Prompt-with-Steer: A Process-Oriented Interaction Framework for Fine-Grained Steering of LLMs

Anonymous ACL submission

Abstract

Aligning Large Language Model (LLM) outputs with user's constraints often requires iterative refinement, yet standard conversational interfaces treat models as black boxes, leading to error propagation and feedback ambiguity. Current interaction paradigms typically force users to regenerate entire responses to correct local errors, resulting in inefficient trial-and-error loops. To address these limitations, we introduce Prompt-with-Steer, an interactive framework that surfaces generation dynamics to enable precise user intervention during inference. Our approach supports prefix rollback regeneration and token-level selection to repair local violations without re-prompting the entire response. We evaluate our framework on an enhanced Instruction Following Evaluation (IFEval) benchmark with verifiable hard constraints. Prompt-with-Steer facilitates higher correction accuracy compared to dialogue-based baselines across different models, achieving high modification success rates on targeted constraints while reducing generation token consumption by up to 70%. These findings demonstrate that fine-grained, inspectable steering is a more interaction-efficient mechanism for aligning LLM outputs with verifiable generation requirements.

1 Introduction

The prevailing paradigm in human-computer interaction for Large Language Models (LLMs) is conversational dialogue (OpenAI et al., 2024; Radford et al., 2019; Fan et al., 2024; Bowden and Walker, 2025; Doran and Dahl, 2024). Within this paradigm, extensive research has been conducted on instruction following (Sun et al., 2024; Ren et al., 2024), and alignment (Ouyang et al., 2022; Ji et al., 2025; Kim and Yang, 2025). While these pre-alignment strategies provide valuable algorithms, they often lack human involvement during inference, making it difficult to align with specific, hard constraints in real-time.

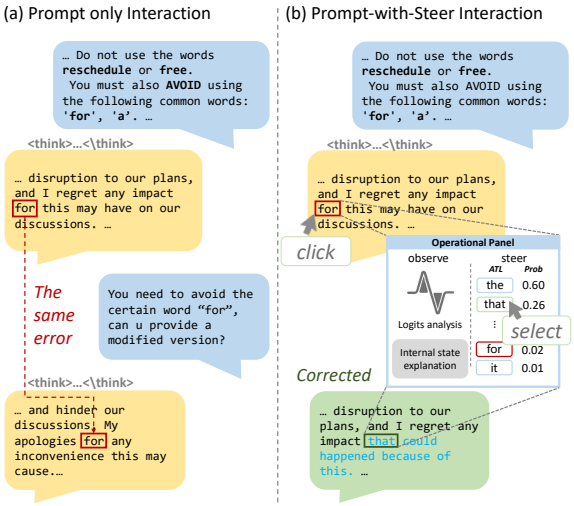

Figure 1: Comparison between (a) Prompt only Interaction and (b) our proposed Prompt-with-Steer Interaction. The dialogue example is taken from the Taboo test of the enhanced IFEval dataset.

In real-world testing and applications, users frequently encounter scenarios where they must correct specific parts of a generated response to meet certain demands. However, current conversational models typically treat the entire previous dialogue—including the erroneous segments—as context (Liu et al., 2024a; Wang and Sun, 2025). This leads to two critical issues during re-generation: dialogue error propagation and ambiguity in user feedback(Siro et al., 2024). Without clear and detailed instructions in multi-turn dialogue, models often fail to judge which parts of the generation need to be preserved and which parts need to be modified. Meanwhile, users cannot predict what will be resampled in this round due to the probabilistic nature of language models. In this interaction paradigm, alignment remains implicit, making it difficult for users to provide precise corrective guidance. Furthermore, re-generating the entire response to fix a local error results in significant

computational inefficiency and token waste.

From a Human-Computer Interaction (HCI) perspective, we argue that effective cooperation in high-precision scenarios requires augmenting the dialogue paradigm with steering capabilities. Merely communicating through conversational interfaces is often insufficient for strict constraints when models lack introspection into *what* the user truly needs. Instead, we propose a complementary interaction modality that aligns specific generation requirements with the model's internal understanding: click-and-choose. Users can take advantage of the auto-regressive mechanism of language models, inspect the model's step-wise generation trajectory and intervene precisely at the point of divergence.

Based on this insight, we propose **Prompt-with-Steer**, an interactive framework for fine-grained inference-time intervention. Prompt-with-Steer leverages internal model signals—such as layer shifts and logit distributions—to assist human decision-making by exposing heuristic proxies of generation instability. Crucially, it empowers users to act as a "surgeon," pinpointing the exact token where generation diverges. We acknowledge that this granular control introduces higher cognitive load compared to conversational interfaces; however, we demonstrate that this trade-off offers necessary precision for expert users or critical tasks where "approximate correctness" is unacceptable. In our system, users can apply prefix rollback regeneration and/or select alternative next-token candidates, which removes the erroneous suffix from the conditioning context and mitigates error propagation.

In summary, our contributions are as follows:

- We propose a steerable, process-oriented interaction mechanism that mitigates error propagation and feedback ambiguity during generation. We formalise an interactive steering workflow, demonstrating effectiveness in aligning the model with verifiable constraints and improving token efficiency in human-AI interaction.

- We introduce Prompt-with-Steer, a modular system that integrates logit-lens based diagnostic cues and proposes certain interaction-oriented metrics to guide human intervention through an interactive steering interface.

- We construct an enhanced instruction following dataset based on IFEval (Zhou et al., 2023), specifically tailored for hard constraint alignment, and conduct extensive experiments with human volunteers. The results show that our system-level workflow improves over dialogue-only interaction under the same checker protocol across different LLMs, particularly for reasoning-intensive models.

2 Related Work

Human-AI Interaction Current Human-AI interaction is mostly defined by chat-based interfaces (Chatterji et al., 2025; Amershi et al., 2019; OpenAI et al., 2024), treating the model as a black box where users influence outputs solely through prompting. While prompt engineering remains the primary mechanism for instruction following, it provides only indirect and coarse-grained control over model behavior (Brown et al., 2020; Wei et al., 2023). As task instructions become more complex or constraint-heavy, users are forced to iteratively reformulate prompts without visibility into how instructions are internally interpreted (Liu et al., 2023; Zamfirescu-Pereira et al., 2023).

Recent HCI research has explored more "inspectable" interfaces to bridge this gap. For instance, ReasonGraph (Li et al., 2025) visualizes the reasoning process as a graph, allowing users to manage context more effectively. Similarly, Vis-CoT (Pather et al., 2025) introduces "direct manipulation" capabilities, enabling users to edit or infill text by pruning incorrect paths and grafting new premises.

Activation Steering Steering LLM behavior at inference time without parameter updates has emerged as a powerful alignment technique. Activation Steering methods, such as Inference-Time Intervention (ITI) (Li et al., 2024) and Representation Engineering (RepE) (Zou et al., 2025), identify and manipulate "steering vectors" in the model's activation space to enforce behaviors like truthfulness or harmlessness. However, these approaches are typically applied offline or by experts, offering limited affordances for end users to express or refine task-specific instructions through interaction (Zou et al., 2025). Complementary to this, decoding strategies like Contrastive Decoding (Li et al., 2023) and DoLa (Chuang et al., 2024) exploit layer-wise discrepancies (e.g., "Logit Lens") to reduce hallucinations and improve factuality.

