Bridging Reasoning and Action: Hybrid LLM–RL Framework for Efficient Cross-Domain Task-Oriented Dialogue

Anonymous ACL submission

Abstract

Cross-domain task-oriented dialogue requires reasoning over implicit and explicit feasibility constraints while planning long-horizon, multi-turn actions. Large language models (LLMs) can infer such constraints but are unreliable over long horizons, while Reinforcement learning (RL) optimizes long-horizon behavior yet cannot recover constraints from raw dialogue. Naively coupling LLMs with RL is therefore brittle: unverified or unstructured LLM outputs can corrupt state representations and misguide policy learning. Motivated by this, we propose **V**erified **LLM-K**nowledge empowered **RL** (**VLK-RL**), a hybrid framework that makes LLM-derived constraint reasoning usable for RL. VLK-RL first elicits candidate constraints with an LLM and then verifies them via a dual-role cross-examination procedure to suppress hallucinations and cross-turn inconsistencies. The verified constraints are mapped into ontology-aligned slot–value representations, yielding a structured, constraint-aware state for RL policy optimization. Experiments across multiple benchmarks demonstrate that VLK-RL significantly improves generalization and robustness, outperforming strong single-model baselines on long-horizon tasks.

1 Introduction

Task-oriented dialogue systems have shown strong performance in single-domain and loosely coupled multi-domain settings (Fernández et al., 2025). However, cross-domain dialogues, where decisions in one domain impose feasibility constraints on others, remain particularly challenging (Zhu et al., 2020a; Qin et al., 2023). These constraints are often only partially stated and may be *explicit* or *implicit*, requiring commonsense or temporal reasoning, as illustrated in Fig. 1. Though rarely expressed verbatim, these constraints are critical for ensuring globally valid actions across tasks.

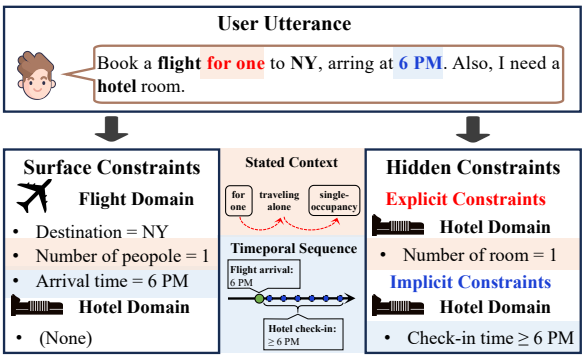


Figure 1: Example of *explicit* (traveling alone implies single-occupancy accommodation) and *implicit* (hotel check-in must follow flight arrival) constraints in a cross-domain scenario.

Existing methods largely address this problem through either dialogue state construction or policy optimization. Dialogue state tracking aims to uncover latent user information within a predefined ontology (Dong et al., 2024; Lin et al., 2021), but they are brittle when cross-domain constraints are not directly grounded in surface text or require commonsense inference. On the other hand, multi-task or hierarchical reinforcement learning (RL) improves long-horizon decision making (Rohmatillah et al., 2023; Zhou et al., 2024), yet typically assumes accurate and complete states and degrades when critical constraints are missing (Zhao et al., 2024). Together, these observations suggest a bottleneck: cross-domain policy learning is fundamentally constrained by whether the dialogue state captures (implicit and explicit) feasibility constraints.

Large language models (LLMs) offer a promising source of such constraint knowledge, as they encode substantial commonsense knowledge and can infer such constraints from context (Zhang et al., 2024; He et al., 2022). Nevertheless, using LLMs as end-to-end dialogue agents or directly integrating them into long-horizon RL policy pipelines is unreliable for task-oriented systems (Yi et al., 2024; Nguyen et al., 2025). LLM outputs may hal-

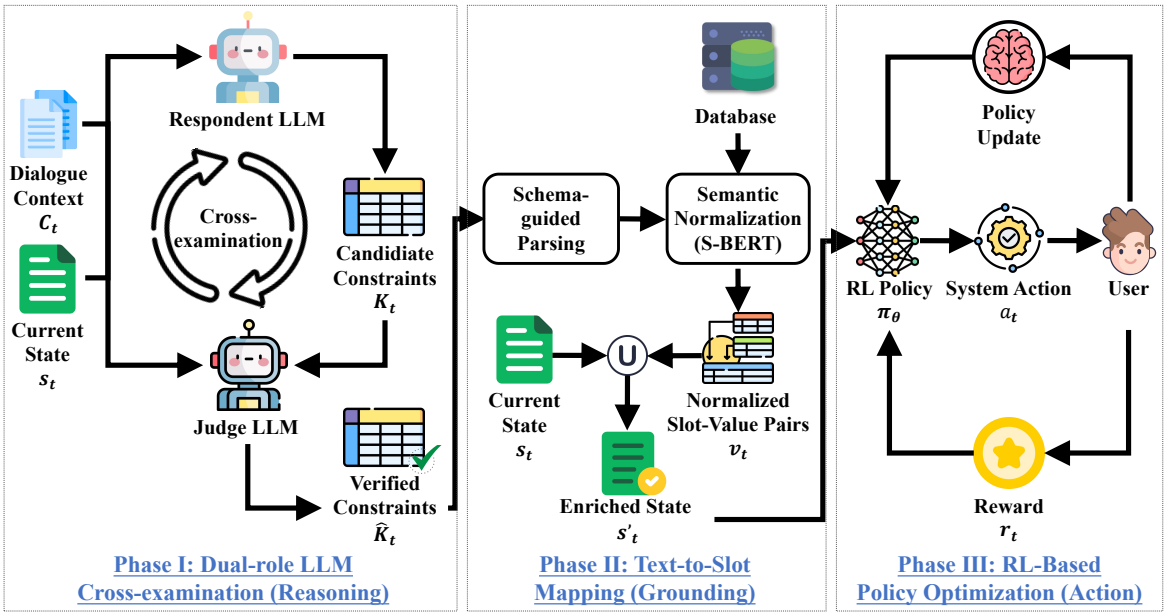


Figure 2: Overview of the proposed VLK-RL framework. Candidate constraints are inferred from the respondent LLM and verified by dual-role cross-examination, then grounded into structured normalized slot-value pairs to enrich the dialogue state for downstream RL policy optimization.

lucinate and drift across turns, and their free-form generations are hard to verify, align with ontologies, or safely encode as state inputs. Consequently, naively coupling LLM reasoning with RL policy can corrupt state representations and misguide policy learning over long horizons.

We address this gap by reframing constraint modeling as *constraint-aware dialogue state construction* and propose **V**erified **LLM**-**K**nowledge empowered **RL** (**VLK-RL**), a hybrid framework that makes LLM-derived constraint reasoning reliable and actionable for RL policies. Given a dialogue context, VLK-RL first elicits candidate explicit and implicit feasibility constraints with an LLM. A key challenge is that such inferences are not directly trustworthy for long-horizon decision making. To this end, we introduce a *dual-role cross-examination* mechanism inspired by fact verification (Cohen et al., 2023), in which two LLMs assume differentiated roles—a *respondent* and a *judge*—to collaboratively validate inferred constraints through cross-examination dialogue without external supervision. Even after verification, constraints remain free-form and cannot be directly consumed by downstream RL. We therefore design a *text-to-slot mapper* that grounds verified constraints into structured normalized representations compatible with downstream RL. The resulting structured, constraint-aware states can be consumed by standard RL policies without changing policy architectures, enabling robust long-horizon planning under cross-domain feasibility. We evaluate VLK-RL on MultiWOZ 2.1 and Frames, and observe consistent gains in cross-domain generalization and policy robustness over strong cross-domain single-model baselines.

In summary, our contributions are threefold:

- We identify feasibility-constraint completeness as a central bottleneck for cross-domain task-oriented dialogue, and formulate constraint modeling as a state construction problem that unifies explicit and implicit dependencies across domains.
- We propose VLK-RL, which integrates LLM reasoning with RL policy optimization, while ensuring both inferred knowledge reliability and modality alignment without external supervision.
- Extensive experiments on multiple benchmarks demonstrate that injecting verified, structured constraint knowledge improves cross-domain generalization and robustness.

