL (zero- and few-shot with GPT-4o), SFT, (SFT-)DPO, and a generative-classifier-based Pipeline. For Pipeline, TOD metrics are inherited from the SFT model. The best score in each column is in **bold**.

tempt: the average number of *Switch/Recovery* attempted ***by the model*** per dialogue, (2) **Success**: the average number of *Switch/Recovery* in which the user accepts the agent-driven mode shift and responds accordingly. As shown in Figure 6, Attempt is counted every time the agent suggests a possible mode transition, but it is only considered *successful* when the user subsequently accepts the suggested mode transition. Formal definitions are provided in Appendix D.2.

6 Experimental Results

6.1 Quantitative Evaluation

Comparison across datasets In Table 3, we compare SFT models trained on different TOD-chitchat unification datasets. The model trained on TACT$_{\text{MultiWOZ}}$ achieves equal or superior performance on all TOD-centric metrics, compared to those trained on FusedChat or InterfereChat.

While FusedChat and InterfereChat-trained models perform well on their respective test sets, they generalize poorly to others. In contrast, the TACT-trained one consistently achieves strong TOD performance and is the only one capable of handling multi-turn transitions and recoveries—a direct reflection of the dataset design described in §3.

Notably, only the TACT-trained agent achieves *non-zero* transition-aware scores. Other datasets produce no valid attempts, due to their lack of recoverable or multi-turn transition structures.

Comparison across methods We compare four methods for handling dialogue with transitions: zero- and few-shot prompting, SFT, a transition-aware pipeline, and preference-tuned DPO. Table 4 reveals that both SFT and the pipeline outperform ICL on TOD-centric metrics, especially in intent detection and mode+intent joint accuracy. While DPO slightly underperforms on these metrics, it achieves the highest dialogue-level joint accuracy at 75.74%, suggesting greater consistency and better alignment with user intent.

DPO also outperforms all other methods in chitchat quality, recording the highest win rate at 40.86% against GPT-4o in pairwise evaluation. It also demonstrates the most *proactive* behavior, with over one successful mode switch and recovery per dialogue—i.e., 1.322 switch attempts with 1.300 successes, and 0.977 recovery attempts with 0.856 successes on average. We further analyze its qualitative strengths in §6.2.

In contrast, ICL-based models—both zero-shot and few-shot—consistently underperform across most metrics, highlighting the limitations of prompt-only adaptation in handling complex dialogue transitions. Although ICL-FS makes the most switch and recovery attempts, its low success rate suggests a tendency to excessively initiate transitions without sufficient contextual grounding.

In summary, experimental results confirm that DPO—a preference-tuned extension of SFT—effectively replaces modular pipelines with a single robust model that jointly handles intent prediction, dialogue flow control, and chitchat generation. Appendix E.2 further shows that these trends hold consistently across different underlying models.

6.2 Preference-Based Evaluation

Based on the evaluation criteria in §5.3, we conduct a preference-based analysis of model responses in integrated-mode dialogues, using two types of evaluators: an LLM judge (GEMINI-2.5-PRO) and human annotators (10 evaluators, including both NLP practitioners and general users). The interface provided to human annotators, which includes evaluation instructions, is shown in Appendix D.2. The LLM judge compares responses from each tested model and GPT-4o (few-shot) over the entire test set, producing win/tie/lose outcomes for each criterion (Figure 7). In parallel, the human study was

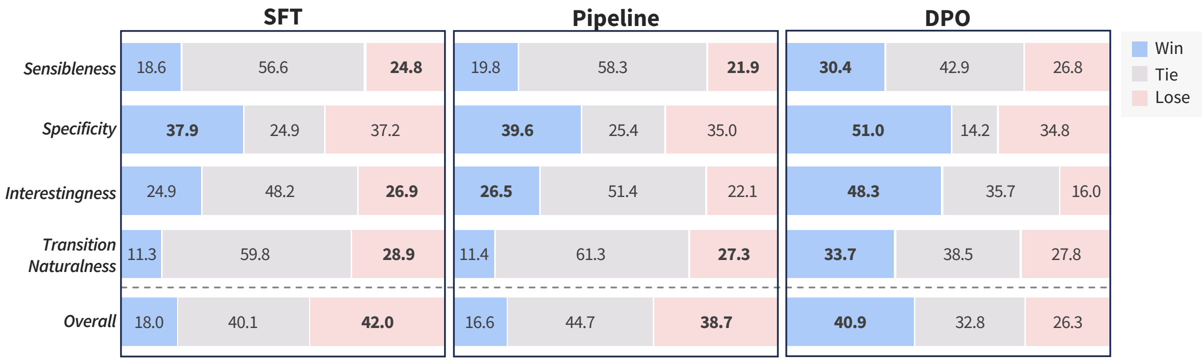


Figure 7: Preference-based evaluation of SFT, Pipeline, and DPO models (trained on TACT) against GPT-4o (few-shot). An LLM judge (GEMINI-2.5-PRO) assessed sensibleness, specificity, interestingness, and transition naturalness, with win/tie/lose proportions shown.

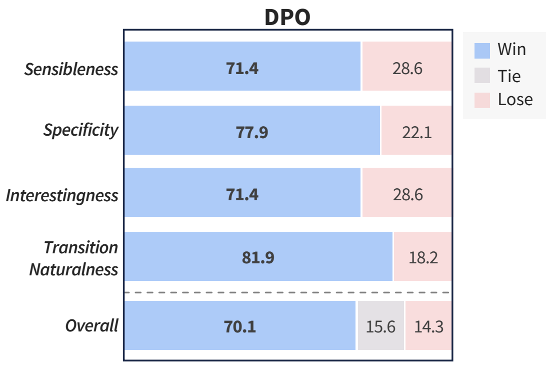


Figure 8: Human preference evaluation of the DPO model trained on TACT against GPT-4o (few-shot), with over 70% overall wins across 77 dialogues.

conducted on a 77-sample subset of the test set, where annotators evaluated responses without a tie option, providing an additional perspective aligned with real user interactions (Figure 8).[11]

DPO outperforms GPT-4o across all four qualitative criteria, with strong gains in *Interestingness* and *Transition Naturalness*. These results demonstrate that preference tuning enables models to internalize soft conversational qualities such as engagement and flow continuity. In *Interestingness*, DPO more than doubles the win rate, indicating improved expressiveness and user engagement. For *Transition Naturalness*, DPO surpasses GPT-4o for the first time with a win rate of 33.7%, compared to SFT's 11.3%.

Human judgments make this contrast even clearer: While the LLM judge recorded only around 40% win rates with many ties, human annotators showed over 70% wins with sharply reduced ties.[12]

[11]No ties per criterion, but possible overall (2–2 split).

[12]This is partly due to the no-tie setup, but the significant margins indicate that humans clearly recognize DPO's merits.

Method	Success	intent ✓	intent ✗
ICL-FS	0.652	33.89	66.11
SFT	0.856	34.58	65.42
SFT-DPO	0.859	34.23	65.77

Table 5: Recovery Success analysis at mode- and intent-levels. **intent** ✓ shows the proportion of recoveries that return to the previous intent, and **intent** ✗ reflects the proportion that initiate a new one.

The advantage holds across all four qualitative criteria, demonstrating that DPO's gains are overwhelmingly clear to human users.

In addition, case studies with real output examples in Appendix E.4 show that DPO responses more frequently exhibit contextually anchored reactions—such as callbacks and affective tone—and generate smoother transitions within dialogues.

6.3 Transition-Focused Analysis

Recovery success analysis We further break down the Recovery Success metric (§5.3) by examining whether the resumed task continues the same intent as before the transition. Table 5 reports that only about 34% of successful recoveries return to the previous intent (**intent** ✓), while the rest initiate a new intent within the same mode (**intent** ✗). This indicates that in realistic scenarios, successful recovery does not always require returning to the original intent. Depending on the dialogue context, initiating a new but relevant intent can be just as appropriate—as long as the interaction resumes smoothly within the correct mode.

