Activation-Steered Personas in Task-Oriented LLM Agent Simulations

Maryam Shoaeinaeini, Bernardo Rodrigues, Adib Mosharrof, A.B. Siddique,& Brent Harrison
Department of Computer Science
University of Kentucky
Lexington, KY 40506, USA
{maryam.shoaei, bernardorodrigues, adib.mosharrof, ab.siddique, brent.harrison}@uky.edu

Abstract

Large language models are increasingly used to simulate both user and system behavior in task-oriented dialogue (TOD). Yet personality control in these simulations remains largely prompt-based, making trait expression difficult to regulate and verify across dialogue turns. We investigate activation steering as an inference-time mechanism for inducing and adapting personality in TOD simulation. In our framework, user-side steering (US) applies fixed Big Five trait-pole activation vectors with scalar coefficients to generate users with different trait intensities. System-side adaptive steering (SA) dynamically adapts the system agent at each turn by predicting system-side steering coefficients from the dialogue context. We evaluate four steering conditions: an unsteered Baseline, SA, US, and US+SA, using the Schema-Guided Dialogue benchmark across the Restaurants and Movies domains. Qwen2.5-7B-Instruct is used as the primary model, and Llama-3.1-8B-Instruct is used as a robustness setting.

Our results show that activation-level personality control affects TOD behavior, but its benefits depend on domain, trait observability, and steering strength. Compared with the unsteered Baseline, SA improves standard task metrics slightly, with larger gains in pairwise-judged user satisfaction and truthfulness, mainly in Restaurants, where dialogues contain richer personality-relevant slots and preferences. User-side steering produces observable personality expression only for some traits: socially or affectively expressive traits such as careless, outgoing, compassionate, and nervous become more detectable at stronger steering, while consistent and solitary remain difficult to observe in short TOD interactions.Adding SA to steered users yields modest task-level changes but a statistically significant improvement in pairwise-judged truthfulness (+8.3 pp) and user satisfaction (+2.7 pp). System-side composition analysis further shows that SA does not directly mirror the user's trait; it usually defaults to Conscientiousness-centered system behavior and becomes more socially adaptive when user trait expression is observable especially at higher steering scalar. Overall, activation steering is a viable alternative to prompt-based personality conditioning for controlled TOD simulation, but its effects must be evaluated with respect to trait observability, steering strength, and the personality relevance of the task domain. [1]

1 Introduction

Simulating realistic user behaviour is a longstanding challenge in the development and evaluation of task-oriented dialogue (TOD) systems (Schatzmann et al., 2005; Wen et al., 2016). Traditional user simulators often rely on rule-based policies or fixed behavioural profiles,

[1]Code and generated dialogues are available at https://github.com/MultifacetedNLP/TODPersonaSteering

limiting their ability to capture the diversity of preferences, constraints, and communicative styles that real users exhibit (Kreyssig et al., 2018; Shi et al., 2019). Large language models (LLMs) offer a more flexible alternative for generating naturalistic user behaviour through prompting (Lin et al., 2022; Davidson et al., 2023). However, prompt-based personality conditioning remains brittle: trait expression is difficult to verify across turns, and the strength of the conditioning cannot be systematically controlled (Serapio-García et al., 2023; Jiang et al., 2024; Yeo et al., 2025).

Prior work suggests that many LLM behaviours are linearly encoded in activation space, enabling inference-time control through vector addition and scalar modulation (Chen et al., 2025). Activation steering builds on this observation by directly modifying internal model activations during generation rather than relying only on prompt-based conditioning (Turner et al., 2024; Marks & Tegmark, 2023). Related work on persona vectors further shows that the Big Five OCEAN personality traits (Goldberg, 2013) are represented in LLM activations and can be probed as trait-specific directions (Feng et al., 2026). However, most prior evaluations focus on open-ended persona conversations, where the primary goal is expressive trait adherence. It remains unclear how activation-level personality control behaves in realistic *task-oriented* dialogue, where agents must complete API calls, satisfy slot constraints, provide grounded information, and achieve user goals. This question is especially important in LLM-based simulations, where user-side personality induction and system-side adaptive steering can be applied independently or jointly. In this setting, steering effects may depend on which agent is steered, how strongly personality is induced, and which trait is expressed. Motivated by this gap, our experiments are organized around three research questions:

RQ1: Does system-side adaptive steering improve task-oriented dialogue when the user is neutral?

RQ2: Does user-side activation steering produce observable personality traits in task-oriented dialogue?

RQ3: When users are activation-steered, how does system-side adaptive steering perform across different traits and scalar strengths?

To answer these questions, we design a dual-agent activation-steering framework for task-oriented dialogue simulation. A single LLM is instantiated twice: once as a simulated user and once as a system agent. We study two complementary steering mechanisms. User-side activation steering (US) injects Big Five trait-pole vectors into the user model with scalar coefficients from $\{-1, 0, 1, 2\}$, producing users with different trait intensities. System-side adaptive activation steering (SA) applies dynamic, turn-level steering to the system model by predicting trait-specific coefficients from the recent dialogue context and composing a system-side steering vector at each system turn. We evaluate four conditions: an unsteered Baseline, SA, US, and their combination, US+SA. Experiments use the Schema-Guided Dialogue dataset (Rastogi et al., 2020) across Restaurants and Movies, with Qwen2.5-7B-Instruct as the primary model and Llama-3.1-8B-Instruct as a cross-model robustness setting. We evaluate generated dialogues using both standard task metrics and LLM-judged pairwise metrics, allowing us to measure task execution, factual grounding, constraint satisfaction, and perceived user satisfaction.

Our results show that activation-level personality control affects both task behaviour and perceived response quality, but its effects depend on domain, trait observability, and steering strength. For RQ1, SA improves neutral-user performance mainly in Restaurants, where dialogue decisions involve richer personality-relevant preference; in Movies, gains are weaker and appear primarily in judged user satisfaction. For RQ2, US produces observable personality expression only for some traits. Socially expressive traits such as careless, outgoing, compassionate, and nervous become detectable, especially at stronger steering, while internally oriented or interaction-suppressing traits such as consistent and solitary remain difficult to observe in task-oriented dialogues. For RQ3, adding SA to directionally steered users yields slight task-level gains and clearer pairwise-quality improvements, increasing truthfulness by $+8.3$ pp and user satisfaction by $+2.7$ pp over US, with a smaller, non-significant gain in constraint satisfaction ($+0.6$ pp). Trait-level analysis shows that SA benefits nervous, self-interested, compassionate, careless, and outgoing users more in

satisfaction, but reduces satisfaction for less observable traits such as solitary and consistent. Scalar-level analysis further shows that SA is most reliable when the user-personality signal is clearly expressed. System-side adaptation analysis shows that SA does not directly mirror the user's target trait; it usually defaults to a Conscientiousness-centered service strategy and becomes more socially adaptive when user-side steering produces stronger behavioural cues. The Llama robustness setting confirms the same pairwise trend at high statistical significance across all qualitative pairwise metrics, indicating that the response-quality benefit is not specific to a single model, though trait- and scalar-level patterns remain model-dependent. Together, these findings show that activation steering provides a controllable mechanism for simulating personality variation in both user and system behavior in task-oriented dialogue.