2 Method

As shown in Fig. 2, VLK-RL decouples *reasoning* from *control* via a modular state-construction interface with three components: (1) a *dual-role LLM cross-examination module* that verifies explicit and implicit constraints via cross-examination dialogue; (2) a *text-to-slot mapper* that grounds verified constraints into structured normalized slot-value pairs

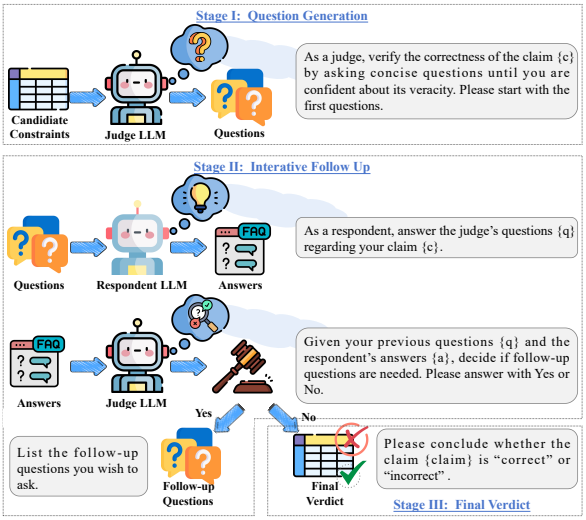


Figure 3: Dual-role cross-examination. A *respondent* proposes constraints; a *judge* probes with targeted questions and returns a verification verdict, filtering hallucinations and cross-turn inconsistencies.

and augments the dialogue state from s_t to an enriched state s'_t; and (3) an *RL-based policy* that makes long-horizon decisions conditioned on s'_t.

2.1 Dual-role LLM Cross-examination Module

Although LLMs possess world knowledge and strong commonsense reasoning ability, their outputs often contain hallucinations and inconsistencies. Directly injecting such unreliable reasoning into policy learning risks severe performance degradation. To address this, we introduce a *dual-role cross-examination module*, inspired by Cohen et al. (2023), where two LLMs assume complementary roles—*respondent* and *judge*—and engage in interactive reasoning to filter factual errors. The core rationale of this module is that interactions between different roles can reveal hidden inconsistencies in reasoning. Such inconsistencies can be considered as signals of uncertainty in the respondent's original claims, and thus provide a critical basis for judging whether its inferences are correct.

Respondent. The respondent LLM takes the dialogue context C_t and current state s_t as input and proposes a set of candidate constraint inferences $K_t = \mathcal{R}(C_t, s_t)$. It covers both explicit constraints and implicit constraints. We design structured CoT prompts (see App. A for details) to guide the respondent LLM in identifying domains, extracting facts, and deriving candidate constraints, thereby improving both reasoning accuracy and interpretability by exposing intermediate steps. During cross-examination, the respondent LLM further defends its claims by answering the judge's challenges to simulate dialogue-based validation.

Judge. Conditioned on (C_t, K_t), the judge LLM evaluates each candidate $k \in K_t$ through an interactive probing procedure. Concretely, the cross-examination proceeds in three prompted stages, as shown in Fig. 3:

In Stage I (*question generation*), the judge LLM formulates clarification questions regarding the claims of respondent LLM, which are sequentially answered by the respondent and appended to the context. In Stage II (*iterative follow-up*), the judge LLM inspects the answers and, if contradictions or gaps are detected, issues further questions until no new queries arise or a predefined round limit is reached (set to 5 in our experiments). In Stage III (*final verdict*), the judge outputs a binary decision (`True`/`False`) for each constraint claim, indicating whether it is logically valid or unsupported. Intuitively, contradictions or evasive responses in this process are treated as signals of uncertainty, leading to the rejection of the corresponding inference. Only the subset of verified constraints endorsed by the judge LLM, denoted $\hat{K}_t$, are retained as reliable knowledge for downstream processing.

2.2 Text-to-Slot Mapper

Although the verified constraints $\hat{K}_t$ are reliable, they remain expressed in natural language, which is incompatible with structured policy representations. Moreover, the raw outputs of LLMs may contain values that do not exactly correspond to entries in the TOD database (e.g., 'NYC downtown' versus the ontology term 'Manhattan'), which would cause execution failures.

To bridge this gap, we design a text-to-slot mapper $\mathcal{M} : \hat{K}_t \rightarrow V_t$, where $V_t = \{(s_i, v_i)\}$ denotes a set of slot–value pairs consistent with the TOD database. Each verified constraint $k \in \hat{K}_t$ is first parsed into candidate slot–value pairs using schema-guided extraction rules and ontology-based templates. This rule-driven approach requires no additional training and ensures that free-form natural language descriptions are aligned with the predefined slot schema across domains.

To resolve the mismatch between LLM-generated values and database-predefined values, we perform semantic similarity-based normalization, guaranteeing that all slot–value pairs are fully compatible with the database entries. For each (s, v), if $s \in DB$, we retrieve valid values for s

and compute embedding similarity with v using Sentence-BERT (Reimers and Gurevych, 2019). The most similar entry is selected:

$$\tilde{v} = \arg \max_{v' \in DB(s)} \cos(\mathbf{e}(v), \mathbf{e}(v')).$$

If $\cos(\mathbf{e}(v), \mathbf{e}(\tilde{v})) \geq \tau$ (with $\tau = 0.7$), v is replaced with $\tilde{v}$; otherwise, the pair is discarded. If s is not in the ontology, we first match the most similar slot and then normalize its value. Finally, normalized slot–value pairs (e.g., `hotel_area = Midtown Manhattan`) are integrated into the dialogue state:

$$s'_t = s_t \cup v_t,$$

where s_t is the original state and v_t contains structured, verified knowledge. The enriched state s'_t is then passed to the RL-based policy optimizer for robust decision-making.

2.3 RL-based Policy Optimizer

Although LLMs excel at knowledge reasoning, they struggle with long-horizon decision-making. RL, on the other hand, provides a principled framework for optimizing policies based on long-horizon rewards. By combining RL with validated knowledge from LLMs, our framework aims to achieve both policy robustness and cross-domain generalization.

We formulate TOD policy optimization as a Markov Decision Process defined by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. The state $s \in \mathcal{S}$ is the enriched dialogue state, which encodes the dialogue context, historical slot–value pairs, validated cross-domain constraints, and dialogue turn information. The action $a \in \mathcal{A}$ corresponds to a system response action. The reward $R(s, a)$ follows the MultiWOZ convention: completing all domain goals yields a reward of $2L$, completing only a single domain yields $+5$, failure yields $-L$, and each intermediate turn incurs a -1 penalty to encourage concise dialogues. To update the policy, we adopt Proximal Policy Optimization (PPO) as a representative RL algorithm due to its stability and strong empirical performance in dialogue policy learning. The clipped surrogate objective is given by:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \Big[\min \big(r_t(\theta)\, \hat{A}_t^{s'}, \\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, \hat{A}_t^{s'} \big) \Big] \quad (1)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s'_t)}{\pi_{\theta_{\text{old}}}(a_t|s'_t)}$ is the probability ratio between the updated and the previous policy on the enriched state s'_t. $\hat{A}_t^{s'}$ denotes the advantage estimate for the enriched state s'_t (i.e., the state that incorporates both historical slots and the verified, normalized slot–value pairs). The expectation $\hat{\mathbb{E}}_t$ is taken over timesteps in a sampled minibatch, and ϵ is the clipping hyperparameter.

