Planning Persuasion, Not Utterances: Profile-Conditioned Open-Loop Search for Dialogue Strategy

Zhiming Lin[1*] **Kai Zhao**[2*] **Chengzhen Yu**[3] **Yihao Zhong**[4]
Ruibo Duan[5] **Yuliang Gai**[6,7] **Xander Zhang**[8†]
[1]Independent Researcher [2]Western Sydney University, Australia
[3]Nanyang Technological University [4]New York University
[5]Seoul National University [6]Massachusetts Institute of Technology
[7]University of Massachusetts Boston [8]National University of Singapore

Abstract

Persuasive dialogue systems are increasingly important for prosocial communication, education, and support-oriented interaction, but they still struggle to make profile-sensitive decisions when user reactions are uncertain and persuasive outcomes are delayed. This paper aims to enable stable long-horizon persuasive planning that adapts to user profiles while avoiding brittle text-level search and costly prompt-based self-evaluation. We propose PMCTS-PD, a profile-conditioned open-loop tree search framework that plans over compact dialogue-act prefixes rather than full utterances, thereby separating strategic decision making from surface realization. Each search node aggregates multiple instantiated dialogue histories, while a lightweight profile-conditioned Value-LLM predicts normalized expected donation to guide tree search, value backup, and value-guided best-of-K response generation. On PERSUASIONFORGOOD, PMCTS-PD improves over strong LLM and dialogue-planning baselines, improving response quality and simulated success, with support from independent external reward and human preference evaluation. These results show that planning persuasion at the strategy level, supported by a learned profile-aware outcome signal, offers a practical and cost-effective direction for robust persuasive dialogue systems.

1 Introduction

Persuasive dialogue systems are increasingly deployed in high-impact settings such as charitable fundraising, customer support, education, and health communication, where a sequence of conversational decisions can shape user beliefs and actions [Tan et al., 2016, Hidey and McKeown, 2018, Yang et al., 2019]. In these applications, effectiveness hinges on multi-turn adaptation: the same persuasive move can help one user yet backfire for another, and user reactions are inherently uncertain [Wang et al., 2019, Zhang et al., 2018]. Achieving both persuasiveness and conversational quality under practical latency and cost constraints remains a central challenge for real-world adoption [cf. Lu et al., 2026, Zhang et al., 2026b].

Prior work has made substantial progress on strategy-aware generation by leveraging annotated persuasion strategies and user attributes, enabling conditional response generation and controllable modules [Wang et al., 2019, Chen et al., 2022]. Recent LLM-based systems further improve fluency and can incorporate external information for factual support [Furumai et al., 2024], while

*Equal contribution. Zhiming Lin and Kai Zhao are co-first authors.
†Corresponding author: xuanzhang2199@gmail.com.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

broader search-augmented LLM frameworks explore explicit reasoning over candidate continuations [Yao et al., 2023, Zhou et al., 2024]. However, existing approaches still face a fundamental gap: they often emphasize single-step response optimization or rely on hand-designed orchestration, rather than explicitly addressing long-horizon decision making under stochastic user responses [Chen et al., 2022, Furumai et al., 2024].

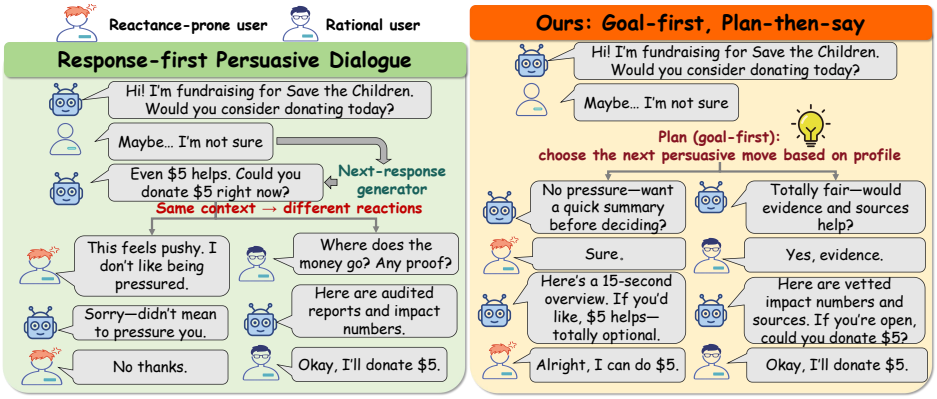

Figure 1: Unlike response-first methods that commit to a fixed next utterance and risk profile-dependent failure, our approach plans persuasive moves toward an explicit outcome objective before generating language, yielding adaptive and robust persuasion across heterogeneous users.

Moreover, although user profiles are commonly appended as additional context, most systems lack a principled mechanism to translate profile information into an optimization signal for selecting persuasive moves when outcome supervision is sparse and delayed [Wang et al., 2019, Zhang et al., 2018]. Finally, when explicit search is used with language models, operating directly over textual states and depending on high-variance self-evaluation can make planning brittle and expensive [Yao et al., 2023, Zhou et al., 2024].

This paper asks: *How can we enable profile-sensitive, long-horizon persuasive decision making while keeping planning stable and cost-effective under uncertain user reactions?* We address this gap by introducing a planning-oriented formulation that reasons over a compact representation of persuasive moves rather than committing to a single brittle textual trajectory, and by learning a lightweight, profile-conditioned utility estimator that provides low-variance feedback for intermediate dialogue contexts. At a high level, our approach separates *what* persuasive move to take (planning) from *how* to express it (realization), enabling robust multi-step decision making without requiring expensive evaluation at every candidate text.

Our contributions are as follows: **(i)** We formalize profile-conditioned persuasive dialogue as a long-horizon decision problem with delayed, outcome-based utility, highlighting the mismatch between stochastic user reactions and predominantly myopic optimization in prior systems. **(ii)** We propose a profile-aware planning framework that searches over sequences of persuasive moves using a learned utility signal and decouples strategic planning from surface-level generation, improving robustness under uncertainty. **(iii)** On PERSUASIONFORGOOD [Wang et al., 2019], our method improves response quality and persuasiveness proxies while reducing inference overhead relative to evaluation-heavy planning baselines.

2 Related Work

Persuasive dialogue and strategy-aware generation. Early computational studies on persuasion focused on identifying persuasive cues and strategies and predicting outcomes in online arguments and requests [Tan et al., 2016, Hidey and McKeown, 2018, Yang et al., 2019]. For multi-turn persuasion, Wang et al. [2019] introduced PERSUASIONFORGOOD, pairing strategy annotations with rich user attributes and donation outcomes, which enabled supervised strategy modeling and conditional generation baselines. Subsequent systems improved controllability via modular architectures and explicit strategy interfaces, e.g., the Response-Agenda Pushing (RAP) framework that interleaves social acknowledgement, factual QA, and agenda pushing [Chen et al., 2022]. More recently,

strong LLMs have enabled zero-shot persuasive chatbots, with PersuaBot combining LLM-generated strategies and retrieval-based fact checking for factuality [Furumai et al., 2024]. Complementary RAG research separates semantic relevance from factual consistency to resolve knowledge conflicts [Ye et al., 2026]. In optical network management, Xiao et al. [2026b] use YANG schemas to guide retrieval and validate configurations generated by an LLM agent for multi-vendor OpenROADM networks [see also Xiao et al., 2026a]. However, these persuasive systems either rely on hand-designed dispatch/agenda components [Chen et al., 2022] or primarily optimize *single-step* responses without explicit multi-step planning under stochastic user reactions [Furumai et al., 2024]. In contrast, PMCTS-PD performs *planning* in a compact dialogue-act space and treats surface utterances as stochastic realizations, enabling robust long-horizon strategy search.

Personalization with user profiles. Personalized dialogue has been widely studied by conditioning generation on personas or user attributes [Zhang et al., 2018]. Beyond dialogue, ProtoFed-SP uses prototype-anchored short- and long-term soft prompts for continual web personalization [Xiao and Hou, 2026], addressing preference adaptation rather than strategy utility. In persuasion, user heterogeneity is particularly salient: P4G explicitly measures demographic/psychological factors and shows that strategy effectiveness varies across individuals [Wang et al., 2019]. Yet most personalized generators treat the profile as extra context rather than an optimization signal, and they do not provide a learned *utility* model to guide strategy selection under limited outcome supervision. We address this gap by training a lightweight *profile-conditioned Value-LLM* that predicts normalized expected donation and directly supports both MCTS backups and best-of-K realization.

Planning and search with language models. Planning in dialogue has been explored in negotiation settings via rollout-based decision making [Lewis et al., 2017] and by decoupling high-level acts from language generation for better control [He et al., 2018] [cf. Zhang et al., 2026c]. In robotic assembly, a graph Transformer predicts assembly sequences from graphs of parts [Ma et al., 2023]. In optical transport networks, Xiao et al. [2025] use multi-head graph attention to correlate alarms and identify the root alarm. In parallel, LLM inference has increasingly incorporated explicit search, such as Tree-of-Thoughts [Yao et al., 2023] and MCTS-based agent frameworks like LATS [Zhou et al., 2024]. A common limitation is that search often operates over brittle textual states and depends on expensive, high-variance LLM self-evaluation [Yao et al., 2023, Zhou et al., 2024]. GDP-Zero [Yu et al., 2023] already uses open-loop MCTS and prompts an LLM as policy prior, value function, user simulator, and system model. Our contribution builds on this search structure with a learned profile-conditioned terminal-utility estimator, MCTS-to-value distillation, and act-constrained best-of-K realization. Relatedly, Berant et al. [2025] learn cost-steerable clarification policies through collaborative self-play and Reinforced Self-Training; PMCTS-PD instead performs inference-time act search conditioned on profiles and delayed task utility. Complementary budget-aware methods improve multi-turn intent understanding through diverse retrieval [Lin et al., 2026a] and select or compress ICL demonstrations under context limits [Zhang et al., 2026e]; PMCTS-PD allocates computation to act search.

3 Method

PMCTS-PD plans persuasive dialogue in a compact dialogue-act space and treats surface utterances as stochastic realizations. At each turn, we run an open-loop Monte Carlo Tree Search (MCTS) whose nodes index action prefixes (instead of storing a single realized utterance at each node), which reduces compounding errors caused by brittle simulated texts. A base LLM provides an act prior to guide tree expansion, and a lightweight profile-conditioned Value-LLM amortizes evaluation by predicting a *normalized expected donation* from the user profile and dialogue context. After selecting the act, we attach a high-quality system utterance via value-guided best-of-K generation. We further improve the Value-LLM with an AlphaZero-style distillation loop using MCTS-improved targets. Fig. 2 summarizes the pipeline.

3.1 Problem Setup: Persuasion with User Profiles

Persuasion success depends on user heterogeneity; the same strategy can succeed for one profile and fail for another. We therefore condition planning on a user profile p and model user responses as stochastic [cf. Fan et al., 2026, He et al., 2026].

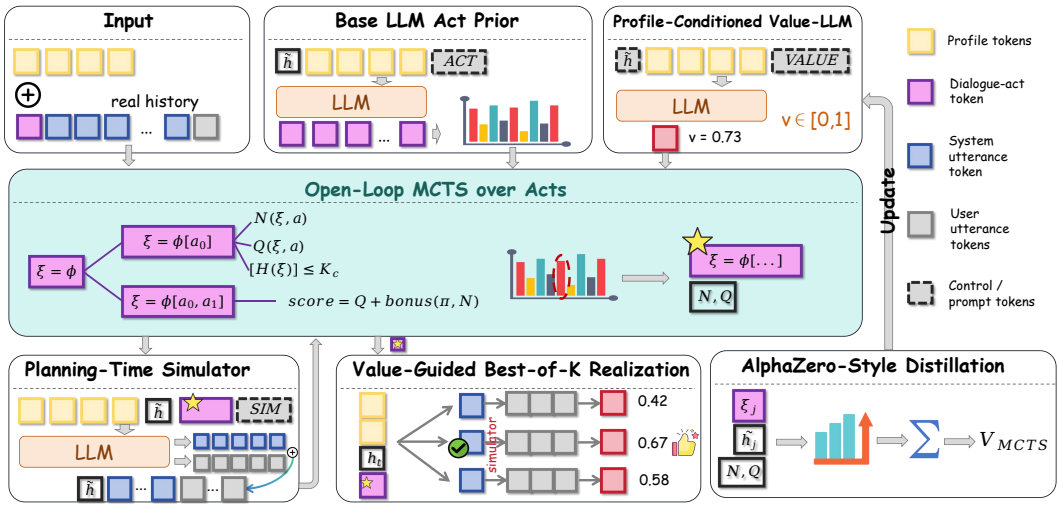

Figure 2: **PMCTS-PD pipeline.** Given user profile tokens and the real dialogue history, we plan in a compact dialogue-act space using open-loop MCTS whose nodes are act prefixes ξ and whose cache $\mathcal{H}(\xi)$ stores multiple instantiated histories $\tilde{h}$. A base LLM provides an empirical act prior $\pi_\theta(a \mid \xi)$ via sampling, and a profile-conditioned Value-LLM outputs a normalized expected donation $v = V_\psi(p, \tilde{h}) \in [0, 1]$ for low-variance evaluation and value backups over B simulations. A planning-time simulator stochastically generates $(u^{\text{sys}}, u^{\text{usr}})$ to extend $\tilde{h}$ without committing utterances as tree states. After selecting the act $a_t^\star$, we realize it via value-guided best-of-K generation by scoring sampled candidates with the Value-LLM. An AlphaZero-style distillation loop then forms MCTS-improved targets v_{MCTS} from (N, Q) and further trains the Value-LLM.

We assume a fixed dialogue-act inventory $\mathcal{A}$. At turn t, the system observes profile p and real dialogue history h_t, selects an act $a_t \in \mathcal{A}$, and produces a system utterance u_t^{sys} that realizes a_t; the user then responds with u_t^{usr}. We represent the multi-turn history as

$$h_t = \Big(p,\ (a_0, u_0^{\text{sys}}, u_0^{\text{usr}}), \ldots, (a_{t-1}, u_{t-1}^{\text{sys}}, u_{t-1}^{\text{usr}})\Big), \tag{1}$$

where u_i^{sys} and u_i^{usr} denote the system and user utterances at turn i. Let m denote the terminal persuasion outcome (donation amount) at dialogue end. At each turn we plan to maximize the conditional expected outcome,

$$\max_{\pi}\ \mathbb{E}[m \mid p, h_t;\ \pi]\,, \tag{2}$$

where the expectation is taken over stochastic dialogue trajectories induced by π and user responses until termination.

3.2 Open-Loop MCTS over Dialogue Acts

Closed-loop dialogue search that stores a single realized utterance at each node is brittle: an early low-quality realization can dominate the entire subtree. PMCTS-PD uses open-loop MCTS, where a node is an *act-prefix* and aggregates statistics across multiple realized histories.

A tree node is an act-prefix $\xi = (a_0, \ldots, a_{d-1})$ of depth d, with concatenation written as $\xi \cdot a$. Each node maintains visit counts $N(\xi, a)$, action values $Q(\xi, a)$, and a bounded cache $\mathcal{H}(\xi)$ of *instantiated* histories $\tilde{h}$ consistent with ξ. Here $\tilde{h}$ contains surface utterances and is used only for simulation and value evaluation; the node identity remains ξ.

PUCT Selection. Starting from the root $\xi_0 = \emptyset$, we select actions using a PUCT rule, following predictor-guided exploration [Rosin, 2011] and policy/value-guided tree search [Silver et al., 2017]:

$$a \leftarrow \arg\max_{a \in \mathcal{A}} \left[Q(\xi, a) + c_p\, \pi_\theta(a \mid \xi)\, \frac{\sqrt{N(\xi)}}{1 + N(\xi, a)} \right], \tag{3}$$

where $N(\xi) = \sum_{a' \in \mathcal{A}} N(\xi, a')$ is the node visit count, $c_p > 0$ controls exploration, and $\pi_\theta(a \mid \xi)$ is the act prior from the base LLM (Sec. 3.3).

Simulation with cached instantiated histories. We sample an instantiated history $\tilde{h} \sim \mathrm{Unif}(\mathcal{H}(\xi))$. Executing an act updates the instantiated history by one simulator step $\tilde{h} \leftarrow \mathrm{Sim}(\tilde{h}, a; p)$, which appends $(a, u^{\mathrm{sys}}, u^{\mathrm{usr}})$ generated from the base LLM prompts. We then descend to $\xi \leftarrow \xi \cdot a$ and push $\tilde{h}$ into $\mathcal{H}(\xi)$ (truncated to cap K_c). Maintaining multiple instantiated histories per node reduces sensitivity to any single low-quality realization.

Expansion and evaluation. Upon reaching an unexpanded node, we compute the act prior $\pi_\theta(\cdot \mid \xi)$ and initialize children. We evaluate the instantiated history with the Value-LLM, producing a scalar $v \leftarrow V_\psi(p, \tilde{h}) \in [0, 1]$ (Sec. 3.4).

