Zero-Shot Cross-Domain Dialogue State Tracking with small LLMs: Learning to Think through Reinforcement Learning

Anonymous ACL submission

Abstract

Dialogue State Tracking (DST) is essential for task-oriented dialogue systems to track user goals, but zero-shot adaptation to unseen domains remains challenging. This paper proposes an approach that enhances small LLMs for zero-shot cross-domain DST through reinforcement learning (RL) with verifiable rewards. We introduce two techniques: a Dynamic Difficulty Sampling strategy, which adaptively selects training examples matched to the model's evolving ability, and a Difficulty-Weighted Fuzzy Match Reward Function, which provides granular feedback to address sparse rewards and prioritize difficult slots. Building on a debiased Group Relative Policy Optimization (GRPO) variant (Liu et al., 2025), our method strengthens the reasoning capabilities of small LLMs and enables robust generalization to unseen domains without further training. Experiments on MultiWOZ 2.1 and 2.4 show that our approach achieves state-of-the-art performance among small models and rivals larger ones, while remaining computationally efficient. In a more challenging cross-dataset setting, our 3B model trained only on MultiWOZ transfers zero-shot to a Schema-Guided Dialogue (SGD) subset and outperforms a 7B baseline by 5.4 points average JGA, confirming that the improvements generalize beyond a single ontology. This work demonstrates the effectiveness of RL-based post-training for compact LLMs, paving the way for scalable, resource-efficient dialogue systems. Our code and models are available at https://anonymous.4open.science/r/DSTRL-6747.

1 Introduction

Dialogue State Tracking (DST) maintains a structured representation of user goals, typically as slot-value pairs, to support effective dialogue management (Budzianowski et al., 2018). Although general-purpose instruction-tuned LLMs can hold fluent task-oriented conversations, explicit DST remains essential for production systems: it produces auditable, machine-readable state that can be passed to downstream APIs and databases, enforced against business logic, and integrated with function-calling interfaces (Li et al., 2024). It is also a prerequisite for deploying smaller, efficient, latency-sensitive systems (e.g., voice assistants on edge devices), where running a large general LLM is impractical. This paper therefore investigates how reinforcement learning (RL) with verifiable rewards (Shao et al., 2024) can enhance *small* LLMs for zero-shot cross-domain DST (Williams et al., 2016), enabling them to handle unseen domains without additional training while retaining the efficiency required for real-world deployment.

Traditional DST research focused on specialized architectures designed to capture dialogue context (Wu et al., 2019a; Heck et al., 2020; Hosseini-Asl et al., 2020; Jacqmin et al., 2022). Recent efforts have shifted toward leveraging LLMs with techniques such as synthetic data generation and knowledge distillation to obtain smaller, more efficient models (Dong et al., 2024a; Finch and Choi, 2024; Hu et al., 2022; Wu et al., 2023). Supervised fine-tuning (SFT) has been the dominant paradigm, aligning predicted dialogue states with ground-truth annotations. While SFT has driven steady progress, recent work shows that RL with test-time scaling can substantially improve complex reasoning in LLMs (OpenAI, 2024; Muennighoff et al., 2025; Snell et al., 2024; Shao et al., 2024), and that small LLMs benefit similarly in domains such as mathematics and coding when optimized with RL (Zeng et al., 2025; Pan et al., 2025). Despite these advances, the use of RL to tackle DST's unique challenges—particularly for small LLMs—remains largely unexplored.

Applying verifiable RL to DST is not a plug-and-play extension of recent RL-for-reasoning methods. DST differs from math and code tasks in two ways. First, DST datasets such as MultiWOZ (Eric

et al., 2020) have a highly imbalanced *difficulty* distribution: most dialogue turns are easy, while a smaller fraction require resolving multi-domain interactions, ambiguous user goals, or long-range slot dependencies. Uniform RL sampling therefore wastes many updates on trivial examples while providing little useful signal on very hard ones. Second, the standard DST evaluation metric, Joint Goal Accuracy (JGA) (Budzianowski et al., 2018), is an exact-match objective over all slot-value pairs, so it gives no partial credit for nearly correct predictions. This sparse reward signal is especially challenging for small LLMs.

These challenges suggest that effective RL for DST requires both adaptive data selection and a reward function that exposes finer-grained learning signals. We therefore propose a three-stage post-training framework for small LLMs: CoT distillation from a larger teacher, difficulty estimation over training turns, and RL optimization with Dr. GRPO (Liu et al., 2025). Within this framework, Dynamic Difficulty Sampling selects examples matched to the model's evolving ability, while a Difficulty-Weighted Fuzzy Match Reward provides slot-level partial credit and emphasizes difficult slots.

Our contributions are threefold:

1. We identify two task-specific obstacles for applying verifiable RL to DST: imbalanced difficulty and sparse exact-match rewards.

2. We introduce a difficulty-aware RL framework that combines dynamic sampling with a difficulty-weighted fuzzy reward for zero-shot cross-domain DST.

3. We show that small LLMs trained with our method achieve state-of-the-art results among compact models on MultiWOZ 2.1/2.4 and transfer zero-shot to SGD, where our 3B model outperforms a 7B baseline by 5.4 points average JGA.

2 Preliminary

2.1 Dialogue State Tracking (DST)

DST maintains a structured representation of user goals in task-oriented dialogue systems, typically as slot-value pairs (e.g., restaurant name, cuisine type). At turn t the dialogue state is $b_t = \{(s_1, v_1), (s_2, v_2), \ldots, (s_k, v_k)\}$, where s_i is a slot and v_i is its value. The dialogue history is $h_t = [u_1, r_1, u_2, r_2, \ldots, u_t, r_t]$, with u_i the user's utterance and r_i the system's response at turn i. The DST model, parameterized by θ, predicts the state:

$$\hat{b}_t = f_\theta(h_t).$$

For LLMs with test-time scaling capabilities, the output additionally contains intermediate reasoning:

$$o_t = \{\textit{think}_t, \hat{b}_t\} = f_\theta(h_t),$$

where o_t is the complete output and $\textit{think}_t$ the reasoning content.

