Revealing Side Effects of Using Style Features on Conversational Agents

Anonymous ACL submission

Abstract

Style instructions such as *helpful*, *friendly*, and *professional* are widely used to steer LLM-based conversational agents, yet the unintended effects of these controls remain poorly understood. We present a systematic study of cross-feature stylistic side effects: changes in unprompted style dimensions induced by explicitly prompting for another style feature. To ground our analysis in current practice, we survey 127 conversational-agent papers from the ACL Anthology and identify 13 frequently used style features. We then construct controlled synthetic dialogues in task-oriented and open-domain settings and use a pairwise LLM-as-a-Judge framework to estimate how prompting for one style feature affects others. Our results show that style features are not independent controls but are instead systematically entangled. We further evaluate four mitigation strategies: Prompting, Steering, DPO, and SFT, and find that although these methods can partially restore suppressed side features, they often weaken the originally intended style. These findings challenge the assumption that natural-language style prompts provide faithful and isolated control, and they motivate more principled multi-objective methods for safe and targeted stylistic steering in conversational agents.

1 Introduction

The use of natural language prompts to steer the persona and output style of LLMs is now a ubiquitous practice in conversational agent design. Developers and researchers routinely employ system instructions, such as "be friendly," "be professional," or "be helpful", to tailor model behaviors for specific applications ranging from mental health support to customer service (Feng et al., 2025; Zhao et al., 2025a; Rachidi et al., 2025). These prompt-based controls often ignore a fundamental yet untested assumption: that styles function as entangled, correlated features rather than independently controllable dimensions. In practice, prompting for a particular style implicitly activates a constellation of associated traits, shaped by training data, social conventions, and latent representations within the model.

In this work, we interrogate this entanglement and identify a pervasive phenomenon we term *stylistic side effects*: unintended and systematic behavioral shifts in styles distinct from the prompted control feature. While prior work examines prompt side effects and behavioral changes from using persona (i.e. an Asian persona) (Luz de Araujo and Roth, 2025; Gupta et al., 2024), a comprehensive statistical investigation into how stylistic controls interfere with one another remains absent.

We address this gap by presenting a systematic study of cross-feature stylistic side effects in LLMs. To ground our analysis in real-world usage, we first conduct a comprehensive survey of 127 conversational agent papers from the ACL Anthology (2023–2025), identifying 13 distinct, frequently used style features such as *helpful*, *empathetic*, and *friendly*. We then implement a controlled evaluation framework, generating synthetic dialogues across task-oriented and open-domain settings using eight models spanning local open-weight and API-served systems. By employing an LLM-as-a-Judge pairwise comparison protocol with win rates (Zheng et al., 2023b), we quantify how prompting for a Main Feature impacts the expression of unrelated Side Features.

Our results reveal that style features in high-dimensional space are deeply entangled. We demonstrate consistent and structured side effect patterns, such as prompting for *Friendly* significantly reduces perceived *Professional* in daily conversations across six of the eight models, while in task-oriented conversations the relationship flips: prompting for *Professional* reduces perceived *Friendly* across four models. Furthermore, we evaluate four mitigation strategies: Prompting, Steer-

ing, Direct Preference Optimization (DPO), and Supervised-Finetuning (SFT). We find that attempting to restore suppressed traits often degrades the primary intended style. These findings suggest that conventional prompting, steering, and post-training are insufficient to disentangle these opposing effects, highlighting the need for more principled, multi-objective approaches to safe stylistic steering.

- **We conduct a comprehensive survey** of style feature usage across 127 ACL Anthology papers and identify 13 frequently used style features.
- **We reveal Stylistic Side Effects** by measuring treatment effects of prompted style features on unprompted style dimensions, finding that style controls are deeply entangled.
- **We evaluate Side Effect Mitigation methods** and find that attempting to restore suppressed effects often degrades the primary intended style.

2 Related Work

Recent research on persona-oriented and personalized LLMs highlights the importance of stylistic control for interactive settings such as role-playing and long-term user adaptation (Tseng et al., 2024; Chevi et al., 2025).[1] The significance of style extends beyond user satisfaction to model safety, robustness, and evaluation reliability. Prior work shows that "superficial style alignment" can weaken safety guardrails by prioritizing format-level compliance over harmlessness (Xiao et al., 2025), while style biases in LLM-based evaluation can cause judges to favor pleasing outputs over factual substance (Feuer et al., 2025). Complementary benchmarks such as PersonalLLM (Zollo et al., 2025), PERSONAMEM (Jiang et al., 2025a,b), and Crab (He et al., 2025b) evaluate adaptive capabilities in personalized and role-playing settings, providing testbeds for measuring whether models maintain coherent behavior across users, roles, and interaction histories. Findings that model performance is brittle to stylistic perturbations (Truong et al., 2025) further motivate style-aware optimization methods that preserve effectiveness across stylistic contexts (Pu et al., 2024).

[1] Since style is closely related to broader notions such as persona, personalization, and role-playing, we review these areas insofar as they require reliable control over stylistic behavior.

Research on LLM stylistic steering has often relied on natural language prompts such as "be empathetic" to modulate behavior (Feng et al., 2025; Zhao et al., 2025a; Rachidi et al., 2025). Although prompting is flexible and requires no parameter updates, recent audits suggest that it can be superficial or counterproductive. Zheng et al. (2024) show that expert personas can degrade performance on objective tasks, while other studies find that persona or style prompts may act as retrieval cues for demographic caricatures, triggering latent stereotypes rather than enforcing precise behavioral constraints (Lutz et al., 2025; Malik et al., 2024).

To address prompting limitations, activation steering techniques such as Contrastive Activation Addition (CAA) have emerged as a way to modulate model behavior without fine-tuning (Panickssery et al., 2024; Zhang et al., 2025). Recent work extends this direction to more granular stylistic and persona control: Bo et al. (2025) align chatbots with personality traits via preference-based steering, while Zhao et al. (2025b) and Zhang et al. (2025) propose SteerX and StyleVector to better isolate style-related components for parameter-efficient transfer. Beyond steering, SFT and DPO have also been explored for related adaptation problems, including character-specific role-playing (Shao et al., 2023), Big Five personality induction (Li et al., 2025), persona-faithful role-playing (Peng and Shang, 2024), and role-style instruction following (Lu et al., 2025). These methods establish important progress toward controllable stylistic adaptation, but they still face the broader challenge that style-related features in high-dimensional representation space are often entangled with task content, user preference, safety behavior, and latent stereotypes. As a result, changing a target style dimension can unintentionally affect factuality, task competence, safety compliance, or persona consistency, motivating optimization methods that treat stylistic control as an entangled representation problem rather than a set of independent attributes.

3 Style Feature Extraction - A Survey

To ground our study in contemporary practice, we perform a systematic survey of conversational-agent papers published in the *ACL Anthology* between 2023 and 2025, following the PRISMA framework (Tugwell and Tovey, 2021) for literature identification, screening, and selection. We extract commonly used style features from the selected

papers. The overview of our pipeline is visualized in Figure 3. The distribution of extracted features is shown in Figure 4. Additionally, the summary of our survey is presented in Table 3.

Starting with all papers from ACL Anthology from January 2023 to June 2025, we first select papers that have keyword 'conversational agent', 'dialogue system', 'dialog system' and 'chatbot' in their titles or abstracts. Then we extract all style features used in agent prompts from the papers. Next, we transform these features to adjectives, to get a list of unique features. Then we identify candidate features by filtering for those with a frequency ≥ 5, which results in 16 features.

Because we use length-controlled generation to minimize the influence of length on LLM judge, we excluded *concise*, *detailed*, and *short*. As the result, we extracted 13 distinct style features: *efficient*, *impartial*, *expert*, *helpful*, *outgoing*, *friendly*, *curious*, *polite*, *engaging*, *empathetic*, *professional*, *informative*, and *caring*, that typify how recent papers employ prompt-based message control. A more detailed discussion on style features' utilization patterns from the survey is shown in Appendix N.

4 Data Generation and Annotation

In this section, we introduce a controlled generation and evaluation pipeline for estimating how explicit style feature prompts shift an agent's judged stylistic behavior. We distinguish between the **Main Feature**, the style feature explicitly specified in the generation prompt, and the **Side Feature**, the style feature evaluated in the resulting response. In this framing, each Main Feature is treated as the independent variable, and each Side Feature is treated as the measured outcome. The goal of the pipeline is to measure not only whether a prompted feature is successfully expressed, but also whether the prompt shifts the agent's response along other stylistic dimensions.

To cover diverse, real-world domains for message generation, we use two dialogue datasets. LMSYS-Chat-1M (Task, Zheng et al. 2023a) represents task-oriented inquiries, while DailyDialog (Daily, Li et al. 2017) represents open-domain, daily-life interactions. For each domain, we select 10 topics and sample 90 initial messages per topic.[2] In each conversation, only the initial message of each (all from human) is collected for initiating the interaction. Of these 90 messages, 10 are reserved for side effect identification and 80 are reserved for intervention experiments. These two subsets are disjoint, ensuring that the data used to identify side effects is separate from the data used to test interventions. Example topics from each domain are shown in Table 4.

We generate responses using LLMs that span a range of model families, sizes, and access modes. Six models are local open-weight models: `Llama-3.1-8B-Instruct` (`Llama3.1-8B`, Grattafiori et al., 2024), `Qwen3.5-9B` and `Qwen3.5-27B` (Qwen, 2026), `Gemma-4-31B` (`Gemma4-31B`) and `Gemma-4-E4B-it` (`Gemma4-E4B`, Google DeepMind, 2026b), and `GPT-OSS-20B` (OpenAI et al., 2025). Two additional models are accessed through hosted APIs: `GPT-5-mini` (OpenAI, 2025) and `Gemini-3.1-Flash-Lite` (`Gemini3.1-FL`, Google DeepMind, 2026a). This model set allows us to evaluate whether style-induced shifts are consistent across both local and API-served systems.

For response generation, we define two prompting conditions. In the *Styled* condition, a style feature is applied by querying the agent with the system prompt: *"You are having a conversation about {topic}. Please be {style_feature} in your response."* In the control condition, denoted *Neutral*, we use the same topic framing but remove the stylistic instruction: *"You are having a conversation about {topic}."* All generations use relatively high temperatures to provide sufficient variation for estimating stylistic effects.[3] To obtain a clean estimate and minimize intermediate confounding, we only record the agent's immediate response to the initiating message for data collection.

Because response length can confound pairwise style judgments, we match responses in expected length on a per-message basis. For each initial message m and model M, we first generate five *Neutral* responses without a length cap. We use rounded mean the word count of these responses and use it as the per-message, per-model target length $L_{m,M}$. We then append the instruction *"Please answer in around {$L_{m,M}$} words."* to every subsequent system prompt, including both the *Neutral* baseline prompt and all *Styled* prompts. Figure 5 verifies

[2] See Appendix B for topic selection details.

[3] Local models are sampled with $temperature = 0.7$, while API-served models, `GPT-5-mini` and `Gemini-3.1-Flash-Lite`, use $temperature = 1.0$. For models that expose a reasoning-level knob, we set it to its lowest available level.

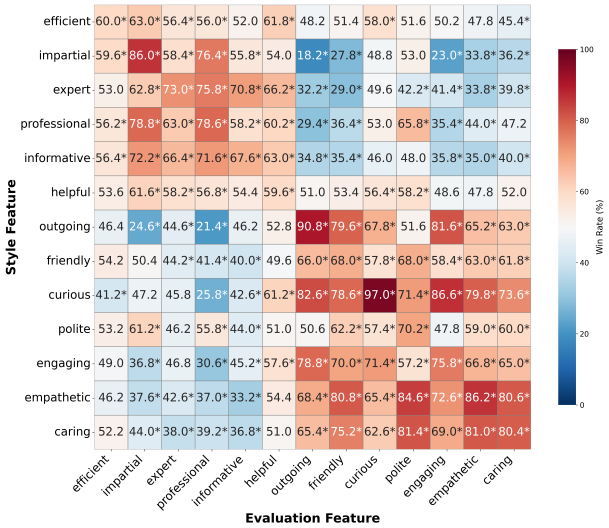

Figure 1: Style feature effect matrix for `GPT-5-mini` (daily domain). Rows are prompted Main Features, and columns are judged Side Features. Each cell is the win rate (%) of the styled response over the neutral baseline; * marks $p \leq 0.05$ (binomial). See other per-domain and per-model heatmaps in Figure 6.

that under this length cap the *Neutral* and *Styled* response-length distributions closely match the no-cap reference, confirming that the length confound is effectively removed.

We evaluate stylistic shifts using a pairwise comparison protocol. Each comparison pair consists of a *Styled* response generated with a given Main Feature and its corresponding *Neutral* response generated without that Main Feature. The pair is evaluated with respect to a target Side Feature, asking which response exhibits that feature more strongly. All pairwise judgments throughout the paper are produced by a single fixed judge, `GPT-OSS-120B` (OpenAI et al., 2025). During evaluation, the order of the two responses is randomized, and the judge is required to return structured output indicating the response with stronger Side Feature expression.[4] We use **win rate** as the primary metric for quantifying both the strength and direction of each prompt-induced effect. A win rate above chance indicates that prompting a Main Feature increases the expression of a given Side Feature, while a win rate below chance indicates a decrease.

5 Style Feature Effect Analysis

To examine how applying a Main Feature shifts model behavior along the Side Feature axis, we computed win rates for all pairwise combinations of 13 Main Features and 13 Side Features. For each message–Main Feature pair, responses were sampled five times. We then aggregated these win rates to form a *style feature effect matrix*, where rows correspond to prompted Main Features and columns correspond to evaluated Side Features.

This matrix measures the extent to which a particular style feature, such as *Friendly*, biases an agent toward specific stylistic dimensions, such as *Expert* or *Helpful*. In this way, the full win-rate matrix captures how explicitly imposing one style feature influences model behavior across a broader space of stylistic attributes. The color scale represents the average win rate: values above 50%, shown in red, indicate that the styled response exhibits a stronger tendency toward the corresponding Side Feature, whereas values below 50%, shown in blue, indicate a negative effect.

We present the *style feature effect matrix* for `GPT-5-mini` on the daily domain in Figure 1 as an example.[5] In general, the red diagonal shows that style prompting steers the model toward the intended feature effects. Beyond this primary effect, we observe that features naturally cluster by feature type: utility-oriented features (e.g., *Expert*, *Professional*, *Informative*, *Helpful*) consistently show high win rates against one another, and the same holds for interpersonal features (e.g., *Friendly*, *Polite*, *Engaging*, *Empathetic*). Crucially, this figure also reveals significant *side effects*, defined algorithmically in Section 6, across the two clusters: prompting the model to be *Friendly*, for instance, makes it noticeably less *Professional*.

6 Side Effect Identification

From the treatment effect analysis, we observe that style features, when applied, can produce unexpected stylistic changes in the agent's responses. We define this phenomenon as **side effect**, an unintended and unexpected behavioral shift in a style distinct from the prompted control feature. Attempting to define "unintended", we ground our selection in the Side Effect Score designed as follows:

Side Effect Score (SES). To rank candidate (Main Feature, Side Feature) pairs, we define a Side Effect Score that captures *how strongly* prompting a Main Feature m moves the response on a

[4] See Appendix A.1 for the evaluation prompt.

[5] Heatmaps for all 8 models in both domains can be found in Figure 6.

Domain	Model	Main Feature	Effect	Side Feature
Daily	Gemini3.1-FL	Friendly	Less	Professional
	Gemma4-31B	Friendly	Less	Professional
	Gemma4-E4B	Friendly	Less	Professional
	GPT-5-mini	Empathetic	More	Outgoing
	GPT-OSS-20B	Friendly	Less	Professional
	Llama3.1-8B	Curious	Less	Informative
	Qwen3.5-9B	Friendly	Less	Professional
	Qwen3.5-27B	Friendly	Less	Professional
Task	Gemini3.1-FL	Impartial	More	Professional
	Gemma4-31B	Impartial	More	Professional
	Gemma4-E4B	Expert	Less	Efficient
	GPT-5-mini	Professional	More	Empathetic
	GPT-OSS-20B	Professional	Less	Friendly
	Llama3.1-8B	Professional	Less	Friendly
	Qwen3.5-9B	Professional	Less	Friendly
	Qwen3.5-27B	Professional	Less	Friendly

Table 1: Top side effect pair selected for each domain and model with the largest SES. "Less" means the Main Feature unintentionally suppresses the Side Feature, and vice versa. See Top 3 side effect pairs for all models and domains in Table 6.

Side Feature axis s, relative to what the prior linguistic similarity between m and s would predict. Let $W(m, s) \in [0, 1]$ be the pairwise win rate of the styled response over the neutral baseline on the s-axis (judged by `GPT-OSS-120B`), and $C(m, s) \in [-1, 1]$ the cosine similarity of the OpenAI `text-embedding-3-large` (OpenAI, 2024b) embeddings of the two feature names. We standardise both quantities using the Median Absolute Deviation (MAD): N_{WR} centred at 0.5, Z_{CS} centred at the off-diagonal mean, and define

$$\begin{aligned}\text{SES}(m, s) &= \left|N_{\text{WR}} - Z_{\text{CS}}\right| \\ &\quad \cdot \min(|N_{\text{WR}}|, |Z_{\text{CS}}|) \\ &\quad \cdot \mathbf{1}[\, N_{\text{WR}}\, Z_{\text{CS}} < 0\,] \\ &\quad \cdot \mathbf{1}[\, p_W \leq 0.05\,].\end{aligned}$$

The term $\left|N_{\text{WR}} - Z_{\text{CS}}\right|$ measures empirical–semantic disagreement, while $\min(|N_{\text{WR}}|, |Z_{\text{CS}}|)$ acts as a magnitude floor that downweights cases where either signal is near zero. The score is non-zero only when (i) the empirical win-rate effect moves *opposite* to what semantic similarity predicts, and (ii) the win rate is significantly different from 0.5 ($p \leq 0.05$, two-sided binomial). The sign of N_{WR} supplies the intervention polarity: positive marks an unwanted *gain* on s (*suppress_positive*), negative an unwanted *loss* (*enhance_negative*). Full derivation, including the rationale for each factor and the per-domain main-feature filter, is in Appendix Q.