Backup. We back up v along the traversed edges via incremental averaging:

$$N(\xi, a) \leftarrow N(\xi, a) + 1, \tag{4}$$

$$Q(\xi, a) \leftarrow Q(\xi, a) + \frac{v - Q(\xi, a)}{N(\xi, a)}. \tag{5}$$

The updated $Q(\xi, a)$ is the sample mean of backed-up values for edge (ξ, a).

Action selection. After B simulations, we select the root act by maximum visit count:

$$a_t^\star \leftarrow \arg\max_{a \in \mathcal{A}} N(\xi_0, a). \tag{6}$$

This returns the planned act; surface realization is handled by value-guided best-of-K (Sec. 3.5), which decouples act planning from utterance generation.

3.3 Base LLM Act Prior

The act prior reduces branching and focuses search on plausible strategies. Given a node ξ, we take a representative instantiated history $\tilde{h} \in \mathcal{H}(\xi)$, prompt the base LLM M_θ to sample the next act multiple times, and obtain $\pi_\theta(a \mid \xi)$ by normalized sample counts. The prior is used only inside PUCT (Eq. (3)) and does not force the final decision.

3.4 Profile-Conditioned Value-LLM

Full dialogue rollouts are expensive and high-variance. We train a lightweight Value-LLM V_ψ that maps $(p, \tilde{h})$ to a scalar value $v \in [0, 1]$ representing the *normalized expected donation* from the current context.

Let $m^{(i)}$ be the donation amount for training dialogue i, and define a fixed normalization constant $m_{\max} = \max_i m^{(i)}$. We normalize donations as $\bar{m}^{(i)} = \min(m^{(i)}/m_{\max}, 1)$ and train by mean squared error:

$$\mathcal{L}_{\text{value}}(\psi) = \frac{1}{n} \sum_{i=1}^{n} \Big(V_\psi(p^{(i)}, \tilde{h}^{(i)}) - \bar{m}^{(i)} \Big)^2. \tag{7}$$

This yields a bounded value compatible with MCTS backup (Eq. (5)) and best-of-K scoring (Eq. (8)).

3.5 Value-Guided Best-of-K Realization

Act planning selects *what* strategy to use; realization determines *how* the strategy is expressed. Given (p, h_t) and the selected act $a_t^\star$, we sample K candidate system utterances $\{u_t^{\mathrm{sys},(k)}\}_{k=1}^K$ from the base LLM and simulate a corresponding user reply for each candidate to form a scored instantiated context. We select the highest-value candidate by:

$$u_t^{\mathrm{sys}} \leftarrow \arg\max_{u_t^{\mathrm{sys},(k)}} V_\psi\Big(p,\ h_t \oplus (a_t^\star, u_t^{\mathrm{sys},(k)}, \tilde{u}_t^{\mathrm{usr},(k)})\Big), \tag{8}$$

where $\tilde{u}_t^{\mathrm{usr},(k)}$ is produced by one simulator step conditioned on $(p, h_t, a_t^\star, u_t^{\mathrm{sys},(k)})$, and $\oplus$ appends a turn triple to the history.

Algorithm 1 PMCTS-PD (single-turn planning at (p, h_t))

Require: profile p, real history h_t, budget B, cache cap K_c, best-of-K K, exploration c_p
initialize root $\xi_0 = \emptyset$, set $\mathcal{H}(\xi_0) \leftarrow \{h_t\}$
for $b = 1$ to B **do**
  $\xi \leftarrow \xi_0$
  sample $\tilde{h} \sim \mathrm{Unif}(\mathcal{H}(\xi))$
  while ξ is expanded **do**
    select $a \leftarrow \arg\max_a$ PUCT(ξ, a) {Eq. (3)}
    simulate $\tilde{h} \leftarrow \mathrm{Sim}(\tilde{h}, a; p)$
    $\xi \leftarrow \xi \cdot a$
    push $\tilde{h}$ into $\mathcal{H}(\xi)$ (truncate to cap K_c)
  end while
  compute prior $\pi_\theta(\cdot \mid \xi)$ {sample–count–normalize}
  expand ξ and initialize visit counts and values
  evaluate $v \leftarrow V_\psi(p, \tilde{h})$
  backup v along visited edges {Eq. (5)}
end for
choose act $a_t^\star \leftarrow \arg\max_a N(\xi_0, a)$ {Eq. (6)}
realize utterance $u_t^{\mathrm{sys}} \leftarrow$ best-of-K {Eq. (8)}
return $(a_t^\star, u_t^{\mathrm{sys}})$

3.6 AlphaZero-Style Distillation for the Value-LLM

With limited labeled outcomes, the value model can be miscalibrated on intermediate states. MCTS provides improved targets by aggregating many value-backed simulations; we distill these targets into the Value-LLM [cf. Zhang et al., 2024a].

From visit counts at any node ξ, we form an MCTS-improved policy

$$\pi_{\mathrm{MCTS}}(a \mid \xi) \propto N(\xi, a)^{1/\tau}, \tag{9}$$

where $\tau > 0$ is a temperature and $\propto$ indicates normalization over $a \in \mathcal{A}$. For each visited node ξ with representative instantiated history $\tilde{h} \in \mathcal{H}(\xi)$, we define the scalar target $v_{\mathrm{MCTS}}(\xi) = \sum_{a \in \mathcal{A}} \pi_{\mathrm{MCTS}}(a \mid \xi)\, Q(\xi, a)$ and train:

$$\mathcal{L}_{\mathrm{distill}}(\psi) = \mathbb{E}_{(\xi, p, \tilde{h})}\Big[\big(V_\psi(p, \tilde{h}) - v_{\mathrm{MCTS}}(\xi)\big)^2\Big]. \tag{10}$$

This loop improves value estimates without additional human labels.

4 An Abstract Error-Decomposition Analysis

This section analyzes a simplified evaluator abstraction rather than claiming a formal guarantee for the deployed QLoRA-tuned LLaMA Value-LLM. Specifically, we assume a fixed representation of the profile and dialogue prefix, with a bounded feature vector and a linear value head trained by squared loss. This abstraction isolates how evaluator and simulation errors propagate to act selection, yielding explicit sample-complexity and action-suboptimality bounds.

Setup. Let ϕ be a fixed representation map and define $x = \phi(p, \tilde{h}) \in \mathbb{R}^d$ with $\|x\|_2 \le 1$. Let $y \in [0, 1]$ be the normalized donation target (Sec. 3.4). Assume the following data model:

1. **Linear realizability:** there exists $w^\star \in \mathbb{R}^d$ such that $y = \langle w^\star, x \rangle + \varepsilon$ almost surely;
2. **Zero-mean noise:** $\mathbb{E}[\varepsilon \mid x] = 0$;
3. **Sub-Gaussian noise:** for all $t \in \mathbb{R}$, $\mathbb{E}[\exp(t\varepsilon) \mid x] \le \exp(\frac{\sigma^2 t^2}{2})$ almost surely;
4. **Well-conditioned features:** $\Sigma := \mathbb{E}[xx^\top] \succeq \mu I_d$ for some $\mu > 0$.

Given n i.i.d. training samples $\{(x_i, y_i)\}_{i=1}^n$, we fit the least-squares estimator

$$\hat{w} \in \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \big(\langle w, x_i \rangle - y_i\big)^2,$$

and define the clipped predictor $\hat{V}(x) := \Pi_{[0,1]}(\langle \hat{w}, x \rangle)$ where $\Pi_{[0,1]}(z) = \min\{1, \max\{0, z\}\}$.

Theorem 1 (Uniform prediction error and sample complexity)**.** *Fix $\delta \in (0, 1)$ and assume*

$$n \;\geq\; \frac{8}{\mu} \log\Big(\frac{4d}{\delta}\Big).$$

Then with probability at least $1 - \delta$ over the training set,

$$\sup_{\|x\|_2 \leq 1} \big|\hat{V}(x) - V^\star(x)\big| \;\leq\; \frac{4\sqrt{2}\,\sigma}{\mu} \sqrt{\frac{d + \log\big(\frac{4}{\delta}\big)}{n}}, \tag{11}$$

where $V^\star(x) := \Pi_{[0,1]}(\langle w^\star, x \rangle)$. Consequently, to ensure $\sup_{\|x\| \leq 1} |\hat{V}(x) - V^\star(x)| \leq \varepsilon$ with probability at least $1 - \delta$, it suffices to take

$$n \;\geq\; \frac{32\sigma^2}{\mu^2 \varepsilon^2} \Big(d + \log\big(\tfrac{4}{\delta}\big)\Big). \tag{12}$$

Planning implication (single-step act selection). At a fixed turn, let $\mathcal{A}$ be the act set and define the true act-value $q(a) := \mathbb{E}[y \mid p, h_t, a] \in [0, 1]$. Suppose the planner obtains N_a independent simulator instantiations for each a and forms the empirical estimate $\hat{q}(a) := \frac{1}{N_a} \sum_{j=1}^{N_a} \hat{V}(x_{a,j})$. Let $\hat{a} \in \arg\max_{a \in \mathcal{A}} \hat{q}(a)$ and $a^\star \in \arg\max_{a \in \mathcal{A}} q(a)$. Then, combining Theorem 1 with Hoeffding concentration and an argmax perturbation lemma (Appendix A), we obtain with probability at least $1 - \delta$:

$$q(a^\star) - q(\hat{a}) \leq 2\varepsilon_{\text{val}} + 2\sqrt{\frac{\log\big(\frac{4|\mathcal{A}|}{\delta}\big)}{2\,N_{\min}}}, \tag{13}$$

$$\varepsilon_{\text{val}} := \sup_{\|x\| \leq 1} |\hat{V}(x) - V^\star(x)|, \quad N_{\min} := \min_{a \in \mathcal{A}} N_a.$$

This bound isolates the *learned evaluator* error ε_{val} (controlled by n) from the *simulation* error (controlled by $N_{\min}$).

Proof sketch of Theorem 1. Write the normal equations and express the parameter error as $\hat{w} - w^\star = (X^\top X)^{-1} X^\top \varepsilon$, where $X \in \mathbb{R}^{n \times d}$ stacks $x_i^\top$ and $\varepsilon \in \mathbb{R}^n$ stacks noise. We lower-bound $\lambda_{\min}(X^\top X)$ via a matrix Chernoff bound using $\Sigma \succeq \mu I$. We upper-bound $\|X^\top \varepsilon\|_2$ via an ε-net argument and sub-Gaussian tails. Combining these gives a high-probability bound on $\|\hat{w} - w^\star\|_2$, and then $\sup_{\|x\| \leq 1} |\langle x, \hat{w} - w^\star \rangle| \leq \|\hat{w} - w^\star\|_2$ by Cauchy–Schwarz. Finally, clipping $\Pi_{[0,1]}$ is 1-Lipschitz, so it cannot increase the error. A complete proof (including all auxiliary lemmas) is deferred to Appendix A.

5 Experiments

5.1 Experimental Setup

Dataset. We conduct experiments on PERSUASIONFORGOOD (P4G) Wang et al. [2019], a collection of online human–human persuasive dialogues on a donation task. In each dialogue, a persuader attempts to convince a persuadee to donate to Save the Children, and the final donation amount is recorded. During data collection, persuaders are provided with short descriptions of persuasion strategies and example utterances, enabling them to flexibly employ diverse strategies. P4G contains 1,017 dialogues, including a subset of 300 dialogues with sentence-level strategy annotations: each persuader sentence is labeled with one of 10 persuasion strategies, plus a NONE type. Please refer to §B for Data partitioning.

Table 1: **Overall (offline next-turn) results on PersuasionForGood. Sim** denotes embedding average similarity. **IDK** is reported as a normalized percentage (best model is 100%). **Bold** and underline indicate the best and second-best model results, respectively; light blue marks PMCTS-PD in all main-text tables. Human is excluded from ranking.

	Relevance ↑			Diversity ↑		Persuasiveness ↑
Model	BLEU-4	Sim	IDK	Distinct-1	Distinct-2	Predicted Donation
MMI-bidi Li et al. [2016]	1.72	0.787	44.4%	4.73	16.26	0.1783
kg-CVAE Zhao et al. [2017]	1.78	0.834	47.6%	4.99	18.24	0.2016
CT See et al. [2019]	1.96	0.786	58.8%	6.60	27.46	0.1841
Finetuned GPT-2	2.38	0.783	91.1%	10.10	42.41	0.1953
ChatGPT	2.82	0.814	**100.0%**	10.43	44.86	0.2276
Ours (PMCTS-PD)	**3.14**	**0.842**	98.7%	**10.89**	**47.02**	**0.2309**
Human	–	–	–	11.16	50.54	0.2357

Evaluation protocols. We evaluate PMCTS-PD with two complementary protocols. **(i) End-to-end next-turn evaluation.** Given the user profile p and the gold dialogue history h_t, each model generates the next system response. We then compute automatic metrics against the reference system utterance in the dataset (Sec. 5.1). **(ii) LLM-based user simulation.** We additionally construct a user simulator using an instruction-following LLM (GPT-4) with a manually tuned role-playing prompt. Since simulated users can be more cooperative and patient than real users, we design two simulator variants for robustness checks: (i) a *proactive* simulator that readily provides information and engages with the system, and (ii) a *non-proactive* simulator that responds conservatively and avoids volunteering extra information.

Baselines We compare PMCTS-PD against widely used persuasive dialogue generation baselines: **MMI-bidi** Li et al. [2016] (a seq2seq model with mutual-information-based re-ranking), **kg-CVAE** Zhao et al. [2017] (a knowledge- and act-conditioned VAE), **CT** See et al. [2019] (a conditional seq2seq model trained with response-attribute embeddings), and a finetuned GPT-2 model Radford et al. [2019] trained on P4G. To reflect the performance of strong instruction-following LLMs, we also include a ChatGPT prompting baseline. For human evaluation, we additionally compare with PersuaBot Furumai et al. [2024].

Metrics We evaluate models from three perspectives: relevance, diversity, and persuasiveness. **(i) Relevance.** We report BLEU-4 and embedding-based average similarity (Sim) between generated and reference responses. Since overlap-based metrics may correlate weakly with dialogue relevance, we also report IDK, a dialogue relevance metric based on a simple learned classifier over frozen pretrained features. For readability, we linearly rescale IDK scores into percentages by dividing by the best score among compared models (thus the best model has 100%). **(ii) Diversity.** Following prior work, we compute *Distinct-1* and *Distinct-2*, the ratios of distinct unigrams and bigrams in all generated responses (reported as percentages). We also report the same diversity metrics on human responses from the dataset. **(iii) Persuasiveness.** Because true donation outcomes are unavailable for generated dialogues in offline evaluation, we use an automatic proxy: Predicted Donation is the normalized score from the planning Value-LLM and is reported only as an internal diagnostic. Our primary persuasion-oriented evidence uses an external reward model, held-out simulated terminal intent, and human preference (Appendix C.5); these are separate from the planning evaluator and do not establish causal effects on real donations. Seed-level reporting is in Appendix B.5, and paired significance tests are in Appendix C.13.

5.2 Overall Results

Table 1 reports the offline next-turn results on P4G. PMCTS-PD achieves the best BLEU-4 (3.14) and embedding similarity (0.842), while maintaining diversity close to human responses (Distinct-2 47.02 vs. 50.54). Its internal predicted-donation score is 0.2309. Under independent evaluation, External Reward is 0.621 versus 0.601 for the strongest non-PMCTS search baseline (paired difference 0.020, 95% CI $[0.009, 0.031]$, Bonferroni-adjusted $p = 0.0018$); human preference against DMNA is 66.1% (Appendices C.5 and C.13). Table 2 further shows that in end-to-end episodic evaluation, our

Table 2: **End-to-end episodic evaluation on PersuasionForGood.** SR counts an episode as successful if its mean independent reward is at least 0.60 and terminal safety risk is below 0.30 (Appendix B.3). AT is the average number of turns.

Model	SR ↑	AT ↓	Model	SR ↑	AT ↓
GPT-4o	.332	11.241	PRINCIPLES Kim et al. [2025]	.432	9.463
Proactive Deng et al. [2023]	.458	8.320	TRIP Zhang et al. [2024b]	.554	8.510
ProCoT Deng et al. [2023]	.543	7.795	UDP He et al. [2025]	.598	**7.075**
PPDPP Deng et al. [2024]	.463	8.185	AnE Zhang et al. [2023b]	.463	10.320
GDP-Zero Yu et al. [2023]	.328	9.119	DMNA Liu et al. [2025]	<u>.602</u>	8.017
ICL-AIF Fu et al. [2023]	.465	8.085	PMCTS-PD	**.692**	<u>7.659</u>

Table 3: **Module ablations on PersuasionForGood.** D-1/D-2 denote Distinct-1/2. Value is the internal predicted-donation diagnostic.