2 Related Work

User simulation for task-oriented dialogue. User simulation is central to training and evaluating task-oriented dialogue (TOD) systems without requiring costly human interaction (Schatzmann et al., 2006; Kreyssig et al., 2018). Early simulators commonly use agenda-based policies, hand-crafted rules, or statistical user models (Schatzmann et al., 2007). These methods provide controllable task behaviour, but they require domain-specific design and often encode limited assumptions about user preferences, communication styles, and behavioural diversity. Neural and data-driven simulators improve naturalness and reduce manual rule construction, but they can still struggle with controllability, goal consistency, and coverage of diverse user behaviours (Shi et al., 2019). More recently, LLM-based user simulators have improved linguistic flexibility and goal coverage (Lin et al., 2022; Davidson et al., 2023). However, most LLM-based simulators still rely on prompt-based conditioning to shape user behaviour, which makes personality expression difficult to control systematically across turns. This motivates the need for more controllable mechanisms for personality variation in simulated users.

Personality control in language models. Personality control in LLMs has been approached through prompting, training-based alignment, and activation-level intervention. Prompt-based methods condition models with personality descriptors or OCEAN trait labels but produce inconsistent trait expression across contexts and offer limited control over intensity (Jiang et al., 2024; Serapio-García et al., 2023; Yeo et al., 2025; La Cava & Tagarelli, 2025). Training-based approaches provide more stable conditioning through supervised fine-tuning or preference optimization (Li et al., 2025), but require additional training and cannot adapt personality dynamically during inference. Activation-level methods offer a training-free alternative by steering internal representations directly (Turner et al., 2024; Zou et al., 2022). Deng et al. (2025) manipulate personality-relevant neurons to induce traits; Feng et al. (2026) extract contrastive vectors for all ten Big Five poles and introduce a dynamic predict-then-steer mechanism for context-aware adaptation. We extend this line of work to TOD, where personality adaptation must coexist with slot constraints, API calls, and user-goal completion.

Activation steering in dialogue and social simulation. Activation steering has been proposed to control behaviors such as toxicity, sycophancy, truthfulness, and refusal in single-agent settings (Rimsky et al., 2024; Arditi et al., 2024), but its role in interactive multi-agent dialogue remains largely unexplored. LLM-based social simulation has grown rapidly (Mou et al., 2026; Park et al., 2023), yet most systems rely on prompt-based role assignment rather than controllable activation-level interventions. We study dual-agent activation steering in TOD by isolating user-side fixed steering, system-side dynamic steering, and their combination. We evaluate each condition using both structural task metrics and pairwise LLM-judged comparisons of system-response quality, reporting win rates to measure which steering condition produces more preferred responses.

3 Method and Experimental Setup

We formulate TOD simulation as an interaction between two LLM agents. The *user agent* receives a task goal from the dataset and generates user turns conditioned on that goal. The *system agent* observes the dialogue history and responds with the objective of satisfying the user request and completing the task. Both agents are instantiated from the same base LLM and assigned different roles through prompting; depending on the experimental condition, they also receive different activation-steering interventions at inference time. This dual-agent setup allows us to isolate whether personality steering affects user behaviour, system adaptation, or their interaction.

3.1 Persona Activation Steering

We build on activation steering, where a personality trait is represented as a direction in the model's residual stream and applied by addition during inference:

$$\mathbf{h}_l \leftarrow \mathbf{h}_l + \alpha\, \mathbf{v}_t^l \tag{1}$$

where $\mathbf{h}_l$ is the residual activation at layer l, $\mathbf{v}_t^l$ is the persona vector for trait pole t at that layer, and α is a scalar coefficient controlling steering strength and direction (Turner et al., 2024; Feng et al., 2026). We use the ten Big Five trait poles: inventive, consistent, dependable, careless, outgoing, solitary, compassionate, self-interested, nervous, and calm (Goldberg, 2013). Each vector is extracted via contrastive activation analysis: the model generates responses under trait-expressing and trait-suppressing system prompts, and the vector is computed as the difference between the mean residual activations of the two groups (Feng et al., 2026). We use the publicly released vectors for Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct from the `xiachongfeng/persona` repository.[2] Steering is applied at layer 20 in all experiments, the layer reported as most effective for personality control on Qwen2.5-7B-Instruct (Feng et al., 2026).

3.2 Persona Steering Conditions

To evaluate how steering affects each side of the dialogue interaction, we define four experimental conditions by varying whether persona steering is applied to the user agent, the system agent, both, or neither. These conditions are: an unsteered **Baseline**, system-side adaptive steering alone (**SA**), user-side steering alone (**US**), and simultaneous user-side and system-side steering (**US+SA**).

User Steering (US). User Steering applies a fixed personality vector to the simulated user agent. For each dialogue, the user is assigned one of the ten Big Five trait poles and a scalar coefficient $\alpha \in \{-1, 0, 1, 2\}$. The corresponding trait vector $\mathbf{v}_{t,l}$ is injected at layer 20 at every user turn throughout the dialogue. A negative α moves generation away from the target trait; α=0 activates the steering pipeline without applying a directional shift, serving as a diagnostic condition; α=2 applies the strongest trait intervention. Across ten trait poles and four scalar values, US produces forty distinct user configurations, each evaluated independently per domain. The user agent prompt is provided in Appendix C.2.

System-side adaptive steering (SA). SA applies dynamic steering to the system agent via a predict-then-steer mechanism (Feng et al., 2026). Before each system response, the dialogue context is passed to a state-of-the-art LLM, which predicts a coefficient $\hat{\alpha}_t \in [-2, 2]$ for each Big Five trait t. Coefficients whose magnitude falls below the sparsity threshold τ=0.5 are zeroed out. The remaining coefficients are composed into a single steering vector injected into the residual stream of the system agent at layer 20 during decoding:

$$\mathbf{v}_{\text{comp}} = \sum_{t \in \mathcal{T}} \mathbf{1}[|\hat{\alpha}_t| \geq \tau] \cdot |\hat{\alpha}_t| \cdot \tilde{\mathbf{v}}_t^{\,\text{sign}(\hat{\alpha}_t)} \tag{2}$$

[2] https://huggingface.co/datasets/xiachongfeng/persona

where $\mathcal{T}$ is the set of Big Five traits, and $\tilde{\mathbf{v}}_t^{(+)}$ and $\tilde{\mathbf{v}}_t^{(-)}$ are the unit-normalised steering vectors for the positive and negative poles of trait t, respectively. Because each Big Five dimension has two independently extracted pole vectors, the sign of $\hat{\alpha}_t$ selects which pole is activated — the positive pole if $\hat{\alpha}_t \geq 0$ and the negative pole otherwise — while $|\hat{\alpha}_t|$ controls its intensity. Unlike US, which holds a single trait fixed across the entire dialogue, SA recomputes and reinjects the composite vector at every system turn, allowing the system to adapt its personality configuration to the evolving dialogue context. The system agent prompt is provided in Appendix C.1.

