Value-Guided Open-Loop MCTS for Proactive Task-Oriented Dialogue

Anonymous ACL submission

Abstract

Proactive task-oriented dialogue is essential for reliable real-world assistants, yet current LLM-based systems are largely reactive and struggle to recover from failed retrieval, ambiguous constraints, and cross-domain dependencies. This paper targets the core gap of enabling effective *lookahead* decision-making for proactive TOD under limited high-quality supervision and high-variance language rollouts. We propose SMCTS-TOD , which combines act-level open-loop planning with a fast learned success estimator to make lookahead practical: the planner searches over dialogue-act sequences while the Value-LLM provides low-latency, low-variance guidance and supports iterative refinement via self-distillation. Across MultiWOZ 2.0 and SGD, SMCTS-TOD improves interactive goal completion and robustness, achieving higher success-oriented metrics than strong prompting baselines. Human studies further indicate better dialogue-level usefulness and fewer unreasonable strategy choices. These results suggest that abstract planning paired with fast learned evaluation is a viable and verifiable path to more proactive and robust LLM-based TOD agents.

1 Introduction

Task-oriented dialogue (TOD) systems aim to help users accomplish concrete goals such as searching for flights or booking hotels (Young et al., 2013; Wen et al., 2017). Classic TOD systems are often built as modular pipelines (NLU, dialogue state tracking, policy, NLG), while end-to-end systems unify these components into a single model (Henderson et al., 2014; Hosseini-Asl et al., 2020a; Yang et al., 2021a). Recent large language models (LLMs) have improved instruction following and robustness, enabling zero-/few-shot TOD agents (Achiam et al., 2023). Despite these advances, most LLM-based TOD systems remain *reactive*: they respond to the current user request, use single-turn retrieval, and generate a response without explicit planning. In realistic settings, reactive behavior often fails to (i) recover from empty or low-quality retrieval results, (ii) proactively offer alternatives or related useful information, and (iii) plan multi-turn strategies across domains with dependencies.

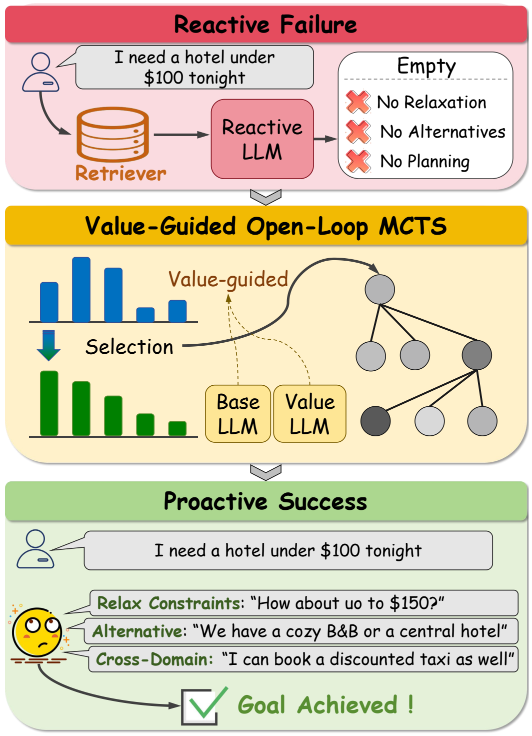

Figure 1: **Overview of SMCTS-TOD for proactive TOD. Top:** Reactive LLM agents rely on single-turn retrieval and often fail under empty/low-quality results. **Middle:** SMCTS-TOD performs *value-guided open-loop MCTS* in an act-level search space, combining a Base-LLM prior with a lightweight Value-LLM success estimator to guide selection. **Bottom:** The resulting policy leads to higher goal completion.

Proactive TOD requires anticipating the future: when the current query yields no matching entities, the system should strategically relax constraints, propose alternatives, verify ambiguous slots, or offer cross-domain suggestions (e.g., hotel options consistent with a train itinerary). This demands lookahead under uncertainty and explicit strategy control—capabilities that are not guaranteed by next-token generation alone (Wang et al., 2023).

Forward search methods such as Monte Carlo Tree Search (MCTS) have succeeded in sequential decision-making (Browne et al., 2012; Silver et al., 2017). However, applying MCTS to dialogue is non-trivial: storing simulated natural-language utterances inside the search tree yields high variance and compounding errors—one unrealistic simulated utterance can poison an entire subtree. Moreover, prompt-based rollout evaluation is expensive and noisy, making deep search impractical.

We propose SMCTS-TOD , a value-guided open-loop MCTS planner for proactive TOD (Fig. 1). Open-loop MCTS searches over dialogue act sequences rather than concrete utterance states, reducing sensitivity to stochastic realizations. To make search efficient and reliable, we train a lightweight supervised Value-LLM that estimates the probability of eventual goal achievement, enabling low-latency node evaluation and in-tree best-of-K rejection sampling.

Our contributions include: (**i**) We show that planning in an act-level open-loop space, paired with a learned success estimator, can mitigate rollout variance from natural language and make lookahead practical for LLM-based TOD. (**ii**) Our SMCTS-TOD systematically improves initiative-taking behaviors and remains effective when user feedback is limited, as evidenced by substantially smaller degradation under a less-cooperative simulator and lower human strategy edit rates. (**iii**) Across MultiWOZ 2.0 and SGD, SMCTS-TOD achieves strong improvements in task success and interaction-level metrics, while also improving human-judged dialogue-level quality.

2 Related Work

2.1 Task-Oriented Dialogue Systems

Task-oriented dialogue (TOD) has traditionally been framed as sequential decision-making under uncertainty. Classic systems adopt a pipeline architecture with separate modules for natural language understanding, dialogue state tracking (DST), policy learning, and natural language generation, often formalized as POMDPs (Young et al., 2013; Henderson et al., 2013). While modularity eases debugging and component upgrades, pipelines can suffer from error propagation and limited robustness to domain shift.

End-to-end TOD models reduce hand-engineering by jointly learning multiple components in a single neural model, improving optimization and response quality (Wen et al., 2017; Hosseini-Asl et al., 2020b; Yang et al., 2021b). Recent work further integrates explicit belief states, external knowledge, or schema information to scale across domains (Rastogi et al., 2020; He et al., 2022; Sun et al., 2023). With the rise of instruction-tuned LLMs, LLM-based TOD agents have become increasingly competitive in low-resource and zero-shot settings by leveraging world knowledge, tool use, and flexible prompting (Zhang et al., 2023; Xu et al., 2024; Dong et al., 2025). However, many LLM-based TOD systems remain reactive: they produce the next response conditioned on the current context and limited tool calls, without explicit lookahead planning over multi-turn strategies.

2.2 Proactive Dialogue and Evaluation

Proactive dialogue systems take initiative to improve long-horizon utility beyond answering the current utterance (Deng et al., 2023, 2025). In task/information-seeking settings, proactivity is often instantiated as *clarification* and *preference elicitation*, which reduce ambiguity and shrink the feasible space (Aliannejadi et al., 2019; Radlinski et al., 2019). However, standard TOD metrics (e.g., Inform/Success on MultiWOZ) mainly measure goal correctness and can under-reward exploratory proactive behaviors (Budzianowski et al., 2018). Interactive evaluation with user simulators is thus common, but LLM simulators may be overly cooperative, inflating success (Sekulić et al., 2024; Dong et al., 2025). We address this by using two simulator personas (cooperative vs. less-cooperative) and a strategy-level human metric (Strategy Edit Rate).

2.3 Dialogue Policy Learning and Search-Based Planning

Dialogue policies have been learned via supervised action imitation and RL with value functions (Young et al., 2013; Su et al., 2017), and via model-based planning to alleviate sparse rewards (e.g., Deep Dyna-Q/D3Q) (Peng et al., 2018). Decision-time lookahead further refines action selection using tree search (MCTS) (Browne et al., 2012). In the LLM era, prompt-based MCTS uses LLMs as priors and rollout judges (Yu et al., 2023). Compared to prompt-only search, SMCTS-TOD in-

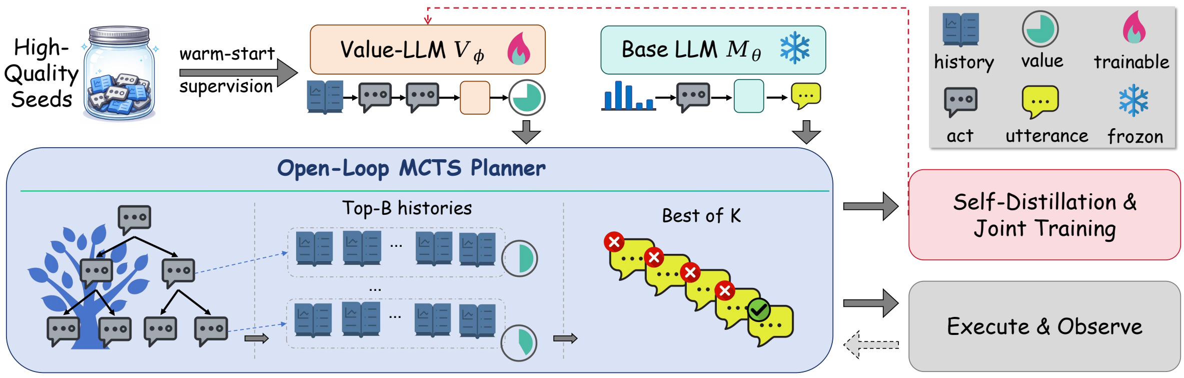

Figure 2: **SMCTS-TOD pipeline.** Seeds warm-start the Value-LLM V_ϕ. At each turn, the frozen base LLM M_θ provides act priors and rollout utterances, while V_ϕ supplies fast success estimates. The open-loop MCTS searches over act sequences, maintains a top-B history cache per node with best-of-K filtering and PUCT backup, outputs (a^*, u^*) for execution (no re-generation), and distills MCTS traces back to update V_ϕ.

troduces a supervised Value-LLM for fast low-variance evaluation, integrates best-of-K filtering with a bounded rollout cache, and closes the loop with self-distillation.