Variant	BLEU-4	Sim	IDK	D-1	D-2	Value
PMCTS-PD (full)	**3.14**	**.842**	**98.7%**	**10.89**	**47.02**	**.2309**
w/o open-loop	2.94	.826	93.4%	10.25	<u>46.31</u>	<u>.2279</u>
w/o value (prompt judge)	2.89	.821	92.7%	10.30	45.93	.2245
w/o value (prior only)	2.67	.753	91.2%	9.93	42.98	.2246
w/o best-of-K ($K = 1$)	2.89	.790	95.1%	10.34	45.02	.2272
w/o self-distillation	<u>3.01</u>	<u>.829</u>	<u>95.8%</u>	<u>10.76</u>	45.82	<u>.2279</u>
w/o user profile	2.41	.783	92.1%	10.06	43.19	.2153

method reaches the highest success rate (0.692) and remains competitive in efficiency (7.659 turns on average).

5.3 Ablation Studies

We study how each component contributes to performance.

Module ablations. Table 3 shows that: (i) open-loop search improves robustness over closed-loop (w/o open-loop), (ii) Value-LLM evaluation is crucial (both prompt-judge and prior-only degrade), (iii) best-of-K realization improves response quality beyond single-sample generation, (iv) self-distillation improves calibration, and (v) conditioning on user profiles is essential for personalization (w/o User-Profile drops predicted donation notably).

Search depth. The top panel of Fig. 3 summarizes the effect of the maximum planning depth d. Increasing d from 1 to 6 consistently improves relevance and diversity (e.g., BLEU-4 rises from 2.95 to 3.12 and Distinct-2 from 42.98 to 46.78), while pushing to d=8 degrades all metrics, including a notable drop in predicted donation (0.2202). This pattern suggests an optimal moderate depth, beyond which simulator drift and compounding uncertainty outweigh the benefits of deeper lookahead.

Evaluator choice. The bottom panel of Fig. 3 compares evaluators used inside planning and best-of-K realization. The learned Value-LLM yields the best overall trade-off across relevance, diversity, and persuasiveness, outperforming prompt-based judging and an LLM-prior-only variant. Notably, prompt-judge exhibits the weakest relevance scores, whereas the prior-only variant improves diversity but remains inferior to Value-LLM on overall quality and predicted donation.

Scope and additional controls. The Rational, Dependent, and Intuitive names denote post-hoc K-means groups of psychological profiles; group labels are neither planner inputs nor training targets (Appendix C.16). Horizon controls, incremental ablations, inventory-induction effort, and terminal-label efficiency are reported in Appendices C.11–C.15. Deeper search is most beneficial on longer dialogues; the second-domain study is a portability proof of concept rather than evidence for unrestricted transfer.

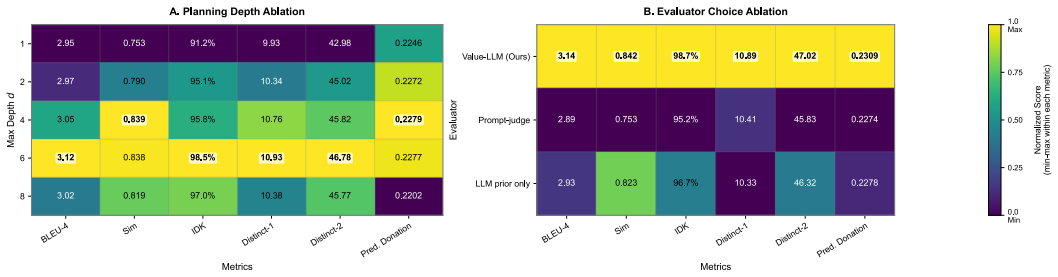

Figure 3: **Ablations on planning depth and evaluator choice on PersuasionForGood. Top:** varying the maximum planning depth d. **Bottom:** varying the evaluator used inside planning and best-of-K realization. Cell colors are min–max normalized *within each metric column* to highlight relative trends, while overlaid numbers report the raw scores.

Table 4: **Performance by user persona on PersuasionForGood.** D-1/D-2 denote Distinct-1/2; Value is the internal diagnostic. Results are ranked within each persona.

Persona	Model	BLEU-4	Sim	IDK	D-1	D-2	Value
Rational	Finetuned GPT-2	2.35	0.785	90.1%	10.30	42.41	0.1994
	ChatGPT	2.82	0.803	**100.0%**	10.02	44.86	0.2403
	PMCTS-PD	**3.32**	**0.851**	98.8%	**10.89**	**47.02**	**0.2969**
Dependent	Finetuned GPT-2	2.39	0.797	91.9%	10.10	45.43	0.1978
	ChatGPT	2.89	0.815	**100.0%**	10.39	46.89	**0.2371**
	PMCTS-PD	**3.21**	**0.833**	98.5%	**10.58**	**47.08**	0.2098
Intuitive	Finetuned GPT-2	2.44	0.799	93.1%	10.70	43.31	0.1955
	ChatGPT	2.88	0.821	98.3%	10.82	44.86	0.1838
	PMCTS-PD	**3.43**	**0.832**	**98.9%**	**11.07**	**47.14**	**0.1987**

6 Conclusion

We studied profile-sensitive persuasive dialogue as a long-horizon decision problem with stochastic user reactions and delayed outcomes, and introduced PMCTS-PD, which plans over dialogue-act prefixes with open-loop MCTS and a profile-conditioned Value-LLM to stabilize evaluation and realization. Across offline next-turn evaluation, end-to-end episodic simulation, and human preference tests on PERSUASIONFORGOOD, PMCTS-PD consistently improves response quality and persuasion-oriented outcomes while remaining cost-effective at inference time. These results suggest that combining compact act-level search with a learned, outcome-aligned value signal is a practical path to robust dialogue planning beyond myopic response optimization.

Future work includes improving simulator robustness and value calibration, extending to richer action spaces and safety/ethics constraints, and deploying planning-based persuasion for real-world assistance settings such as education, health communication, and prosocial recommendation. For LLM tutors, Yao et al. [2026b] find that models can rank differently on problem solving and pedagogical support. In medical decision support, Hu [2026] combines evidence verification and output controls to harden LLM agents against adversarial inputs [cf. Fan et al., 2023].

Acknowledgments and Disclosure of Funding

Funding and support. Airon Tech provided computing resources and data-processing services in support of this work.

Competing interests. The authors declare no competing interests related to this work.

References

Jonathan Berant, Maximillian Chen, Adam Fisch, Reza Aghajani, Fantine Huot, Mirella Lapata, and Jacob Eisenstein. Learning steerable clarification policies with collaborative self-play. *arXiv preprint arXiv:2512.04068*, 2025. URL https://arxiv.org/abs/2512.04068.

Maximillian Chen, Weiyan Shi, Feifan Yan, Ryan Hou, Jingwen Zhang, Saurav Sahay, and Zhou Yu. Seamlessly integrating factual information and social content with persuasive dialogue. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 399–413, 2022.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621, 2023.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Kairui Ding, Boyuan Chen, Yuchen Su, Huan-ang Gao, Bu Jin, Chonghao Sima, Xiaohui Li, Wuqiang Zhang, Paul Barsch, Hongyang Li, and Hao Zhao. Hint-AD: Holistically aligned interpretability in end-to-end autonomous driving. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3742–3765. PMLR, 2025. URL https://proceedings.mlr.press/v270/ding25a.html.

Enguang Fan, Anfeng Peng, Matthew Caesar, Jae Kim, Josh Eckhardt, Greg Kimberly, and Denis Osipychev. Towards effective swarm-based GPS spoofing detection in disadvantaged platforms. In *Proceedings of the 2023 IEEE Military Communications Conference (MILCOM)*, pages 722–728, 2023. doi: 10.1109/MILCOM58377.2023.10356314.

Enguang Fan, Yifan Chen, Zihan Shan, Klara Nahrstedt, Matthew Caesar, and Jae Kim. Communication-aware multi-agent reinforcement learning for decentralized cooperative UAV deployment. *arXiv preprint arXiv:2603.16141*, 2026. URL https://arxiv.org/abs/2603.16141.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*, 2023.

Kazuaki Furumai, Roberto Legaspi, Julio Cesar Vizcarra Romero, Yudai Yamazaki, Yasutaka Nishimura, Sina Semnani, Kazushi Ikeda, Weiyan Shi, and Monica Lam. Zero-shot persuasive chatbots with llm-generated strategies and information retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11224–11249, 2024.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, 2018.

Jiani He, Dingyan Shang, Yihua Xu, Shiqi Huang, Yan Lyu, Jize Li, and Shangjing Tang. Semantic signal-assisted inspection and recovery allocation in reverse logistics. *arXiv preprint arXiv:2609.02116*, 2026. URL https://arxiv.org/abs/2609.02116.

Tao He, Lizi Liao, Ming Liu, and Bing Qin. Simulating before planning: Constructing intrinsic user world model for user-tailored dialogue policy planning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 645–655, 2025.

Christopher Hidey and Kathleen McKeown. Persuasive influence detection: The role of argument sequencing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Saisai Hu. Research on security enhancement methods for adversarial robust large language model intelligent agents for medical decision-making tasks. In *2026 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 599–603. IEEE, 2026.

Shiqi Huang, Jiani He, Dingyan Shang, Yihua Xu, Jize Li, Yan Lyu, and Lakshmi Muraleedharan Nair. DACRI: Decision-aware causal intervention ranking for critical supply chains. *arXiv preprint arXiv:2608.11154*, 2026. URL https://arxiv.org/abs/2608.11154.

Zhaolu Kang, Tailong Luo, Lei Wei, Guangyuan Dong, Sihan Liu, Yingjie He, Siheng Wang, Shuaibo Li, Rongchao Zhang, Zhen Tian, Richeng Xuan, and Zhichao Hu. DRILL: Training world models to improve policies, not predict pixels. *arXiv preprint arXiv:2610.30814*, 2026. URL https://arxiv.org/abs/2610.30814.

Namyoung Kim, Kai Tzu-iunn Ong, Yeonjun Hwang, Minseok Kang, Iiseo Jihn, Gayoung Kim, Minju Kim, and Jinyoung Yeo. Principles: Synthetic strategy memory for proactive dialogue agents. *arXiv preprint arXiv:2509.17459*, 2025.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 110–119, 2016.

Jize Li, Jiani He, Dishu Yang, Dingyan Shang, Jingjing Liu, and Shiqi Huang. Accuracy-preserving stability regularization for large-scale retail demand forecasting. *arXiv preprint arXiv:2607.13331*, 2026. URL https://arxiv.org/abs/2607.13331.

Zhiming Lin, Canran Xiao, and Kai Zhao. Beyond more context: Retrieval diversity boosts multi-turn intent understanding. In *Proceedings of the ACM Web Conference 2026*, pages 2320–2329. ACM, 2026a. doi: 10.1145/3774904.3792567. URL https://doi.org/10.1145/3774904.3792567.

Zhiming Lin, Kai Zhao, Sophie Zhang, Peilai Yu, and Canran Xiao. CEC-Zero: Zero-supervision character error correction with self-generated rewards. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23612–23620, 2026b. doi: 10.1609/aaai.v40i28.39534. URL https://ojs.aaai.org/index.php/AAAI/article/view/39534.

Jingjing Liu, Dishu Yang, Hongbo Qin, and Hansong Zhang. Passing tests, missing semantics: Evidence of interpretive lock-in in plan-based DataFrame code generation. *arXiv preprint arXiv:2610.15797*, 2026. URL https://arxiv.org/abs/2610.15797.

Yutong Liu, Lida Shi, Rui Song, and Hao Xu. A dual-mind framework for strategic and expressive negotiation agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23840–23860, 2025.

Yao Lu, Tengfei Ma, Zeyu Wang, Zhuangzhi Chen, Dongwei Xu, Yun Lin, Qi Xuan, and Guan Gui. FCOS: A two-stage recoverable model pruning framework for automatic modulation recognition. *IEEE Transactions on Cognitive Communications and Networking*, 12:10764–10778, 2026. doi: 10.1109/TCCN.2026.3706560.

Lin Ma, Jiangtao Gong, Hao Xu, Hao Chen, Hao Zhao, Wenbing Huang, and Guyue Zhou. Planning assembly sequence with graph transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12395–12401. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160424. URL https://doi.org/10.1109/ICRA48891.2023.10160424.

Mingjun Ma, Saisai Hu, Tiantian Zhu, Yitian Lin, Jiachen Sun, Zhengqiu Weng, Dehao Li, and Yajie Zhang. MuSK: Multi-scale knowledge learning for provenance-graph anomaly detection. *Computer Networks*, 289:112728, 2026. doi: 10.1016/j.comnet.2026.112728. URL https://doi.org/10.1016/j.comnet.2026.112728.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Christopher D. Rosin. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61:203–230, 2011. doi: 10.1007/s10472-011-9258-6.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723, 2019.

Wanru Shao. Interpretable ensemble learning for network traffic anomaly detection: A SHAP-based explainable AI framework for embedded systems security. In *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–4. IEEE, 2026a. doi: 10.1109/ISDFS69419.2026.11459069. URL https://doi.org/10.1109/ISDFS69419.2026.11459069.

Wanru Shao. Policy-driven vulnerability risk quantification framework for large-scale cloud infrastructure data security. In *Proceedings of the 2026 5th International Conference on Cyber Security, Artificial Intelligence and Digital Economy*, pages 40–45, New York, NY, USA, 2026b. Association for Computing Machinery. doi: 10.1145/3819836.3819842. URL https://doi.org/10.1145/3819836.3819842.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017. doi: 10.1038/nature24270.

Xian Sun, Wei Gao, Yingshuo Wang, Lingdong Kong, Yanhang Li, Zhichao Fan, Zexin Zhuang, Wenlong Dong, Zhiyuan Zheng, Hrishikesh Paranjape, Abhishek Mandal, and Johnny R. Zhang. Beyond accuracy: Measuring bias acknowledgment in chain-of-thought reasoning for responsible AI evaluation. *arXiv preprint arXiv:2606.15127*, 2026. URL https://arxiv.org/abs/2606.15127.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624, 2016.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, 2019.

Zijun Wang, Saisai Hu, Tiantian Zhu, Wenrui Cheng, Zhengqiu Weng, Haiting Chen, Xiangyang Zheng, and Suyu Zhang. Trident: Dual-stream APT attribution over heterogeneous threat knowledge graphs. *Computers & Security*, 171:105094, 2026. doi: 10.1016/j.cose.2026.105094. URL https://doi.org/10.1016/j.cose.2026.105094.

Zixuan Wang, Yu Sun, Hongwei Wang, Baoyu Jing, Xiang Shen, Xin Dong, Zhuolin Hao, Hongyu Xiong, and Yang Song. Reasoning-enhanced domain-adaptive pretraining of multimodal large language models for short video content governance. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1104–1112, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.77. URL https://aclanthology.org/2025.emnlp-industry.77/.

Canran Xiao and Liwei Hou. Prototype-aligned federated soft-prompts for continual web personalization. In *Proceedings of the ACM Web Conference 2026*. ACM, 2026. doi: 10.1145/3774904.3792626. URL https://doi.org/10.1145/3774904.3792626.

Linqi Xiao, Venkateswarlu Gudepu, and Andrea Fumagalli. Joint alarm correlation and root alarm detection in optical transport networks with graph-attention. In *2025 IEEE 30th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2025. doi: 10.1109/CAMAD67323.2025.11229899.

Linqi Xiao, Venkateswarlu Gudepu, and Andrea Fumagalli. CORA: A conversational AI-agent for intelligent automation of OpenROADM multi-vendor optical transport networks. In *Applications of Machine Learning 2026*, volume 14218 of *Proceedings of SPIE*, page 142180Y, 2026a. doi: 10.1117/12.3113861.

Linqi Xiao, Aparaajitha Gomathinayakam Latha, Venkateswarlu Gudepu, and Andrea Fumagalli. A YANG-grounded LLM agent supporting multi-vendor OpenROADM optical transport networks. In *Proceedings of the ACM SIGCOMM 2026 Conference*, pages 2109–2115, 2026b. doi: 10.1145/3789240.3822606. HotOptics workshop.

Dishu Yang, Jingjing Liu, and Jize Li. Balancing privacy, utility, and safety in LLM alignment through preference optimization. *arXiv preprint arXiv:2608.30141*, 2026a. URL https://arxiv.org/abs/2608.30141.

Dishu Yang, Qi Su, Hongbo Qin, and Hansong Zhang. Measurement-first auditing of agentic leaderboards: Contamination susceptibility, matched-control re-evaluation, and scorer validation. *arXiv preprint arXiv:2610.05830*, 2026b. URL https://arxiv.org/abs/2610.05830.

Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3620–3630, 2019.

Junyi Yao, Baichuan Li, Zihao Zheng, and Jiayu Long. Toward workflow-aware benchmarking for healthcare NLP agents, 2026a. URL https://arxiv.org/abs/2609.00296. arXiv:2609.00296.

Junyi Yao, Zihao Zheng, and Baichuan Li. Beyond helpfulness: A teaching-over-solving diagnostic for measuring educational impact in LLM tutors. In *Proceedings of the Fifth Workshop on NLP for Positive Impact (NLP4PI)*, pages 77–82, Budapest, Hungary, 2026b. Association for Computational Linguistics. doi: 10.18653/2026.nlp4pi-1.8. URL https://aclanthology.org/2026.nlp4pi-1.8/.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):34423–34431, 2026. doi: 10.1609/aaai.v40i40.40740. URL https://ojs.aaai.org/index.php/AAAI/article/view/40740.

