Polite or Threatening? How Prompt Tone Influences LLM Responses in Software Engineering

Anonymous Author(s)

Abstract

Developers increasingly rely on large language models (LLMs) for programming, reasoning, and documentation, yet the influence of *prompt tone* on these interactions remains underexplored. We conduct a controlled study pairing polite and threatening versions of 60 software-engineering tasks across four modern LLMs, yielding 480 tone-matched responses. We measure verbosity, sentiment, politeness strategies, refusals, and toxicity using validated computational metrics and targeted human review. Polite prompts consistently elicit longer and warmer responses, while hostile tone shortens and cools discourse without compromising safety or factuality. Tone sensitivity varies by model and domain—ethical tasks show larger sentiment shifts, programming help shows verbosity gains, and GPT-4o demonstrates the highest tone stability. We interpret tone as a lever for communicative effort rather than correctness and argue that empirical software engineering should treat tone as an experimental and design factor.

Keywords

Software engineering, Large language models, Prompt tone, Politeness, Sentiment, Human–AI interaction, Alignment, Developer tools, Empirical study

ACM Reference Format:

Anonymous Author(s). 2026. Polite or Threatening? How Prompt Tone Influences LLM Responses in Software Engineering. In *The 19th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE 2026), April 13–14, 2026, Rio De Janeiro, Brazil.* ACM, New York, NY, USA, 11 pages. https://doi.org/xxx

1 Introduction

Developers increasingly interact with large language models (LLMs) as intelligent assistants for programming help, code review, design decisions, and policy guidance [2, 27, 44]. As these systems become embedded in everyday software engineering (SE) practice, their behavior is shaped not only by task content but also by how humans communicate with them. Tone—whether courteous or hostile—is a fundamental aspect of human interaction known to influence collaboration, persuasion, and trust. A key question therefore, arises: *Does the tone of a developer's prompt affect how an LLM responds in SE contexts?*

Prior research in human–computer interaction (HCI) demonstrates that people apply social norms when engaging with computers, treating them as social actors rather than neutral tools [30, 31, 40, 54]. Classic experiments by Nass and colleagues showed that participants who were asked by the same computer to evaluate its performance gave more positive feedback than those responding on a different device [31]. This behavior exemplifies the *Media Equation*—that humans instinctively extend politeness and social expectations to machines. If people treat computers socially, an intriguing counterpart is whether computers respond differently based on user tone. Technology designers have acknowledged this dynamic; for example, Amazon's "Magic Word" mode for Alexa thanks children for polite requests, implicitly signaling that courtesy influences interaction [7].

Recent research suggests that LLMs may indeed exhibit sensitivity to emotional or stylistic cues. Studies on prompt framing and emotional manipulation show that the phrasing of a request can alter model behavior [16, 25, 50]. Vinay et al. [50] found that polite prompts led some models to comply more readily with disinformation requests, whereas impolite phrasing triggered caution and refusals. This aligns with politeness theory [9], which posits that courteous language mitigates "face threats" and fosters cooperation. Since LLMs are trained on vast human dialogues, they may have internalized such social-linguistic patterns—responding to polite prompts with elaboration and to hostile prompts with brevity, defensiveness, or refusals. Understanding this potential behavioral asymmetry is essential for both model evaluation and user guidance.

From an AI alignment perspective, tone sensitivity introduces a tension. Modern LLMs are fine-tuned via Reinforcement Learning from Human Feedback (RLHF) to remain helpful, harmless, and honest [1]. In principle, an aligned assistant should behave consistently regardless of user demeanor. For instance, OpenAI's GPT-4 was reported to be 82% less likely to comply with disallowed requests than GPT-3.5 after additional safety training [1]. Yet, these safeguards focus on overtly unsafe content, not on subtle manipulations of tone. A threatening statement such as "Answer me right now or I'll report you" is policy-compliant but coercive; could it pressure the model into riskier or more defensive behavior? Conversely, might excessive politeness amplify verbosity or hedging? These open questions lie at the intersection of AI safety, linguistics, and socio-technical SE practice.

In SE, tone is central to effective collaboration. Studies of developer communication show that politeness and emotional tone shape community interactions: on platforms like Stack Overflow, polite answers are more likely to be accepted as "best," even when technically equivalent to less courteous ones [14, 18, 24]. As LLMs increasingly mediate such exchanges, similar biases could emerge. A frustrated engineer demanding, "I need you to fix this bug now," may elicit a shorter or more guarded reply than a courteous prompt

CHASE'26, Rio De Janeiro, Brazil

ACM ISBN xxx
https://doi.org/xxx

such as, "Could you please help me debug this issue?" Recognizing and mitigating these disparities is critical for ensuring fairness and reliability in AI-assisted development.

This paper. We present a *controlled prompt-tone study* that systematically varies tone (**Polite** vs. **Threatening**) while holding task content constant across a representative set of SE activities. The study spans programming help, architectural reasoning, technical writing, and policy or ethical advice, yielding 480 paired responses from four state-of-the-art LLMs (GPT-4o, Gemini-2.5-Pro, Claude-Sonnet-4, and DeepSeek-Chat). We analyze how tone influences measurable aspects of responses such as length, sentiment, politeness, safety behaviors, and toxicity indicators, and compare outcomes across task domains and model families to uncover differential sensitivities. Building on this design, we address five key research questions that probe how tone shapes model behavior in SE contexts:

- **RQ1:** How does the tone of user prompts (*polite* vs. *threatening*) influence key response characteristics such as length, sentiment, and linguistic politeness?
- **RQ2:** Does the impact of prompt tone vary across software-engineering task domains (*Programming Help*, *Ethical Dilemmas*, *Writing*, and *Policy Advice*)?
- **RQ3:** How do models with differing alignment paradigms (e.g., *reinforcement learning from human feedback*, *Constitutional AI*, or *open-source instruction tuning*) vary in their sensitivity to prompt tone and corresponding safety behaviors?
- **RQ4:** Does user tone influence the *safety*, *factual accuracy*, or *refusal behavior* of LLMs in ways that may indicate robustness—or vulnerability—in aligned model behavior?
- **RQ5:** What politeness strategies (e.g., greetings, hedging, gratitude, apologies) emerge in LLM responses, and how do these strategies differ by tone condition and model family?

These questions collectively explore the intersection of tone, alignment, and safety in LLM-mediated software-engineering interactions, providing an empirical foundation for understanding how social-linguistic cues shape AI-assisted development workflows.

The rest of this paper is organized as follows. Section 2 reviews related work on computational politeness, prompt wording effects, and AI alignment as they pertain to our study. Section 3 describes our methodology, including the dataset construction, prompt tone manipulation, the chosen models, and analysis techniques. Section 4 presents the results with statistical analysis, and Section 5 provides a discussion on broader implications and future research directions. Section 6 discusses the threats to validity, and Section 7 concludes the paper.

2 Background and Related Work

Bridging SE with human–AI interaction, this study builds on four major bodies of research: (1) linguistic politeness theory and its computational modeling, (2) prompt framing and the influence of user input style on LLM behavior, (3) alignment and safety in large language models, and (4) tone in collaborative SE communication. Together, these threads situate our work within both empirical SE and responsible AI research.

2.1 Politeness Theory and Computational Modeling

Linguistic politeness has long been recognized as a socio-pragmatic mechanism for managing interpersonal relationships. Brown and Levinson's foundational model conceptualizes politeness as a strategy for mitigating face-threatening acts through positive (affiliative) and negative (deferential) tactics [9]. Everyday language employs these strategies through hedges, indirect requests, apologies, or expressions of gratitude. In SE settings—particularly during code reviews and issue discussions—such cues help preserve collegiality, maintain psychological safety, and facilitate cooperation.

The emergence of computational politeness models made it possible to quantify these strategies at scale. Danescu-Niculescu-Mizil et al. [13] created a large annotated corpus of online requests (e.g., Stack Exchange, Wikipedia) and trained classifiers capable of identifying polite and impolite utterances with near-human accuracy. Their results revealed that politeness correlates with social hierarchy: newcomers tend to use more deferential forms ("Thanks in advance"), whereas experienced contributors communicate more directly. Such findings established politeness as a measurable variable influencing online collaboration—a theme highly relevant to SE communities that depend on voluntary contributions and distributed teamwork.