2.2 Group Relative Policy Optimization (GRPO)

GRPO (Shao et al., 2024) is a reinforcement learning algorithm that enhances the reasoning capabilities of LLMs by evaluating groups of generated responses relative to one another. Its efficiency, demonstrated in models such as DeepSeek-R1, stems from eliminating the need for a separate value model, unlike Proximal Policy Optimization (PPO) (Schulman et al., 2017).

In the context of DST, each dialogue turn is treated as a question–answer pair (h_t, b_t), where h_t is the dialogue history and b_t is the ground-truth dialogue state. The policy $\pi_{\theta_{\text{old}}}$ generates a group of G candidate responses $\{o_i\}_{i=1}^{G}$, each containing a predicted state $\hat{b}_i$ and intermediate reasoning $\textit{think}_i$. A reward function evaluates each predicted state, yielding rewards $\{R_i = \text{reward}(\hat{b}_i, b_t)\}_{i=1}^{G}$. The advantage for each response is computed relative to the group mean:

$$\hat{A}_{i,t} = R_i - \text{mean}\big(\{R_j\}_{j=1}^{G}\big).$$

Recent work (Liu et al., 2025) has identified two systematic biases in standard GRPO—a response-level length bias that favors longer outputs, and a question-level difficulty bias that distorts the advantage estimate across tasks of varying difficulty—and proposes Dr. GRPO, an unbiased variant that also shows the KL-divergence term is unnecessary for verifiable-reward RL. We adopt Dr. GRPO as our base optimizer because, in our preliminary experiments, both biases were pronounced for small LLMs ($<$ 3B) on DST and the KL term further restricted exploration. The policy is updated by maximizing the following objective:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} L_{i,t}(\theta)\right],$$

where the expectation is over $(h, b) \sim \mathcal{D}$ and $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|h)$, and $L_{i,t}(\theta)$ is:

$$L_{i,t}(\theta) = \min\big(r_{i,t}(\theta)\hat{A}_{i,t}, \\ \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\big),$$

and

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|h, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|h, o_{i,<t})}.$$

While DST and Dr. GRPO together provide a foundational framework, applying RL to DST raises additional, task-specific challenges—imbalanced difficulty distributions and sparse rewards—that the optimizer alone does not address. In the following section, we tackle these challenges with a tailored RL pipeline and novel sampling and reward strategies.

3 Proposed Method

3.1 A Novel Framework for Cross-domain DST Post-training

To address difficulty imbalance and improve generalization in zero-shot cross-domain DST, we propose a three-stage reinforcement learning (RL) pipeline to optimize small LLMs (<3B parameters), as illustrated in Figure 1.

Stage 1: Chain-of-Thought (CoT) Generation and Distillation. We first use a large LLM to generate high-quality Chain-of-Thought (CoT) reasoning traces for a single source domain. These traces are then distilled into the small LLM via Supervised Fine-Tuning (SFT) to establish a reasoning foundation that small models struggle to develop from instructions alone.

Stage 2: Difficulty Evaluation and Data Preparation. We next quantify the difficulty of the multi-domain training data to guide RL optimization. Using k-fold evaluation with the SFT model from Stage 1, we obtain a robust per-turn difficulty score defined as the Average Goal Accuracy (AGA) (Rastogi et al., 2020), i.e., the fraction of correctly predicted slots. These annotations feed both the Dynamic Difficulty Sampling strategy (Section 3.2) and the Weighted Fuzzy Match Reward Function (Section 3.3), addressing the imbalanced difficulty distribution typical of datasets such as MultiWOZ.

Stage 3: Reinforcement Learning with Dr. GRPO. Finally, we apply Dr. GRPO (Section 2.2) for RL across all source domains. Building on the difficulty annotations from Stage 2 and overcoming the single-domain coverage of Stage 1, this phase encourages the small LLM to learn generalized patterns and substantially improves its zero-shot performance on held-out domains.

3.2 Dynamic Difficulty Sampling

In rule-based verifiable RL methods such as GRPO, optimization relies on group-level advantages derived from multiple generated responses. When the training distribution is heavily skewed toward easy examples, uniform sampling produces weak optimization signals from trivially solved turns and yields little gradient information from intractably hard ones. As shown in Figure 2, the difficulty distribution of MultiWOZ—measured by per-turn AGA via k-fold evaluation—exhibits exactly this imbalance.

While static filtering of mid-difficulty samples can initially enhance optimization, it introduces bias as the model's proficiency evolves: a fixed filter cannot adapt to the model's changing capabilities. The proposed **Dynamic Difficulty Sampling** strategy instead tracks the model's ability by drawing samples from a Gaussian distribution centered on a target difficulty μ with spread σ:

$$p_i = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_i - \mu)^2}{2\sigma^2}\right),$$

where d_i is the difficulty of sample i. We update μ and σ periodically based on the average reward, shifting focus to harder or easier samples as performance changes. To ensure multi-domain balance, we apply stratified sampling that maintains proportional domain representation. This dynamic, balanced procedure addresses the imbalanced difficulty inherent in DST datasets.

3.3 Weighted Fuzzy Match Reward Function

In rule-based RL, the reward function guides the model toward desired behavior. For DST, a conventional reward relies on an exact match between the predicted state $\hat{b}_t$ and the ground-truth state b_t:

$$R(\hat{b}_t, b_t) = \begin{cases} 1 & \text{if } \hat{b}_t = b_t, \\ 0 & \text{otherwise.} \end{cases}$$

This formulation is inadequate for DST in three ways. (i) A predicted state comprises multiple slot–value pairs and partial correctness is common, yet exact match assigns positive reward only when all pairs match, discarding partial progress. (ii) For non-categorical slots with open-ended values,

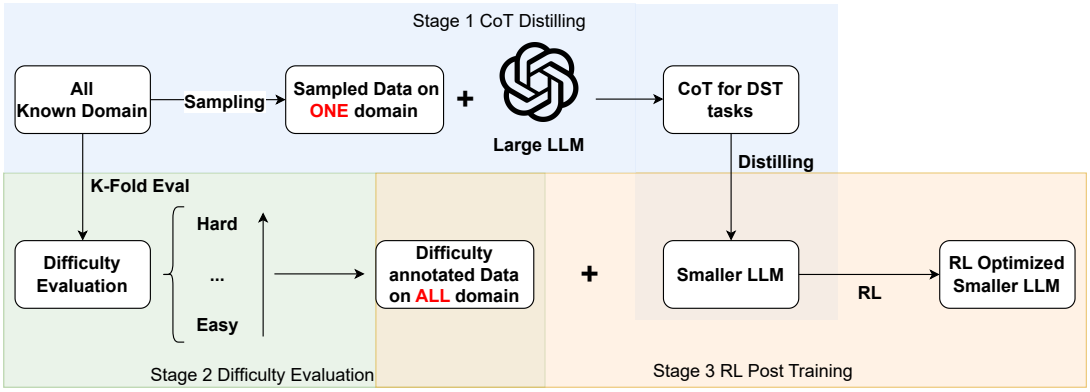

Figure 1: Proposed three-stage RL framework for zero-shot cross-domain DST post-training, featuring CoT distillation, difficulty evaluation, and RL optimization with Dr. GRPO.