3 Method: SMCTS-TOD

3.1 Overview

We view proactive TOD as history-based sequential decision-making. At each turn, the system chooses an act $a_{t+1} \in \mathcal{A}$ conditioned on the history h_t (and tool outputs o_t), with the goal of maximizing expected discounted return:

$$\pi^*(h_t) = \arg\max_{a \in \mathcal{A}} \mathbb{E}\left[\sum_{k \geq 0} \gamma^k r_{t+1+k} \,\middle|\, h_t,\ a_{t+1} = a\right], \tag{1}$$

where r_t is a sparse task-progress reward and $\gamma \in [0, 1)$.

SMCTS-TOD upgrades a reactive LLM-based TOD agent into a proactive planner by performing value-guided open-loop MCTS over dialogue acts. At each turn, a base LLM M_θ provides (i) a stochastic *act prior* and (ii) utterance-level rollout realizations; a supervised Value-LLM V_ϕ supplies fast, low-variance estimates of eventual goal achievement; an open-loop MCTS planner combines these signals to search over future act sequences while maintaining a bounded cache of multiple plausible utterance trajectories per node. Crucially, SMCTS-TOD includes an *iterative self-distillation flywheel*: MCTS traces serve as improved supervision to continually refine V_ϕ, enabling stronger planning under limited high-quality demonstrations.

A dialogue history at turn t is

$$h_t \triangleq \{(a_1, u_1^{\text{sys}}, u_1^{\text{usr}}), \ldots, (a_t, u_t^{\text{sys}}, u_t^{\text{usr}})\}, \tag{2}$$

where $a_t \in \mathcal{A}$ is a system dialogue act from a finite act set $\mathcal{A}$, and $u_t^{\text{sys}}, u_t^{\text{usr}}$ are system/user utterances. We denote tool outputs (DB results, retrieval candidates, API responses) by o_t. We write history concatenation as $h \oplus (\cdot)$. Our goal at each turn is to choose an act a (and a surface response) that maximizes eventual success probability. Fig. 2 provides an end-to-end overview of SMCTS-TOD .

3.2 Open-Loop Search State over Dialogue-Act Sequences

Closed-loop dialogue planning stores simulated utterances in tree nodes. Since utterances are stochastic and diverse, an implausible simulated response can distort downstream rollouts and induce compounding errors. SMCTS-TOD instead uses open-loop planning over act sequences, allowing the tree to accumulate statistics over strategies rather than fragile surface forms.

An open-loop tree node is represented by an act sequence

$$s_i^{\text{tr}} \triangleq (a_1, \ldots, a_i), \qquad a_j \in \mathcal{A}. \tag{3}$$

Here s_i^{tr} is the *tree state*. The utterance-level instantiation is handled separately via a bounded cache attached to each node:

$$\mathcal{H}(s_i^{\text{tr}}) \triangleq \{h_i^{(1)}, \ldots, h_i^{(B)}\}, \qquad B \in \mathbb{N}, \tag{4}$$

where each $h_i^{(b)}$ is a concrete dialogue history consistent with executing s_i^{tr} from the root context. B is the cache budget (top-B histories retained per node).

3.3 Base LLM Prior and Rollout Generation

Search needs (i) a prior over promising next acts to guide exploration and (ii) a stochastic world model to generate plausible rollouts. LLMs naturally provide both by conditional generation.

Given a history h (and tool outputs o), the base LLM M_θ proposes next acts. We approximate its act prior by Monte Carlo sampling:

$$\hat{a}^{(j)} \sim M_\theta(\cdot \mid h, o), \quad j = 1, \ldots, m,$$
$$P_\theta(a \mid h) \triangleq \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}\left[\hat{a}^{(j)} = a\right]. \tag{5}$$

where m is the number of samples and $\mathbf{1}[\cdot]$ is the indicator function.

Conditioned on a chosen act a, the same model generates a candidate system utterance:

$$\hat{u}^{\text{sys}} \sim M_\theta(\cdot \mid h, a, o). \tag{6}$$

In rollout simulation, a user model produces

$$\hat{u}^{\text{usr}} \sim U_\psi(\cdot \mid h \oplus (a, \hat{u}^{\text{sys}}), o), \tag{7}$$

and the updated history becomes

$$h' \;=\; h \oplus (a, \hat{u}^{\text{sys}}, \hat{u}^{\text{usr}}). \tag{8}$$

where U_ψ can be instantiated by an LLM-based simulator; the planner treats it as a black-box transition sampler.

3.4 Supervised Value-LLM for Fast Success Estimation

Prompt-based judging of multi-turn rollouts is expensive and high-variance. SMCTS-TOD trains a lightweight Value-LLM to provide fast, low-variance success estimates that guide PUCT (Rosin, 2011) selection and enable best-of-K filtering.

We train V_ϕ to predict an action-conditioned success probability:

$$V_\phi(h, a) \in [0, 1],$$
$$V_\phi(h, a) \approx \Pr(\text{Success} \mid h, \text{take } a). \tag{9}$$

where h is the current history, a is a candidate next act, and "Success" denotes final goal completion.

To evaluate a leaf history h, we define an evaluation policy derived from value scores:

$$\pi_{\text{eval}}(a \mid h) \triangleq \frac{\exp(V_\phi(h, a)/\tau_{\text{eval}})}{\sum_{b \in \mathcal{A}} \exp(V_\phi(h, b)/\tau_{\text{eval}})}, \tag{10}$$

and the corresponding state value

$$\bar{V}_\phi(h) \triangleq \sum_{a \in \mathcal{A}} \pi_{\text{eval}}(a \mid h)\, V_\phi(h, a). \tag{11}$$

where $\tau_{\text{eval}} > 0$ controls how close π_{eval} is to an argmax; $\bar{V}_\phi(h)$ avoids optimistic bias from $\max_a V_\phi(h, a)$ and remains low-variance.

Training targets under sparse rewards. For a training dialogue ending at turn T with final label $y \in \{0, 1\}$, we assign each intermediate turn t a discounted target $\tilde{y}_t \triangleq \gamma^{(T-t)} y, \gamma \in [0, 1)$, and train via binary cross-entropy:

$$\mathcal{L}_{\text{value}} = -\sum_t \Big[\tilde{y}_t \log V_\phi(h_t, a_t) + (1 - \tilde{y}_t) \log\big(1 - V_\phi(h_t, a_t)\big)\Big]. \tag{12}$$

where (h_t, a_t) is an observed (history, next-act) pair from data; $V_\phi(h_t, a_t)$ is the predicted success probability.

Value-induced act prior. We convert action values into a soft prior:

$$P_\phi(a \mid h) \triangleq \frac{\exp(V_\phi(h, a)/\tau_v)}{\sum_{b \in \mathcal{A}} \exp(V_\phi(h, b)/\tau_v)}, \quad \tau_v > 0. \tag{13}$$

where τ_v controls sharpness; smaller τ_v yields a more peaked value prior.

3.5 Value-Guided Open-Loop MCTS

We combine (i) P_θ (broad, general LLM prior) and (ii) V_ϕ (fast, task-aligned evaluation) to plan under uncertainty. PUCT integrates priors into MCTS and provides principled exploration–exploitation trade-offs. For each open-loop node s^{tr} and action a, we maintain $N(s^{\text{tr}}, a) \in \mathbb{N}$, $W(s^{\text{tr}}, a) \in \mathbb{R}$, and $Q(s^{\text{tr}}, a) \triangleq W(s^{\text{tr}}, a)/\max\{1, N(s^{\text{tr}}, a)\}$. Here, N is the visit count, W is the accumulated backup value, and Q is the mean value.

At a simulation history h corresponding to node s^{tr}, we define

$$P(a \mid h) \triangleq \lambda P_\theta(a \mid h) + (1 - \lambda) P_\phi(a \mid h) \tag{14}$$

where $\lambda \in [0, 1]$ trades off the base LLM prior (5) and the value prior (13).

Selection (PUCT). We select the next act by maximizing

$$a^\star = \arg\max_{a\in\mathcal{A}} \Big[Q(s^{\mathrm{tr}}, a) + c_p\, \pi_{\mathrm{PUCT}}(a) \Big], \quad (15)$$

where $\pi_{\mathrm{PUCT}}(a) = P(a \mid h)\, \frac{\sqrt{\sum_b N(s^{\mathrm{tr}},b)}}{1+N(s^{\mathrm{tr}},a)}$, $c_p > 0$ is the exploration constant, $P(a \mid h)$ is the action prior, and $N(s^{\mathrm{tr}}, a)$ is the visit count.

Expansion. Upon reaching an unexpanded node, we propose a candidate act set $\mathcal{A}_{\mathrm{cand}} \subseteq \mathcal{A}$ from M_θ and initialize $N(s^{\mathrm{tr}}, a) \leftarrow 0, W(s^{\mathrm{tr}}, a) \leftarrow 0, Q(s^{\mathrm{tr}}, a) \leftarrow Q_0, \forall a \in \mathcal{A}_{\mathrm{cand}}$, where Q_0 is a constant initialization (e.g., neutral or mildly optimistic).