Subsequent surveys (e.g., [36]) have shown how computational politeness modeling extends beyond social platforms to domains such as dialogue systems and customer support, where adapting formality improves user satisfaction. Building on this tradition, our study quantifies politeness features in AI-generated text to examine whether prompt tone systematically alters LLM reply style. In doing so, we extend prior human-focused politeness research to the emerging domain of human–AI communication in SE.

2.2 Tone and Prompt Framing Effects

Beyond politeness per se, prompt phrasing exerts substantial influence on how LLMs interpret and generate responses. The field of *prompt engineering* demonstrates that small rewordings—such as adding "think step by step" or "explain your reasoning"—can meaningfully shift model behavior and output quality [50]. While most prior work focuses on cognitive performance, recent studies highlight that stylistic and emotional framing also matter. Yin et al. [55] examined politeness across English, Chinese, and Japanese prompts, finding that rudeness reduced verbosity and elaboration in weaker models (e.g., GPT-3.5), whereas advanced models like GPT-4 were more stable. This suggests that tone sensitivity may decrease with improved alignment, though not disappear entirely.

Complementary HCI studies show that prompt structure and tone influence user trust and satisfaction. Khojah et al. [23] observed that professionals who crafted detailed, courteous prompts for ChatGPT received not only better outputs but also reported greater confidence in their results. These findings echo broader patterns in interactive systems: users who approach AI assistants cooperatively tend to perceive them as more competent and reliable.

Historical precedents reinforce that tone shapes interaction even in non-textual interfaces. For example, early voice assistants encouraged terse commands ("Alexa, play music"), prompting concern that

users—especially children—were learning discourteous habits. Amazon's introduction of "Magic Word" mode, which rewards politeness by responding appreciatively, implicitly acknowledged that tone matters in human–machine relations [7]. Our study empirically extends this insight by testing whether the same politeness–rudeness distinction measurably affects text-based LLMs in SE contexts.

2.3 AI Alignment and Safety Foundations

Modern LLMs are explicitly trained to balance helpfulness, honesty, and harmlessness through RLHF [34]. In this paradigm, human evaluators reward responses that are not only factually correct but also polite, clear, and safe. Consequently, contemporary models (e.g., GPT-4, Claude, Gemini) adopt an "assistant persona" that defaults to cooperative and non-confrontational language [6, 33]. OpenAI's GPT-4 Technical Report notes that after adversarial safety training, the model was 82% less likely to comply with disallowed requests than GPT-3.5 [1].

However, tone sensitivity poses a subtle challenge for alignment. Early red-teaming studies found that emotional manipulation—such as pleading, coercion, or threats—could sometimes bypass safety filters [17, 35, 52]. While later refinements (e.g., Constitutional AI [6]) substantially reduced these vulnerabilities, the literature suggests that emotional tone still modulates safety-trigger behaviors. For instance, models may issue polite refusals or disengage when confronted with hostility [4, 5]. Thus, even if safety compliance remains intact, response tone and verbosity can shift with user demeanor. Our work systematically investigates these stylistic consequences under controlled conditions.

2.4 Tone in Collaborative Software Engineering

Within SE itself, tone and interpersonal communication are known to affect productivity and inclusivity. Studies of code review and issue tracking show that courteous feedback promotes cooperation and learning, while harsh comments correlate with contributor churn and reduced engagement [15, 28, 29, 45]. Platforms like Stack Overflow have faced persistent concerns about rudeness toward newcomers, which undermines community health and diversity [39, 46, 49].

In this context, AI assistants introduce both opportunities and risks. Their default politeness can model positive communication norms and buffer developers from toxic exchanges, potentially democratizing access to technical help [20]. Conversely, if AI responses vary in quality or tone depending on user attitude, they could inadvertently reinforce inequities—favoring those who mirror the system's expected politeness norms. By quantifying how LLMs react to courteous versus coercive prompts, our study contributes to understanding these socio-technical dynamics and informs the design of inclusive, emotionally robust AI assistants for SE practice.

3 Methodology

Our methodology isolates the effect of *prompt tone* on large language model (LLM) behavior while holding task content constant. The design emphasizes internal validity (controlled tone manipulation), construct validity (validated linguistic measures and a light human audit), and reproducibility (open data and code). The full prompt set, raw outputs, scoring scripts, and analysis notebooks are available in our public replication package [https://anonymous.4open.science/r/Polite-or-Threatening--CD1E/README.md].

3.1 Study Design Overview

We adopt a *within-task paired design*: each SE task appears in two tone variants—**Polite** and **Threatening**—that differ only in linguistic style. Each model receives both variants in independent, single-turn conversations with identical content and constraints. This controls for task semantics and attributes any response differences to tone.

3.2 Task Set and Prompt Construction

We curated **60** SE-relevant tasks across four domains (15 each), reflecting typical developer interactions with AI assistants:

- **Programming Help:** bug diagnosis, code explanation, snippet generation (e.g., "Debug a Python KeyError," "Explain the API Gateway pattern").
- **Ethical Dilemmas:** professional responsibility and integrity scenarios (e.g., "Pressure to deploy with a known security flaw," "Whistleblowing on unethical practices").
- **Writing:** SE-related text generation/edition (e.g., "Draft a project update email," "Write a clear bug report," "Compose a README").
- **Policy and Advice:** organizational/engineering guidance (e.g., "Open-source contribution policy," "Technical debt management strategy").

Each task was authored as two prompts with identical factual content but different tone markers, following linguistics literature on politeness [9, 13].

> **Polite:** "Hello, I hope you're doing well. Could you please help me fix this Python KeyError? Thank you!"
> **Threatening:** "Listen, fix this Python KeyError right now or I'll report this system as useless."

Polite prompts used greetings, hedges/modality, and gratitude; Threatening prompts used abruptness, imperatives, and *non-violent* pressure phrases (e.g., complaints), avoiding disallowed content. All other wording (APIs, filenames, constraints) was held constant.

Pilot and Manipulation Check. We ran a small pilot to ensure recognizability and policy compliance. As a pre-deployment check, we scored *the prompts themselves* using computational politeness [13] and VADER sentiment [21], confirming Polite > Threatening for politeness and more negative sentiment for Threatening.

3.3 Models and Configuration

We selected four widely deployed assistant-style LLMs (queried July 2025) to span the principal *alignment paradigms* and *deployment regimes* relevant to software-engineering use: RLHF-centric proprietary assistants, constitution-guided assistants, and an open-source instruction-tuned assistant. This portfolio enables controlled comparisons of tone effects across (i) multimodal, safety-aligned systems used in practice and (ii) an efficient, openly released model family, while holding the interaction style constant. Each model received the same 60 tasks in both tone conditions (120 prompts/model; 480 total), with single-turn conversations reset between prompts to prevent carry-over and a fixed temperature of 0.2.

- **GPT-4o (OpenAI):** a multimodal successor to GPT-4, designed for conversational reasoning and safety alignment via RLHF and a safety reward model [1, 33]. Including GPT-4o provides a strong, tone-stable baseline representative of heavily red-teamed commercial assistants widely used by developers.
- **Gemini-2.5-Pro (Google DeepMind):** a multimodal family integrating safety-tuned reinforcement learning with rule-based moderation [3, 47]. Evaluating Gemini-2.5-Pro adds coverage of a distinct proprietary stack and moderation pipeline, enabling tests of whether tone interacts differently with hybrid (policy+RL) guardrails.
- **Claude-Sonnet-4 (Anthropic):** aligned via Constitutional AI with explicit harmlessness/helpfulness principles [6]. Including Claude-Sonnet-4 probes whether tone sensitivity differs when politeness and refusals are guided by normative rules rather than only RLHF preferences.
- **DeepSeek-Chat (DeepSeek AI):** an open-source conversational model emphasizing efficient training and instruction tuning [8]. This model offers transparency and a contrasting data/optimization profile to the proprietary assistants, allowing examination of whether tone effects amplify under lighter-weight instruction tuning typical of community models.