Algorithm 1 VLK-RL Framework for Multi-domain TOD

Require: Dialogue context C, database DB, RL policy π, dialogue state s

1: **for** each dialogue turn t **do**
2: $K_t \leftarrow$ Respondent LLM inference on C_t
3: $\hat{K}_t \leftarrow$ Judge LLM verifies K_t
4: $v_t \leftarrow \mathcal{M}(\hat{K}_t, DB)$ {text-to-slot mapping and normalization}
5: $s'_t \leftarrow s_t \cup v_t$
6: Sample action $a_t \sim \pi_\theta(\cdot \mid s'_t)$
7: Execute system response action a_t
8: **end for**

VLK-RL enhances s_t with verified feasibility constraints by modifying only state construction, while keeping the policy architecture, action space, and optimization objective unchanged. As a result, it is fully compatible with standard RL policy learning. For concreteness, we instantiate the framework with PPO as our default optimizer. Importantly, other RL backbones can be plugged in without altering our verification or grounding modules, a claim we further validate in App. C.

Alg. 1 summarizes the workflow. At each turn, the framework first performs cross-examination to infer and validate constraints (lines 2–3), then maps the validated constraints into ontology-aligned slot–value pairs (line 4), merges them into the dialogue state (line 5), and finally selects the next system action through RL policy (lines 6–7).

3 Experiments

We evaluate VLK-RL on cross-domain task-oriented dialogue along three axes: (i) overall performance and robustness against strong baselines in simulated and human evaluations (Sec. 3.5.1 and Sec. 3.5.2), (ii) the contribution of each module via ablations (Sec. 3.6), and (iii) constraint-oriented analyses that combine explicit/implicit failure statistics with qualitative case studies (Sec. 3.7). Additional studies (e.g., low-resource training (App. I), cross-model cross-examination (App. D), and alternative RL backbones (App. C) are deferred to the appendix.

3.1 Datasets

We conduct experiments on two widely used multi-domain benchmarks: MultiWOZ 2.1 (Zhu et al., 2020b) and Frames[1]. MultiWOZ 2.1 contains over 10k human–human dialogues spanning seven domains with annotated slot–value states, and exhibits rich cross-domain dependencies. Frames is a Wizard-of-Oz dataset with interrelated subtasks and strong cross-task feasibility requirements; following Peng et al. (2017), we adopt a modified schema that encodes inter-subtask constraints and preferences. We use ConvLab-2[2] for simulation, database access, and evaluation to ensure reproducibility and fair comparison with prior work[3].

3.2 Experimental Setup

Unless stated otherwise, we follow ConvLab-2 defaults. Key hyperparameters are: training epochs $= 300$, maximum dialogue length $L = 30$, batch size $= 100$, cross-examination rounds $R = 5$, and normalization threshold $\tau = 0.7$. We use PPO as the RL optimizer. For the LLM backbone, we evaluate Qwen2-7B-Instruct, Qwen1.5-14B-Chat (GPTQ-Int4), and GPT-4o-mini to cover different capability and deployment regimes. The Qwen family provides strong reasoning for multi-turn dialogue and supports efficient local deployment via quantization, while GPT-4o-mini represents a high-performance commercial model, serving as a strong reference point. By default, the *judge* and *respondent* roles share the same backbone with role-specific prompts, and we report cross-model variants in App. D. All LLMs are used off-the-shelf without task-specific fine-tuning. All results are averaged over 5 runs with different random seeds.

3.3 Evaluation Metrics

We report standard ConvLab-2 metrics: average dialogue-act Precision/Recall/F1, task Complete and Success rates, and average turns for successful dialogues and for all dialogues (lower is better). Note that shorter dialogues may reflect premature failures rather than efficiency, hence these metrics are interpreted jointly. Following common practice, we interpret success/complete jointly with turns to reflect both task achievement and efficiency.

[1] https://datasets.maluuba.com/Frames.

[2] https://github.com/thu-coai/ConvLab-2.

[3] We use ConvLab-2 for stable RL training; ConvLab-3 RL pipelines remain less stable in our setting (see repository issue #179 and #191 discussions).

3.4 Baselines

We compare VLK-RL with RL-based cross-domain policy baselines (PPO (Schulman et al., 2017), ACGOS (Cordier et al., 2022)), LLM-based dialogue models (GALAXY (He et al., 2022), GDP-Zero (Yu et al., 2023), TransferTOD (Zhang et al., 2024)), and a cross-domain DST baseline (CAPID (Dong et al., 2024)).

3.5 Main Results

3.5.1 Simulated Environments

Tab. 1 reports results on MultiWOZ 2.1 and Frames. Across both datasets, VLK-RL achieves the strongest overall performance, with consistent improvements in Complete and Success and fewer average turns, indicating that it completes more user goals and reduces redundant interactions. The improvement is notably larger on Frames, where subtasks are more tightly coupled and feasibility constraints often determine whether a plan is globally valid, supporting our claim that explicitly modeling and enforcing cross-domain constraints is central to robust long-horizon decision making. RL-only baselines (PPO, ACGOS) remain limited, especially on Frames, suggesting that long-horizon exploration and credit assignment alone are insufficient to recover missing feasibility knowledge from raw interaction. LLM-based approaches (GALAXY, GDP-Zero, TransferTOD) achieve stronger dialogue-act metrics, but their end-to-end success lags behind VLK-RL, consistent with the brittleness of unverified or weakly grounded single-pass reasoning that can accumulate errors across turns. CAPID improves cross-domain state tracking and narrows the gap, yet it primarily strengthens states with information grounded in dialogue and ontology, so implicit feasibility constraints that require commonsense or temporal inference can still be under-specified, limiting downstream policy success. In contrast, VLK-RL explicitly verifies candidate constraints and grounds them into ontology-aligned slot–value representations that are executable against the database, yielding a more stable state interface for policy learning and mitigating drift over long dialogues. Among VLK-RL variants, stronger backbones tend to perform better, and we observe that Qwen-based variants achieve the highest scores in our setting; we attribute this mainly to more consistent constraint-to-slot grounding and fewer cross-turn inconsistencies under our prompts, and

Table 1: Performance Comparison of different dialogue agents on MultiWOZ 2.1 and Frames. Top performance per metric is highlighted with bold and a background. All differences statistically significant ($p < 0.05$).

Dataset	Model	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)
MultiWOZ 2.1	PPO	0.4273	0.4997	0.7121	0.4912	0.3815	13.21	20.94
	ACGOS	0.4857	0.5328	0.7316	0.5524	0.4521	14.10	19.82
	GALAXY	0.5236	0.5789	0.7654	0.6031	0.5216	13.50	21.56
	GDP-Zero	0.5819	0.6357	0.7928	0.7025	0.6024	15.31	22.22
	TransferTOD	0.5648	0.6105	0.7817	0.6716	0.5823	14.83	20.40
	CAPID	0.5763	0.6152	0.7871	0.6820	0.5875	14.00	20.00
	VLK-RL (GPT-4o-mini)	0.6354	0.6886	0.8237	0.7619	0.6812	13.00	17.91
	VLK-RL (Qwen-7B)	0.6483	0.7027	0.8354	0.7815	0.6958	12.80	17.62
	VLK-RL (Qwen-14B)	**0.6628**	**0.7182**	**0.8429**	**0.8006**	**0.7214**	**12.51**	**17.35**
Frames	PPO	0.4231	0.4802	0.6721	0.6031	0.4235	15.34	18.56
	ACGOS	0.4852	0.5331	0.7309	0.5236	0.4315	13.50	18.80
	GALAXY	0.5186	0.5734	0.7598	0.5810	0.4975	12.90	20.10
	GDP-Zero	0.5819	0.6327	0.7578	0.7215	0.5890	14.12	17.84
	TransferTOD	0.5596	0.6087	0.7803	0.6507	0.5678	13.80	19.60
	CAPID	0.5712	0.6154	0.7883	0.6701	0.5782	14.00	20.00
	VLK-RL (GPT-4o-mini)	0.6302	0.6875	0.8204	0.7512	0.6789	12.30	17.80
	VLK-RL (Qwen-7B)	0.6437	0.6989	0.8325	0.7701	0.6903	12.10	17.50
	VLK-RL (Qwen-14B)	**0.7034**	**0.7512**	**0.8357**	**0.8063**	**0.7239**	**12.65**	**15.91**

we provide additional analysis and cross-model cross-examination results in App. D.