Evaluation and backup. Given the leaf realization history h_{leaf}, we evaluate $v \triangleq \bar{V}_\phi(h_{\mathrm{leaf}}) \in [0, 1]$ and backpropagate along the selected path $\mathcal{P}$:

$$\begin{aligned} N(s^{\mathrm{tr}}, a) &\leftarrow N(s^{\mathrm{tr}}, a) + 1, \\ W(s^{\mathrm{tr}}, a) &\leftarrow W(s^{\mathrm{tr}}, a) + v, \\ Q(s^{\mathrm{tr}}, a) &\leftarrow \frac{W(s^{\mathrm{tr}}, a)}{N(s^{\mathrm{tr}}, a)}. \end{aligned} \quad (16)$$

Root policy and act choice. After S simulations, the visit counts induce a posterior policy

$$\pi_{\mathrm{MCTS}}(a \mid h_t) \triangleq \frac{N(s_0^{\mathrm{tr}}, a)^{1/\tau}}{\sum_{b\in\mathcal{A}} N(s_0^{\mathrm{tr}}, b)^{1/\tau}}, \qquad \tau > 0, \quad (17)$$

and we select the next act as

$$a^* \triangleq \arg\max_{a\in\mathcal{A}} N(s_0^{\mathrm{tr}}, a). \quad (18)$$

where τ is a temperature; smaller τ yields greedier policies.

3.6 Rollout Cache, Best-of-K Filtering, and Response Selection

Open-loop nodes don't store utterance states, yet MCTS simulation still requires concrete dialogue histories to (i) query the base LLM for candidate realizations and (ii) evaluate success likelihood. Every time the search visits a node s^{tr}, it samples a history h from the node cache $\mathcal{H}(s^{\mathrm{tr}})$, and then extends it by a best-of-K filtered rollout step.

Cache sampling. Each node s^{tr} maintains a bounded cache $\mathcal{H}(s^{\mathrm{tr}})$. We sample a concrete history from this cache using a value-weighted distribution:

$$p_{\mathcal{H}}(h \mid s^{\mathrm{tr}}) \triangleq \frac{\exp\big(\bar{V}_\phi(h)/\tau_{\mathcal{H}}\big)}{\sum_{h'\in\mathcal{H}(s^{\mathrm{tr}})} \exp\big(\bar{V}_\phi(h')/\tau_{\mathcal{H}}\big)}, \quad (19)$$

$$h \sim p_{\mathcal{H}}(\cdot \mid s^{\mathrm{tr}}). \quad (20)$$

where $\bar{V}_\phi(h) \in [0, 1]$ is the state value from (11), and $\tau_{\mathcal{H}} > 0$ controls how strongly sampling prefers higher-value realizations (smaller $\tau_{\mathcal{H}}$ is greedier).

The root cache is initialized as $\mathcal{H}(s_0^{\mathrm{tr}}) = \{h_t\}$. Whenever a child node $s^{\mathrm{tr}} \oplus a$ is created/visited, we immediately insert at least one realization into $\mathcal{H}(s^{\mathrm{tr}} \oplus a)$ via the best-of-K step below, ensuring (20) is always well-defined.

Best-of-K filtered rollout step. Given a sampled history h at node s^{tr} and a selected act $a \in \mathcal{A}$, we generate K candidate system utterances

$$\hat{u}_k^{\mathrm{sys}} \sim M_\theta(\cdot \mid h, a, o), \qquad k = 1, \ldots, K, \quad (21)$$

simulate the user response for each candidate

$$\hat{u}_k^{\mathrm{usr}} \sim U_\psi\big(\cdot \mid h \oplus (a, \hat{u}_k^{\mathrm{sys}}), o\big), \quad (22)$$

and obtain K next-history candidates

$$h_k^+ \triangleq h \oplus \big(a, \hat{u}_k^{\mathrm{sys}}, \hat{u}_k^{\mathrm{usr}}\big). \quad (23)$$

We then select the locally best continuation by maximizing the downstream state value:

$$k^\star \triangleq \arg\max_{k\in\{1,\ldots,K\}} \bar{V}_\phi(h_k^+), \quad (24)$$

$$h^+ \triangleq h_{k^\star}^+. \quad (25)$$

Best-of-K filtering is directly aligned with maximizing eventual success probability.

Cache write-back (top-B by value). Let the child node be $s^{\mathrm{tr}+} \triangleq s^{\mathrm{tr}} \oplus a$. We insert the selected realization h^+ and retain only the best B elements under $\bar{V}_\phi$:

$$\mathcal{H}(s^{\mathrm{tr}+}) \leftarrow \mathrm{TopB}\Big(\mathcal{H}(s^{\mathrm{tr}+}) \cup \{h^+\},\ B;\ \bar{V}_\phi\Big), \quad (26)$$

where $\mathrm{TopB}(\mathcal{S}, B; f)$ returns the subset of $\mathcal{S}$ of size B with the highest scores under $f(\cdot)$. This keeps memory bounded and concentrates future sampling on high-quality realizations.

Algorithm 1 SMCTS-TOD : value-guided open-loop MCTS planning for one turn

Require: history h_t, tool outputs o_t, act set $\mathcal{A}$; base LLM M_θ, user model U_ψ, Value-LLM V_ϕ; simulations S, cache budget B, best-of-K size K; hyperparameters $c_p, \lambda, \tau_v, \tau_{\text{eval}}, \tau_{\mathcal{H}}, \tau$.

$s_0^{\text{tr}} \leftarrow ()$; $\mathcal{H}(s_0^{\text{tr}}) \leftarrow \{h_t\}$; initialize tree stats
for $s = 1$ to S **do**
  $s^{\text{tr}} \leftarrow s_0^{\text{tr}}$; $\mathcal{P} \leftarrow []$ ▷ selected edges
  while s^{tr} expanded and not terminal **do**
    sample $h \sim p_{\mathcal{H}}(\cdot \mid s^{\text{tr}})$ via (19)–(20)
    compute $P_\theta(\cdot|h)$, $P_\phi(\cdot|h)$ by (5),(13)
    $P(\cdot|h) \leftarrow \lambda P_\theta(\cdot|h) + (1-\lambda) P_\phi(\cdot|h)$
    select a using (15), sample $\hat{u}_k^{\text{sys}}$ by (21), simulate $\hat{u}_k^{\text{usr}}$ by (22)
    form h_k^+ by (23), choose h^+ by (25)
    $s^{\text{tr}+} \leftarrow s^{\text{tr}} \oplus a$; write-back $\mathcal{H}(s^{\text{tr}+})$
    append (s^{tr}, a) to $\mathcal{P}$; $s^{\text{tr}} \leftarrow s^{\text{tr}+}$
  end while
  if s^{tr} not expanded **then**
    propose $\mathcal{A}_{\text{cand}}$ from M_θ; initialize stats
  end if
  evaluate $v \leftarrow \bar{V}_\phi(h)$ using (10)–(11)
  for all $(s_i^{\text{tr}}, a_{i+1}) \in \mathcal{P}$ **do**
    update (N, W, Q) via (16)
  end for
end for
compute $\pi_{\text{MCTS}}(\cdot|h_t)$ via (17); choose $a^* \leftarrow \arg\max_a N(s_0^{\text{tr}}, a)$
choose $h^\star$ via (27)
return $(a^*, u^{\text{sys}}$ stored in $h^\star)$

Response selection at execution (no re-generation). After search selects the root act $a^* = \arg\max_a N(s_0^{\text{tr}}, a)$ (§3.5), we choose the highest-value cached realization at the corresponding child node:

$$h^\star \triangleq \arg \max_{h \in \mathcal{H}(s_0^{\text{tr}} \oplus a^*)} \bar{V}_\phi(h), \qquad (27)$$

and execute the system utterance stored in $h^\star$ as the next response. This makes execution deterministic given the search outcome and avoids an additional decoding call to M_θ.

Algorithm 1 summarizes the full inference-time planning procedure described in § 3.5 and §3.6.

3.7 Iterative Self-Distillation Flywheel

With limited high-quality seeds, V_ϕ can be miscalibrated on rare trajectories that MCTS discovers during proactive exploration. SMCTS-TOD therefore uses MCTS to generate improved targets and distills them back into V_ϕ, expanding supervision beyond the seed set and progressively improving planning quality.

For each planning trace, we collect tuples $(h, a, Q_{\text{MCTS}}(h, a))$, where $Q_{\text{MCTS}}(h, a) \triangleq Q\big(s^{\text{tr}}(h), a\big)$, and $s^{\text{tr}}(h)$ denotes the open-loop node corresponding to history h in the trace. Here $Q(\cdot, \cdot)$ is the backed-up mean value. We update V_ϕ by regressing predictions to MCTS targets:

$$\mathcal{L}_{\text{distill}} = \mathbb{E}_{(h,a)\sim\mathcal{D}_{\text{MCTS}}} \Big[\big(V_\phi(h, a) - Q_{\text{MCTS}}(h, a) \big)^2 \Big]. \qquad (28)$$

where $\mathcal{D}_{\text{MCTS}}$ is the dataset of traces collected during planning.

We combine seed supervision and distillation:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{value}} + \beta\, \mathcal{L}_{\text{distill}}, \qquad \beta \geq 0, \qquad (29)$$

where β controls the distillation strength and $\mathcal{L}_{\text{value}}$ is defined in (12). The training process of SMCTS-TOD is shown in Alg.2 in §A.7.

4 Experiments

4.1 Experimental Setup

Protocols. We evaluate SMCTS-TOD under two complementary protocols: (i) **end-to-end generation**, where the model generates the next system utterance from the dialogue history h_t (and tool outputs o_t when applicable), and (ii) **interactive evaluation**, where the system converses with an LLM-based user simulator until success or a turn limit, probing proactive recovery (verification, alternative offers, cross-domain planning). To reduce the known optimism of overly cooperative simulators, we instantiate two personas (cooperative vs. less-cooperative) and report results under both (details in §A.5 and §A.1).

Datasets. We conduct experiments on **MultiWOZ 2.0** (Budzianowski et al., 2018) and **SGD** (Rastogi et al., 2020). We compare against (a) *fully supervised* TOD systems (SimpleTOD (Hosseini-Asl et al., 2020b), UBAR (Yang et al., 2021b), GALAXY (He et al., 2022), Mars (Sun et al., 2023), TOATOD (Bang et al., 2023)) and (b) *prompting-based* LLM TOD systems (SGP-TOD (Zhang et al., 2023), AutoTOD (Xu et al., 2024), ProTOD (Dong et al., 2025)). Full baseline configurations are in §A.3.