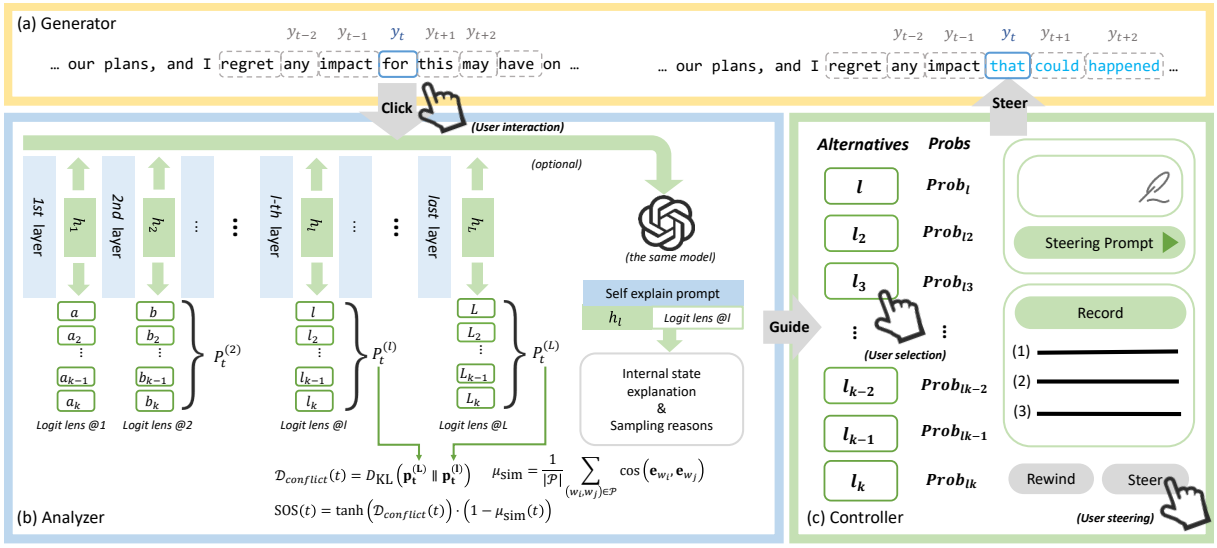

Figure 2: Illustration of the Prompt-with-Steer framework, composed of (a) a Generator, using an LLM to perform autoregressive generation; (b) an Analyzer, analyzing internal signals to provide diagnostic guidance; and (c) a Controller, enabling users to steer the generation process.

Process Supervision Moving beyond outcome-based supervision, recent work views LLM reasoning as a search process. Methods like Tree of Thoughts (ToT) (Yao et al., 2023) and Graph of Thoughts (Besta et al., 2024) utilize branching and backtracking to explore diverse reasoning paths. These are often coupled with Process Reward Models (PRM) (Zhang et al., 2025), which score intermediate steps to guide the search, effectively applying Monte Carlo Tree Search (MCTS) (Świechowski et al., 2022) principles to language generation. While effective at improving reasoning quality, these methods optimize for solution correctness rather than instruction compliance (Creswell and Shanahan, 2022; Hewitt et al., 2024). They provide limited mechanisms for users to intervene when intermediate reasoning diverges from intended constraints or priorities.

3 Methodology

3.1 Overview

The Prompt-with-Steer framework is designed to shift LLM interaction from dialogue-only to prompting with steering. Unlike standard conversational interfaces, where users only see the final output, our framework exposes the model's generation dynamics—specifically proxy signals for uncertainty and potential semantic shifts—allowing users to intervene during the generation process. Besides, it also provides a click-and-choose interaction method for using a token suggested by the model that matches their intent.

The framework consists of three coupled modules as shown in Figure 2: the *Generator*, which manages autoregressive inference and supports prefix rollback regeneration; the *Analyzer*, which computes diagnostic signals and guides steering; and the *Controller*, which provides the interface for users to execute "surgical" interventions. This design enables a tight feedback loop where human intent modulates the generation trajectory.

3.2 Generator

The generation process acts as a trajectory through a high-dimensional state space. At time step t, the model's state $\mathcal{S}_t$ is encapsulated by the Key-Value (KV) cache of all preceding tokens, denoted as $\mathcal{S}_t = \{(\mathbf{K}_i, \mathbf{V}_i)\}_{i=1}^{t-1}$, where $\mathbf{K}_i, \mathbf{V}_i$ representing the key and value vectors at position i. The probability of the next token y_t is conditioned on this state and the current input x(or y_{t-1}):

$$\mathbf{p}(y_t|y_{<t}, x) = \text{softmax}\left(\mathbf{W}_U \cdot \mathbf{h}_t^{(L)}\right), \quad (1)$$

where

$$\mathbf{h}_t^{(L)} = f(\mathcal{S}_t, y_{t-1}) \quad (2)$$

represents the final-layer hidden state produced by the Transformer forward pass f and $\mathbf{W}_U$ is the unembedding matrix. In standard autoregressive decoding, the trajectory extends linearly:

$$\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{(\mathbf{K}_t, \mathbf{V}_t)\}. \quad (3)$$

Our method interrupts this linear flow by treating $\mathcal{S}_t$ as a mutable object that can be dynamically inspected and pruned.

3.3 Analyzer

To guide user intervention, we must translate high-dimensional internal states into interpretable signals. We employ three primary lenses:

Internal Conflict and Logit Lens While final-layer state $\mathbf{h}_t^{(L)}$ dictates the model's committed output, intermediate layers often encode suppressed alternatives. Motivated by prior analyses of layer-wise representations (Liu et al., 2024b; Jin et al., 2025), we inspect an intermediate layer for analysis. Let $\mathbf{h}_t^{(l)}$ denote the hidden state at layer l. We utilize the *Logit Lens* technique to project $\mathbf{h}_t^{(l)}$ directly into the vocabulary space using the shared unembedding matrix $\mathbf{W}_U$, yielding an intermediate distribution $\mathbf{p}_t^{(l)} = \text{softmax}\left(\mathbf{W}_U \mathbf{h}_t^{(l)}\right)$. We quantify internal conflict as the divergence between this layer-wise "intuition" and the final output $\mathbf{p}_t^{(L)}$:

$$\mathcal{D}_{\text{conflict}}(t) = D_{\text{KL}}\left(\mathbf{p}_t^{(L)} \parallel \mathbf{p}_t^{(l)}\right) \tag{4}$$

A high divergence suggests a significant discrepancy between the intermediate representation and the final output distribution. We treat this as a heuristic signal that may indicate potential generation instability or conflict.

Steering Opportunity Score (SOS) Raw entropy is often noisy; a model may be uncertain between two synonyms (e.g., "happy" vs. "glad") which is benign. To identify meaningful intervention points, we propose the Steering Opportunity Score (SOS), which weights internal conflict by semantic diversity.

Let $\mathcal{V}_t^{(k)} \subset \mathcal{V}$ be the set of the top-k tokens predicted by the final layer. We define the semantic cohesiveness μ_{sim} as the average pairwise cosine similarity between the embeddings of these tokens:

$$\mu_{\text{sim}} = \frac{1}{|\mathcal{P}|} \sum_{(w_i, w_j) \in \mathcal{P}} \cos(\mathbf{e}_{w_i}, \mathbf{e}_{w_j}) \tag{5}$$

where $\mathbf{e}_w$ denotes the embedding vector of token w, and $\mathcal{P}$ is the set of all unique pairs in $\mathcal{V}_t^{(k)}$. We then define the SOS as:

$$\text{SOS}(t) = \tanh(\mathcal{D}_{\text{conflict}}(t)) \cdot (1 - \mu_{\text{sim}}(t)) \tag{6}$$

Here, the term $1 - \mu_{\text{sim}}(t)$ acts as a semantic filter: it vanishes when candidates are synonymous ($\mu_{\text{sim}}(t) \approx 1$) and amplifies the score when candidates are semantically distinct ($\mu_{\text{sim}} \ll 1$). Prompt-with-Steer assigns different shades of color to different tokens based on their SOS. A high SOS suggests a potential "bifurcation point" in the generation trajectory, serving as a heuristic hint for human steering.

Internal Explanation To complement the numeric SOS signal, we optionally provide a natural-language explanation view (Ghandeharioun et al., 2024) that helps users interpret why certain next-token candidates are considered at a given step. This feature is intended as a lightweight usability aid (especially for novices) rather than a core algorithmic component, and we do not attribute our main quantitative gains to this explanation module.

3.4 Controller

Prompt-with-Steer provides a set of operators that manipulate the generative trajectory. Formally, we define an intervention operator $\mathcal{I} : \mathbb{S} \times \mathcal{A} \rightarrow \mathbb{S}$ that maps a current state $\mathcal{S}_t$ and a user action $a \in \mathcal{A}$ to a modified state $\mathcal{S}'$.