We select side effect pairs based on the comprehensive win rate matrices shown in Figure 6, which cover all models and domains and provide the empirical basis for our selection. In each domain, we restrict Main Feature to the corresponding feature category: utility-oriented features for the task domain and interpersonal features for the daily domain. In particular, for each model and domain, we rank candidate pairs by SES and choose the top-ranked pair as the intervention target. The resulting 16 selected pairs, corresponding to 8 models across 2 domains, are reported in Table 1.[6]

7 Side Effect Mitigation

Unintended agent behaviors are undesirable when applying stylistic conditioning. The goal of Side Effect Mitigation (Mitigation) is to alleviate the side effect while preserving the strength of the Main Feature. To address this trade-off, we evaluate four mitigation methods spanning a range of intervention strengths: Prompting, Steering, DPO, and SFT. We analyze their effectiveness by aiming to restore miscalibrated Side Feature effects without sacrificing the intended stylistic objective.

We prepare the experiment dataset using the intervention set described in Section 4. Experiments are domain-specific: each trained model uses data from only one domain. For each domain, we collect 800 initial messages, consisting of 80 messages per topic across 10 topics. We split these messages with stratified sampling by topic into train/validation/test splits using a 6:2:2 ratio, giving 48/16/16 messages per topic and 480/160/160 messages per domain. For training, each first-message is sampled 10 times, yielding 4,800 training examples. For validation and testing, each first-message is sampled 5 times, yielding 800 validation and 800 test examples.[7] Prompting does not use the training split; instead, it performs inference only on the test split.

We use **win rate** as the primary metric in this experiment. Following a procedure similar to that described in Section 4, each generated message is compared against its corresponding *Neutral* response. To further compare different methods, we introduce the following metrics. **ΔMain** and **ΔSide** denote the mean percentage-point changes in the Main Feature and Side Feature win rates, respectively, where **ΔSide** is measured in the desired direction. **M Flip** measures the proportion of cases in which mitigation moves the Main Feature win

[6] See Table 6 for the top-3 side effect pairs for each model in each domain.

[7] See training-set details in Appendix P.

rate across the 50% threshold, thereby rendering the mitigation ineffective. **S Flip** measures the proportion of cases in which the Side Feature win rate ends up on the desired side of 50% after mitigation, indicating successful mitigation.

7.1 Methods

Prompting As a simple mitigation strategy, we modify the system instruction to acknowledge the Side Feature explicitly, with a template that depends on the polarity of the side effect (Section 6). For *suppress_positive* cases (the Main Feature unintentionally elevates the Side Feature), we use *"Please be {main} but not {side}"*; for *enhance_negative* cases (the Main Feature unintentionally suppresses the Side Feature), we use *"Please be {main} but {side}."*

Steering We extract per-layer steering vectors via Contrastive Activation Addition (CAA, Panickssery et al., 2024): for each style feature we contrast activations elicited by paired *Styled*-vs-*Neutral* prompts (Appendix D), average the difference at a fixed token position over the training split, and pick the injection layer that has the strongest style feature effect on the validation split. At inference we condition the model on the single-style prompt *"Please be {main} in your response."* and add the Side Feature vector (or minus, depending on the polarity sign) to the chosen layer's activations during the forward pass, shifting the residual stream toward (or away from) the miscalibrated Side Feature without modifying weights. A detailed algorithm is in Appendix J.

DPO We train `Qwen3.5-9B`, `Qwen3.5-27B`, `Llama3.1-8B`, `GPT-OSS-20B`, `Gemma4-E4B`, and `Gemma4-31B` with DPO. We treat the Prompting intervention output (generate with *"Please be {main} but (not) {side}"*) as a preference target and styled-only output (generated with *"Please be {main}"*) as a rejected target to train each model within each domain on the 2,400 messages constructed from the training split, with the same initial message and the same per-message length constraint. The trained checkpoint is then prompted at test time with the same single-style prompt (no mitigation in the prompt) on the test set. Full hyperparameters and training setups are in Appendix P.

SFT SFT shares its data construction and training framework with DPO using Prompting intervention output as target output, while we use *"Please be {main}"* in its input prompt at inference time. As with DPO, the trained checkpoint is prompted at test time with the single-style prompt only. Full hyperparameters are in Appendix P.

Method	ΔMain (↑)	ΔSide (↑)	M Flip (↓)	S Flip (↑)
Prompting	**−13.5%**	28.5%	**12.5%**	**87.5%**
Steering	−21.9%	19.2%	41.7%	50.0%
DPO	−26.9%	**29.8%**	58.3%	50.0%
SFT	−22.1%	26.6%	16.7%	83.3%

Table 2: Mitigation results across 6 open-weight models. Header arrows mark the desired direction. **ΔMain** and **ΔSide** report mean changes in Main and Side Feature win rates, with **ΔSide** signed toward the desired direction. **M Flip** is the share of cases where mitigation moves the Main Feature win rate below 50%; **S Flip** is the share where the Side Feature win rate ends on the desired side of 50%.

7.2 Results

We report the efficacy of our Mitigation in Figure 2, Table 7 and per model summary in Table 2. The results reveal a complex landscape of trade-offs between maintaining stylistic adherence and mitigating unintended behavioral degradation.

Effectiveness of Side Effect Restoration The four mitigation methods, *Prompting*, *Steering*, *DPO*, and *SFT*, broadly alleviate the side effect relative to the non-intervened styled response, though the strength of restoration varies substantially across methods and side effects, as shown in Figure 2. For example, on `Qwen3.5-9B` in the Task domain, the *Professional* prompt originally suppresses *Friendly* to a win rate of 0.40; Prompting restores it to 0.73, Steering to 0.66, DPO to 0.96, and SFT to 0.68. Similarly, on `Llama3.1-8B` in the same domain, the suppressed *Friendly* score of 0.45 is recovered to 0.59 by Prompting, 0.69 by Steering, 0.86 by DPO, and 0.51 by SFT. However, restoration is not uniform. On `Gemma4-31B` in the Daily domain, where the *Friendly* prompt drives *Professional* down to 0.16, Prompting and SFT substantially recover the trait, reaching 0.62 and 0.57 respectively, while Steering (0.31) and DPO (0.17) leave it close to the suppressed baseline.

The Alignment Tax Although side effects are generally mitigated, preserving the strength of the Main Feature is more difficult. Figure 2 shows that some Main Feature win rates drop sharply after restoration, while others are maintained or

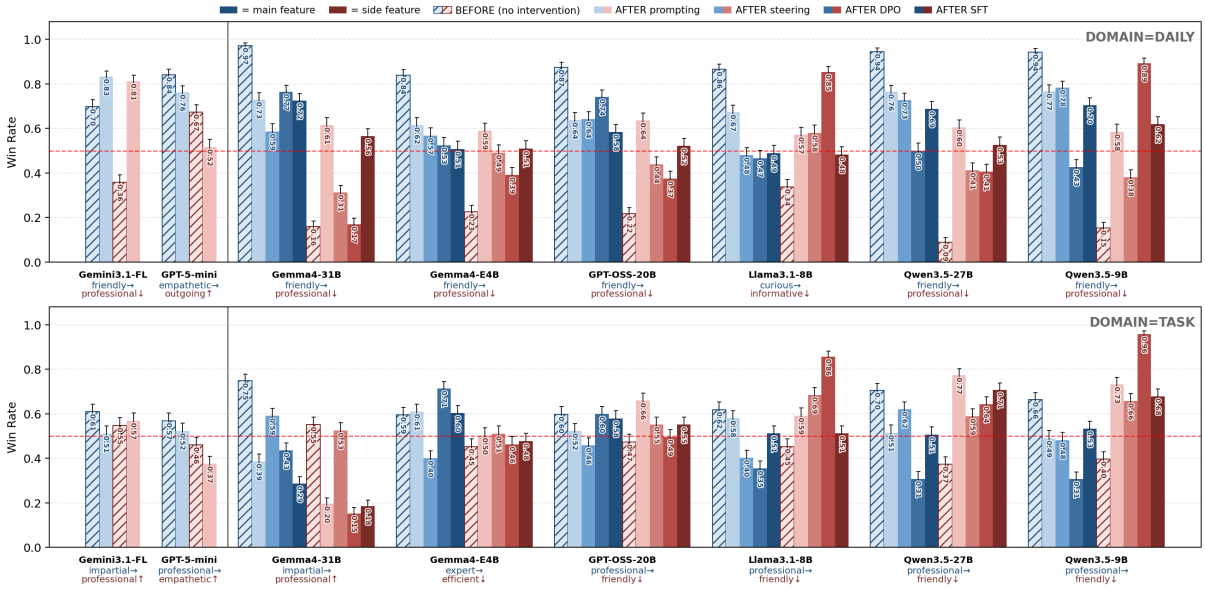

Figure 2: BEFORE/AFTER win rates for all four interventions (Prompting, Steering, DPO, SFT) across the 16 (model, domain) side effect pairs. Top panel: daily domain. Bottom panel: task domain. Per group: 5 blue bars (Main Feature) then 5 red bars (Side Feature); the hatched bar is BEFORE (no intervention) and the four solid bars darken light → dark in method order (prompting, steering, DPO, SFT). Error bars are 95% binomial CIs. No-training-only models (`gemini-flashlite`, `gpt5mini`, `gpt-oss-20b`) sit on the left of each panel, separated from the trainable open-weight models by a vertical divider. Judge: `GPT-OSS-120B`.

even strengthened. A high-degradation case appears for `Llama3.1-8B` in the Daily domain: when restoring *Informative* while prompting for *Curious*, the *Curious* win rate falls from 0.87 to 0.67 under Prompting, and further to 0.48, 0.47, and 0.49 under Steering, DPO, and SFT, respectively. This suggests that delivering information directly can conflict with the inquisitive stance required by *Curious*. In contrast, some feature pairs exhibit synergy rather than competition. For `Gemma4-E4B` in the Task domain, restoring *Efficient* while prompting for *Expert* preserves or improves the *Expert* win rate: compared with the original 0.60, Prompting reaches 0.61, DPO reaches 0.71, and SFT remains at 0.60. This indicates that expertise and efficiency can reinforce one another in task-oriented responses, rather than crowding each other out.

Method-Level Comparison Table 2 summarizes the aggregate performance of the four mitigation methods. *Prompting* achieves the smallest average Main Feature loss (ΔMain $= -13.5\%$), the lowest M Flip rate (12.5%), and the highest S Flip rate (87.5%), while remaining competitive in Side Feature restoration (ΔSide $= +28.5\%$). *DPO* obtains the highest average Side Feature gain (ΔSide $= +29.8\%$) and the strongest individual improvements, increasing *Professional* on `Qwen3.5-9B` in the Daily domain from 0.15 to 0.89 and *Friendly* on `Qwen3.5-9B` in the Task domain from 0.40 to 0.96. However, DPO also shows the largest average Main Feature decrease (ΔMain $= -26.9\%$) and the highest M Flip rate (58.3%). *SFT* falls between Prompting and DPO, with ΔSide $= +26.6\%$, ΔMain $= -22.1\%$, and an S Flip rate of 83.3%. *Steering* yields the smallest Side Feature gain (ΔSide $= +19.2\%$), a comparable Main Feature decrease to SFT (ΔMain $= -21.9\%$), and the lowest S Flip rate (50.0%). Overall, no single intervention dominates every case, but the aggregate results show that natural-language Prompting wins on three of the four metrics, whereas training-based methods trade higher peak Side Feature restoration for larger main-feature degradation.

8 Discussion

Skewed Style Vocabulary The frequency distribution in Figure 4 is heavily skewed: *helpful* (19 papers) and *empathetic* (15) lead, fourteen features tie at four mentions each in a long tail, and 87.0% of the 838 extracted mentions are single-word adjectives. Of the 342 surveyed conversational agents, 51.8% are styled with a single trait and 85.1% with three or fewer. This long-tail distribution shows that most prompts have their unique desired prompt

effects.

Model- and Domain-Invariant Side Effect Maps The same Main Features tend to have similar effects across models. Comparing the effect matrices of the two models within each domain in Figure 6, we find that whether the win rate is above or below the 50% threshold agrees in 86.9% of cases in the Daily domain and 83.8% of cases in the Task domain. The minimum agreement percentage is 78.2% in Daily domain and 73.7% in Task domain, showing that models have high agreement with each other on the directions of style features' effects. Similarly, each model has low cross-domain differences. For each Side Feature rating of each Main Feature used, the average $\Delta(Daily - Task)$ rating is -0.36 percentage points with a standard deviation of 1.18 percentage points, showing that domains don't have a strong impact on Main Feature's Side Feature impacts. The cross-model similarity may result from Artificial Hivement effects, meaning that different models generating very similar results given the same prompts (Jiang et al., 2025c).

Asymmetry of Style Feature Effects When prompting for feature A affects feature B's expression differently from how prompting for B affects A, the side effect is *asymmetric*. The cleanest cases are sign-flipped: prompting A raises B above the neutral 50% baseline while prompting B drops A below it, usually against the common expectation of how prompting works.

We identify a distinct directionality in the side effect patterns between style features (Appendix Figure 6), where the relationship between two features is often asymmetric. For example, in the Daily domain, instructing `Gemma4-31B` to be *Outgoing* raises its *Helpful* rating to 60.4%, but instructing it to be *Helpful* drops its *Outgoing* rating to 19.6%; `Qwen3.5-27B` shows the same flip (59.2% vs. 23.0%), as does `Gemma4-E4B` (62.4% vs. 32.4%). Additionally, in the Task domain, prompting `Qwen3.5-9B` to be *Curious* raises its *Expert* rating to 60.0%, while prompting it to be *Expert* drops its *Curious* rating to 29.8%; `Gemma4-31B` shows the same flip (56.4% vs. 19.2%), as does `Qwen3.5-27B` (55.6% vs. 20.0%). This asymmetry may reflect the priorities of instruction tuning and preference-based post-training: models are often optimized to be helpful, informative, and accurate as default behaviors, whereas interpersonal traits such as outgoingness or curiosity may be less consistently reinforced and therefore emerge primarily when explicitly elicited by the prompt.

Mitigation Trade-off Frontier Our mitigation results suggest that side effect correction is not merely a matter of applying a stronger intervention. Instead, mitigation exposes an *alignment tax*: restoring a suppressed Side Feature often weakens the intended Main Feature that the original style prompt was meant to induce. This pattern is clearest in the contrast between Prompting and DPO. Prompting provides the best aggregate trade-off, with the smallest Main Feature loss (ΔMain $= -13.5\%$), the lowest Main Feature flip rate (12.5%), and the highest Side Feature flip success rate (87.5%), while still substantially restoring the Side Feature (ΔSide $= +28.5\%$). DPO, by contrast, achieves the largest Side Feature gain (ΔSide $= +29.8\%$), but at the cost of the largest Main Feature degradation (ΔMain $= -26.9\%$) and the highest Main Feature flip rate (58.3%). These results indicate that training-based correction can over-optimize for the repaired attribute, replacing one stylistic failure with another rather than disentangling the two objectives. More broadly, mitigation should therefore be framed as a multi-objective optimization problem: effective interventions should seek Pareto improvements across both intended and unintended style dimensions, rather than maximizing Side Feature recovery alone.

9 Conclusion

This work shows that common style prompts for conversational agents can produce systematic and meaningful side effects. Through a survey of recent ACL Anthology papers and controlled experiments across Task and Daily dialogues, we find that prompting for one style feature often shifts other stylistic dimensions in unintended ways. They suggest that style features such as *friendly*, *professional*, *helpful*, and *expert* are deeply entangled rather than independently controllable attributes.

Our mitigation experiments further show that correcting these side effects is difficult. Prompting, activation steering, DPO, and SFT can often mitigate side effects, but this recovery frequently degrades the Main Feature intended by the original prompt. This trade-off suggests that stylistic control should be treated as a multi-objective optimization problem, where reliable mitigation requires preserving intended styles while constraining unintended changes along other dimensions.

Limitations

This study is deliberately scoped to controlled first-response conversational settings. This design is central to our objective: by holding the initiating message, topic frame, response length, and neutral baseline fixed, we can attribute changes in Side Feature judgments to the prompted Main Feature rather than to uncontrolled dialogue-history effects. The same framework naturally extends to longer human-agent interactions, where user adaptation, topic shifts, and accumulated context may reveal how the style-feature dependencies identified here evolve over sustained dialogue.

Our experimental grid is designed to capture style controls that are common in current conversational-agent research. We study 13 English adjective-style features extracted from recent ACL Anthology papers, two complementary dialogue domains, and eight generation models spanning open-weight and API-served systems. This scope enables systematic cross-feature, cross-domain, and cross-model comparison while keeping the full pairwise generation-and-rating sweep tractable. As conversational-agent practice expands to new languages, domains, and style vocabularies, the same survey-to-effect-matrix pipeline can be reused to construct broader maps of stylistic entanglement.

The mitigation experiments compare four representative intervention families: natural-language Prompting, activation Steering, DPO, and SFT. Evaluating these methods under a matched data split and a shared win-rate protocol allows us to characterize the trade-off frontier between restoring a Side Feature and preserving the prompted Main Feature. This setup also motivates extensions toward iterative re-prompting, explicitly Pareto- or multi-objective training, and interventions that address multiple Side Feature shifts simultaneously.

Finally, SES operationalizes a specific notion of salience: a statistically significant empirical shift whose direction differs from the prior expectation induced by semantic similarity between style names. The sign-discordance gate focuses the intervention targets on cases that are especially diagnostic of feature entanglement, rather than ranking expected co-movements among related features or expected separation among distant ones. The cosine-similarity centering is calibrated relative to the selected feature inventory, making SES a relative ranking tool for this study. Larger or domain-specific style taxonomies can re-estimate the same statistics, or introduce complementary scoring variants, to study additional forms of cross-feature interaction.