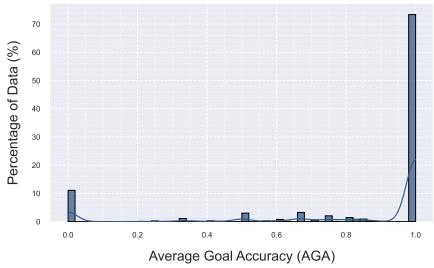

Figure 2: AGA distribution in MultiWOZ, showing difficulty imbalance

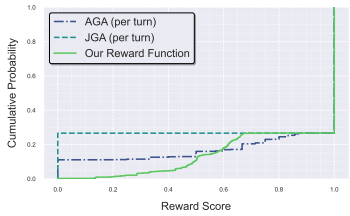

Figure 3: Comparison of reward distributions for exact and fuzzy match functions

semantically equivalent but textually distinct expressions (e.g., "Saint Thomas Hospital" vs. "St. Thomas Hospital") are incorrectly penalized. (iii) Slots vary in difficulty due to ambiguity or contextual dependency, yet exact match treats all slots uniformly.

Our Weighted Fuzzy Match Reward Function addresses these issues with three refinements:

- **Fuzzy Matching for Partial Credit.** For each slot–value pair we compute a fuzzy match ratio; if it exceeds a threshold τ, a partial reward $\delta < 1$ is assigned, providing feedback beyond binary matches.
- **Difficulty-Based Slot Weighting.** Using the difficulty estimates from Stage 2, we assign each slot a weight proportional to its error rate. Harder slots receive larger weights, directing the model's attention to areas most in need of refinement.
- **Per-turn Reward Aggregation.** For each slot we compute an individual reward (1 for exact match, δ if the fuzzy ratio meets τ, 0 otherwise), multiply by its difficulty weight, and average across slots in the turn. When not all slots match exactly, we further apply a discount factor $\gamma < 1$ to balance recognition of progress with the ultimate goal of complete accuracy.

Formally, for a turn with slots $s_1, \ldots, s_k$ the turn-level reward is

$$R_t = \begin{cases} 1 & \text{if } \hat{b}_t = b_t, \\ \gamma \cdot \frac{1}{k} \sum_{i=1}^{k} w_i \, r_i & \text{otherwise,} \end{cases}$$

where w_i is the difficulty weight for slot s_i and r_i is the per-slot reward

$$r_i = \begin{cases} 1 & \text{if exact match,} \\ \delta & \text{if fuzzy match ratio} \geq \tau, \\ 0 & \text{otherwise.} \end{cases}$$

As shown in Figure 3, this approach provides nuanced feedback, rewarding partial matches and prioritizing difficult slots, enhancing DST performance in RL frameworks.

4 Experiment

4.1 Evaluation Datasets

We evaluate our approach on three task-oriented dialogue datasets. **MultiWOZ 2.1** (Eric et al., 2020)

and **MultiWOZ 2.4** (Ye et al., 2022) serve as our primary in-domain benchmarks under the standard zero-shot cross-domain protocol (train on four domains, evaluate on the held-out domain). To further test generalization to entirely different schemas, we evaluate on a subset of the **Schema-Guided Dialogue (SGD)** dataset (Rastogi et al., 2020) in a cross-dataset zero-shot setting, where models trained only on MultiWOZ are tested on SGD services with no SGD-specific training. We report Joint Goal Accuracy (JGA) (Budzianowski et al., 2018), the exact-match accuracy over all slot-value pairs, and Average Goal Accuracy (AGA) (Rastogi et al., 2020), the accuracy of individual slot predictions.

4.2 Evaluation Baseline

We follow the standard zero-shot cross-domain protocol, training on MultiWOZ with one domain held-out for testing. We benchmark against baselines grouped by parameter count:

- **<1B**: TRADE (Wu et al., 2019b), T5DST (Lin et al., 2021b), TransferQA (Lin et al., 2021a), Prompter (Aksu et al., 2023), D3ST (Zhao et al., 2022), and CAPID (Dong et al., 2024b).
- **1B–100B**: D0T (Finch and Choi, 2024), SDT (Gupta et al., 2022), FNCTOD (Li et al., 2024), and LDST (Feng et al., 2023).
- **>100B**: IC-DST (Hu et al., 2022), Ref-PyDST (King and Flanigan, 2023), and ParsingDST (Wu et al., 2023). These models are typically closed-source and evaluated in a few-shot setting.

4.3 Implementation Details

We use Qwen2.5-0.5B and Qwen2.5-3B (Yang et al., 2024) as base models. For Chain-of-Thought (CoT) distillation, QwQ-32B (QwenTeam, 2025) generates 2,000 reasoning samples each for the hotel and train domains, which are then distilled into the base models via Supervised Fine-Tuning (SFT). We calculate initial difficulty scores using 2-fold validation on a fine-tuned Qwen2.5-0.5B.

For the Reinforcement Learning (RL) stage, we use Dynamic Difficulty Sampling with initial parameters $\mu = 0.75$ and $\sigma = 0.15$, step sizes $\mu_s = 0.05$ and $\sigma_s = 0.01$, and minimums $\mu_{\min} = 0.3$ and $\sigma_{\min} = 0.05$. Reward thresholds are set to 0.7 (0.5B model) and 0.75 (3B model), and the fuzzy and partial matching ratios are both 0.8. We use Llama-factory (Zheng et al., 2024) for SFT and Verl (Sheng et al., 2024) for RL; see Appendix C for full hyperparameters.