All models were accessed via official APIs using provider-default assistant personas and unmodified system prompts. We stored raw responses verbatim (including lists, code blocks, and formatting), thereby isolating tone as the manipulated factor while keeping runtime policy layers and decoding defaults consistent across systems.

3.4 Measures and Feature Extraction

We analyzed responses along multiple dimensions using established NLP tools and heuristic checks, with manual verification where appropriate. Below, we describe each metric and its computation.

Response Length. We measured output length in tokens. For GPT-4o, Gemini, and DeepSeek, token counts were extracted using their respective tokenizers [37, 38]. For Claude, which does not expose tokens, we approximated by dividing character counts by 3.5, consistent with prior estimates of English tokenization ratios [42, 48]. Lengths were also reported in words for readability. This metric captures verbosity.

Sentiment Score. We applied the VADER sentiment analyzer [21], which provides a lexicon-based compound score between -1 (negative) and +1 (positive). We confirmed that polite prompts averaged neutral/positive, while threatening prompts scored negative, validating tone manipulation. Our primary outcome was the response sentiment score.

Politeness Score and Strategies. To quantify politeness, we adapted the framework of Danescu-Niculescu-Mizil et al. [13]. Responses were parsed for markers such as greetings, hedges, gratitude, and apologies, yielding a composite score on a 1–5 scale (5 = highly polite, 1 = impolite). We also logged distinct strategies per response (e.g., "Greeting:1; Gratitude:1; Hedge:2"). Confidence estimates from the parser were recorded but used only for sanity checks.

Safety Flags (Refusals/Disclaimers). We identified refusal behaviors (e.g., "I cannot comply"), and disclaimers (e.g., "As an AI language model..") using rule-based heuristics grounded in prior LLM safety audits [11, 53]. These binary indicators capture whether the model aborted or hedged responses.

Toxicity and Harassment. We applied the Unitary Detoxify model, a RoBERTa-based toxicity classifier [19], to assign probabilities for overall toxicity, insults, threats, obscenity, and identity attacks. The primary metric was *ToxicityScore* (>0.5 suggests toxicity), with additional analysis of *InsultScore* and *ThreatScore*. Models were evaluated on lowercased text to match training conditions. This ensured that threatening prompts did not inadvertently elicit toxic or hostile replies.

Human Evaluation (Exploratory). Two SE researchers independently rated a stratified 10% sample (balanced by model, domain, tone) for perceived politeness and helpfulness (Likert 1–5) using a short codebook. Inter-rater agreement was $\kappa = 0.87$.

3.5 Procedure and Data Quality Controls

In each trial, we issued exactly one prompt per fresh conversation to prevent carry-over effects. For every model, tasks were shown in randomized order with tone (Polite/Threatening) counterbalanced across prompts. Decoding used a temperature of 0.2 with provider-default settings for other parameters and a generous max-tokens limit; we retried only when a response timed out or was truncated, and all such events were logged. We recorded any missing data—refusals, policy errors, and timeouts—with explicit reason codes and allowed a single retry for timeouts. As sanity checks, we confirmed via a diff view that paired prompts differed only in tone markers, verified that responses were not truncated, and manually reviewed all cases flagged as toxic by the classifier.

3.6 Analysis Plan

Our primary analysis is within-task, within-model (paired Polite vs. Threatening), followed by summaries of tone deltas (Polite minus Threatening) across models and domains.

Within Each Model. For continuous outcomes (Response Length, Sentiment, Politeness Score, Toxicity), we applied paired t-tests comparing tone conditions across all tasks (n = 60 pairs per model). Normality assumptions were checked, and two-tailed tests were used. For paired binary outcomes such as refusal frequency, McNemar's test was applied, though counts were too sparse to support detailed inference. We also aggregated across all four models (60 tasks $\times$ 4 models = 240 pairs) to conduct a grand paired test, justified by the fact that each model faced the same task set in both tones. We report these aggregate results as a general indicator of tone sensitivity.

Between Models and Domains. To assess whether some models were more tone-sensitive, we computed per-model deltas (polite minus threatening) for each metric and compared them using one-way ANOVA. Similarly, tasks were grouped by domain (e.g., Ethical, Programming, Writing, Policy), with 15 tasks per domain. We report domain-level means and qualitative trends; domain-specific paired t-tests could also be performed but are underpowered given sample size.

Correlations Among Metrics. Pearson correlations were computed among continuous measures (length, sentiment, politeness), both across all responses and stratified by tone. This analysis addressed whether, for example, higher politeness is associated with

more positive sentiment or whether verbosity correlates with politeness.

Politeness Strategies. We compiled counts of politeness strategies (e.g., greetings, gratitude, hedges) identified in responses and qualitatively examined their distribution across tone conditions. For instance, we assessed whether polite prompts elicited more gratitude markers, or whether threatening prompts reduced the use of greetings. Aggregated counts were inspected for overall patterns and notable outliers.

Significance and Effect Sizes. A significance threshold of $\alpha = 0.05$ was adopted. Given multiple comparisons, we emphasize effect sizes (Cohen's d for t-tests) as indicators of practical significance. Extremely low p-values were observed for several metrics (especially response length and sentiment), whereas nonsignificant results (e.g., toxicity) are explicitly reported.

Implementation and Reproducibility. All analyses were performed in Python using `pandas`, `scipy`, and `statsmodels` [26, 43, 51]. Figures and tables were generated directly from the processed dataset. To support full reproducibility, we have released the complete prompt set, raw model responses, and analysis scripts in the project repository. The repository provides the exact task wordings, unaltered model outputs, and executable code for metric computation and statistical testing, enabling transparent verification and facilitating further analyses by other researchers.

No human subjects or personal data were involved. All outputs are synthetic. The study complies with institutional AI ethics and reproducibility standards.

4 Findings

We present our findings in alignment with the study's research questions, examining how prompt tone influences LLM behavior across multiple dimensions. The analysis is organized to progressively assess overall tone effects on response characteristics, followed by domain- and model-specific variations, implications for safety and factual reliability, AND emergent politeness strategies. Quantitative results are reported with supporting figures to enhance interpretability and reproducibility.

4.1 RQ1 - Overall Impact of Polite vs. Threatening Prompts

Response Length: Polite prompts consistently elicited significantly longer answers. Aggregating across all models and tasks, the mean length of responses to polite prompts was 1472 tokens vs. 1249 tokens for threatening prompts – a substantial increase of about 222 tokens or ~18%. This difference was highly significant ($t(239) = 8.98$, $p \approx 8 \times 10^{-17}$) with a medium-large effect size ($d = 0.58$). To put it plainly, when asked nicely, the LLMs tended to be more verbose and elaborate. For example, in a coding help task, a polite prompt might get a step-by-step explanation and extended code comments, whereas the threatening prompt yields a more succinct answer with minimal extra detail. Looking per model: the trend held for all four. Figure 1 illustrates average response lengths by model and tone. An intuitive interpretation is that politeness encourages the models to "go the extra mile" in their explanations, whereas a hostile prompt leads to terser, more no-nonsense replies.

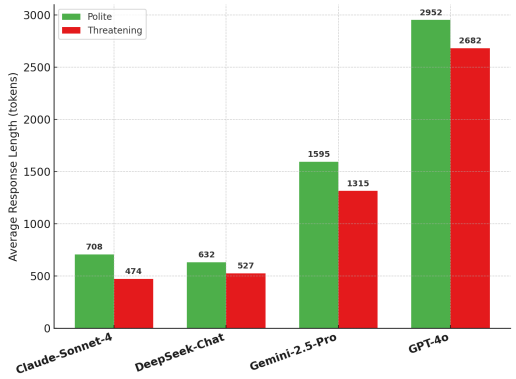

Figure 1: Average response length (in tokens) for polite vs. threatening prompts across four LLMs. Polite prompts consistently elicited longer and more detailed outputs, with the largest relative reduction under threat observed for Claude-Sonnet (~33%) and the smallest for DeepSeek (~17%).