Prefix Rollback Regeneration (Rewind) When a user identifies an error at step t, simply editing the surface text is insufficient because the erroneous suffix remains in the conditioning context for subsequent re-generation. Prompt-with-Steer therefore performs a rewind operation. We define the action $r(t) = \textsc{Rewind}(t)$ as a truncation of the state history:

$$\mathcal{I}(\mathcal{S}_{t+l}, r(t)) = \mathcal{S}_t = \{\mathbf{K}_i, \mathbf{V}_i\}_{i=1}^{t-1} \tag{7}$$

Conceptually, this reverts the KV cache to its state at time t. In our implementation, we realize this by reconstructing a prompt comprising the preserved prefix text and re-running the generation. This guarantees that the subsequent trajectory is conditioned purely on the validated prefix, mitigating error propagation.

Steering and Resampling Beyond truncation, users can force the model to explore alternative paths. The Steering operator allows the user to select a token y'_t from the top-k candidates (e.g., via the Logit Lens) that aligns with their intent. This trajectory then branches explicitly:

$$\mathcal{S}'_{t+1} = \mathcal{S}_t \cup \{(\mathbf{K}(y'_t), \mathbf{V}(y'_t))\} \tag{8}$$

where $(\mathbf{K}(y'_t), \mathbf{V}(y'_t))$ denote the key and value vectors computed given the forced y'_t. This transforms the generation from a single linear chain

into a user-guided tree search, where the human effectively acts as the value function guiding the policy.

4 Experiments

To validate the effectiveness of Prompt-with-Steer in addressing the correction problem and enhancing user agency, we focus our evaluation on steerability and constraint satisfaction rather than general open-ended generation. In these scenarios, success is objectively verifiable (e.g., "did the model use a forbidden word?"), making them ideal for measuring the precision of human-in-the-loop steering compared to prompting alone.

4.1 Datasets and Experimental Settings

Datasets We utilize the Instruction Following Evaluation (IFEval) dataset (Zhou et al., 2023), a benchmark designed to measure a model's ability to follow verifiable instructions. To conduct a focused study on steerability and provide an appropriate test set for the volunteers, we constructed an enhanced 60-item Hard Test set derived from IFEval, supplemented with custom templates.

The tasks are categorized into two distinct types of hard constraints:

Forbidden Words (Taboo): Negative constraints where the model must generate text without using specific forbidden words. This tests the model's ability to suppress high-probability tokens and the user's ability to repair violations via *prefix rollback regeneration*.

Format(JSON): Structural constraints where the model must generate valid JSON output, often adhering to a specific schema. This tests the model's syntax adherence and allows users to correct structural errors via user's choice (token selection).

Reason Our core goal is to evaluate human steerability and local repair under constraints, which requires an unambiguous success criterion to trigger interventions reliably and attribute improvements to interaction rather than to subjective scoring or model knowledge. We therefore prioritize benchmarks with deterministic, auditable checkers. We note that many real-world user intents are dynamic and not fully formalizable a priori. However, there is currently no widely adopted public dataset that simultaneously provides realistic evolving intent traces and objective, automatic verifiers for end-to-end success.

Taboo and JSON are intentionally chosen because violations are local and verifiable, making them ideal for studying repair operations. We emphasize that these tasks can also be satisfied by automated constrained decoding. Our claim is not that human steering is a replacement for constrained decoding in settings where constraints are fully specified and stable. Instead, we evaluate whether interactive, inspectable local repair provides interaction-efficiency benefits under checker-verified constraints; extending these benefits to partially specified or evolving constraints is left for future work.

Model Configuration Our primary backbone model is Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We also evaluate DeepSeek-Distill-Llama-3-8B (DeepSeek-AI, 2025) and Qwen3-8B (Yang et al., 2025) as additional backbone models to validate the generalizability of our findings. For the Prompt-with-Steer setup, we employ a deterministic checker for each task type. In the real-time simulation, an intervention is triggered immediately by the user when the Checker detects a violation.

Evaluation Protocol We cap the interaction budget at $K = 5$ turns for all methods. For automated baselines (Standard Generation and Reflexion), we use deterministic decoding (greedy generation) and apply the checker after each turn. For human-in-the-loop conditions, we keep the underlying model fixed within a session and apply the same checker-defined success criterion.

4.2 Comparative Methods

We compare the Prompt-with-Steer framework against methods that represent the current state of interaction. To ensure a fair comparison between human-driven and automated approaches, we employed both automated scripts and human volunteers across different settings.

Baselines *Standard Generation (Zero-shot):* The model generates the response in a single pass given the instruction. This serves as the lower bound, establishing the base model's capability to satisfy constraints without any intervention.

Reflexion (Automated Self-Correction): This baseline simulates black-box prompting with an automated feedback loop. When the model violates a constraint (detected by the Checker), the error details are fed back as a new prompt (e.g., *"You failed the constraint: used forbidden word*

Model	Dataset	Method	Pass@1	Pass@2	Pass@K	Mod. SR	Avg. Turns	Avg. Tokens	Overhead	Token Saving
Llama-3.1	Taboo	Reflexion	46.7	**86.7**	90.0	81.2	1.87	526.0	+1.06x	-
		Chat	50.0	83.3	96.7	93.3	1.73	389.4	+0.71x	-
		Steer (*Ours*)	**56.7**	**86.7**	**100.0**	**100.0**	**1.57**	**287.5**	**+0.41x**	**45.3%**
	JSON	Reflexion	76.7	96.7	**100.0**	**100.0**	1.27	82.3	**+0.25x**	-
		Chat	**80.0**	**100.0**	**100.0**	**100.0**	**1.20**	60.8	+0.59x	-
		Steer (*Ours*)	**80.0**	**100.0**	**100.0**	**100.0**	**1.20**	**49.4**	+0.89x	**40.0%**
DeepSeek	Taboo	Reflexion	**33.3**	36.7	46.7	20.0	3.37	3488.3	+3.79x	-
		Chat	26.7	30.0	40.0	18.2	3.70	2413.6	+2.91x	-
		Steer (*Ours*)	26.7	**56.7**	**86.7**	**81.8**	**2.60**	**1031.7**	**+0.56x**	**70.4%**
	JSON	Reflexion	**66.7**	**80.0**	80.0	40.0	**1.93**	916.4	+1.01x	-
		Chat	36.7	56.7	76.7	63.2	2.57	1209.9	+2.19x	-
		Steer (*Ours*)	43.3	**80.0**	**93.3**	**88.2**	**1.93**	**581.8**	**+0.20x**	**36.5%**
Qwen3	Taboo	Reflexion	**50.0**	66.7	80.0	60.0	2.23	2082.7	+1.42x	-
		Chat	40.0	66.7	80.0	66.7	2.37	1612.2	+1.43x	-
		Steer (*Ours*)	43.3	**80.0**	**96.7**	**94.1**	**1.93**	**814.8**	**+0.12x**	**60.9%**
	JSON	Reflexion	96.7	96.7	96.7	96.7	1.13	632.2	+0.09x	-
		Chat	**100.0**	**100.0**	**100.0**	-	**1.00**	**562.4**	**+0.00x**	-
		Steer (*Ours*)	**100.0**	**100.0**	**100.0**	-	**1.00**	568.9	**+0.00x**	**10.0%**

Table 1: Main results on Taboo and JSON tasks. Comparison between Llama-3.1-8B, DeepSeek-Distill-Llama-3-8B, and Qwen3-8B models. Overhead indicates the average ratio of additional tokens consumed for modification relative to the initial generation.

'apple'. Please regenerate."). The model then attempts to self-correct by re-generating the entire response. This baseline isolates the (token) cost of full-response regeneration under a turn budget.

Detailed Prompt (Human-in-the-loop Prompting): This baseline represents real-world human-LLM cooperation via standard dialogue. We recruited volunteers to interact with the models. Unlike the automated Reflexion, human users can provide precise and context-aware instructions to guide the model when it fails. This serves as the strong baseline for current human-AI interaction paradigms.

Our Method *Prompt-with-Steer (Human-in-the-loop Steering):* The same group of volunteers used our interactive interface to perform local repairs. Instead of re-prompting after a failure, users can roll back to a clean prefix (prefix rollback regeneration) and/or select alternative next-token candidates to avoid violating tokens, guiding the model to a valid generation path.