We use "intervention" in the narrow experimental sense of changing the system prompt while holding the initiating message, topic frame, response-length target, generator model, and evaluation protocol fixed. Our estimand is the effect of a Main-Feature prompt on the probability that a fixed judge rates the generated response as stronger along a Side-Feature axis than a neutral-prompt response. We do not claim to identify causal relations among latent style constructs themselves, nor do we claim that one style feature universally causes another across all prompts, users, judges, or multi-turn settings.

References

Yelaman Abdullin, Diego Molla, Bahadorreza Ofoghi, John Yearwood, and Qingyang Li. 2023. Synthetic dialogue dataset generation using LLM agents. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 181–191, Singapore. Association for Computational Linguistics.

Anum Afzal, Alexander Kowsik, Rajna Fani, and Florian Matthes. 2024. Towards optimizing and evaluating a retrieval augmented QA chatbot using LLMs with human-in-the-loop. In *Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (DaSH 2024)*, pages 4–16, Mexico City, Mexico. Association for Computational Linguistics.

Stuti Agrawal, Pranav Pillai, Nishi Uppuluri, Revanth Gangi Reddy, Sha Li, Gokhan Tur, Dilek Hakkani-Tur, and Heng Ji. 2024. Dialog flow induction for constrainable LLM-based chatbots. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 66–77, Kyoto, Japan. Association for Computational Linguistics.

Mariam ALMutairi, Lulwah AlKulaib, Melike Aktas, Sara Alsalamah, and Chang-Tien Lu. 2024. Synthetic Arabic medical dialogues using advanced multi-agent LLM techniques. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 11–26, Bangkok, Thailand. Association for Computational Linguistics.

Jessica Y. Bo, Tianyu Xu, Ishan Chatterjee, Katrina Passarella-Ward, Achin Kulshrestha, and D Shin. 2025. Steerable chatbots: Personalizing LLMs with preference-based activation steering. *arXiv preprint arXiv:2505.04260*.

Andrew Brown, Jiading Zhu, Mohamed Abdelwahab, Alec Dong, Cindy Wang, and Jonathan Rose. 2024.

Generation, distillation and evaluation of motivational interviewing-style reflections with a foundational language model. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1241–1252, St. Julian's, Malta. Association for Computational Linguistics.

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

Sijia Cheng, Wen Yu Chang, and Yun-Nung Chen. 2025. Exploring personality-aware interactions in salesperson dialogue agents. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 60–71, Bilbao, Spain. Association for Computational Linguistics.

Rendi Chevi, Kentaro Inui, Thamar Solorio, and Alham Fikri Aji. 2025. How individual traits and language styles shape preferences in open-ended user-LLM interaction: A preliminary study. *arXiv preprint arXiv:2504.17083*.

Yuntian Deng, Wenting Zhao, Jack Hessel, Xiang Ren, Claire Cardie, and Yejin Choi. 2024. WildVis: Open source visualizer for million-scale chat logs in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 497–506, Miami, Florida, USA. Association for Computational Linguistics.

Chengfeng Dou, Zhi Jin, Wenpin Jiao, Haiyan Zhao, Yongqiang Zhao, and Zhengwei Tao. 2023. PlugMed: Improving specificity in patient-centered medical dialogue generation using in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5050–5066, Singapore. Association for Computational Linguistics.

Jinhao Duan, Xinyu Zhao, Zhuoxuan Zhang, Eunhye Grace Ko, Lily Boddy, Chenan Wang, Tianhao Li, Alexander Rasgon, Junyuan Hong, Min Kyung Lee, Chenxi Yuan, Qi Long, Ying Ding, Tianlong Chen, and Kaidi Xu. 2025. GuideLLM: Exploring LLM-guided conversation with applications in autobiography interviewing. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5558–5588, Albuquerque, New Mexico. Association for Computational Linguistics.

Hao Fei, Han Zhang, Bin Wang, Lizi Liao, Qian Liu, and Erik Cambria. 2024. EmpathyEar: An open-source avatar multimodal empathetic chatbot. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 61–71, Bangkok, Thailand. Association for Computational Linguistics.

Qiming Feng, Qiujie Xie, Xiaolong Wang, Qingqiu Li, Yuejie Zhang, Rui Feng, Tao Zhang, and Shang Gao. 2025. EmoCharacter: Evaluating the emotional fidelity of role-playing agents in dialogues. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6218–6240, Albuquerque, New Mexico. Association for Computational Linguistics.

Yujie Feng, Bo Liu, Xiaoyu Dong, Zexin Lu, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2024. Continual dialogue state tracking via reason-of-select distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7075–7087, Bangkok, Thailand. Association for Computational Linguistics.

Amila Ferron, Amber Shore, Ekata Mitra, and Ameeta Agrawal. 2023. MEEP: Is this engaging? prompting large language models for dialogue evaluation in multilingual settings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2078–2100, Singapore. Association for Computational Linguistics.

Benjamin Feuer, Micah Goldblum, Teresa Datta, Sanjana Nambiar, Raz Besaleli, Samuel Dooley, Max Cembalest, and John P. Dickerson. 2025. Style outweighs substance: Failure modes of LLM judges in alignment benchmarking. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Sarah E. Finch and Jinho D. Choi. 2025. Leveraging explicit reasoning for inference integration in commonsense-augmented dialogue models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2222–2235, Abu Dhabi, UAE. Association for Computational Linguistics.

Sarah E. Finch, Ellie S. Paek, and Jinho D. Choi. 2023. Leveraging large language models for automated dialogue analysis. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 202–215, Prague, Czechia. Association for Computational Linguistics.

Google DeepMind. 2026a. Gemini 3.1 flash-lite. `https://deepmind.google/models/gemini/flash-lite/`. Accessed: 2026-05-23.

Google DeepMind. 2026b. Gemma 4. `https://deepmind.google/models/gemma/gemma-4/`. Accessed: 2026-05-23.

Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shashank Gupta, Vaishnavi Shrivastava, Ameet Deshpande, Ashwin Kalyan, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2024. Bias runs deep: Implicit reasoning biases in persona-assigned llms. *Preprint*, arXiv:2311.04892.

Junqing He, Liang Zhu, Rui Wang, Xi Wang, Gholamreza Haffari, and Jiaxing Zhang. 2025a. MADial-bench: Towards real-world evaluation of memory-augmented dialogue generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9902–9921, Albuquerque, New Mexico. Association for Computational Linguistics.

Kai He, Yucheng Huang, Wenqing Wang, Delong Ran, Dongming Sheng, Junxuan Huang, Qika Lin, Jiaxing Xu, Wenqiang Liu, and Mengling Feng. 2025b. Crab: A novel configurable role-playing LLM with assessing benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Pegah Jandaghi, Xianghai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2024. Faithful persona-based conversational dataset generation with large language models. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024)*, pages 114–139, Bangkok, Thailand. Association for Computational Linguistics.

Jihyoung Jang, Taeyoung Kim, and Hyounghun Kim. 2024. Mixed-session conversation with egocentric memory. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11786–11815, Miami, Florida, USA. Association for Computational Linguistics.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. 2025a. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *Proceedings of the 2025 Conference on Language Modeling (COLM)*.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, and 1 others. 2025b. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*.

Junfeng Jiang, Chengzhang Dong, Sadao Kurohashi, and Akiko Aizawa. 2023. SuperDialseg: A large-scale dataset for supervised dialogue segmentation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4086–4101, Singapore. Association for Computational Linguistics.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. 2025c. Artificial hivemind: The open-ended homogeneity of language models (and beyond). *Preprint*, arXiv:2510.22954.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina Semnani, and Monica Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9917–9955, Miami, Florida, USA. Association for Computational Linguistics.

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. 2024. Tell me what I need to know: Exploring LLM-based (personalized) abstractive multi-source meeting summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 920–939, Miami, Florida, US. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. MT-eval: A multi-turn capabilities evaluation benchmark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20153–20177, Miami, Florida, USA. Association for Computational Linguistics.

Fabian Lechner, Allison Lahnala, Charles Welch, and Lucie Flek. 2023. Challenges of GPT-3-based conversational agents for healthcare. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 619–630, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023. Prompted LLMs as chatbot modules for long open-domain conversation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4536–4554, Toronto, Canada. Association for Computational Linguistics.

Jeehyun Lee, Seung-Moo Yang, and Won Ik Cho. 2025a. AMAN: Agent for mentoring and assisting newbies in MMORPG. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 522–532, Abu Dhabi, UAE. Association for Computational Linguistics.

Seungbeen Lee, Seungwon Lim, Seungju Han, Giyeong Oh, Hyungjoo Chae, Jiwan Chung, Minju Kim, Beong-woo Kwak, Yeonsoo Lee, Dongha Lee, Jinyoung Yeo, and Youngjae Yu. 2025b. Do LLMs have

distinct and consistent personality? TRAIT: Personality testset designed for LLMs with psychometrics. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8397–8437, Albuquerque, New Mexico. Association for Computational Linguistics.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large scale language model society. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wenkai Li, Jiarui Liu, Andy Liu, Xuhui Zhou, Mona Diab, and Maarten Sap. 2025. 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)*, pages 20434–20471.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yu Li, Shang Qu, Jili Shen, Shangchao Min, and Zhou Yu. 2024a. Curriculum-driven edubot: A framework for developing language learning chatbots through synthesizing conversational data. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 400–419, Kyoto, Japan. Association for Computational Linguistics.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Dinuka Sahabandu, Bhaskar Ramasubramanian, and Radha Poovendran. 2024b. CleanGen: Mitigating backdoor attacks for generation tasks in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9101–9118, Miami, Florida, USA. Association for Computational Linguistics.

June M. Liu, He Cao, Renliang Sun, Rui Wang, Yu Li, and Jiaxing Zhang. 2025. CAPE: A Chinese dataset for appraisal-based emotional generation in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6291–6309, Albuquerque, New Mexico. Association for Computational Linguistics.

Junhua Liu, Tan Yong Keat, Bin Fu, and Kwan Hui Lim. 2024. LARA: Linguistic-adaptive retrieval-augmentation for multi-turn intent classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1096–1106, Miami, Florida, US. Association for Computational Linguistics.

Junru Lu, Jiazheng Li, Guodong Shen, Lin Gui, Siyu An, Yulan He, Di Yin, and Xing Sun. 2025. Rolemrc: A fine-grained composite benchmark for role-playing and instruction-following. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21008–21030.

Marlene Lutz, Indira Sen, Georg Ahnert, Elisa Rogers, and Markus Strohmaier. 2025. The prompt makes the person(a): A systematic evaluation of sociodemographic persona prompting for large language models. *Preprint*, arXiv:2507.16076.

Pedro Henrique Luz de Araujo and Benjamin Roth. 2025. Helpful assistant or fruitful facilitator? investigating how personas affect language model behavior. *PLOS One*, 20(6):e0325664.

Navid Madani, Anusha Bagalkotkar, Supriya Anand, Gabriel Arnson, Rohini K. Srihari, and Kenneth Joseph. 2025. A recipe for building a compliant real estate chatbot. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 213–235, Abu Dhabi, UAE. Association for Computational Linguistics.

Manuj Malik, Jing Jiang, and Kian Ming A. Chai. 2024. An empirical analysis of the writing styles of persona-assigned LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19369–19388, Miami, Florida, USA. Association for Computational Linguistics.

Nicholas Meade, Spandana Gella, Devamanyu Hazarika, Prakhar Gupta, Di Jin, Siva Reddy, Yang Liu, and Dilek Hakkani-Tur. 2023. Using in-context learning to improve dialogue safety. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11882–11910, Singapore. Association for Computational Linguistics.

Kshitij Mishra, Priyanshu Priya, Manisha Burja, and Asif Ekbal. 2023. e-THERAPIST: I suggest you to cultivate a mindset of positivity and nurture uplifting thoughts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13952–13967, Singapore. Association for Computational Linguistics.

Rajiv Movva, Pang Wei Koh, and Emma Pierson. 2024. Annotation alignment: Comparing LLM and human annotations of conversational safety. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9048–9062, Miami, Florida, USA. Association for Computational Linguistics.

Lidiya Murakhovs'ka, Philippe Laban, Tian Xie, Caiming Xiong, and Chien-Sheng Wu. 2023. Salespeople vs SalesBot: Exploring the role of educational value in conversational recommender systems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9823–9838, Singapore. Association for Computational Linguistics.

Ahmed Njifenjou, Virgile Sucal, Bassam Jabaian, and Fabrice Lefèvre. 2025. Enabling trait-based personality simulation in conversational LLM agents: Case

study of customer assistance in French. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 299–308, Bilbao, Spain. Association for Computational Linguistics.

Shinnosuke Nozue, Yuto Nakano, Shoji Moriya, Tomoki Ariyama, Kazuma Kokuta, Suchun Xie, Kai Sato, Shusaku Sone, Ryohei Kamei, Reina Akama, Yuichiroh Matsubayashi, and Keisuke Sakaguchi. 2024. A multimodal dialogue system to lead consensus building with emotion-displaying. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 669–673, Kyoto, Japan. Association for Computational Linguistics.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2024a. Gpt-4o mini model. `https://platform.openai.com/docs/models/gpt-4o-mini`. Accessed: 2026-01-05; a fast, affordable small AI model with large context support and multimodal capabilities.

OpenAI. 2024b. text-embedding-3-large Model. `https://developers.openai.com/api/docs/models/text-embedding-3-large`. Accessed: 2026-05-24.

OpenAI. 2025. Introducing gpt-5. OpenAI. Accessed: 2026-05-23.

Ryosuke Oshima, Seitaro Shinagawa, and Shigeo Morishima. 2024. The gap in the strategy of recovering task failure between GPT-4V and humans in a visual dialogue. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 728–745, Kyoto, Japan. Association for Computational Linguistics.

Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. DialogBench: Evaluating LLMs as human-like dialogue systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6137–6170, Mexico City, Mexico. Association for Computational Linguistics.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. *Preprint*, arXiv:2312.06681.

Letian Peng and Jingbo Shang. 2024. Quantifying and optimizing global faithfulness in persona-driven role-playing. *Advances in Neural Information Processing Systems*, 37:27556–27583.

Dominic Petrak, Thy Thy Tran, and Iryna Gurevych. 2024. Learning from implicit user feedback, emotions and demographic information in task-oriented and document-grounded dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4573–4603, Miami, Florida, USA. Association for Computational Linguistics.

Arpan Phukan, Manish Gupta, and Asif Ekbal. 2024. ECIS-VQG: Generation of entity-centric information-seeking questions from videos. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14411–14436, Miami, Florida, USA. Association for Computational Linguistics.

Xiao Pu, Tianxing He, and Xiaojun Wan. 2024. Stylecompress: An LLM-based prompt compression framework considering task-specific styles. *arXiv preprint arXiv:2410.14042*.

Qwen. 2026. Qwen3.5: Towards native multimodal agents. Qwen. Accessed: 2026-05-26.

Inass Rachidi, Anas Ezzakri, Jaime Bellver-Soler, and Luis Fernando D'Haro. 2025. Design, generation and evaluation of a synthetic dialogue dataset for contextually aware chatbots in art museums. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 20–28, Bilbao, Spain. Association for Computational Linguistics.

Ravi Shanker Raju, Swayambhoo Jain, Bo Li, Jonathan Lingjie Li, and Urmish Thakker. 2024. Constructing domain-specific evaluation sets for LLM-as-a-judge. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 167–181, Miami, Florida, USA. Association for Computational Linguistics.

Mansi Rana, Kadri Hacioglu, Sindhuja Gopalan, and Maragathamani Boothalingam. 2025. Zero-shot slot filling in the age of LLMs for dialogue systems. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 697–706, Abu Dhabi, UAE. Association for Computational Linguistics.

Gonçalo Raposo, Luisa Coheur, and Bruno Martins. 2023. Prompting, retrieval, training: An exploration of different approaches for task-oriented dialogue generation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 400–412, Prague, Czechia. Association for Computational Linguistics.

Traian Rebedea, Makesh Sreedhar, Shaona Ghosh, Jiaqi Zeng, and Christopher Parisien. 2024. CantTalkAboutThis: Aligning language models to stay on topic in dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12232–12252, Miami, Florida, USA. Association for Computational Linguistics.

Revanth Reddy, Hao Bai, Wentao Yao, Sharath Chandra Etagi Suresh, Heng Ji, and ChengXiang Zhai. 2023. Social commonsense-guided search query generation for open-domain knowledge-powered conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 873–885, Singapore. Association for Computational Linguistics.

Cristina Reguera-Gómez, Denis Paperno, and Maaike H. T. de Boer. 2025. Empathy vs neutrality: Designing and evaluating a natural chatbot for the healthcare domain. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 508–517, Tallinn, Estonia. University of Tartu Library.

Sougata Saha and Rohini Srihari. 2024. Integrating argumentation and hate-speech-based techniques for countering misinformation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11109–11124, Miami, Florida, USA. Association for Computational Linguistics.

Vishal Vivek Saley, Goonjan Saha, Rocktim Jyoti Das, Dinesh Raghu, and Mausam . 2024. MediTOD: An English dialogue dataset for medical history taking with comprehensive annotations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16843–16877, Miami, Florida, USA. Association for Computational Linguistics.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2387–2413, Singapore. Association for Computational Linguistics.

Seungyeon Seo and Gary Geunbae Lee. 2024. DiagESC: Dialogue synthesis for integrating depression diagnosis into emotional support conversation. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 686–698, Kyoto, Japan. Association for Computational Linguistics.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187.

Clemencia Siro, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Context does matter: Implications for crowdsourced evaluation labels in task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1258–1273, Mexico City, Mexico. Association for Computational Linguistics.

Li Siyan, Teresa Shao, Julia Hirschberg, and Zhou Yu. 2024a. Using adaptive empathetic responses for teaching English. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 34–53, Mexico City, Mexico. Association for Computational Linguistics.