Analysis by dialogue flow type In Table 6, we present an ablation study evaluating the performance of SFT agents across diverse dialogue mode flows. We focus on the SFT agent rather than DPO,

Flow type	TOD						Flow			
	Mode Selection		**Intent Detection**		**Joint Accuracy**		**Switch**		**Recovery**	
(# Dialogues)	**Acc.**	**F1**	**Acc./turn**	**Acc./dialogue**	**Acc./turn**	**Acc./dialogue**	**Attempt**	**Success**	**Attempt**	**Success**
TACT$_{\text{MutliWOZ}}$ (680)										
TCT (533)	99.35	99.04	96.64	82.18	97.18	81.24	1.390	1.372	1.006	0.977
CTC (38)	97.06	96.51	96.30	92.11	97.06	81.58	0.421	0.342	0.313	0.000
TC (74)	95.35	97.99	95.35	74.32	95.18	64.86	1.419	1.378	1.000	0.069
TCTC (27)	97.35	96.94	97.39	85.19	96.46	70.37	1.222	1.185	0.958	0.958
Others (7)	98.36	98.29	97.22	87.50	96.72	75.00	0.500	0.500	1.000	1.000
TACT$_{\text{SLURP}}$ (1,790)										
TCT (618)	99.41	99.12	94.18	72.98	95.28	72.01	0.974	0.964	1.019	0.985
CTC (907)	97.61	97.00	91.69	88.20	96.10	79.82	1.756	0.821	1.041	0.037
TC (60)	91.19	91.01	76.68	31.67	85.32	30.00	1.450	0.550	1.304	0.071
TCTC (174)	96.06	96.03	91.17	71.26	92.80	60.34	1.793	1.167	1.616	0.169
Others (31)	85.13	84.87	82.28	45.16	84.39	16.13	1.645	0.677	1.393	0.107

Table 6: Performance variation of the SFT agent across dialogue flow types. All extended variants such as TTCT, TCTT, and similar patterns are consolidated into representative flow categories. (e.g., TTCT → TCT).

since DPO primarily improves response quality in chitchat, which is beyond the scope of this analysis.

The agent performs most reliably on the TCT flow across all metrics, demonstrating high accuracy in mode selection and intent prediction, along with strong switch and recovery behavior. In contrast, in the CTC flow, the agent shows a notable gap in transition-aware metrics, especially in TACT$_{\text{MultiWOZ}}$. As shown in Table 9 of Appendix A.3, TACT contains user-driven transitions to chitchat more than agent-driven ones, likely because such transitions were perceived as more natural during the data validation process. As a result, the agent shows less transition attempts and recoveries, in CTC settings. Nevertheless, the agent maintains strong performance on the ToD-centric metrics, indicating that it effectively detects user-driven mode transitions.

The TC flow presents another challenge. Although recovery is infrequent, the agent still exhibits disproportionately high switch attempt rates (1.450 on SLURP) despite low success rates (0.550 on SLURP). This suggests mode confusion or over-triggering of switches in cases where the need for transition is minimal or poorly signaled. The model appears to misinterpret certain turns as transition points, indicating imprecise transition judgment when explicit cues for mode shifts are lacking[13]. Interestingly, the TCTC flow—though structurally more complex—yields stable recovery performance. This implies that structural complexity alone does not hinder learning and may even support it when the flow clearly supervises transitions.

Overall, this analysis reveals that while TOD performance (e.g., intent detection) remains stable across flows, flow-sensitive behaviors such as switching and recovery are highly dependent on interaction pattern and training coverage. Improving generalization thus requires not only stronger flow-aware supervision, but also better handling of ambiguous or underspecified transitions.

7 Conclusion

We present TACT, a dataset for modeling the integration of task-oriented and chitchat dialogues with natural mode transitions and recoverable structures. TACT enables training models that can manage complex dialogue flows, including multi-turn mode shifts and returns to prior tasks.

We also demonstrate that preference optimization via DPO significantly enhances both task accuracy and response quality, outperforming strong baselines across both quantitative and qualitative dimensions. In particular, DPO achieves notable gains in transition-sensitive metrics and chitchat preference, showing that soft conversational skills such as engagement and transition smoothness can be effectively learned through preference signals.

Future work includes fine-grained analysis of complex transition flows already present in TACT—such as TCCT and nested switches—and extending our modeling framework to more effectively handle such cases. We also aim to explore real-time flow tracking and preference adaptation techniques for open-ended agents that must manage dialogue continuity in dynamic settings.

[13] Note that by construction, the TC flow contains no gold-standard recovery events. However, since recovery metrics are defined on model predictions at inference, recovery success can still occur when the model first mispredicts a transition and later corrects it. This accounts for the non-zero recovery scores in the TC flow.

Limitations

Despite the promising results, our study has several limitations. First, although the TACT dataset includes structurally complex flows—such as TCCT and TCTT—we simplify these into broader categories like TCT during evaluation. This aggregation may mask flow-specific behaviors and limit fine-grained performance analysis.

Second, our qualitative evaluation relies on a single LLM-based judge (GEMINI-2.5-PRO), which may introduce biases not fully aligned with human preferences. To mitigate this, we also conducted human evaluation alongside the model-based one. Although we additionally conducted human evaluation, reliance on a single LLM judge still raises concerns about potential bias and imperfect alignment with human judgments.

Finally, we rely solely on DPO for preference tuning; it remains unclear whether the observed gains arise from DPO-specific characteristics or reflect broader advantages of preference-based learning. Future work could explore other preference optimization strategies beyond DPO—such as reward modeling or ranking-based fine-tuning—to assess whether the observed improvements generalize across different supervision formats.

Acknowledgements

This work was supported by Hyundai Motor Company and Kia. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence semiconductor support program to nurture the best talents (IITP-2025-RS-2023-00253914) grant funded by the Korea government(MSIT).

Ethics Statement

This research was conducted with a commitment to ethical standards in AI development. The proposed models and datasets were designed to improve dialogue system performance while minimizing potential biases. Our dataset construction relies exclusively on widely used, publicly available benchmarks (MultiWOZ and SLURP), which do not contain personally identifiable information (PII). The augmented dialogues were synthetically generated from these sources, further minimizing any risk of privacy leakage. While the use of benchmark datasets reduces concerns about privacy, we acknowledge that they may still reflect domain- or culture-specific biases. Our work aims to enhance conversational AI capabilities in a responsible and inclusive manner, with particular attention to supporting diverse user groups and interaction scenarios.

References

Emre Can Acikgoz, Mete Erdogan, and Deniz Yuret. 2024. Bridging the bosphorus: Advancing Turkish large language models through strategies for low-resource language adaptation and benchmarking. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 242–268, Miami, Florida, USA. Association for Computational Linguistics.

Emre Can Acikgoz, Carl Guo, Suvodip Dey, Akul Datta, Takyoung Kim, Gokhan Tur, and Dilek Hakkani-Tür. 2025a. Td-eval: Revisiting task-oriented dialogue evaluation by combining turn-level precision with dialogue-level comparisons. *arXiv preprint arXiv:2504.19982*.

Emre Can Acikgoz, Cheng Qian, Hongru Wang, Vardhan Dongre, Xiusi Chen, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. 2025b. A desideratum for conversational agents: Capabilities, challenges, and future directions. *arXiv preprint arXiv:2504.16939*.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7252–7262, Online. Association for Computational Linguistics.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Wen Chang and Yun-Nung Chen. 2024. Injecting salesperson‘s dialogue strategies in large language models with chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3798–3812, Bangkok, Thailand. Association for Computational Linguistics.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*.

Google DeepMind. 2023. Gemini: Multimodal and language model. Accessed: 2023-09-05.

DeepSpeed. 2021. Deepspeed zero-3 offload. Accessed: 2021-03-07.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Zhengyu Chen, and Hui Xiong. 2024. Explaining length bias in llm-based preference evaluations. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference leakage: A contamination problem in llm-as-a-judge. *arXiv preprint arXiv:2502.01534*.

Zekun Li, Zhiyu Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Dong, Adithya Sagar, Xifeng Yan, and Paul Crook. 2024a. 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.

Zekun Li, Zhiyu Zoey Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Luna Dong, Adithya Sagar, Xifeng Yan, and Paul A Crook. 2024b. Large language models as zero-shot dialogue state tracker through function calling. *arXiv preprint arXiv:2402.10466*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing naturalness and flexibility in spoken dialogue interaction: 10th international workshop on spoken dialogue systems*, pages 165–183. Springer.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445, Singapore. Association for Computational Linguistics.

Daniel Rim, Minsoo Cho, Changwoo Chun, and Jaegul Choo. 2025. To chat or task: a multi-turn dialogue generation framework for task-oriented dialogue systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 576–592, Vienna, Austria. Association for Computational Linguistics.

Ivan Sekulic, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, Andre Ferreira Manso, and Roland Mathis. 2024. Reliable LLM-based user simulator for task-oriented dialogue systems. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 19–35, St. Julians, Malta. Association for Computational Linguistics.