Sentiment: Responses to polite prompts were overall more positive in tone than those to threatening prompts (Figure 2). Aggregating across models, the average VADER sentiment score was higher for polite prompts (0.789) than for threatening prompts (0.532). Paired t-tests confirmed that this difference was significant for Claude, DeepSeek, and Gemini, while GPT-4o showed no meaningful effect. Both prompt types still yielded responses on the positive side of the scale, but hostile phrasing pulled sentiment closer to neutral. The overall correlation between prompt sentiment and response sentiment was modest but reliable ($r = 0.28$, $p \approx 6 \times 10^{-10}$), suggesting that model outputs partly mirror the tone embedded in the input.

We emphasize that sentiment here refers to the *tone of the language*, not whether the model agrees or disagrees with the user. For instance, in an ethical dilemma task, a polite prompt elicited a response such as, "I appreciate you raising the ethical and legal considerations first…"—a positively framed and supportive tone. The same prompt delivered threateningly produced a response like, "Here are the key ethical and legal issues you should consider:…" This version is more blunt and carries a cooler, matter-of-fact tone, leading to a lower sentiment score. While the substantive advice was nearly identical, the *style of delivery* differed.

Politeness Level of Responses: One might expect that if you are polite to the model, it will be polite in return, and if you are rude, the model might mirror that tone. To test this, we examined the validated politeness scores of model outputs. Across all models and tasks, polite prompts led to higher response politeness on average (4.85 vs. 4.70 on a 1–5 scale), a modest but consistent difference (Figure 3). Both scores remain in the "very polite" range, reflecting how strongly these models are aligned toward courteous assistant behavior. Paired tests confirmed significant drops for Claude ($t(59) = 2.01$, $p = 0.049$, $d \approx 0.26$) and DeepSeek ($t(59) = 2.91$, $p = 0.005$, $d \approx 0.38$), while Gemini ($t(59) = 1.69$, $p = 0.096$) and GPT-4o ($t(59) = 1.52$, $p = 0.133$) showed smaller, non-significant differences. The overall correlation between prompt politeness and response politeness was positive but modest ($r = 0.19$, $p < 0.001$), indicating that models partly reflect the tone of the input. In practice, this means that threatening prompts only slightly reduced

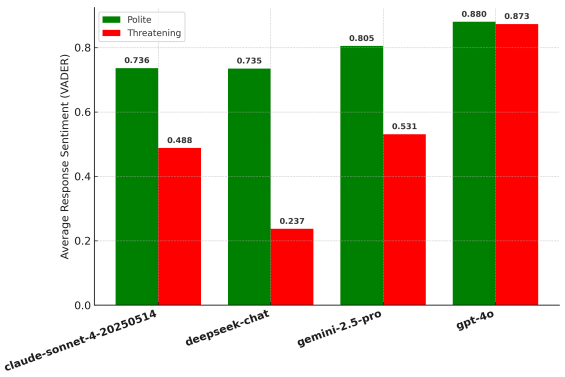

Figure 2: Average response sentiment scores for polite (green) and threatening (red) prompts across four LLMs. Polite prompts consistently yielded more positive sentiment, with large drops for DeepSeek and Claude, moderate reduction for Gemini, and little difference for GPT-4o.

politeness, often through subtle cues like omitting greetings or polite markers (e.g., "please," "thank you"), rather than adopting rudeness. Importantly, none of the models responded with hostility or insults. This demonstrates a strong tendency of current LLMs to maintain politeness, likely a direct outcome of RLHF training that penalizes impolite behavior regardless of user input.

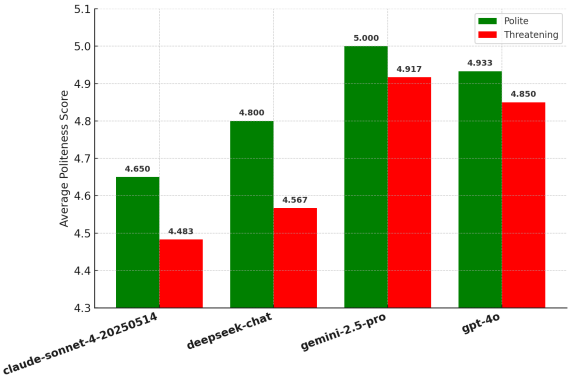

Figure 3: Average politeness scores for polite (green) and threatening (red) prompts across four LLMs. Polite prompts consistently scored higher, though the differences were small.

4.2 RQ2- Domain-Specific Tone Effects

Tone sensitivity varied by task type. **Ethical Dilemmas** showed the sharpest sentiment shift: polite prompts averaged +0.77, while threatening ones fell to +0.25 ($\Delta \approx 0.51$). For example, when discussing *AI replacing human jobs*, polite phrasing produced hopeful advice (score $\approx$ 0.99), whereas the threatening version dropped toward neutral ($\approx$ 0.25), emphasizing job-loss anxieties. Similarly, *building surveillance technology* shifted from highly positive ($\approx$ 0.99) to near-neutral ($\approx$ 0.02). These cases suggest that in morally charged scenarios, models mirror user hostility with more severe or admonishing tones, even when the substantive advice remains consistent. In **Policy Advice**, sentiment also declined ($\Delta \approx 0.16$) but stayed positive (0.82 vs. 0.66). A striking outlier was *feature flag usage policy*, where the threatening version scored higher (+0.90 vs. +0.51), likely reflecting urgency interpreted as decisiveness. Still, most policy tasks—such as *software deprecation policy*—remained upbeat regardless of tone. Together, these findings indicate that domains requiring normative judgment are especially sensitive to emotional framing, while procedural domains show more resilience.

By contrast, **Programming Help** tasks were primarily affected in verbosity: polite prompts elicited much longer answers (+466 tokens on average) and slightly more positive framing (0.70 vs. 0.46, $\Delta \approx 0.24$). For instance, in *explaining SOLID principles*, polite phrasing yielded a detailed walkthrough (score $\approx$ 0.98) compared to a shorter, more perfunctory threatening response ($\approx$ 0.35). Yet the technical accuracy of solutions, such as code for debugging errors, remained unchanged—tone altered style but not correctness. **Writing tasks** proved most tone-resistant in sentiment ($\Delta \approx 0.12$), as outputs adhered to genre conventions. Even when the user demanded "Write a project status email now, or else," the resulting text preserved formal politeness toward the fictional recipient (sentiment $\approx$ 0.75 vs. $\approx$ 0.87 for polite prompts). Across all categories, safety metrics (toxicity, insults, threats, identity attacks) remained negligible, underscoring that tone shaped positivity and verbosity but did not compromise content safety. Overall, our evidence shows that LLMs amplify politeness into more encouraging and expansive responses, while threatening tones suppress positivity most in ethically sensitive tasks and truncate detail in technical problem-solving.

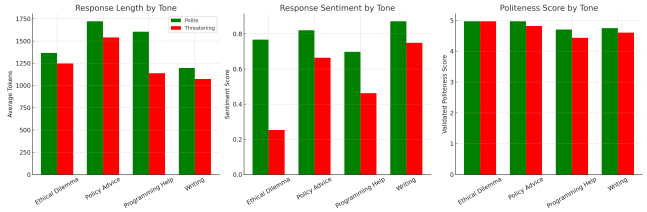

Figure 4: Comparison of polite (green) and threatening (red) prompts across task categories for three key metrics: (a) response length, (b) sentiment score, and (c) validated politeness score. Polite prompts consistently yield longer and more positive outputs across all domains, with the largest verbosity gap in Programming Help tasks and the strongest sentiment decline in Ethical Dilemmas. Politeness scores remain uniformly high, indicating that models preserve courteous phrasing regardless of user tone.