4.4 Experiment Results

To evaluate our zero-shot cross-domain DST approach, we first address the choice of query strategy. Prior methods use either per-domain or per-slot queries. While per-slot queries simplify the task and improve accuracy (e.g., LDST (Feng et al., 2023), D0T (Finch and Choi, 2024), and CAPID (Dong et al., 2024b)), they incur high computational costs: for a 12-turn hotel-domain dialogue with 10 slots, per-slot queries require 60 calls versus 6 for per-domain. We adopt the per-domain strategy for its efficiency, despite the added per-call complexity. Further details are provided in Appendix D.

Our results on MultiWOZ 2.1 and 2.4 are shown in Table 1 and Table 2. For models with fewer than 1B parameters, our Qwen2.5-0.5B (494M) model achieves state-of-the-art performance, with an average JGA of 61.0% on MultiWOZ 2.1 (vs. CAPID's 50.1%) and 62.6% on MultiWOZ 2.4 (vs. CAPID's 47.9%), reflecting superior accuracy and adaptability in lightweight settings. For the 1B–100B range, our Qwen2.5-3B model scores 76.1% JGA on MultiWOZ 2.1 and 77.7% on MultiWOZ 2.4, closely rivaling CAPID's 7B model and matching top baselines with roughly half the parameters.

We further compare our RL-based approach with SFT and CoT distillation on the Qwen2.5-0.5B model. Table 2 shows that SFT alone achieves 36.7% JGA and SFT with CoT only 32.6% JGA, both well below our RL method. CoT distillation particularly struggles on unknown domains, indicating that small LLMs find it difficult to absorb CoT directly from much larger LLMs. These results demonstrate that RL with test-time scaling outperforms traditional fine-tuning approaches, more effectively unlocking the reasoning potential of small LLMs for cross-domain DST.

4.5 Cross-Dataset Generalization to SGD

To probe whether our improvements transfer beyond the MultiWOZ ontology, we further evaluate our largest model (Qwen2.5-3B) on a subset of the Schema-Guided Dialogue (SGD) dataset (Rastogi et al., 2020), following the subset-selection protocol used in prior work. Critically, neither our model nor the CAPID baseline sees any SGD data

Method	Model	Parms	Attraction	Hotel	Restaurant	Taxi	Train	Average
			JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA
TRADE	ELMo	93.6M	19.9 / 55.5	13.7 / 65.3	11.5 / 53.4	60.6 / 73.9	22.4 / 49.3	25.6 / 59.5
T5DST	T5	60.5M	33.1 / -	21.2 / -	21.7 / -	64.6 / -	35.4 / -	35.2 / -
TransferQA	T5	770M	31.2 / 60.6	22.7 / 77.8	26.3 / 81.7	61.9 / 86.5	36.7 / **87.2**	35.8 / 78.8
Prompter	PPTOD	60.5M	35.8 / -	19.2 / -	26.0 / -	66.3 / -	39.0 / -	37.3 / -
D3ST	T5	220M	56.4 / -	21.8 / -	38.2 / -	78.4 / -	37.7 / -	46.5 / -
CAPID	T5	220+60.5M	40.9 / 69.0	31.1 / 72.6	31.6 / 69.1	65.4 / 83.8	34.3 / 65.9	40.7 / 72.1
CAPID	T5	220+220M	33.3 / 64.4	43.5 / 83.3	37.1 / 75.2	**87.1** / 92.0	**49.5** / 73.4	50.1 / 77.7
Ours	Qwen2.5	494M	**65.4 / 81.9**	**54.2 / 90.1**	**52.7 / 85.4**	84.1 / **94.9**	48.5 / 83.4	**61.0 / 87.1**
D0T	T5	11B	61.1 / -	27.6 / -	64.3 / -	46.9 / -	49.7 / -	49.9 / -
D0T	Llama2	13B	66.6 / -	- / -	67.2 / -	48.8 / -	66.5 / -	58.5 / -
FNCTOD	Llama2	13B	62.2 / -	46.8 / -	60.3 / -	67.5 / -	60.9 / -	59.5 / -
SDT	T5	11B	74.4 / -	33.9 / -	72.0 / -	86.4 / -	62.9 / -	65.9 / -
LDST	Llama2	7B	75.6 / -	63.3 / -	73.7 / -	**91.5** / -	75.0 / -	75.8 / -
CAPID	T5+Llama2	7B+220M	**83.6 / 92.6**	**71.6 / 94.2**	**77.5** / 95.3	91.2 / **96.0**	**90.0 / 97.8**	**82.8 / 95.2**
Ours	Qwen2.5	3B	75.3 / 89.7	64.7 / 93.0	75.6 / **95.5**	86.5 / 93.6	78.7 / 95.7	76.1 / 93.5
IC-DST	Codex	>100B	**62.1** / -	53.2 / -	54.9 / -	71.9 / -	51.4 / -	58.7 / -
FNCTOD	GPT-4	>100B	58.8 / -	45.1 / -	63.2 / -	**76.4** / -	69.5 / -	62.6 / -
RefPyDST	Codex	>100B	**62.1** / -	**56.6** / -	**68.2** / -	71.9 / -	**76.1** / -	**68.8** / -