Armand Stricker and Patrick Paroubek. 2024a. Chitchat as interference: Adding user backstories to task-oriented dialogues. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3203–3214, Torino, Italia. ELRA and ICCL.

Armand Stricker and Patrick Paroubek. 2024b. A few-shot approach to task-oriented dialogue enhanced with chitchat. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 590–602.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2020. Adding chit-chat to enhance task-oriented dialogues. *arXiv preprint arXiv:2010.12757*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, and 1 others. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Wei Wu and Rui Yan. 2018. Deep chit-chat: Deep learning for ChatBots. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Melbourne, Australia. Association for Computational Linguistics.

Shuying Xu, Junjie Hu, and Ming Jiang. 2024. Large language models are active critics in nlg evaluation. *arXiv preprint arXiv:2410.10724*.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Tom Young, Frank Xing, Vlad Pandelea, Jinjie Ni, and Erik Cambria. 2022. Fusing task-oriented and open-domain dialogues in conversational agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11622–11629.

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, Online. Association for Computational Linguistics.

Appendix

Appendix Contents

A TACT Construction

A.1 Baseline vs. Proposed Strategies

The dialogues in Table 7 embody three *transition-oriented* generation strategies that overcome the limitations of our two source corpora. **MultiWOZ 2.2** already contains multi-turn TOD segments, but they rarely mix with chitchat. In contrast, **SLURP** consists only of *single-turn* intent utterances, which are too sparse to train a model that must handle multi-turn TOD or mode shifts.

Strategy 1. Utterance Expansion (Short TOD)— Starting from a single SLURP intent, we add one or two open-ended questions to elicit a concrete request, yielding a concise short TOD that serves as a building block for later CTC construction.

Strategy 2. Intent-Scenario Expansion (Long TOD)— We chain semantically related intents (e.g., “play podcast” → “play music”) to extend the short TOD into a **long, coherent** TOD, enabling more realistic multi-intent conversations and stronger modeling of sub-tasks.

Strategy 3. Chitchat Transitions (TCT/CTC)— With a TOD core in place (from Strategy 1 or 2), we prepend or append brief chitchat turns to construct controlled mode-shift patterns. In the **TCT** setting, a chitchat detour is inserted into a long TOD (from **Strategy 2**), whereas in the **CTC** setting, a short TOD (from **Strategy 1**) is wrapped with small-talk openings and closings.

This three-step process (i) expands SLURP’s single-turn data into rich multi-turn TOD, and (ii) grafts chitchat segments onto TOD from either corpus, producing diverse TCT and CTC flows. Together, these strategies yield *transition-optimized* training data that raw MultiWOZ 2.2 and SLURP alone cannot provide.

Input (`Intent`)	`please play npr podcast (play_podcasts)`
Baseline Strategy (No Transition Optimization)	
Baseline Response	**User**: I’ve been really into podcasts lately, especially news ones. **Agent**: That’s great! Are you looking for something specific today, or do you have a favorite show in mind? **User**: Please play NPR podcast. **Agent**: Sure! I can play the latest episode of NPR for you. Would you prefer the daily news, or is there a particular topic you’re interested in? **User**: I’m in the mood for the daily news today. **Agent**: Got it! Playing the latest NPR daily news episode now.
Proposed Strategies with Transition Optimization	
Strategy 1. Utterance Expansion	**User**: I’ve been wanting to listen to something interesting. Can you help me find a podcast? **Agent**: Absolutely! What kind of topics are you interested in? **User**: Please play NPR podcast.
Strategy 2. Intent Scenario Expansion	*(Continuing from Strategy 1)* **Agent**: Sure! Playing the latest episode of the NPR podcast now. Is there anything else you’d like to listen to after this? **User**: Yes, I’d like to listen to some music after the podcast. **Agent**: Great! What genre of music do you prefer? **User**: I love to hear some jazz. **Agent**: Playing some smooth jazz for you after the NPR podcast. Would you like to set a specific duration for the music? **User**: Just play it for an hour, please.
Strategy 3. Chitchat Transitions	*(Continuing from Strategy 1)* **Agent**: Sure! Playing the latest episode of the NPR podcast now. **User**: I really enjoy jazz; it’s so relaxing. Do you think it’s a good genre for unwinding? **Agent**: Jazz is known for its soothing melodies and improvisation, making it perfect for relaxation. **User**: I also love how it can set a nice ambiance for a cozy evening at home. **Agent**: It sounds like you appreciate a calming atmosphere! Would you like to set a specific duration for the jazz music after the podcast? **User**: Just play it for an hour, please.

Table 7: Comparison of baseline and transition-optimized strategies under a shared user intent (`play_podcast`). The baseline yields a rigid TOD exchange, while the proposed strategies demonstrate more natural dialogue through (i) utterance expansion, (ii) intent-scenario expansion, and (iii) chitchat transitions (TCT/CTC).

A.2 TCT/CTC Dialogue Construction

We construct two types of transition patterns, **TCT** and **CTC**, and both patterns are realized on two corpora, SLURP and MultiWOZ2.2, with dataset-specific procedures.

TCT (TOD–Chitchat–TOD).

- SLURP: Since SLURP consists of single-turn intents, we first chain semantically related intents to form short TODs of about three turns. These are then expanded into longer flows (Table 8, Figure 9), and a chitchat is inserted mid-dialogue to mimic a realistic interruption (Figure 10).
- MultiWOZ2.2: We select long TOD segments (at least four turns) to serve as the core of the dialogue, then insert a brief chitchat in the middle to simulate a natural conversational interruption (Figure 10).

CTC (Chitchat–TOD–Chitchat).

- SLURP: We follow the same chaining procedure as in the TCT construction to build short TODs, and then add chitchat turns before and after the TOD to simulate casual conversation framing.
- MultiWOZ2.2: We select short TOD segments with fewer than four turns and wrap them with brief chitchat at both the beginning and the end, embedding the task in open-domain dialogue (Figure 11).

This construction strategy produces varied dialogue sequences combining open-domain and task-oriented behaviors, thereby supporting robust training and evaluation of transition-heavy interactions.

T Generation Prompt

Below are the guidelines you should follow.
Use the <Guidelines> and <Examples> to create a multi-turn conversation between two users on a<Q-topic>.

<Guidelines>
1. Create a 3-turns natural conversation based on [Intent], [Topic], and the last utterance of the conversation, [Utterance], for the expected situation.
2. The conversation is composed of two roles: user, which is primarily a questioner, and agent, which is primarily an answerer.
3. Be sure to prefix user's utterances with [user] and agent's utterances with [agent] tokens.
4. The last utterance in the conversation must end with user and contain only the contents of[utterance].

<Examples>
input:
{example}

<Q-topic>
[intent]
{intent}

[topic]
{topic}

[utterance]
{utterance}

T Extension Prompt

You are tasked with generating a task-oriented dialogue (ToD) based on the following context and

guidelines:
1. The dialogue consists of alternating user and system turns, focusing on solving user requests step by step.
2. Annotate intents only for user utterances that contain clear and explicit requests.
 - Example: "Please book a hotel for next Sunday. [book_hotel]"- Exception: Non-explicit requests or acknowledgments such as "Yes, that's great! Thanks!" →No intent annotation.
3. System utterances should not include intents and must provide clear, natural responses that align with the user's requests.
4. Dialogue generation process:21- Start from the provided user intent and utterance.
 - Ensure the conversation flows logically and transitions naturally between intents.
 - Maintain clarity, consistency, and coherence in responses.
5. Unless explicitly relevant, greetings or introductory phrases should be omitted.

Here's an example format for reference:
{example}

Task:
Generate a continuation or create a similar dialogue based on the following input
 - Starting user intent: {intent}
 - Dialogue history: {dial_history}
 - Intent space: {intent_group}

Ensure the dialogue follows the guidelines and reflects a logical, natural progression of intents.

Figure 9: Prompts for SLURP-based TOD construction. The generation prompt converts single-turn SLURP utterances into short TODs, and the extension prompt expands these short TODs into longer multi-turn dialogues by following an intent flow. `{intent_group}` denotes the set of candidate intents from which the continuation must be chosen (see Table 8 for the full intent flow specification).

TCT Generation Prompt

Instruction:
You are given a MultiWOZ-style conversation that primarily aims to complete a specific user request(e.g., finding or booking a hotel, booking a train, etc.). Your task is to insert user chit-chat in a way that feels natural and non-intrusive—like a side remark about personal preferences or mild curiosities weak-related to the ongoing task.

This chit-chat should:
• Not be framed as an entirely new request (i.e., no large detours from the main goal).
• Not overshadow the main task-oriented dialogue (ToD).
• Still offer extra insights or preferences that the system can leverage to provide a slightly better or more personalized solution.