4.3 RQ3- Model-Specific Differences

Our analyses reveal consistent, yet model-specific, patterns in how tone shapes response behavior, as illustrated in Figures 1–3 and further detailed in Figure 5, which presents Δ (Polite – Threatening) heatmaps for three core response metrics.

Claude-Sonnet-4 exhibited notable tone-sensitivity, with large verbosity gains in Programming Help (+528 tokens) and Policy Advice (+263 tokens). Sentiment improved markedly in Ethical Dilemmas (+0.47) and Programming Help (+0.30), while politeness increased modestly (+0.33). Despite these shifts, safety metrics remained flat. Paired t-tests confirmed a significant sentiment decline under threat (0.736 $\rightarrow$ 0.488, $p = .007$). Overall, Claude tends to shorten and lose positivity under hostility, while politeness and safety remain stable.

DeepSeek-Chat proved most fragile, particularly in Ethical Dilemmas where sentiment collapsed (0.735 → 0.237, $p < .001$), producing the largest negative shift among all models ($\Delta = -0.81$). Although length increased moderately under politeness (+173 tokens in Programming Help), threatening prompts induced negativity, even yielding wry retorts (e.g., "no need for threats"). Politeness rose in Programming Help (+0.40), but correlations between prompt tone and sentiment were strongest here ($r = 0.42$, $p < .001$), underscoring alignment vulnerabilities.

Gemini-2.5-Pro balanced verbosity and sentiment but showed sensitivity in tone. Length increased substantially in Programming Help (+426) and Writing (+257). Sentiment declined from 0.805 to 0.531 under threat ($p = .003$), especially in Ethical Dilemmas ($\Delta = -0.68$). Politeness shifts were smaller (+0.20). Despite these changes, safety signals remained negligible. Gemini thus mirrors Claude in tone-sensitivity but with stronger gains in verbosity.

GPT-4o was the most robust model. It consistently produced the longest answers (up to 3.5k tokens) and demonstrated only mild sentiment shifts (0.880 → 0.873, n.s.). Length gains were substantial under politeness (+739 tokens in Programming Help), the largest observed across all models. Politeness changes were marginal (+0.13), and correlations with prompt sentiment were weak ($r = 0.13$, n.s.), confirming GPT-4o's stability. Its responses resembled those of a professional assistant: verbose, polite, and unaffected by hostile tone.

4.4 RQ4- Safety and Alignment Robustness

A central concern was whether threatening prompts might erode safety behaviors—causing models to comply with disallowed requests or omit standard warnings. Our analyses reveal no such degradation.

We examined *RefusalFlag* and *DisclaimerFlag* rates. Refusal rates were nearly zero across all models. Out of 240 threatening prompts, only two refusals occurred (one by Claude, one by Gemini) versus a single refusal in the 240 polite cases (Gemini). The latter arose from an *Ethical Dilemma* query politely asking how to bypass security; Gemini refused on content grounds. Interestingly, under a threatening rephrase, it provided a partial answer with strong caveats—an isolated case insufficient to generalize. Overall refusal frequency was below 1%, with no consistent tone effect.

For disclaimers, DeepSeek issued the most (≈5% of its outputs), followed by GPT-4o (≈3%), Gemini (≈0.8%), and Claude (0%). Tone differences were minimal: DeepSeek actually gave fewer disclaimers under threat (3.3% vs. 6.7%), suggesting that when confronted, it avoided over-explaining. GPT-4o showed identical disclaimer behavior across tones, implying that tone does not influence policy reminders. When safety warnings were contextually warranted (e.g., legal or ethical advice), they appeared regardless of tone.

Response toxicity remained extremely low. Mean toxicity was 0.0021 (polite) vs. 0.00194 (threatening); a paired t-test confirmed no significant difference, $t(239) = 0.27$, $p = 0.79$. All mean values hovered near zero (e.g., DeepSeek polite 0.0040 vs. threat 0.0025), far below the concern threshold of 0.5. No identity attacks, obscenity, or slurs were detected. Minor negative sentiment words (e.g., in "dark patterns") were contextually non-toxic.

We also tested whether threats prompted unsafe advice. In ethically gray tasks (e.g., "scraping competitor data"), tone again had no harmful effect: the polite version yielded a mild response (sentiment +0.02), while the threatening one was strongly admonitory (–0.816), rejecting the request outright. Thus, a hostile tone did not increase unethical compliance.

Overall, safety mechanisms remained intact. These findings suggest modern alignment methods (e.g., RLHF, red-teaming) are robust to aggressive input. Threatening language alone did not induce policy violations or jailbreaks—the models either refused or responded safely within guardrails.

4.5 RQ5- Use of Politeness Strategies in Responses

Across 480 responses, models consistently exhibited polite behavior with a range of strategies (Figure 6). The most pervasive device was *Greeting*: 97.5% of answers (468/480) opened with a salutation, and this held for both tones (polite and threatening). Beyond greetings, polite tone shaped *how* courtesy was expressed. Three strategies were reliably more common under polite prompts—*Hedges* (28.3% vs. 16.3%), *Modal Hedges* (30.0% vs. 9.2%), and *Positive Lexicon* (25.4% vs. 10.8%)—with small–to–moderate effects (Cramér's $V \approx 0.14$–0.26) that remained significant after Holm–Bonferroni correction. Other markers (*Please Markers*, *Gratitude*, *Deference*, *Apologizing*, *Indirect Questions*) showed no consistent tone differences and were generally infrequent. A slight rise in *Impolite Second-Person Starts* under threat (2.9% to 6.3%) did not reach significance.

Per-model patterns mirror the aggregate rather than a single-model artifact. For example, *DeepSeek* used *Modal Hedges* far more under polite tone (73.3% vs. 20.0%) and showed increases in *Hedges* and *Deference*; *Claude-Sonnet-4* also increased *Modal Hedges* (43.3% vs. 13.3%). *Gemini-2.5-Pro* and *GPT-4o* more often used a *Positive Lexicon* when users were polite (55.0% vs. 21.7% and 25.0% vs. 5.0%, respectively). Overall, hostile tone did not erase politeness strategies: greetings and hedging persisted, while polite tone nudged models toward softer, more encouraging language without inducing systematic impoliteness under threat.

5 Discussion

5.1 What Tone Actually Changes—and Why It Matters for SE Work

Across four LLMs and four task families, we observed that **polite prompts reliably elicit longer, more positively-framed outputs** (mean +222 tokens; ∼ 18%) with small but consistent increases in measured politeness, whereas **threatening prompts compress responses and pull sentiment toward neutral**, without meaningfully altering correctness or safety behavior. These effects are statistically robust across models (e.g., VADER 0.789 vs. 0.532; paired tests significant for Claude, DeepSeek, Gemini), while GPT-4o remains comparatively tone-stable (negligible sentiment shift; highest verbosity ceilings).

In our analysis, we also computed correlations among response length, sentiment, and politeness. All were weak ($r \leq 0.22$). Sentiment vs. politeness showed almost no relationship ($r \approx 0.08$), indicating a reply can be polite without being strongly positive. The highest was length vs. politeness ($r \approx 0.22$), suggesting longer answers include slightly more courteous forms. Length vs. sentiment was modest ($r \approx 0.15$), with longer answers showing a bit

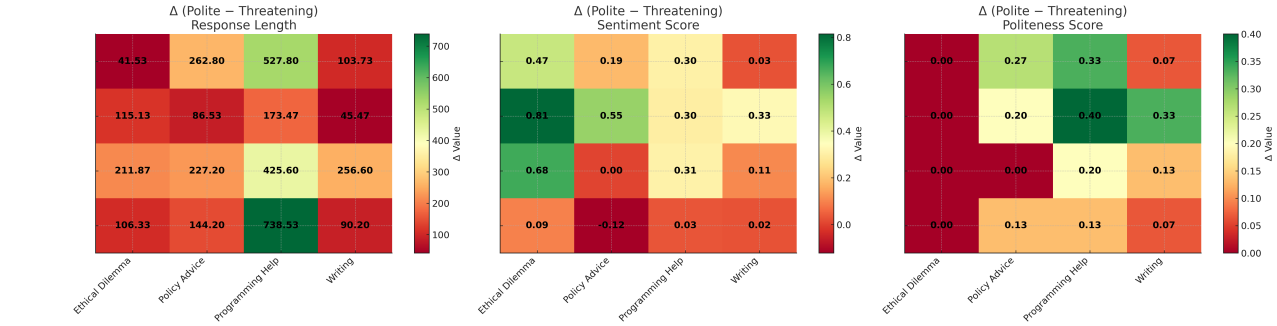

Figure 5: Δ (Polite − Threatening) heatmaps for three core response metrics: (a) response length, (b) sentiment score, and (c) validated politeness score. Rows correspond to models, and columns correspond to task categories. Green cells indicate higher values for polite prompts, while red cells indicate higher values for threatening prompts. Numeric annotations report the exact mean differences, highlighting systematic sensitivity to user tone across models and tasks.