4.3 Metrics

We evaluate performance using following metrics:

Pass@k: The percentage of tasks successfully completed within k turns. We report Pass@1 (immediate success), Pass@2, and Pass@K (success within the maximum allowed turns $K = 5$). Formally, let N be the total number of tasks. For each task i, we define a binary indicator $s_{i,t} \in \{0, 1\}$, where $s_{i,t} = 1$ if the task passes the checker at turn t, and 0 otherwise. Define the cumulative success indicator $c_{i,k} = \max_{1 \le t \le k} s_{i,t}$. The metric is defined as:

$$\text{Pass@}k = \frac{1}{N} \sum_{i=1}^{N} c_{i,k}. \tag{9}$$

Modification Success Rate (Mod. SR): The success rate of the repair process for tasks that fail at the initial generation (turn 1). Let $\mathcal{F} = \{i \mid s_{i,1} = 0\}$. Then

$$\text{Mod.SR} = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} c_{i,K}. \tag{10}$$

Average Turns: The average number of interaction rounds required to satisfy the constraint. A lower value indicates a more efficient human-AI collaboration.

Token Efficiency (Overhead & Token Saving): We report token-based efficiency (not wall-clock compute). Let T_i^{init} be the number of generated tokens at turn 1, and T_i^{total} be the total generated tokens across all turns. We denote the sample mean across the dataset as $\bar{T}$. Overhead is defined as

$$\text{Overhead} = \frac{\bar{T}^{\,\text{total}} - \bar{T}^{\,\text{init}}}{\bar{T}^{\,\text{init}}}. \tag{11}$$

Token Saving is computed relative to the Reflexion baseline on the same model and dataset:

$$\text{Token Saving} = 1 - \frac{\bar{T}^{\,\text{total}}_{\text{method}}}{\bar{T}^{\,\text{total}}_{\text{Reflexion}}}. \tag{12}$$

4.4 Main Results

The results on the enhanced IFEval tasks (Taboo and JSON) are presented in Table 1. Our proposed Prompt-with-Steer framework improves over baselines under the same checker protocol across Llama-3.1, DeepSeek, and Qwen models in both effectiveness and efficiency.

On the Llama-3.1-8B model, Prompt-with-Steer achieves a 100% Pass@K and 100% Modification Success Rate (Mod. SR) on both Taboo and JSON tasks, significantly surpassing Reflexion (Mod. SR 81.2% on Taboo) and Chat baselines. Notably, on the DeepSeek model, which struggles with the Taboo task (Reflexion Pass@K 46.7%), Prompt-with-Steer improves performance to 86.7% Pass@K, indicating benefits even when the base model is weaker. Qwen also sees a substantial improvement in the Taboo task, reaching 96.7% Pass@K with our method.

A key advantage of Prompt-with-Steer is its inspectable local repair mechanism. Unlike Reflexion, which regenerates the entire response, Prompt-with-Steer incurs significantly lower token overhead. On the Taboo task with Llama-3.1, our method achieves a 45.3% Token Saving compared to Reflexion. The efficiency gain is even more pronounced on DeepSeek (70.4% saving) and Qwen (60.9% saving), showing that local steering can be more token-efficient than full-response re-prompting under hard constraints.

4.5 Ablation Study

The main results validate the overall effectiveness of human-in-the-loop local repair, but do not isolate the contribution of diagnostic signals. We therefore conducted an ablation study to test whether exposing signals changes user outcomes and behavior.

We recruited 5 volunteers to complete 30 Taboo items per model under two conditions with the same turn budget $K = 5$: *Steer-lite (Signals Off)* enables local repairs (prefix rollback regeneration / token selection) while hiding diagnostic signals, and *Steer (Signals On)* adds SOS and token-level cues while keeping the interaction capabilities identical. We report Pass@K, average turns, average total completion time (minutes) for the 30-item set, and the average total number of interventions.

In this study, enabling signals is associated with improved success rates and efficiency, particularly for models that struggle with the task. On DeepSeek, Steer increases Pass@K while reducing average completion time and interventions. On Qwen, Steer also slightly improves Pass@K and reduces completion time. On Llama-3.1, both conditions reach 100% Pass@K with similar efficiency, likely due to the model's strong baseline performance.

Model	Cond.	Pass@K	Turns	Time	#Intv.
Llama-3.1	Lite	100.0	1.30	67.5	18.5
	Steer	100.0	1.20	69.6	17.6
DeepSeek	Lite	75.3	2.83	123.4	53.4
	Steer	88.2	2.60	112.6	48.2
Qwen3	Lite	95.2	1.91	86.6	31.2
	Steer	96.3	1.93	80.5	28.3

Table 2: Ablation study on Taboo tasks ($K = 5$) with 5 volunteers. Cond.: Lite = Steer-lite (signals off), Steer = signals on. Time is average total minutes for 30 items; #Intv. is average total interventions.

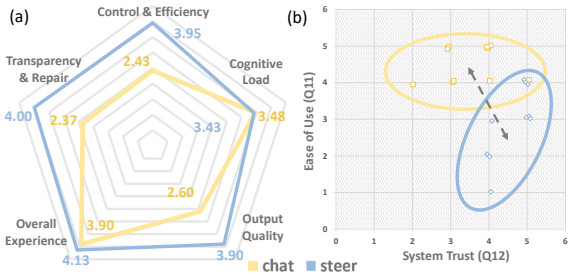

Figure 3: Key Insights from User Study ($N = 10$). The radar chart compares five interaction dimensions, suggesting an advantage for Steer in Control and Transparency within our sample. The scatter plot highlights a trade-off: while Chat is rated higher on Ease of Use, Steer is rated higher on System Trust.

4.6 Results on User Study

We conducted a within-subject survey ($N = 10$) to evaluate the subjective experience of the Prompt-with-Steer method compared to the Chat baseline. This sample size ($N = 10$) is consistent with standard usability testing practices for discovering major interaction issues and trends (Nielsen and Landauer, 1993). Participants rated both modes across five dimensions on a 5-point Likert scale. Given the small sample size, we treat these results as descriptive evidence of usability trends rather than making claims of statistical significance.

Control and Transparency. As shown in Figure 3 (a), the multidimensional ratings suggest an advantage for the Steer mode within our sample. In terms of *Control & Efficiency*, the proposed method achieved a mean score of 3.95 compared to 2.42

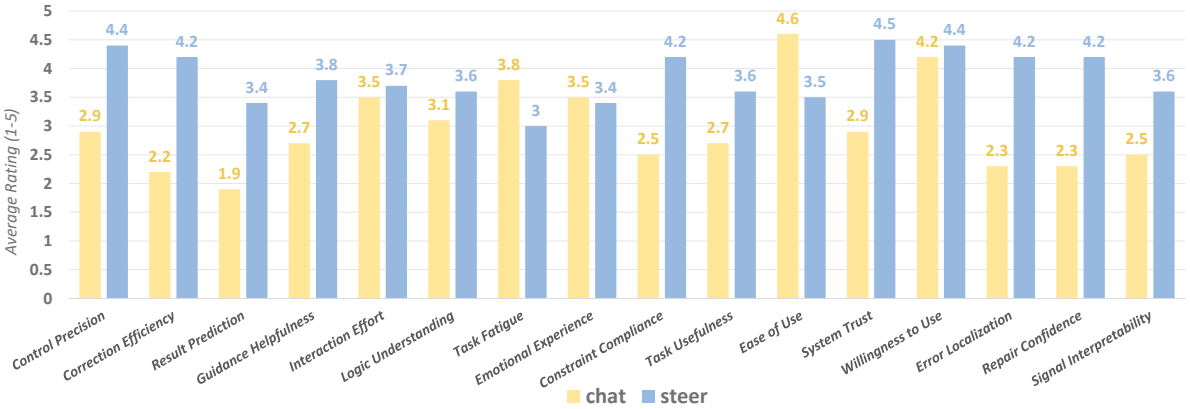

Figure 4: Detailed Subjective Evaluation Results. The bar chart details average Likert ratings for all 16 individual items across five dimensions.

for the baseline. Similarly, *Transparency & Repair* increased from 2.37 to 4.00. Notably, for *Error Localization* (Q14) (see Figure 4), the Steer mode was rated higher by 1.90 points (4.20 vs. 2.30). These gaps are consistent with the intended affordance of the steering interface: making it easier for users to localize and repair errors beyond black-box prompting.