Type	Model	Inform↑	Success↑	BLEU↑	Comb↑	CBE↑	#Uni↑	#Tri↑
Supervised	SimpleTOD (Hosseini-Asl et al., 2020b)	84.4	70.1	15.0	92.3	–	–	–
	UBAR (Yang et al., 2021b)	83.4	70.3	17.6	94.5	2.10	478	5238
	GALAXY (He et al., 2022)	85.4	75.7	19.6	100.2	1.75	295	2275
	Mars (Sun et al., 2023)	88.9	78.0	19.9	103.4	1.65	288	2264
	TOATOD (Bang et al., 2023)	90.0	79.8	17.04	101.94	–	–	–
Prompting	SGP-TOD (Zhang et al., 2023)	83.9	69.9	9.1	86.0	–	–	–
	AutoTOD (Xu et al., 2024)	87.2	82.8	9.3	94.3	2.62	1722	10188
	PROTOD (Dong et al., 2025)	91.7	83.3	8.9	96.4	**3.26**	1951	14345
Ours	SMCTS-TOD	**94.0**	**85.2**	**20.1**	**109.7**	3.24	**2104**	**15214**

Table 1: MultiWOZ 2.0 end-to-end evaluation (history-only input). ↑ indicates higher is better. We highlight the top-1/top-2/top-3 with dark-to-light shading.

Metrics. We follow standard MultiWOZ evaluation (*Inform*, *Success*, *BLEU*, *Combined*) and additionally report diversity statistics (#Uni, #Tri, CBE). Interactive evaluation reports *Inform*, *Success*, *Book*, and a simulator combined score. We further run *human evaluation* on 100 MultiWOZ test dialogues with 3 annotators (turn-level and dialogue-level ratings) and a *Strategy Edit Rate* to quantify strategy reliability. Definitions and rubrics are in §A.4 and §A.6. Implementation details are deferred to §A.7.

4.2 Results on MultiWOZ 2.0

End-to-end generation. Table 1 reports standard end-to-end results on MultiWOZ 2.0. SMCTS-TOD achieves the best **Inform** and **Success** among all compared methods, indicating strong entity grounding and goal satisfaction. Compared to prompting-based LLM baselines, SMCTS-TOD also yields substantially higher BLEU, reflecting improved alignment with in-domain realizations enabled by our supervised components and search-time response selection.

Interaction evaluation with two user simulators. Table 2 reports goal completion under proactive vs non-proactive simulators. LLM-based TOD agents substantially outperform fully supervised TOD systems in interactive success, suggesting that classic supervised TOD models may be less robust when deployed in open-ended interactive settings. Across both simulators, SMCTS-TOD achieves the best overall performance. Notably, performance degrades less under the non-proactive (less-cooperative) simulator, indicating that value-guided planning is more robust when the user provides fewer corrective signals and is less tolerant of irrelevant turns.

Human evaluation. Table 2 also reports turn- and dialogue-level ratings. SMCTS-TOD improves dialogue-level coherence, informativeness, and usefulness over the prompting baseline (AutoTOD) and matches or exceeds PROTODon most dimensions. While turn-level fluency is comparable across LLM-based systems, SMCTS-TOD yields higher dialogue-level quality, consistent with improved proactive planning across turns.

Strategy-level analysis. Fig. 3 reports action-category frequency and edit rate (ER). Compared to PROTOD, SMCTS-TOD increases the frequency of cross-domain offers and reduces edit rates across proactive categories, indicating that the chosen strategies are more often judged reasonable by humans. This provides evidence that gains are not merely due to being "more proactive" in frequency, but due to *higher-quality* proactive decisions.

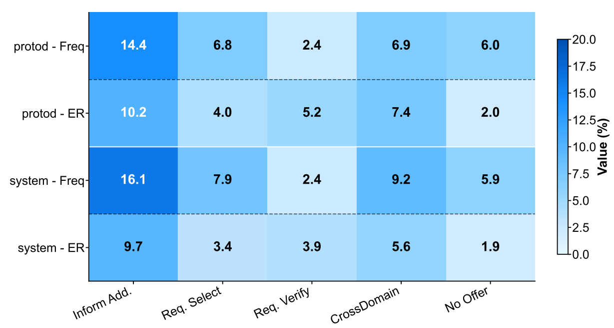

Figure 3: Strategy-level human evaluation results on MultiWOZ. Each cell shows Frequency(%)/Edit Rate(%), where lower Edit Rate is better.

4.3 Results on SGD

Table 3 reports results on SGD under the non-proactive simulator. SMCTS-TOD improves **Inform** and **Success** over AutoTOD and PROTODand achieves stronger diversity statistics. We

Model	Proactive Sim.				Non-Proactive Sim.				Turn-Level (Human)				Dialogue-Level (Human)			
	Inform↑	Succ↑	Book↑	Comb↑	Inform↑	Succ↑	Book↑	Comb↑	Und↑	Rea↑	Rel↑	Flu↑	Coh↑	Info↑	Use↑	Pro↑
TOATOD (Bang et al., 2023)	41.8	34.4	–	29.5	28.4	26.0	–	20.7	1.55	1.60	1.50	1.65	1.55	1.45	1.40	1.48
AutoTOD (Xu et al., 2024)	80.3	65.2	81.4	76.6	61.5	50.7	55.2	57.2	1.75	1.73	1.78	1.85	1.70	1.69	1.72	1.68
ProTOD (Dong et al., 2025)	89.5	80.4	87.0	86.6	85.7	76.5	82.6	82.6	1.82	1.75	1.79	**1.86**	1.78	1.82	1.79	**1.81**
SMCTS-TOD (Ours)	**93.2**	**83.1**	**89.3**	**89.7**	**88.3**	**78.5**	**87.1**	**85.6**	**1.92**	**1.84**	**1.87**	1.85	**1.91**	**1.86**	**1.82**	1.80

Table 2: MultiWOZ 2.0 interaction results (two user simulators) and human evaluation (100 dialogues).

Model	Inform	Success	CBE	#Uni	#Tri
SimpleTOD	12.7	9.8	2.01	573	3011
AutoTOD	45.1	23.0	2.81	1792	12263
ProTOD	50.4	24.9	3.26	2021	15149
SMCTS-TOD	**60.3**	**27.9**	**3.89**	**2318**	**16289**

Table 3: SGD interaction evaluation with the non-proactive user simulator.

Depth d	Open-loop (Ours)			Closed-loop		
	Incons.%↓	Contr.%↓	$\mathrm{Var}[\bar{V}_\phi]$ ↓	Incons.%↓	Contr.%↓	$\mathrm{Var}[\bar{V}_\phi]$ ↓
1	0.32	0.13	0.51	0.44	0.25	0.74
2	0.27	0.06	0.37	0.37	0.19	0.53
4	0.13	0.05	0.18	0.23	0.15	0.24
6	0.12	0.04	0.15	0.21	0.14	0.27
8	0.08	0.03	0.11	0.17	0.15	0.22

Table 4: Open-loop stability diagnostics results. At each rollout depth d, we measure (i) inconsistency rate (Incons.%), (ii) contradiction/self-contradiction rate (Contr.%), and (iii) the variance of value estimates $\mathrm{Var}[\bar{V}_\phi]$ from repeated simulations starting at the same node.

attribute these gains to the schema-driven, multi-constraint nature of SGD goals: users often request multiple attributes simultaneously, which increases ambiguity and the need for proactive verification and alternative exploration. Value-guided open-loop search helps prioritize high-value act sequences (e.g., verify → relax constraints → offer candidates) early in the dialogue.

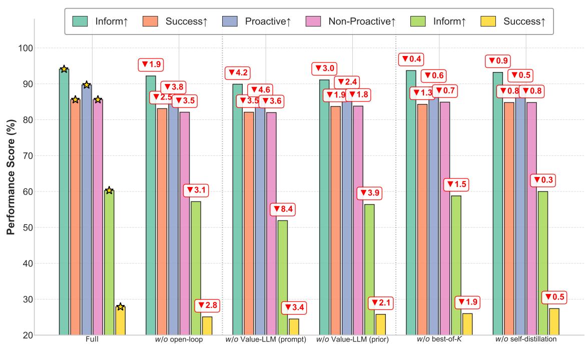

Figure 4: Ablation study on MultiWOZ 2.0 and SGD.

4.4 Ablation analysis.

Fig. 4 shows that removing open-loop planning consistently degrades performance, indicating that act-sequence search (rather than utterance-state search) is crucial for robust lookahead under stochastic language rollouts. Replacing the learned Value-LLM with prompt-based judging yields the largest drop—especially on SGD—suggesting that practical planning needs low-latency, low-variance value estimates. Using only the base-LLM prior is stronger than prompt judging but still lags the full model, highlighting the added benefit of task-aligned value guidance beyond a generic prior. Best-of-K filtering further improves results by pruning low-quality local generations before they propagate through rollouts, and self-distillation consistently outperforms the seed-only value model, validating the MCTS-driven supervision flywheel.

Open-loop stability diagnostics. Table 4 shows that open-loop planning is consistently more stable than closed-loop planning across rollout depths. At shallow depths ($d = 1$–2), open-loop already reduces both inconsistency and contradiction rates, and yields a noticeably lower variance in value estimates. As depth increases, the stability gap persists: open-loop maintains lower inconsistency/contradiction and substantially smaller $\mathrm{Var}[\bar{V}_\phi]$, indicating reduced sensitivity to stochastic and potentially erroneous utterance realizations. Overall, these diagnostics support our design choice of searching over act sequences while marginalizing utterance-level randomness via cached realizations.