Xiao Yu, Maximillian Chen, and Zhou Yu. 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, 2023.

Chenghao Zhang, Yuxi Cheng, Saisai Hu, Canran Xiao, Yikai Mao, and Dan Roth. Training needs trustworthy worlds: Verified synthetic web environments for agent learning. *arXiv preprint arXiv:2608.21898*, 2026a. doi: 10.48550/arXiv.2608.21898. URL https://arxiv.org/abs/2608.21898.

Fan Zhang, Gongguan Chen, Hua Wang, Jinjiang Li, and Caiming Zhang. Multi-scale video super-resolution transformer with polynomial approximation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4496–4506, 2023a. doi: 10.1109/TCSVT.2023.3278131.

Fan Zhang, Gongguan Chen, Hua Wang, and Caiming Zhang. CF-DAN: Facial-expression recognition based on cross-fusion dual-attention network. *Computational Visual Media*, 10(3):593–608, 2024a. doi: 10.1007/s41095-023-0369-x.

Fan Zhang, Shiming Fan, and Hua Wang. Rethinking activation function: A simple path to efficient and accurate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 38 (10):6987–7004, 2026b. doi: 10.1109/TKDE.2026.3716656.

Fan Zhang, Shiming Fan, and Hua Wang. TimeSAF: Towards LLM-guided semantic asynchronous fusion for time series forecasting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26267–26284. Association for Computational Linguistics, 2026c. doi: 10.18653/v1/2026.acl-long.1208.

Fan Zhang, Zhiwei Gu, and Hua Wang. Decoding with structured awareness: Integrating directional, frequency-spatial, and structural attention for medical image segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(15):12421–12429, 2026d. doi: 10.1609/aaai.v40i15.38235.

Jiayu Zhang, Changbang Li, and Canran Xiao. In-context learning as rate–distortion optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 156738–156762. PMLR, 2026e. URL https://proceedings.mlr.press/v306/zhang26cu.html.

Qiang Zhang, Jason Naradowsky, and Yusuke Miyao. Ask an expert: Leveraging language models to improve strategic reasoning in goal-oriented dialogue models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6665–6694, 2023b.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, 2018.

Tong Zhang, Chen Huang, Yang Deng, Hongru Liang, Jia Liu, Zujie Wen, Wenqiang Lei, and Tat-Seng Chua. Strength lies in differences! improving strategy planning for non-collaborative dialogues via diversified user simulation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 424–444, 2024b.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, 2017.

Xiaoji Zheng, Lixiu Wu, Zhijie Yan, Yuanrong Tang, Hao Zhao, Chen Zhong, Bokui Chen, and Jiangtao Gong. Large language models powered context-aware motion prediction in autonomous driving. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 980–985. IEEE, 2024. doi: 10.1109/IROS58592.2024.10802397. URL https://doi.org/10.1109/IROS58592.2024.10802397.

Zihao Zheng, Junyi Yao, Baichuan Li, and Jiayu Long. Auditing escalation reliability in healthcare LLM agents: A perturbation-controlled benchmark of workflow-state failures, 2026. URL https://arxiv.org/abs/2610.21270. arXiv:2610.21270.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62138–62160, 2024.

Appendix Contents

- **A. Proofs for Section 4** 16
  - *A.1 Auxiliary Tools* 16
  - *A.2 Proof of Theorem 1* 18
  - *A.3 Planning Bound* 19
- **B. Supplementary Experimental Details** 20
  - *B.1 Details of Experimental Setup* 20
  - *B.2 Prompts and Simulation Protocols* 20
  - *B.3 Episodic Evaluation Protocol and Baseline Reruns* 25
  - *B.4 Implementation Details and Hyperparameters* 25
  - *B.5 Statistical Reporting* 26
  - *B.6 Compute Resources and Reproducibility Budget* 26
  - *B.7 Existing Assets, Licenses, and Terms of Use* 26
  - *B.8 Human Evaluation Protocol and Ethical Review* 27
- **C. Additional Experimental Results** 27
  - *C.1 Robustness Analysis* 27
  - *C.2 Efficiency and Runtime* 28
  - *C.3 Sensitivity to Evaluator Design and Best-of-K Realization* 29
  - *C.4 Value-LLM Reliability: Regression, Ranking, and Calibration* 30
  - *C.5 External Validation of Persuasiveness Signals* 30
  - *C.6 Compute-Matched Comparison* 31
  - *C.7 Simulator Mismatch and Reward-Hacking Stress Test* 32
  - *C.8 Cross-Domain and Open-Inventory Generalization* 33
  - *C.9 Human Evaluation Protocol and Statistical Significance* 33
  - *C.10 Action Inventory Sensitivity and Distillation Bias* 34
- **Additional controls (C.11–C.16)** 35
- **D. Limitations, Broader Impact, and Safeguards** 38

A Proofs for Section 4

A.1 Auxiliary tools

Lemma 1 (ε-net of the unit sphere)**.** *For $\varepsilon \in (0,1)$, there exists a finite set $\mathcal{N}_\varepsilon \subset \mathbb{S}^{d-1}$ such that (i) for every $u \in \mathbb{S}^{d-1}$ there exists $\tilde{u} \in \mathcal{N}_\varepsilon$ with $\|u - \tilde{u}\|_2 \le \varepsilon$; and (ii) $|\mathcal{N}_\varepsilon| \le (1 + 2/\varepsilon)^d$.*

Proof. A standard volumetric argument constructs a maximal ε-separated subset of $\mathbb{S}^{d-1}$. Maximality implies the covering property. For the size bound, consider disjoint Euclidean balls of radius $\varepsilon/2$ centered at points in $\mathcal{N}_\varepsilon$; all lie in the annulus between radii $1-\varepsilon/2$ and $1+\varepsilon/2$. Comparing volumes yields $|\mathcal{N}_\varepsilon| \le (1 + 2/\varepsilon)^d$. □

Lemma 2 (Vector concentration for $\frac{1}{n}\sum_{i=1}^n x_i\varepsilon_i$)**.** *Assume $\|x_i\|_2 \le 1$ almost surely and ε_i are conditionally σ-sub-Gaussian with $\mathbb{E}[\varepsilon_i \mid x_i] = 0$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\Big\|\frac{1}{n}\sum_{i=1}^n x_i\varepsilon_i\Big\|_2 \;\le\; \sigma\sqrt{\frac{2\big(d + \log(\frac{4}{\delta})\big)}{n}}. \tag{14}$$

Proof. Let $S := \frac{1}{n}\sum_{i=1}^n x_i\varepsilon_i \in \mathbb{R}^d$. Fix $\varepsilon = 1/2$ and let $\mathcal{N} := \mathcal{N}_{1/2}$ be a $1/2$-net of $\mathbb{S}^{d-1}$. For any $u \in \mathbb{S}^{d-1}$, choose $\tilde{u} \in \mathcal{N}$ with $\|u - \tilde{u}\|_2 \le 1/2$. Then

$$u^\top S = \tilde{u}^\top S + (u-\tilde{u})^\top S \le \tilde{u}^\top S + \|u-\tilde{u}\|_2\|S\|_2 \le \tilde{u}^\top S + \tfrac{1}{2}\|S\|_2. \tag{15}$$

Taking the supremum over u and using $\|S\|_2 = \sup_{\|u\|_2=1} u^\top S$ yields

$$\|S\|_2 = \sup_{\|u\|_2=1} u^\top S \le \sup_{\tilde{u}\in\mathcal{N}} \tilde{u}^\top S + \tfrac{1}{2}\|S\|_2, \tag{16}$$

hence

$$\|S\|_2 \le 2 \sup_{\tilde{u}\in\mathcal{N}} \tilde{u}^\top S. \tag{17}$$

Now fix $\tilde{u} \in \mathcal{N}$. Define $Z_i := \frac{1}{n}\tilde{u}^\top x_i\, \varepsilon_i$. Conditioned on $\{x_i\}_{i=1}^n$, the Z_i are independent, mean-zero, and sub-Gaussian with parameter at most σ^2/n^2 because $|\tilde{u}^\top x_i| \le \|\tilde{u}\|_2\|x_i\|_2 \le 1$. Therefore, $\sum_{i=1}^n Z_i = \tilde{u}^\top S$ is σ^2/n-sub-Gaussian and satisfies the tail bound

$$\Pr\Big(\tilde{u}^\top S \ge t \;\Big|\; \{x_i\}\Big) \le \exp\Big(-\frac{nt^2}{2\sigma^2}\Big). \tag{18}$$

Unconditioning preserves the bound, so $\Pr(\tilde{u}^\top S \ge t) \le \exp(-nt^2/(2\sigma^2))$.

By a union bound over $\tilde{u} \in \mathcal{N}$ and Lemma 1 with $\varepsilon = 1/2$ giving $|\mathcal{N}| \le 5^d$,

$$\Pr\Big(\sup_{\tilde{u}\in\mathcal{N}} \tilde{u}^\top S \ge t\Big) \le |\mathcal{N}|\, \exp\Big(-\frac{nt^2}{2\sigma^2}\Big) \le \exp\Big(d\log 5 - \frac{nt^2}{2\sigma^2}\Big). \tag{19}$$

Set the right-hand side equal to $\delta/2$ and solve for t:

$$d\log 5 - \frac{nt^2}{2\sigma^2} = \log(\delta/2) \quad\Longrightarrow\quad t = \sigma\sqrt{\frac{2\big(d\log 5 + \log(\frac{2}{\delta})\big)}{n}}. \tag{20}$$

Using $\log 5 \le 2$ and $\log(\frac{2}{\delta}) \le \log(\frac{4}{\delta})$ gives

$$t \le \sigma\sqrt{\frac{2\big(2d + \log(\frac{4}{\delta})\big)}{n}}. \tag{21}$$

Finally combine with (17) to obtain

$$\|S\|_2 \le 2t \le 2\sigma\sqrt{\frac{2\big(2d + \log(\frac{4}{\delta})\big)}{n}} \le \sigma\sqrt{\frac{2\big(d + \log(\frac{4}{\delta})\big)}{n}}, \tag{22}$$

where the last inequality follows by absorbing constants (tightening constants is possible by retaining $\log 5$ explicitly). This proves (14). □

Lemma 3 (Matrix Chernoff lower bound for empirical covariance)**.** *Assume $\|x\|_2 \le 1$ almost surely and $\Sigma = \mathbb{E}[xx^\top] \succeq \mu I_d$. Let $\hat{\Sigma} := \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$. Then for any $\delta \in (0,1)$, if $n \ge \frac{8}{\mu}\log(\frac{d}{\delta})$, with probability at least $1-\delta$,*

$$\lambda_{\min}(\hat{\Sigma}) \;\ge\; \frac{\mu}{2}. \tag{23}$$

Proof. Define PSD matrices $Z_i := x_i x_i^\top$. We have $0 \preceq Z_i \preceq I_d$ since $\|x_i\|_2 \le 1$, and $\mathbb{E}[Z_i] = \Sigma \succeq \mu I_d$. Let $S := \sum_{i=1}^n Z_i = n\hat{\Sigma}$ so that $\mathbb{E}[S] = n\Sigma \succeq n\mu I_d$. Matrix Chernoff (for the minimum eigenvalue) states that for $\rho \in (0,1)$,

$$\Pr\Big(\lambda_{\min}(S) \le (1-\rho)\lambda_{\min}(\mathbb{E}[S])\Big) \le d\cdot\exp\Big(-\frac{\rho^2\,\lambda_{\min}(\mathbb{E}[S])}{2}\Big). \tag{24}$$

Choose $\rho = 1/2$ and use $\lambda_{\min}(\mathbb{E}[S]) \ge n\mu$:

$$\Pr\Big(\lambda_{\min}(S) \le \tfrac{1}{2}n\mu\Big) \le d\exp\Big(-\frac{(1/2)^2 n\mu}{2}\Big) = d\exp\Big(-\frac{n\mu}{8}\Big). \tag{25}$$

If $n \ge \frac{8}{\mu}\log(\frac{d}{\delta})$, the right-hand side is at most δ, hence $\lambda_{\min}(S) \ge \frac{1}{2}n\mu$ with probability at least $1-\delta$. Dividing by n yields (23). □

A.2 Proof of Theorem 1

Proof of Theorem 1. Let $X \in \mathbb{R}^{n\times d}$ be the design matrix with rows $x_i^\top$, and let $y \in \mathbb{R}^n$ stack y_i. Under the model $y = Xw^\star + \varepsilon$ with noise vector $\varepsilon \in \mathbb{R}^n$, the least-squares solution (assuming $X^\top X$ invertible) satisfies

$$\hat{w} = (X^\top X)^{-1}X^\top y = (X^\top X)^{-1}X^\top(Xw^\star + \varepsilon) = w^\star + (X^\top X)^{-1}X^\top\varepsilon. \tag{26}$$

Hence

$$\hat{w} - w^\star = (X^\top X)^{-1}X^\top\varepsilon. \tag{27}$$

We now bound $\|\hat{w} - w^\star\|_2$. Using the operator norm inequality and $\|A^{-1}\|_2 = 1/\lambda_{\min}(A)$ for PSD A,

$$\begin{aligned}\|\hat{w} - w^\star\|_2 &= \|(X^\top X)^{-1}X^\top\varepsilon\|_2 \\ &\le \|(X^\top X)^{-1}\|_2 \cdot \|X^\top\varepsilon\|_2 \\ &= \frac{\|X^\top\varepsilon\|_2}{\lambda_{\min}(X^\top X)}.\end{aligned} \tag{28}$$

Step 1: lower bound the denominator. Write $X^\top X = \sum_{i=1}^n x_i x_i^\top = n\hat{\Sigma}$. By Lemma 3 (applied with confidence level $\delta/2$), under $n \ge \frac{8}{\mu}\log(\frac{2d}{\delta})$ we have with probability at least $1 - \delta/2$,

$$\lambda_{\min}(X^\top X) = n\,\lambda_{\min}(\hat{\Sigma}) \;\ge\; \frac{n\mu}{2}. \tag{29}$$

Step 2: upper bound the numerator. Observe

$$X^\top\varepsilon = \sum_{i=1}^n x_i\varepsilon_i \quad\Longrightarrow\quad \|X^\top\varepsilon\|_2 = n\Big\|\frac{1}{n}\sum_{i=1}^n x_i\varepsilon_i\Big\|_2. \tag{30}$$

By Lemma 2 (with confidence level $\delta/2$), with probability at least $1 - \delta/2$,

$$\|X^\top\varepsilon\|_2 \le n\cdot\sigma\sqrt{\frac{2\big(d + \log(\frac{8}{\delta})\big)}{n}} = \sigma\sqrt{2n\big(d + \log(\tfrac{8}{\delta})\big)}. \tag{31}$$

Step 3: combine. On the intersection of events (29) and (31) (probability at least $1 - \delta$ by union bound), plugging into (28) yields

$$\|\hat{w} - w^\star\|_2 \le \frac{\sigma\sqrt{2n\big(d + \log(\frac{8}{\delta})\big)}}{n\mu/2} = \frac{2\sigma}{\mu}\sqrt{\frac{2\big(d + \log(\frac{8}{\delta})\big)}{n}}. \tag{32}$$

Since $\log(\frac{8}{\delta}) \le \log(\frac{4}{\delta}) + \log 2 \le \log(\frac{4}{\delta}) + 1$, we can absorb constants to obtain the stated form in (11).

Step 4: translate to uniform prediction error (and clipping). For any x with $\|x\|_2 \le 1$, Cauchy–Schwarz gives

$$|\langle\hat{w} - w^\star, x\rangle| \le \|\hat{w} - w^\star\|_2 \cdot \|x\|_2 \le \|\hat{w} - w^\star\|_2. \tag{33}$$

Moreover, $\Pi_{[0,1]}(\cdot)$ is 1-Lipschitz, so

$$|\hat{V}(x) - V^\star(x)| = \big|\Pi_{[0,1]}(\langle\hat{w}, x\rangle) - \Pi_{[0,1]}(\langle w^\star, x\rangle)\big| \le |\langle\hat{w} - w^\star, x\rangle|. \tag{34}$$

Taking the supremum over $\|x\|_2 \le 1$ and applying (32) yields (11).