Trade-off: Learning Curve vs. System Trust. As highlighted in Figure 3 (b), our results reveal an important trade-off between ease of use and reliability. The Chat mode was rated higher on *Ease of Use* (4.60 vs. 3.60) and lower on *Task Fatigue*, reflecting the low entry barrier of natural language conversation. However, this increased effort in the Steer mode translated directly into higher *System Trust*, which jumped from 2.90 to 4.50. Likewise, *Constraint Compliance Confidence* saw a marked increase to 4.20 from the baseline of 2.50. This suggests that while active steering requires more initial engagement, users consider this investment worthwhile for the assurance of precise outputs.

Overall Preference. Despite the steeper learning curve, the direct comparison suggests a mixed preference pattern within our sample. On a 7-point scale, participants reported relatively high *Sense of Control* and *Process Fluency*, scoring 5.40 and 5.30 respectively. The *Future Usage Tendency* was 4.70, suggesting that after familiarization, some participants may value the added agency and predictability of the "prompt-and-steer" paradigm over trial-and-error prompting.

5 Conclusion

In this paper, we propose Prompt-with-Steer, an interactive framework that enables fine-grained, process-driven steerability for human–LLM interaction. Building upon conventional prompt-based interaction, Prompt-with-Steer leverages the autoregressive next-token prediction mechanism of large language models to introduce click-and-choose interventions at critical generation points, allowing users to precisely guide both generation and regeneration through lightweight steering prompts. Importantly, Prompt-with-Steer requires no modification to model architectures or parameters, and operates solely by exploiting the intrinsic generation dynamics of autoregressive LLMs, making it readily applicable to existing models. Extensive experiments demonstrate that Prompt-with-Steer enables users to achieve more accurate edits on verifiable lexical and syntactic constraint tasks with substantially fewer interaction rounds and reduced token consumption. Furthermore, our ablation analysis suggests that exposing diagnostic signals—such as the Steering Opportunity Score (SOS) and token-level cues—helps users identify effective intervention points rather than steering blindly. In our pilot user study ($N = 10$), while traditional chat is perceived as easier to use, Prompt-with-Steer shows descriptive trends of higher perceived control and system trust. We observe these trends more clearly with self-prompting LLMs in our setting, suggesting potential benefits for controllability in practice. Future work will explore scaling this approach to more open-ended, dynamic intent scenarios.

Limitations

Our study focuses on verifiable hard constraints (Taboo/JSON) to precisely measure interaction benefits, a trade-off that may under-represent dynamic, partially formalizable user intents. While static constraints can often be handled by constrained decoding, our goal is to study inspectable, localized repair for scenarios where constraints evolve or are negotiated. Consequently, our findings highlight the benefits of interactive repair but may not fully apply to settings where constraints are fully pre-specified.

Our evaluation relies on automatic verifiers and controlled tasks, which may not fully transfer to open-ended real-world writing with subjective success criteria. Verifier-defined success can incentivize shortcut behaviors, and iterative local repairs may introduce drift in long-form outputs. Additionally, our approach is primarily designed for local syntactic repairs; for global semantic errors or high-level stylistic changes requiring large-scale rewriting, token-level steering may be less efficient than re-prompting.

While Prompt-with-Steer improves token efficiency, it imposes a higher cognitive load than conversational interfaces, requiring users to interpret diagnostic cues. This makes it currently best suited for expert users or high-stakes scenarios where precision is paramount, rather than casual interactions. Future work should focus on abstracting these technical details to lower the barrier for general users.

From a systems perspective, our framework requires model-specific integration and incurs computational overhead for diagnostic computation, potentially affecting latency for large models. Furthermore, steering behaviors may vary across architectures and decoding settings. Future research should validate these benefits with larger-scale studies and develop benchmarks that better capture evolving intents in realistic editing workflows.

AI Assistance We used GitHub Copilot for coding support and Grammarly for grammatical error correction and writing.

Ethical Considerations

This work involved a low-risk controlled user study in which volunteers interacted with the system on predefined Taboo and JSON tasks with explicit success criteria. The study protocol was approved by an Institutional Review Board (IRB); details are omitted for double-blind review. The study was classified as low-risk and exempt from full committee review.

Participants were recruited via convenience sampling. All participants provided written informed consent acknowledging their understanding of the task, the data collected (e.g., interaction logs necessary for analysis), and their right to withdraw at any time without consequence. Participants received a small non-monetary token of appreciation. All collected data was anonymized by removing user identifiers. We manually reviewed the task templates and collected data to ensure they contain no personally identifying information or offensive content.

Potential Risks While Prompt-with-Steer empowers users to align model outputs with their intents, we acknowledge that fine-grained control mechanisms could potentially be misused to bypass safety guardrails or steer models toward generating harmful content. However, our framework operates on top of the base model's probability distribution; it does not modify the model weights or remove safety training. We believe that as user control increases, safety checks should be integrated dynamically into the interaction loop rather than relying solely on static post-generation filtering.

References

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA. Association for Computing Machinery.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Kevin Bowden and Marilyn Walker. 2025. Dialogue scaffolding: Producing a realistic corpus of human-computer open-domain dialogues using a spoken dialogue system and ChatGPT. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 538–560, Avignon, France. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. 2025. How people use chatgpt. Working Paper 34255, National Bureau of Economic Research.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. *Preprint*, arXiv:2309.03883.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *Preprint*, arXiv:2208.14271.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Christy Doran and Deborah A. Dahl. 2024. It's not under the lamppost: Expanding the reach of conversational AI. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9441–9451, Torino, Italia. ELRA and ICCL.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2024. Uncovering the potential of ChatGPT for discourse analysis in dialogue: An empirical study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16998–17010, Torino, Italia. ELRA and ICCL.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. *Preprint*, arXiv:2401.06102.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

John Hewitt, Nelson F. Liu, Percy Liang, and Christopher D. Manning. 2024. Instruction following without instruction tuning. *Preprint*, arXiv:2409.14254.

Jiaming Ji, Kaile Wang, Tianyi Alex Qiu, Boyuan Chen, Jiayi Zhou, Changye Li, Hantao Lou, Josef Dai, Yunhuai Liu, and Yaodong Yang. 2025. Language models resist alignment: Evidence from data compression. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23411–23432, Vienna, Austria. Association for Computational Linguistics.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573, Abu Dhabi, UAE. Association for Computational Linguistics.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of LLMs with mis-aligned responses. 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 11943–11974, Albuquerque, New Mexico. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Preprint*, arXiv:2306.03341.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. *Preprint*, arXiv:2210.15097.

Zongqian Li, Ehsan Shareghi, and Nigel Collier. 2025. ReasonGraph: Visualization of reasoning methods and extended inference paths. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 140–147, Vienna, Austria. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, prompt, and recommendation: A comprehensive survey of language modeling paradigm adaptations in recommender systems. *Transactions of the Association for Computational Linguistics*, 11:1553–1571.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. 2024b. Fantastic semantics and where to find them: Investigating which layers of generative LLMs reflect lexical semantics. In *Findings of the Association*

for Computational Linguistics: ACL 2024, pages 14551–14558, Bangkok, Thailand. Association for Computational Linguistics.

Jakob Nielsen and Thomas K. Landauer. 1993. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 206–213, New York, NY, USA. Association for Computing Machinery.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Kaviraj Pather, Elena Hadjigeorgiou, Arben Krasniqi, Claire Schmit, Irina Rusu, Marc Pons, and Kabir Khan. 2025. Vis-cot: A human-in-the-loop framework for interactive visualization and intervention in llm chain-of-thought reasoning. *Preprint*, arXiv:2509.01412.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. 2024. Learning or self-aligning? rethinking instruction fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6090–6105, Bangkok, Thailand. Association for Computational Linguistics.

Clemencia Siro, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Rethinking the evaluation of dialogue systems: Effects of user feedback on crowdworkers and llms. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 1952–1962. ACM.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Parrot: Enhancing multi-turn instruction following for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9729–9750, Bangkok, Thailand. Association for Computational Linguistics.

Chupei Wang and Jiaqiu Vince Sun. 2025. Unable to forget: Proactive interference reveals working memory limits in llms beyond context length. *Preprint*, arXiv:2506.08184.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *Preprint*, arXiv:2501.07301.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. 2022. Monte carlo tree search: a review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562.