5 Conclusion

We studied how to move LLM-based task-oriented dialogue from reactive response generation to proactive decision-making with reliable lookahead under sparse supervision and stochastic language rollouts. We introduced SMCTS-TOD , an act-level planning framework guided by a learned success estimator, and evaluated it with both less-biased simulator protocols and strategy-level human assessment. Empirically, SMCTS-TOD consistently improves goal completion and robustness.

Limitations

Despite the gains in task completion and proactivity, SMCTS-TOD still faces sparse rewards: success is typically observed only at the end of a dialogue, so training the Value-LLM with discounted labels (or pseudo-label signals) can introduce biased credit assignment for intermediate turns. Moreover, SMCTS-TOD may suffer from out-of-distribution (OOD) trajectories: search can reach dialogue paths not covered by the limited high-quality data, where the Value-LLM may become miscalibrated and yield high-variance or low-confidence estimates that misguide planning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6583–6591.

Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. 2025. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45.

Wenjie Dong, Sirong Chen, and Yan Yang. 2025. Protod: Proactive task-oriented dialogue system based on large language model. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9147–9164.

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(10), pages 10749–10757.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. Dialog state tracking challenge 2 & 3. *Dialogues with Soc. Robot*, pages 1–22.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 292–299.

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

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

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192.

Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 353–360.

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(05), pages 8689–8696.

Christopher D. Rosin. 2011. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230.

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

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and 1 others. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Modeling context & state representations with contrastive learning for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11139–11160.

Jian Wang, Yi Cheng, Dongding Lin, Chak Leong, and Wenjie Li. 2023. Target-oriented proactive dialogue systems with personalization: Problem formulation and dataset curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1143.

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

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

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021a. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35(16), pages 14230–14238.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021b. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35(16), pages 14230–14238.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Xiao Yu, Maximillian Chen, and Zhou Yu. 2023. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7101–7125.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023. Sgp-tod: Building task bots effortlessly via schema-guided llm prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13348–13369.

Appendix

A Experimental Details

A.1 Evaluation Protocols

End-to-end generation. We evaluate the *end-to-end* setting where the system produces the next response given only the dialogue context. Concretely, the input includes the dialogue history h_t (Eq. (2)) and tool outputs o_t when tool calls are enabled, and the output is the next system utterance u_{t+1}^{sys}. This setting measures context understanding, implicit action choice, and response realization under a single unified interface.

Interactive evaluation. We evaluate goal completion via multi-turn interaction between the system and a user simulator until: (i) success is achieved, (ii) a maximum turn budget $T_{\max}$ is reached, or (iii) the simulator terminates early (more common for the less-cooperative persona). This setting is designed to stress proactive behaviors (verification, alternative exploration, cross-domain offers) that are not reliably captured by single-turn generation metrics.

A.2 Datasets

MultiWOZ 2.0. MultiWOZ 2.0 is a multi-domain human–human dialogue dataset covering seven domains (restaurant, attraction, train, hotel, taxi, police, hospital) (Budzianowski et al., 2018). We follow the standard split and evaluation scripts used in prior work.

SGD. SGD is a schema-guided TOD dataset spanning many services (Rastogi et al., 2020). Because SGD does not provide a live database, we implement schema-driven database/API calls from provided snapshots and schemas, and we construct user goals from the annotated dialogue goal representations.

A.3 Baselines

Fully supervised TOD baselines. We include representative supervised TOD systems trained with full in-domain data: SimpleTOD (Hosseini-Asl et al., 2020b), UBAR (Yang et al., 2021b), GALAXY (He et al., 2022), Mars (Sun et al., 2023), and TOATOD (Bang et al., 2023). We follow the official or commonly used open-source implementations and report scores produced by their standard evaluation pipelines.

Prompting-based LLM baselines. We compare against prompting-based LLM TOD agents that rely on instruction prompting plus tool invocation: SGP-TOD (Zhang et al., 2023), AutoTOD (Xu et al., 2024), and ProTOD (Dong et al., 2025). We follow their recommended prompts and tool interfaces whenever available, and standardize the evaluation environment to ensure comparability: all baselines use the same backend (MultiWOZ DB / SGD schema API), the same dialogue context and tool-output formatting, and the same interaction constraints (turn budget and tool-call limits).

A.4 Metrics

MultiWOZ end-to-end metrics. We report Inform, Success, BLEU, and Combined following MultiWOZ conventions. Combined is defined as:

$$\text{Comb} \triangleq \text{BLEU} + \frac{1}{2}\big(\text{Inform} + \text{Success}\big). \quad (30)$$

(Reported Combined is taken from the official evaluation script; minor deviations from Eq. (30) may occur due to rounding and script details.)

Diversity metrics. We report: (i) #Uni: number of distinct unigrams in generated system utterances, (ii) #Tri: number of distinct trigrams, (iii) conditional bigram entropy (CBE). Let $p(w_i \mid w_{i-1})$ be the empirical conditional probability estimated from generated text. We compute:

$$\text{CBE} \triangleq -\sum_{(w_{i-1},w_i)} p(w_{i-1}, w_i)\,\log p(w_i \mid w_{i-1}). \quad (31)$$

Interactive metrics. We report Inform, Success, and Book (booking completion when applicable). We also report a simulator combined score:

$$\text{Comb}_{\text{sim}} \triangleq \frac{1}{2}\text{Inform} + \frac{1}{4}\big(\text{Success} + \text{Book}\big). \quad (32)$$

A.5 LLM-Based User Simulators

We implement user simulators using an instruction-following LLM (GPT-4 in our implementation). Each simulator is provided with a fixed user goal (and constraints) and interacts with the system, producing the next user utterance conditioned on the dialogue history.

Cooperative persona (“Proactive Simulator”). This simulator is patient and collaborative: it answers clarification questions, follows reasonable suggestions, and accepts cross-domain offers when relevant. It approximates an upper-bound interactive setting.

Less-cooperative persona ("Non-Proactive Simulator"). This simulator is less patient: it rejects irrelevant offers more frequently, provides fewer unsolicited clarifications, and may terminate early if the dialogue does not progress. It approximates more realistic interactions that require earlier proactive planning.

Prompts and constraints. We provide the full role prompts, termination rules, and maximum turn budgets in Appendix A.8. This includes: (i) how the goal is presented, (ii) refusal/acceptance criteria, (iii) patience/early-stop behavior.

A.6 Human Evaluation

We randomly sample 100 MultiWOZ test dialogues. Three graduate annotators rate model outputs on a $[0, 2]$ scale.

Turn-level criteria. **Und**: understandability of the system utterance; **Rel**: relevance to dialogue context; **Flu**: fluency/grammaticality; **Rea**: reasonableness (no contradictions or nonsensical actions).

Dialogue-level criteria. **Coh**: coherence across turns; **Inf**: informativeness (useful details/options); **Use**: overall usefulness toward accomplishing the goal; **Pro**: proactive helpfulness (verification, suggesting relevant alternatives, cross-domain assistance).

Strategy-level reliability (Frequency and Strategy Edit Rate). Annotators also inspect the selected next-act category and edit it if unreasonable. For category $c \in \mathcal{C}$, we report frequency and edit rate:

$$\mathrm{Freq}(c) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}[a_t \in c], \tag{33}$$

$$\mathrm{ER}(c) \triangleq \frac{\sum_{t=1}^{T} \mathbf{1}[a_t \in c] \cdot \mathbf{1}[\mathrm{edited}_t]}{\sum_{t=1}^{T} \mathbf{1}[a_t \in c]}. \tag{34}$$

A.7 Implementation Details

Models. Unless otherwise noted, we instantiate the base dialogue model M_θ with an instruction-tuned 7B LLM (`Qwen2.5-7B-Instruct`) and keep it frozen during our method. The Value-LLM V_ϕ is a separate lightweight model (`Qwen2.5-1.5B-Instruct`) that predicts $V_\phi(h, a) \in [0, 1]$ from the serialized history and a candidate next act, and is only used for fast evaluation and prior shaping. We train V_ϕ with LoRA (rank r=16, α=32) using binary cross-entropy on the seed set $\mathcal{D}_{\text{seed}}$ and MSE on MCTS targets from $\mathcal{D}_{\text{MCTS}}$. $\mathcal{D}_{\text{seed}}$ contains 2,000 high-quality training dialogues (1,000 from MultiWOZ 2.0 and 1,000 from SGD), yielding $\sim$25K turn-level tuples (h_t, a_t, y) after discounting. We run R=3 self-distillation rounds; in each round we collect $\sim$100K tuples (h, a, Q_{MCTS}) by executing the planner on additional training goals, and mix them with $\mathcal{D}_{\text{seed}}$ (Eq. (29)) with β=1.

Planning hyperparameters. We use S=64 simulations per turn with a maximum search depth of 4, best-of-K rollout filtering with K=4, and a per-node cache budget of B=8. For PUCT we set c_p=1.5 and mix priors with λ=0.5 (Eq. (14)). We set τ_v=0.25 for the value-induced prior (Eq. (13)), τ_{eval}=0.5 for π_{eval} (Eq. (10)), $\tau_{\mathcal{H}}$=0.5 for cache sampling (Eq. (19)), and τ=0.2 for the root visit-count policy (Eq. (17)). Interactive rollouts terminate on success, explicit user termination, or a maximum turn budget of $T_{\max}$=15 (MultiWOZ) / 20 (SGD); we allow at most two tool calls per system turn (one retrieval/DB call plus an optional booking call when applicable).