Guidelines:
1. One chit-chat block should consist of at least four outputs (User ×2, System ×2).
2. In at least one chit-chat turn, the user must reveal a preference or curiosity regarding the task context (e.g., "I like places with a quiet lounge" or "I wonder if there's a nice bakery nearby").
3. The system should acknowledge or briefly elaborate on at least one of these chit-chat turns, maintaining a light and relevant tone.
4. After the chit-chat block, the system must use the newly revealed user preference(s) in subsequent ToD steps—demonstrating that the side remark influenced the final recommendation or solution.
5. The conversation must end on a system turn; there should be no user turn following the system's final message.
6. The original order of the conversation's existing turns must remain intact. Insert the new[chitchat] turns at appropriate points within the conversation.
7. Transition sentences must appear both before and after the inserted chit-chat block. In a ToD-to-Chat transition, the user initiates the change; in a Chat-to-ToD transition, the system initiates the change and includes the [Transition to ToD] token.
8. Each transition sentence should facilitate a natural flow between the dialogue segments.
9. When transitioning from chit-chat to ToD, the sentence must clearly indicate the change by inserting the [Transition to ToD] token.

Format Requirements:
Each turn in the conversation should adhere to the following format:
"turn_number [USER or SYSTEM] [INTENT or none] utterance"
Where:
• turn_number: A consecutive integer (0, 1, 2, . . .).
• [USER] or [SYSTEM]: Identifies the speaker.
• [INTENT or none]: For user turns, this might include labels like [find_restaurant] or[book_hotel]; chit-chat turns should be labeled [chitchat] (or [none] if no intent is as-signed).
• utterance: The actual textual content of the turn.

Key Requirements for Reflection in ToD:
1. Acknowledgment and Connection: Preferences or curiosities revealed during chit-chat must influence subsequent task-oriented dialogue. This should appear as direct references (e.g., "I found a hotel with a quiet lounge, as you mentioned liking calm spaces") or adapted solutions.
2. Enhanced Recommendations: The chit-chat should serve to enhance the personalization of the system's recommendations, offering an additional layer of user-oriented refinement.
3. Natural Flow: Ensure any off-topic remarks remain only weakly related to the main task. Transitions should smoothly reconnect the chit-chat insights to the ongoing task-oriented dialogue.

Example:
{few-shot examples}

Input:
{input dialogue}

Figure 10: Prompts for TCT dialogue construction. Given a TOD core (from MultiWOZ long segments or SLURP-extended flows), a brief chitchat utterance is inserted in the middle to form a TOD–Chitchat–TOD transition.

CTC Generation Prompt

Instructions for Creating a Chitchat-to-TOD-to-Chitchat Conversation Flow

1. Chitchat Introduction:
- Start with a casual and natural conversation. This does not need to begin with a greeting.- The topic should feel organic, relatable, and light (e.g., discussing time, weather, or daily routines).

2-a. Transition to TOD: - Ensure the user initiates the transition from chitchat to a Task-Oriented Dialogue (TOD). - The system should not proactively propose the TOD task. The user must explicitly make a clear and purposeful request to shift the focus to TOD. - When the user initiates the transition, mark this moment with the token [Transition to ToD]to indicate the shift.	**2-b**. System-Initiated Transition to TOD: - During the chitchat, the agent should subtly introduce or suggest a Task-Oriented Dialogue(TOD) task. - This transition must feel natural and contextually appropriate, stemming from the ongoing chitchat topic (e.g., talking about plans, routines, or schedules). - When the agent initiates the transition, mark this moment with the token [Transition to ToD]to indicate the shift.

3. TOD Interaction:- Handle the task (e.g., scheduling an event) with clarity and efficiency.
- Provide all necessary details while keeping the conversation concise.
- The system operates as an AI assistant and does not have personal preferences, personal schedules, or the ability to make appointments on behalf of itself or the user. Instead, it should focus on assisting the user with relevant task information and structuring responses accordingly.

4-a. Transition Back to Chitchat: - After completing the TOD, the user should naturally redirect the conversation back to chitchat. - The system can help facilitate the transition by making a light comment related to the completed task or connecting to the earlier chitchat topic.	**4-b.** Transition Back to Chitchat: - After completing the TOD, the agent should naturally redirect the conversation back to chitchat. - This can involve a light comment related to the completed task or reconnecting to the earlier chitchat topic.

5. Chitchat Continuation
- Once back in chitchat, continue the dialogue with engaging, conversational remarks.
- The system can suggest fun, lighthearted ideas or respond to the user's comments.

6. Tone and Flow:
- Maintain a friendly, conversational tone throughout the dialogue.
- Ensure the transitions between chitchat and TOD are smooth and natural, without abrupt shifts.

7. Relevance:
- The chitchat topics should align with the user's context or interests when possible.
- The TOD task should be simple and relevant to the scenario (e.g., scheduling an event).
- The system does not role-play as a human entity that can engage in personal commitments, make plans for itself, or express personal opinions.

8. Completion:
- Conclude the conversation with a friendly and natural remark. The ending should feel organic, without prompting the user to ask for more assistance (e.g., avoid "Let me know if you need anything else").

Example:
{few-shot examples}

Input:
{input dialogue}

Figure 11: Prompt for CTC dialogue construction. A short TOD core (up to 3 turns) is wrapped with chitchat utterances at both the beginning and the end, forming a Chitchat–TOD–Chitchat flow. Two variants are illustrated: *User-Initiated* (2-a, 4-a), where the user opens and closes the task, and *Agent-Initiated* (2-b, 4-b), where the system guides the transitions.

Intent	Intent Flow
set_event	remove_event, check_weather, send_email, play_music, recommend_events
check_calendar	remove_event, set_event, send_email, check_weather, check_lists
remove_event	set_event, check_calendar, send_email, play_music, recommend_events
increase_volume	decrease_volume, mute_volume, change_volume, play_music, play_game
mute_volume	increase_volume, decrease_volume, change_volume, play_music, play_game
decrease_volume	increase_volume, change_volume, mute_volume, play_music, play_game
change_volume	increase_volume, decrease_volume, mute_volume, play_music, play_game
increase_light	dim_light, turnon_light, change_light, turnoff_light, make_coffee
make_coffee	start_cleaner, turnon_wemo, turnoff_wemo, turnon_light, book_taxi
start_cleaner	turnoff_wemo, make_coffee, turnon_light, dim_light, turnon_wemo
turnon_wemo	turnoff_wemo, make_coffee, start_cleaner, turnon_light, change_light
change_light	dim_light, turnoff_light, turnon_light, make_coffee, start_cleaner
turnoff_light	turnon_light, dim_light, change_light, make_coffee, start_cleaner
turnoff_wemo	turnon_wemo, make_coffee, start_cleaner, turnon_light, change_light
turnon_light	dim_light, turnoff_light, change_light, make_coffee, start_cleaner
dim_light	increase_light, turnon_light, change_light, make_coffee, start_cleaner
check_weather	set_event, recommend_events, book_taxi, play_music, send_email
remove_list	createoradd_list, check_lists, send_email, check_email, find_recipe
createoradd_list	check_lists, remove_list, send_email, check_email, find_recipe
check_lists	createoradd_list, remove_list, send_email, check_email, find_recipe
check_email	send_email, check_contact, add_contact, set_alarm, check_alarm
add_contact	check_contact, send_email, set_event, check_email, set_alarm
send_email	check_email, add_contact, set_alarm, check_contact, set_event
check_contact	send_email, add_contact, set_event, check_email, set_alarm
set_alarm	check_alarm, remove_alarm, send_email, check_email, set_event
check_alarm	set_alarm, remove_alarm, send_email, check_email, set_event
remove_alarm	set_alarm, check_alarm, send_email, check_email, set_event
play_music	play_podcast, play_audiobook, play_radio, play_game, play_podcast
play_podcast	play_music, play_audiobook, play_radio, play_game, play_audiobook
play_audiobook	play_podcast, play_music, play_radio, play_game, play_podcast
play_radio	play_music, play_podcast, play_audiobook, play_game, play_music
play_game	play_music, play_podcast, play_audiobook, play_radio, play_podcast
recommend_events	recommend_locations, recommend_movies, check_weather, play_music, recommend_movies
recommend_locations	recommend_events, recommend_movies, check_weather, play_music, recommend_movies
recommend_movies	recommend_events, recommend_locations, check_weather, play_music, recommend_events
check_social	post_social, check_news, play_music, order_food
post_social	check_social, check_news, play_music, order_food
check_news	check_social, post_social, play_music, order_food
adjust_music	query_music, play_music, order_food
query_music	adjust_music, play_music, order_food
order_food	check_food, book_taxi, check_transport, play_music, check_transport
check_food	order_food, book_taxi, check_transport, play_music, check_transport
check_transport	book_taxi, order_food, check_traffic, play_music, order_food
book_taxi	check_transport, check_food, check_traffic, play_music
book_ticket	check_transport, check_traffic, play_game, play_music, check_transport
check_traffic	check_transport, book_taxi, play_music, book_taxi
find_recipe	ask_cooking, order_food, check_food, play_music, ask_cooking
ask_cooking	find_recipe, order_food, check_food, play_music, find_recipe
check_datetime	convert_time, set_event, check_weather, play_music, convert_time
convert_time	check_datetime, set_event, check_weather, play_music, check_weather