3.5.2 Human Environments

To assess robustness in realistic interactions, we conduct human evaluation with 30 annotators on MultiWOZ 2.1 and Frames. For each dataset, we use ConvLab-2's `goal_generator` to sample user goals, and each annotator completes three dialogues by interacting with anonymized agents whose identities are shuffled and hidden to avoid bias. After each dialogue, the annotator provides two judgments: task completion as SUCCESS RATE (SR), and overall dialogue quality as HUMAN RATING (HR) on a 1–5 Likert scale (higher is better), considering fluency, naturalness, and redundancy. Annotators are allowed to terminate dialogues early when interactions become incoherent or unproductive; such cases are counted as failures for SR, and the corresponding HR is still recorded to reflect perceived quality under failure.

Tab. 2 shows that VLK-RL consistently achieves higher SR and HR than all baselines on both datasets, with larger gains on Frames where cross-task feasibility constraints are more salient. Compared with RL-only agents, VLK-RL reduces redundant clarification turns and improves cross-domain coordination; compared with LLM-based and DST-based baselines, verifying and grounding constraints at the state level helps maintain feasibility over long horizons, improving both task validity and human-perceived coherence. These human results corroborate the simulated evaluation and support the central claim that constraint verification and ontology-aligned grounding provide a reliable interface for long-horizon policy learning.

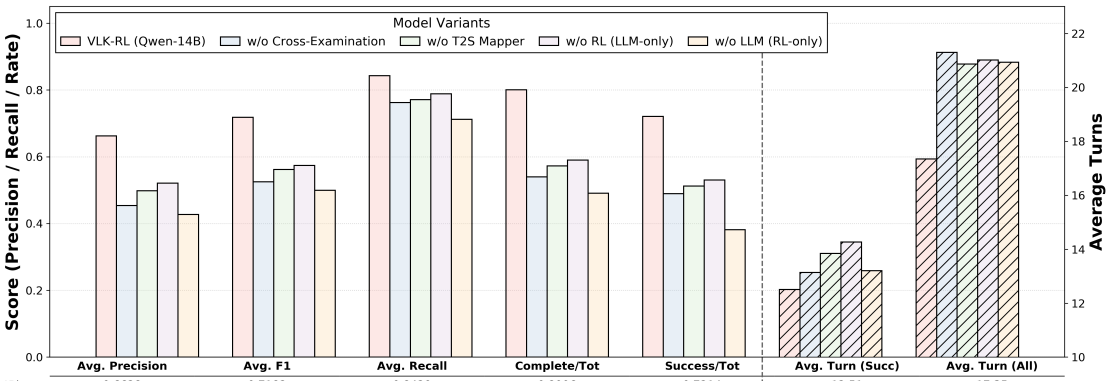
Table 2: Human evaluation results of different agents.

Dataset	Model	SR	HR
MultiWOZ 2.1	PPO	0.2850	2.21
	ACGOS	0.3410	2.35
	GALAXY	0.3875	2.46
	GDP-Zero	0.4120	2.61
	TransferTOD	0.3982	2.58
	CAPID	0.4056	2.63
	VLK-RL (GPT-4o-mini)	0.4713	3.04
	VLK-RL (Qwen-7B)	0.4936	3.07
	VLK-RL (Qwen-14B)	**0.5124**	**3.18**
Frames	PPO	0.2618	2.08
	ACGOS	0.3185	2.22
	GALAXY	0.3612	2.34
	GDP-Zero	0.3896	2.49
	TransferTOD	0.3741	2.45
	CAPID	0.3927	2.51
	VLK-RL (GPT-4o-mini)	0.4589	3.11
	VLK-RL (Qwen-7B)	0.4823	3.16
	VLK-RL (Qwen-14B)	**0.5057**	**3.32**

3.6 Ablation Study

We analyze the contribution of each component in VLK-RL using the best-performing variant, VLK-RL (Qwen-14B), and report results in Fig. 4. We consider four ablations. (1) **w/o Cross-Examination** removes the dual-role verification and uses the respondent LLM outputs directly as constraints. (2) **w/o T2S Mapper** removes ontology-aligned grounding and instead encodes verified textual constraints into dense representations that are concatenated with the dialogue state, without slot normalization. (3) **w/o RL (LLM-only)** removes RL optimization and prompts the LLM to select an action from the predefined action set (prompt in App. B). (4) **w/o LLM (RL-only)** removes both LLM-based modules, reducing the system to PPO on the original state.

Removing any component substantially degrades performance, indicating that VLK-RL's gains arise from the interaction between verification, state grounding, and long-horizon optimization rather



	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)
VLK-RL (Qwen-14B)	0.6628	0.7182	0.8429	0.8006	0.7214	12.51	17.35
w/o Cross-Examination	0.4545	0.5252	0.7624	0.5400	0.4900	13.14	21.30
w/o T2S Mapper	0.4987	0.5623	0.7716	0.5732	0.5124	13.85	20.87
w/o RL (LLM-only)	0.5214	0.5742	0.7885	0.5901	0.5308	14.27	21.02
w/o LLM (RL-only)	0.4273	0.4997	0.7121	0.4912	0.3815	13.21	20.94

Figure 4: Ablation study on VLK-RL (Qwen-14B) to analyze the individual contributions of its three core modules.

than from any single dominant module. In particular, **w/o Cross-Examination** sharply reduces success and precision, showing that hallucinations and cross-turn inconsistencies in single-pass constraint inference corrupt the state signal used for policy learning, motivating explicit verification. **w/o T2S Mapper** also leads to a pronounced drop, despite using verified constraints, because free-form text fails to provide an executable and stable interface to the ontology and database; without slot-level grounding and value normalization, constraints may be non-fillable, schema-inconsistent, or ambiguous across domains, weakening their utility for downstream RL. The **LLM-only** variant achieves moderate act-level quality but lags in end-to-end task success and turns, reflecting that myopic action selection without RL struggles to maintain goal consistency and recover from earlier errors over long horizons. Finally, the **RL-only** variant performs worst overall, confirming that long-horizon optimization alone cannot compensate for missing implicit feasibility constraints in the state.

To make these failure modes concrete, App. G quantifies grounding errors and shows that a substantial fraction of verified constraints cannot be executed reliably against the ontology and database without slot-value mapping and normalization, explaining the drop for w/o T2S Mapper. App. F analyzes verification errors and reports that cross-examination filters hallucinated constraints, leading to higher-precision state augmentation and more stable long-horizon policy learning.

3.7 Constraint-oriented Analyses

We further analyze whether agents satisfy cross-domain feasibility constraints by combining explicit and implicit failure statistics with qualitative inspection. Following prior TOD analyses, we distinguish *explicit* constraints that are directly stated or trivially implied by user utterances from *implicit* constraints that require commonsense, temporal, or inter-domain reasoning. Concretely, we sample 50 user goals from each dataset and run each system to obtain 250 dialogues per dataset. For each failed dialogue, we attribute the failure to missing explicit or implicit constraints and report, among failures, the fraction attributable to each type.

Fig. 5 summarizes the results. RL-only baselines exhibit high failure rates on both constraint types, reflecting their reliance on incomplete states and the difficulty of enforcing feasibility purely through long-horizon optimization. LLM-based baselines reduce explicit failures but remain unstable on implicit constraints, consistent with reasoning errors and cross-turn inconsistencies accumulating in long interactions. CAPID achieves relatively low explicit failure rates due to stronger ontology-grounded state tracking, yet implicit failures remain high because such constraints are rarely grounded in surface text and not explicitly represented at the state level. In contrast, VLK-RL substantially reduces failures for both explicit and implicit constraints across datasets, indicating that verifying and grounding LLM-inferred constraints into slot–value states improves the system's ability to recognize and enforce feasibility throughout the dialogue trajectory. We observe the largest relative improvements on Frames, which contains tightly coupled subtasks and makes feasibility constraints more consequential to end-to-end success.