Table 1: Evaluation Results on MultiWOZ 2.1 dataset

Method	Model	Parms	Attraction	Hotel	Restaurant	Taxi	Train	Average
			JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA	JGA / AGA
CAPID	T5	220+60.5M	47.9 / 74.3	38.7 / 77.1	29.4 / 67.9	73.3 / 88.2	47.9 / 74.4	47.4 / 76.4
CAPID	T5	220+220M	22.8 / 59.3	31.3 / 79.0	39.1 / 78.7	**89.3** / 93.8	**56.7** / 76.9	47.9 / 77.5
Ours(SFT only)	Qwen2.5	494M	24.9 / 60.1	28.4 / 67.5	31.5 / 73.7	63.2 / 76.7	35.4 / 74.1	36.7 / 70.4
Ours(SFT with CoT)	Qwen2.5	494M	18.0 / 58.1	21.7 / 57.0	27.3 / 63.2	67.5 / 79.2	28.7 / 72.3	32.6 / 66.0
Ours	Qwen2.5	494M	**69.7 / 84.5**	**54.4 / 90.1**	**54.0 / 89.5**	85.5 / **95.2**	49.4 / **83.0**	**62.6 / 88.5**
D0T	T5	11B	68.1 / -	32.0 / -	72.3 / -	50.6 / -	55.8 / -	55.7 / -
D0T	Llama2	13B	76.8 / -	56.4 / -	78.8 / -	54.7 / -	76.1 / -	68.6 / -
CAPID	T5+Llama2	7B+220M	**84.4** / 93.1	**71.3** / 94.5	**79.1** / 95.4	**91.6 / 96.0**	**89.6 / 97.6**	**83.2 / 95.3**
Ours	Qwen2.5	3B	79.2 / **94.5**	65.2 / **95.2**	75.4 / **95.9**	85.2 / 94.1	79.4 / 95.5	76.9 / 95.0
IC-DST	Gpt Codex	>100B	60.0 / -	46.7 / -	57.3 / -	71.3 / -	49.4 / -	56.9 / -
ParsingDST	Gpt-3.5	>100B	65.6 / -	46.8 / -	**67.7** / -	**80.6** / -	62.6 / -	**64.7** / -
RefPyDST	Gpt Codex	>100B	**70.9** / -	**51.2** / -	65.6 / -	67.1 / -	**69.2** / -	**64.7** / -

Table 2: Evaluation Results on MultiWOZ 2.4 dataset

during training: both are trained only on MultiWOZ and evaluated zero-shot on SGD services with previously unseen schemas. The supervised baselines (SGD-baseline, TransferQA, Prompter) are included for context only—they are trained on SGD itself and therefore operate in a fundamentally easier setting.

Table 3 reports per-service JGA. In the zero-shot transfer setting, our 3B model achieves an average JGA of 30.3%, a 5.4-point absolute improvement over the much larger CAPID 7B baseline (24.9%). Our model wins on seven of the eight evaluated services, with particularly large margins on Media (+9.8), Music (+8.4), Flights (+7.7), and Messaging (+7.3); it trails CAPID by only 1.2 points on Events. Notably, on Buses and Media our zero-shot 3B model also surpasses the fully supervised SGD-baseline and TransferQA models, and approaches Prompter's supervised numbers on several services. These results indicate that the difficulty-aware sampling and reward design generalize beyond the MultiWOZ schema and confer genuine zero-shot transfer benefits, not merely in-domain fitting.

5 Ablation Study

5.1 Handling Imbalanced Difficulty and Sparse Reward

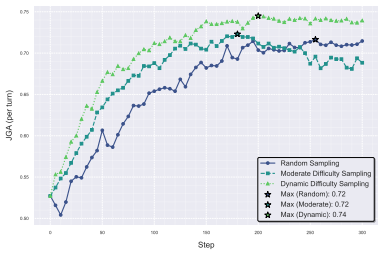

Figure 4: Comparison of Average convergence speed and JGA across different sampling strategies on hotel and train domain

To assess the effectiveness of sampling strategies in RL-based DST post-training, we used the Qwen2.5-0.5B model and the MultiWOZ 2.4

Setting	Method	Buses	Events	Flights	Media	Messaging	Music	Payment	Trains	Average
Supervised	SGD-baseline (Rastogi et al., 2020)	9.7	23.5	23.9	18.0	10.2	15.5	11.5	13.6	15.7
	TransferQA (Lin et al., 2021a)	15.9	15.6	3.6	30.2	13.3	8.9	24.7	17.4	16.2
	Prompter (Aksu et al., 2023)	48.4	51.5	–	65.3	59.2	–	21.9	50.8	49.5†
Zero-shot (MultiWOZ → SGD)	CAPID (Dong et al., 2024b), 7B	25.4	**29.7**	10.5	42.2	37.8	12.1	8.1	33.4	24.9
	Ours, Qwen2.5-3B	**31.0**	28.5	**18.2**	**52.0**	**45.1**	**20.5**	**12.0**	**35.2**	**30.3**

Table 3: Per-service JGA (%) on a subset of the SGD test set. Supervised baselines are trained on SGD; zero-shot rows are trained only on MultiWOZ and evaluated on SGD without any SGD-specific training. Bold marks the best result within the zero-shot block. †Prompter's average is computed over the 6 services it reports.

Reward function	JGA	Steps
JGA (Exact Match)	0.45	500+
AGA (turn-level)	0.68	320
+ fuzzy matching	0.65	240
+ weighted slot	0.71	180
+ partial factor (ours)	0.74	200

Table 4: Comparison of reward functions. AGA refers to turn-level partial matching.

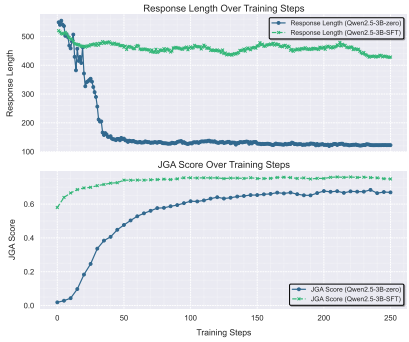

Figure 5: Comparison of reasoning length and JGA between models with and without initial SFT.

dataset. We conducted zero-shot testing, where the model generalizes to unseen domains without prior training, on the hotel and train domains. For evaluation, we used per-turn JGA, which measures the correctness of all slot-value predictions at each dialogue turn.

We compared three sampling strategies: (1) random sampling (baseline), (2) static moderate difficulty sampling (selecting 8,000 data points with difficulty scores between 0.1 and 0.8), and (3) our proposed dynamic difficulty sampling. Figure 4 shows that our dynamic method converged faster (at 200 steps) and achieved higher JGA than the alternatives. Static sampling outperformed random sampling, confirming the importance of addressing imbalanced difficulty in DST datasets. However, it risked overfitting due to its fixed data pool, while our adaptive approach adjusted to the model's progress, proving more effective.