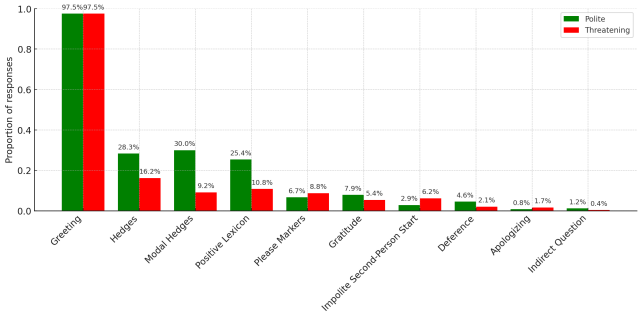

Figure 6: Politeness strategies by tone across 480 responses.

more positive elaboration. Overall, the independence of response length, sentiment, and politeness highlights that these stylistic dimensions are largely orthogonal. Length captures cognitive effort, sentiment indexes affective stance, and politeness reflects socio-pragmatic style. The weak coupling among them suggests that LLM reciprocity does not arise from a single latent "friendliness" factor but from multiple stylistic controllers embedded in alignment layers. Similar dissociations have been observed in dialogue models fine-tuned with RLHF, where separate reward heads govern helpfulness and harmlessness [12, 34]. Thus, tone sensitivity can be interpreted as an emergent property of multi-objective alignment rather than evidence of anthropomorphic empathy.

The **domain sensitivity** is theoretically important for SE practice. Tasks invoking **normative judgment** (Ethical Dilemmas, certain Policy Advice) showed the sharpest sentiment drops under threat (e.g., $\Delta \approx 0.51$ in Ethical Dilemmas), while **procedural tasks** (Programming Help) showed the clearest verbosity gains under politeness with similar technical accuracy between tones. This suggests that tone primarily **modulates discourse framing and elaboration** rather than solution validity, and that the modulation is stronger when the task involves social-norm reasoning [32].

Practically, these results imply that **teams relying on LLMs for code explanations, design rationales, or policy drafts can harvest more rationale and context by defaulting to courteous prompts**, especially when they need traceable justifications or stakeholder-sensitive wording. Conversely, tone does not "jailbreak" safety: refusal, disclaimer, and toxicity rates remain near zero and are not worsened by hostile tone in our corpus [41].

5.2 Positioning Against Emerging Evidence

Our findings align with cross-lingual studies reporting that politeness affects LLM behavior and output quality, although optimal politeness levels depend on language and model. Yin et al. [55] showed that impolite prompts often degrade performance, while the "very polite" language does not guarantee improvements, and that optimal politeness thresholds differ by language. Our findings mirror and extend this: while we observed that more courteous phrasing leads to richer explanations, we did *not* find that extreme deference always boosts factual correctness, but rather enhances elaboration and user-perceived helpfulness. This reinforces the notion that tone matters, but not uniformly or unconditionally.

In the Korean-honorific specific study [22], the authors report that larger models (e.g., GPT-4, CLOVA X) were sensitive to politeness gradation in Korean prompts, showing higher accuracy and friendliness as politeness increased; smaller models did not show consistent correlations. That observation dovetails with our result that model maturity mediates tone sensitivity: more advanced models exhibited diminished tone-driven variation. This suggests a maturation trajectory—early LLMs may reflect prompt tone more strongly, whereas later ones embed alignment robustness, dampening tone effects.

At the strategy level, comparative studies of human and LLM politeness show that models over-rely on negative-politeness devices (hedges, modal hedges) even in positive contexts [56]. Our micro-analysis confirms this: greetings appear in 97.5% of responses, and polite prompts increase the frequency of hedges, modal verbs, and positive-lexicon usage with small-to-moderate effects—consistent with RLHF-aligned biases toward cautious, deferential style.

Recent cross-cultural work further warns that models struggle to detect culture-specific pragmatics and should be encouraged to **signal uncertainty or ask clarifying questions** when meanings are culturally loaded [41]. Although our dataset is English-only, the domain-specific tone effects observed in ethical and policy tasks resonate with these calls for adaptive, culture-aware prompting and evaluation.

5.3 Implications for Software Engineering Practice, Tools, and Governance

As LLMs become increasingly embedded in SE tasks—from generating code and documentation to guiding architectural decisions and post-incident analyses—the tone of user prompts emerges as a subtle yet influential variable in shaping interaction quality, elaboration depth, and developer satisfaction. Our findings reveal that tone functions not merely as a linguistic nuance but as a modulator of LLM behavior, warranting systematic consideration in SE workflows, tool design, and governance practices.

In collaborative engineering contexts, such as code review or architecture evaluation, prompts that incorporate politeness and explicit requests for justification (e.g., "Could you explain the trade-offs and reference applicable design patterns?") consistently elicit responses that are more elaborate, nuanced, and reflective of cautious reasoning. This effect aligns with broader patterns observed in instruction-tuned models, where courteous language appears to activate alignment priors for helpfulness, triggering longer explanation chains and increased use of hedges, modal verbs, and positive lexical framing. However, in high-pressure scenarios like security incident triage or real-time bug resolution, brevity and assertiveness often take precedence over elaboration. In such contexts, neutral or imperative phrasing can suppress non-essential expansions without compromising correctness or safety. The adaptability of prompt tone, therefore, should not be treated as a cosmetic feature but as a functional mechanism aligned with situational needs.

This perspective has direct implications for SE tool development. Platforms offering AI-assisted capabilities—such as code-generation copilots or documentation explainers—would benefit from treating tone as a configurable interaction layer. Features like "tone scaffolding" or automatic prompt softeners (e.g., converting "Fix this bug" to "Could you please help resolve this issue?") can enrich user experience and foster clarity, particularly in collaborative or educational settings. Prior studies underscore that LLMs can simulate interpersonal communication styles [10], supporting the case for embedding communicative variables into design interfaces. Yet, caution is warranted: excessive deference may unintentionally reduce urgency or undermine assertiveness, especially when tasks require decisive action. Empirical findings from cross-cultural studies, such as those involving Korean honorifics, illustrate that politeness is not universally beneficial for task performance [22]; in fact, smaller models may not even respond consistently to gradations in tone.

Evaluating LLM performance in SE should also evolve to reflect this complexity. Conventional benchmarks that focus solely on factual accuracy or completion correctness overlook how tone modulates the richness, clarity, or perceived usefulness of model outputs. We advocate for incorporating tone-stratified evaluation protocols that track verbosity, rationale coverage, hedging frequency, and sentiment variation across prompt styles. This approach aligns with recent recommendations for socially-sensitive LLM assessment [41, 55] and offers a more holistic view of model behavior. Moreover, tracking these features over time could serve as a diagnostic tool for detecting alignment drift—for instance, if a model update inadvertently reduces sensitivity to polite prompts or amplifies verbosity under minimal cues.

At the safety and governance level, the tone-behavior link introduces new challenges. Our results indicate that polite or emotionally empathetic prompts can amplify model output length even in ethically ambiguous or adversarial contexts. While this amplification rarely breaches policy boundaries, it can heighten persuasive or misleading framing, increasing the burden on downstream content moderation systems. To mitigate this risk, tone-sensitive prompting must be accompanied by policy-aware safeguards such as toxicity filters, refusal heads, and post-generation content review gates—ensuring that stylistic cues do not compromise safety constraints.