A Appendix

To ensure the reproducibility and auditability of our results, we provide a detailed mapping of artifacts, execution protocols, and a qualitative failure analysis.

A.1 Code and Data Artifacts

For double-blind review, we do not include identifying URLs or repositories. We will release the full source code, datasets, and checker implementations upon acceptance. The codebase components are detailed in Table 3. Detailed usage instructions and file mappings will be provided in an accompanying `README.md`.

Artifact	Description
Task Sets	Curated subset of IFEval and custom hard constraint tasks.
Checkers	Deterministic scripts for validating Taboo and JSON constraints.
Baselines	Implementations of the Standard and Reflexion loops.
Interface	Streamlit-based UI used for human-in-the-loop experiments.

Table 3: Overview of released code and data artifacts.

A.2 Hyperparameters and Configuration

We employ a consistent set of generation hyperparameters across Llama-3, DeepSeek, and Qwen models for the human-in-the-loop interaction conditions (Chat / Steer), as detailed in Table 4.

Hyperparameter	Value
Temperature	0.4
Top-p	0.7
Top-k	40
Repetition Penalty	1.15
Max New Tokens	1024

Table 4: Generation hyperparameters used across all models.

For automated baseline evaluations (Standard Generation / Reflexion), we use deterministic decoding (greedy generation with `do_sample=False`, `num_beams=1`). In this setting, sampling-related hyperparameters such as Temperature, Top-p, and Top-k are not applied.

For the internal analysis, we compute KL divergence using intermediate layer representations. Motivated by prior analyses of layerwise representations (Liu et al., 2024b; Jin et al., 2025), unless otherwise specified we capture hidden states at a mid-layer ($l = 15$) for 8B models to analyze mid-generation uncertainty.

A.3 User Study Instructions

Participants were provided with the following instructions before starting the session:

> *"You will be presented with a series of 60 tasks involving specific constraints (e.g., avoiding certain words or formatting output as JSON). Your goal is to solve these problems as accurately and efficiently as possible. You are free to use the provided tools—either by writing detailed prompts or by using the 'Prompt-with-Steer' interface to intervene during generation. Please explore the system freely; the developers will not intervene in your process. Focus on achieving the correct result in the shortest time."*

Participants were also given a brief 5-minute tutorial on how to use the steering interface (rollback and token selection) prior to the data collection phase.

A.4 Computational Environment

All experiments were conducted on a workstation equipped with an NVIDIA RTX PRO 6000. The software stack includes PyTorch 2.1.2 and Transformers 4.36.2. We evaluated the following model versions: Llama-3.1-8B-Instruct, DeepSeek-R1-Distill-Llama-8B, and Qwen3-8B.

A.5 Experimental Protocol and Data Schema

We strictly enforce a turn budget of $K = 5$ across all experimental conditions.

Automated Conditions (Reflexion). Evaluations are executed via `run_evaluation.py`. The script performs greedy decoding, deterministic checking, and feedback looping. It automatically records metrics in `.csv` format and execution traces in `.log` format.

Human-in-the-Loop (Chat / Steer). Volunteers interact with the models via a Streamlit application. To minimize cognitive load, the UI calculates real-time metrics (e.g., Total Tokens, Correction Turns) and displays them in a sidebar. Participants transcribe these metrics into a structured spreadsheet. The UI also supports a "Download Logs" feature for timestamped JSON traces.

Data Schema for Audit. The recorded CSV data contains the fields detailed in Table 5, which are used to compute the main results.

Field	Description
`Init Tokens`	Tokens generated in the first turn.
`Total Token`	Cumulative tokens across all turns.
`Total Turn`	Turn number of success (or 5 if failed).
`Corr. Tokens`	Tokens consumed during correction (Steer only).

Table 5: Data schema for audit logs.

A.6 Failure Analysis (Case Studies)

We analyze typical failure modes to contextualize the quantitative results presented in the main text.

Case 1: Taboo Word Collisions (Reflexion Failure). *Task:* `ifeval_taboo_2395_hard` (Constraints: No 'flea', 'json', 'in', 'of').
Observation: Reflexion often fails because regenerating the entire response to fix a specific violation (e.g., 'flea') inadvertently reintroduces high-probability function words like 'in' or 'of'.

Case 2: Structural Fragility (Chat Failure). *Task:* `curated_json_013_hard` (Constraint: Single-line JSON).
Observation: Chat models frequently include polite conversational fillers or markdown code fences, violating the strict "single-line" constraint.

Case 3: Local Repair (Steer Success). *Observation:* Steer allows the user to surgically regenerate only the violating segment (e.g., a specific word or newline), preserving the correct context and avoiding the reintroduction of errors.

B Dataset Examples

This appendix lists the tasks used in our evaluation, including their prompts and hard constraints.

B.1 Taboo Tasks

ifeval_taboo_2143_hard **TABOO**

Prompt:
"I'm sorry to inform you that I can't make it to the meeting today. I apologize for any inconvenience this may cause." Please expand it into at least 5 sentences. Do not use the words reschedule or free.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'for', 'a'.

Constraints:
Forbidden Words: 'reschedule', 'free', 'for', 'a'

ifeval_taboo_2395_hard **TABOO**

Prompt:
Write a facebook post about a flea market in JSON format. Do not include keywords 'flea' and 'JSON' in the response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'in', 'of'.

Constraints:
Forbidden Words: 'flea', 'json', 'in', 'of'

ifeval_taboo_1691_hard **TABOO**

Prompt:
Request:
1. What are the best places to visit in Bohemia, Czech Republic?
2. Include a list of recommended hotels.
3. Wrap the ENTIRE output in JSON format.
4. Do not include the following keywords: Moser, Glassworks, Pravcice, Karlovy, Vary
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that'.

Constraints:
Forbidden Words: 'moser', 'glassworks', 'pravcice', 'karlovy', 'vary', 'that'

ifeval_taboo_2577_hard **TABOO**

Prompt:
What do prehistoric megaliths in Europe look like? Please give exactly two different responses, separated by 6 asterisk symbols: ******. Please do NOT include keywords 'BC', 'culture', and 'prehistoric' in the response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'and', 'the'.

Constraints:
Forbidden Words: 'BC', 'culture', 'prehistoric', 'and', 'the'

ifeval_taboo_1147_hard **TABOO**

Prompt:
Rewrite the following statement to make it sound more formal, like a President of the United States:
"Hi guys. The work was done to add in a fix for the issue that was observed in the field with the SSO. We are working with our collaborators closely. We will get it done. Thanks ya all."
Do not include the following keywords: field, thanks, issue, collaborator.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'is'.

Constraints:
Forbidden Words: 'field', 'thanks', 'issue', 'collaborator', 'is'

ifeval_taboo_1629_hard **TABOO**

Prompt:
Make the sentence "The bus arrived at the station" sound more interesting. Avoid using the word "station".
[HARD CONSTRAINT] You must also AVOID using the following common words: 'a'.

Constraints:
Forbidden Words: 'station', 'a'

ifeval_taboo_1580_hard **TABOO**

Prompt:
Write a story for kids about how a person learns to ride a bike. Do not include 'can' and 'ride' in your response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'a'.

Constraints:
Forbidden Words: 'can', 'ride', 'a'

ifeval_taboo_2207_hard TABOO

Prompt:
Here is the summary of a research paper on the effect of VHF radio waves on a certain type of bacteria: "Our results show that VHF radio waves have no effect on the growth of bacteria." Can you help me rewrite this summary in a more formal way, using APA format? Do not use words "ours" or "have".
[HARD CONSTRAINT] You must also AVOID using the following common words: 'for', 'it'.

Constraints:
Forbidden Words: 'ours', 'have', 'for', 'it'

ifeval_taboo_371_hard TABOO

Prompt:
Write a riddle that describes the word "key" but doesn't use the word "key". Wrap all words into one JSON block. The word "key" should not appear in your entire reply.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that', 'the'.

Constraints:
Forbidden Words: 'key', 'that', 'the'

curated_taboo_004_hard TABOO

Prompt:
Write a short email to reschedule a meeting. Do not use the words meeting or schedule.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that'.