3.3 Dataset

We evaluate our framework on the Schema-Guided Dialogue (SGD) benchmark (Rastogi et al., 2020), a multi-domain task-oriented dialogue dataset with schema-defined services, intents, slots, and API calls. Each dialogue is associated with a user goal, and the system must collect the required slot values, issue the appropriate API call, and use the returned search results to satisfy the user's request.

We focus on two domains: *Restaurants* and *Movies*. The Restaurants domain requires the system to handle preference-rich booking decisions, such as cuisine, price range, availability of vegetarian options, and availability of outdoor seating. This makes it a useful setting for testing whether personality steering affects how users express preferences and how systems adapt to them. The Movies domain is more lookup-oriented, involving movie search and ticket-related requests. It provides fewer personality-relevant decision points at the task level, but still allows variation in conversational style and perceived response quality.

3.4 Evaluation Metrics

We evaluate each experimental condition using two complementary layers of metrics. Standard task metrics measure whether each dialogue achieves its transactional objective, while LLM-judged pairwise metrics measure perceived response quality beyond task completion. Both layers are necessary because activation steering may improve conversational quality without improving task structure, or may improve task execution while introducing trade-offs in response style and user satisfaction.

Standard Task Metrics. We report six standard task-oriented dialogue metrics computed from the generated dialogues and reference annotations. **Method Accuracy** measures whether the predicted API call uses the same method as the reference API call. **Dialogue Success** measures the fraction of dialogues in which all reference API calls are predicted correctly. **Full API Accuracy** measures whether the predicted API call matches the reference in both method and required slot values; a call is counted as correct only when the method, parameter keys, and parameter values are all correct. **BLEU** measures lexical overlap between the generated dialogue and the reference dialogue after removing search-result blocks and formatting artifacts. **Inform Accuracy** measures how many user-requested slots are correctly answered by the system after an API call, normalized by the total number of requested slots. **Dialogue Completion Rate** measures the fraction of dialogues that reach a valid completed state rather than stopping prematurely.

LLM-Judged Pairwise Metrics. Because task metrics alone may not capture perceived response quality, we also conduct pairwise evaluation with an LLM judge. For each matched dialogue pair from two conditions, such as US vs. US+SA, the judge compares the system responses and selects the preferred condition on three dimensions, or declares a tie: **Constraint Satisfaction**: compares how many required slots are correctly provided by the system. A slot is counted only if its value is clear, supported by the search results, and not duplicated. The dialogue with the higher correct-slot count wins. **Truthfulness**: compares factual grounding with respect to the shared API calls and search results. The judge prefers the dialogue that reports facts correctly and avoids unsupported, contradictory, or false claims about API outcomes. **Personality-Conditioned User Satisfaction**: compares how well the system satisfies the user given the user's trait, steering coefficient, trait definition, and domain. The judge considers whether the tone, wording, and adaptation fit the user's

	Movies			Restaurants		
Metric	**Base**	**SA**	**Δ**	**Base**	**SA**	**Δ**
Standard evaluation						
Method Accuracy	45.9	45.9	0.0	54.3	58.7	+4.3
Dialog Success	20.0	20.0	0.0	16.0	24.0	+**8.0**
Full API Accuracy	27.0	27.0	0.0	39.1	41.3	+2.2
BLEU	35.7	35.2	−0.5	32.6	34.2	+1.6
Inform Accuracy	45.0	45.0	0.0	39.5	41.9	+2.3
Dialog Completion	8.0	4.0	−4.0	16.0	16.0	0.0
Pairwise judge						
User Satisfaction	8.0	16.0	+8.0	4.0	20.0	+**16.0**
Truthfulness	8.0	8.0	0.0	4.0	8.0	+4.0
Constraint	0.0	0.0	0.0	9.5	9.5	0.0

Table 1: The comparison of Baseline vs. SA by domain.

personality and preferences, without prioritizing task completion unless it clearly affects the interaction.

4 Results

We organize the results around the three research questions introduced in Section 1, using Qwen2.5-7B-Instruct as the primary evaluation model. We then report Llama-3.1-8B-Instruct separately as a cross-model robustness setting. Across all comparisons, we report both standard task metrics and LLM-judged pairwise quality metrics. The Baseline–SA comparison uses 50 dialogue pairs, with 25 per domain, while the US–US+SA comparison uses 2,000 dialogue pairs, covering 10 traits, 4 scalar strengths, 2 domains, and 25 dialogues per setting.

4.1 RQ1: Does system-side adaptive steering improve task-oriented dialogue when the user is neutral?

To answer RQ1, we compare Baseline and system-side adaptive steering (SA) under the same neutral-user setting. Because this comparison does not involve user-trait or scalar variation, we analyze the effect of SA by domain. This breakdown is important because averaging across domains can obscure whether SA provides broad improvements or benefits domains with more personality-relevant decision points. As shown in Table 1, SA's task-metric gains are concentrated in Restaurants, while Movies shows little or no task improvement but still improves judged user satisfaction. This suggests that SA is most useful in domains with richer personality-relevant slots or preferences, where adaptive system behavior can better tailor recommendations and communicate relevant options.

In Restaurants, SA improves every standard task metric except dialogue completion, which remains unchanged. Dialogue success rises from 16.0% to 24.0%, method accuracy improves by +4.3 pp, full API accuracy by +2.2 pp, inform accuracy by +2.3 pp, and BLEU by +1.6 pp. The pairwise judge shows the same direction: SA improves user satisfaction by +16.0 pp and truthfulness by +4.0 pp, while constraint satisfaction is tied. This suggests that in a structured slot-filling domain, SA improves both task behavior and perceived response quality.

Movies shows a different pattern. Method accuracy, full API accuracy, dialogue success, and inform accuracy are unchanged, while dialogue completion and BLEU decrease slightly. However, SA still improves judged user satisfaction by +8.0 pp, with truthfulness and constraint unchanged. Thus, in Movies, SA appears to affect response style more than structural task completion. This domain split explains why the Baseline–SA average should not be interpreted as a uniform improvement across tasks.