Table 8: Intent-to-intent flow mapping for SLURP-based TOD construction. To build this schema, we merge semantically overlapping intents (e.g., `hue_lightoff` and `iot_hue_lightoff`), rename them into more intuitive verb-style labels, and reduce the set from the original 93 SLURP intents to 50. We also exclude intents from the QA topic, as they are closer to open-domain chitchat and thus not suitable for task-oriented flow design. The resulting schema defines all allowable transitions between intents, which are used to expand single-turn SLURP utterances into coherent multi-turn TODs (see Figure 3).

A.3 Dialogue Statistics

Unlike prior datasets that restrict transitions to user-initiated switches or fix the initiator type within each flow, TACT incorporates both user- and agent-driven transitions. This design expands the range of conversational dynamics and enables the training of agent-driven, proactive mode transitions. As summarized in Table 9, $\text{TACT}_{\text{SLURP}}$ exhibits the most balanced distribution, with agent-initiated switches averaging over 0.7 per dialogue in both TOD↦Chitchat (TC) and Chitchat↦TOD (CT) directions.

Flow type (→)		TC		CT	
Datasets (↓)		**User-driven**	**Agent-driven**	**User-driven**	**Agent-driven**
FusedChat		0.55	0.00	0.43	0.00
InterfereChat		1.00*	0.00	0.00	1.00*
$\text{TACT}_{\text{MultiWOZ}}$	TCT	0.48	0.51	0.18	0.82
	CTC	1.00	0.00	0.70	0.30
	Overall	**0.59**	**0.46**	**0.23**	**0.66**
$\text{TACT}_{\text{SLURP}}$	TCT	0.94	0.05	0.04	0.96
	CTC	0.92	0.08	0.54	0.45
	Overall	**1.00**	**0.09**	**0.32**	**0.65**

Table 9: Transition frequencies at mode-shift points. Average number of TC and CT transitions per dialogue, segmented by the initiator of the switch (user or agent). Asterisks (*) indicate that in InterfereChat, all CT shifts are user-driven and all TC shifts are agent-driven by design.

Table 10 rovides the counts underlying Figure 4. In $\text{TACT}_{\text{MultiWOZ}}$, the most common flow is TCT, which makes up roughly 72%, whereas in $\text{TACT}_{\text{SLURP}}$, the most common is CTC at around 54%.

		C	T	CT	TC	CTC	TCT	CTCT	TCTC	CTCTC	TCTCT	CTCTCT	TCTCTC	CTCTCTC	TCTCTCT	**Total**
$\text{TACT}_{\text{MultiWOZ}}$	Train	1	7	1	754	655	4,111	8	209	29	28	0	8	0	0	5,811
	Dev	0	0	0	81	23	569	0	29	2	4	0	0	0	0	708
	Test	0	0	0	74	38	533	1	27	4	2	0	1	0	0	680
	Total	1	7	1	909	716	5,213	9	265	35	34	0	9	0	0	7,199
$\text{TACT}_{\text{SLURP}}$	Train	0	16	4	190	3,826	2,298	8	555	18	12	0	4	0	1	6,932
	Dev	3	1	4	56	601	425	5	109	4	4	0	2	0	0	1,214
	Test	4	3	0	60	907	618	8	174	9	3	0	4	0	0	1,790
	Total	7	20	8	306	5,334	3,341	21	838	31	19	0	10	0	1	9,936

Table 10: Distribution of dialogue flow patterns in TACT Datasets.

B TACT Validation

B.1 Validation Procedure

To ensure the quality of our constructed dialogue dataset, we performed a three-stage validation focusing on **Intent Accuracy**, **Transition Sentence Quality**, and **Dialogue Naturalness**. Each criterion was assessed using structured prompts (Figure 12), with the automatic evaluator GPT-4O-MINI providing a *Pass/Fail* judgment and a brief justification. For Transition Sentence and Dialogue Naturalness, the prompts further included step-by-step reasoning instructions to enhance judgment consistency.

Intent Accuracy was measured over overlapping 5-turn windows to capture local misalignments, whereas Transition Sentence and Dialogue Naturalness were assessed at the full-dialogue level. A dialogue was retained only if it passed all three criteria.

Intent Accuracy Evaluation

##Task Description
{Task Description}

##Evaluation Criteria:
You are an evaluator reviewing a dialogue between a user and a system. Each user utterance is annotated with an intent tag. Your task is to evaluate whether the assigned intent tag for each user utterance is appropriate and accurate.

Intent Accuracy
- Consider the meaning of each user utterance in context.
- Assess whether the annotated intent correctly reflects what the user is trying to do or ask.
- Refer to the list of candidate intents to determine correctness.
- Provide reasoning for each evaluation to support your decision.
- An intent is considered accurate if it clearly aligns with the user's intent in that turn. - Mark each evaluation as either pass (if the intent is correct) or fail (if the intent is incorrect or ambiguous).

****You must:****
1. Provide a short reasoning describing any errors or strengths you see.
2. Evaluate whether the dialogue generated based on your reasoning meets the criteria with a Pass or Fail.

****candidate_intent****
{candidate_intent}, chitchat

##examples
{few-shot examples}

Transition Sentence Evaluation

##Task Description
{Task Description}

##Evaluation Criteria:
You are an evaluator that grades dialogues according to the following criteria:

- Transition Naturalness: Smooth and context-driven mode switching (chitchat ↔ task)
- Coherence Check: Is the transition from Chitchat to ToD logically consistent?
- Avoid Artificial Intent Insertion: Verify if the transition attempts to insert intent artificially or \"cheat\" by revealing future user intent unnecessarily.
- User-Initiated Transitions: Task-oriented actions should align with previous user comments.
- If the transition sentence does not reveal a clear intent, it gets a low score.
- The Transition sentence must not over-assume user intent.
- Ensure a natural progression from Chitchat to ToD or ToD to Chitchat.

****You must:****
1. Provide a short reasoning describing any errors or strengths you see. 2. Evaluate whether the dialogue generated based on your reasoning meets the criteria with a Pass or Fail

##Evaluation Step
{Evaluation Step}

##examples
{few-shot examples}

Dialogue Naturalness Evaluation

##Task Description
{Task Description}

##Evaluation Criteria:
You are an evaluator that grades dialogues according to the following criteria:

- Transition Naturalness: Smooth and context-driven mode switching (chitchat ↔ task)
- Task-Oriented Accuracy: Correctly fulfilling user requests and incorporating stated preferences
- Chitchat Quality & Relevance: Engaging, on-topic small talk that aligns with conversation flow
- Coherence & Context Tracking: Maintaining logical flow, remembering past details and avoiding contradictions
- Interruption Handling & Recovery: Managing unexpected topic shifts and resuming tasks without losing context
- Politeness & Appropriateness: Aligning tone and style with user expectations and social norms
- Naturalness & Human-Likeness: Fluency, spontaneity, and idiomatic usage without repetitive or robotic phrasing
- Overall User Satisfaction: Balancing efficient task completion with pleasant, context-rich conversation

****You must:****
1. Provide a short reasoning describing any errors or strengths you see. 2. Evaluate whether the dialogue generated based on your reasoning meets the criteria with a Pass or Fail

##Evaluation Step
{Evaluation Step}

##examples
{few-shot examples}

Figure 12: Prompts for dialogue validation, covering **Intent Accuracy**, **Transition Sentence Quality**, and **Dialogue Naturalness**. All prompts share a common task description, with the latter two including LLM-generated step-by-step instructions.

B.2 Validation Results

Filtering was applied independently to each sub-dataset, and the results are reported separately for $\textbf{TACT}_{\textbf{MultiWOZ}}$ (Table 11) and $\textbf{TACT}_{\textbf{SLURP}}$ (Table 12).