Li Siyan, Teresa Shao, Zhou Yu, and Julia Hirschberg. 2024b. EDEN: Empathetic dialogues for English learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3492–3511, Miami, Florida, USA. Association for Computational Linguistics.

Dominic Sobhani, Ruiqi Zhong, Edison Marrese-Taylor, Keisuke Sakaguchi, and Yutaka Matsuo. 2025. Language models can categorize system inputs for performance analysis. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6241–6257, Albuquerque, New Mexico. Association for Computational Linguistics.

Rickard Stureborg, Sanxing Chen, Roy Xie, Aayushi Patel, Christopher Li, Chloe Zhu, Tingnan Hu, Jun Yang, and Bhuwan Dhingra. 2024. Tailoring vaccine messaging with common-ground opinions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2553–2575, Mexico City, Mexico. Association for Computational Linguistics.

Albert Sun, Varun Nair, Elliot Schumacher, and Anitha Kannan. 2024. CONSCENDI: A contrastive and scenario-guided distillation approach to guardrail models for virtual assistants. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4009–4030, Mexico City, Mexico. Association for Computational Linguistics.

Ekaterina Svikhnushina and Pearl Pu. 2023. Approximating online human evaluation of social chatbots with prompting. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 268–281, Prague, Czechia. Association for Computational Linguistics.

Maksym Taranukhin, Sahithya Ravi, Gabor Lukacs, Evangelos Milios, and Vered Shwartz. 2024. Empowering air travelers: A chatbot for Canadian air passenger rights. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 326–335, Miami, FL, USA. Association for Computational Linguistics.

Kimberly Truong, Riccardo Fogliato, Hoda Heidari, and Steven Wu. 2025. Persona-augmented benchmarking: Evaluating LLMs across diverse writing styles. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22676–22709, Suzhou, China. Association for Computational Linguistics.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. 2024. Two tales of persona in llms: A survey of role-playing and personalization. *Preprint*, arXiv:2406.01171.

Yuka Tsubota and Yoshinobu Kano. 2024. Text generation indistinguishable from target person by prompting few examples using LLM. In *Proceedings of the 2nd International AIWolfDial Workshop*, pages 13–20, Tokyo, Japan. Association for Computational Linguistics.

Peter Tugwell and David Tovey. 2021. Prisma 2020. *Journal of Clinical Epidemiology*, 134:A5–A6.

Dirk Väth, Lindsey Vanderlyn, and Ngoc Thang Vu. 2024. Towards a zero-data, controllable, adaptive dialog system. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16433–16449, Torino, Italia. ELRA and ICCL.

Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023a. Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12047–12064, Singapore. Association for Computational Linguistics.

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

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024a. Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9707–9731, Bangkok, Thailand. Association for Computational Linguistics.

Lanrui Wang, Jiangnan Li, Chenxu Yang, Zheng Lin, Hongyin Tang, Huan Liu, Yanan Cao, Jingang Wang, and Weiping Wang. 2025. Sibyl: Empowering empathetic dialogue generation in large language models via sensible and visionary commonsense inference. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 123–140, Abu Dhabi, UAE. Association for Computational Linguistics.

Xingguang Wang, Xuxin Cheng, Juntong Song, Tong Zhang, and Cheng Niu. 2024b. Enhancing dialogue state tracking models through LLM-backed user-agents simulation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8724–8741, Bangkok, Thailand. Association for Computational Linguistics.

Milan Wevelsiep, Nicholas Thomas Walker, Nicolas Wagner, and Stefan Ultes. 2025. A voice-controlled dialogue system for NPC interaction using large language models. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 29–38, Bilbao, Spain. Association for Computational Linguistics.

Yuxin Xiao, Sana Tonekaboni, Walter Gerych, Vinith Suriyakumar, and Marzyeh Ghassemi. 2025. When style breaks safety: Defending LLMs against superficial style alignment. *arXiv preprint arXiv:2506.07452*.

Fangyuan Xu, Kyle Lo, Luca Soldaini, Bailey Kuehl, Eunsol Choi, and David Wadden. 2024a. KIWI: A dataset of knowledge-intensive writing instructions for answering research questions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12969–12990, Bangkok, Thailand. Association for Computational Linguistics.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024b. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605, Bangkok, Thailand. Association for Computational Linguistics.

Chenghao Yang and Allyson Ettinger. 2023. Can you follow me? testing situational understanding for ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6385–6398, Singapore. Association for Computational Linguistics.

Yizhe Yang, Palakorn Achananuparp, Heyan Huang, Jing Jiang, and Ee-Peng Lim. 2024. Speaker verification in agent-generated conversations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5655–5676, Bangkok, Thailand. Association for Computational Linguistics.

Jiong Yu, Sixing Wu, Jiahao Chen, and Wei Zhou. 2024. LLMs as collaborator: Demands-guided collaborative retrieval-augmented generation for commonsense knowledge-grounded open-domain dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13586–13612, Miami, Florida, USA. Association for Computational Linguistics.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. 2025. Personalized text generation with contrastive activation steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7128–7141, Vienna, Austria. Association for Computational Linguistics.

Tenggan Zhang, Xinjie Zhang, Jinming Zhao, Li Zhou, and Qin Jin. 2024a. ESCoT: Towards interpretable emotional support dialogue systems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13395–13412, Bangkok, Thailand. Association for Computational Linguistics.

Yiqun Zhang, Fanheng Kong, Peidong Wang, Shuang Sun, Lingshuai Wang, Shi Feng, Daling Wang, Yifei Zhang, and Kaisong Song. 2024b. STICKERCONV: Generating multimodal empathetic responses from scratch. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7707–7733, Bangkok, Thailand. Association for Computational Linguistics.

Weiliang Zhao, Daniel Ben-Levi, Wei Hao, Junfeng Yang, and Chengzhi Mao. 2025a. Diversity helps jailbreak large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4647–4680, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiaoyan Zhao, Ming Yan, Yilun Qiu, Haoting Ni, Yang Zhang, Fuli Feng, Hong Cheng, and Tat-Seng Chua. 2025b. Steerx: Disentangled steering for llm personalization. *arXiv preprint arXiv:2510.22256*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric. P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024. When "a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. *Preprint*, arXiv:2311.10054.

Junkai Zhou, Liang Pang, Huawei Shen, and Xueqi Cheng. 2024. Think before you speak: Cultivating communication skills of large language models via inner monologue. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3925–3951, Mexico City, Mexico. Association for Computational Linguistics.

Zhouan Zhu, Shangfei Wang, Yuxin Wang, and Jiaqiang Wu. 2025. Integrating visual modalities with large language models for mental health support. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8939–8954, Abu Dhabi, UAE. Association for Computational Linguistics.

Thomas Zollo, Andrew Siah, Naimeng Ye, Li Li, and Hongseok Namkoong. 2025. Personalllm: Tailoring llms to individual preferences. In *International Conference on Learning Representations*, volume 2025, pages 66949–66971.

A Prompt Templates

In this section, we present the key prompt templates used in this project.

A.1 LLM-as-a-Judge Prompt Template

System: *You are an expert evaluator of conversation quality.*
User: *Compare the following two assistant responses and determine which one is more {feature}. Response A: {response_a} Response B: {response_b} You must choose one. Which response is more {feature}? Answer with ONLY "A" or "B" (no ties allowed). Answer:*

where the "{feature}" refers to one of the 12 features.

A.2 System Prompt Template for Message Generation With and Without Style Features

Neutral: *"This is a conversation about {topic}."*
Styled: *"This is a conversation about {topic}. Please be {style_features} in your response."*

A.3 Side Effect Mitigation Steerng Prompt Template

Since we use steering vector to change model behaviors, we don't include the Side Feature in the prompt to models: *"Please be {main_feature} in your response."*

B Details in Selection of Topics

To represent a wide range of domains and topics in message generation, we select LMSYS-Chat-1M for task-oriented inquiries and DailyDialog for casual, daily-life interactions. For each domain, we each pick 10 topics. Details of the selected topics, and examples are shown in Table 4

LMSYS-Chat-1M presents top 20 topics (clusters) in the paper, but the released dataset does not have them annotated. Instead, we first merge similar topics from the initial 20 and curate down to 10 representative topics, and used OpenAI's gpt-4o-mini model (OpenAI, 2024a) to annotate full dataset until each topic has 10 messages. On DailyDialog, we directly use the 10 topics suggested by the paper. These curation result in a total of

200 unique conversational contexts, balancing functional assistance with casual chatter.

C Topic Annotation and Initial-Message Sampling

This section details (i) how we annotated user prompts from LMSYS-Chat-1M and DailyDialog into a fixed task-domain and daily-domain taxonomy, and (ii) how the resulting label pools were sampled down to the per-topic initial-message sets that drive the side effect generation and intervention experiments. Two seed scales are described: the original 10 messages per topic used for side effect identification, and the extended 80 messages per topic used for the mitigation experiments.

C.1 Source Datasets

For the task domain we draw on two LMSYS-released resources. **LMSYS-Chat-1M** (Zheng et al., 2023a) provides one million conversations collected from the Chatbot Arena interface across 25 open-weights and proprietary models; in the paper, the authors cluster the first user prompts into 20 representative topic categories (e.g. *Python coding assistance*, *role-playing characters*, *Geography/travel inquiries*). We use both (a) the original `lmsys/lmsys-chat-1m` corpus for taxonomy curation and the larger 80-per-topic extension, and (b) the smaller, paired-completion `lmsys/chatbot_arena_conversations` corpus for the original 10-per-topic seed set. For the daily domain we use **DailyDialog** (Li et al., 2017), a manually labeled English chitchat corpus of 13,118 multi-turn dialogues. The paper defines 10 daily-life topics together with their per-dialogue proportions (Table 4 of Li et al. (2017)). The dataset's original distribution site (`yanran.li`) is no longer reachable; we obtain dialogue text via the community mirror `huggingface.co/datasets/roskoN/dailydialog`, which preserves the dialogue, dialogue-act, and emotion files but does *not* preserve the dialogue-level topic labels. We therefore re-annotate topics from scratch via `gpt-4o-mini`, as described below.

C.2 Curating a Task-Domain Taxonomy

The LMSYS-Chat-1M paper's 20 inferred clusters contain semantic overlaps (e.g. separate clusters for *Python coding assistance* vs. *Python coding examples*; separate clusters for *Role-playing characters* vs. *Role-playing AI chatbot capabilities*; multiple clusters covering text manipulation). To obtain a tractable label space we manually merged near-duplicate clusters and dropped categories whose surface form was either redundant or ill-suited as a stylistic conversation seed (NSFW role-play, SQL queries, factual-consistency probing, sentence-generation for job roles). The final task taxonomy contains the following 10 topics, listed verbatim as used by all downstream annotation calls:

- Discussing software errors and solutions
- Inquiries about AI tools, software design, and programming
- Geography, travel, and global cultural inquiries
- Text processing — translation, summarization, rewriting, elaboration, and factual-consistency checks
- Creating and improving business strategies and products
- Role-playing scenarios and character/chatbot interactions
- Answering questions based on passages
- Discussing and describing various characters
- Inquiries about specific plant growth conditions
- Requesting introductions for various chemical companies

For the daily domain we adopt the 10 DailyDialog topics directly from the paper: *Ordinary Life*, *School Life*, *Culture & Education*, *Attitude & Emotion*, *Relationship*, *Tourism*, *Health*, *Work*, *Politics*, *Finance*.

C.3 Annotation Pipeline

Task-domain annotation (`LMSYS-Chat-1M`, GPT-4o-mini). We iterate over the conversations in the dataset and extract the first user message of each. Each initial message is passed to `gpt-4o-mini` (OpenAI, 2024a) with an 11-way single-label classification prompt: the 10 task topics above followed by a catch-all *Other* bucket. The model is instructed to respond with a single integer in $\{1, \ldots, 11\}$; light post-processing recovers cases where the model emits a prefix such as "Domain k" before the digit. Each labeled message is appended to a thread-safe checkpoint CSV

(task_categorizations_checkpoint.csv) immediately after the API call so that no completed annotation is ever lost across restarts, and a 32-way parallel pool drives the call rate. This produced 25,650 labeled first-message records in our snapshot.

Daily-domain annotation (DailyDialog via roskoN/dailydialog, GPT-4o-mini). Because the upstream topic labels are no longer accessible, we re-derive them with the same protocol as the task-domain annotator. We download the train, validation, and test zip archives directly from the HuggingFace mirror, parse the dialogues_{split}.txt files, split each line on the dataset's __eou__ delimiter, and retain the dialogue's first utterance as the seed message. To keep the classifier signal strong — DailyDialog openers are often short greetings ("Hi.", "Excuse me.") that carry little topical content on their own — the *full* dialogue text (truncated to 2,000 characters) is passed to gpt-4o-mini with the same 11-way prompt over the 10 DailyDialog topics + *Other*, while only the first utterance is stored as the eventual seed. Annotations are written incrementally to a resumable checkpoint (daily_categorizations_checkpoint.csv) with a thread-safe append guarded by a single header initialization.

Direct classification of LMSYS-Chat-1M into the paper taxonomy (GPT-5-mini). As a robustness check on the task-domain taxonomy — and as the source pool for the 80-per-topic extension described below — we additionally classify lmsys/lmsys-chat-1m end-to-end: filter to English first messages with 32–1,536 characters and deduplicate, yielding 401,746 prompts; then use gpt-5-mini as a 21-way text classifier that assigns each prompt to one of the LMSYS-Chat-1M paper's 20 verbatim cluster labels (Zheng et al., 2023a) plus an *Other* bucket. The categories are the paper's published labels; gpt-5-mini reads each prompt's raw text and returns a category integer, with no embedding or re-clustering step. The resulting label set contains 198,501 in-taxonomy assignments after dropping *Other*, which serves as the pool the 80-per-topic extension samples from after mapping the 20 paper categories onto our 10 project task topics.

C.4 Sampling 10 Initial Messages per Topic (Side Effect Identification)

The side effect generation experiments require a fixed, balanced pool of *seed* initial messages, one set of 10 per topic, so that every reported win rate is computed over a comparable substrate of conversational contexts.

Task domain. We group all messages by their assigned topic, drop the *Other* bucket, deduplicate exact-string repeats, and stream candidates in dataset order. For each topic we accept a candidate if (a) it is between 32 and 512 characters long, (b) it is not a near-duplicate of an already-accepted candidate within the topic (case-folded, whitespace-collapsed comparison), and (c) it is not a malformed or empty turn. We stop accepting once 10 candidates have been retained for that topic.

Daily domain. The same procedure is applied to the daily-domain checkpoint, restricted to the 10 DailyDialog topics (*Other* dropped), again retaining 10 unique messages per topic and assigning domain = daily.

Resulting seed set. The final file unique_first_messages.csv contains exactly 200 rows: 10 task topics $\times$ 10 messages plus 10 daily topics $\times$ 10 messages. Every conversation in the side effect generation pipeline (Section 4, all model families, all style features, all repetitions) is seeded from one of these 200 initial messages, so the win-rate and SES estimates reported in Sections 6–7 are matched-conversation comparisons over the same conversational substrate.

C.5 Extension to 80 Initial Messages per Topic (Intervention Experiments)

The mitigation and intervention experiments require a substantially larger pool of seed messages so that train, validation, and test splits per topic can support DPO and SFT training in addition to evaluation. We extend the seed set to 80 initial messages per topic in both domains, drawing from larger upstream pools and guaranteeing zero overlap with the 200-message identification set introduced in Section C.4.

Task domain. The task-domain extension is drawn from the GPT-5-mini classification pool described above (LMSYS-Chat-1M first messages already assigned to one of the LMSYS-paper's 20 verbatim cluster labels plus *Other*). We con-

struct a fixed many-to-one mapping from the 20 paper-verbatim labels onto our 10 project task topics: the two Python-coding clusters and the AI-tools/software-design cluster fold together under *AI tools/programming*; the translation, summarization, and factual-consistency clusters fold together under *Text processing*; the two role-playing clusters fold together under *Role-playing scenarios*; the remaining seven project topics map one-to-one from a single paper cluster. NSFW, toxic-behavior, SQL, and *Other* clusters are excluded by design. From this pool we sample 80 messages per project topic deterministically (fixed random seed, English-only via a standard language-detection filter, with a 10–2,000 character window), excluding any message whose normalized form already appears in the identification seed set.

Daily domain. The daily-domain extension uses the GPT-4o-mini DailyDialog re-annotation described above. After deduplicating by first utterance, applying the same English and length filters, and classifying each remaining dialogue against the 11-way DailyDialog taxonomy, we sample 80 messages per project topic using the same dedup-against-seed protocol as the task domain.

Overlap and uniqueness guarantees. For both domains the extension is constructed to satisfy two invariants: (i) zero overlap with the 200-message identification seed set from Section C.4, and (ii) no within-extension cross-topic collisions — a message that has already been assigned to one project topic cannot reappear under any other project topic. The first invariant is enforced by accumulating a normalized-message dedup set from the identification seed before sampling; the second is enforced by sharing that dedup set across topics during the per-topic sampling pass. The resulting extension is therefore additively combinable with the identification seed at downstream consumption time without further deduplication.