To complement the aggregate statistics, we provide qualitative case studies in App. H. These exam-

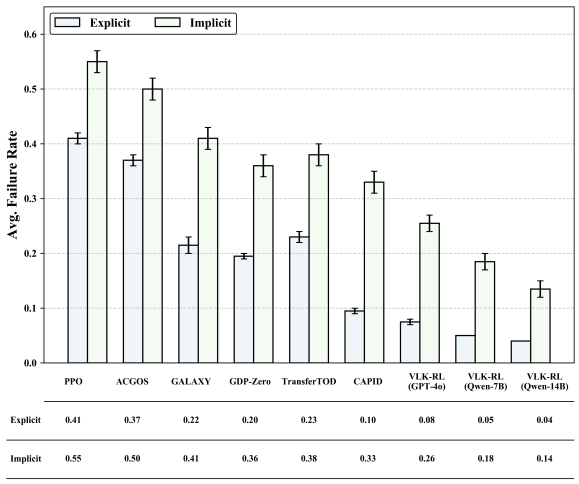


Figure 5: Constraint-related failure breakdown on MultiWOZ 2.1 and Frames. For failed dialogues, we report the fraction whose failure is attributable to missing *explicit* constraints or missing *implicit* constraints. Per-dataset breakdowns are provided in App. E.

ples illustrate how verified constraints are surfaced during cross-examination, how they are grounded into ontology-aligned slot–value updates, and how the resulting constraint-aware state leads to fewer redundant clarification turns and fewer globally invalid decisions, especially in cases where the user does not state inter-domain dependencies explicitly.

4 Related Work

Task-oriented dialogue systems in cross-domain settings require managing long-horizon dependencies, where decisions in one domain impose feasibility constraints on others. Traditional cross-domain dialogue state tracking (DST) methods, such as TRADE (Wu et al., 2019) and TripPy (Heck et al., 2020), focus on extracting slot-value representations to model user goals. While effective for explicit constraints, these methods struggle to handle implicit constraints, which require commonsense or temporal reasoning (e.g., hotel check-in must follow flight arrival). Recent work has extended DST with large language models (LLMs) to improve generalization across domains, allowing LLMs to infer both explicit and implicit constraints from the dialogue context (Dong et al., 2024; Feng et al., 2023). However, LLM-generated outputs often lack grounding and verifiability, making them challenging to integrate with structured databases and decision-making pipelines. Another line of research addresses cross-domain decision-making through reinforcement learning (RL). Composite-task and hierarchical RL frameworks aim to decompose complex goals into smaller sub-tasks and improve decision-making across multiple domains (Peng et al., 2017; Rohmatillah and Chien, 2023; Kwan et al., 2023). These approaches typically assume that the dialogue state is accurate and complete. However, when cross-domain constraints are missing or misrepresented, RL policies become brittle, especially in long-horizon tasks where decisions in one domain influence others.

Recent hybrid LLM-RL frameworks have combined the strengths of LLMs and RL to bridge high-level reasoning and sequential decision-making. These methods use LLMs for planning, reward assignment, or task scheduling, while RL is responsible for decision-making (Alon and David, 2025; Wei et al., 2025; Cao et al., 2025). However, many of these approaches tightly couple reasoning and action selection, which can lead to instability, especially in long-horizon tasks. In contrast, our approach decouples reasoning from control by using LLMs for constraint extraction, which are grounded into normalized states for RL optimization. This modular design allows LLM reasoning to inform RL policy without entangling the two processes, enhancing robustness and stability in cross-domain decision-making.

5 Conclusion

We present VLK-RL, a hybrid framework for cross-domain task-oriented dialogue that connects LLM reasoning and RL decision making through verified, state-level constraint grounding. VLK-RL verifies LLM-inferred explicit and implicit feasibility constraints via dual-role cross-examination and grounds the verified knowledge into ontology-aligned slot–value states, providing a stable and executable interface for downstream policy optimization. Experiments on MultiWOZ 2.1 and Frames show that VLK-RL consistently improves cross-domain generalization, policy robustness, and dialogue efficiency over strong RL-only, LLM-only, and cross-domain DST baselines in both simulated and human evaluations. Overall, VLK-RL offers a modular and principled approach to integrating verified LLM knowledge into long-horizon dialogue policy learning.

Limitation

VLK-RL has several limitations. First, it relies on pre-trained LLMs for constraint inference and verification, which introduces additional latency and

computation and may limit deployment in resource-constrained settings. Second, cross-examination improves reliability but is not guaranteed to be complete, and it may fail to surface subtle or rare commonsense dependencies, especially in complex contexts with ambiguous user goals. Third, the text-to-slot grounding depends on the coverage and granularity of the underlying ontology and database; constraints that cannot be cleanly expressed in the predefined schema may be partially lost. Future work may mitigate these issues by distilling the reasoning and verification modules into smaller models, incorporating retrieval or external knowledge to support rare constraints, and extending grounding mechanisms to handle richer constraint forms beyond slot–value representations.

References

Yoav Alon and Cristina David. 2025. Integrating large language models and reinforcement learning for non-linear reasoning. In *FSE 2025*.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2025. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Trans. Neural Networks Learn. Syst.*, 36(6):9737–9757.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12621–12640. Association for Computational Linguistics.

Thibault Cordier, Tanguy Urvoy, Fabrice Lefèvre, and Lina M Rojas-Barahona. 2022. Graph neural network policies and imitation learning for multi-domain task-oriented dialogues. *arXiv preprint arXiv:2210.05252*.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024. 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, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8527–8540. Association for Computational Linguistics.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 739–755. Association for Computational Linguistics.

Cristina Fernández, Izaskun Fernández, and Cristina Aceta. 2025. Lamia: An llm approach for task-oriented dialogue systems in industry 5.0. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 205–214.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, and 1 others. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 10749–10757.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 35–44. Association for Computational Linguistics.

Wai-Chung Kwan, Hong-Ru Wang, Hui-Min Wang, and Kam-Fai Wong. 2023. A survey on recent advances and challenges in reinforcement learning methods for task-oriented dialogue policy learning. *Machine Intelligence Research*, 20(3):318–334.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul A. Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021. Zero-shot dialogue state tracking via cross-task transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7890–7900. Association for Computational Linguistics.

Vinh Quang Nguyen, Nguyen Quang Chieu, Hoang Viet Pham, and Khac-Hoai Nam Bui. 2025. Spec-tod: A specialized instruction-tuned llm framework for efficient task-oriented dialogue systems. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 133–145.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2231–2240. Association for Computational Linguistics.

Libo Qin, Wenbo Pan, Qiguang Chen, Lizi Liao, Zhou Yu, Yue Zhang, Wanxiang Che, and Min Li. 2023. End-to-end task-oriented dialogue: A survey of tasks, methods, and future directions. In *Proceedings of the 2023 Conference on Empirical Methods*

in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 5925–5941. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Mahdin Rohmatillah and Jen-Tzung Chien. 2023. Hierarchical reinforcement learning with guidance for multi-domain dialogue policy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:748–761.

Mahdin Rohmatillah, Jen-Tzung Chien, and 1 others. 2023. Advances and challenges in multi-domain task-oriented dialogue policy optimization. *APSIPA Transactions on Signal and Information Processing*, 12(1).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*.

Yuan Wei, Xiaohan Shan, and Jianmin Li. 2025. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. *arXiv*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819. Association for Computational Linguistics.

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. *CoRR*, abs/2402.18013.

Xiao Yu, Maximillian Chen, and Zhou Yu. 2023. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning. *arXiv preprint arXiv:2305.13660*.

Ming Zhang, Caishuang Huang, Yilong Wu, Shichun Liu, Huiyuan Zheng, Yurui Dong, Yujiong Shen, Shihan Dou, Jun Zhao, Junjie Ye, and 1 others. 2024. Transfertod: A generalizable chinese multi-domain task-oriented dialogue system with transfer capabilities. *arXiv preprint arXiv:2407.21693*.

Yangyang Zhao, Mehdi Dastani, Jinchuan Long, Zhenyu Wang, and Shihan Wang. 2024. Rescue conversations from dead-ends: Efficient exploration for task-oriented dialogue policy optimization. *Trans. Assoc. Comput. Linguistics*, 12:1578–1596.