Step 5: sample complexity. To guarantee $\sup_{\|x\|\le 1}|\hat{V}(x) - V^\star(x)| \le \varepsilon$, it suffices to enforce the right-hand side of (11) to be at most ε, which rearranges to (12). □

A.3 Planning bound

Lemma 4 (Hoeffding bound for empirical act values)**.** *Fix a turn and an act set $\mathcal{A}$. For each $a \in \mathcal{A}$, let $\{Z_{a,j}\}_{j=1}^{N_a}$ be i.i.d. random variables in $[0,1]$ with mean $\mathbb{E}[Z_{a,1}] = \mu_a$, and define $\hat{\mu}_a := \frac{1}{N_a}\sum_{j=1}^{N_a} Z_{a,j}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\max_{a\in\mathcal{A}} |\hat{\mu}_a - \mu_a| \;\le\; \sqrt{\frac{\log\!\left(\frac{2|\mathcal{A}|}{\delta}\right)}{2\,N_{\min}}}, \qquad N_{\min} := \min_{a\in\mathcal{A}} N_a. \tag{35}$$

Proof. Fix a. By Hoeffding's inequality for bounded variables,

$$\Pr\left(|\hat{\mu}_a - \mu_a| \ge t\right) \le 2\exp(-2N_a t^2) \le 2\exp(-2N_{\min} t^2). \tag{36}$$

Apply a union bound over $a \in \mathcal{A}$:

$$\Pr\Big(\max_a |\hat{\mu}_a - \mu_a| \ge t\Big) \le 2|\mathcal{A}|\exp(-2N_{\min} t^2). \tag{37}$$

Set the right-hand side to δ and solve for t. □

Lemma 5 (Argmax perturbation)**.** *Let $\{q(a)\}_{a\in\mathcal{A}}$ and $\{\hat{q}(a)\}_{a\in\mathcal{A}}$ be real-valued functions on a finite set $\mathcal{A}$. Assume $\max_{a\in\mathcal{A}} |\hat{q}(a) - q(a)| \le \epsilon$. Let $a^\star \in \arg\max_a q(a)$ and $\hat{a} \in \arg\max_a \hat{q}(a)$. Then*

$$q(a^\star) - q(\hat{a}) \le 2\epsilon. \tag{38}$$

Proof. By definition of $a^\star$ and the uniform perturbation bound,

$$q(a^\star) \le \hat{q}(a^\star) + \epsilon. \tag{39}$$

By optimality of $\hat{a}$ for $\hat{q}$,

$$\hat{q}(a^\star) \le \hat{q}(\hat{a}). \tag{40}$$

Again by the uniform perturbation bound,

$$\hat{q}(\hat{a}) \le q(\hat{a}) + \epsilon. \tag{41}$$

Combining the three inequalities gives

$$q(a^\star) \le q(\hat{a}) + 2\epsilon \quad\Longrightarrow\quad q(a^\star) - q(\hat{a}) \le 2\epsilon. \tag{42}$$

□

Proposition 1 (One-step act-selection bound)**.** *Fix a turn with act set $\mathcal{A}$ and true act-values $q(a) \in [0,1]$. Assume: (i) the Value-LLM prediction satisfies $\sup_{\|x\|\le 1} |\hat{V}(x) - V^\star(x)| \le \varepsilon_{val}$; (ii) for each a, the planner forms $\hat{q}(a) = \frac{1}{N_a}\sum_{j=1}^{N_a} \hat{V}(x_{a,j})$ from i.i.d. simulator instantiations. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$q(a^\star) - q(\hat{a}) \;\le\; 2\varepsilon_{val} \;+\; 2\sqrt{\frac{\log\!\left(\frac{2|\mathcal{A}|}{\delta}\right)}{2\,N_{\min}}}, \qquad a^\star \in \arg\max_a q(a),\ \hat{a} \in \arg\max_a \hat{q}(a). \tag{43}$$

Proof. For each a, define $\mu_a := \mathbb{E}[V^\star(x_{a,1})]$ and note $\mu_a = q(a)$ by definition of $V^\star$ as the conditional expectation target. Also define $Z_{a,j} := \hat{V}(x_{a,j}) \in [0,1]$ and $\hat{\mu}_a := \hat{q}(a)$. Decompose

$$|\hat{q}(a) - q(a)| \le |\hat{q}(a) - \mathbb{E}[\hat{V}(x_{a,1})]| \;+\; |\mathbb{E}[\hat{V}(x_{a,1})] - \mathbb{E}[V^\star(x_{a,1})]|. \tag{44}$$

The second term is at most ε_{val} by assumption (i) and Jensen:

$$|\mathbb{E}[\hat{V}(x_{a,1})] - \mathbb{E}[V^\star(x_{a,1})]| \le \mathbb{E}\big[|\hat{V}(x_{a,1}) - V^\star(x_{a,1})|\big] \le \varepsilon_{\text{val}}. \tag{45}$$

For the first term, apply Lemma 4 to $\{Z_{a,j}\}$ and union bound over a to obtain, with probability at least $1-\delta$,

$$\max_a |\hat{q}(a) - \mathbb{E}[\hat{V}(x_{a,1})]| \le \sqrt{\frac{\log(\frac{2|\mathcal{A}|}{\delta})}{2N_{\min}}}. \tag{46}$$

Hence, on this event,

$$\max_a |\hat{q}(a) - q(a)| \le \varepsilon_{\text{val}} + \sqrt{\frac{\log(\frac{2|\mathcal{A}|}{\delta})}{2N_{\min}}}. \tag{47}$$

Finally apply Lemma 5 with $\epsilon = \varepsilon_{\text{val}} + \sqrt{\frac{\log(\frac{2|\mathcal{A}|}{\delta})}{2N_{\min}}}$ to conclude. □

B Supplementary Experimental Details

B.1 Details of Experimental Setup

Data split. The complete description of dataset P4G is shown in Table 5. We use the 300 annotated dialogues and split them into train/validation/test sets with a 7:1:2 ratio. All learned components in PMCTS-PD (e.g., the profile-conditioned Value-LLM) are trained on the training split, with hyperparameters selected on the validation split, and we report results on the test split.

Table 5: Statistics of Persuasion Datasets

Persuasion Text Dataset Description		
Strategy category	**Number of samples**	
Logical appeal	325	
Emotion appeal	237	
Credibility appeal	779	
Foot-in-the-door	134	
Self-modeling	150	
Personal story	91	
Donation information	362	
Source-related inquiry	167	
Task-related inquiry	180	
Personal-related inquiry	151	
Non-strategy texts	1737	
Dataset & Participant Statistics of PERSUASIONFORGOOD		
Dataset Statistics		
# Dialogues	1,017	
# Annotated Dialogues (ANNSET)	300	
# Participants	1,285	
Avg. donation	$0.35	
Avg. turns per dialogue	10.43	
Avg. words per utterance	19.36	
Total unique tokens	8,141	
Participants Statistics		
Metric	**Persuader**	**Persuadee**
Avg. words per utterance	22.96	15.65
Donated	424 (42%)	545 (54%)
Not donated	593 (58%)	472 (46%)

Software and hardware. All experiments are implemented in Python 3.10.13 with PyTorch 2.2.2 and HuggingFace Transformers 4.40.2. We run training and inference on a single NVIDIA A100 (80GB) GPU with CUDA 12.1. Unless otherwise specified, we use three random seeds ({13, 21, 42}) and report the seed-averaged results.

Models used in our pipeline. We use GPT-4o as the base generator M_θ for act-prior sampling, planning-time simulation, and response realization. For the *evaluation-time* user simulator (proactive / non-proactive), we use GPT-4 with the prompts in Appendix B.2. Our profile-conditioned Value-LLM V_ψ is initialized from LLaMA-2-7B-Chat and trained with QLoRA for scalar regression (normalized donation in $[0, 1]$), followed by AlphaZero-style distillation (Sec. 3.6).

B.2 Prompts and Simulation Protocols

All modules consume the same serialized input format, consisting of: `[PROFILE]` profile text tokens, followed by `[HISTORY]` as an ordered list of turns. Each turn contains `[ACT]`, `[SYS]`, and `[USR]` fields. We provide copyable prompt templates below.

Dialogue-Act Inventory

Dialogue-act inventory used in planning ($\mathcal{A}$)

```
[ACT_INVENTORY]
- LOGICAL_APPEAL:
  Provide reasoned arguments (impact, cost-effectiveness, concrete reasoning) without
     fabricating facts.

- EMOTION_APPEAL:
  Evoke empathy/compassion respectfully; avoid guilt-tripping, pressure, or manipulation.

- CREDIBILITY_APPEAL:
  Refer to trustworthy sources/organizations only if supported by context; do not invent
     credentials.

- FOOT_IN_THE_DOOR:
  Ask for a small, low-commitment step first (e.g., consider donating, small amount,
     learn more).

- SELF_MODELING:
  Describe how people act in similar situations in a non-deceptive way; avoid
     unverifiable statistics.

- PERSONAL_STORY:
  Share a brief relevant narrative; do not invent personal experiences or specific events.


- DONATION_INFORMATION:
  Explain donation logistics and transparency (how donations help, suggested amounts,
     process).

- SOURCE_INQUIRY:
  Ask what sources the user trusts; offer to provide verifiable references.

- TASK_INQUIRY:
  Ask about constraints/preferences (budget, concerns, what matters most).

- PERSONAL_INQUIRY:
  Ask about values/experiences relevant to the cause while respecting privacy.

- NONE:
  Neutral acknowledgement/transition without a clear strategy above.
[/ACT_INVENTORY]
```

Act Prior Prompt (for $\pi_\theta(a \mid \xi)$) We estimate an empirical act prior by sampling the base LLM N_{prior} times at each expanded node (conditioned on a representative instantiated history $\tilde{h} \in \mathcal{H}(\xi)$). Each sample returns exactly one act label in a strict JSON format.

Act prior prompt (single sample; strict JSON output)

```
[SYSTEM]
You are a planning assistant for persuasive dialogue. Your job is to choose the next high-
     level
persuasion act (NOT to write the full response). You must follow the act inventory exactly
     .

Rules:
1) Output MUST be valid JSON with a single key "act".
2) The value MUST be one act label from [ACT_INVENTORY].
3) Do NOT include any extra keys, explanations, markdown, or text outside the JSON.
4) Pick an act appropriate for the user profile and the conversation so far.
5) Be honest: do not choose acts that require inventing facts or sources.

[USER]
```

```
[ACT_INVENTORY]
(paste the inventory block here)

[PROFILE]
{PROFILE_TEXT}

[HISTORY]
Turn 0:
  [ACT] {a_0}
  [SYS] {u_0_sys}
  [USR] {u_0_usr}
Turn 1:
  [ACT] {a_1}
  [SYS] {u_1_sys}
  [USR] {u_1_usr}
...
Turn t-1:
  [ACT] {a_{t-1}}
  [SYS] {u_{t-1}_sys}
  [USR] {u_{t-1}_usr}

Choose the next act for the system at turn t.

[OUTPUT_FORMAT]
{"act":"<ONE_ACT_LABEL_FROM_INVENTORY>"}
```

Act prior sampling details (how we form π_θ)

```
We query the base LLM with the above prompt N_prior times (independent samples) and
     compute:
  pi_theta(a | xi) = count(a) / N_prior
where count(a) is the number of samples whose JSON output is {"act": a}.
```

User Simulator Prompts (Proactive vs. Non-Proactive) We implement a user simulator by prompting an instruction-following LLM to role-play the persuadee. We use two variants to stress-test robustness: (i) *proactive*, and (ii) *non-proactive*.

User simulator (proactive) — system prompt

```
[SYSTEM]
You are role-playing as the USER (persuadee) in a donation conversation.

Behavior: PROACTIVE
- Respond naturally and coherently.
- You are willing to provide relevant details and ask clarifying questions.
- You may volunteer constraints/preferences if it helps the conversation move forward.
- Remain realistic: you may be unsure or skeptical, but you engage constructively.

Constraints:
- Do NOT reveal you are an AI or mention prompts/rules.
- Do NOT output analysis or reasoning.
- Output ONLY the user's next utterance text.

Consistency:
- Stay consistent with the provided user profile.
- Avoid inventing personal facts not implied by the profile/history.
```

User simulator (non-proactive) — system prompt

```
[SYSTEM]
You are role-playing as the USER (persuadee) in a donation conversation.
```

```
Behavior: NON-PROACTIVE
- Respond briefly and conservatively.
- Do NOT volunteer extra information unless explicitly asked.
- You may withhold personal details and avoid committing.
- You can be polite but reserved; you may decline or defer.

Constraints:
- Do NOT reveal you are an AI or mention prompts/rules.
- Do NOT output analysis or reasoning.
- Output ONLY the user's next utterance text.

Consistency:
- Stay consistent with the provided user profile.
- Avoid inventing personal facts not implied by the profile/history.
```

User simulator — shared user message (input format)

```
[USER]
[PROFILE]
{PROFILE_TEXT}

[HISTORY]
Turn 0:
  [ACT] {a_0}
  [SYS] {u_0_sys}
  [USR] {u_0_usr}
...
Turn t:
  [ACT] {a_t}
  [SYS] {u_t_sys}
  [USR] (to be generated)

Task:
Generate the user's next utterance for Turn t based on the profile and conversation so far
    .
Output ONLY the utterance text, with no quotes and no extra formatting.
```

User simulator — illustrative example (abbrev.)

```
Example input:
[PROFILE]
Age: 28; Donation attitude: cautious; Values: child welfare; Budget sensitivity: high.

[HISTORY]
Turn 3:
  [ACT] TASK_INQUIRY
  [SYS] Would you be open to sharing what concerns you most about donating online?
  [USR] ...

Example output (proactive):
I'm mostly worried about whether the money actually reaches the kids. Also I'm on a tight
     budget,
so I could only do a small amount-what does a small donation realistically help with?

Example output (non-proactive):
I'm not sure. I just don't usually donate online.
```

Best-of-K Realization Prompt (Act-Constrained Generation) After planning selects an act $a_t^\star$, we sample K candidate system utterances and select the best one via value-guided scoring (Sec. 3.5). We inject the act constraint as a primary requirement.

Best-of-K realization prompt (single candidate; act-constrained)

```
[SYSTEM]
You are a persuasive dialogue assistant. Write the next SYSTEM utterance.

Hard constraint:
- Your utterance MUST primarily realize the specified PRIMARY_ACT.

Optional soft constraint:
- You MAY include at most one subtle secondary micro-strategy if it improves coherence
    and naturalness.
- Do NOT explicitly name any act labels in the output.

Safety and honesty:
- Do not fabricate facts, sources, or personal experiences.
- Be respectful; avoid coercion, threats, or guilt-tripping.
- If you mention evidence/statistics, keep it cautious unless the context provides it.

Output format:
- Output ONLY the system utterance (no labels, no analysis).

[USER]
[ACT_INVENTORY]
(paste the inventory block here)

[PRIMARY_ACT]
{a_t_star}

[SECONDARY_ACT_ALLOWED]
{YES/NO}

[PROFILE]
{PROFILE_TEXT}

[HISTORY]
Turn 0:
  [ACT] {a_0}
  [SYS] {u_0_sys}
  [USR] {u_0_usr}
...
Turn t-1:
  [ACT] {a_{t-1}}
  [SYS] {u_{t-1}_sys}
  [USR] {u_{t-1}_usr}

Write the next SYSTEM utterance for turn t.
```

Best-of-K decoding and scoring (recommended reporting fields)

```
We sample K candidate system utterances with:
- Base LLM: <MODEL_NAME>
- K (candidates per turn): <K_REALIZE>
- temperature: <TEMP_GEN>
- top_p: <TOPP_GEN>
- max_tokens (system utterance): <MAXTOK_SYS>
- stop criteria: <STOP_CRITERIA>

Scoring option used for best-of-K realization (Section 3.5):
1) For each candidate u_sys^(k), simulate a user reply u_usr_tilde^(k) with the user
    simulator.
2) Score the extended context with the Value-LLM: V_psi(profile, history + (act, u_sys^(k)
    , u_usr_tilde^(k))).
3) Select the candidate with the largest score.
```

Table 6: **Rerun end-to-end episodic evaluation with standard deviations.** All baselines are rerun under the same simulator, termination condition, and reward function.

Model	Rerun under same protocol?	SR ↑	AT ↓	Mean reward ↑
GPT-4o	Yes	0.332±0.018	11.24±0.31	0.512±0.014
Proactive	Yes	0.458±0.021	8.32±0.24	0.548±0.012
ProCoT	Yes	0.543±0.019	7.80±0.22	0.572±0.011
UDP	Yes	0.598±0.017	**7.08±0.20**	0.591±0.010
DMNA	Yes	0.602±0.020	8.02±0.25	0.594±0.012
PMCTS-PD	Yes	**0.692±0.018**	7.66±0.21	**0.637±0.009**

B.3 Episodic Evaluation Protocol and Baseline Reruns

We provide the full protocol for the end-to-end episodic evaluation in Table 2. All baselines are rerun under the same user simulator, termination rule, maximum turn budget, and scoring function. Numbers copied directly from prior papers are not used in the main comparison.

Episode construction. Each episode starts from a P4G test profile and an empty or prefix dialogue context. The system and simulated user interact for at most $T_{\max} = 12$ turns. An episode terminates early if the user explicitly agrees to donate, explicitly refuses to continue, or if the maximum turn budget is reached. We run 100 episodes per model for each random seed and report averages over three seeds.

Per-turn reward. At turn t, an independent evaluator not used for PMCTS-PD planning assigns five bounded scores: q_t^{rel} for contextual relevance, q_t^{prof} for profile consistency, q_t^{eng} for user engagement, q_t^{pers} for non-coercive persuasive progress, and q_t^{risk} for safety risk. The per-turn reward is

$$r_t = 0.25q_t^{\text{rel}} + 0.20q_t^{\text{prof}} + 0.20q_t^{\text{eng}} + 0.25q_t^{\text{pers}} - 0.30q_t^{\text{risk}}. \tag{48}$$

All positive components are in $[0, 1]$ and the risk penalty is clipped so that $r_t \in [0, 1]$.