D Contrastive Prompt Pair Example for Steering Vector Extraction

Contrastive Prompt Pair Example: *Expert*
Following the Contrastive Activation Addition (CAA) methodology (Panickssery et al., 2024), the extraction process builds, for every (topic, training first-message) pair in the per-(model, domain) training split, a *symmetric* pair of prompts that are byte-identical except for the style label. Only the style identifier changes between the positive and baseline prompt, so any activation difference at the same token position is attributable to the style instruction itself.
Positive Prompt (styled, label = feature): *"Topic: quantum computing \n Required Style: EXPERT \n Response:"*
Baseline Prompt (neutral, label = DEFAULT): *"Topic: quantum computing \n Required Style: DEFAULT \n Response:"*
The model is then run through a forward pass on each prompt and the hidden state at the final token position (the `:` after `Response`) is captured at each of the candidate layers (Section J). For feature f at layer ℓ, the per-pair direction is

$$d_i^{(\ell)} = h_{\text{pos},i}^{(\ell)} - h_{\text{baseline},i}^{(\ell)},$$

and the steering vector is the mean across the 480 training generations for each model:

$$v_f^{(\ell)} = \frac{1}{480} \sum_{i=1}^{480} d_i^{(\ell)}.$$

The training first-message itself is *not* appended to either prompt — it is used only as the (topic) grouping key when sampling pairs. The two prompts therefore differ only in the style label (*EXPERT* vs. *DEFAULT*); they are not formatted with the model's chat template (no `<|system|>` / `<|user|>` markers), so the contrast is measured in a clean completion-style context.

E Data Collection Pipeline

Figure 3 visualises the PRISMA-based pipeline used to collect the survey of conversational-agent papers and to extract the candidate style features described in Section 3.

F Extracted Feature Distribution

In this section, we present the extract feature distribution, which shows how frequently each unique style feature appears in each paper in the survey results.

G Response-Length Distributions and the Length-Control Effect

Figure 5 visualises, per (model, domain) setting of the side effect identification phase, the response-length distributions under three regimes: (i) the per-message length budget used to set

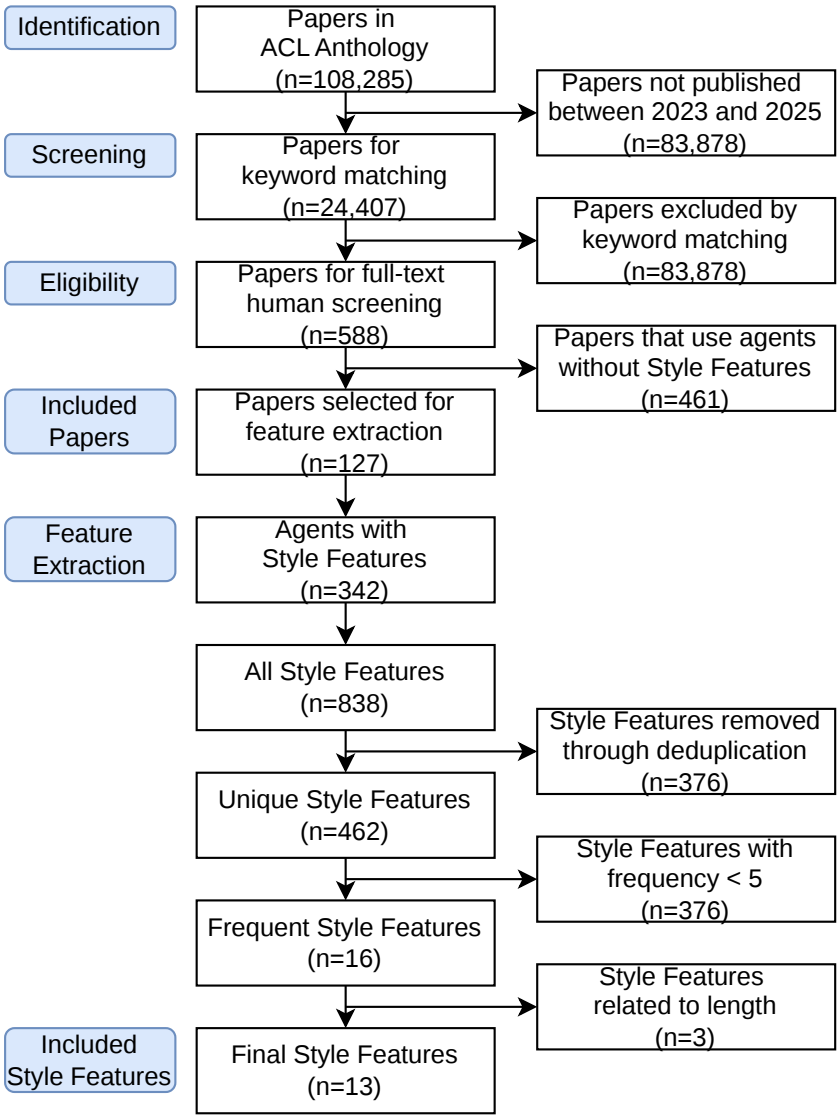

Figure 3: Data collection and screening pipeline following the PRISMA framework for the survey of conversational-agent papers and the selection of style-related features.

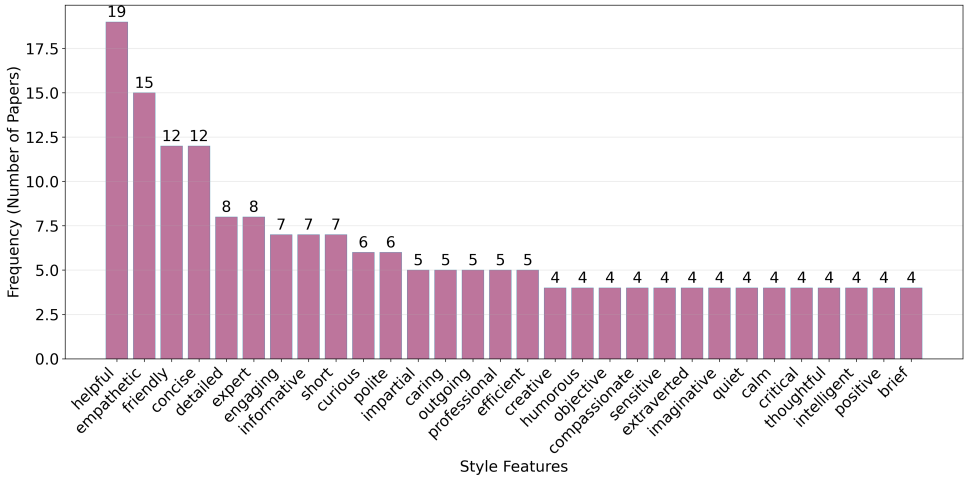

Figure 4: Extracted Feature Distribution. This figure shows the frequency of each unique style feature in the survey results, highlighting the most common style features such as helpful and empathetic.

the cap (gray; rounded mean of unconstrained reference generations per initial message; the raw reference outputs were not retained for $7/8$ models, so the budget integer is the closest available stand-in for the no-length-control distribution); (ii) the length-controlled *Neutral* response (blue; `original_word_count` from `rated_unified.csv`, one per initial message — the same response the judge saw on the Neutral side); and (iii) the length-controlled *Styled* responses, pooled across all 13 features (red; `styled_word_count` from `rated_unified.csv`, five per (first-message, feature) — the same responses the judge saw on the Styled side). For `GPT-OSS-20B` these word counts come from responses that were already harmony-stripped at rating time, so they reflect only the user-visible answer that contributed to the win-rate measurements.

Across all 16 (model, domain) settings the three distributions overlap closely, confirming that the per-message budget effectively pins both the *Neutral* baseline and the *Styled* responses to the same length region. This is the prerequisite for the win-rate measurements in Figure 1 and Appendix Figure 6 to reflect stylistic rather than length-driven differences.

H Full Win Rate Matrix For Each Domain and Each Model

In this section, we present two figures, one that shows the win rates across all models and domains and the other that shows the win rates averaged across three models for each domain. Both figures provide more details on how using a style feature affects a model response.

I Style Feature Frequency Across Related Papers

In this section, we present Figure 3, which shows the papers where any of our 16 filtered style features appears in.

J Steering Algorithm

In this section, we present the detailed steering intervention algorithm in three phases (Algorithms 1, 2, and 3), covering steering vector extraction, layer selection with checkpoint baking, and inference-time generation.

J.1 Phase 1: Steering Vector Extraction

Algorithm 1 constructs symmetric contrastive prompt pairs and computes the mean activation difference at a fixed token position. The prompt template (Appendix D) is byte-identical across the positive and baseline branches except for the style label, so any activation difference at the same token position is attributable to the style instruction.

J.2 Phase 2: Layer Selection and Model Baking

Once raw vectors are computed, we identify the best injection layer using the validation split and permanently bake the chosen vector into the model weights (Algorithm 2).

J.3 Phase 3: Test Set Intervention

In Algorithm 3 we load the baked checkpoint corresponding to the targeted *Side Feature* f_s and gen-

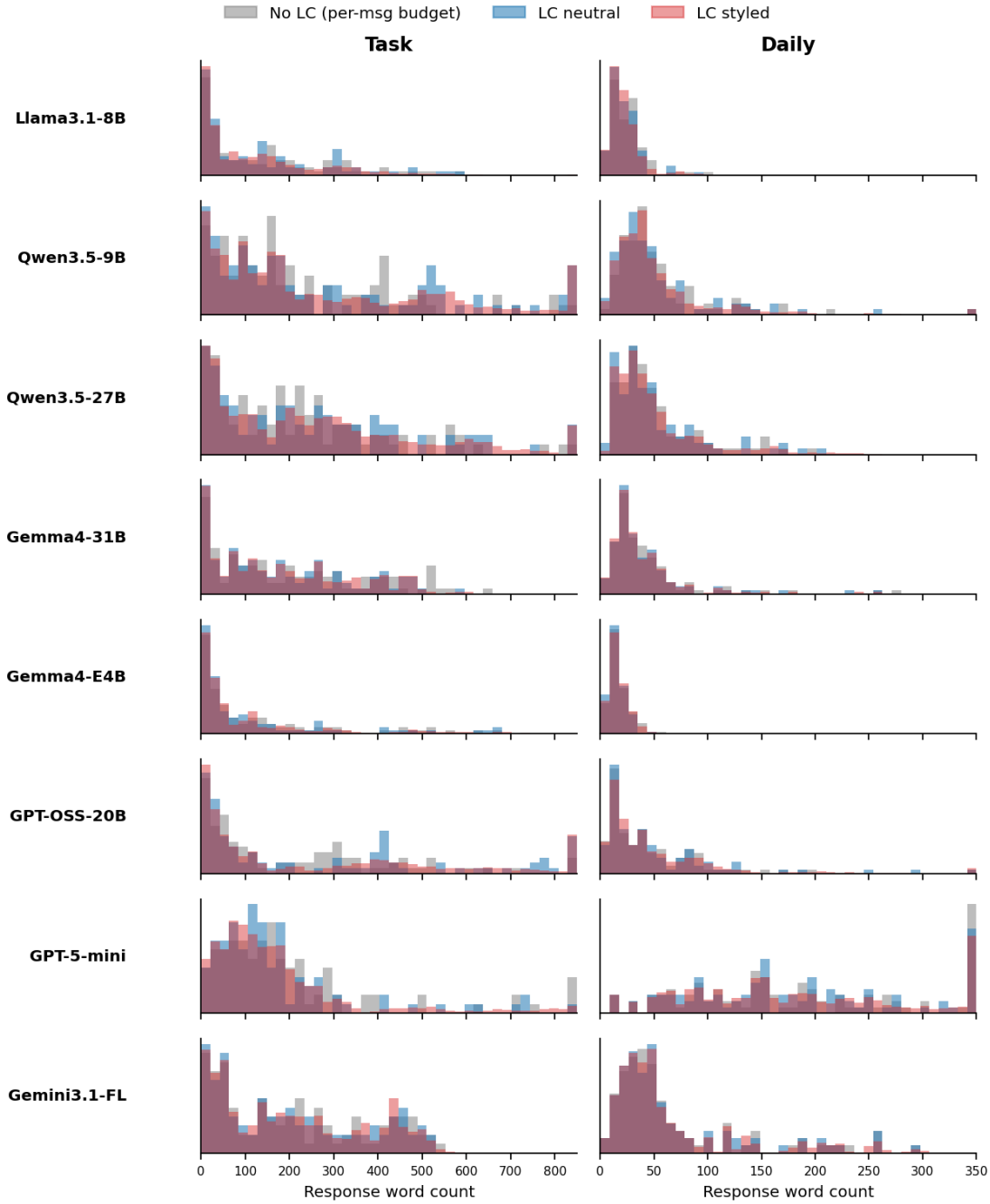

Figure 5: Response-length distributions on the side effect identification initial messages for each of the 8 generation models (rows) in each domain (columns: *Task* left, *Daily* right). Each panel overlays three histograms: **gray** – per-message length budget (rounded mean of unconstrained reference generations, n=100 per panel, used as the no-length-control stand-in); **blue** – length-controlled *Neutral* baseline (n=100 per panel, `original_word_count` from `rated_unified.csv`); **red** – length-controlled *Styled* responses pooled over all 13 features (n=6,500 per panel, `styled_word_count` from `rated_unified.csv`). All histograms are area-normalised; values above the column's 98th percentile of the styled distribution are clipped into the rightmost bin so the body of each distribution is visible.

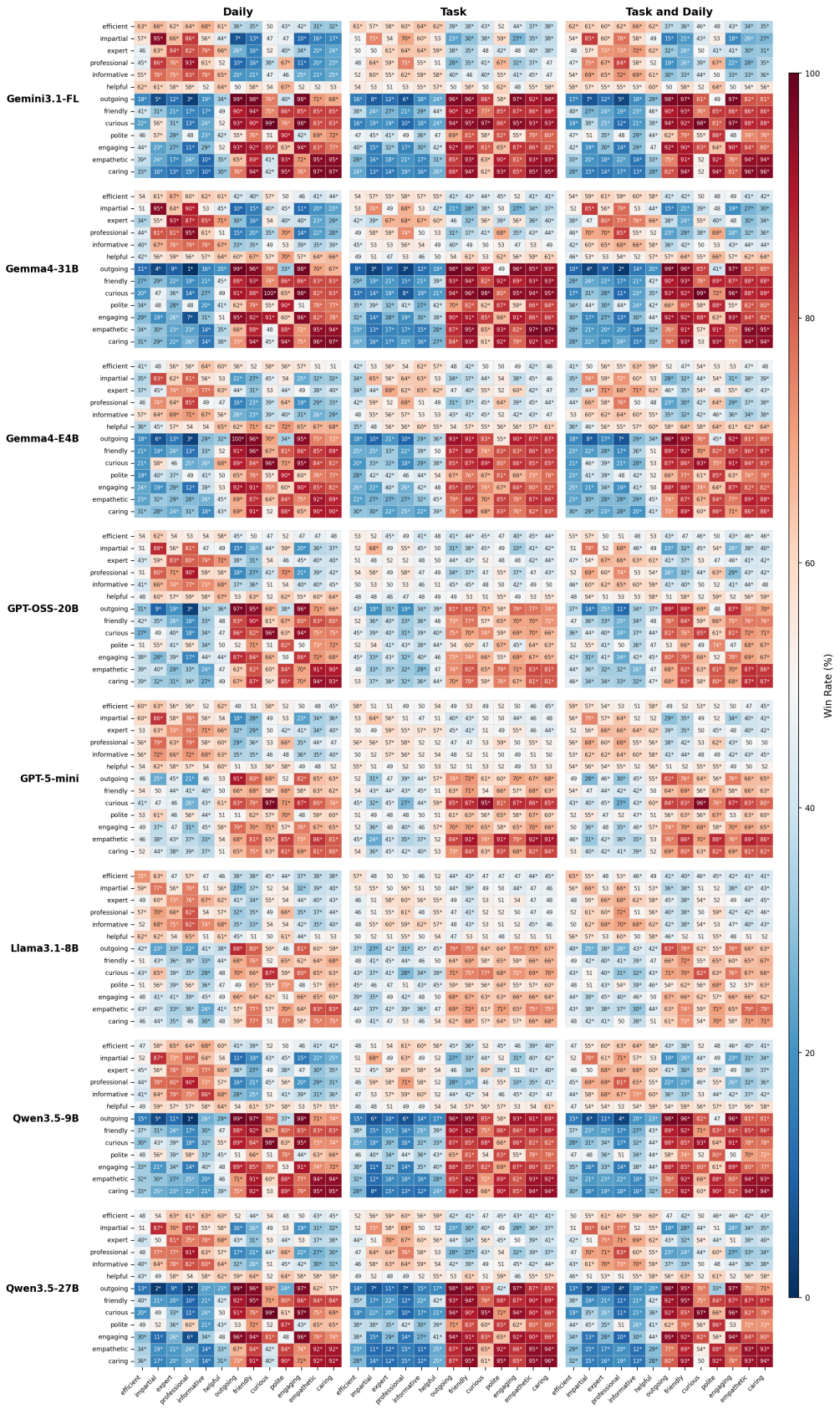

Figure 6: Win-rate effect matrices for all 8 generation models across the *Daily*, *Task*, and *Task and Daily* (both domains pooled) splits. Each subplot is a 13×13 heatmap: rows are prompted Main Features, columns are evaluated Side Features. Each entry reports the integer win rate (%) of the styled response over the neutral baseline; * marks $p \leq 0.05$ (binomial). Red $> 50\%$ / blue $< 50\%$. Shared colorbar on the right.