Zhenyou Zhou, Zhibin Liu, Zhaoan Dong, and Yuhan Liu. 2024. Model discrepancy policy optimization for task-oriented dialogue. *Computer Speech & Language*, 87:101636.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020a. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Trans. Assoc. Comput. Linguistics*, 8:281–295.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020b. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. *arXiv preprint arXiv:2002.04793*.

A Respondent LLM Prompt Design

The Tab. 3 presents the step-by-step CoT prompt designed for the Respondent LLM to infer both explicit and implicit constraints in multi-domain TOD tasks.

B Prompt for LLM-only Policy Optimization

In the **w/o RL (LLM-only)** ablation, the RL-based policy optimizer is removed and the LLM is directly prompted to select an action from the predefined action set. The Tab. 4 presents the exact prompt template used for Qwen-14B.

C Extendability to Different RL Backbones

VLK-RL enhances dialogue policies by augmenting the dialogue state with verified explicit and implicit cross-task constraints, while leaving the policy architecture, action space, and reward design unchanged. This design suggests that the framework should be compatible with different reinforcement learning optimizers. To empirically validate this claim, we replace the default PPO optimizer with alternative RL algorithms and evaluate performance on the MultiWOZ 2.1 benchmark under the same experimental settings. Specifically, we consider Deep Q-Networks (DQN) as a value-based method and vanilla Policy Gradient (PG) as a policy-based method. All models are trained using identical dialogue state representations and hyperparameter tuning procedures, differing only in the RL backbone.

Tab. 5 reports the performance of VLK-RL instantiated with different reinforcement learning backbones on the MultiWOZ 2.1 dataset. For both

Table 3: Respondent LLM CoT Prompt for Inferring Explicit and Implicit Constraints.

Step	Prompt Instruction
1	**System Role (rea_system)**: You are a helpful assistant. Focus only on the `belief_state` of the user status. Fill in blank slots based on the known information across domains, without adding extra slots. Provide a confidence coefficient between 0 and 1 indicating your certainty. Domains refer to top-level keys in `belief_state` (e.g., 'police', 'taxi', 'hotel', 'train').
2	**Main Prompt (rea_main)**: Examples for step-by-step reasoning. **Example 1:** User State: `{user_action: [], system_action: [], belief_state: {...}, request_state: {}, terminated: False, history: []}` Step 1: Identify relevant task domains from the user status. Step 2: Extract known slot information per domain. Step 3: Analyze relationships and infer potential slot values logically (explicit and implicit). Step 4: Assign a confidence coefficient to the inferred values. Step 5: Produce final output in the format: @{updated user status}@, confidence coefficient: 0.95. **Example 2:** Similar procedure with attention to local context inference and uncertainty handling. Confidence coefficient: 0.87. **Example 3:** Inference with high ambiguity due to multiple plausible scenarios; confidence coefficient: 0.65. **Example 4:** Extreme uncertainty in inference from limited information; confidence coefficient: 0.35.
3	**Instructions for Respondent LLM**: • Ensure output format is consistent with input, enclosed with '@' at start and end. • Include confidence coefficient ($0-1$) in the output, enclosed with '$'. • Do not add comments or extra slots, and do not modify `user_action`, `system_action`, `request_state`, `terminated`, `history`. • Focus solely on filling empty slot values.
4	**Question Template (question)**: Analyze and infer the information for the following user status: • Q: {user_status} • Perform step-by-step reasoning to infer missing slot values.

Table 4: Prompt design for the LLM-only setting (w/o RL). The LLM is required to select one action $a \in \mathcal{A}$ based on the dialogue history and database results.

System Instruction:
You are a task-oriented dialogue agent. Your goal is to select the next system action from the predefined action set $\mathcal{A}$ based on the current dialogue history and database state.
Action Set:
{ `inform_slot`, `request_slot`, `confirm_slot`, `book`, `goodbye`,... }
User Dialogue History:
Dialogue context C_t up to current turn t
Database Results:
Relevant slot–value information retrieved from DB
Task:
Based on the dialogue history and database results, select exactly one action from the action set.
Do not generate free text or explanations. Only output the action name.
Output Format:
Action = [selected action]

Table 5: Performance of VLK-RL with different RL backbones on MultiWOZ 2.1.

Model	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)
DQN	0.5396	0.6750	0.7592	0.7093	0.3124	15.60	21.00
VLK-RL (Qwen-14B + DQN)	0.6493	0.7050	0.8087	0.7791	0.6782	13.20	18.50
PG	0.5228	0.6511	0.7429	0.6947	0.2863	16.15	21.84
VLK-RL (Qwen-14B + PG)	0.6216	0.6887	0.7914	0.7579	0.6418	13.85	19.15

Table 6: Effect of Judge-Respondent model instantiation on VLK-RL performance on MultiWOZ 2.1.

Judge / Respondent	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)
Qwen-14B / Qwen-14B	**0.6628**	**0.7182**	**0.8429**	**0.8006**	**0.7214**	**12.51**	**17.35**
GPT-4o-mini / Qwen-14B	0.6481	0.7013	0.8315	0.7794	0.7026	12.84	17.68
Qwen-14B / GPT-4o-mini	0.6517	0.7059	0.8352	0.7831	0.7089	12.77	17.59

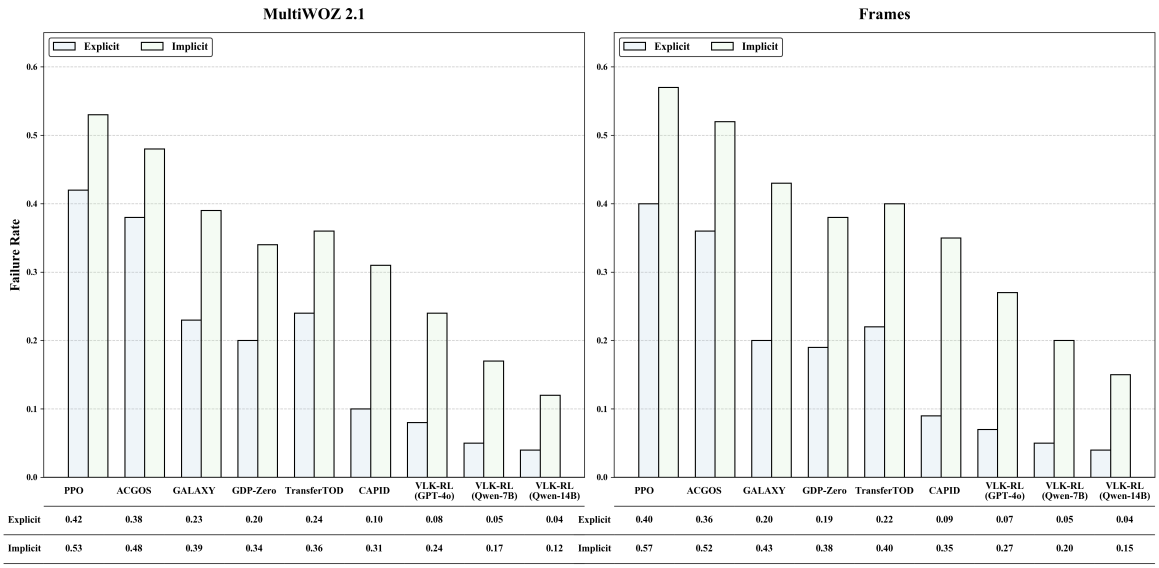


Figure 6: Failure rates for explicit and implicit constraints on MultiWOZ 2.1 and Frames separately.

value-based (DQN) and policy-based (PG) methods, incorporating VLK-RL consistently improves task success rate, dialogue completion rate, and state-level prediction metrics compared to their respective baselines. Notably, while absolute performance varies across RL algorithms, the relative gains introduced by VLK-RL remain stable. This indicates that the improvements primarily stem from enhanced constraint-aware state representations rather than optimizer-specific characteristics. In addition, VLK-RL reduces the average number of dialogue turns required for successful task completion, suggesting improved dialogue efficiency across different optimization paradigms.