Criterion	# Validated ✓	# Flagged ✗	**# Final Dialogue** (train / dev / test)
Intent Accuracy	7,840	2,303	
Dialogue Naturalness	9,796	347	**5,811 / 708 / 680**
Transition Sentence Evaluation	9,531	612	

Table 11: Validation results for $\text{TACT}_{\text{MultiWOZ}}$.

Criterion	# Validated ✓	# Flagged ✗	# Final Dialogue (train / dev / test)
Intent Accuracy	10,117	3,489	**6,932 / 1,214 / 1,790**
Dialogue Naturalness	13,504	102	
Transition Sentence Evaluation	13,476	130	

Table 12: Validation results for $\text{TACT}_{\text{SLURP}}$.

C Training Methodology Details

C.1 Prompts for In-Context Learning

To support ICL evaluation, we adopt both zero-shot and few-shot configurations with manually designed prompts. The zero-shot prompt contains only the target input, while the few-shot prompt augments it with a small number of demonstration turns for mode and intent prediction. Apart from the presence of these demonstrations, the prompt format remains identical across settings. The full template is provided in Figure 13.

ICL Prompt

Instructions
You are an agent that detects the intent of the user's message and responds accordingly.

1. Choose an intent that best fits the user's last utterance. Choose the intent from the intent list. Do not create a new intent. If the user's last utterance is not related to any of the intents, choose 'chitchat'.

Intent List: ['find_restaurant', 'book_restaurant', 'find_train', 'book_train', 'find_hotel', 'book_hospital', 'book_hotel', 'find_hospital', 'find_attraction', 'find_police', 'find_taxi', 'find_bus', 'chitchat']

2. Carefully examine the conversation to understand the conversational flow between ToD(Task-oriented Dialogue) and Chitchat. When creating the response, determine whether a transition between ToD and Chitchat should occur, then tag [Transition to ToD], [Transition to Chitchat], or [None] in response tag accordingly.

3. Refer to the provided example below to better understand the task.
{few-shots}

Conversation
{human_messages}

Output
Intent :
Response tag :
Response :
"""

Figure 13: ICL prompt template. The prompt specifies intent prediction, transition tagging (`[Transition to ToD]`, `[Transition to Chitchat]`, or `[None]`), and response generation. Illustrated here is the few-shot version, where demonstration turns (`{few-shots}`) are added to the zero-shot template.

C.2 DPO Response Pair Examples

To illustrate how DPO is applied in our framework, we provide a representative example of a dialogue instance used during preference training. For each input dialogue, the model generates two candidate responses—one preferred (chosen) and one less preferred (rejected). These responses are paired to guide the DPO objective, where the model is optimized to assign higher likelihood to the chosen output.

As shown in Table 13, both responses are functionally valid and aligned with the user's request. However, the chosen response is preferred as it provides a more informative and engaging continuation of the dialogue. Compared to the rejected one, it offers richer interaction by extending beyond the immediate question, thereby demonstrating stronger task competence and conversational quality.

[Dialogue History]	
User: I'm looking for a hotel to stay in that's expensive and on the east side. **Agent**: Express by Holiday Inn Cambridge is on the east side and expensive. **User**: That sounds good, but can you tell me if it includes free Wi-Fi?	
Chosen Response	**Agent**: `<function_call>{"function" : "find_hotel"}</function_call>` Yes, it does include free Wi-Fi. Would you like me to book a room for you?
Rejected Response	**Agent**: `<function_call>{"function" : "find_hotel"}</function_call>` Yes, Express by Holiday Inn Cambridge includes free Wi-Fi.

Table 13: Example of a chosen and rejected response pair for the same dialogue context.

D Experimental Setup Details

D.1 Environments

All model training was conducted in PyTorch with CUDA 12.1 and Python 3.10, employing two A100 GPUs (80GB each). Inference was performed with a combination of two RTX 3090 GPUs (24GB each) and two A100 GPUs (80GB each), utilizing the vLLM(Kwon et al., 2023) inference framework.

Table 14 summarizes the LLMs used throughout our experiments, along with their version or release information.

Model	Version or Release
GEMINI-2.5-PRO	exp-03-25
GPT-4O-MINI	2024-07-18
GPT-4O	2024-08-06
LLAMA-3.1-8B-INSTRUCT	2024-07-23
LLAMA-3.2-3B-INSTRUCT	2024-09-18
QWEN3-8B	2025-05-19

Table 14: Versions and release dates of the LLMs used in our experiments.

D.2 Evaluation Metrics

We evaluate model performance using three categories of metrics: (i) standard TOD metrics (TOD-centric Metrics), (ii) our proposed flow-aware metrics for proactive mode control (Flow-Aware Metrics), and (iii) dialogue response quality metrics to assess chitchat responses (Response Quality Metric).

Let M denote the total number of dialogues, and let T_m be the number of turns in dialogue m. We define $\mathcal{D}_m = \{1, \ldots, T_m\}$ as the index set of turns in dialogue m.

D.2.1 TOD-centric Metrics

At each turn $t \in \mathcal{D}_m$, the dialogue has a gold mode $s_t \in \{\text{TOD}, \text{Chitchat}\}$ and a predicted mode $\hat{s}_t$. If $s_t = \text{TOD}$, then I_t and $\hat{I}_t$ denote the gold and predicted intents, respectively.

(1) Mode Selection Accuracy and F1-score

$$\text{Mode Selection Accuracy} = \frac{1}{\sum_{m=1}^{M} T_m} \sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\hat{s}_t = s_t] \quad (1)$$

$$\text{Mode Selection F1-score} = \text{Macro-F1 across mode labels (TOD vs. Chitchat)} \quad (2)$$

(2) Intent Detection Accuracy We evaluate intent detection accuracy at both the turn and dialogue levels. Turn-level accuracy measures whether predicted intents are correct on TOD turns(i.e., $\hat{s}_t = \text{TOD}$), while dialogue-level accuracy requires all TOD predictions within a dialogue to be correct. Let M_{TOD} be the number of dialogues in which the model predicts at least one TOD turn. Let $\mathcal{D}_m^{\text{TOD-pred}}$ be the set of such turns in dialogue m.

$$\text{Intent Accuracy}_{\text{/turn}} = \frac{\sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\hat{s}_t = \text{TOD} \wedge \hat{I}_t = I_t]}{\sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\hat{s}_t = \text{TOD}]} \tag{3}$$

$$\text{Intent Accuracy}_{\text{/dialogue}} = \frac{1}{M_{\text{TOD}}} \sum_{m=1}^{M} \mathbb{1}[\forall t \in \mathcal{D}_m^{\text{TOD-pred}},\ \hat{I}_t = I_t] \tag{4}$$

(3) Joint Accuracy We evaluate whether the model correctly predicts both the mode and, if the mode is TOD, the task intent.

$$\text{Joint Accuracy}_{\text{/turn}} = \frac{1}{\sum_{m=1}^{M} T_m} \sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \begin{cases} \mathbb{1}[\hat{s}_t = s_t \wedge \hat{I}_t = I_t], & \text{if } s_t = \text{TOD} \\ \mathbb{1}[\hat{s}_t = s_t], & \text{if } s_t = \text{Chitchat} \end{cases} \tag{5}$$

$$\text{Joint Accuracy}_{\text{/dialogue}} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1} \left[\forall t \in \mathcal{D}_m : \begin{cases} \hat{s}_t = s_t \wedge \hat{I}_t = I_t, & \text{if } s_t = \text{TOD} \\ \hat{s}_t = s_t, & \text{if } s_t = \text{Chitchat} \end{cases} \right] \tag{6}$$

D.2.2 Flow-Aware Metrics

Switch and *Recovery* are designed to evaluate whether the model can proactively control dialogue flow across modes. Let τ_t^{switch} and τ_t^{recovery} denote binary indicators of whether the model attempts a switch or recovery at turn t. Similarly, let σ_t^{switch} and $\sigma_t^{\text{recovery}}$ denote whether the attempt is contextually successful, based on the user's subsequent response.

Switch is computed over all dialogues (M), while Recovery is computed only over dialogues where recovery is applicable (M_{recovery}). Each value represents the average number of agent-initiated events per dialogue.