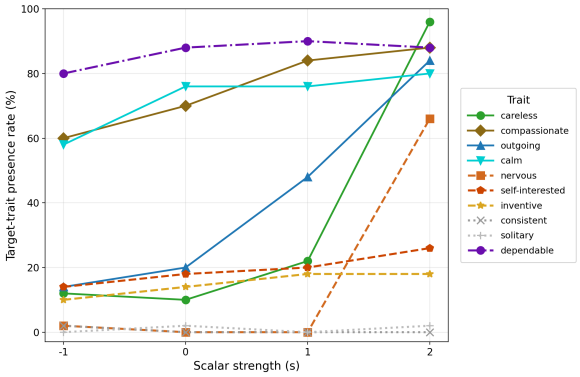


Figure 1: Target-trait presence rate by steering scalar strength. The intended trait is counted as present when it appears in the blind judge's top-three observed user traits.

4.2 RQ2: Does user-side activation steering produce observable personality traits in task-oriented dialogue?

To test whether user-side activation steering produces observable personality expression, we apply a blind trait-recognition judge to Baseline and US dialogues. The full evaluation prompt is shown in Appendix C.3. The judge reads the full dialogue and identifies the top three most evident user traits without being told the steered target. This setting tests whether the intended trait is visible from user behavior alone. The Baseline results show that the unsteered LLM already has a strong default style: 80% of Baseline dialogues are judged as *dependable* in the top position. Against this cooperative prior, target-trait recovery under US is uneven: across all 2,000 US dialogues, the intended trait appears in the judge's top three only 36.4% of the time. Thus, user-side steering does not make all target traits equally observable. Figure 1 shows that observability depends strongly on both trait and scalar strength. Some traits become clearly visible only at the strongest scalar: *careless* rises from 22% at α=1 to 96% at α=2, and *nervous* rises from 0% to 66%. Other traits, such as *outgoing* and *compassionate*, increase more gradually with scalar strength. In contrast, *consistent* and *solitary* remain almost unobservable across all scalar values, suggesting that internally oriented or interaction-suppressing traits are difficult to express in task-oriented dialogues. Overall, user-side activation steering produces observable personality expression, but only for some traits and mainly at sufficient steering strength.

4.3 RQ3: When users are activation-steered, how does system-side adaptive steering perform across different traits and scalar strengths?

We compare US and US+SA to evaluate whether adding system-side adaptation improves interactions with activation-steered users. Table 2 reports the aggregate comparison over directional user-steering coefficients only, $\alpha \in \{-1, 1, 2\}$, excluding the undirected α=0 condition. Under directional user steering, US+SA shows slight task-level changes. A paired t-test over the 60 matched (trait $\times$ scalar $\times$ domain) cells, with a Wilcoxon signed-rank check, shows that none of these standard-evaluation differences reach significance at $p < 0.05$. These results indicate that, at this sample size, system-side adaptation does not yield a statistically reliable advantage in standard task performance for activation-steered users.

The pairwise judge shows a clearer response-quality benefit, and this benefit is statistically supported: US+SA is preferred more often for truthfulness (+8.3 pp; exact McNemar-style test, $p < 0.000001$) and user satisfaction (+2.7 pp, $p = 0.0408$), while constraint satisfaction (+0.6 pp) is not significant ($p = 0.507$).

Overall, these results suggest that system-side adaptive steering provides slight, non-significant task-level gains under directional user steering, alongside statistically significant improvements in judged factual grounding and user satisfaction.

	Standard evaluation (%)						Pairwise judge (%)		
Experiments	**Meth.**	**API**	**Succ.**	**Completion**	**Inform**	**BLEU**	**Truth.**	**USat.**	**Constr.**
US	53.79	32.43	21.60	12.60	34.53	32.67	4.8	11.4	4.5
US+SA	54.49	32.99	22.13	12.53	33.91	32.72	13.1	14.1	5.1
Δpp	+0.70	+0.56	+0.53	−0.07	−0.62	+0.06	+8.3	+2.7	+0.6

Table 2: US vs. US+SA results using Qwen2.5-7B-Instruct, averaged over directional user-steering coefficients only ($\alpha \in \{-1, 1, 2\}$). Δ denotes US+SA minus US. Constraint satisfaction counts N/A pairs (no checkable slot) as non-wins; among decidable pairs, the US+SA advantage is +0.88 pp.

The aggregate averages mask substantial trait-level variation. Figure 2 shows US+SA pairwise win-rate advantages by trait and metric, averaged across all scalar values including α=0. Truthfulness improves for every trait (+4.0 to +10.0 pp). User satisfaction is strongly trait-dependent: it improves for socially expressive users (self-interested, nervous, compassionate, careless, outgoing) but decreases for consistent and solitary users, whose traits remain difficult to detect even at α=2 (Figure 1). When the user trait is weakly expressed, SA may not receive a clear personality signal and may default to a generic adaptive style, improving truthfulness while reducing persona-aligned satisfaction. Constraint satisfaction shows smaller, mixed changes across traits. Overall, SA appears most useful when the user trait creates a need for more careful, supportive, or socially responsive system behavior.

Scalar strength offers a complementary view of how user-side steering interacts with SA. As Figure 3 shows, US+SA improves truthfulness and user satisfaction across all scalar values, with the clearest gains at α=2. Constraint satisfaction is less consistent: it drops at α=0, where the steering pipeline is active but no directional shift is applied, creating a mismatch between user signal and system adaptation, then recovers at α=1 and improves at α=2.

SA compositions in US+SA dialogues (Appendix D) show a stable dominant pattern: SA defaults to *Conscientiousness*-centered behavior across most trait–scalar settings, regardless of the user's steered trait. The main deviation appears at α=2, where the system begins to adapt its composition to the user's expressed behavior: careless and outgoing users shift the system toward *Agreeableness+Extraversion*, while nervous and compassionate users shift it toward *Agreeableness+Conscientiousness*. This suggests that SA can adapt to the user's personality, but only when user-side steering is strong enough to produce a clearly observable behavioral signal.

Cross-model robustness. Full Llama-3.1-8B-Instruct results are reported in Appendix B. Llama confirms the same pairwise trend at high statistical significance on all three metrics, including constraint satisfaction, which was non-significant under Qwen (p=0.507) but highly significant under Llama (p=1.5×10^{-17}), suggesting the response-quality benefit of system-side adaptive steering is not a Qwen-specific artifact. Full significance results are in Appendix B.5. Trait- and scalar-level patterns remain model-dependent, indicating that while the direction of the effect generalizes, its magnitude and trait-specific profile require validation across model families.