D Analysis of Judge-Respondent Model Instantiation

This appendix investigates whether the effectiveness of the proposed dual-role cross-examination mechanism depends on using the same LLM for both the Judge and Respondent roles. While the main experiments instantiate both roles with a unified LLM, the core mechanism relies on role-specific prompting rather than architectural differences. Using the same LLM ensures a shared logical framework and consistent knowledge boundaries, which facilitates the detection of internal inconsistencies by reducing false positives introduced by divergent reasoning styles across models. Although hallucinations may still occur, they tend to manifest differently across role-specific generations, allowing the Judge to identify contradictions relative to its own reasoning trajectory. In contrast, cross-model instantiations may introduce additional noise due to heterogeneous reasoning preferences, leading to missed inconsistencies or incorrect rejections. To empirically validate this claim, we conduct ablation experiments by cross-assigning GPT-4o-mini and Qwen-14B to the Judge and Respondent roles, and compare them with the same-model configurations. Qwen-7B is not included as it consistently underperforms Qwen-14B in our main experiments and does not provide additional insight beyond model scale effects.

As shown in Tab. 6, VLK-RL remains effective under both same-model and cross-model instantiations, indicating that the proposed framework does not rely on a specific LLM pairing. However, same-model configurations consistently yield slightly higher success rates and more stable dialogue efficiency, supporting our choice of a unified LLM as the default setting. These results suggest that role-induced reasoning diversity, rather than model heterogeneity, is the primary driver of effective cross-examination.

E Constraint Failure Rates by Dataset

Fig. 6 provides a dataset-level breakdown of constraint failure patterns, reporting the failure rates of explicit and implicit constraints on MultiWOZ 2.1 and Frames separately to complement the aggregated analysis in the main paper.

F Importance of Dual-role LLM Cross-Examination

F.1 Quantitative Analysis of LLM Failure Modes

To justify the necessity of dual-role LLM cross-examination, we quantify hallucinations, inconsistencies, and constraint violations in MultiWOZ 2.1 dialogues. We sampled 200 user goals and manually labeled three major failure types:

- **T1: Hallucinations** (fabricated entities or attributes)
- **T2: Inconsistencies** (contradictions in responses or with prior turns)
- **T3: Constraint Violations** (ignoring user or system constraints)

The proportion of dialogues affected by each failure type is shown in Tab. 8. The results show that hallucinations, inconsistencies, and constraint violations are frequent and systemic in ToD tasks. Dual-role cross-examination significantly reduces all three failure categories, validating its necessity.

F.2 Qualitative Examples of Knowledge Failures

Tab. 7 illustrates representative dialogue failures before and after introducing dual-role cross-examination.

F.3 Alternative Designs for Dual-role Verification

To examine whether dual-role cross-examination can be replaced by simpler single-model heuristics, we design two *confidence-based gating* variants that rely on an LLM's self-reported confidence to filter its own outputs. Unlike VLK-RL, which verifies knowledge via role-based debate, these variants accept or reject inferred knowledge without

Table 7: Qualitative examples of LLM failures and the effect of cross-examination.

Failure Type	Before Mechanism	After Mechanism
T1: Hallucinations	User: "I'd like a moderately priced restaurant in the centre of town." System: "You can try 'King's Garden Bistro,' which is open from 8 am to 11 pm and serves Italian food."	User: "I'd like a moderately priced restaurant in the centre of town." System: "I recommend 'Cotto,' a moderately priced restaurant in the city centre that matches your request."
T2: Inconsistencies	User: "I need a cheap guesthouse in the north of the city." System: "I found a cheap guesthouse in the north called 'Acorn Guest House.'" User: "Can you remind me of the price range again?" System: "'Acorn Guest House' is in the moderate price range and located in the centre."	User: "I need a cheap guesthouse in the north of the city." System: "I found a cheap guesthouse in the north called 'Acorn Guest House.'" User: "Can you remind me of the price range again?" System: "'Acorn Guest House' is still in the cheap price range and located in the north, as mentioned before."
T3: Constraint Violations	User: "Please book a vegetarian restaurant in the centre that is open after 9 PM." System: "I booked 'The Garden' for you. It's vegetarian and in the centre, but it closes at 8:30 PM."	User: "Please book a vegetarian restaurant in the centre that is open after 9 PM." System: "'The Garden' closes at 8:30 PM and does not satisfy your time constraint. Instead, I booked 'Taste of Cambridge,' which is open until 10 PM."

Table 8: Proportion of dialogues affected by failure types with and without cross-examination.

Failure Type	Before Mechanism	After Mechanism
T1: Hallucinations	23.5%	6.5%
T2: Inconsistencies	22.0%	7.5%
T3: Constraint Violations	17.0%	5.0%

introducing an explicit Judge role. Specifically, we consider the following two alternatives:

- **Confidence-fixed ($\tau = 0.85$).** The LLM output is accepted only if its self-reported confidence exceeds a fixed threshold of 0.85; otherwise, the inferred knowledge is discarded.
- **Confidence-dynamic.** The acceptance threshold is adjusted during training according to validation performance. Formally, the threshold for epoch $e{+}1$ is updated as Eq. 2.

$$\tau^{(e+1)} = \begin{cases} \tau_0 & \text{if } e < T_{\text{th}} \\ \tau_0 + \alpha \cdot \max(0, \text{F1}^{(e)} - \text{F1}_{\text{th}}) & \text{otherwise} \end{cases} \quad (2)$$

where τ_0 denotes the initial threshold, $\text{F1}^{(e)}$ is the validation F1 score at epoch e, F1_{th} is the minimum target F1, and T_{th} marks the training stage after which dynamic adjustment begins. Fig. 7 illustrates the evolution of τ during training.

Tab. 9 reports the performance of these alternatives. Both confidence-based gating strategies outperform naive removal of cross-examination, indicating that filtering unreliable LLM outputs is beneficial. However, neither variant matches the

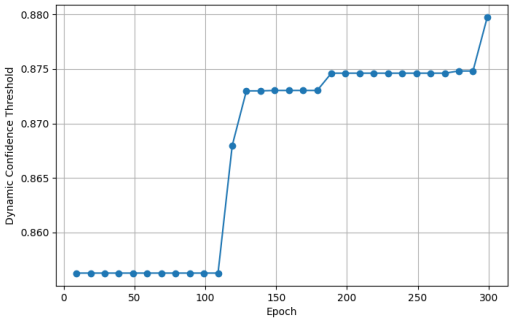


Figure 7: Dynamic changes in confidence threshold during training for VLK-RL (Qwen14b).

full VLK-RL framework. Although the dynamic threshold yields higher recall by gradually relaxing acceptance criteria, both gating strategies remain inferior to dual-role cross-examination. This gap arises because self-confidence estimates are systematically over-optimistic and cannot expose latent inconsistencies or constraint violations. In contrast, explicit role separation enables independent reasoning trajectories and more reliable detection of hallucinations, confirming the necessity of dual-role verification in VLK-RL.

G Importance of Ontology-aware Text-to-Slot Mapping

To simulate direct LLM output without a dedicated mapper, we designed stricter prompts forcing slot-value formatting, *Mapper-prompt with retries (kx)*. It denotes a prompt-only setting in which the LLM

Table 9: Performance of alternative dual-role verification designs.

Model	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)
VLK-RL (Qwen-14B)	0.6628	0.7182	0.8429	0.8006	0.7214	12.51	17.35
w/o Cross-Examination	0.4545	0.5252	0.7624	0.5400	0.4900	13.14	21.30
Confidence-fixed ($\tau = 0.85$)	0.6245	0.6784	0.8194	0.6100	0.6000	14.13	21.22
Confidence-dynamic	0.6728	0.7243	0.8642	0.7100	0.6700	14.53	21.16

Table 10: Ablation and prompt-based variants for the T2S mapper. Runtime is relative to VLK-RL (Qwen-14B).