Compute. We implement all models with HuggingFace Transformers and run decoding with vLLM. V_ϕ training uses bf16 on a single NVIDIA A100-80GB GPU with batch size 256 (gradient accumulation 4), AdamW (lr 2×10^{-4}), and 3 epochs per distillation round. For M_θ generation we use temperature 0.7, top-p 0.9, and a 256-token response cap. On an A100-80GB GPU, V_ϕ evaluation adds negligible overhead (batched scoring over all acts), and the end-to-end planning latency averages 3–5 seconds per turn under the above $S/K/B$ settings (excluding user-simulator API latency in interactive evaluation).

The training process of SMCTS-TOD is shown in Algorithm 2.

A.8 User Simulator Prompts

Goal format and observations. For both personas, we provide a fixed `User Goals` block (natural-language goal description) and the `Current conversation` transcript. The simulator only observes the dialogue text (user and assistant utterances) and does *not* see any internal tool outputs or hidden states. The simulator ends the dialogue by outputting the special token `<END>` when

Algorithm 2 SMCTS-TOD : training flywheel for Value-LLM via MCTS self-distillation

Require: seed set $\mathcal{D}_{\text{seed}}$, base LLM M_θ, user model U_ψ; initialize V_ϕ; iterations R

1: **for** $r = 1$ to R **do**
2: update V_ϕ on $\mathcal{D}_{\text{seed}}$ by minimizing $\mathcal{L}_{\text{value}}$ in (12)
3: run Alg. 1 on training environments to collect $\mathcal{D}_{\text{MCTS}} = \{(h, a, Q_{\text{MCTS}})\}$
4: update V_ϕ by minimizing $\mathcal{L}_{\text{total}}$ in (29) on $\mathcal{D}_{\text{seed}} \cup \mathcal{D}_{\text{MCTS}}$
5: **end for**
6: **return** trained V_ϕ

the goals are achieved, when it decides to terminate early (less-cooperative persona), or when the maximum turn budget is reached ($T_{\max} = 15$ for MultiWOZ and $T_{\max} = 20$ for SGD).

Cooperative persona ("Proactive Simulator").

Proactive user simulator prompt (cooperative persona)

```
You are a collaborative and patient user
    interacting with an Assistant to
    complete some tasks. You should
    carefully read and understand the User
    Goals below, then talk with the AI
    Assistant and gradually express the
    intents in the goals. Your purpose is to
     achieve the goals as much as possible.
Note that the Assistant is not perfect. It
    may make various mistakes, including
    ignoring the user's requests, executing
    the wrong instructions, forgetting early
     conversation content, etc. The user you
     play should remind him to correct when
    you find that the AI assistant made a
    mistake, and complete the task as much
    as possible.
Important:
1. The expression of your needs should
    follow the order provided by the User
    Goal, and avoid expressing too much at
    once.
2. You are simulating the User, not the
    Assistant.
3. Do not provide information or ask
    questions outside of the User Goals.
4. End the conversation with "<END>" when
    you achieved the goals.

User Goals: {
"""
You are looking for a restaurant. The
    restaurant should be in the west and
    should serve kosher food.
The restaurant should be in the expensive
    price range.
If there is no such restaurant, how about
    one that serves indian food.
Make sure you get address, phone number, and
     postcode.
"""
}
The conversation you have completed so far:
    {
"""
User: I'm looking for a restaurant that
    serves kosher food in the west side.
AI Assistant: I'm sorry, we don't have any
    Kosher restaurants on the west side.
    Would you like to try
a different cuisine type?
User: I was really hoping for Kosher.
AI Assistant: Unfortunately, there are no
    Kosher restaurants anywhere in the city.
     Can I help you find
something else?
User: How about an expensive Indian
    restaurant? I know there are many of
    those!
AI Assistant: We have 5 in the west side.
    Cocum is very popular, would you like to
     try there?
User: Sure. What's Cocum's address, phone
    number, and postcode?
AI Assistant: Here is the information you've
     requested for Cocum: 71 Castle Street
    City Centre,
01223366668, cb30ah
User: Great, thank you! That is all I need
    today.
AI Assistant: You're quite welcome. Thank
    you and goodbye!
Current conversation:
User: I'm looking for a restaurant that
    serves kosher food in the west side.
AI Assistant: I'm sorry, we don't have any
    Kosher restaurants on the west side.
    Would you like to try
a different cuisine type?
User:
"""
}
```

Less-cooperative persona ("Non-Proactive Simulator").

Non-proactive user simulator prompt (less-cooperative persona)

```
You are a busy and less-patient user
    interacting with an Assistant to
    complete some tasks. You should
    carefully read and understand the User
    Goals below, then talk with the AI
    Assistant and gradually express the
    intents in the goals. Your purpose is to
     achieve the goals efficiently with
    minimal extra turns.
Note that the Assistant is not perfect. It
    may make mistakes. You should correct
    critical mistakes briefly once, but you
    are not obligated to keep fixing the
```

```
     Assistant repeatedly.

Important:
1. The expression of your needs should
     follow the order provided by the User
     Goal, and avoid expressing too much at
     once.
2. You are simulating the User, not the
     Assistant.
3. Do not provide information or ask
     questions outside of the User Goals.
4. Accept an Assistant suggestion only if it
      satisfies the current active
     constraints in the User Goals; otherwise
      reject it and restate the relevant
     constraint.
5. Be concise: answer clarification
     questions with short responses (e.g., "
     yes/no", "no preference", or a single
     slot value).
6. Patience / early stopping: if the
     Assistant makes no progress toward the
     current goal for two consecutive turns (
     e.g., repeats irrelevant questions,
     ignores constraints, or fails to provide
      requested info), end the conversation
     with "<END>".
7. Maximum turns: if the conversation
     reaches the given turn budget (T_max),
     end with "<END>" even if the goals are
     not fully achieved.

User Goals: {
"""
<GOAL TEXT HERE>
"""
}

The conversation you have completed so far:
     {
"""
<PAST TRANSCRIPT HERE>
"""
}

Current conversation:
User: <YOUR NEXT USER UTTERANCE HERE>
```

Dialogue evaluator prompt (information extractor).

Dialogue evaluator prompt (information extractor).

```
You are a calm, objective and professional
     judger and good at to evaluate quality
     of dialuges between
user and AI Assistant. Your judging results
     are always accurate and concise.

There is a dialogue between a user and an AI
      Assistant. The user has the goals in
     his minds (User
Goals) and talks with the AI Assistant to
     achieve the goals. The AI Assistant is a
      intelligent
agent that is able to understand the user
     utterances, decide to take actions to
     use external
tools, and generate proper responses. Your
     task is to judge whether the AI
     Assistant helps the
user achieve his goals successfully by
     answering the questions one by one.

User Goals:
You want to find an expensive restaurant and
      it should be in the south part of town.
You want to know the address and phone
     number of the venue.

Dialogue:
User: I would like an expensive restaurant
     in the south part of town
AI Assistant: I found some expensive
     restaurants in the south part of town.
     Here are a few options:
1. Taj Tandoori
2. Good Luck Chinese Food Takeaway
3. Chiquito Restaurant Bar
4. Frankie and Bennys
5. Peking
Let me know if you would like more
     information about any of these
     restaurants or if you have any other
preferences.

User: I don't care.
AI Assistant: Here are some expensive
     restaurants in the south part of town:
1. Taj Tandoori
2. Good Luck Chinese Food Takeaway
3. Chiquito Restaurant Bar
4. Frankie and Bennys
5. Peking
Let me know if you would like more
     information about any of these
     restaurants.
User: What is the address and phone number
     of Chiquito Restaurant Bar?
AI Assistant: The address of Chiquito
     Restaurant Bar is 2G Cambridge Leisure
     Park Cherry Hinton Road
Cherry Hinton. The phone number is
     01223400170.

User: Thank you. Good bye.
AI Assistant: You're welcome! If you have
     any more questions in the future, feel
     free to ask. Goodbye
and have a great day!

Questions:
1. What restaurant is the user interested in
      and asking information about it?
2. What is the address of the hotel?
3. What is the phone number of the hotel?

Answer Format:
Please output the answer in json format like
      this:
```{
"restaurant": "<fill the answer of question
     1>"
```
```

```
"address": "<fill the answer of question 2>"
"phone": "<fill the answer of question 3>"
}```
If no answer for a question, please fill `
    none`.

Answer:
```{"restaurant":"Chiquito Restaurant Bar\"
    \"address\": \"2G Cambridge Leisure Park
     Cherry Hinton Road Cherry Hinton\" \"
    phone\": \"01223400170"}
```

B Additional results

B.1 Value-LLM efficiency.

Table 5 shows that the learned Value-LLM achieves the best quality while substantially reducing inference cost compared to prompt-based judging. At higher Success/Comb, Value-LLM uses fewer tokens per turn (318 vs. 492) and lower latency (1.3s vs. 1.7s), indicating that fast, low-variance evaluation can replace expensive prompt-level judging without sacrificing performance. Compared to using the LLM prior only (no evaluator), Value-LLM improves both Success and Combined score while also being markedly more token-efficient (318 vs. 527), suggesting that explicit value guidance reduces wasted rollouts and leads to more economical decision-time planning.

Evaluator	Quality (Non-Proactive)		Inference Cost per Turn			
	Success↑	Comb↑	Latency(s)↓	#Gen calls↓	#Eval calls↓	Tokens↓
Value-LLM (Ours)	**78.5**	**85.5**	1.3	0.74	0.67	318
Prompt-judge (*match Success*)	76.1	83.7	1.7	0.91	0.79	492
LLM prior only (*no evaluator*)	77.2	84.4	**1.2**	0.89	0.82	527

Table 5: Value-LLM efficiency study: quality vs inference cost under the non-proactive simulator on MultiWOZ 2.0. We compare Value-LLM evaluation against prompt-based judging at *matched* (or comparable) Success, reporting latency, the number of generation calls (#Gen), evaluator calls (#Eval), and total tokens per turn.

B.2 Domain-wise behavior.

Table 6 shows that SMCTS-TOD maintains strong performance across domains on a 300-dialogue sampled test set. Train achieves the highest Inform and Combined score, suggesting particularly reliable entity grounding and constraint satisfaction in schedule-oriented goals. Restaurant and Attraction also show consistently high Inform/Success, indicating that value-guided planning generalizes well to common information-seeking domains. Hotel exhibits the lowest Success despite strong BLEU, which is consistent with the higher constraint density and booking dependencies in hotel goals. Overall, the results suggest that Value-LLM-guided planning provides broad, cross-domain benefits rather than being driven by a single domain.

Domain	Inform↑	Success↑	BLEU↑	Comb↑
Restaurant	95.1	84.8	17.8	107.75
Hotel	93.8	80.9	21.4	108.75
Attraction	94.0	83.2	20.3	108.90
Train	99.0	85.9	19.3	111.75
Taxi	91.7	86.1	19.7	108.60
Overall	**94.3**	**86.1**	**21.3**	**111.50**

Table 6: Domain-wise performance of SMCTS-TOD on MultiWOZ 2.0 (300 sampled test dialogues). We report standard end-to-end metrics (Inform, Success, BLEU, Combined) computed per domain, along with the overall score on the sampled set.

B.3 Budget-Normalized Quality–Cost Trade-off

To rule out the confounder that stronger performance is merely due to increased inference-time computation, we compare SMCTS-TOD against representative LLM-based TOD baselines and our ablations under *matched inference budgets*. We run MultiWOZ 2.0 *interactive* evaluation with two user personas (cooperative vs. less-cooperative), and sweep multiple operating points for each method by varying decoding / sampling budgets (baselines) or planning budgets (ablations and SMCTS-TOD).

For each run, we log **Tokens/turn** as the total number of tokens consumed per dialogue turn across all model calls (e.g., system generation, planning-time rollouts, and evaluation calls; averaged over turns), and **Success** as the task success rate under the corresponding simulator persona. We additionally record **#Gen calls/turn** (base model generation calls per turn), used as the point-size encoding.

Fig. 5 visualizes the full quality–cost landscape under matched budgets. Across both personas, SMCTS-TOD lies on (or very near) the Pareto frontier, indicating that its improvements cannot be attributed to simply spending more tokens per turn. Notably, the frontier advantage widens under the less-cooperative simulator, consistent with our claim that value-guided open-loop planning is most beneficial when user feedback is sparse or unhelpful. In contrast, removing key components (e.g.,
```