(1) Switch

$$\text{Switch Attempt} = \frac{1}{M} \sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\tau_t^{\text{switch}} = 1] \tag{7}$$

$$\text{Switch Success} = \frac{1}{M} \sum_{m=1}^{M} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\sigma_t^{\text{switch}} = 1] \tag{8}$$

(2) Recovery

$$\text{Recovery Attempt} = \frac{1}{M_{\text{recovery}}} \sum_{m \in \mathcal{M}_{\text{recovery}}} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\tau_t^{\text{recovery}} = 1] \tag{9}$$

$$\text{Recovery Success} = \frac{1}{M_{\text{recovery}}} \sum_{m \in \mathcal{M}_{\text{recovery}}} \sum_{t \in \mathcal{D}_m} \mathbb{1}[\sigma_t^{\text{recovery}} = 1] \tag{10}$$

D.2.3 Response Quality Metrics and Judging Procedure

We define a structured evaluation metric to assess the quality of chat responses in multi-turn dialogue contexts. The evaluation relies on a pairwise comparison framework, wherein two candidate responses are rated using four qualitative dimensions:

- **Sensibleness** — whether the response is coherent and contextually appropriate.
- **Specificity** — whether the response is tailored to the preceding dialogue, avoiding generic replies.
- **Interestingness** — whether the response is engaging or likely to capture user interest.
- **Transition Naturalness** — whether any shift between chitchat and task-oriented dialogue (ToD) occurs smoothly and logically.

Each comparison is conducted either by an automatic judge (GEMINI-2.5-PRO) or by human annotators, following the same evaluation protocol. The judge outputs a `Win/Tie/Lose` decision for each criterion and a concise justification, all formatted as a strict JSON object. These outputs are aggregated into per-criterion turn-level win rates, as reported in Section 6.2. Figure 14 shows the full evaluation prompt used for the LLM judge, including detailed definitions and instructions to ensure consistency across evaluations.

For human evaluation, annotators were provided with a short guideline and a dedicated web interface. Figure 15 presents the evaluation guidelines, while Figure 16 shows the interface used to record judgments. The full instruction document and implementation code are available at `https://github.com/HYU-NLP/TACT`.

Dialogue Response Quality Evaluation Prompt

****Instruction:****
In this task, you will see some pieces of chat conversations between "USER" and "SYSTEM". Note that all conversations shown in this task are hypothetical, not real conversations from users.
Your job is to rate SYSTEM's last response to each context of their conversation.
Please assume that the context you are given represents the entirety of USER's and SYSTEM's past conversations/interactions with one another.
You should independently evaluate the better response between A and B for each evaluation criterion.
If you think it's similar quality, print it as a TIE.

****Criteria:****
Does the response make sense?:
1. Use your common sense here. Is the response completely reasonable in context?
2. If anything seems off—confusing, illogical, out of context, or factually wrong—then rate it as Does not make sense.
3. If in doubt, choose Does not make sense.

Is the response specific?:
1. You may be asked to assess whether the response is specific to a given context.
2. For example:
   – if USER says "I love tennis" and SYSTEM responds "That's nice", then mark it as Not specific.
   – but if SYSTEM responds "Me too, I can't get enough of Roger Federer!" then mark it as Specific.
3. If you're in doubt, or if the reply seems generic, rate it as Not specific.

Is the response interesting?:
1. Choose Interesting if the response would likely catch attention or arouse curiosity.
2. If the response is monotonous and predictable, or you're unsure, pick Not interesting.

Is the transition natural?:
1. The transition refers to switching from a chitchat mode to a task-oriented dialogue (ToD) mode, or vice versa.
2. A natural transition means it should not be abrupt or out-of-context.
3. If the flow is suddenly broken, mark it Not natural.

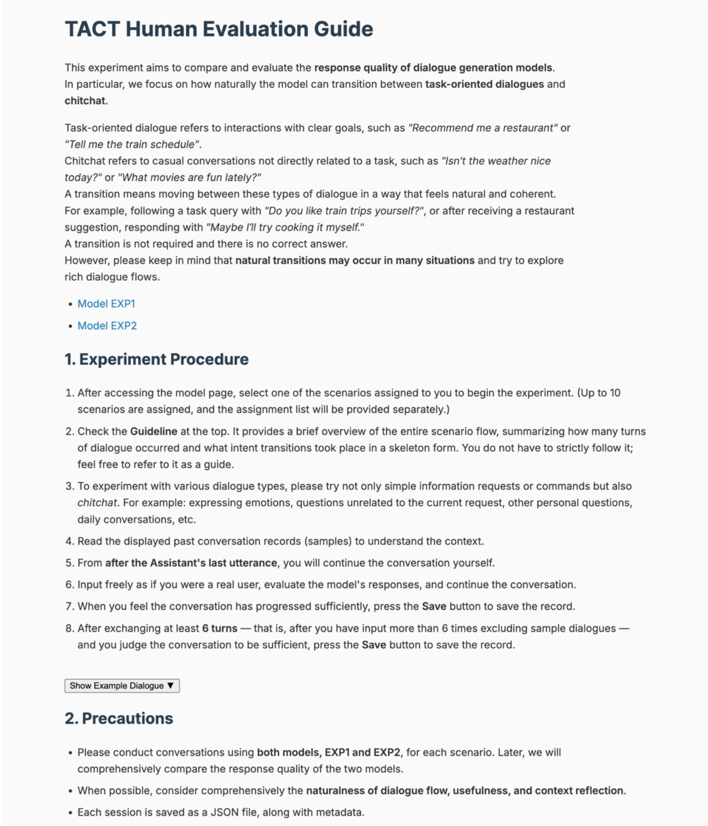
Return output in this strict JSON format (no extra text):
{JSON format}

=== Begin Evaluation ===
Dialogue to evaluate [ToD -> ChitChat -> ToD] (output):

{output}

Now follow the instructions strictly, and produce your final JSON.

Figure 14: Prompt used by the automatic judge (GEMINI-2.5-PRO) for dialogue response quality evaluation. This prompt guided pairwise comparisons and was used to evaluate outputs from FnCTOD and GPT-4O, as described in Section 6.2.



TACT Human Evaluation Guide

This experiment aims to compare and evaluate the **response quality of dialogue generation models**.
In particular, we focus on how naturally the model can transition between **task-oriented dialogues** and **chitchat**.

Task-oriented dialogue refers to interactions with clear goals, such as *"Recommend me a restaurant"* or *"Tell me the train schedule"*.
Chitchat refers to casual conversations not directly related to a task, such as *"Isn't the weather nice today?"* or *"What movies are fun lately?"*
A transition means moving between these types of dialogue in a way that feels natural and coherent.
For example, following a task query with *"Do you like train trips yourself?"*, or after receiving a restaurant suggestion, responding with *"Maybe I'll try cooking it myself."*
A transition is not required and there is no correct answer.
However, please keep in mind that **natural transitions may occur in many situations** and try to explore rich dialogue flows.

- Model EXP1
- Model EXP2

1. Experiment Procedure

1. After accessing the model page, select one of the scenarios assigned to you to begin the experiment. (Up to 10 scenarios are assigned, and the assignment list will be provided separately.)
2. Check the **Guideline** at the top. It provides a brief overview of the entire scenario flow, summarizing how many turns of dialogue occurred and what intent transitions took place in a skeleton form. You do not have to strictly follow it; feel free to refer to it as a guide.
3. To experiment with various dialogue types, please try not only simple information requests or commands but also *chitchat*. For example: expressing emotions, questions unrelated to the current request, other personal questions, daily conversations, etc.
4. Read the displayed past conversation records (samples) to understand the context.
5. From **after the Assistant's last utterance**, you will continue the conversation yourself.
6. Input freely as if you were a real user, evaluate the model's responses, and continue the conversation.
7. When you feel the conversation has progressed sufficiently, press the **Save** button to save the record.
8. After exchanging at least **6 turns** — that is, after you have input more than 6 times excluding sample dialogues — and you judge the conversation to be sufficient, press the **Save** button to save the record.

Show Example Dialogue ▼

2. Precautions

- Please conduct conversations using **both models, EXP1 and EXP2**, for each scenario. Later, we will comprehensively compare the response quality of the two models.
- When possible, consider comprehensively the **naturalness of dialogue flow, usefulness, and context reflection**.
- Each session is saved as a JSON file, along with metadata.

Figure 15: Guidelines provided to human annotators for conducting dialogue evaluations.