5 Conclusion

We studied activation steering for controllable personality variation in task-oriented dialogue simulation. Using a dual-agent framework, we evaluated user-side activation steering (US), system-side adaptive steering (SA), and their combination (US+SA) on the Restaurants and Movies domains of the Schema-Guided Dialogue benchmark, with Qwen2.5-7B-Instruct as the primary model and Llama-3.1-8B-Instruct as a robustness setting.

Our results show that activation steering affects TOD behavior in a domain-, trait-, and steering scalar-dependent manner. With neutral users, SA improves task metrics and judged response quality mainly in Restaurants, while Movies shows weaker task-level effects but

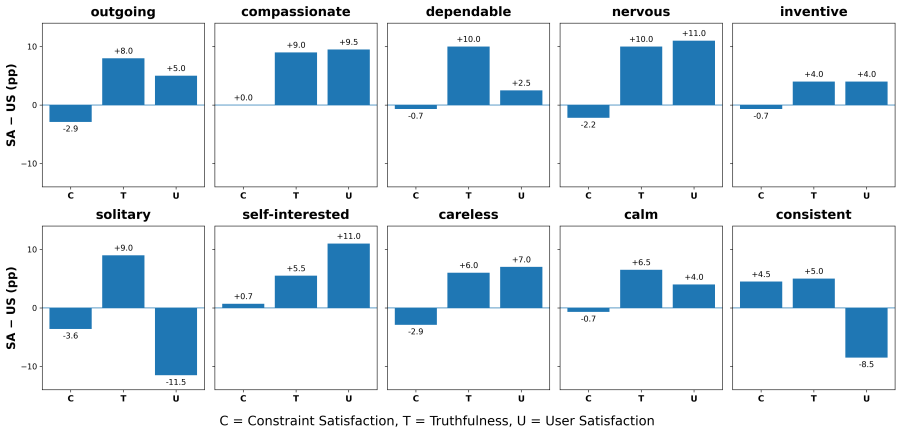


Figure 2: US+SA minus US pairwise win-rate advantage by trait and metric.

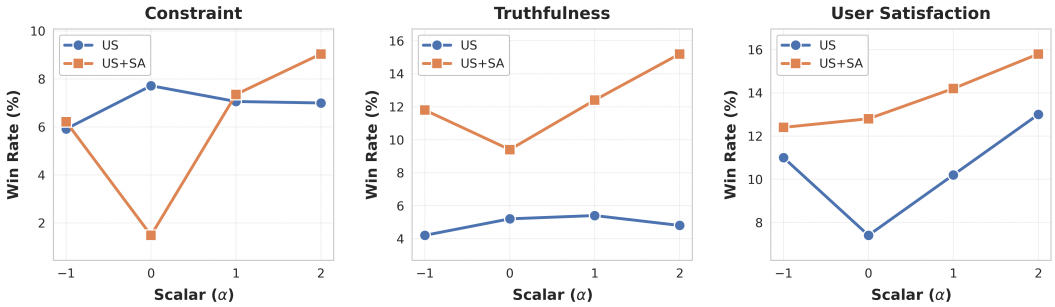


Figure 3: Pairwise win rates across persona scalar strengths for constraint, truthfulness, and user satisfaction.

improved user satisfaction. User-side steering produces observable personality expression only for some traits: socially or affectively expressive traits such as careless, outgoing, compassionate, and nervous become more detectable at stronger steering levels, whereas consistent and solitary remain difficult to observe in short TOD interactions.

When users are directionally steered, $\alpha \in \{-1, 1, 2\}$, adding SA yields slight task-level gains and clearer pairwise-quality improvements. In Qwen, US+SA yields a statistically significant improvement in truthfulness (+8.3 pp) and user satisfaction (+2.7 pp). These benefits are strongest when the user-personality signal is clear, especially at α=2, and vary substantially across traits. System-side composition analysis further shows that SA does not directly mirror the user's target trait; it usually defaults to Conscientiousness-centered service behavior and becomes more socially adaptive when the user trait is strongly expressed. The Llama robustness setting confirms the same pairwise trend at high statistical significance on all three metrics, including constraint satisfaction, which was non-significant under Qwen, suggesting the response-quality benefit generalizes across models, though trait- and scalar-level patterns remain model-dependent.

Overall, activation-level personality steering is a more controllable alternative to prompt-only conditioning for controlled TOD simulation, with clear benefits for perceived response quality and selective gains in task metrics. Future work should extend this framework to more domains and model families, and explore adaptive scalar-selection strategies for more reliable cross-model transfer.

Acknowledgment

This work was supported in part by the National Science Foundation under grant IIS-2401685.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour, and Yi Zhang. User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*, 2023.

Jia Deng, Tianyi Tang, Yanbin Yin, Wenhao Yang, Xin Zhao, and Ji-Rong Wen. Neuron based personality trait induction in large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiachong Feng, Liang Zhao, Weihong Zhong, Yichong Huang, Yuxuan Gu, Lingpeng Kong, Xiaocheng Feng, and Bing Qin. Persona: Dynamic and compositional inference-time personality control via activation vector algebra. *arXiv preprint arXiv:2602.15669*, 2026.

Lewis R Goldberg. An alternative "description of personality": The big-five factor structure. In *Personality and personality disorders*, pp. 34–47. Routledge, 2013.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. Personallm: Investigating the ability of large language models to express personality traits. In *Findings of the association for computational linguistics: NAACL 2024*, pp. 3605–3627, 2024.

Florian Kreyssig, Iñigo Casanueva, Paweł Budzianowski, and Milica Gasic. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 60–69, 2018.

Lucio La Cava and Andrea Tagarelli. Open models, closed minds? on agents capabilities in mimicking human personalities through open large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1355–1363, 2025.

Wenkai Li, Jiarui Liu, Andy Liu, Xuhui Zhou, Mona Diab, and Maarten Sap. Big5-chat: Shaping llm personalities through training on human-grounded data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20434–20471, 2025.

Hsien-chin Lin, Christian Geishauser, Shutong Feng, Nurul Lubis, Carel van Niekerk, Michael Heck, and Milica Gasic. Gentus: Simulating user behaviour and language in task-oriented dialogues with generative transformers. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 270–282, 2022.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Huang Xuanjing, et al. From individual to society: A survey on social simulation driven by large language model-based agents. *ACM Computing Surveys*, 58 (11):1–41, 2026.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

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

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Jost Schatzmann, Kallirroi Georgila, and Steve Young. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pp. 45–54, 2005.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126, 2006.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pp. 149–152, 2007.

Greg Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. Personality traits in large language models. *arXiv preprint arXiv:2307.00184*, 2023.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. How to build user simulators to train rl-based dialog systems. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 1990–2000, 2019.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. *URL https://arxiv. org/abs/2308.10248*, 2308, 2024.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. Conditional generation and snapshot learning in neural dialogue systems. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2153–2162, 2016.