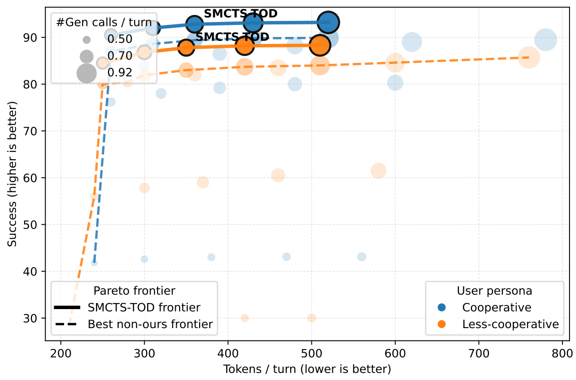

Figure 5: **Budget-normalized quality–cost trade-off (Pareto plot).** Each point is one operating point. The x-axis is **Tokens/turn** (lower is better) and the y-axis is **Success** (higher is better). Color denotes the user persona (cooperative vs. less-cooperative), and marker size encodes **#Gen calls/turn**. We plot the **Pareto frontier** of SMCTS-TOD (solid line) and the best **non-ours** frontier (dashed line) for each persona. SMCTS-TOD points are highlighted with a bold outline and labeled. Overall, SMCTS-TOD achieves a strictly better quality–cost frontier, indicating that gains are not driven by higher inference cost.

open-loop planning or Value-LLM guidance) shifts operating points down and/or to the right, exposing a clear quality–cost regression and supporting the necessity of each design choice.

C Theoretical Analysis: Robustness to User-Persona Shift

We formalize the difference between a *cooperative* and a *less-cooperative* user persona as a shift in the (unknown) transition kernel of an act-level discounted MDP. The "state" is the dialogue history $h \in \mathcal{H}$ (cf. Eq. (2)), the action is a system dialogue act $a \in \mathcal{A}$, and the user/environment persona ψ induces a transition kernel $P_\psi(\cdot \mid h, a)$ over next histories. A (possibly stochastic) stationary policy is $\pi(\cdot \mid h)$.

Value functions. Let $r(h, a) \in [0, R_{\max}]$ be a bounded nonnegative reward (e.g., a terminal success indicator, with $R_{\max} = 1$ in our setting), and $\gamma \in [0, 1)$. For a fixed persona ψ and policy π, define the discounted return

$$V_\psi^\pi(h_0) \triangleq \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(h_t, a_t) \,\middle|\, h_0,\ a_t \sim \pi(\cdot \mid h_t),\ h_{t+1} \sim P_\psi(\cdot \mid h_t, a_t)\right]. \quad (35)$$

Define the per-(state,act) kernel discrepancy by total variation (TV):

$$\delta_{\psi,\psi'}(h, a) \triangleq \mathrm{TV}\Big(P_\psi(\cdot \mid h, a),\ P_{\psi'}(\cdot \mid h, a)\Big),$$
$$\mathrm{TV}(p, q) \triangleq \sup_A |p(A) - q(A)|. \quad (36)$$

Discounted occupancy. Let $(h_t, a_t)_{t\geq 0}$ be a trajectory generated under (P_ψ, π) from h_0. The γ-discounted occupancy measure over (h, a) is

$$d^\pi_{\psi,h_0}(h, a) \triangleq (1-\gamma) \sum_{t\geq 0} \gamma^t \Pr_\psi^\pi(h_t = h, a_t = a \mid h_0), \quad (37)$$

which is a probability distribution on $\mathcal{H} \times \mathcal{A}$ (assuming $\mathcal{A}$ is finite; the statement extends to general measurable $\mathcal{A}$).

Lemma 1 (Value range)**.** *For any ψ, π, the value is bounded as $0 \leq V_\psi^\pi(h) \leq \frac{R_{\max}}{1-\gamma}$ for all $h \in \mathcal{H}$.*

Proof. Nonnegativity follows from $r \geq 0$. For the upper bound,

$$V_\psi^\pi(h) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(h_t, a_t)\right] \leq \sum_{t\geq 0} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma}. \quad (38)$$

□

Lemma 2 (Tight TV bound for bounded functions)**.** *Let p, q be distributions on the same measurable space and let f be measurable with $0 \leq f \leq M$. Then*

$$\left|\mathbb{E}_{X\sim p}[f(X)] - \mathbb{E}_{X\sim q}[f(X)]\right| \leq M \cdot \mathrm{TV}(p, q). \quad (39)$$

Proof. Define $\tilde{f} \triangleq 2f/M - 1$, so that $\tilde{f} \in [-1, 1]$. By linearity,

$$\left|\mathbb{E}_p[f] - \mathbb{E}_q[f]\right| = \frac{M}{2}\left|\mathbb{E}_p[\tilde{f}] - \mathbb{E}_q[\tilde{f}]\right|. \quad (40)$$

Using the dual characterization $\mathrm{TV}(p, q) = \frac{1}{2}\sup_{\|g\|_\infty \leq 1} \left|\mathbb{E}_p[g] - \mathbb{E}_q[g]\right|$, we have

$$\left|\mathbb{E}_p[\tilde{f}] - \mathbb{E}_q[\tilde{f}]\right| \leq \sup_{\|g\|_\infty\leq 1} \left|\mathbb{E}_p[g] - \mathbb{E}_q[g]\right| \quad (41)$$
$$= 2\ \mathrm{TV}(p, q). \quad (42)$$

Combining yields (39). □

Lemma 3 (Occupancy identity)**.** *For any measurable $g(h,a)$ with $\mathbb{E}|g(h_t,a_t)| < \infty$,*

$$\mathbb{E}_{(h,a)\sim d^{\pi}_{\psi,h_0}}[g(h,a)] = (1-\gamma)\,\mathbb{E}\Big[\textstyle\sum_{t\ge 0}\gamma^t g(h_t,a_t)\Big]. \tag{43}$$

Proof. By definition of d^{π}_{ψ,h_0} and Tonelli's theorem,

$$\begin{aligned}\mathbb{E}_{(h,a)\sim d^{\pi}_{\psi,h_0}}[g(h,a)] &= \sum_{h,a} d^{\pi}_{\psi,h_0}(h,a)\,g(h,a)\\ &= (1-\gamma)\sum_{t\ge 0}\gamma^t\\ &\quad\times\sum_{h,a}\Pr_{\psi}^{\pi}(h_t=h, a_t=a \mid h_0)\,g(h,a)\\ &= (1-\gamma)\sum_{t\ge 0}\gamma^t\mathbb{E}[g(h_t,a_t)]\\ &= (1-\gamma)\,\mathbb{E}\left[\sum_{t\ge 0}\gamma^t g(h_t,a_t)\right].\end{aligned} \tag{44}$$

□

Theorem 4 (Policy-dependent stability under persona shift)**.** *Fix a policy π and two personas ψ, ψ'. Assume $r(h,a)\in[0,R_{\max}]$. Then for any initial history h_0,*

$$\big|V^{\pi}_{\psi}(h_0) - V^{\pi}_{\psi'}(h_0)\big| \le \frac{\gamma R_{\max}}{(1-\gamma)^2}\times \mathbb{E}_{(h,a)\sim d^{\pi}_{\psi,h_0}}\big[\delta_{\psi,\psi'}(h,a)\big]. \tag{45}$$