Next, we explored the impact of reward functions on RL-based DST performance, using the same model, dataset, and dynamic sampling. Table 4 summarizes the results after a maximum of 500 training steps. Using JGA directly as the reward led to slow convergence, with gains continuing past 500 steps. Turn-level AGA, which rewards slot-level correctness, markedly improved performance. Adding fuzzy matching—which rewards semantically similar predictions—further sped up convergence. Incorporating both weighted slots (emphasizing harder slots) and partial credit (for partially correct predictions) yielded the highest JGA of 0.74 at 200 steps. Combined in our Weighted Fuzzy Match Reward function, these enhancements improve both efficiency and accuracy.

Together, these results show that our approach effectively tackles the imbalanced-difficulty and sparse-reward challenges of zero-shot cross-domain DST with small LLMs such as Qwen2.5-0.5B.

5.2 Direct RL Training without SFT

We investigated whether small LLMs can effectively learn DST from direct RL without an initial SFT stage, a strategy that has been shown to work for larger models (Shao et al., 2024). We applied RL directly to the base Qwen2.5-0.5B and Qwen2.5-3B models in a one-shot in-context learning framework.

The results highlight clear limitations of this approach for small LLMs. The 0.5B model failed to learn, producing nonsensical outputs throughout training. The 3B model reached a JGA of 0.67 but remained well below the 0.79 JGA of its SFT-initialized counterpart. As illustrated in Figure 5, SFT-initialized models produce longer, more accurate responses (avg. 420 tokens) than direct-RL models (avg. 120 tokens), which exhibit limited reasoning. These findings suggest that SFT is a crucial prerequisite for developing effective DST reasoning in small LLMs.

6 Related Work

6.1 Zero-shot Cross-domain DST

Traditionally, zero-shot cross-domain DST research focused on optimizing model architectures to enable generalization across dialogue domains without domain-specific training (Wu et al., 2019b; Lin et al., 2021a; Wang et al., 2021). The emergence of LLMs has shifted the paradigm due to their robust generalization capabilities. Recent studies have prioritized generating informative prompts or synthetic data to enhance zero-shot cross-domain DST performance, typically through SFT in an imitation learning framework. Hu et al. (2022) explored in-context learning for few-shot DST, which can be adapted to zero-shot settings by providing contextual examples. Dong et al. (2024b) introduced context-aware auto-prompting and contrastive decoding to improve LLM performance in cross-domain DST. Finch and Choi (2024) generated diverse synthetic data to make zero-shot DST more adaptable, further leveraging SFT to align models with task requirements. However, the application of RL to optimize test-time scaling and enhance reasoning during inference remain largely unexplored in DST. Our work addresses this gap by investigating RL-driven optimization for small LLMs in zero-shot cross-domain DST, aiming to balance performance and practicality.

6.2 Verifiable Reinforcement Learning

Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), particularly using Proximal Policy Optimization (PPO) (Schulman et al., 2017), has been shown to significantly improve LLMs' generalization and output quality. Building on this, OpenAI (OpenAI, 2024) demonstrated that RL can enhance LLMs' reasoning capabilities, implementing test-time scaling to produce more accurate responses. However, these methods require extensive human-annotated reasoning data, which is costly and difficult to obtain, limiting their scalability. To address this challenge, recent work has introduced rule-based verifiable RL approaches such as ReMax (Li et al., 2023), RLOO (Ahmadian et al., 2024), and GRPO (Shao et al., 2024) that reduce reliance on external data by allowing the base LLM to self-generate high-quality reasoning tokens. Subsequent analyses have refined the GRPO family: Dr. GRPO (Liu et al., 2025) removes the response-length and difficulty biases of standard GRPO and shows that the KL term is unnecessary for verifiable-reward training, which we adopt as our base optimizer. Despite their success in mathematics and code, verifiable RL methods have rarely been applied to DST, particularly with small LLMs. Our study fills this gap by combining a debiased GRPO objective with DST-specific difficulty-aware sampling and reward design.

7 Conclusion

This work introduces a verifiable RL-based approach for zero-shot cross-domain DST with small LLMs. By combining Dynamic Difficulty Sampling and a Difficulty-Weighted Fuzzy Match Reward Function with the debiased Dr. GRPO optimizer (Liu et al., 2025), we directly address the imbalanced-difficulty and sparse-reward challenges inherent to DST. Our method achieves state-of-the-art performance among models under 1B parameters and rivals significantly larger systems on MultiWOZ 2.1 and 2.4; in a stricter cross-dataset setting, our 3B model trained only on MultiWOZ also outperforms a 7B baseline on a SGD subset by 5.4 absolute JGA, indicating that the gains generalize beyond a single schema. These findings underscore the viability of small LLMs for efficient, scalable dialogue systems and offer a pathway to resource-efficient deployment where computational limits are critical.

Limitations

Our approach has several limitations. (i) Although competitive within their parameter class, small LLMs still underperform much larger models in some domains, and the SFT-then-RL pipeline relies on a large teacher (QwQ-32B) for the initial CoT distillation. (ii) RL post-training demands non-trivial computational resources (4×H100 in our setup), which may limit accessibility for smaller labs. (iii) Several hyperparameters in the sampling distribution and reward function (e.g., μ, σ, τ, δ, γ) are hand-tuned; transferring the recipe to new datasets may require re-tuning. (iv) Difficulty is estimated once and remains static during RL, ignoring concept drift in the policy's actual error pattern; an online or meta-learned difficulty estimator could address this. (v) Our evaluation covers MultiWOZ 2.1, MultiWOZ 2.4, and a subset of SGD; while the SGD transfer results are encouraging, broader empirical validation on the full SGD test set, SGD-X, and other DST benchmarks would further establish generality. (vi) The CoT outputs produced by our

model (avg. 420 tokens) increase per-turn computational load relative to non-reasoning baselines, which is a relevant consideration for real-time deployment.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.

Taha Aksu, Min-Yen Kan, and Nancy Chen. 2023. Prompter: Zero-shot adaptive prefixes for dialogue state tracking domain adaptation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4588–4603, Toronto, Canada. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024a. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8527–8540.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024b. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8527–8540, Miami, Florida, USA. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

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

James D. Finch and Jinho D. Choi. 2024. Diverse and effective synthetic data generation for adaptable zero-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12527–12544, Miami, Florida, USA. Association for Computational Linguistics.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Yuan Cao, Abhinav Rastogi, and Yonghui Wu. 2022. Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4541–4549, Seattle, United States. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. *arXiv preprint arXiv:2005.02877*.

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.