Gerard Yeo, Fiona Tan An Ting, Kokil Jaidka, Shaz Furniturewala, Wu Fanyou, Weijie Xu, Vinija Jain, Aman Chadha, Yang Liu, and See Kiong Ng. Phantom: Persona-based prompting has an effect on theory-of-mind reasoning in large language models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pp. 2124–2142, 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency, 2023. *URL https://arxiv.org/abs/2310.01405*, 97, 2022.

Trait	Meth.	Succ.	Inf.	Truth.	Usat.	Constr.
	Standard evaluation, pp			Pairwise judge, pp		
self-interested	+0.14	+2.5	+0.32	+5.5	+11.0	+0.7
compassionate	+0.68	+1.5	+0.33	+9.0	+9.5	0.0
nervous	+1.56	+1.0	−2.75	+10.0	+11.0	−2.2
consistent	+1.97	+1.5	+0.75	+5.0	−8.5	+4.5
inventive	+1.36	0.0	+0.62	+4.0	+4.0	−0.7
careless	+0.68	+0.5	−2.17	+6.0	+7.0	−2.9
outgoing	+1.02	0.0	−2.17	+8.0	+5.0	−2.9
calm	−0.80	−1.5	−0.90	+6.5	+4.0	−0.7
dependable	+0.27	0.0	−1.25	+10.0	+2.5	−0.7
solitary	+0.20	0.0	−1.21	+9.0	−11.5	−3.6

Table 3: Full US+SA minus US deltas by personality trait. Standard evaluation metrics and pairwise judge metrics are reported in percentage points. Positive values indicate a US+SA advantage.

	Standard evaluation (Δ pp)						**Pairwise judge** (Δ pp)		
α	**Meth.**	**API**	**Succ.**	**Completion**	**Inform**	**BLEU**	**Truth.**	**USat.**	**Constr.**
−1	+0.66	+1.17	+1.00	0.00	−1.23	+0.28	+7.6	+1.4	+0.30
0	+0.73	−0.69	+0.60	−1.20	−1.50	+0.38	+4.2	+5.4	−6.23
1	+0.06	−0.11	0.00	−0.60	−1.23	−0.12	+7.0	+4.0	+0.29
2	+1.38	+0.62	+0.60	+0.40	+0.60	+0.01	+10.4	+2.8	+2.04

Table 4: US+SA minus US deltas by user-steering scalar strength. Values are reported in percentage points.

A Additional Qwen Results

A.1 Full Trait-Level US vs. US+SA Deltas

Table 3 reports US+SA minus US deltas by personality trait. The main paper visualizes the pairwise quality metrics in Figure 2; this table provides the corresponding numerical values for both standard task metrics and pairwise judge metrics. Positive values indicate a US+SA advantage.

A.2 Full Scalar-Level US vs. US+SA Deltas

Table 4 reports US+SA minus US deltas by user-steering scalar strength. The main paper visualizes the corresponding pairwise win rates in Figure 3. Positive values indicate a US+SA advantage.

A.3 Pairwise Win-Rate Interpretation

Pairwise win rates should be interpreted together with tie rates. In comparisons with many ties, a win-rate advantage does not mean that one condition dominates every dialogue; rather, it indicates that the judge prefers that condition more often among distinguishable cases. This is especially relevant for Baseline vs. SA, where several pairwise comparisons have high tie rates and the main SA effect appears in user satisfaction rather than constraint adherence.

Metric	Baseline	SA	US	US+SA
Method Accuracy	8.11	4.05	3.55	3.84
Dialog Success	2.00	0.00	0.53	0.93
Full API Accuracy	4.05	2.70	2.17	2.67
BLEU	12.18	13.15	12.61	12.42
Inform Accuracy	0.00	0.00	0.04	0.31
Dialog Completion	0.00	0.00	0.00	0.00

Table 5: Standard evaluation metrics across four Llama-3.1-8B-Instruct conditions, averaged over Movies and Restaurants. Baseline and SA use 50 dialogues each; US–US+SA uses 1,500 dialogues each ($\alpha \in \{-1, 1, 2\}$)

B Additional Llama-3.1-8B Results

B.1 Cross-Model Robustness: Llama as a Low-Task-Performance Setting

We also run the same four-condition protocol with Llama-3.1-8B-Instruct to test whether the effects of system adaptation persist under a different model. We do not average Llama with Qwen because the two models differ substantially in task execution ability. In particular, Llama performs near the floor on the SGD task: method accuracy remains below 10%, dialogue success is at most 2%, and dialogue completion is 0% in all conditions. Thus, Llama is better interpreted as a low-task-performance stress test rather than as a directly comparable replicate of the Qwen setting.

Despite weak standard task performance, Llama shows the same qualitative Baseline–SA pairwise pattern observed in Qwen. SA is preferred over the Baseline across all three judged quality metrics, improving user satisfaction by $+22.0$ pp, truthfulness by $+29.4$ pp, and constraint satisfaction by $+20.0$ pp overall. This suggests that SA can improve perceived conversational quality even when the underlying model struggles with multi-turn task execution.

For the US–US+SA comparison, Llama shows larger pairwise gains than Qwen: user satisfaction improves by +32.67 pp, truthfulness by +17.47 pp, and constraint adherence by +14.49 pp. This differs from Qwen, where the corresponding gains are smaller and constraint is nearly unchanged. We interpret this difference as a capacity-floor effect: when the base steered-user interactions are weak, SA improves general response quality and raises the constraint floor; when the base model is already stronger, as in Qwen, SA mainly improves truthfulness and satisfaction while leaving constraint adherence comparatively stable.

The trait and scalar analyses also show that some effects are model-specific. Nervous, self-interested, and compassionate users benefit from combined steering in both models, suggesting partial cross-model stability. However, traits that are counterproductive under Qwen, such as solitary and consistent, become positive under Llama. Similarly, the $\alpha = 0$ constraint drop observed in Qwen does not appear in Llama. These differences indicate that system adaptation's qualitative benefits are robust, but the precise trait- and scalar-level failure modes depend on the base model's task competence.

B.2 Standard Evaluation

Table 5 reports standard task metrics across the four Llama conditions. Llama performs near the floor on the SGD task: method accuracy remains below 10%, dialogue success is at most 2%, and dialogue completion is 0% in all conditions. These results indicate that Llama does not reliably complete the multi-turn API structure required by the task in our setup. For this reason, we interpret Llama standard metrics primarily as a diagnostic of model capability rather than as the main evidence for system adaptation effectiveness.