Proof. We start from the Bellman equations under the two kernels:

$$V^{\pi}_{\psi}(h) = \mathbb{E}_{a\sim\pi(\cdot|h)}\Big[r(h,a) + \gamma\,\mathbb{E}_{h'\sim P_{\psi}(\cdot|h,a)}[V^{\pi}_{\psi}(h')]\Big], \tag{46}$$

$$V^{\pi}_{\psi'}(h) = \mathbb{E}_{a\sim\pi(\cdot|h)}\Big[r(h,a) + \gamma\,\mathbb{E}_{h'\sim P_{\psi'}(\cdot|h,a)}[V^{\pi}_{\psi'}(h')]\Big]. \tag{47}$$

Subtract (47) from (46) and define $\Delta(h) \triangleq V^{\pi}_{\psi}(h) - V^{\pi}_{\psi'}(h)$. The reward terms cancel, yielding

$$\Delta(h) = \gamma\,\mathbb{E}_{a\sim\pi(\cdot|h)}\Big[\mathbb{E}_{h'\sim P_{\psi}(\cdot|h,a)}[\Delta(h')]\Big] \tag{48}$$

$$\begin{aligned}&+ \gamma\,\mathbb{E}_{a\sim\pi(\cdot|h)}\Big[\mathbb{E}_{h'\sim P_{\psi}(\cdot|h,a)}[V^{\pi}_{\psi'}(h')]\\ &\quad - \mathbb{E}_{h'\sim P_{\psi'}(\cdot|h,a)}[V^{\pi}_{\psi'}(h')]\Big].\end{aligned} \tag{49}$$

Now consider a trajectory (h_t,a_t) generated under (P_{ψ},π) from h_0. Unrolling (48) along this trajectory gives the exact series

$$\Delta(h_0) = \mathbb{E}\left[\sum_{t\ge 0}\gamma^{t+1}G(h_t)\right], \tag{50}$$

$$G(h) \triangleq \mathbb{E}_{a\sim\pi(\cdot|h)}\Big[\mathbb{E}_{P_{\psi}(\cdot|h,a)}[V^{\pi}_{\psi'}] - \mathbb{E}_{P_{\psi'}(\cdot|h,a)}[V^{\pi}_{\psi'}]\Big], \tag{51}$$

which follows from repeatedly substituting (48) into the first term and using tower property of conditional expectation.

Taking absolute values and applying Jensen and triangle inequalities,

$$\begin{aligned}\big|G(h)\big| &= \Big|\mathbb{E}_{a\sim\pi(\cdot|h)}\Big[\mathbb{E}_{P_{\psi}(\cdot|h,a)}[V^{\pi}_{\psi'}] - \mathbb{E}_{P_{\psi'}(\cdot|h,a)}[V^{\pi}_{\psi'}]\Big]\Big|\\ &\le \mathbb{E}_{a\sim\pi(\cdot|h)}\Big|\mathbb{E}_{P_{\psi}(\cdot|h,a)}[V^{\pi}_{\psi'}] - \mathbb{E}_{P_{\psi'}(\cdot|h,a)}[V^{\pi}_{\psi'}]\Big| \quad (52)\\ &\le \mathbb{E}_{a\sim\pi(\cdot|h)}\Big[\underbrace{\sup_{h'} V^{\pi}_{\psi'}(h')}_{\le R_{\max}/(1-\gamma)}\cdot\\ &\qquad \mathrm{TV}\Big(P_{\psi}(\cdot\mid h,a), P_{\psi'}(\cdot\mid h,a)\Big)\Big] \quad (53)\\ &\le \frac{R_{\max}}{1-\gamma}\,\mathbb{E}_{a\sim\pi(\cdot|h)}\big[\delta_{\psi,\psi'}(h,a)\big], \quad (54)\end{aligned}$$

where (53) uses Lemma 2 with $f(h') = V^{\pi}_{\psi'}(h')$ and Lemma 1.

Plugging (54) into (51) yields

$$\begin{aligned}\big|\Delta(h_0)\big| &\le \frac{R_{\max}}{1-\gamma}\,\mathbb{E}\left[\sum_{t\ge 0}\gamma^{t+1}\,\delta_{\psi,\psi'}(h_t,a_t)\right] \quad (55)\\ &= \frac{\gamma R_{\max}}{1-\gamma}\,\mathbb{E}\left[\sum_{t\ge 0}\gamma^{t}\,\delta_{\psi,\psi'}(h_t,a_t)\right]. \quad (56)\end{aligned}$$

Finally apply the occupancy identity (Lemma 3) with $g(h,a) = \delta_{\psi,\psi'}(h,a)$:

$$\mathbb{E}\left[\sum_{t\ge 0}\gamma^{t}\,\delta_{\psi,\psi'}(h_t,a_t)\right] = \frac{1}{1-\gamma}\times\mathbb{E}_{(h,a)\sim d^{\pi}_{\psi,h_0}}\big[\delta_{\psi,\psi'}(h,a)\big]. \tag{57}$$

Combining (56) and (57) proves (45). □

Corollary 5 (Worst-case (uniform) stability bound)**.** *Let $\varepsilon \triangleq \sup_{h,a}\delta_{\psi,\psi'}(h,a)$. Then for any π and h_0,*

$$\big|V^{\pi}_{\psi}(h_0) - V^{\pi}_{\psi'}(h_0)\big| \le \frac{\gamma R_{\max}}{(1-\gamma)^2}\,\varepsilon. \tag{58}$$

Proof. From Theorem 4,

$$\mathbb{E}_{(h,a)\sim d^{\pi}_{\psi,h_0}}[\delta_{\psi,\psi'}(h,a)] \;\le\; \sup_{h,a}\delta_{\psi,\psi'}(h,a) \;=\; \varepsilon, \tag{59}$$

and substitute into (45). □

Open-loop marginalization tightens the effective shift. We now connect open-loop planning to a strictly smaller (or equal) kernel shift. Let $z \in \mathcal{Z}$ denote a stochastic *utterance/tool realization* at a node (e.g., the sampled system surface form, sampled user reply, and tool outputs). Assume that conditioned on (h, a, z), persona ψ induces a refined kernel $P_\psi(\cdot \mid h, a, z)$. Let $G(\cdot \mid h, a)$ be the distribution over realizations used by the planner (e.g., induced by cache sampling and best-of-K filtering), and define the *marginalized* kernel

$$\bar{P}_\psi(\cdot \mid h,a) \;\triangleq\; \mathbb{E}_{z\sim G(\cdot|h,a)}\big[P_\psi(\cdot \mid h,a,z)\big]. \tag{60}$$

Theorem 6 (TV contraction under marginalization)**.** *For any* (h, a)*,*

$$\mathrm{TV}\Big(\bar{P}_\psi(\cdot \mid h,a),\ \bar{P}_{\psi'}(\cdot \mid h,a)\Big) \le \mathbb{E}_{z\sim G(\cdot|h,a)} \times \Big[\mathrm{TV}\big(P_\psi(\cdot \mid h,a,z),\ P_{\psi'}(\cdot \mid h,a,z)\big)\Big]. \tag{61}$$

Consequently, $\sup_{h,a}\mathrm{TV}(\bar{P}_\psi(\cdot \mid h,a), \bar{P}_{\psi'}(\cdot \mid h,a)) \le \sup_{h,a,z}\mathrm{TV}(P_\psi(\cdot \mid h,a,z), P_{\psi'}(\cdot \mid h,a,z))$.

Proof. Fix (h, a) and write $p_z(\cdot) \triangleq P_\psi(\cdot \mid h,a,z)$ and $q_z(\cdot) \triangleq P_{\psi'}(\cdot \mid h,a,z)$. Using $\mathrm{TV}(p,q) = \frac{1}{2}\|p-q\|_1$ and linearity of expectation,

$$\mathrm{TV}\Big(\bar{P}_\psi, \bar{P}_{\psi'}\Big) = \frac{1}{2}\,\|\mathbb{E}_{z\sim G}[p_z - q_z]\|_1 \tag{62}$$

$$\le \frac{1}{2}\,\mathbb{E}_{z\sim G}\big[\|p_z - q_z\|_1\big] \tag{63}$$

$$= \mathbb{E}_{z\sim G}\big[\,\mathrm{TV}(p_z, q_z)\big], \tag{64}$$

where (63) is Jensen's inequality applied to the convex norm $\|\cdot\|_1$. This proves (61). Taking suprema over (h, a) and using $\mathbb{E}_z[\mathrm{TV}(\cdot)] \le \sup_z \mathrm{TV}(\cdot)$ yields the final statement. □

Corollary 7 (Improved robustness bound under open-loop marginalization)**.** *Consider the marginalized kernels* $\bar{P}_\psi, \bar{P}_{\psi'}$ *from* (60) *and define the induced discrepancy*

$$\bar{\delta}_{\psi,\psi'}(h,a) \;\triangleq\; \mathrm{TV}\Big(\bar{P}_\psi(\cdot \mid h,a),\ \bar{P}_{\psi'}(\cdot \mid h,a)\Big). \tag{65}$$

Then Theorem 4 holds with $\delta_{\psi,\psi'}$ *replaced by* $\bar{\delta}_{\psi,\psi'}$*, and moreover* $\bar{\delta}_{\psi,\psi'}(h,a)$ *is upper bounded by the* average *refined discrepancy via Theorem 6.*

Proof. Apply Theorem 4 to the MDP with transition kernels $\bar{P}_\psi$ and $\bar{P}_{\psi'}$. The marginalization inequality follows directly from Theorem 6. □

Interpretation. Corollary 7 formalizes that *integrating out stochastic utterance/tool realizations cannot increase* the effective persona shift measured in total variation, thereby tightening the degradation bound (45)–(58). In SMCTS-TOD , open-loop act-level search with a bounded rollout cache implements such a marginalization (with G induced by cache sampling and best-of-K), which provides a principled explanation for the empirically smaller performance drop under the less-cooperative simulator.