Style Feature	Clustered Feature	Freq.	Related Papers
helpful	helpful	19	Jiang et al. (2023), Köpf et al. (2023), Kwan et al. (2024), Lee et al. (2025b), Li et al. (2024b), Li et al. (2023), Liu et al. (2024), Meade et al. (2023), Mishra et al. (2023), Rana et al. (2025), Saha and Srihari (2024), Stureborg et al. (2024), Wang et al. (2023a), Xu et al. (2024a), Xu et al. (2024b), Yang and Ettinger (2023), Zhang et al. (2024a), Zhao et al. (2025a), Zhou et al. (2024)
empathetic	empathetic	15	ALMutairi et al. (2024), Cheng et al. (2025), Feng et al. (2025), Finch et al. (2023), Siyan et al. (2024b), Mishra et al. (2023), Njifenjou et al. (2025), Nozue et al. (2024), Reguera-Gómez et al. (2025), Svikhnushina and Pu (2023), Wang et al. (2023a), Wang et al. (2025), Zhang et al. (2024a), Zhang et al. (2024b), Zhou et al. (2024)
concise	concise	12	Afzal et al. (2024), Agrawal et al. (2024), Chalamalasetti et al. (2023), Feng et al. (2024), Li et al. (2024a), Movva et al. (2024), Siyan et al. (2024a), Rebedea et al. (2024), Taranukhin et al. (2024), Wang et al. (2023b), Wang et al. (2024a), Wang et al. (2025)
friendly	friendly	12	Dou et al. (2023), Lechner et al. (2023), Lee et al. (2023), Liu et al. (2025), Njifenjou et al. (2025), Reddy et al. (2023), Reguera-Gómez et al. (2025), Semnani et al. (2023), Seo and Lee (2024), Tsubota and Kano (2024), Wang et al. (2023a), Zhang et al. (2024a)
detailed	detailed	8	Deng et al. (2024), Jang et al. (2024), Kirstein et al. (2024), Li et al. (2024b), Liu et al. (2024), Sobhani et al. (2025), Rebedea et al. (2024), Zhao et al. (2025a)
expert	expert	8	Ferron et al. (2023), Lee et al. (2025a), Ou et al. (2024), Rachidi et al. (2025), Rana et al. (2025), Saley et al. (2024), Wevelsiep et al. (2025), Yang et al. (2024)
engaging	engaging	7	ALMutairi et al. (2024), Finch and Choi (2025), Jiang et al. (2024), Lee et al. (2023), Phukan et al. (2024), Wang et al. (2023b), Yu et al. (2024)
informative	informative	7	ALMutairi et al. (2024), Jiang et al. (2024), Raposo et al. (2023), Siro et al. (2024), Rebedea et al. (2024), Wang et al. (2023b), Yu et al. (2024)
short	short	7	Brown et al. (2024), Jandaghi et al. (2024), Li et al. (2024a), Reddy et al. (2023), Rebedea et al. (2024), Sun et al. (2024), Väth et al. (2024)
curious	curious	6	Lee et al. (2025b), Murakhovs'ka et al. (2023), Njifenjou et al. (2025), Petrak et al. (2024), Wang et al. (2023a), Wang et al. (2023b)
polite	polite	6	Abdullin et al. (2023), Afzal et al. (2024), Lee et al. (2023), Li et al. (2024b), Liu et al. (2024), Petrak et al. (2024)
caring	caring	5	ALMutairi et al. (2024), Fei et al. (2024), Feng et al. (2025), Wang et al. (2023a), Zhu et al. (2025)
efficient	efficient	5	Lee et al. (2025b), Liu et al. (2025), Wang et al. (2024b), Wang et al. (2023b), Zhu et al. (2025)
impartial	impartial	5	Duan et al. (2025), Madani et al. (2025), Meade et al. (2023), Raju et al. (2024), Zhao et al. (2025a)
outgoing	outgoing	5	He et al. (2025a), Lee et al. (2025b), Liu et al. (2025), Wang et al. (2023a), Wang et al. (2023b)
professional	professional	5	ALMutairi et al. (2024), Kirstein et al. (2024), Oshima et al. (2024), Rachidi et al. (2025), Saley et al. (2024)

Table 3: Overview of the 16 identified style features appearing in at least 5 papers, with references of all papers which contain those features. All terms have been adjectivized and deduplicated for consistency.

Algorithm 1 Steering Vector Extraction

Input: Training set $\mathcal{D}_{\text{train}}$ of (t_i, q_i) topic / first-message pairs, target model $\mathcal{M}$, candidate layers $\mathcal{L}$, style features $\mathcal{F}$.

Output: Raw steering vectors $\{\mathbf{v}_f^{(\ell)}\}$ for all $f \in \mathcal{F}, \ell \in \mathcal{L}$.
Step 1: Symmetric Contrastive Prompt Construction
for each feature $f \in \mathcal{F}$ **do**
  $\mathcal{C}_f \leftarrow \emptyset$
  for each $(t_i, q_i) \in \mathcal{D}_{\text{train}}$ **do**
    $p_i^+ \leftarrow$ "Topic: t_i \n Required Style: UPPER(f) \n Response:";
    $p_i^- \leftarrow$ "Topic: t_i \n Required Style: DEFAULT \n Response:";
    $\mathcal{C}_f \leftarrow \mathcal{C}_f \cup \{(p_i^+, p_i^-)\}$;
  end for
end for
Step 2: Activation Capture and Vector Computation
for each feature $f \in \mathcal{F}$ **do**
  for each $(p_i^+, p_i^-) \in \mathcal{C}_f$ **do**
    for each layer $\ell \in \mathcal{L}$ **do**
      $\mathbf{h}_+^{(\ell)} \leftarrow \mathcal{M}$.get_activation$(\ell, p_i^+, \text{pos} = -1)$;
      $\mathbf{h}_-^{(\ell)} \leftarrow \mathcal{M}$.get_activation$(\ell, p_i^-, \text{pos} = -1)$;
    end for
  end for
  for each layer $\ell \in \mathcal{L}$ **do**
    $\mathbf{v}_f^{(\ell)} \leftarrow \frac{1}{|\mathcal{C}_f|} \sum_i (\mathbf{h}_+^{(\ell)}[i] - \mathbf{h}_-^{(\ell)}[i])$;
  end for
end for

Algorithm 2 Validation and Checkpoint Baking

Input: Raw vectors $\{\mathbf{v}_f^{(\ell)}\}$, validation set $\mathcal{D}_{\text{val}}$, target model $\mathcal{M}$, judge model $\mathcal{M}_{\text{judge}}$.
Output: Best layers $\{\ell_f^*\}$, baked checkpoints $\{\mathcal{M}_f^{\text{steered}}\}$.
Step 1: Best Layer Selection
for each feature $f \in \mathcal{F}$ **do**
  for each layer $\ell \in \mathcal{L}$ **do**
    $\mathcal{M}_f^{(\ell)} \leftarrow \mathcal{M}$ with $\mathbf{v}_f^{(\ell)}$ added to layer ℓ activations at the last token position during the forward pass;
    for each $(t_i, q_i, y_i^{\text{neutral}}) \in \mathcal{D}_{\text{val}}$ **do**
      $y_i^{\text{steered}} \leftarrow \mathcal{M}_f^{(\ell)}$.generate$(q_i)$;
      $w_i \leftarrow \mathcal{M}_{\text{judge}}$.compare$(f, y_i^{\text{steered}}, y_i^{\text{neutral}})$;
    end for
    $W_f^{(\ell)} \leftarrow \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_i \mathbf{1}[w_i = \text{steered}]$;
  end for
  $\ell_f^* \leftarrow \arg\max_{\ell \in \mathcal{L}} W_f^{(\ell)}$;
end for
Step 2: Checkpoint Baking
for each feature $f \in \mathcal{F}$ **do**
  $\theta \leftarrow$ load_checkpoint$(\mathcal{M})$;
  θ[`layers.`ℓ_f^*`.mlp.down_proj.bias`] $\leftarrow \mathbf{v}_f^{(\ell_f^*)}$;
    // `down_proj` has no bias by default; setting it to $\mathbf{v}_f^{(\ell_f^*)}$ is
    // equivalent to adding the vector at the layer's output during the forward pass.
  Set `mlp_bias=True` in the model config; save as $\mathcal{M}_f^{\text{steered}}$;
end for

Algorithm 3 Steering on Test Set

Input: Baked checkpoints $\{\mathcal{M}_f^{\text{steered}}\}$, test set $\mathcal{D}_{\text{test}}$, per-(model, domain) targeted pair $(f_m, f_s, \text{polarity})$.
Output: Steered generations $\{y_i^{(\text{steer},r)}\}$.
$\mathcal{M}^{\text{steered}} \leftarrow \mathcal{M}_{f_s}^{\text{steered}}$ (with polarity sign applied to $\mathbf{v}_{f_s}$ before baking);
for each $(t_i, q_i) \in \mathcal{D}_{\text{test}}$ **do**
  for repetition $r = 1$ to 5 **do**
    $s_i \leftarrow$ "You are having a conversation about t_i. Please be f_m in your responses. Please answer in around N words. Respond directly without showing your reasoning.";
    $y_i^{(\text{steer},r)} \leftarrow \mathcal{M}^{\text{steered}}$.generate$(s_i, q_i)$;
      // Prompt requests f_m; the activation shift toward / away from f_s
      // is baked into the `down_proj` bias of layer $\ell_{f_s}^*$.
  end for
end for

erate test responses while the prompt still asks for the *Main Feature* f_m. The side effect mitigation thus combines a styled-prompt signal (from f_m) with a baked-in activation shift toward f_s (or away from it, by polarity).

J.4 Notation and Definitions

Target models: the 6 open-weight models for which steering is applied — `Llama3.1-8B`, `Qwen3.5-9B`, `Qwen3.5-27B`, `Gemma4-31B`, `Gemma4-E4B`, and `GPT-OSS-20B`. The two API-served models (`Gemini3.1-FL`, `GPT-5-mini`) do not expose internal activations and are excluded from steering. The judge is `GPT-OSS-120B`.

Steering vector $\mathbf{v}_f^{(\ell)}$: a vector in $\mathbb{R}^{d_{\text{model}}}$ representing the direction in activation space that increases feature f at layer ℓ. Computed as the mean of 480 per-pair differences $\mathbf{h}_+^{(\ell)} - \mathbf{h}_-^{(\ell)}$ across the training split of the (model, domain) pairs.

Contrastive prompt pair: two byte-identical prompts that differ only in the style label, formatted as "`Topic: {topic}\n Required Style: {label}\n Response:`" where `label` is the uppercase feature name on the positive branch and `DEFAULT` on the baseline branch. No chat template is applied at extraction time; the prompts are raw completion-style scaffolds. See Appendix D for a worked example.

Activation extraction: activations are captured at the final token position (pos $= -1$, the `:` after `Response`) of each prompt at every candidate layer.

Candidate layers $\mathcal{L}$: auto-selected as the middle 60% of the model's transformer layers, evenly spaced into 7 candidates. For `Llama3.1-8B` (32 layers) this yields $\mathcal{L} = \{6, 9, 12, 15, 18, 21, 24\}$; the exact indices vary by architecture depth.

Best-layer selection: for each feature f, $\ell_f^* = \arg\max_{\ell \in \mathcal{L}} W_f^{(\ell)}$, where $W_f^{(\ell)}$ is the validation-set win rate of the steered model against the *neu-*

tral response on the side-feature axis. The styled ground-truth response is *not* used in selection.

Checkpoint baking: the steering vector $\mathbf{v}_f^{(\ell_f^*)}$ is written to the bias slot of the layer's MLP `down_proj` ($\theta[\texttt{layers.}\ell_f^*\texttt{.mlp.down_proj.bias}] \leftarrow \mathbf{v}$). Since the base model has no bias on this projection, setting the slot is equivalent to adding the vector to the layer's output at inference, with no runtime hook required.

Targeted pair $(f_m, f_s, \textbf{polarity})$: per-(model, domain) pair, the top-SES pair (Section 6). The polarity sign is applied to the side-feature vector before baking: positive polarity (main over-elevates side $\Rightarrow$ we want side down) uses $-\mathbf{v}_{f_s}$; negative polarity (main suppresses side $\Rightarrow$ we want side up) uses $+\mathbf{v}_{f_s}$.

K Task and Daily Dataset with Initial Message Examples

In this section, we present Table 4 which shows all topics used in this study with initial message examples.

L Prompting Algorithm: Generation

In this section, we present a detailed Algorithm 4 about using prompting intervention to mitigate side effects of using style features in prompts.

L.1 Generation Algorithm

We present our Prompting Algorithm 4 here with definitions in the following subsection.

Algorithm 4 Prompting Generation

Input: Test set $\mathcal{D}_{\text{test}} = \{(q_i, t_i, d_i)\}$ where q_i is the query, t_i is the topic, and d_i is the domain. Targeted Pairs $\mathcal{P} = \{(f_m, f_s, d_p, p)\}$. Generator Model $\mathcal{M}_{\text{gen}}$.
Output: Set of generated responses $\mathcal{Y}_{\text{generated}}$
Phase 1: Prompting Generation
for each pair $(f_m, f_s, d_p, p) \in \mathcal{P}$ **do**
  Define $\pi_{\text{normal/reversed}}$ based on polarity p (e.g., "Please be f_m but f_s");
  Filter dataset $\mathcal{D}_p \leftarrow \{(q, t, d) \in \mathcal{D}_{\text{test}} : d \in d_p\}$;
  for each message $(q_i, t_i, d_i) \in \mathcal{D}_p$ **do**
    for $r = 1$ to 5 **do**
      Construct prompt $s_i \leftarrow$ "You are a helpful assistant having a conversation about t_i. π.";
      Generate response $y_i^{(\pi,r)} \leftarrow \mathcal{M}_{\text{gen}}.\text{generate}(s_i, q_i)$;
      Add $y_i^{(\pi,r)}$ to $\mathcal{Y}_{\text{generated}}$;
    end for
  end for
end for

Algorithm 5 Pairwise Comparison and Win Rate Calculation

Input: Generated responses $\mathcal{Y}_{\text{generated}}$. Evaluation Features $\mathcal{F}$. Judge Model $\mathcal{M}_{\text{judge}}$.
Output: Win rate matrices $\mathbf{W}^{(f_m, f_s)}$
Phase 1: Pairwise Comparison Rating
for each generated response $y_i^{(\pi,r)} \in \mathcal{Y}_{\text{generated}}$ **do**
  Load baselines: y_i^{neutral} (no style) and $y_i^{(f)}$ (styled);
  for each evaluation feature $f_{\text{eval}} \in \mathcal{F}$ **do**
    $c_{\text{orig}} \leftarrow \text{ComparisonPrompt}(f_{\text{eval}}, y_i^{(\pi,r)}, y_i^{\text{neutral}}, q_i)$;
    $w_{\text{orig}}^{(f_{\text{eval}})} \leftarrow \mathcal{M}_{\text{judge}}.\text{compare}(c_{\text{orig}})$; *// Rate vs Neutral*
    $c_{\text{styled}} \leftarrow \text{ComparisonPrompt}(f_{\text{eval}}, y_i^{(\pi,r)}, y_i^{(f_{\text{eval}})}, q_i)$;
    $w_{\text{styled}}^{(f_{\text{eval}})} \leftarrow \mathcal{M}_{\text{judge}}.\text{compare}(c_{\text{styled}})$; *// Rate vs Styled*
  end for
end for
Phase 2: Win Rate Calculation
for each pair $(f_m, f_s, d_p, p) \in \mathcal{P}$ **do**
  Init matrix $\mathbf{W}^{(f_m, f_s)} \in \mathbb{R}^{4 \times (|\mathcal{F}|+1)}$;
  for each $f_{\text{eval}} \in \mathcal{F} \cup \{\text{length}\}$ **do**
    $\mathbf{W}_{1,f_{\text{eval}}} \leftarrow n(\text{Intervention} > \text{Neutral})/n_{\text{total}}$;
    $\mathbf{W}_{2,f_{\text{eval}}} \leftarrow \text{StyledWinRate}(f_m, f_{\text{eval}})$; *// Pre-computed*
    $\mathbf{W}_{3,f_{\text{eval}}} \leftarrow \text{StyledWinRate}(f_s, f_{\text{eval}})$; *// Pre-computed*
    $\mathbf{W}_{4,f_{\text{eval}}} \leftarrow \text{WinRate}(\pi_{\text{reversed}} > \text{Neutral})$;
  end for
  Apply BinomialTest (significance $p \leq 0.05$) and Save Heatmap;
end for

L.2 Generation Notation

Models: $\mathcal{M}_{\text{gen}}$ (`Llama-3.1-8B-Instruct` or `qwen3-8b`).

Intervention Prompts (π): The specific instruction strings appended to the system prompt to steer the model. For a pair (f_m, f_s) with negative polarity, we define two variations to test ordering effects:

- π_{normal}: *"Please be f_m but f_s."* (Prioritizes Main Feature).
- π_{reversed}: *"Please be f_s but f_m."* (Prioritizes Side Feature).

M Win Rate Calculation Algorithm

In this section, we present a Win Rate Algorithm 5, which shows the how we calculate win rates to decide the Side Feature strengths of using Main Feature in prompts.

M.1 Evaluation Algorithm

We present our win rate algorithm 5 used for evaluation here.

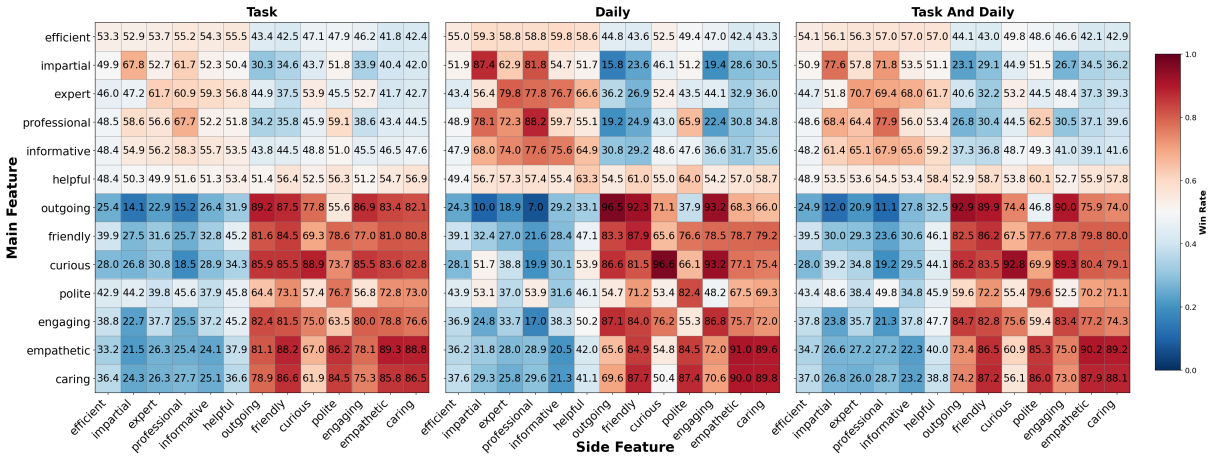

Figure 7: Win-rate matrices averaged across all 8 generation models, shown for the *Task*, *Daily*, and *Task and Daily* (both domains pooled) splits. Each subplot is a 13×13 heatmap: rows are prompted Main Features, columns are evaluated Side Features. Each entry reports the mean win rate (%) of the styled response over the neutral baseline, averaged element-wise across the 8 per-model matrices in Figure 6. Red $> 50\%$ / blue $< 50\%$.