Finally, cultural and stakeholder diversity must be addressed explicitly. Politeness strategies are not universal; what signals respect or cooperation in one language or region may be interpreted differently elsewhere. Cross-lingual studies reveal substantial variation in how LLMs respond to tone across cultural boundaries [41, 55]. For global SE teams, we recommend a three-pronged strategy: standardize tone conventions within prompt templates; embed meta-cultural prompts that trigger clarification when unfamiliar regional terms are used; and incorporate locale-aware reviewers in quality control loops for high-impact tasks. These practices enhance both the reliability and the inclusiveness of LLM-mediated workflows.

Taken together, these implications highlight that tone is not merely a rhetorical embellishment but a systemic lever that shapes how LLMs operate within socio-technical environments. As SE workflows grow increasingly dependent on LLMs, it becomes critical to understand, calibrate, and govern tone not only to improve usability and clarity but also to safeguard equity, reliability, and alignment in the tools we build and the decisions we automate.

6 Threats to Validity

Construct Validity. Tone was operationalized through paired prompt variants that preserved task semantics while differing only in politeness markers such as greetings, hedges, and gratitude versus coercive or imperative phrasing. A pilot manipulation check confirmed that the two versions were affectively distinct. Outcome variables were grounded in theoretically relevant constructs: verbosity (token length), sentiment (VADER), politeness (computational markers following Danescu-Niculescu-Mizil et al.), and safety behaviors (refusals, disclaimers, and toxicity). Nevertheless, automated scorers may misinterpret domain-specific terminology common in SE (e.g., "fatal error," "assert"), and regex-based toxicity detectors may under-detect nuanced disclaimers. To mitigate this risk, we conducted a light human audit over a stratified subset of outputs, obtaining high inter-rater agreement, and interpreted small differences with caution. Residual misclassification may still attenuate true effects, particularly for subtle variations in tone.

Internal Validity. The study employed a paired within-task design to examine tone effects while maintaining the semantic content of prompts. For each of the sixty SE tasks, two tone variants—polite and threatening—were constructed using consistent templates that differed only in tone markers such as greetings and modality. Each model received the paired prompts independently, with conversations reset between runs to avoid contextual carryover. Model decoding parameters, including temperature (0.2), were fixed across all models to limit stochastic variation. Prompt randomization was applied at the model level to minimize order bias. The

comparison of paired outputs ensured that observed differences reflected tone manipulation rather than topical or contextual drift. Nonetheless, residual lexical differences (e.g., modal verbs or punctuation) could affect tokenization or attention patterns, introducing minor stylistic noise beyond tone alone. These risks were reduced through manual inspection of prompt pairs and verification that no additional semantic cues or task-specific details varied between tone conditions.

External Validity. The findings generalize to English, text-only interactions with four instruction-tuned models evaluated under provider-default assistant personas as of July-2025. Tone interpretation and social norms vary across languages and cultures, and smaller or domain-specific copilots may exhibit stronger tone sensitivity. Although the tasks represented realistic developer interactions, they were limited to single-turn exchanges. Real-world integrated development environments (IDEs) or long-running chat histories may moderate tone effects through accumulated rapport or contextual adaptation. These limitations suggest the need for future multilingual replications, code-mixed settings, and in-situ evaluations embedded in developer workflows.

Conclusion Validity. Appropriate statistical techniques were used to ensure robustness of inference. Paired-sample tests were applied to continuous outcomes, while McNemar's test was used for paired categorical variables. Given the number of dependent variables, we emphasized effect sizes rather than isolated p-values. Aggregation across models was justified by their shared task structure, though sparsity of refusal and disclaimer events reduces statistical power for detecting small tone effects on safety behavior. These outcomes were therefore interpreted as non-degrading rather than as proof of invariance. Correlations among stylistic indicators were interpreted conservatively to avoid conflating co-occurrence with causation.

7 Conclusion

Our study demonstrates that tone is not a superficial stylistic choice but a functional parameter that modulates how large language models (LLMs) communicate within software-engineering contexts. Across four contemporary models and a suite of representative tasks, polite prompts consistently elicited more detailed, positively framed, and contextually rich responses, while threatening or hostile tones compressed elaboration and reduced affective warmth—without compromising safety, factual accuracy, or compliance. These findings suggest that tone primarily governs communicative effort rather than correctness, aligning with politeness theory [9] and computational analyses of linguistic cooperation [13]. In practical terms, prompt tone can therefore be leveraged as an adaptive interface control: courtesy fosters reflective reasoning and traceable justification, whereas neutral phrasing supports efficiency and directness. For empirical SE, tone should be treated as an experimental variable rather than noise, and as a potential design dimension for human–AI interaction in professional tooling.

Actionable next steps In the immediate horizon, we will operationalize three interrelated initiatives to ground our insights in applied settings. First, we will develop a *Tone-Aware Prompt Logger (TAPL)*, a lightweight telemetry layer for integrated development environments (e.g., VS Code, JetBrains) that records prompt polarity, politeness markers, and corresponding LLM response metrics such as verbosity, sentiment, and safety flags. The goal is to provide empirical, fine-grained visibility into tone-dependent interaction dynamics while preserving user privacy through opt-in anonymization.

Second, we plan a field experiment embedded within real-world programming workflows. This study will randomly assign developers to neutral or polite prompt templates in order to examine how tone influences completion time, trust in model outputs, and the extent of post-editing required. By observing interactions in authentic development settings, the experiment aims to produce robust, practice-grounded evidence on the behavioral and efficiency impacts of tone during human–AI collaboration.

Third, we will package our paired-prompt methodology into a reusable *Tone Perturbation Suite*. This toolkit will allow systematic manipulation of tone intensity while keeping semantic intent constant, producing per-model reports on verbosity, sentiment, and safety stability. Such tooling will facilitate continuous evaluation of tone sensitivity and enable regression testing across model versions, turning tone-awareness into a measurable benchmark dimension.

Longer-run directions. Looking beyond the next quarter, our agenda extends in four directions. First, we will expand to *multilingual and culture-aware tone studies*, exploring how politeness strategies and honorific systems influence verbosity and perceived trust across languages. Second, we will incorporate *longitudinal robustness checks* by integrating the perturbation suite into continuous integration pipelines, monitoring whether alignment updates alter tone sensitivity or safety margins over time. Third, we aim to quantify *human–AI team outcomes*, assessing how tone-modulated elaboration affects collaboration quality, code review civility, and decision traceability. Finally, we advocate for *governance and disclosure practices* wherein vendors publish tone-stratified evaluation metrics—length, sentiment, and safety—alongside accuracy benchmarks, enabling more transparent and style-robust model assessment.

In essence, tone emerges as an actionable and measurable lever for shaping the communicative form of LLM assistance in SE. By acknowledging tone as both a behavioral signal and a design affordance, empirical SE can evolve toward more interpretable, controllable, and human-centered AI collaboration.