Constraints:
Forbidden Words: 'meeting', 'schedule', 'that'

ifeval_taboo_3506_hard TABOO

Prompt:
Write a proposal for a research project that will determine whether pupils who have been exposed to a fast-paced environment are more likely to develop ADHD. Please wrap the entire output in JSON format. You can use markdown ticks like
"'JSON
[json content]
"'
Please do not include the words "proposal" or "project" in the response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'in', 'and'.

Constraints:
Forbidden Words: 'proposal', 'project', 'in', 'and'

ifeval_taboo_2471_hard TABOO

Prompt:
Write a college academic paper about President of the United States being stressed. Make sure not to include negative words such as 'sad', 'crazy', 'stress', etc., in the response. Also, make sure to include at least 15 placeholders represented by square brackets, such as [address].
[HARD CONSTRAINT] You must also AVOID using the following common words: 'a', 'is'.

Constraints:
Forbidden Words: 'sad', 'crazy', 'stress', 'a', 'is'

curated_taboo_001_hard TABOO

Prompt:
Write a short product description for a new note-taking app. Do not use the words notes or app.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'the', 'to'.

Constraints:
Forbidden Words: 'notes', 'app', 'the', 'to'

ifeval_taboo_301_hard TABOO

Prompt:
Explain to me how to ride a bike like I am a kid. Also, do not include the keywords "slow", "like" and "kid".
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that'.

Constraints:
Forbidden Words: 'slow', 'like', 'kid', 'that'

ifeval_taboo_2328_hard TABOO

Prompt:
Write a startup pitch for a time capsule service. The words startup and capsule cannot be in the response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'to'.

Constraints:
Forbidden Words: 'startup', 'capsule', 'to'

ifeval_taboo_1675_hard TABOO

Prompt:
Can you provide a translation for "今天天气很好" in German? Do not use "heute". Please use another word.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'is'.

Constraints:
Forbidden Words: 'heute', 'is'

ifeval_taboo_1132_hard TABOO

Prompt:
Write the lyrics to a hit song by the rock band 'The Gifted and The Not Gifted'. To make it rocky, the response should be in all capital letters. The word "rock" should not appear in your response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'is', 'that'.

Constraints:
Forbidden Words: 'rock', 'is', 'that'

ifeval_taboo_3114_hard TABOO

Prompt:
You visited a beach and a park. Which one is more likely to have a dog in it? Write at least 900 words. Do not include the words 'bark' or 'run' in your answer. Finish your response with the phrase "Does this make sense?"
[HARD CONSTRAINT] You must also AVOID using the following common words: 'for'.

Constraints:
Forbidden Words: 'bark', 'run', 'for'

ifeval_taboo_127_hard TABOO

Prompt:
Take the text below as a starting point, and make it a complete article: "You may have to meet with a helper to work out a parenting plan. The first would be to estimate how many time you have everyday for parenting, and is that enough...."
Avoid using the following keywords: sleep, cook, feed
Mention the keyword 'schedule' for more than 5 times.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'it', 'the'.

Constraints:
Forbidden Words: 'sleep', 'cook', 'feed', 'it', 'the'

ifeval_taboo_1137_hard TABOO

Prompt:
Explain in French why it is important to eat healthy foods to heal the body, without using the word "nourriture". Make sure your entire response is wrapped in JSON format.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'in'.

Constraints:
Forbidden Words: 'nourriture', 'in'

ifeval_taboo_2957_hard TABOO

Prompt:
Rewrite the limerick in a strange way. In particular, the limerick is about nursery and storytelling. But do not mention nursery and storytelling in your entire response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'of', 'the'.

Constraints:
Forbidden Words: 'nursery', 'storytelling', 'of', 'the'

ifeval_taboo_2028_hard TABOO

Prompt:
Are the weather conditions in the Arctic very cold most of the year? Do not say 'yes' or 'no' throughout your entire response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'for'.

Constraints:
Forbidden Words: 'yes', 'no', 'for'

ifeval_taboo_3445_hard TABOO

Prompt:
How to tell others that your major is computer science, without saying the word computer or science? You entire response should not contain the word computer and science.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'and'.

Constraints:
Forbidden Words: 'computer', 'science', 'and'

ifeval_taboo_3371_hard TABOO

Prompt:
Is the following true? "People use time to buy money." Explain your answer in a way that makes it easy for kids to understand. Include the word "farmer". Exclude the words "economy", "demand" and "supply".
Please do not say any words or characters before repeating the request.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'and', 'that'.

Constraints:
Forbidden Words: 'economy', 'demand', 'supply', 'and', 'that'

ifeval_taboo_1402_hard TABOO

Prompt:
I'm interested in a college with open enrollment and a regional accreditation. Which college would you recommend? Don't include the keywords "DuPage" and "Dade" in your response. Let's make it a constrained writing problem: be sure the letter p appears at least 15 times in your response.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'is'.

Constraints:
Forbidden Words: 'dupage', 'dade', 'is'

ifeval_taboo_3081_hard TABOO

Prompt:
Can you re-create a story from a fictional newspaper with title: "A man mysteriously died in his house, and police are investigating"? Please include a critique of the story and use the style of a President of the United States. Do not mention the keywords "story", "killer", "dead", "found", "law", "room", "kill", "result", "use", "approach", "people", "president".
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that', 'is'.

Constraints:
Forbidden Words: 'story', 'killer', 'dead', 'found', 'law', 'room', 'kill', 'result', 'use', 'approach', 'people', 'president', 'that', 'is'

ifeval_taboo_3315_hard TABOO

Prompt:
Students are travelling to a university campus for 3 days. Create a hilarious itinerary for them. Do not use the word 'university'. Your entire response should have exactly 4 paragraphs. Separate paragraphs with the markdown divider: ***
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that', 'a'.

Constraints:
Forbidden Words: 'university', 'that', 'a'

curated_taboo_006_hard TABOO

Prompt:
Write a friendly welcome message for a community forum. Do not use the words welcome or community.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'that'.

Constraints:
Forbidden Words: 'welcome', 'community', 'that'

ifeval_taboo_3166_hard TABOO

Prompt:
What are the steps to be followed for the documentation of a GM in SAP? Just list the steps without saying the word steps or step.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'to'.

Constraints:
Forbidden Words: 'steps', 'step', 'to'

ifeval_taboo_1466_hard TABOO

Prompt:
Write a song that critiques the song "We Are Never Ever Getting Back Together" by Taylor Swift. Wrap your entire response with double quotation marks. Do not mention the word Taylor, Swift, or Together.
[HARD CONSTRAINT] You must also AVOID using the following common words: 'in', 'it'.

Constraints:
Forbidden Words: 'taylor', 'swift', 'together', 'in', 'it'

B.2 JSON Tasks

curated_json_013_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: steps, duration_minutes. Do not include any extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `steps`: `list[string]`, `duration_minutes`: `int`

curated_json_021_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: title, summary, tags. Do not include any extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

ifeval_json_2395_hard — JSON

Prompt:
Write a facebook post about a flea market in JSON format. Do not include keywords 'flea' and 'JSON' in the response.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

curated_json_016_hard — JSON

Prompt:
Respond in strict JSON with keys: title, summary, tags. No code fences.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

curated_json_029_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: question, answer, confidence. Do not include any extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `question`: `string`, `answer`: `string`, `confidence`: `float`

ifeval_json_13_hard — JSON

Prompt:
What is the history of NYC prospect park? Please wrap your entire answer in JSON format. You can use markdown ticks such as "'. For example:
"'JSON
...
"'
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

curated_json_033_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: steps, duration_minutes. Do not include any extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `steps`: `list[string]`, `duration_minutes`: `int`

curated_json_039_hard — JSON

Prompt:
Provide a JSON object with exactly these keys: question, answer, confidence. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `question`: `string`, `answer`: `string`, `confidence`: `float`

curated_json_002_hard — JSON

Prompt:
Output must be strictly valid JSON (no markdown). Use keys: name, age, city. No extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `name`: `string`, `age`: `int`, `city`: `string`

curated_json_035_hard — JSON

Prompt:
Provide a JSON object with exactly these keys: items, priority. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

curated_json_015_hard — JSON

Prompt:
Provide a JSON object with exactly these keys: items, priority. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

curated_json_005_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: items, priority. Do not include any extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:

Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

curated_json_003_hard JSON

Prompt:
Provide a JSON object with exactly these keys: steps, duration_minutes. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `steps`: `list[string]`, `duration_minutes`: `int`

ifeval_json_1094_hard JSON

Prompt:
Please provide the names of 5 famous moms in JSON format. Please, use any interesting or weird tone. Your entire output should just contain a JSON block, nothing else.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

ifeval_json_3223_hard JSON

Prompt:
List all facts about Lionel Messi in a structured output. In particular, Format your entire output in JSON.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

curated_json_042_hard JSON

Prompt:
Output must be strictly valid JSON (no markdown). Use keys: name, age, city. No extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `name`: `string`, `age`: `int`, `city`: `string`

curated_json_019_hard JSON

Prompt:
Provide a JSON object with exactly these keys: question, answer, confidence. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `question`: `string`, `answer`: `string`, `confidence`: `float`

curated_json_010_hard JSON

Prompt:
Output must be strictly valid JSON (no markdown). Use keys: items, priority. No extra keys.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

ifeval_json_371_hard JSON

Prompt:
Write a riddle that describes the word "key" but doesn't use the word "key". Wrap all words into one JSON block. The word "key" should not appear in your entire reply.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

curated_json_012_hard JSON

Prompt:
Respond in strict JSON with keys: name, age, city. No code fences.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `name`: `string`, `age`: `int`, `city`: `string`

curated_json_036_hard JSON

Prompt:
Respond in strict JSON with keys: title, summary, tags. No code fences.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

ifeval_json_1148_hard JSON

Prompt:
What are the advantages and disadvantages of having supernatural powers? Make it short. Wrap the entire output in JSON format. You can use markdown ticks such as "'.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Single Line: Yes

curated_json_040_hard JSON

Prompt:
Respond in strict JSON with keys: items, priority. No code fences.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

curated_json_031_hard JSON

Prompt:
Provide a JSON object with exactly these keys: title, summary, tags. No additional text.
[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

curated_json_025_hard JSON

Prompt:

Return ONLY a valid JSON object with keys: items, priority. Do not include any extra keys.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `items`: `list[string]`, `priority`: `string`

curated_json_007_hard — JSON

Prompt:
Provide a JSON object with exactly these keys: name, age, city. No additional text.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `name`: `string`, `age`: `int`, `city`: `string`

curated_json_008_hard — JSON

Prompt:
Respond in strict JSON with keys: steps, duration_minutes. No code fences.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `steps`: `list[string]`, `duration_minutes`: `int`

curated_json_001_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: title, summary, tags. Do not include any extra keys.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

curated_json_032_hard — JSON

Prompt:
Respond in strict JSON with keys: name, age, city. No code fences.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `name`: `string`, `age`: `int`, `city`: `string`

curated_json_041_hard — JSON

Prompt:
Return ONLY a valid JSON object with keys: title, summary, tags. Do not include any extra keys.

[HARD CONSTRAINT] The JSON must be compacted into a SINGLE line. Do not use any newlines.

Constraints:
Strict JSON: Yes *Single Line:* Yes
Schema Keys: `title`: `string`, `summary`: `string`, `tags`: `list[string]`

C User Experience Survey: Prompt-with-Steer vs. Chat

This section presents the full User Experience Survey used in our study. The detailed questionnaire items and rating scales are shown in Figure 5 and Figure 6 on the following pages.

1 User Experience Survey: Prompt-with-Steer vs. Chat

1.1 Instructions

Participants are asked to evaluate two interaction modes based on their experience:

- **Mode A: Traditional Chat** (Conversational free interaction)
- **Mode B: Prompt-with-Steer** (Controllable segmented generation)

Please rate each mode on a 5-point Likert scale *(1 = Strongly Disagree/Poor, 5 = Strongly Agree/Excellent)*. Unless otherwise specified, please answer based on your experience **in the tasks of this study** (Taboo forbidden words / JSON formatting) rather than general long-term use.

1.2 Detailed Evaluation

Table 1: User Experience Questionnaire Items. Participants rated both modes side-by-side.

ID	Assessment Item & Anchors	Chat Mode					Steer Mode				
Dimension 1: Control & Efficiency											
1	**Control Precision** I can precisely control the model to output what I want. (1: Hard to control ↔ 5: Precise control)	1	2	3	4	5	1	2	3	4	5
2	**Correction Efficiency** When I find an error, I can fix it quickly. (1: Hard to fix ↔ 5: Quick fix)	1	2	3	4	5	1	2	3	4	5
3	**Result Prediction** I can predict the model's next output. (1: Unexpected ↔ 5: As expected)	1	2	3	4	5	1	2	3	4	5
4	**Guidance Helpfulness** The system provided helpful signals to guide my actions. (1: Useless/None ↔ 5: Very helpful)	1	2	3	4	5	1	2	3	4	5
Dimension 2: Cognitive Load											
5	**Interaction Effort** Figuring out what to do next to fix an error required little trial-and-error. (1: Lots of trial-and-error ↔ 5: Very straightforward)	1	2	3	4	5	1	2	3	4	5
6	**Logic Understanding** I can understand the model's generation logic. (1: Black box ↔ 5: Clear logic)	1	2	3	4	5	1	2	3	4	5
7	**Task Fatigue** After completing the tasks in this mode, I felt mentally relaxed (not exhausted). (1: Exhausted ↔ 5: Relaxed)	1	2	3	4	5	1	2	3	4	5
8	**Emotional Experience** I felt calm and not frustrated during the task. (1: Frustrated ↔ 5: Calm/Positive)	1	2	3	4	5	1	2	3	4	5
Dimension 3: Output Quality											
9	**Perceived Constraint Compliance** I am confident the output complied with the task constraints (format / forbidden words). (1: Not confident ↔ 5: Very confident)	1	2	3	4	5	1	2	3	4	5

Figure 5: User Experience Survey (Page 1)

ID	Assessment Item & Anchors	Chat Mode					Steer Mode				
10	**Task Usefulness** The output was useful for completing the task (beyond merely passing the checker). (1: Not useful ↔ 5: Very useful)	1	2	3	4	5	1	2	3	4	5
Dimension 4: Overall Experience											
11	**Ease of Use** The system is easy to learn and use. (1: Hard to learn ↔ 5: Easy to use)	1	2	3	4	5	1	2	3	4	5
12	**System Trust** I trust the system's output to be correct. (1: Distrust ↔ 5: Fully trust)	1	2	3	4	5	1	2	3	4	5
13	**Willingness to Use** I am willing to use this interaction mode long-term. (1: Not willing ↔ 5: Very willing)	1	2	3	4	5	1	2	3	4	5
Dimension 5: Transparency & Repair											
14	**Error Localization** When something went wrong, I could quickly locate *where* the error started (which token/segment). (1: Hard to locate ↔ 5: Easy to locate)	1	2	3	4	5	1	2	3	4	5
15	**Repair Confidence** I felt confident that my intervention (editing / rollback / token choice) would improve the outcome. (1: Not confident ↔ 5: Very confident)	1	2	3	4	5	1	2	3	4	5
16	**Signal Interpretability** I could understand what the system signals meant (e.g., uncertainty/diagnostic indicators) and how to act on them. (1: Hard to understand ↔ 5: Easy to understand)	1	2	3	4	5	1	2	3	4	5

1.3 Direct Comparison

Which mode do you prefer?
(7-point scale; 1 = Strongly Prefer Chat, 4 = Neutral, 7 = Strongly Prefer Steer)

Comparison Aspect	Preference (Chat ↔ Steer)						
1. Effectiveness of task completion	1	2	3	**4**	5	6	7
2. Fluency of the process	1	2	3	**4**	5	6	7
3. Sense of control over output	1	2	3	**4**	5	6	7
4. Overall satisfaction	1	2	3	**4**	5	6	7
5. Future usage tendency	1	2	3	**4**	5	6	7

1.4 Open-ended Feedback

- **Pros of Steer Mode:** _____
- **Cons of Steer Mode:** _____
- **Suggestions:** _____

Figure 6: User Experience Survey (Page 2)