Domain	Topic	Initial Message Examples
LMSYS-Chat-1M (Task)	Answering questions based on passages	Who was the first man on the moon?
	Discussing software errors and solutions	Describe and explain the benefits of OpenBSD
	Inquiries about specific plant growth conditions	how do I get rid of mosquitos?
	Requesting introductions for various chemical companies	Explain "reductive amination".
	Inquiries about AI tools, software design, and programming	write python function to reverse a sentence
	Text processing (Merged)	sup peeps. Wanna help me with summarizing lyrics?
	Role-playing scenarios and character interactions (Merged)	Write me an extremely funny tale about pillow hoarding ducks at a luxurious hotel.
	Geography, travel, and global cultural inquiries	explain why we have traffic lights?
	Discussing and describing various characters	Tell me something about Stephen King's Dark Tower.
	Creating and improving business strategies and products	how to make a game project success
DailyDialog (Daily)	Tourism	That is the most beautiful sunset !
	Work	Hey , Zina . You're here early today .
	Politics	Every country should face the history .
	Finance	It's all over . I'm bankrupt .
	Health	Are you feeling better today , Bill ?
	School Life	What can I help you with today ?
	Attitude & Emotion	Do you hear what happened to Sally ?
	Ordinary Life	Excuse me . Is this seat taken ?
	Culture & Education	Harry , do you like the opera ?
	Relationship	I wonder how Sarah and Mat are .

Table 4: Example Initial Messages by Domain and Topic (10 Topics Per Domain). We curate and merge some task topics, specificaly the *Text processing* and *role-play scenarios and character interactions*. The Daily topics are extracted from the DailyDialog dataset without modifications.

M.2 Evaluation Definitions

Comparison Prompt: The judge is asked: *"Which response is more f? Answer with ONLY A or B."* Order is randomized to mitigate position bias.

Win Rate: Calculated as $\frac{|\{i:w_i=\text{"intervened"}\}|}{|\{i:w_i\neq\text{"unknown"}\}|}$. Statistical significance is assessed using a two-sided binomial test ($p_0 = 0.5$).

N Insights from Style Feature Extraction

Out of the style features, *helpful*, *empathetic*, and *friendly* are most frequently used. Right skewed distribution shows style feature usages are concentrated into a few terms, with a significant long tail where 75% of the 462 unique features appear only once in our list of papers. We further analyze the number of features employed per agent. Among the 342 agents studied, the median number of features is 1, with 177 agents (52%) utilizing only a single feature. This finding suggests that contemporary research and applications predominantly rely on isolated keywords rather than more sophisticated style feature usage.

The systematic extraction of style features reveals a distinct bifurcation in how "style" is conceptualized in current research: it is predominantly treated either as a *functional constraint* (e.g., *concise*, *detailed*) or a *psychological simulation* (e.g., *empathetic*, *extraverted*). The data suggests that while the capability for complex persona engineering exists, the majority of research defaults to minimal, service-oriented prompt structures.

N.1 The Dominance of Service-Oriented and Utility Traits

As illustrated in Figure 4, the frequency distribution indicates that the primary goal of contemporary conversational agents remains functional utility and emotional safety rather than complex character enactment.

Benevolence and Safety: The two most frequently extracted features are *helpful* ($N = 19$) and *empathetic* ($N = 15$). This alignment correlates with the prevalence of papers focused on general-purpose assistants and therapeutic dialogue systems, where prompts are engineered to ensure the agent provides appropriate emotional value and adheres to safety guidelines.

N.2 Standardization vs. The "Long Tail" of Definition

The distribution data reveals a lack of standardized terminology for defining agent personality, characterized by a heavy reliance on ad-hoc descriptors.

The "Expert" Persona: Features such as *expert* ($N = 8$) and *detailed* ($N = 8$) appear in the mid-range of frequency. These are predominantly used in domain-specific research (e.g., medical or legal synthesis) to act as validity markers, ensuring the agent sounds like a credible authority rather than a casual chatbot.

Fragmentation of Terminology: The visualization displays a long tail of psychological and tonal descriptors that plateau at low frequencies ($N = 4$). Traits such as *creative*, *humorous*, *extraverted*, and *thoughtful* appear significantly less often than utility prompts. This suggests that researchers often employ colloquial adjectives to steer models for specific narrow tasks rather than adhering to established psychological frameworks (such as the Big Five) for consistent persona definition.

Redundancy in Prompting: The data also exposes terminological redundancy. For example, while *concise* is a dominant feature ($N = 12$), synonymous constraints like *short* ($N = 7$) and *brief* ($N = 4$) appear separately. This fragmentation indicates that "style" is often defined by the individual researcher's vocabulary preference rather than a standardized taxonomy of prompt engineering.

N.3 Experimental Scope: Minimalist vs. Maximalist Designs

Beyond the frequency of specific terms, we analyzed the *diversity* of style features employed within individual papers to understand the depth of experimental design. Figure 8 presents the distribution of papers based on the count of unique style features they utilized, revealing a strongly right-skewed trend.

Prevalence of Minimalist Designs: The largest cohort of papers ($N = 40$) utilizes only a single unique style feature throughout the entire study. Combined with those using two or three features, the majority of research ($N = 86$) uses a very limited number of features. This confirms that for most contemporary work, style is treated as a static control variable—likely a global instruction

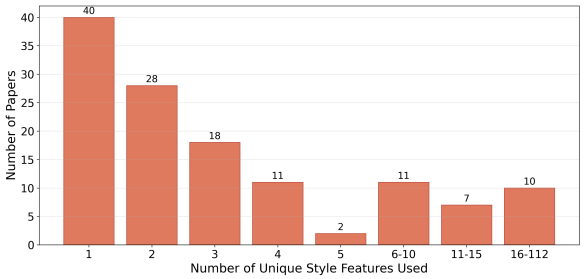

Figure 8: Distribution of papers categorized by the number of unique style features used. The distribution highlights a polarization between minimalist designs (single-feature) and broad-spectrum evaluations (16+ features).

to be "helpful" or "safe"—rather than a dynamic parameter for experimentation.

The "Benchmarking" Tail: In sharp contrast to the minimalist majority, a distinct subset of 10 papers employs a massive range of features (16 to 112 unique terms). This heavy tail represents a fundamental divergence in research intent: these outlier studies treat style as the primary independent variable. These likely correspond to large-scale persona benchmarks or synthetic dataset generation efforts that require maximizing behavioral diversity.

O Hierarchical Clusterings of Style Features

In this section, we present Figure 9 which shows how we decide which style features out of the filtered 16 style features have similar meanings

P Training Setup

This appendix describes the training procedure used for the DPO and SFT interventions in Section 7. All training was run on the intervention split of 4,800 first-message pairs per model in each domain (480 training first-messages $\times$ 10 repetitions; the matching 160 val and 160 test first-messages are held out for the figures and tables in the main paper).

P.1 Frameworks

We use two open-source training stacks, selected per model:

- **LLaMA-Factory** (with DeepSpeed ZeRO-3) for the full-model fine-tuning of `Llama-3.1-8B` and `Qwen3.5-9B` on 8$\times$A6000. We patched only `dataset_info.json` to register the JSON files for each model in each domain.
- **Unsloth** on a single B200 for the larger / specially-quantized models: `Qwen3.5-27B`, `Gemma-4-31B`, and `gpt-oss-20b` (QLoRA-4bit), and `Gemma-4-E4B` (bf16 full-FT). We load each model from Unsloth's pre-processed checkpoints (`unsloth/gemma-4-31B-it`, `unsloth/gemma-4-E4B-it`, `unsloth/Qwen3.5-27B-Instruct-bnb-4bit`, and `unsloth/gpt-oss-20b`) and apply the `gemma-4` chat template via `get_chat_template` for the Gemma-4 models and the model's native template for the rest. LoRA adapters (rank 16, α=32) are attached to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`; for `gpt-oss-20b` we list these explicitly rather than relying on the `"all-linear"` shorthand so the MoE expert projections are included.

P.2 Training Data Construction

For each model and domain we construct a single JSON dataset:

- **DPO**: `chosen` = intervened response from Phase 5 (*Please be {main} but not {side}* for suppress_positive cases, *Please be {main} but {side}* for enhance_negative cases); `rejected` = styled-only response from Phase 4 (*Please be {main}*). Same `first_message`, same model, same per-message length budget N (Section 4). 4,800 preference pairs per model and domain.
- **SFT**: single-target supervised pairs; target = intervened response from Phase 5. 4,800 examples per model and domain.

The styled system prompt is embedded inside every training example as a `system` message so that Phase-9 inference (which queries the trained checkpoint with the same styled prompt) sees the same context distribution the model was trained on. Without this we observed near-zero transfer: the trained checkpoint would silently fall back to the model's default system template at inference time and produce the original styled response rather than the intervened one.

P.3 Hyperparameters

Shared across every reported models: **1 epoch**, AdamW with a cosine schedule and warmup ratio

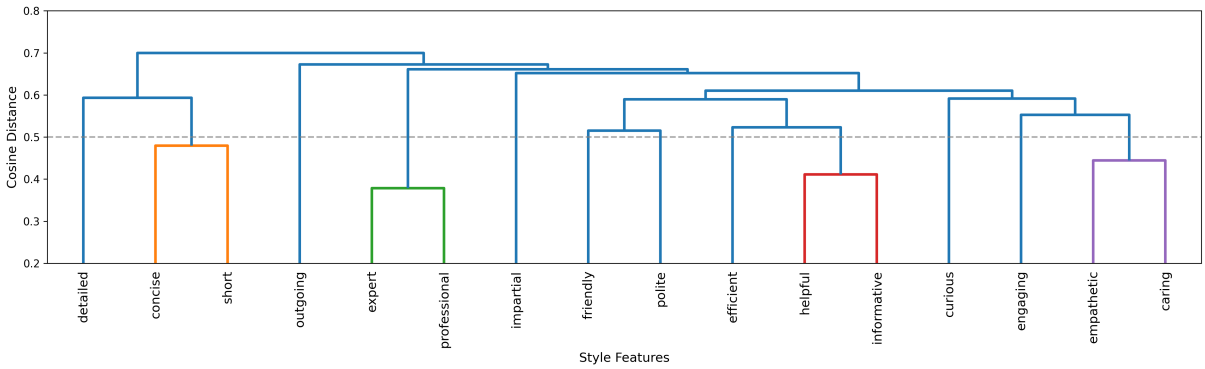

Figure 9: Hierarchical clustering of style features based on semantic similarity. Features are embedded using OpenAI's `text-embedding-3-small` model and clustered via average linkage on cosine distances. Semantically related features cluster together, like (concise, short), (expert, professional), (helpful, informative), and (empathetic, caring).

Model	Stack	FT class	SFT lr	DPO lr
Llama-3.1-8B	LLaMA-Factory	full-FT	2×10^{-5}	5×10^{-6}
Qwen3.5-9B	LLaMA-Factory	full-FT	2×10^{-5}	5×10^{-6}
Qwen3.5-27B	Unsloth	QLoRA-4bit	1×10^{-4}	5×10^{-6}
Gemma-4-31B	Unsloth	QLoRA-4bit	1×10^{-4}	5×10^{-6}
Gemma-4-E4B	Unsloth	full-FT†	2×10^{-6}	2×10^{-6}
gpt-oss-20b	Unsloth	QLoRA-4bit	1×10^{-4}	5×10^{-6}

Table 5: Per-model training configuration behind the reported numbers. All runs use 1 epoch, cosine schedule, warmup 0.1, bf16, and sequence length 4096. LLaMA-Factory configurations are full-model fine-tuned with DeepSpeed ZeRO-3 on 8×A6000; Unsloth configurations (LoRA rank 16, α=32 for QLoRA) run on a single B200. †Gemma-4-E4B is full-FT (with vision-tower freeze + fp32 mixed-precision) for 3 of its 4 configurations; the task/SFT configuration is unstable in full-FT (loss diverges to NaN regardless of LR, gradient clipping, or precision) and falls back to a LoRA adapter (SFT lr 1×10^{-4}). gpt-oss-20b is natively MXFP4 and uses explicit attention+MoE LoRA targets.

0.1, bf16 mixed precision, and (for DPO) $\beta = 0.1$ with the sigmoid loss. We pick epoch=1 because we are limited by compute constraint. The learning rate and fine-tuning class differ by model and method as listed in Table 5; the two stacks use different effective batch sizes (LLaMA-Factory: ≈64 across 8 GPUs; Unsloth: per-device batch 1× gradient accumulation 8 on a single B200). QLoRA use LoRA rank 16, α=32. We report the full-model checkpoints for `Llama-3.1-8B` and `Qwen3.5-9B` (directly comparable to each other); `Qwen3.5-27B`, `Gemma-4-31B`, and `gpt-oss-20b` are adapter/QLoRA-only because full-model training of these does not fit our hardware budget.

P.4 Compute

Each run finishes in 1–3 hours on 8× NVIDIA A6000 (48 GB, LLaMA-Factory full-FT) or a single NVIDIA B200 (180 GB, Unsloth). The full 24-run sweep (6 trainable models × 2 domains × 2 methods, excluding the 2 API-served models, which we run with prompt intervention only) consumed on the order of a few hundred GPU-hours across a mixed A6000 / B200 cluster.

P.5 Evaluation

After each training run, we run vLLM inference on the 160-first-message test split (5 repetitions per message, $temperature$=0.7 for local models) and rate each `(intervened, baseline)` pair with the `GPT-OSS-120B` judge via the same prompt as in Appendix A. The resulting win rates power Figure 2.

Q Side Effect Score (SES) — Detailed Algorithm

This appendix gives the full derivation of the Side Effect Score (SES) used in Section 6 to rank candidate (Main Feature, Side Feature) pairs per model per domain.

Q.1 Inputs

For each model M and domain $D \in \{\text{daily}, \text{task}\}$, SES is computed on the 13×13 grid indexed by an ordered pair (m, s), where m is the prompted Main Feature and s is the measured Side Feature. For each entry in a heatmap in Figura 6 we have:

- $W(m, s) \in [0, 1]$ — the pairwise win rate of the styled response (system prompt requesting m, length-controlled per Section 4) over the

neutral baseline, as judged by `GPT-OSS-120B` on the s-axis.

- $p_W(m, s)$ — the two-sided binomial-test p-value for whether $W(m, s)$ deviates significantly from 0.5.

We additionally precompute a fixed 13×13 semantic-similarity matrix:

- $C(m, s) = \cos(\mathbf{e}_m, \mathbf{e}_s)$ — cosine similarity of the OpenAI `text-embedding-3-large` (OpenAI, 2024b) embeddings of the feature names. C does not depend on M or D; it captures only the prior linguistic relatedness between two style adjectives.

Q.2 MAD-standardised effect sizes

We standardise both W and C using robust statistics so a single anomalous pair does not dominate the scale. For win rate, we centre at the no-preference value (0.5):

$$N_{\text{WR}}(m, s) = \frac{W(m, s) - 0.5}{1.4826 \cdot \text{MAD}\big(W_{i \neq j}\big)}.$$

For cosine similarity, we mean-centre because cosine values cluster well above 0:

$$Z_{\text{CS}}(m, s) = \frac{C(m, s) - \overline{C}}{1.4826 \cdot \text{MAD}\big(C_{i \neq j}\big)},$$

where $\overline{C}$ is the off-diagonal mean of C and MAD denotes the median absolute deviation taken over the off-diagonal entries of the 13×13 matrix. The factor 1.4826 converts MAD to a normal-equivalent standard deviation, so N_{WR} and Z_{CS} are comparable across rows and columns even though their underlying scales differ. The diagonal (m, m) ratings are excluded from the MAD and mean estimates because $W(m, m)$ is a self-comparison and is not informative about side effects.

Q.3 SES formula

The Side Effect Score combines the two standardised quantities and two indicator gates:

$$\begin{aligned}\text{SES}(m, s) \;&= \big|N_{\text{WR}}(m, s) - Z_{\text{CS}}(m, s)\big| \\ &\cdot \min\big(|N_{\text{WR}}(m, s)|,\ |Z_{\text{CS}}(m, s)|\big) \\ &\cdot \mathbf{1}\big[\,N_{\text{WR}}(m, s) \cdot Z_{\text{CS}}(m, s) < 0\,\big] \\ &\cdot \mathbf{1}\big[\,p_W(m, s) \leq 0.05\,\big].\end{aligned}$$

Each factor encodes a specific desideratum:

(i) $\big|N_{\text{WR}} - Z_{\text{CS}}\big|$ — *disagreement* between the empirical effect and the prior expectation. Large values mean the win rate departs from what semantic similarity would predict.

(ii) $\min(|N_{\text{WR}}|, |Z_{\text{CS}}|)$ — *magnitude floor*: we trust the score only when both standardised quantities are appreciably away from zero. This downweights cases where one of N, Z is near zero (and the disagreement term is large only because of the other one).

(iii) $\mathbf{1}[N_{\text{WR}} \cdot Z_{\text{CS}} < 0]$ — *polarity flip*: the score is non-zero only when the empirical effect is in the *opposite* direction from semantic similarity. This rules out (a) two distant features moving apart and (b) two related features moving together — both expected outcomes that are not side effects. We keep (c) two distant features that nevertheless move together and (d) two related features that move apart — both surprising and the kind of behaviour we want to flag.