Comparison	Metric	Win rates A/B/Tie (%)	Δ
Baseline vs. SA	User Satisfaction	6.00 / 28.00 / 66.00	+22.00 pp
	Truthfulness	10.00 / 30.00 / 60.00	+20.00 pp
	Constraint	3.85 / 34.98 / 61.17	+31.14 pp
US vs. US+SA	User Satisfaction	6.00 / 38.67 / 55.33	+32.67 pp
	Truthfulness	5.47 / 22.93 / 71.60	+17.47 pp
	Constraint	10.20 / 24.69 / 65.11	+14.49 pp

Table 6: Llama pairwise win rates for neutral-user and steered-user comparisons. A denotes the first condition in each comparison and B denotes the second condition. Δ denotes B win rate minus A win rate in percentage points. Baseline–SA is evaluated over 50 dialogue pairs, and US–US+SA over 1,500 dialogue pairs, except for constraint satisfaction, which is evaluated over 34 of 50 pairs (Baseline–SA) and 1,018 of 1,500 pairs (US–US+SA) due to pairs lacking a checkable constraint slot.

B.3 Pairwise Quality Evaluation

Although Llama performs poorly on standard task metrics, Table 6 shows that system adaptation improves LLM-judged response quality in both the neutral-user and steered-user comparisons. In the Baseline–SA setting, SA is preferred over Baseline across all three pairwise metrics: user satisfaction improves by +22.0 pp, truthfulness by +20.0 pp, and constraint satisfaction by +31.14 pp. These gains occur despite high tie rates, indicating that many dialogue pairs are judged as comparable, but SA is preferred more often when the judge identifies a difference.

The US–US+SA comparison shows the same overall direction. Adding system-side adaptation to user-side steering produces large US+SA advantages in user satisfaction (+32.67 pp), truthfulness (+17.47 pp), and constraint satisfaction (+14.49 pp). This differs from the Qwen setting, where US+SA mainly improves truthfulness and user satisfaction while constraint satisfaction is nearly unchanged. We interpret the larger Llama gains as a capacity-floor effect: because Llama has weak task execution in the US condition, system adaptation can improve general response quality and raise the pairwise constraint-satisfaction floor even when standard task metrics remain low.

B.4 Trait and Scalar Patterns

Trait patterns. Table 7 reports US+SA minus US pairwise advantages by trait for Llama. Unlike Qwen, where solitary and consistent users showed negative user-satisfaction deltas, Llama shows positive US+SA gains for every trait. The strongest gains appear for nervous (+52.0 pp), self-interested (+44.0 pp), and compassionate (+35.3 pp) users, partially matching the strongest Qwen traits. However, the Qwen failure cases do not replicate: solitary and consistent users also benefit under Llama. This suggests that Llama's weaker US baseline leaves more room for system adaptation to improve judged response quality across traits.

Scalar patterns. Table 8 reports US+SA minus US advantages by scalar strength. US+SA improves user satisfaction, truthfulness, and constraint adherence at every scalar value. This differs from Qwen, where the $\alpha = 0$ setting produced a constraint drop. In Llama, constraint improves at $\alpha = 0$ (+19.9 pp), suggesting that the Qwen $\alpha = 0$ failure mode is model-specific rather than a general property of combined steering.

B.5 Statistical Significance

We test the Llama pairwise and standard-metric results for significance. For the pairwise judgments, which are paired binary preferences per dialogue pair, we use McNemar's exact test on discordant pairs. For the standard task metrics, we use a Wilcoxon signed-rank test over the 60 (trait $\times$ scalar $\times$ domain) cells at $\alpha \in \{-1, 1, 2\}$. For Baseline vs. SA, only two domain-level observations exist per condition, which is too few for a paired test; Fisher's

Trait	Usersat Δ	Truthfulness Δ	Constraint Δ
nervous	+52.0 pp	+18.7 pp	+20.6 pp
self-interested	+44.0 pp	+15.3 pp	+10.8 pp
compassionate	+35.3 pp	+11.3 pp	+10.9 pp
dependable	+35.3 pp	+13.3 pp	+14.9 pp
outgoing	+35.3 pp	+14.7 pp	+10.8 pp
careless	+30.7 pp	+16.7 pp	+8.8 pp
inventive	+28.0 pp	+24.7 pp	+23.5 pp
consistent	+26.7 pp	+24.7 pp	+23.5 pp
calm	+23.3 pp	+14.7 pp	+16.7 pp
solitary	+16.0 pp	+20.7 pp	+13.7 pp

Table 7: Llama US+SA minus US pairwise win-rate advantages by trait. All values are positive, indicating that no trait favors US over US+SA, using 150 dialogue pairs per trait for usersat/truthfulness and ∼101–102 evaluable pairs per trait for constraint ($s \in \{-1, 1, 2\}$).

Scalar	Usersat Δ	Truthfulness Δ	Constraint Δ
$\alpha = -1$	+33.0 pp	+16.8 pp	+17.7 pp
$\alpha = 0$	+32.2 pp	+22.2 pp	+19.9 pp
$\alpha = 1$	+33.4 pp	+19.8 pp	+18.3 pp
$\alpha = 2$	+31.6 pp	+15.8 pp	+10.3 pp

Table 8: Llama US+SA minus US pairwise win-rate advantages by scalar strength. All values are positive across all scalar levels.

exact test on the recoverable per-dialogue binary rates shows no significant difference (dialogue success 1/50 vs. 0/50, p=1.0).

Pairwise judgments. US+SA is favored over US at high significance on all three metrics: constraint (p=1.5 × 10^{-17}, 252 vs. 95 discordant pairs), truthfulness (p=3.1 × 10^{-39}, 344 vs. 82), and user satisfaction (p=1.4 × 10^{-88}, 580 vs. 90). The Baseline vs. SA comparison is also significant on all three metrics despite its smaller sample (p=6.3 × 10^{-3}, 4.1 × 10^{-2}, and 1.3 × 10^{-2} for constraint, truthfulness, and user satisfaction).

Standard metrics. No standard task metric shows a significant US vs. US+SA difference (Wilcoxon, $p > 0.05$ for method accuracy, full API accuracy, dialogue success, and BLEU); inform accuracy reaches nominal significance (p=0.02) but on only six nonzero-difference cells, and dialogue completion is uniformly zero. These nulls mirror the Qwen finding (§4.3) that system-side adaptation improves judged response quality without producing a reliable standard-metric advantage. For Baseline vs. SA, only two domain-level observations exist per condition, which is too few for a paired test; Fisher's exact test on the recoverable per-dialogue binary rates shows no significant difference (e.g., dialogue success 1/50 vs. 0/50, p=1.0).

C Prompts

C.1 System-Agent Instruction Prompt

We used the following instruction prompt for the system agent in each task-oriented dialogue domain. The prompt instructs the system to collect required slot values, make API calls using schema-defined parameters, and ground its final response in the returned database results.