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

Léo Jacqmin, Lina M Rojas-Barahona, and Benoit Favre. 2022. " Do you follow me?": A Survey of Recent Approaches in Dialogue State Tracking. *arXiv preprint arXiv:2207.14627*.

Brendan King and Jeffrey Flanigan. 2023. Diverse retrieval-augmented in-context learning for dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5570–5585, Toronto, Canada. Association for Computational Linguistics.

Zekun Li, Zhiyu Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Dong, Adithya Sagar, Xifeng Yan, and Paul Crook. 2024. Large language models as zero-shot dialogue state tracker through function calling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8688–8704, Bangkok, Thailand. Association for Computational Linguistics.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023. Remax: A

simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021a. Zero-shot dialogue state tracking via cross-task transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021b. Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

OpenAI. 2024. Learning to reason with LLMs.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. 2025. Tinyzero. https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

QwenTeam. 2025. Qwq-32b: Embracing the power of reinforcement learning.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Chenghong Wang, Jieren Deng, Xianrui Meng, Yijue Wang, Ji Li, Sheng Lin, Shuo Han, Fei Miao, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. A secure and efficient federated learning framework for NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7676–7682, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.

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

Yuxiang Wu, Guanting Dong, and Weiran Xu. 2023. Semantic parsing by large language models for intricate updating strategies of zero-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11093–11099, Singapore. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK. Association for Computational Linguistics.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 2025. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason. Notion Blog.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Role of Large Language Models in This Study

In this study, LLMs were integral to both the technical implementation and the final manuscript preparation. For software development, we leveraged an LLM as an intelligent assistant to aid in coding and debugging. Furthermore, to enhance the clarity and quality of this publication, we utilized the LLM to perform comprehensive grammatical checks and proofread the manuscript.

B Prompt Template

In this section, we list the prompt we used in our proposed DST RL strategy.

Prompt for DST CoT distilling from large LLMs:

You are an expert in Dialogue State Tracking (DST). Your task is to generate chain of thought for solving provided DST tasks.
Instructions:
- Generate your chain of thought that helps to arrive at the provided dialogue state in target domain.
- Pay more attention to the difficult and complex dialogue states, and less attention to the easy ones.
- If the dialogue is not related to the target domain (the ground-truth dialogue states are all None), please indicate that in your CoT with a short explanation.
- The length of CoT should be in 100 to 500 words.
Input:
- Target Domain:
{domain}
- Dialogue:
{dialogue}
- Related Slots and explanations:
{slots}
- The ground-truth dialogue states:
{dialogue_state}
Output Format:
- Generate answer in <think> tags. Example: <think>Your CoT...</think>
Now, analyze the given content and generate your chain of thought.

Prompt for small LLM generate both CoT and DST results:

For the given dialogue, generate step-by-step reasoning and determine the dialogue state for the domain {domain}.
Input:
- Target Domain:
{domain}
- Dialogue:
{dialogue}
- Related Slots and explanations:
{slots}
Output Format:
Output your reasoning in <think> tag and the dialogue state in <answer> tag using following format:
<think>your reasoning..</think>
<answer>slot1:value1,slot2:value2</answer>
Now, analyze the given content and generate your step-by-step reasoning and determine the dialogue state:

C Detail Hyperparameter

In this appendix, we provide the detailed hyperparameter settings used in our experiments for both Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) tasks. All experiments were conducted on a 4×H100 cluster.

C.1 Supervised Fine-Tuning (SFT)

For SFT tasks, we utilized the llama-factory tool as our fine-tuning framework. The hyperparameters are listed in Table 5.

Table 5: Hyperparameters for SFT tasks

Hyperparameter	Value
per_device_train_batch_size	1
gradient_accumulation_steps	10
learning_rate	1.0e-5
num_train_epochs	3.0
lr_scheduler_type	cosine
warmup_ratio	0.1

C.2 Reinforcement Learning (RL)

For RL tasks, we utilized the verl framework with vLLM as the rollout tool. The hyperparameters are listed in Table 6.

For more details, please refer to our code.

Table 6: Hyperparameters for RL tasks

Hyperparameter	Value
TRAIN_BATCH_SIZE	512
MAX_PROMPT_LENGTH	1280
MAX_RESPONSE_LENGTH	2048
LEARNING_RATE	1e-6
TEMPERATURE	1.0
ROLLOUT_N	5

D Efficiency Analysis of Per-Domain and Per-Slot Query Strategies

In this section, we evaluate the efficiency of per-domain and per-slot query strategies with respect to computational resource utilization, focusing on token usage and the number of queries required. As outlined in Section 4.4, the per-slot query strategy, while simplifying the problem and potentially enhancing accuracy, introduces a substantial computational burden by necessitating individual queries for each slot at every turn. To assess this trade-off quantitatively, we compare a representative per-slot query method, LDST, with our proposed per-domain query model, which is built upon Qwen2.5-0.5B. Two variants of our model are examined: one employing SFT without CoT reasoning, and another incorporating RL post-training with CoT reasoning. The evaluation is conducted using 100 randomly selected dialogues from the MultiWOZ2.4 training set.

The efficiency metrics, including the average number of tokens used for prompts, responses, total tokens, and the average number of queries per dialogue, are summarized in Table 7 below.

The results reveal a marked disparity in computational efficiency between the two strategies. The per-slot query approach, exemplified by LDST, consumes an average of 37510 total tokens and requires 77.5 queries per dialogue. In contrast, our per-domain query strategy without CoT reasoning utilizes only 4665.5 total tokens and 7.75 queries, reflecting reductions of approximately 87.6% in token usage and 90% in query count. Even when enhanced with CoT reasoning, which increases token consumption due to the generation of reasoning steps, our per-domain model requires 7982.5 total tokens—still significantly lower than the per-slot strategy.

Table 7: Efficiency comparison of query strategies in terms of token usage and query count

Query type	Method	Avg. Prompt	Avg. Response	Avg. total	Avg. #Query
Per Slot	LDST	36657.5	852.5	37510	77.5
Per Domain	Ours (DST only)	3975.75	689.75	4665.5	7.75
Per Domain	Ours (CoT)	4247	3735.5	7982.5	7.75

E Example Result

In this section, we include several failure cases from the Qwen2.5-0.5B CoT DST model in the hotel domain. The incorrect reasoning is marked in red.