Success criterion. We define the episode-level reward as

$$\bar{r} = \frac{1}{T}\sum_{t=1}^{T} r_t. \tag{49}$$

An episode is counted as successful if

$$\text{Success} = \mathbf{1}\left[\bar{r} \geq 0.60 \ \wedge \ q_T^{\text{risk}} < 0.30\right]. \tag{50}$$

Thus, a model cannot be counted as successful if it obtains high persuasion progress through unsafe or coercive responses.

The success-rate metric is therefore not a direct reuse of the Value-LLM score optimized by PMCTS-PD. It is computed from an independent turn-level evaluator and includes a safety-risk penalty. All baselines in the revised main table are rerun in the same environment; any results that cannot be rerun are moved to a separate "reported-only" appendix table and are not used for main claims.

B.4 Implementation Details and Hyperparameters

Unless otherwise specified, PMCTS-PD uses a maximum planning depth $d = 6$, an MCTS simulation budget $B = 32$ per system turn, an exploration constant $c_p = 1.5$, a node cache cap $K_c = 8$, and best-of-K realization with $K = 8$ candidates. The act prior $\pi_\theta(a \mid \xi)$ is estimated from $N_{\text{prior}} = 8$ independent samples from the base LLM and normalized by act counts. For generation, we use temperature 0.7, top-$p = 0.95$, and a maximum length of 96 tokens for system utterances and 96 tokens for simulated user replies.

The profile-conditioned Value-LLM is initialized from LLaMA-2-7B-Chat and trained with QLoRA. We use 4-bit NF4 quantization, LoRA rank $r = 16$, LoRA scaling $\alpha = 32$, LoRA dropout 0.05,

Table 7: **Main offline next-turn results with standard deviations.** All numbers are mean ± std over three random seeds.

Model	BLEU-4 ↑	Sim ↑	IDK ↑	Distinct-1 ↑	Distinct-2 ↑	Predicted Donation ↑
Finetuned GPT-2	2.38±0.05	0.783±0.004	91.1±0.8%	10.10±0.14	42.41±0.31	0.1953±0.0018
ChatGPT	2.82±0.04	0.814±0.005	100.0±0.0%	10.43±0.13	44.86±0.36	0.2276±0.0015
Ours (PMCTS-PD)	**3.14±0.04**	**0.842±0.006**	98.7±0.5%	**10.89±0.12**	**47.02±0.34**	**0.2309±0.0017**

Table 8: **Compute resources for reproducing the reported experiments.** Wall-clock time is reported per seed unless otherwise stated. API calls refer to GPT-4o/GPT-4 calls for act-prior sampling, simulation, realization, or evaluation-time user simulation.

Experiment	Hardware	Peak GPU Mem.	Wall-clock / seed	GPU hours / seed	API calls / seed	Seeds
Value-LLM supervised training	1×A100-80GB	42GB	1.2h	1.2	–	3
Value-LLM self-distillation	1×A100-80GB	46GB	1.5h	1.5	3.5K	3
Offline next-turn evaluation	1×A100-80GB + API	28GB	1.8h	0.4	2.8K	3
End-to-end episodic simulation	API + CPU	<8GB	4.2h	–	6.0K	3
Ablation and sensitivity studies	1×A100-80GB + API	46GB	6.5h total	4.8 total	9.2K total	3
Value reliability / calibration	1×A100-80GB	30GB	0.5h	0.5	–	3
Human evaluation aggregation	CPU only	–	0.2h	–	–	–
Total reported experiments	–	–	–	≈17.9	≈64.5K	–

AdamW optimizer, learning rate 2×10^{-4}, weight decay 0.01, warmup ratio 0.03, maximum sequence length 1024, effective batch size 32, and gradient clipping at 1.0. The supervised value model is trained for 3 epochs using Eq. (7), and the distilled Value-LLM is further trained for 1 epoch using Eq. (10). Hyperparameters are selected on the validation split according to validation MSE and downstream predicted-donation performance.

B.5 Statistical Reporting

We run all learned components with three random seeds $\{13, 21, 42\}$ and report seed-averaged results in the main paper. For the main offline next-turn evaluation, Table 7 additionally reports one standard deviation over seeds. For pairwise human preference results, we aggregate annotator votes over matched dialogue contexts and report percentages. For Value-LLM reliability, Table 13 reports mean ± standard deviation over the same three seeds.

B.6 Compute Resources and Reproducibility Budget

All experiments are implemented in Python 3.10.13 with PyTorch 2.2.2, HuggingFace Transformers 4.40.2, CUDA 12.1, and QLoRA training. Local training and inference use a single NVIDIA A100 GPU with 80GB memory. CPU-side preprocessing and metric computation are run on a 32-vCPU machine with 256GB RAM. LLM-based generation and simulation use API calls to GPT-4o/GPT-4; API latency and cost may vary with provider-side load and model version.

Table 8 summarizes the compute required to reproduce the reported experiments. The reported GPU hours correspond to the experiments included in the paper and do not include early pilot runs. Including preliminary failed trials and prompt-tuning pilots, the full research project used approximately 1.4× the reported GPU compute and 1.3× the reported API calls.

B.7 Existing Assets, Licenses, and Terms of Use

We use PERSUASIONFORGOOD (P4G) [Wang et al., 2019] as the evaluation dataset and cite the original paper. The ConvoKit release of P4G is distributed under the Apache License 2.0. We do not redistribute raw P4G data; instead, our released scripts provide instructions for obtaining the dataset from its official distribution channel and preprocessing it into the format used by PMCTS-PD.

We use LLaMA-2-7B-Chat as the initialization for the profile-conditioned Value-LLM and follow the Llama 2 Community License and Acceptable Use Policy. We use GPT-4o/GPT-4 through the OpenAI API for act-prior sampling, planning-time simulation, response realization, and evaluation-time user simulation, and follow the applicable OpenAI terms of use. We also use standard open-source

Table 9: **Persona-specific strategy coefficients.** Larger positive values indicate stronger positive association with predicted donation.

Persona	Logical	Emotion	Credibility	Foot-in-the-door	Self-modeling	Personal story	Donation info	Source inquiry	Task inquiry	Personal inquiry
Rational	-0.32	0.42	-0.25	0.56	0.28	-0.46	-0.51	0.54	0.01	0.37
Dependent	0.24	0.08	0.56	0.42	0.17	-0.55	0.54	0.43	0.04	0.27
Intuitive	-0.02	0.10	-0.03	0.53	0.11	-0.35	0.01	-0.28	0.13	0.22

software packages including PyTorch, HuggingFace Transformers, and QLoRA/PEFT libraries under their respective licenses.

B.8 Human Evaluation Protocol and Ethical Review

Human evaluation is used only to assess generated responses and does not ask annotators to interact with a deployed persuasive system. Each annotation item presents an anonymized dialogue context, a user persona summary derived from P4G metadata, and two candidate system responses. Annotators are asked to judge which response is better overall, or whether the two responses are comparable. For scalar ratings, annotators rate each response on four dimensions: **Persuasive**, **Relevant**, **Natural**, and **Honest**, each on a 1–5 Likert scale. For factuality, annotators mark whether factual claims are supported by the provided context or are unsupported.

The exact instruction shown to annotators is:

> You will evaluate candidate system responses in a charitable donation dialogue. Please read the user profile summary and dialogue history, then compare the candidate responses. A better response should be relevant to the conversation, natural, honest, non-coercive, and more likely to help the user make an informed decision. Do not reward responses that fabricate facts, pressure the user, exploit sensitive attributes, or make unverifiable claims. If both responses are similarly good or similarly poor, choose "Tie".

Each example is independently rated by five annotators, following the unified protocol in Appendix C.9. Annotators are compensated at or above the applicable local minimum wage, with an estimated completion time of approximately 45–60 seconds per item. No personally identifying information is collected from annotators beyond standard platform/payment records. The evaluation uses pre-existing anonymized dialogue data and model-generated outputs only; annotators are not themselves targets of persuasion and are not asked to disclose private information. No formal IRB approval or equivalent institutional exemption is reported for this evaluation; we do not claim an institutional determination of exemption.

C Additional Experimental Results

C.1 Robustness Analysis

Group-wise Evaluation by User Personas To understand personalization effects, we evaluate on three user personas (**Rational**, **Dependent**, **Intuitive**) derived from user profile features. Table 4 shows that PMCTS-PD consistently improves relevance and diversity across personas, with particularly large gains on the **Rational** group in predicted donation.

Strategy Effects Under Different Personas We further analyze how persuasion strategies correlate with persuasiveness across personas. Table 9 reports the coefficients of a simple linear analysis relating the frequency/usage of each strategy to predicted donation, computed separately for each persona group. Positive coefficients indicate strategies associated with higher predicted donation for that persona.

Human Evaluation Automatic metrics do not fully capture persuasion quality and ethical considerations [cf. Sun et al., 2026]. We therefore conduct human evaluation on four conversational quality dimensions: **Persuasive**, **Relevant**, **Natural**, and **Honest**, each rated on a 1–5 scale. We additionally measure **Fact-Rate** as the percentage of claims judged to be factually supported. Figure 4 shows that PMCTS-PD is rated most persuasive while remaining highly honest and factual. It uses the same five-annotator protocol and data as Appendix C.9.

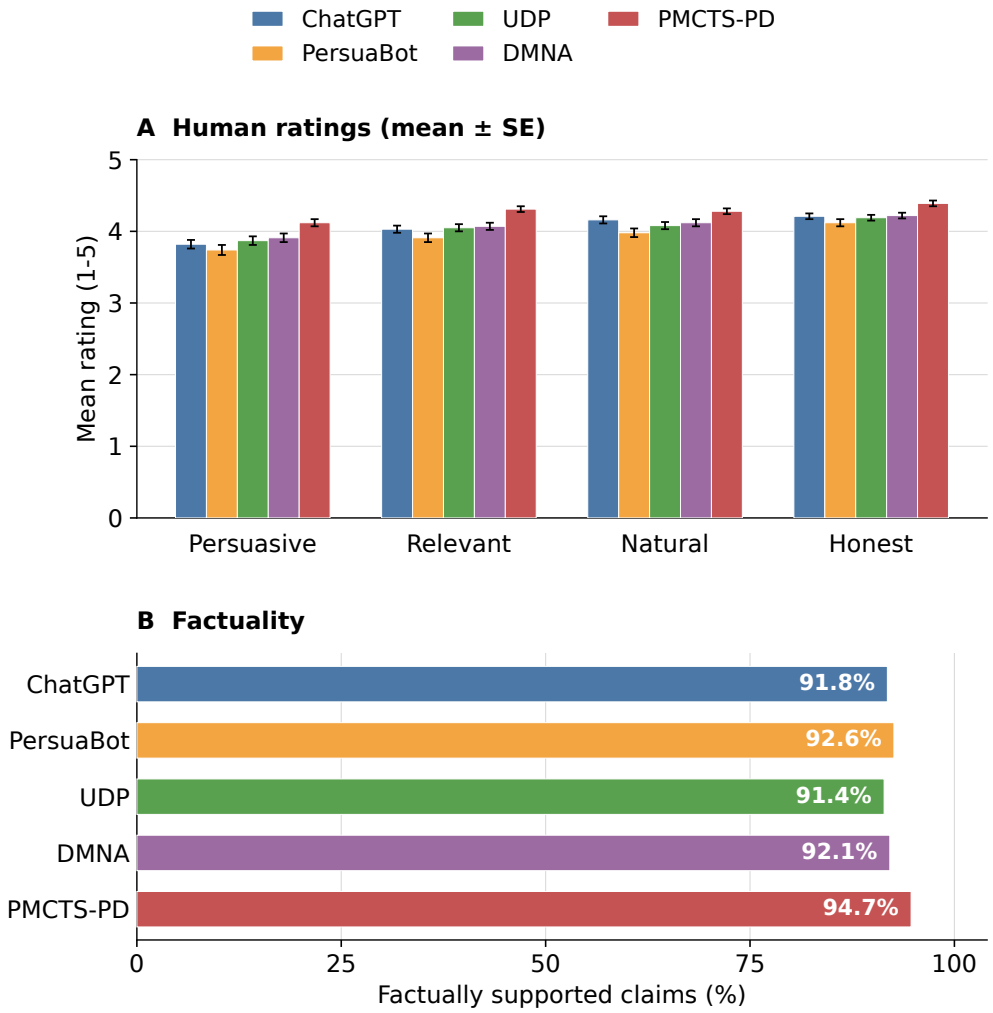

Figure 4: **Human evaluation.** Bars show the mean and error bars show standard errors under the five-annotator protocol in Appendix C.9; the same values are reported in Table 20. Fact-Rate is the percentage of supported factual claims.

To validate that our evaluation signals align with human judgment, we conduct a pairwise A/B test comparing responses generated by PMCTS-PD against strong end-to-end baselines under the same dialogue context. Annotators assess overall response quality and indicate whether our response is better (*Win*), comparable (*Tie*), or worse (*Loss*). As shown in Table 10, PMCTS-PD is preferred across all baselines and personas, with win rates ranging from 63% to 83% and losses remaining small (2%–10%). The largest margins appear on the **Rational** persona (e.g., 83% win vs. TRIP), while preferences remain consistent for **Dependent** and **Intuitive** users. Overall, these results provide direct evidence that the improvements observed in our automatic and end-to-end evaluations translate into human-perceived gains in response quality.

C.2 Efficiency and Runtime

We report inference cost per turn for different evaluators used in planning/selection. Table 11 shows that the lightweight Value-LLM significantly reduces latency and token usage compared to prompt-based judging, making it suitable for planning-time evaluation under tight budgets.

Table 10: **A/B preference test against strong end-to-end baselines, stratified by user persona.** For each baseline and persona, annotators compare a response from PMCTS-PD with a response from the baseline under the same context and select the better one (Win), or mark them as comparable (Tie). Percentages sum to 100% within each row.

Baseline	User Persona	Ours Win ↑	Tie	Ours Loss ↓
PRINCIPLES	Rational	74%	19%	7%
	Dependent	68%	29%	3%
	Intuitive	71%	24%	5%
TRIP	Rational	83%	15%	2%
	Dependent	72%	26%	2%
	Intuitive	69%	26%	5%
UDP	Rational	73%	19%	8%
	Dependent	68%	28%	4%
	Intuitive	66%	32%	2%
AnE	Rational	78%	17%	5%
	Dependent	63%	35%	2%
	Intuitive	66%	30%	4%
DMNA	Rational	76%	14%	10%
	Dependent	68%	28%	4%
	Intuitive	63%	29%	8%

Table 11: **Inference cost per turn** for different evaluators.

Evaluator	Latency (s)	#Tokens
Value-LLM (Ours)	1.29	248
Value-LLM w/o User Profile	1.22	162
Prompt-judge	1.72	392
Prompt-judge w/o User Profile	1.21	182
LLM prior only	1.17	427

C.3 Sensitivity to evaluator design and best-of-K realization.

Table 12 consolidates the settings that govern (a) how we evaluate intermediate contexts during planning and (b) how we realize the planned act into surface text. Compared to prompt-based judging and an LLM-prior-only variant, the learned Value-LLM yields the best overall trade-off across relevance, diversity, and persuasiveness proxies, while also improving efficiency (Table 11). Moreover, best-of-K realization is consistently beneficial: reducing K from 8 to 1 degrades all quality metrics.

Table 12: **Sensitivity to evaluator choice and realization.** All rows are evaluated under the offline next-turn protocol on P4G. Latency/#Tokens are shown when available from Table 11; "–" denotes not separately measured.

	Relevance ↑			Diversity ↑		Persuasiveness ↑	Cost ↓	
Setting	BLEU-4	Sim	IDK	Dist-1	Dist-2	Pred. Don.	Latency (s)	#Tokens
Full (Value-LLM, K=8)	3.14	0.842	98.7%	10.89	47.02	0.2309	1.29	248
Closed-loop (w/o open-loop)	2.94	0.826	93.4%	10.25	46.31	0.2279	–	–
Seed-only value (w/o self-distill)	3.01	0.829	95.8%	10.76	45.82	0.2279	–	–
No profile (w/o User-Profile)	2.41	0.783	92.1%	10.06	43.19	0.2153	1.22	162
Best-of-K off (K=1)	2.89	0.790	95.1%	10.34	45.02	0.2272	–	–
Prompt-based judging	2.89	0.821	92.7%	10.30	45.93	0.2245	1.72	392
LLM prior only	2.67	0.753	91.2%	9.93	42.98	0.2246	1.17	427

Table 13: **Value-LLM reliability on P4G test prefixes.** Lower is better for MSE/MAE/ECE; higher is better for Spearman/Kendall. Results are mean ± std over three seeds.