System-Agent Instruction Prompt

Think of yourself as an expert chat assistant specialized in the {domain_name} domain. As an expert, your task is to generate the most natural and helpful responses for a given task-oriented dialogue context.
I will provide the Schema for the {domain_name} domain, a sample dialogue conversation between a System and a User, and optionally search results from the database. Understand the dialogue in relation to the Schema.
You can request slot values from the User to fulfill the user's current intent. Remember that required slots are more important than optional slots. Make sure you restrict your conversation to the slots relevant to the user's intent.
When making API calls, use column names from the Schema as parameters. Match the required and optional slots with the column names and use them to search in the API calls. Use the date format YYYY-MM-DD if a date is needed in the API call. Before making the call, ensure that you have gathered all required slots from the user. You can skip any parameters that are not necessary.
API Call Format:

```
APICall( method='intent_name',
        parameters={required_slot_1: value required_slot_2: value})
```

Once you have gathered enough information about the slot values, output the API call. You will be provided with the API call search results from the database right after the API call. Use the search results to respond appropriately and fulfill the user's current intent.
Schema:
Here is a Schema for the {domain_name}:

C.2 User-Simulator System Prompt

We used the following system prompt to instruct the simulated user agent. The prompt constrains the user simulator to produce a single grounded user utterance that is consistent with the dialogue history and aligned with the next expected API call.

User-Simulator System Prompt

You are a User Simulator for the {domain_name} domain.
Generate exactly one realistic user utterance. Output only the utterance text.
Do not include role tags such as `User:` or `System:`. Do not output an end-of-dialogue marker. The system is responsible for closing the conversation. Your response should be a natural user utterance only.
Grounding and Consistency Rules

- The dialogue history is authoritative. Your utterance must be consistent with it.
- Do not introduce new goals, requests, or tasks that are not implied by the dialogue history.
- You are given the next API call that the system is expected to execute.
- Your task is to steer the dialogue so that the system uses that API call, and only that API call.
- Do not request actions that would require an API call other than the provided next API call.

How to Use the Next API Call Information

- If `Next API Call` is provided and is not `N/A`, your utterance should naturally provide missing information needed for that call, ask for information aligned with the next required slots, or confirm information relevant to that call.
- If `Next API Call` is `N/A`, you must not request any new action from the system. Do not ask for booking, reservation, cancellation, ticket purchase, another search, or any follow-up task.
- If the last system message asks a question, answer it directly without adding a new request.

- If the last system message offers further help, respond politely that you have nothing else to ask. For example: *No, that's all, thanks.* The system will close the dialogue on its next turn.

Input Fields

- **Domain:** {domain_name}
- **Next API Call:** {next_api_call}
- **Next Required Slots:** {next_required_slots}
- **Last API Call:** {last_api_call}
- **Last Required Slots:** {last_required_slots}

Dialogue History
{dialogue_history}

C.3 Trait-Expression Evaluation Prompt

Trait-Expression Evaluation Prompt

You are an expert evaluator of personality expression in task-oriented dialogue.
Task: Identify the top three most evident personality traits expressed by the USER across the full dialogue, and rate the expression strength of each trait.
Possible user traits and definitions: {trait_descriptions}
Scoring scale: 1 = very low evidence of the trait 2 = low evidence of the trait 3 = moderate or unclear evidence 4 = high evidence of the trait 5 = very high evidence of the trait
Instructions:
1. Use the full dialogue, not a single utterance. 2. Judge only observable USER behavior, not SYSTEM behavior. 3. Do not infer personality beyond what is clearly supported. 4. Rank the three most evident traits from strongest to weakest. 5. The three traits must be different. 6. Return only valid JSON. 7. Do not include extra commentary, explanation, or markdown. 8. In reason fields, do not use double quotation marks. If quoting dialogue text, use single quotation marks.
Dialogue:{dialogue}
Return exactly this JSON format:
"top1_evident_trait": "", "top1_evident_trait_expression": "score": 1, "reason": "" , "top2_evident_trait": "", "top2_evident_trait_expression": "score": 1, "reason": "" , "top3_evident_trait": "", "top3_evident_trait_expression": "score": 1, "reason": ""

User trait	$\alpha = -1$	$\alpha = 0$	$\alpha = 1$	$\alpha = 2$
calm	C, A+C	C, A+C	C, A+C	C, A+C
careless	C, A+C	C, A+C	C, A+C	A+E, A
compassionate	C, A+C	C, A+C	C, A+C	A+C, C
consistent	C, A+C	C, A+C	C, A+C	C, A+C
dependable	C, A+C	C, A+C	C, A+C	C, A+C
inventive	C, A+C	C, A+C	C, A+C	C, A+C
nervous	C, A+C	C, A+C	C, A+C	A+C, C
outgoing	C, A+C	C, A+C	C, A+C	C, A+E
self-interested	C, A+C	C, A+C	C, A+C	C, A+C
solitary	C, A+C	C, A+C	C, A+C	C, A+C

Table 9: Dominant system-side adaptive steering (SA) compositions across user traits and steering scalar strengths. Each cell lists the two most frequent active SA composition vectors for that user trait and scalar value. The results show that SA is usually dominated by Conscientiousness-based compositions, while stronger user steering at $alpha$=2 induces more Agreeableness- and Extraversion-based adaptation for traits such as careless, nervous, and outgoing. C = Conscientiousness, A = Agreeableness, E = Extraversion.

D System-Side Adaptive Steering Composition Analysis

We further analyze the system-side adaptive steering (SA) coefficients in US+SA dialogues. Across all 2,000 dialogues, SA is applied over 9,999 system turns and activates in 7,304 turns (73.0%). Activation is more frequent during API-call turns, where SA activates in 2,930 out of 3,599 turns (81.4%), compared with 4,374 out of 6,400 text-response turns (68.3%). This suggests that SA is especially active when the system must make structured task decisions, while still frequently steering natural-language responses.

Table 9 summarizes the two most frequent active SA composition vectors for each user trait and steering scalar. The dominant pattern is highly stable: for most trait–scalar settings, SA is led by Conscientiousness-based compositions, especially C and A+C. This indicates that the adaptive system mostly defaults to a reliable, organized, and cooperative system agent style.

The main deviations appear at the strongest user-steering scalar, α=2. Careless users shift toward A+E and A, nervous and compassionate users shift toward A+C, and outgoing users show A+E as the second dominant composition. In contrast, consistent, dependable, self-interested, and solitary users remain dominated by C and A+C. Overall, SA does not directly mirror the user's target trait; instead, it defaults to Conscientiousness-centered system behavior and becomes more socially adaptive when the user-side trait is strongly expressed.