LLM hallucination on no existing slots

<think>The dialogue revolves around the user requesting a place to stay (hotel) in the north side of town, specifically asking for a hotel with a star of 4, free parking, and free wifi. Let's break down the dialogue to identify the relevant slots for the hotel domain.

1. **Initial Request for Hotel**: The user transitions from requesting a train to requesting a hotel. The key statement is: "I would like a place to stay that is in the north side of town. [SYSTEM]: I can certainly help you with that. There are many accommodations on the north side of town. Do you have any preferences?" This explicitly sets the 'hotel_area' to "north" and the 'hotel_location' to "north". The user specifies "north side of town," which directly sets the 'hotel_region' to "north."

2. **Hotel Domain Establishment**: The user explicitly states: "I would like a place with a star of 4 that includes free parking and free wifi." This directly sets the 'hotel_stars' to "4" and the 'hotel_internet' to "yes." The mention of "free parking" and "free wifi" clearly sets 'hotel_parking' and 'hotel_internet' to "yes."

3. **Hotel Domain Details**: The user specifies "moderate hotel with free wifi and parking." This sets 'hotel_parking' to "yes" and 'hotel_pricerange' to "moderate." The mention of "moderate hotel" indicates a moderate price range, which is captured in 'hotel_pricerange'. The phrase "free wifi and parking" directly sets 'hotel_internet' to "yes" and 'hotel_parking' to "yes."

4. **Hotel Domain Validation**: The user's request for a hotel is explicitly confirmed by the system, and no new slots are added. The user's focus on hotel details (e.g., area, price, parking) is directly mapped to the corresponding slots. The absence of any other slots (e.g., date, number of people, stay) is checked, and all these slots are correctly filled based on the dialogue.

5. **Conclusion**: The dialogue state correctly reflects the user's explicit requirements for the hotel domain, including the slot values specified by the user. No ambiguities arise. The user's request for a hotel is directly addressed, and all relevant slots are properly populated based on the dialogue.</think>

<answer>hotel_area:north, hotel_book-day:NONE, hotel_book-people:NONE, hotel_bookstay:NONE, hotel_internet:yes, hotel_name:NONE, hotel_parking:yes, hotel_pricerange:moderate, hotel_stars:4, hotel_type:guesthouse, hotel_destination:NONE, hotel_contact:NONE </answer>

Misunderstanding on Yes, No and None option.

<think>The dialogue revolves around the user seeking a hotel in the centre of Cambridge for a cheap price range. The target domain is "hotel", so we focus on the hotel-related slots.

1. **Hotel Domain Analysis**: The user's query is strictly about a hotel, so the hotel-related slots (hotel_name, hotel_parking, hotel_book-day, hotel_book-people, hotel_book-stay, hotel_area, hotel_price, hotel_internet, hotel_star, hotel_type) are directly addressed. The user specifies "Alexander Bed and Breakfast" as the hotel, so "hotel_name" is clearly "Alexander Bed and Breakfast".

2. **Hotel Book-Day**: The user selects "Sunday" for the hotel stay, so "hotel_book-day" is "sunday".

3. **Hotel Book-People**: The user confirms "2 people" for the hotel, so "hotel_book-people" is "2".

4. **Hotel Book-Stay**: The user specifies "2 nights" for the hotel, so "hotel_stay" is "2".

5. **Hotel Price Range**: The user specifies "cheap price range" for the hotel, so "hotel_pricerange" is "cheap".

6. **Hotel Area**: The user mentions "centre" for the hotel, so "hotel_area" is "centre".

7. **Hotel Internet**: The user does not mention any internet service provision, so "hotel_internet" is "no".

The user's query is strictly about a hotel, and the dialogue is exclusively about hotel booking. The hotel slot is clearly addressed by the user, and all other slots are directly derived from the user's explicit specification. The absence of any other mentions or requirements related to the hotel (e.g., room type, schedule, or amenities) results in the dialogue state being straightforward.</think>

<answer>hotel_area:centre, hotel_book-day:sunday, hotel_book-people:2, hotel_book-stay:2, hotel_internet:no, hotel_name:alexander bed and breakfast, hotel_parking:NONE, hotel_pricerange:cheap, hotel_stars:NONE, hotel_type:NONE </answer>

LLM confused on hotel and guesthouse. Here, hotel refers to both the domain and a value of the hotel_type slot.

<think>The dialogue revolves around the user seeking a cheap hotel. The target domain is "hotel", so we focus on the relevant slots.

1. **Hotel Domain Identification**: The user explicitly mentions "cheap guesthouse" and "cheap hotel" as requirements. The system provides "Alexander Bed and Breakfast" as the hotel, and the user's query is about a cheap hotel. The domain is "hotel", so all slots related to the hotel are addressed.

2. **Slot Analysis**: - **Hotel Name**: The user specifies "Alexander Bed and Breakfast" as the hotel. This directly fills the ‘hotel_name‘ slot. - **Hotel Price Range**: The user specifies "cheap" for the hotel, so ‘hotel_pricerange‘ is "cheap". - **Hotel Book-Day**: The user does not mention a specific day for checking in. The guesthouse is booked for Friday, but the hotel is not specified. The system's response is about the guesthouse, not the hotel. - **Hotel Book-People**: The user booked a guesthouse for 8 people, not a hotel. The hotel request is unrelated. - **Other Slots**: The user's query is about a hotel, not a guesthouse. The parking, restaurant, or other details are not mentioned in the hotel context.

3. **No Other Slots**: The user's query is strictly about a hotel, and the system provides details about the hotel. There is no mention of the hotel's area, booking day, or other parameters. The guesthouse's details (e.g., parking, WiFi) are unrelated to the hotel domain.

The dialogue is strictly about the user seeking a cheap hotel, and the relevant slots are directly addressed. There is no overlap with the guesthouse's details, so all slots related to the hotel are properly populated.</think>

<answer>hotel_area:NONE, hotel_book-day:NONE, hotel_book-people:NONE, hotel_book-stay:NONE, hotel_internet:NONE, hotel_name:NONE, hotel_parking:NONE, hotel_pricerange:cheap, hotel_stars:NONE, hotel_type:NONE</answer>