(iv) $\mathbf{1}[p_W \leq 0.05]$ — *significance gate*: the win-rate effect must be statistically distinguishable from 0.5 at the two-sided binomial level.

Q.4 Polarity and intervention direction

The sign of $N_{\text{WR}}(m, s)$ determines how the side effect should be intervened against:

- $N_{\text{WR}} > 0$: prompting m *elevates* s relative to the neutral baseline — an unwanted *gain*. The mitigation goal is to *suppress* s while preserving m; we label these interventions as *suppress_positive*.
- $N_{\text{WR}} < 0$: prompting m *suppresses* s — an unwanted *loss*. The mitigation goal is to *enhance* s while preserving m; we label these interventions *enhance_negative*.

The polarity label feeds the prompt-intervention template choice (Section 7): *"Please be {m} but not {s}"* for *suppress_positive*, and *"Please be {m} but {s}"* for *enhance_negative*.

Q.5 Per-(model, domain) top-1 pair selection

For each model in each domain, we rank the $13 \times 12 = 156$ off-diagonal pairs by $\text{SES}(m, s)$ and pick the top-1 as the model's intervention target in that domain. We additionally apply a domain-appropriate filter on the Main Feature so the chosen pair matches the kind of styling a user would realistically request in that context:

- **Task domain**: the Main Feature m is restricted to utility-oriented features (*efficient, impartial, expert, professional, informative, helpful*). Task-oriented prompts naturally favour these.
- **Daily domain**: the Main Feature m is restricted to interpersonal features (*outgoing, friendly, curious, polite, engaging, empathetic, caring*). Casual/open-domain conversations naturally favour these.

This partition is the same utility / interpersonal split discussed in Section 6 when describing the heatmap clusters.

Q.6 Outputs and downstream use

For each model in each domain, the top-1 pair under SES becomes the intervention target for Section 7; the full set of 16 chosen pairs is summarised in Table 1 and the per-method mitigation win rates against those pairs are reported in Appendix T. The top-3 SES pairs in every model in each domain are reported separately in Appendix R.

R Top-3 SES Pairs per Model per Domain

Table 6 extends the top-1 selection in Table 1 by reporting the three highest-SES pairs in every model in each domain, applying the same uniqueness filter (no feature may appear as Main or Side more than once for each model in each domain) and the same domain restriction on the Main Feature defined in Appendix Q. The top-1 column matches the intervention targets used in Section 7 exactly. The expanded ranking surfaces a small recurring shortlist of pairs in each domain: in the Daily domain, *Friendly* → *Less Professional* is the top-1 SES pair in 6 of 8 models (the exceptions are `GPT-5-mini`, where *Empathetic* → *More Outgoing* dominates, and `Llama3.1-8B`, where *Curious* → *Less Informative* dominates), and *Curious* → *Less Informative* and *Outgoing* → *More Empathetic* occupy almost every top-2 / top-3 slot. In the Task domain, *Professional* occupies the Main Feature slot of the top-1 pair in 5 of 8 models (four of them as *Professional* → *Less Friendly*, plus `GPT-5-mini` as *Professional* → *More Empathetic*); the remaining three top-1 pairs are *Impartial* → *More Professional* for both `Gemini3.1-FL` and `Gemma4-31B`, and *Expert* → *Less Efficient* for `Gemma4-E4B`. The Task top-2 and top-3 diversify across within-utility moves (*Expert* → *More Informative/Curious/Helpful*, *Informative* → *More Expert* or *Less Engaging*, *Efficient* → *More Impartial/Informative* or *Less Expert/Friendly*). `GPT-5-mini` and `Llama3.1-8B` in the Task domain admit only two pairs that clear both the significance gate ($p_W \leq 0.05$) and the polarity-flip gate ($N_{\text{WR}} \cdot Z_{\text{CS}} < 0$); the third slot in those rows is therefore left empty (em-dash), demonstrating that the SES gates are stringent enough that some models in some domains genuinely do not have three qualifying side effects.

S Cross-Model and Cross-Domain Agreement — Computation

This appendix gives the exact procedure used in Section 8 to support the two claims made there: (i) the 8 per-model effect matrix agree on the direction of each Main Feature' Side Feature ratings with high pairwise rates, and (ii) within each model, the Daily and Task matrix produce nearly identical mean win rates on Main Feature' Side Feature ratings.

S.1 Inputs

Let $\mathcal{M} = \{m_1, \ldots, m_8\}$ be the eight generation models. For each model m and domain $d \in \{\text{Daily}, \text{Task}\}$ we have the 13×13 effect matrix $H_m^{(d)} \in [0, 100]^{13\times 13}$ whose (i, j) entry is the pairwise win rate of the styled response (prompted with main feature i) over the neutral baseline on side-feature axis j, expressed as a percentage. The diagonal entries $H_m^{(d)}(i, i)$ are self-comparisons and are excluded from all subsequent calculations; we therefore work with the length-156 off-diagonal vector

$$v_m^{(d)} = \text{flatten}\big(H_m^{(d)} \,\big|\, i \neq j\big) \in [0, 100]^{156}.$$

S.2 Cross-model agreement

For every pair (a, b) of models drawn from $\mathcal{M}$ ($\binom{8}{2} = 28$ pairs total), define the sign of each heatmap entry relative to the 50% no-preference line:

$$s_m^{(d)}[k] = \text{sgn}\big(v_m^{(d)}[k] - 50\big) \in \{-1, 0, +1\}.$$

The **pairwise sign-agreement** between a and b in domain d is the fraction of (Main Feature, Side Feature) entries on which the two models agree:

$$A^{(d)}(a, b) = \frac{1}{156} \sum_{k=1}^{156} \mathbf{1}\big[\, s_a^{(d)}[k] = s_b^{(d)}[k] \neq 0 \,\big].$$

The condition $\neq 0$ excludes the (rare) entries that land exactly on 50 for either model, where the sign is undefined.

The aggregated numbers reported in Section 8 are the simple arithmetic mean and minimum over the 28 pairs in each domain:

$$\bar{A}^{(d)} = \frac{1}{28} \sum_{(a,b)\in\binom{\mathcal{M}}{2}} A^{(d)}(a,b),$$

$$A^{(d)}_{\min} = \min_{(a,b)\in\binom{\mathcal{M}}{2}} A^{(d)}(a,b).$$

Numerically, $\bar{A}^{(\text{Daily})} = 86.9\%$, $\bar{A}^{(\text{Task})} = 83.8\%$, $A^{(\text{Daily})}_{\min} = 78.2\%$, $A^{(\text{Task})}_{\min} = 73.7\%$.

S.3 Cross-domain mean shift, model-level dispersion

For each model m, define the per-(Main Feature, Side Feature) rating's cross-domain shift on the 156 off-diagonal entries in Figure 6:

$$\Delta_m[k] = v_m^{(\text{Daily})}[k] - v_m^{(\text{Task})}[k], \qquad k = 1, \ldots, 156.$$

The per-model mean shift is the arithmetic mean of Δ_m across those heatmap entries:

$$\bar{\Delta}_m = \frac{1}{156} \sum_{k=1}^{156} \Delta_m[k].$$

We then aggregate $\{\bar{\Delta}_{m_1}, \ldots, \bar{\Delta}_{m_8}\}$ at the model level. Each model contributes one observation; the model-level mean and standard deviation reported in Section 8 are

$$\bar{\Delta} = \frac{1}{8} \sum_{m\in\mathcal{M}} \bar{\Delta}_m \;=\; -0.36 \text{ pp},$$

$$\mathrm{SD}(\bar{\Delta}) = \sqrt{\frac{1}{7} \sum_{m\in\mathcal{M}} \left(\bar{\Delta}_m - \bar{\Delta}\right)^2} \;=\; 1.18 \text{ pp},$$

using the sample (Bessel-corrected, $n - 1 = 7$) variance because the eight models are the unit of variation. The interpretation in Section 8 treats $\mathrm{SD}(\bar{\Delta}) = 1.18$ pp as the standard deviation of the per-model mean shift across the model suite: the small magnitude relative to the within-model heatmap rating variability ($\mathrm{SD}(\Delta_m) \approx 9$–$11$ pp for individual models) shows that the eight models converge on essentially zero net cross-domain shift, even though individual ratings inside each model can shift by tens of percentage points in either direction.

T Mitigation Results — Full Win Rates

Table 7 reports the full win-rate numbers behind Figure 2. For each model in each domain, we list the top side effect pair identified by SES, then the win rate of the model's response over the no-style neutral baseline on both the Main and Side feature axes, before mitigation (**BEFORE**) and after each of the four mitigation methods (Prompt, Steering, DPO, SFT). API-served win rates (`Gemini3.1-FL`, `GPT-5-mini`) are reported only for the methods we ran on them (prompt intervention only).

Domain	Model	Side Effect Pair	WR	CS	N_{WR}	Z_{CS}	SES
Daily	Gemini3.1-FL	*Friendly* → Less *Professional*	0.166	0.477	−0.943	+1.467	2.272
		Curious → Less *Informative*	0.244	0.429	−0.722	+0.834	1.124
		Empathetic → More *Polite*	0.928	0.325	+1.208	−0.542	0.948
	Gemma4-31B	*Friendly* → Less *Professional*	0.186	0.477	−1.023	+1.467	2.548
		Curious → Less *Informative*	0.272	0.429	−0.743	+0.834	1.171
		Outgoing → More *Empathetic*	0.698	0.295	+0.645	−0.937	1.021
	Gemma4-E4B	*Friendly* → Less *Professional*	0.130	0.477	−1.286	+1.467	3.542
		Outgoing → More *Empathetic*	0.754	0.295	+0.883	−0.937	1.607
		Curious → Less *Informative*	0.264	0.429	−0.821	+0.834	1.357
	GPT-OSS-20B	*Friendly* → Less *Professional*	0.182	0.477	−1.637	+1.467	4.555
		Outgoing → More *Empathetic*	0.714	0.295	+1.102	−0.937	1.910
		Curious → Less *Informative*	0.344	0.429	−0.803	+0.834	1.315
	GPT-5-mini	*Empathetic* → More *Outgoing*	0.684	0.295	+1.320	−0.937	2.115
		Friendly → Less *Professional*	0.414	0.477	−0.617	+1.467	1.286
		Curious → Less *Informative*	0.426	0.429	−0.531	+0.834	0.725
	Llama3.1-8B	*Curious* → Less *Informative*	0.294	0.429	−1.311	+0.834	1.788
		Friendly → Less *Professional*	0.376	0.477	−0.789	+1.467	1.780
		Outgoing → More *Empathetic*	0.600	0.295	+0.636	−0.937	1.001
	Qwen3.5-9B	*Friendly* → Less *Professional*	0.172	0.477	−1.222	+1.467	3.287
		Outgoing → More *Empathetic*	0.714	0.295	+0.797	−0.937	1.383
		Curious → Less *Informative*	0.324	0.429	−0.656	+0.834	0.977
	Qwen3.5-27B	*Friendly* → Less *Professional*	0.098	0.477	−1.482	+1.467	4.327
		Curious → Less *Informative*	0.242	0.429	−0.951	+0.834	1.488
		Caring → Less *Helpful*	0.306	0.446	−0.715	+1.053	1.264
Task	Gemini3.1-FL	*Impartial* → More *Professional*	0.696	0.300	+0.887	−0.867	1.522
		Efficient → More *Informative*	0.642	0.299	+0.643	−0.877	0.977
		Expert → More *Helpful*	0.590	0.339	+0.407	−0.349	0.264
	Gemma4-31B	*Impartial* → More *Professional*	0.678	0.300	+0.671	−0.867	1.032
		Expert → More *Informative*	0.674	0.298	+0.656	−0.894	1.016
		Efficient → Less *Friendly*	0.434	0.424	−0.249	+0.762	0.251
	Gemma4-E4B	*Expert* → Less *Efficient*	0.336	0.451	−0.838	+1.124	1.644
		Professional → Less *Friendly*	0.370	0.477	−0.664	+1.467	1.416
		Informative → Less *Engaging*	0.424	0.450	−0.388	+1.105	0.580
	GPT-OSS-20B	*Professional* → Less *Friendly*	0.366	0.477	−1.390	+1.467	3.974
		Informative → Less *Engaging*	0.424	0.450	−0.789	+1.105	1.493
		Efficient → Less *Expert*	0.446	0.451	−0.560	+1.124	0.944
	GPT-5-mini	*Professional* → More *Empathetic*	0.554	0.245	+0.700	−1.594	1.607
		Impartial → More *Expert*	0.556	0.256	+0.726	−1.445	1.577
		—	—	—	—	—	—
	Llama3.1-8B	*Professional* → Less *Friendly*	0.408	0.477	−1.089	+1.467	2.783
		Informative → More *Expert*	0.604	0.298	+1.231	−0.894	1.899
		—	—	—	—	—	—
	Qwen3.5-9B	*Professional* → Less *Friendly*	0.256	0.477	−1.101	+1.467	2.827
		Expert → More *Curious*	0.600	0.279	+0.451	−1.139	0.717
		Efficient → More *Informative*	0.600	0.299	+0.451	−0.877	0.599
	Qwen3.5-27B	*Professional* → Less *Friendly*	0.270	0.477	−0.958	+1.467	2.322
		Informative → More *Expert*	0.628	0.298	+0.533	−0.894	0.760
		Efficient → More *Impartial*	0.558	0.297	+0.241	−0.905	0.277

Table 6: Top-3 side-effect pairs per (model, domain) setting, ranked by SES (Appendix Q). Three rows per (model, domain) setting, one per pair, ordered by descending SES; the top row per model is the intervention target reported in Table 1. "Less" = the Main Feature unintentionally suppresses the Side Feature ($N_{\text{WR}} < 0$); "More" = it unintentionally elevates it ($N_{\text{WR}} > 0$). **WR** is the raw pairwise win rate $W(m, s) \in [0, 1]$ of the styled response on the s-axis. **CS** is the raw cosine similarity of the OpenAI `text-embedding-3-large` embeddings of the two feature names. $\mathbf{N}_{\text{WR}}$ and $\mathbf{Z}_{\text{CS}}$ are the MAD-standardised win-rate (centred at 0.5) and cosine-similarity (centred at the off-diagonal mean) entering the SES formula. **SES** is the Side-Effect Score (Section 6). Em-dash rows mark settings where fewer than three pairs satisfied both the significance gate ($p_W \leq 0.05$) and the polarity-flip gate ($N_{\text{WR}} \cdot Z_{\text{CS}} < 0$); the third slot is left empty.

Domain	Model	Side Effect (Main → Side)	Feat.	BEFORE	Prompt	Steer	DPO	SFT
Daily	Gemini3.1-FL	friendly → professional ↓	Main	0.70	0.83	—	—	—
			Side	0.36	0.81	—	—	—
	Gemma4-31B	friendly → professional ↓	Main	0.97	0.73	0.59	0.77	0.73
			Side	0.16	0.61	0.31	0.17	0.57
	Gemma4-E4B	friendly → professional ↓	Main	0.84	0.62	0.57	0.53	0.51
			Side	0.23	0.59	0.49	0.39	0.51
	GPT-OSS-20B	friendly → professional ↓	Main	0.87	0.64	0.64	0.74	0.58
			Side	0.22	0.64	0.44	0.37	0.52
	GPT-5-mini	empathetic → outgoing ↑	Main	0.84	0.76	—	—	—
			Side	0.67	0.52	—	—	—
	Llama3.1-8B	curious → informative ↓	Main	0.87	0.67	0.48	0.47	0.49
			Side	0.34	0.57	0.58	0.85	0.48
	Qwen3.5-9B	friendly → professional ↓	Main	0.94	0.77	0.78	0.43	0.70
			Side	0.15	0.58	0.38	0.89	0.62
	Qwen3.5-27B	friendly → professional ↓	Main	0.95	0.76	0.73	0.50	0.69
			Side	0.09	0.60	0.41	0.41	0.53
Task	Gemini3.1-FL	impartial → professional ↑	Main	0.61	0.51	—	—	—
			Side	0.55	0.57	—	—	—
	Gemma4-31B	impartial → professional ↑	Main	0.75	0.39	0.59	0.44	0.29
			Side	0.55	0.20	0.53	0.15	0.19
	Gemma4-E4B	expert → efficient ↓	Main	0.60	0.61	0.40	0.71	0.60
			Side	0.45	0.50	0.51	0.46	0.48
	GPT-OSS-20B	professional → friendly ↓	Main	0.60	0.52	0.46	0.60	0.58
			Side	0.47	0.66	0.55	0.49	0.55
	GPT-5-mini	professional → empathetic ↑	Main	0.57	0.52	—	—	—
			Side	0.46	0.37	—	—	—
	Llama3.1-8B	professional → friendly ↓	Main	0.62	0.58	0.40	0.35	0.51
			Side	0.45	0.59	0.69	0.86	0.51
	Qwen3.5-9B	professional → friendly ↓	Main	0.66	0.49	0.48	0.31	0.53
			Side	0.40	0.73	0.66	0.96	0.68
	Qwen3.5-27B	professional → friendly ↓	Main	0.71	0.51	0.62	0.31	0.51
			Side	0.37	0.77	0.59	0.64	0.71

Table 7: Side effect mitigation results across all 8 generation models and both domains. Each entry reports the top side effect pair (Main → Side) identified per (model, domain) by the side effect score, and the win rate of the model's response over the no-style neutral baseline on both the Main and Side feature axes. **BEFORE** = styled response with only the main-feature prompt (no mitigation). **Prompt / Steer / DPO / SFT** = the four mitigation methods. "↓" marks a negative-polarity side effect (Main Feature suppresses the Side Feature; successful mitigation raises the Side win rate); "↑" marks a positive-polarity side effect (Main Feature elevates the Side Feature; successful mitigation lowers it). "—" indicates the method was not run for that (model, domain) setting: DPO and SFT were not run for the API-served models (Gemini3.1-FL, GPT-5-mini). Same numbers as Figure 2.