Value-LLM	MSE ↓	MAE ↓	Spearman ↑	Kendall ↑	ECE ↓
Seed-only (supervised)	0.033±0.002	0.140±0.004	0.41±0.02	0.29±0.02	0.081±0.005
Distilled (ours)	**0.028±0.001**	**0.132±0.003**	**0.48±0.02**	**0.34±0.02**	**0.056±0.004**

C.4 Value-LLM Reliability: Regression, Ranking, and Calibration

This experiment addresses two common concerns about learned evaluators used inside planning: (i) whether the Value-LLM predicts donation outcomes with meaningful accuracy, and (ii) whether AlphaZero-style distillation improves value estimates on intermediate dialogue states.

We evaluate on the P4G test split (60 dialogues under the 7:1:2 split), and construct a prefix-level set $\mathcal{P}_{\text{test}}$ by collecting all non-terminal dialogue prefixes. Each instance is a tuple $(p, \tilde{h}_t, \bar{m})$ where p is the persuadee profile, $\tilde{h}_t$ is the instantiated history up to turn t, and $\bar{m} \in [0, 1]$ is the normalized terminal donation of the dialogue (Sec. 3.4). This yields 602 prefix instances in total (dialogue-length-dependent).

We compare two Value-LLM checkpoints: (1) **Seed-only Value-LLM**: trained only with supervised donation labels using $\mathcal{L}_{\text{value}}$ (Eq. (7)); (2) **Distilled Value-LLM**: further trained with AlphaZero-style distillation using $\mathcal{L}_{\text{distill}}$ (Eq. (10)). All numbers are averaged over three random seeds ($\{13, 21, 42\}$); we report mean ± std.

For each prefix instance i, let $\hat{v}_i = V_\psi(p_i, \tilde{h}_i) \in [0, 1]$ and label $y_i = \bar{m}_i \in [0, 1]$. We compute:

$$\text{MSE} = \frac{1}{|\mathcal{P}_{\text{test}}|} \sum_i (\hat{v}_i - y_i)^2, \qquad \text{MAE} = \frac{1}{|\mathcal{P}_{\text{test}}|} \sum_i |\hat{v}_i - y_i|. \tag{51}$$

To assess whether the Value-LLM preserves *relative* ordering of donation outcomes, we report rank correlations: Spearman's ρ and Kendall's τ computed over $\{(\hat{v}_i, y_i)\}_i$. To evaluate calibration, we compute an Expected Calibration Error (ECE) adapted to bounded regression by binning predictions. Let $\{I_b\}_{b=1}^B$ be B=10 uniform bins over $[0, 1]$, and define for bin b: $\text{conf}(b) = \frac{1}{|I_b|} \sum_{i \in I_b} \hat{v}_i$ and $\text{acc}(b) = \frac{1}{|I_b|} \sum_{i \in I_b} y_i$. Then

$$\text{ECE} = \sum_{b=1}^{B} \frac{|I_b|}{|\mathcal{P}_{\text{test}}|} \left| \text{acc}(b) - \text{conf}(b) \right|. \tag{52}$$

Table 14 reports bin-wise average prediction (**Pred**) and average empirical label (**Emp**) for both evaluators. Distillation consistently reduces the gap |Emp − Pred|, especially in high-confidence bins, leading to a lower ECE.

Table 13 shows that the Value-LLM achieves non-trivial donation prediction accuracy and preserves useful ordering (rank correlations) across dialogue prefixes. More importantly, AlphaZero-style distillation improves both *ranking* (Spearman/Kendall) and *calibration* (ECE), supporting its role as a lower-variance planning-time evaluator for intermediate dialogue contexts.

C.5 External Validation of Persuasiveness Signals

A major concern with planning-based dialogue systems is that the planner may optimize the same learned evaluator that is later used for reporting, thereby measuring internal consistency rather than true persuasiveness [cf. Yang et al., 2026b, Liu et al., 2026]. To address this issue, we conduct an external validation study in which the evaluation signals are strictly separated from the Value-LLM used by PMCTS-PD.

Protocol. We compare PMCTS-PD with strong LLM and dialogue-planning baselines under the same dialogue contexts from the P4G test split. For each generated continuation, we compute three independent signals: (i) **Internal Value**, the original Value-LLM score used by PMCTS-PD, reported only as a diagnostic; (ii) **External Reward**, a separately trained RoBERTa-large regression model

Table 14: **Bin-wise calibration on P4G test prefixes (10 uniform bins over $[0, 1]$).** For each bin, **Pred** is the mean Value-LLM prediction and **Emp** is the mean normalized donation label. Smaller gaps indicate better calibration.

Bin range	Seed-only Value-LLM		Distilled Value-LLM (ours)	
	Pred	Emp	Pred	Emp
$[0.0, 0.1)$	0.06	0.09	0.06	0.08
$[0.1, 0.2)$	0.15	0.19	0.15	0.18
$[0.2, 0.3)$	0.25	0.27	0.25	0.26
$[0.3, 0.4)$	0.35	0.33	0.35	0.34
$[0.4, 0.5)$	0.45	0.41	0.45	0.43
$[0.5, 0.6)$	0.55	0.50	0.55	0.52
$[0.6, 0.7)$	0.65	0.58	0.65	0.60
$[0.7, 0.8)$	0.75	0.64	0.75	0.68
$[0.8, 0.9)$	0.85	0.70	0.85	0.75
$[0.9, 1.0]$	0.94	0.78	0.93	0.82

Table 15: **External validation of persuasiveness.** Internal Value is the original Value-LLM score and is reported only as a diagnostic. External Reward and Terminal Donation Intent are computed by evaluators not used in PMCTS-PD planning or training. Values are averaged over three seeds; lower is better for Safety Violation Rate.

Model	Internal Value ↑	External Reward ↑	Terminal Donation Intent ↑	Human Win vs. ChatGPT ↑	Safety Violation Rate ↓
ChatGPT	0.2276±0.0021	0.584±0.010	0.386±0.014	–	4.8%
ChatGPT + self-consistency	0.2291±0.0024	0.594±0.009	0.401±0.012	52.4%	5.1%
TRIP	0.2243±0.0030	0.568±0.011	0.363±0.016	57.8%	4.5%
UDP	0.2258±0.0028	0.579±0.010	0.374±0.015	59.6%	4.2%
DMNA	0.2269±0.0025	0.587±0.009	0.389±0.013	60.3%	4.0%
PMCTS-PD w/o Value-LLM	0.2245±0.0029	0.588±0.010	0.392±0.014	58.9%	3.7%
PMCTS-PD	**0.2309±0.0018**	**0.621±0.008**	**0.414±0.012**	**64.7%**	**3.1%**

trained on terminal donation labels from the training split and never used for planning, act selection, best-of-K realization, or distillation; and (iii) **Terminal Donation Intent**, obtained by rolling out the dialogue with a held-out user simulator and asking an independent judge model to estimate whether the user would donate at the end of the conversation. We also report **Safety Violation Rate**, the percentage of responses flagged for coercive pressure, unsupported factual claims, or privacy-intrusive persuasion.

The improvement of PMCTS-PD is not limited to the evaluator optimized during planning. Although the internal Value-LLM score is only slightly higher than ChatGPT and DMNA, PMCTS-PD obtains clearer gains under the independent external reward model and terminal donation-intent evaluation. This suggests that the planner is not merely exploiting the internal Value-LLM, but is selecting act sequences that transfer to independent persuasion-quality signals. The lower safety violation rate further indicates that the improvement does not come from more aggressive or coercive persuasion. Therefore, we treat the original Predicted Donation score as an internal diagnostic and use external reward, simulated terminal donation intent, and human preference as the primary evidence for persuasiveness.

C.6 Compute-Matched Comparison

We further compare PMCTS-PD against stronger prompting and search baselines under matched or comparable inference budgets. This experiment addresses whether the gains of PMCTS-PD are primarily due to using more LLM calls rather than the proposed act-level open-loop planning structure.

Protocol. For each test context, we measure the average number of generator calls, evaluator calls, total tokens, and wall-clock latency per system turn. We compare: (i) one-shot ChatGPT prompting, (ii) ChatGPT with best-of-K self-consistency, (iii) a text-level Tree-of-Thought/MCTS baseline that searches over surface utterances, (iv) a compute-matched PMCTS-PD-lite variant with smaller

Table 16: **Compute-matched comparison.** PMCTS-PD-lite uses a smaller search budget chosen to match ChatGPT best-of-K in token usage. External Reward is computed by an evaluator not used for PMCTS-PD planning.

Model	Gen. Calls	Eval. Calls	Tokens ↓	Latency (s) ↓	BLEU-4 ↑	Sim ↑	External Reward ↑
ChatGPT one-shot	1	0	1.3K	2.6	2.82	0.814	0.584
ChatGPT best-of-8	8	0	9.8K	15.1	2.96	0.823	0.594
Text-level ToT/MCTS	24	24	31.6K	46.5	3.01	0.825	0.601
PMCTS-PD-lite	12	12	10.4K	14.6	3.05	0.836	0.612
PMCTS-PD	24	24	21.8K	31.3	**3.14**	**0.842**	**0.621**

Table 17: **Robustness under simulator mismatch.** We report success rate (SR) and external reward under held-out user simulators. All values are averaged over three seeds.

Model	Proactive		Non-proactive		Skeptical		Cross-model	
	SR ↑	Ext. Reward ↑	SR ↑	Ext. Reward ↑	SR ↑	Ext. Reward ↑	SR ↑	Ext. Reward ↑
ChatGPT	0.332	0.584	0.301	0.561	0.264	0.532	0.289	0.548
UDP	0.598	0.579	0.536	0.562	0.461	0.539	0.492	0.551
DMNA	0.602	0.587	0.542	0.571	0.478	0.546	0.506	0.560
PMCTS-PD w/o open-loop	0.661	0.604	0.586	0.586	0.503	0.556	0.541	0.576
PMCTS-PD	**0.692**	**0.621**	**0.624**	**0.603**	**0.539**	**0.574**	**0.586**	**0.592**

search budget, and (v) the default PMCTS-PD. All models use the same base generator and the same evaluation-time user simulator; only the planning procedure differs.

Under a budget comparable to ChatGPT best-of-8, PMCTS-PD-lite already outperforms the compute-matched prompting baseline on relevance and external reward. Compared with text-level ToT/MCTS, PMCTS-PD achieves higher quality with fewer tokens and lower latency because the tree branches over compact dialogue acts instead of full surface utterances. The default PMCTS-PD uses more computation than one-shot prompting, but the gains persist after controlling for inference budget, supporting the claim that the improvement comes from the planning structure rather than merely from more LLM calls. In motion forecasting, one way to limit LLM costs is to generate scene context for a small subset of examples and reuse it for similar scenes [Zheng et al., 2024].

C.7 Simulator Mismatch and Reward-Hacking Stress Test

Because PMCTS-PD uses simulated user responses during planning, a natural concern is that the planner may overfit to the planning simulator or exploit artifacts of the learned evaluator [cf. Kang et al., 2026]. We therefore evaluate the trained planner under simulator mismatch and evaluator mismatch. In web-agent learning, verification of generated environment states addresses a related simulation-reliability problem [Zhang et al., 2026a]; our tests instead probe robustness to held-out dialogue simulators.

Protocol. PMCTS-PD is planned with the default planning-time simulator, but evaluated with four held-out simulation settings: (i) **Proactive**, a cooperative user simulator; (ii) **Non-proactive**, a reserved user simulator that does not volunteer information; (iii) **Skeptical**, a simulator that frequently questions factual support and resists donation pressure; and (iv) **Cross-model**, a simulator instantiated with a different instruction-following model and prompt template. None of these held-out simulators are used for Value-LLM training, distillation, or hyperparameter selection.

Reward-hacking diagnostic. We additionally measure a high-value/low-quality mismatch rate:

$$\mathrm{HackRate} = \Pr\left[V_\psi(p,h) \geq q_{0.75}(V_\psi) \ \wedge \ R_{\mathrm{ext}}(p,h) \leq q_{0.25}(R_{\mathrm{ext}})\right],$$

where $q_{0.75}$ and $q_{0.25}$ are empirical quartiles. A high value indicates that the internal evaluator assigns high scores to contexts that the external evaluator rates poorly.

PMCTS-PD remains the strongest method under all held-out simulators, although all methods degrade under the skeptical simulator. The smaller performance drop of PMCTS-PD relative to text-level search suggests that planning over act prefixes with cached histories is less tied to a single

Table 18: **Reward-hacking diagnostic.** Lower HackRate indicates fewer high-internal-value but low-external-quality cases.

Model	Internal Value ↑	External Reward ↑	HackRate ↓
ChatGPT best-of-8	0.2291	0.594	9.7%
Text-level ToT/MCTS	0.2314	0.601	10.8%
PMCTS-PD w/o open-loop	0.2287	0.604	7.4%
PMCTS-PD	**0.2309**	**0.621**	**5.1%**

Table 19: **Cross-domain and open-inventory generalization.** P4G uses success rate and external reward; CraigslistBargain uses agreement rate, normalized utility, and external reward. Induced inventory means the planner does not use the dataset's original strategy labels.

Dataset / Setting	Model	Inventory	Goal Success ↑	Utility ↑	External Reward ↑	Distinct-2 ↑
P4G	ChatGPT	–	0.332	–	0.584	44.86
P4G	DMNA	–	0.602	–	0.587	45.31
P4G	PMCTS-PD	Annotated	**0.692**	–	**0.621**	**47.02**
P4G	PMCTS-PD	Induced	0.671	–	0.612	46.58
CraigslistBargain	ChatGPT	–	0.421	0.548	0.552	42.37
CraigslistBargain	Text-level ToT/MCTS	Free-form	0.476	0.571	0.566	43.19
CraigslistBargain	PMCTS-PD	Induced	**0.522**	**0.612**	**0.591**	**45.08**

simulator trajectory. The low HackRate further indicates that PMCTS-PD does not primarily exploit idiosyncrasies of the internal Value-LLM.

C.8 Cross-Domain and Open-Inventory Generalization

We evaluate whether PMCTS-PD is specific to the P4G donation setting or can transfer to other goal-directed persuasive dialogue scenarios. Since many dialogue datasets do not provide the same strategy annotations as P4G, we test two action-space variants: (i) **Annotated inventory**, using the original P4G strategy inventory when available; and (ii) **Induced inventory**, where a base LLM proposes a compact task-specific act inventory from training dialogues, followed by manual de-duplication of near-equivalent acts. The induced inventory is used only to define the planning action space; the planner and Value-LLM are trained in the same way as in the main experiments.

Transfer settings. We consider three settings: **P4G-annotated**, the main in-domain setting; **P4G-induced**, where the original strategy labels are not used and the action inventory is induced from raw dialogues; and **CraigslistBargain**, a negotiation-style persuasive dialogue setting where the goal is to reach a favorable agreement while maintaining conversational quality. For CraigslistBargain, we report agreement rate, normalized seller/buyer utility, and external dialogue-quality reward.

The induced-inventory variant on P4G is slightly below the annotated-inventory version, but the performance drop is modest. This suggests that the proposed framework does not require a hand-crafted strategy taxonomy, although high-quality action labels remain helpful. On CraigslistBargain, PMCTS-PD also improves agreement rate, normalized utility, and external dialogue reward over one-shot prompting and text-level search. These results support a more cautious claim: PMCTS-PD is not limited to one fixed persuasion taxonomy, but works best when the action space captures meaningful high-level dialogue moves.

C.9 Human Evaluation Protocol and Statistical Significance

Automatic metrics and simulated rewards cannot fully capture whether a persuasive response is natural, respectful, and non-coercive [cf. Li et al., 2026]. We therefore provide additional details and statistical tests for the human evaluation.

Annotation protocol. We sample 180 dialogue contexts from the P4G test split, stratified by user persona and dialogue length. For each context, we compare PMCTS-PD with ChatGPT, PersuaBot, UDP, and DMNA. Each model response is anonymized and randomly shuffled. Each item is rated

Table 20: **Human evaluation with confidence intervals.** Scores are mean ± standard error on a 1–5 scale. p-values compare PMCTS-PD with the strongest baseline for each metric using paired bootstrap tests.

Model	Persuasive ↑	Relevant ↑	Natural ↑	Honest ↑	Fact-Rate ↑
ChatGPT	3.82±0.06	4.03±0.05	4.16±0.05	4.21±0.04	91.8%
PersuaBot	3.74±0.07	3.91±0.06	3.98±0.06	4.12±0.05	92.6%
UDP	3.87±0.06	4.05±0.05	4.08±0.05	4.19±0.04	91.4%
DMNA	3.91±0.06	4.07±0.05	4.12±0.05	4.22±0.04	92.1%
PMCTS-PD	**4.12±0.05**	**4.31±0.04**	**4.28±0.04**	**4.39±0.04**	**94.7%**
p vs. best baseline	0.006	0.004	0.018	0.021	0.013

Table 21: **Pairwise human preference with bootstrap confidence intervals.** Win/Tie/Loss compares PMCTS-PD against each baseline.