Model	Avg. Precision	Avg. F1	Avg. Recall	Complete/Tot	Success/Tot	Avg. Turn (Succ)	Avg. Turn (All)	Runtime
VLK-RL (Qwen-14B)	0.6628	0.7182	0.8429	0.8006	0.7214	12.51	17.35	1x
w/o T2S Mapper	0.4987	0.5623	0.7716	0.5732	0.5124	13.85	20.87	1x
Mapper-prompt with retries (5x)	0.5887	0.6415	0.8016	0.6503	0.5812	13.94	20.85	1.2x
Mapper-prompt with retries (20x)	0.6402	0.7056	0.8005	0.7882	0.7105	14.35	19.02	8.4x

is instructed to directly output ontology-aligned slot-value pairs. After each generation attempt, the output is validated by a deterministic parser against the predefined ontology schema. If the output contains invalid slot names, malformed values, missing mandatory fields, or violates the expected key–value format, it is rejected and the model is re-prompted. This process repeats until either a valid structured output is obtained or the maximum number of retries k is reached. If all attempts fail, the output for that turn is discarded. We consider two retry budgets:

- **Mapper-prompt with retries (5x):** The LLM is allowed up to five regeneration attempts per dialogue turn. This setting balances computational cost and structural validity, but often discards semantically correct outputs due to formatting errors when the retry limit is reached.
- **Mapper-prompt with retries (20x):** The LLM is allowed up to twenty regeneration attempts per turn. This aggressive retry strategy substantially increases the probability of obtaining a valid structured output, at the expense of significantly higher latency and inference cost.

As shown in Tab. 10, increasing the retry budget from 5x to 20x consistently improves task success and completion rates, indicating that brute-force regeneration can partially recover structural correctness. However, this improvement comes with diminishing returns and substantial overhead: the 20x variant incurs an approximately $8\times$ increase in runtime compared to the original VLK-RL pipeline, rendering it impractical for real-world deployment. Moreover, both retry-based variants exhibit higher dialogue length and instability, reflecting the inherent variance of prompt-based structured generation.

In summary, the dual-role cross-examination and T2S mapper are complementary: the cross-examination ensures correctness and consistency of generated knowledge, while the T2S mapper guarantees structured, database-compatible states. Together, they provide high-quality input for RL optimization, leading to robust multi-turn dialogue performance.

H Case Study

To complement the quantitative results, we present a case study comparing PPO and VLK-RL in a multi-domain travel planning scenario. As shown in Tab. 11, the task involves coordinating transportation, attraction visits, and hotel booking under realistic user constraints.

The PPO agent struggles with implicit associations: although it retrieves the train information, it fails to leverage contextual cues to infer the attraction's location or connect the date with the hotel booking request. This results in redundant clarification questions (e.g., asking for the obvious location of Cambridge University Botanic Gardens) and longer dialogues. In contrast, VLK-RL successfully resolves both explicit and implicit constraints. It infers that the attraction is on Bateman Street and proactively associates the hotel booking with the same day (Wednesday), while explicitly confirming the user's room type request. These mechanisms reduce unnecessary turns and improve dialogue coherence. Overall, the case study confirms that VLK-RL enhances policy efficiency by grounding decisions in both explicit slot-value matches and implicit contextual reasoning, leading to smoother and more natural multi-domain dialogues.

I Low-resource Environments

We evaluate all models under an extreme low-resource setting on MultiWOZ 2.1, where agents

Table 11: Dialogue examples for a multi-domain travel planning task. Comparison between PPO and VLK-RL. Red text highlights implicit associations (inferred from dialogue context, such as dates and locations), while green text highlights explicit associations (direct responses to user constraints). (U = User, S = System).

PPO	**VLK-RL**
U0: Hi, I need to go to Cambridge from London on Wednesday morning. I'd like to go by myself. S1: Sure, I can help you with that. There is a train leaving from London Liverpool Street at 09:15 on Wednesday, heading to Cambridge. Would you like to book a ticket? U2: No need to book, just give me the train information. Also, I'm planning to visit the Cambridge University Botanic Gardens. S3: Okay, so what's the area of the Cambridge University Botanic Gardens? U4: Well, the area is definitely Bateman Street, hasn't it? S5: Okay, the area is Bateman Street. U6: Oh, by the way, I also need to book a hotel. I'm looking for a four-star hotel, and it should be on the east side of the scenic area. S7: Okay, for the hotel booking, what are the dates and how many people will it be for? Also, do you have any parking requirements? U8: The hotel is for Wednesday, for one person, and I don't need parking. ...	U0: Hi, I need to go to Cambridge from London on Wednesday morning. I'd like to go by myself. S1: Sure, I can help you with that. There is a train leaving from London Liverpool Street at 09:15 on Wednesday, heading to Cambridge. Would you like to book a ticket? U2: No need to book, just give me the train information. Also, I'm planning to visit the Cambridge University Botanic Gardens. S3: Alright, the area of the Cambridge University Botanic Gardens should be Bateman Street (implicit association). Do you need a hotel reservation for Wednesday (implicit association)? I can book a single room for you (explicit association). Also, will you require a parking space? U4: That would be great, please book a hotel for me. I won't need a parking space. Oh, by the way, it needs to be a four-star hotel on the east side of the scenic area. ...

Table 12: Performance under low-resource environments, where all models are trained from scratch without pre-trained weights, warm-start initialization, or external data. Each epoch only stores one dialogue for training.

Model	**Avg. Precision**	**Avg. F1**	**Avg. Recall**	**Complete/Tot**	**Success/Tot**	**Avg. Turn (Succ)**	**Avg. Turn (All)**
PPO	0.1456	0.0935	0.0737	0.0312	0.0000	–	28.45
ACGOS	0.1628	0.1154	0.0979	0.0421	0.0193	26.10	27.92
GALAXY	0.1875	0.1316	0.1164	0.0517	0.0326	24.83	27.35
GDP-Zero	0.2059	0.1587	0.1432	0.0562	0.0448	23.74	26.81
TransferTOD	0.1984	0.1492	0.1379	0.0541	0.0412	24.25	26.94
CAPID	0.2146	0.1659	0.1473	0.0618	0.0524	23.10	26.40
VLK-RL (GPT-4o-mini)	0.2651	0.2053	0.1827	0.0912	0.0890	21.35	25.10
VLK-RL (Qwen-7B)	0.2789	0.2176	0.1941	0.0983	0.1034	20.62	24.71
VLK-RL (Qwen-14B)	**0.2952**	**0.2351**	**0.2058**	**0.1105**	**0.1217**	**19.48**	**24.21**

are trained entirely from scratch without pre-trained weights, warm-start initialization, or external data. We focus on MultiWOZ in this setting because it provides sufficient domain diversity and interaction complexity to stress-test policy learning under sparse supervision, while remaining computationally feasible for repeated from-scratch training. In contrast, Frames contains fewer but more tightly coupled subtasks, where training instability dominates under such limited data, making reliable comparison difficult.

As shown in Tab. 12, all methods suffer substantial performance degradation. Classical RL baselines (PPO, ACGOS) exhibit near-zero success rates, indicating their inability to explore effective cross-domain strategies without prior knowledge. LLM-based approaches (GALAXY, GDP-Zero, TransferTOD) perform slightly better, leveraging in-context reasoning, but still struggle to generate executable and consistent slot–value pairs, leading to low task success and long dialogues. DST methods (CAPID) further improve over LLM-only baselines by enhancing cross-domain state tracking; however, without explicit modeling of commonsense-driven feasibility constraints, its gains remain limited under sparse supervision. In contrast, all VLK-RL variants achieve markedly higher success rates and shorter dialogues. Notably, VLK-RL (Qwen-14B) attains over 12% success, demonstrating that verified constraint extraction and structured slot grounding effectively compensate for the lack of training data. These results

highlight the robustness of VLK-RL: by explicitly grounding validated constraints at the state level, the framework enables more stable and data-efficient policy optimization even in severely low-resource environments.