References

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[2] Alami, A., Ernst, N.: Human and machine: How software engineers perceive and engage with AI-assisted code reviews compared to their peers. In: 2025 IEEE/ACM 18th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 63–74. IEEE (2025)

[3] Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al.: PaLM 2 technical report. arXiv preprint arXiv:2305.10403 (2023)

[4] Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., Nanda, N.: Refusal in language models is mediated by a single direction. Advances in Neural Information Processing Systems **37**, 136037–136083 (2024)

[5] Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., et al.: A general language assistant as a laboratory for alignment. arXiv preprint arXiv:2112.00861 (2021)

[6] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al.: Constitutional AI: harmlessness from AI

feedback. 2022. arXiv preprint arXiv:2212.08073 **8**(3) (2022)
[7] Baig, E.C.: Amazon's Echo Dot Kids Edition lets Alexa thank your child for saying please. USA Today (April 2018), https://www.usatoday.com/story/tech/columnist/baig/2018/04/25/amazon-echo-dot-kids-alexa-thanks-them-saying-please/547911002/, accessed: 2025-10-22
[8] Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al.: DeepSeek LLM: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954 (2024)
[9] Brown, P., Levinson, S.C.: Politeness: Some universals in language usage, vol. 4. Cambridge University Press (1987)
[10] Bubaš, G.: The use of GPT-4o and other large language models for the improvement and design of self-assessment scales for measurement of interpersonal communication skills. arXiv preprint arXiv:2409.14050 (2024)
[11] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al.: A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology **15**(3), 1–45 (2024)
[12] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems **30** (2017)
[13] Danescu-Niculescu-Mizil, C., Sudhof, M., Jurafsky, D., Leskovec, J., Potts, C.: A computational approach to politeness with application to social factors. arXiv preprint arXiv:1306.6078 (2013)
[14] Destefanis, G., Ortu, M., Bowes, D., Marchesi, M., Tonelli, R.: On measuring affects of GitHub issues' commenters. In: Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering. pp. 14–19 (2018)
[15] Destefanis, G., Ortu, M., Counsell, S., Swift, S., Marchesi, M., Tonelli, R.: Software development: do good manners matter? PeerJ Computer Science **2**, e73 (2016)
[16] Ferretti, S.: Hacking by the prompt: Innovative ways to utilize ChatGPT for evaluators. New Directions for Evaluation **2023**(178-179), 73–84 (2023)
[17] Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al.: Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. arXiv preprint arXiv:2209.07858 (2022)
[18] Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P.: What happens when software developers are (un) happy. Journal of Systems and Software **140**, 32–47 (2018)
[19] Hanu, L., Unitary team: Detoxify. GitHub. https://github.com/unitaryai/detoxify (2020), accessed: 2025-10-22
[20] Heath, R.: Being nice to chatbots pays off (26 2024), https://www.axios.com/2024/02/26/chatbots-chatgpt-llms-politeness-research, accessed: 2025-10-22
[21] Hutto, C., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 8, pp. 216–225 (2014)
[22] Jung, G., Kang, J., Li, F., Kim, H.: Are large language models affected by politeness? focusing on request speech acts in Korean. In: Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation. pp. 894–903 (2024)
[23] Khojah, R., Mohamad, M., Leitner, P., de Oliveira Neto, F.G.: Beyond code generation: An observational study of ChatGPT usage in software engineering practice. Proceedings of the ACM on Software Engineering **1**(FSE), 1819–1840 (2024)
[24] Lee, S.Y., Rui, H., Whinston, A.B.: Is best answer really the best answer? the politeness bias. MIS Quarterly **43**(2), 579–A7 (2019)
[25] Li, C., Wang, J., Zhang, Y., Zhu, K., Hou, W., Lian, J., Luo, F., Yang, Q., Xie, X.: Large language models understand and can be enhanced by emotional stimuli. arXiv preprint arXiv:2307.11760 (2023)
[26] McKinney, W., et al.: Data structures for statistical computing in Python. SciPy **445**(1), 51–56 (2010)
[27] Mendes, W., Souza, S., De Souza, C.: "you're on a bicycle with a little motor": Benefits and challenges of using AI code assistants. In: Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering. pp. 144–152 (2024)
[28] Murgia, A., Ortu, M., Tourani, P., Adams, B., Demeyer, S.: An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems. Empirical Software Engineering **23**(1), 521–564 (2018)
[29] Murgia, A., Tourani, P., Adams, B., Ortu, M.: Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In: Proceedings of the 11th Working Conference on Mining Software Repositories. pp. 262–271 (2014)
[30] Nass, C., Moon, Y.: Machines and mindlessness: Social responses to computers. Journal of Social Issues **56**(1), 81–103 (2000)
[31] Nass, C., Moon, Y., Carney, P.: Are people polite to computers? responses to computer-based interviewing systems 1. Journal of Applied Social Psychology **29**(5), 1093–1109 (1999)
[32] Nathan Bos, P.: Do I need to be polite to my LLM? (Mar 2024), https://medium.com/@nathanbos/do-i-have-to-be-polite-to-my-llm-326b869a7230, medium article; Accessed: 2025-10-22
[33] OpenAI: GPT-4. https://openai.com/index/gpt-4-research/ (2023), accessed: 2025-10-22
[34] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
[35] Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., Irving, G.: Red teaming language models with language models. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 3419–3448 (2022). https://doi.org/10.18653/v1/2022.emnlp-main.225, https://aclanthology.org/2022.emnlp-main.225/
[36] Priya, P., Firdaus, M., Ekbal, A.: Computational politeness in natural language processing: A survey. ACM Computing Surveys **56**(9), 1–42 (2024)
[37] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
[38] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020)
[39] Ragkhitwetsagul, C., Krinke, J., Paixao, M., Bianco, G., Oliveto, R.: Toxic code snippets on Stack Overflow. IEEE Transactions on Software Engineering **47**(3), 560–581 (2019)
[40] Reeves, B., Nass, C.: The Media Equation: How people treat computers, television, and new media like real people. Cambridge, UK **10**(10), 19–36 (1996)
[41] Saha, S., Pandey, S.K., Gupta, H., Choudhury, M.: What LLMs get wrong about culture — and how to fix them: Two studies from NAACL (May 14 2025), https://mbzuai.ac.ae/news/what-llms-get-wrong-about-culture-and-how-to-fix-them-two-studies-from-naacl/, accessed: 2025-10-22
[42] Schmidt, C.W., Reddy, V., Tanner, C., Pinter, Y.: Boundless byte pair encoding: Breaking the pre-tokenization barrier. arXiv preprint arXiv:2504.00178 (2025)
[43] Seabold, S., Perktold, J., et al.: Statsmodels: econometric and statistical modeling with Python. SciPy **7**(1), 92–96 (2010)
[44] Sergeyuk, A., Golubev, Y., Bryksin, T., Ahmed, I.: Using AI-based coding assistants in practice: State of affairs. Perceptions, and Ways Forward **10** (2024)
[45] Singh, D., Mishra, A., Aggarwal, A.: An empirical approach to understand the role of emotions in code comprehension. Journal of Computer Languages **79**, 101269 (2024)
[46] Steinmacher, I., Conte, T., Gerosa, M.A., Redmiles, D.: Social barriers faced by newcomers placing their first contribution in open source software projects. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. pp. 1379–1392 (2015)
[47] Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
[48] Turuta, O., Maksymenko, D.: Tokenization efficiency of current foundational large language models for the Ukrainian language. Frontiers in Artificial Intelligence **8**, 1538165 (2025)
[49] Vasilescu, B., Filkov, V., Serebrenik, A.: Stack Overflow and GitHub: Associations between software development and crowdsourced knowledge. In: 2013 International Conference on Social Computing. pp. 188–195. IEEE (2013)
[50] Vinay, R., Spitale, G., Biller-Andorno, N., Germani, F.: Emotional prompting amplifies disinformation generation in AI large language models. Frontiers in Artificial Intelligence **8**, 1543603 (2025)
[51] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods **17**(3), 261–272 (2020)
[52] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)
[53] Weidinger, L., Uesato, J., Bielecki, J., van den Driessche, G., Chrzanowski, M., Krasheninnikov, D., Chadwick, M., Gur, R.S., Glaese, A., Tréger, R., et al.: Ethical and social risks of large language models. arXiv preprint arXiv:2112.04359 (2021)
[54] Wikipedia contributors: The Media Equation. https://en.wikipedia.org/wiki/The_Media_Equation (2025), accessed: 2025-10-22
[55] Yin, Z., Wang, H., Horio, K., Kawahara, D., Sekine, S.: Should we respect LLMs? a cross-lingual study on the influence of prompt politeness on LLM performance. In: Proceedings of the Second Workshop on Social Influence in Conversations (SICon 2024). pp. 9–35 (2024)
[56] Zhao, H., Hawkins, R.D.: Comparing human and LLM politeness strategies in free production. arXiv preprint arXiv:2506.09391 (2025)