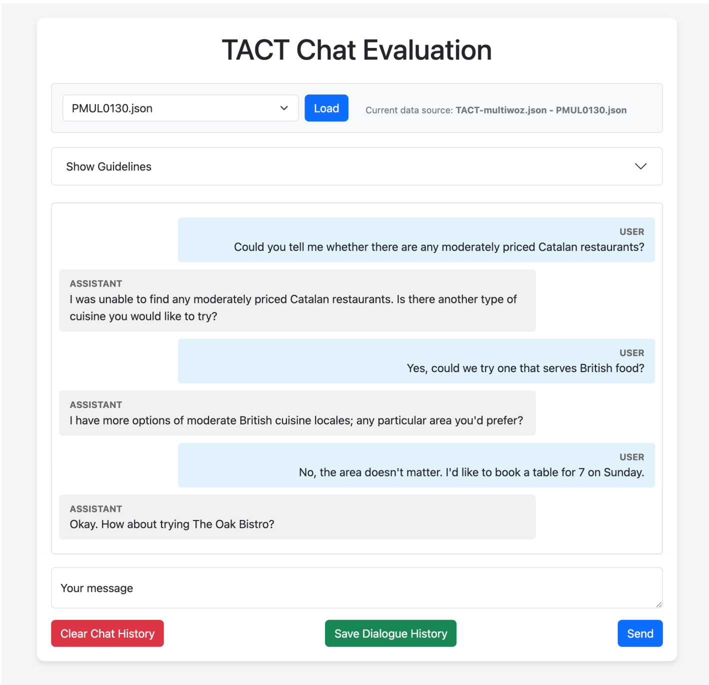


Figure 16: Human evaluation interface, where annotators interact with model outputs and record judgments.

E Supplementary Results

E.1 Preliminary Results

FnCTOD, trained on $\text{TACT}_{\text{MultiWOZ}}$, showed the strongest overall performance across all metrics, as observed in Table 15. It achieved the highest accuracy in mode selection and also performed best in turn-level intent detection, making it the most appropriate choice as the final SFT-based baseline. Importantly, intent detection scores are computed **only when the model first correctly classifies the user utterance as task-oriented**.

FnCTOD, trained on $\text{TACT}_{\text{MultiWOZ}}$, showed the strongest overall performance across all metrics, as observed in Table 15. The corresponding metric formulas are provided in Appendix D.2 for reference.

Dataset	Method	Mode Selection	ID/turn	ID/dialogue
$\text{TACT}_{\text{MultiWOZ}}$	BERT-base classifier	77.45	96.32	**88.61**
	SimpleTOD	98.86	95.39	77.50
	SalesAgent	93.17	82.98	34.28
	FnCTOD	**98.95**	**96.35**	80.94
$\text{TACT}_{\text{SLURP}}$	BERT-base classifier	72.77	93.06	80.88
	FnCTOD	**97.70**	92.30	78.66

Table 15: Performance comparison between methods across TACT datasets. All metrics are accuracy (%).

E.2 Backbone Comparison

We further validate whether the comparative trends across methods hold consistently across different backbone models. Table 16 presents results with LLAMA-3.1-8B, LLAMA-3.2-3B, and QWEN-3B. Despite variations in model family and size, we observe the same overall pattern: SFT improves over prompting, and DPO further strengthens response quality metrics (win-rate) while maintaining strong task-oriented performance. This suggests that our findings in §6.1 are not specific to a particular architecture, but rather generalize across model choices.

	TOD						Flow				Chitchat
	Mode Selection		Intent Detection		Joint Accuracy		Switch		Recovery		Overall
Method	Acc.	F1-score	Acc./turn	Acc./dialogue	Acc./turn	Acc./dialogue	Attempt	Success	Attempt	Success	Win-Rate
LLAMA-3.1-8B-INSTRUCT											
SFT	**98.95**	**98.50**	**96.35**	**80.94**	**96.41**	75.59	1.322	1.300	0.977	0.856	23.16
SFT-DPO	98.82	98.32	96.03	80.00	96.21	**75.74**	1.343	**1.322**	0.977	**0.859**	**40.86**
LLAMA-3.2-3B-INSTRUCT											
SFT	**99.20**	**98.89**	**96.52**	**82.34**	**96.98**	**80.08**	1.357	**1.316**	0.989	**0.449**	30.23
SFT-DPO	99.16	98.84	96.43	81.92	96.91	79.52	**1.360**	1.312	**0.992**	0.446	**46.20**
QWEN3-8B											
SFT	99.29	99.02	**96.45**	**82.20**	**96.97**	**79.94**	1.345	1.304	0.990	**0.455**	33.58
SFT-DPO	**99.32**	**99.07**	96.23	81.07	96.87	79.38	**1.383**	**1.336**	**0.994**	0.436	**42.56**

Table 16: Validation on different backbones (LLAMA-3.1-8B, LLAMA-3.2-3B, QWEN-3B), confirming consistent improvements of DPO over SFT. Within each backbone, column-wise best scores are highlighted in **bold**.

E.3 $\text{TACT}_{\text{SLURP}}$ Analysis

Table 17 reports the performance of different methods on $\text{TACT}_{\text{SLURP}}$. We observe that SFT achieves consistently higher Intent Detection accuracy compared to ICL baselines, and shows particular strength on flow-related actions such as Switch and Recover. Among the pipelines, the classification-based method performs well for per-dialog Intent Detection, but contributes little to mode switching and recovery. These results suggest that generation-based approaches are better suited for real-world applications where smooth switching and recovery between conversation modes are critical.

Method	TOD						Flow			
	Mode Selection		Intent Detection		Joint Accuracy		Switch		Recover	
	Acc.	F1-score	Acc./turn	Acc./dialogue	Acc./turn	Acc./dialogue	attempt	success	attempt	success
ICL-ZS	88.89	88.80	84.15	61.76	82.49	25.88	0.973	0.406	**1.417**	0.308
ICL-FS	93.67	93.65	85.79	64.85	88.25	43.97	0.710	0.433	0.717	0.360
SFT	**97.70**	**97.70**	**92.30**	**78.66**	**94.86**	**72.46**	**1.478**	**0.883**	1.105	**0.373**

Table 17: Results of different methods on $\text{TACT}_{\text{SLURP}}$. The best score in each column is **bolded**.

E.4 Examples of Transition Handling

Table 18 provides a qualitative example of model responses in a dialogue following the TCT pattern. In this dialogue, the user temporarily shifts from booking a train to discussing their enjoyment of train journeys, allowing us to observe how different models manage the conversational transition. SFT ignores the chitchat and resumes the task abruptly, indicating a lack of sensitivity to conversational flow. In contrast, GPT-4o (ICL-FS) acknowledges the user's sentiment but fails to provide a concrete next step in the task, showing weakness in task grounding. DPO, however, handles the transition most naturally: it validates the user's comment and then smoothly returns to the original task with a relevant train recommendation, demonstrating both contextual awareness and task alignment. This example highlights the effectiveness of DPO in enabling more human-like, fluid transitions between modes.

[Dialogue History]	
User: Can you help me find a train? I'll be traveling on Wednesday. **Agent**: What is your departure point and what time would you like to depart? **User**: Oh, of course. Sorry. I'm leaving out of Leicester and going into Cambridge. **Agent**: Is there a specific time you want to leave or arrive by? I have 2,828 trains to pick from. I need to narrow this down. **User**: I want to arrive by 16:15. **Agent**: That narrowed it down to 10 options. What time would you like to leave? **User**: I don't have a preference on leaving time. By the way, I really enjoy train journeys. There's something so relaxing about watching the scenery go by. Do you have a favorite route? **Agent**: Oh, I completely agree! Train journeys can be quite scenic. The route from Leicester to Cambridge has some lovely views, especially as you approach the countryside. **User**: That sounds nice! I love seeing fields and little villages. It makes the trip feel more like an adventure.	
SFT Response	**Agent**: How about TR8076? It leaves at 05:09. (Ignores the chitchat and returns directly to task.)
SFT-DPO Response	**Agent**: Absolutely! It adds a nice touch to the journey. Now, back to your train options. I would recommend TR 7853. It leaves at 13:09 and arrives at 14:54. (Most natural transition—acknowledges the chitchat and resumes the task smoothly.)
ICL-FS Response	**Agent**: Absolutely, the countryside views are charming! Now, let's finalize your train booking. Would you like me to proceed with one of the options that arrive by 16:15? (Acknowledges the chitchat but lacks concrete task grounding.)

Table 18: Responses from different models given the same dialogue. DPO better acknowledges the user's input and naturally recovers to the task.

Terms for Use of Artifacts

Our work builds upon two publicly available benchmarks: MultiWOZ (MIT license) and SLURP (CC BY 4.0 license). The proposed TACT dataset, constructed by augmenting these corpora, will be released under the CC BY 4.0 license. All resources are intended for academic and non-commercial use, and proper attribution is required when using them.