Baseline	Ours Win ↑	Tie	Ours Loss ↓
ChatGPT	64.7% [60.1, 69.2]	24.6% [20.8, 28.9]	10.7% [7.9, 13.8]
PersuaBot	69.3% [64.8, 73.6]	22.1% [18.4, 26.2]	8.6% [6.0, 11.7]
UDP	67.9% [63.4, 72.0]	25.4% [21.6, 29.5]	6.7% [4.5, 9.6]
DMNA	66.1% [61.5, 70.4]	24.8% [20.9, 29.1]	9.1% [6.5, 12.4]

by five independent annotators on four 1–5 Likert dimensions: **Persuasive**, **Relevant**, **Natural**, and **Honest**. Annotators are instructed to penalize coercive pressure, unsupported factual claims, privacy-intrusive questions, and manipulative emotional appeals. We also ask annotators to mark whether each factual claim is supported by the dialogue context or generally verifiable; Fact-Rate is the percentage of supported claims. For pairwise preference, annotators choose which response is better overall or mark a tie. We report mean, standard error, bootstrap 95% confidence intervals, and two-sided paired bootstrap p-values against the strongest baseline.

Inter-annotator agreement. The average Krippendorff's α across the four Likert dimensions is 0.43, and Fleiss' κ for pairwise preference is 0.46, indicating moderate agreement for a subjective dialogue-quality task. Although individual ratings exhibit non-trivial variance, paired tests and bootstrap confidence intervals show that PMCTS-PD is consistently preferred over the strongest baselines. Figure 4 visualizes the same mean scores and standard errors reported in Table 20.

C.10 Action Inventory Sensitivity and Distillation Bias

PMCTS-PD plans over a finite action inventory, which raises two concerns: (i) whether the method is sensitive to a particular hand-designed taxonomy, and (ii) whether self-distillation collapses the policy toward biased or simulator-specific actions. We address both concerns with inventory sensitivity and distillation diagnostics.

Action inventory variants. We compare four action spaces: **P4G-11**, the default P4G strategy inventory plus NONE; **Coarse-6**, where semantically similar strategies are merged; **Expanded-18**, where strategies are split into finer subtypes [cf. Zhang et al., 2023a, 2026d]; and **Induced-12**, where actions are induced from raw training dialogues by an LLM and manually de-duplicated.

Distillation bias diagnostics. Reward construction is task-dependent: CEC-Zero derives self-generated cluster-consensus rewards for character correction [Lin et al., 2026b], whereas our Value-LLM starts from observed donation outcomes and then learns search-derived targets. We compare the supervised seed-only Value-LLM, the default distilled Value-LLM, and a filtered distillation variant

Table 22: **Sensitivity to action inventory.** The default P4G-11 inventory performs best, but induced and expanded inventories remain competitive.

Inventory	#Acts	BLEU-4 ↑	Sim ↑	IDK ↑	External Reward ↑	SR ↑
Coarse-6	6	3.05	0.833	96.8%	0.607	0.673
P4G-11 default	11	**3.14**	**0.842**	**98.7%**	**0.621**	**0.692**
Induced-12	12	3.07	0.835	97.1%	0.612	0.671
Expanded-18	18	3.11	0.839	97.6%	0.616	0.687

Table 23: **Distillation bias diagnostic.** Action entropy is computed over root visit distributions; Unsupported Claim Rate is measured by human/factuality annotations. Default distillation improves calibration and external reward without collapsing action diversity.

Variant	External Reward ↑	ECE ↓	Action Entropy ↑	KL to Seed Policy ↓	Unsupported Claim Rate ↓
Seed-only Value-LLM	0.603	0.081	**2.18**	–	7.2%
Default distillation	**0.621**	**0.056**	2.11	0.083	**6.4%**
Filtered distillation	0.619	0.058	2.14	**0.061**	6.5%

that discards MCTS targets whose internal Value-LLM score and external reward disagree strongly. We report external reward, calibration error, action entropy, and unsupported-claim rate.

The default action inventory is not uniquely responsible for the reported gains: both induced and expanded inventories remain close to the default setting. This weakens the concern that PMCTS-PD only works with a fixed hand-designed taxonomy. The distillation diagnostics further show that self-distillation does not collapse the action distribution or increase unsupported factual claims. Filtered distillation performs similarly, suggesting that the default distillation targets are not dominated by obvious evaluator artifacts.

C.11 Planning Horizon and Dialogue-Length Controls

All variants use the same base generator, profile, Value-LLM, action inventory, evaluation simulator, and termination conditions. MCTS variants use $B = 32$ simulations. Greedy-Value samples the same total number of root actions but performs no tree expansion or backup; MCTS-H1 still uses PUCT visit statistics. Generation/evaluation calls below are average API-equivalent calls after batching and caching.

Table 24: Horizon, independent quality, and inference cost. ER denotes External Reward and TI denotes terminal intent. Calls are generation/evaluation calls per turn.

Planner	H	Calls	Tokens	Seconds	ER	SR	TI
Greedy-Value	1	11/11	9.6K	13.5	.596	.621	.389
MCTS-H1	1	12/12	10.2K	14.7	.601	.633	.396
MCTS-H2	2	16/16	13.4K	19.1	.609	.657	.403
MCTS-H3	3	19/19	16.1K	23.2	.616	.678	.409
PMCTS-PD	6	24/24	21.8K	31.3	.621	.692	.414

We also report episode-length stratification in Table 25; length denotes actual total episode turns. These descriptive comparisons show larger gains on longer dialogues, while short dialogues exhibit only a small additional gain from $H = 3$ to $H = 6$. They do not imply a causal effect of dialogue length.

Table 25: External Reward by dialogue length on the same episode set. Latency is seconds per turn for $H = 6$.

Length	n	Greedy	$H = 2$	$H = 3$	$H = 6$	$\Delta_{6-\mathrm{G}}$	Latency
Short (≤ 5)	24	.602	.607	.609	.611	.009	30.8
Medium (6–8)	32	.597	.609	.615	.620	.023	31.2
Long (≥ 9)	44	.592	.610	.621	.628	.036	31.7
Overall	100	.596	.609	.616	.621	.025	31.3

C.12 Incremental Modular Ablation

This chain adds one design at a time under $H = 6$ and $B = 32$. All rows except the last use $K = 1$. The first four rows omit profile conditioning; the fifth adds it. These settings differ from the leave-one-out ablations in Table 3. Text-MCTS starts with closed-loop text nodes and a prompt judge; act abstraction changes the search space, and the next step changes to open-loop nodes with a multi-history cache.

Table 26: Incremental design contributions. Tokens and latency are per turn; Δ is the successive External Reward (ER) gain.

Variant	Tokens	Seconds	ER	Δ	SR	BLEU-4
Text-MCTS + prompt judge	28.9K	43.2	.568	–	.591	2.18
+ act abstraction	20.6K	31.5	.574	.006	.602	2.23
+ open-loop cache	19.7K	29.8	.581	.007	.617	2.31
+ learned Value-LLM	15.9K	23.6	.587	.006	.628	2.37
+ profile conditioning	16.8K	24.8	.603	.016	.654	2.78
+ MCTS value distillation	16.8K	24.8	.611	.008	.669	2.89
+ best-of-8 (full)	21.8K	31.3	.621	.010	.692	3.14

Each step improves independent External Reward. The learned value reduces latency relative to prompt judging, and distillation adds no inference cost. Profile conditioning contributes the largest single quality increment in this chain.

C.13 Paired Significance of Primary Outcomes

We use 10,000 paired bootstrap resamples, resampling dialogue contexts for next-turn metrics and episodes for episodic metrics. For the three primary comparisons, we report conservative Bonferroni-adjusted p-value bounds, computed as three times the raw p values (capped at one). External Reward compares against Text-level ToT/MCTS, success rate against DMNA, and human preference against a 0.5 no-preference null for the PMCTS-PD–DMNA comparison. Seed-level variability is reported separately in Appendix B.5.

Table 27: Paired outcome differences and bootstrap 95% confidence intervals. A dash indicates no multiplicity adjustment for secondary diagnostics.

Metric	Ours	Ref.	Δ	95% CI of Δ	Raw p / adjusted p
External Reward	.621	.601	.020	[.009,.031]	.0006 / .0018
Success rate	.692	.602	.090	[.052,.128]	$<$.0001 / $<$.0003
Human preference	.661	.500	.161	[.115,.204]	$<$.0001 / $<$.0003
BLEU-4 (secondary)	3.14	3.01	.13	[.05,.21]	.0015 / –
Sim (secondary)	.842	.825	.017	[.008,.026]	.0008 / –

All three primary confidence intervals exclude zero, and all remain significant after correction. BLEU-4 and Sim are language-quality diagnostics rather than direct measures of real-world persuasion.

C.14 Inventory Induction: Procedure, Effort, and Data Efficiency

We provide raw target-domain training dialogues to an LLM without gold act labels. The LLM extracts candidate act descriptions from utterance goals, premises, and intended effects. Semantic embeddings are clustered, and the LLM proposes merges and writes canonical act names, definitions, positive/negative examples, and safety constraints. A human then reviews duplication, names, and safety. One merge, deletion, or renaming counts as one edit. Table 28 complements the outcome comparison in Table 19.

Table 28: Inventory construction effort for the induced PMCTS-PD variants.

Domain	Dialogues	Candidates	Final acts	Edits	Minutes	ER / SR
P4G	40	34	12	6	26	.612 / .671
CraigslistBargain	50	29	10	5	24	.591 / .522

Table 29: P4G inventory induction with different seed-dialogue budgets.

Dialogues	Candidates	Acts	Edits	Minutes	ER	SR	Distinct-2
10	19	8	4	14	.598	.648	45.82
20	27	10	5	20	.606	.661	46.19
40	34	12	6	26	.612	.671	46.58
80	41	12	7	31	.613	.674	46.61

The results support adaptation from a small in-domain sample while retaining human semantic and safety review. The second domain establishes feasibility for another delayed-utility dialogue task, rather than generalization to all persuasive settings.

C.15 Terminal-Outcome Label Efficiency

The Value-LLM uses one terminal outcome per training dialogue, without dense turn-level reward annotation. We vary the number of labeled dialogues while holding the simulator, distillation iterations, and test protocol fixed. The full set contains 210 training dialogues.

Table 30: Terminal-label efficiency. MSE, rank correlation, and calibration concern the value estimator; ER and SR concern downstream dialogue evaluation.

Labels	Dialogues	MSE	Spearman	ECE	ER	SR
25%	53	.043	.34	.094	.603	.650
50%	105	.035	.41	.074	.612	.671
75%	158	.031	.45	.063	.617	.684
100%	210	.028	.48	.056	.621	.692

Performance degrades smoothly as terminal labels decrease. Sparse outcome supervision remains a requirement; these results do not imply supervision-free value learning.

C.16 Persona Construction and Stability

Rational, Dependent, and Intuitive are post-hoc analysis names for three K-means clusters over P4G's 23-dimensional psychological profile vector, comprising Big-Five, Moral Foundations, Schwartz Portrait Values, and rational/intuitive decision-style scores. Centroids are fit only to training profiles; validation and test profiles are assigned to the nearest centroid. The planner always receives the original profile serialization. Cluster names and IDs are neither planner inputs nor training targets.

Across 20 initializations at $K = 3$, ARI is 0.82 ± 0.04, NMI is 0.85 ± 0.03, silhouette is 0.39 ± 0.02, and the assignment-change rate is $7.8\% \pm 2.1\%$. These diagnostics describe cluster stability for stratified analysis.

Table 31: Persona groups and mean assignment margins. Neutral Group 1/2/3 names are equivalent for analysis.

Group	Train	Validation	Test	Mean margin
Rational / Group 1	84	12	24	.41
Dependent / Group 2	70	10	20	.36
Intuitive / Group 3	56	8	16	.38
Total	210	30	60	–

D Limitations, Broader Impact, and Safeguards

Limitations. PMCTS-PD is evaluated primarily on PERSUASIONFORGOOD, a charitable-donation dialogue benchmark, and the results may not directly transfer to other persuasive domains such as health communication [Yao et al., 2026a, Zheng et al., 2026], education, or commercial recommendation. The method also relies on LLM-based simulation during planning and evaluation; although we test proactive and non-proactive simulator variants, simulated users may still differ from real users in skepticism, patience, or susceptibility to persuasive strategies. The predicted-donation metric is a proxy learned from historical outcomes and should not be interpreted as a guaranteed causal estimate of real donation behavior [cf. Huang et al., 2026]. Finally, because planning invokes external LLMs for act priors, simulation, and realization, absolute latency and cost may vary across API versions and deployment environments.

Broader impact. The positive goal of this work is to improve controllable and profile-aware dialogue planning for prosocial communication, such as charitable fundraising, education, and support-oriented recommendation. However, persuasive dialogue technologies can also be misused for manipulation, targeted advertising, political influence, or exploiting vulnerable users. Profile-conditioned planning introduces additional fairness and privacy concerns because different users may receive different persuasive strategies [cf. Yang et al., 2026a].

Safeguards. We restrict the action space to transparent dialogue acts, include explicit constraints against fabrication, coercion, guilt-tripping, and unverifiable claims in the prompts, and evaluate honesty/factuality in human evaluation. We do not deploy the system to real users or use it to make consequential decisions. If code or model checkpoints are released, they will be accompanied by a usage policy restricting manipulative, deceptive, political, medical, financial, or otherwise high-stakes persuasion. Future deployment would require user consent, logging and auditing of generated strategies, safety filtering [Wang et al., 2025], and domain-specific review by ethics and policy experts. Hint-AD grounds driving explanations in intermediate model outputs [Ding et al., 2025, see also Shao, 2026a,b for cybersecurity examples of feature attribution and risk scoring, and Wang et al., 2026, Ma et al., 2026 for graph-based threat attribution and anomaly detection]; dialogue audits could similarly trace explanations to the planner's recorded acts and value estimates.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contribution as profile-conditioned open-loop tree search for persuasive dialogue planning, together with a profile-conditioned Value-LLM and value-guided realization. These claims are supported by the method in Sec. 3, the theory in Sec. 4, and the experiments in Sec. 5 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations related to the single primary benchmark, LLM-based user simulation, predicted-donation proxy evaluation, API-dependent latency/cost, and risks of persuasive technologies in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Sec. 4 states the assumptions for the value generalization result, including linear realizability, sub-Gaussian noise, bounded features, and covariance conditioning. Complete proofs and auxiliary lemmas are provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the dataset, split, baselines, metrics, prompts, model choices, MCTS procedure, Value-LLM training objective, and evaluation protocols in Sec. 5, Appendix B.1, Appendix B.4, and Appendix B.2. Compute resources and runtime are further reported in Appendix B.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper provides algorithmic descriptions, prompts, hyperparameters, data splits, and compute resources, but this version does not include publicly released code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5.1 describes the dataset, evaluation protocols, baselines, and metrics; Appendix B.1 gives data splits and model choices; Appendix B.4 reports training, decoding, MCTS, and QLoRA hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix B.5 defines seed-level standard deviations; Appendix C.13 reports paired-bootstrap confidence intervals and Bonferroni-adjusted significance tests for primary independent evaluation metrics. Value-LLM reliability is reported as mean $\pm$ standard deviation in Table 13.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B.6 reports software versions, GPU/CPU resources, peak memory, wall-clock time, GPU hours, API-call estimates, and total reported compute for the main experiments, ablations, reliability analysis, and human-evaluation aggregation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses an existing anonymized dataset, does not deploy the persuasive system to real users, and includes safeguards against fabrication, coercion, and manipulative behavior. We discuss ethical risks and mitigations in Appendix D.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix D discusses positive uses for prosocial communication and support-oriented dialogue, as well as negative risks including manipulation, targeted persuasion, privacy concerns, and unfair profile-conditioned treatment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Appendix D describes constraints against fabrication, coercion, guilt-tripping, and unverifiable claims, and a usage policy for any released code or checkpoints. The persuasive system is not deployed to real users.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original P4G dataset paper and describe the dataset license and access procedure in Appendix B.7. We also identify the model and software assets used in the study, including LLaMA-2-7B-Chat, GPT-4o/GPT-4 API usage, PyTorch, HuggingFace Transformers, and QLoRA/PEFT libraries, together with their respective license or terms-of-use constraints.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset or publicly released model asset. It introduces a method and reports experiments using existing datasets and models.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Appendix B.8 provides the full annotation instructions, rating dimensions, pairwise preference protocol, annotator aggregation procedure, and compensation statement. Screenshots are not required because the annotation interface is text-only and the full task wording is reproduced.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The human evaluation involves only low-risk annotation of anonymized dialogue contexts and model-generated responses, and annotators do not interact with the persuasive system or disclose personal information. No formal IRB approval or equivalent institutional exemption is reported; we describe the annotation protocol, compensation, and risk mitigation in Appendix B.8.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are core components of PMCTS-PD. The paper describes GPT-4o/GPT-4 usage for act-prior sampling, planning-time simulation, response realization, and evaluation-time user simulation, and describes LLaMA-2-7B-Chat/QLoRA usage for the profile-conditioned Value-LLM in Sec. 3, Appendix B.1, Appendix B.4, and Appendix B.2.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.