MinSteer: Minimal-Pair Steering via Two-Stage Cached Continuation

Muxuan Liu[*,‡], **Tatsuya Ishigaki**[‡], **Yusuke Miyao**[†,‡],
Hiroya Takamura[‡], **Ichiro Kobayashi**[*,‡]
[*]Ochanomizu University [†]The University of Tokyo
[‡]National Institute of Advanced Industrial Science and Technology
{liu.muxuan,koba}@is.ocha.ac.jp yusuke@is.s.u-tokyo.ac.jp
{ishigaki.tatsuya,takamura.hiroya}@aist.go.jp

Abstract

Do behavioral styles in large language models correspond to reusable directions in residual space, or do they mainly appear after averaging many noisy contrastive examples? We introduce **MinSteer**, a *two-stage cached-continuation* method for estimating a contrastive residual direction (CRD) within one decoding trajectory. Stage X generates a continuation in one style; a style-flip suffix is then processed using Stage X's final KV cache; Stage Y continues from this inherited trajectory. The CRD is computed from residuals recorded only during generation, excluding the prompt and suffix prefill passes. Across sentiment transfer, politeness control, and a ParaDetox-based bidirectional diagnostic, MinSteer extracts effective steering directions from very small contrasts. On TweetEval sentiment transfer, one MinSteer contrast outperforms a CAA baseline averaged over 50 independent pairs. Japanese keigo (honorific language) CRDs transfer to English and Chinese outputs and outperform trajectory-free **TwoSeq** controls. In ParaDetox, MinSteer produces bidirectional Detoxify movement under our rewrite setup, while exposing semantic-preservation trade-offs. The directions transfer across tested scenarios and languages and show clear depth dependence. Together, these results show that cached self-rewrite trajectories can provide stronger steering signals than independent-pair averaging.

1 Introduction

A central question in LLM interpretation is whether behavioral styles such as sentiment, register, and toxicity are encoded in identifiable residual-space directions, or whether they are diffuse patterns that only emerge from large-scale averaging. Contrasts built from independently sampled positive and negative examples can mix behavioral differences with unrelated content differences. A direction extracted from 100 independent positive/negative pairs is an average over both signal and noise; we cannot tell whether $n \geq 50$ is needed because behavioral directions are intrinsically weak, or simply because the noise floor is high.

We address this by designing **MinSteer**[1], a controlled extraction probe that keeps the behavioral contrast *inside one cached decoding trajectory*. MinSteer first generates a continuation in one style (Stage X). It then appends a style-flip instruction as incremental tokens processed with Stage X's final KV cache (*cache inheritance*) and generates a second continuation (Stage Y). The CRD is the difference between mean residual states recorded during the generation of Stage Y and Stage X; prompt and suffix prefill passes are excluded from these averages. This is a sequential rewrite protocol: Stage Y is conditioned on the concrete content produced in Stage X, not on a symmetric pair of

[1] https://github.com/Liumx2020/MinSteer

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

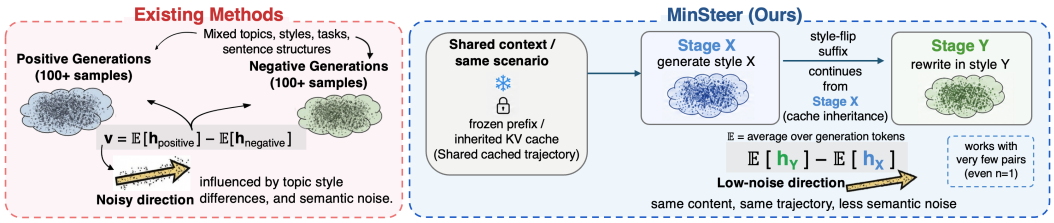


Figure 1: Comparison between class-mean contrastive steering and MinSteer. The left panel illustrates contrasts between independently sampled positive and negative examples. Means are taken over residual states recorded during generation. Self-rewriting is designed to reduce content-related variation.

branches from an identical prefix state. We call the resulting vector a *contrastive residual direction* (CRD). It may reflect both the intended style shift and rewrite-transform components induced by the follow-up instruction. Content-controlled comparisons separate transcript construction from cache execution, and transfer tests probe whether CRDs remain useful outside the extraction template (Section 5.6, Appendix P). This design makes MinSteer useful not only as a practical steering method, but primarily as a controlled instrument for probing behavioral structure in LLM residual space. Figure 1 illustrates the key difference between class-mean averaging and our two-stage cached continuation approach.

This setup lets us ask three questions about LLM internal structure. First, how much data is needed? If $n = 1$ suffices, large unaligned averages may be compensating for extraction noise rather than weak behavioral signal. Second, do the directions transfer? Cross-scenario and cross-lingual tests distinguish model-internal regularities from prompt-surface artifacts. Third, where do they act, and what are their limits? Layer localization, depth dependence, and safety-related interventions show when the direction remains useful.

To answer these questions, we apply the same extraction procedure to three settings: sentiment transfer, politeness control, and a toxicity evaluation on ParaDetox-style rewriting. In sentiment, the external **CAA baseline** remains below MinSteer at $n = 1$ even after averaging 50 independent pairs. In politeness, we pair MinSteer with **TwoSeq** trajectory-drop controls on the same scenarios (Table 2). In toxicity, TwoSeq is reported as a trajectory-free baseline: **TwoSeq-same** pairs both arms on the same toxic line, providing a fairer test than independent prompts. A harsher **TwoSeq-cross** diagnostic is discussed in the appendix for completeness (Table 16). The main table (Table 7) compares MinSteer with the task-specific TwoSeq controls across all three tasks.

Our contributions are:

- **We introduce MinSteer, a two-stage cached-continuation method for extracting steering directions without large human-written paired datasets.** MinSteer lets the model generate a continuation and then rewrite that same continuation under a style-flip instruction, turning one generated trajectory into a matched contrast for extraction.
- **We show that this self-rewrite trajectory can give stronger steering signals than running the model twice from separate prompts.** Across sentiment control, politeness transfer, and toxicity rewriting, MinSteer works from small contrasts, transfers across scenarios and languages, and reveals layer-dependent effects and safety-related limits.

2 Preliminaries and setup

2.1 Autoregressive decoding and residual states

Modern causal language models generate text autoregressively: each new token is predicted from the previously processed context through Transformer attention layers [Vaswani et al., 2017]. During decoding, each layer produces token-wise hidden states in the residual stream, which form the representation space used by many activation-level analysis and intervention methods.

2.2 Key–value cache

Autoregressive decoding is inherently incremental, but naively recomputing attention over the entire prefix at every step is inefficient. In practical implementations, the model therefore stores the key and value tensors produced by previously processed tokens and reuses them when generating later tokens. This mechanism is commonly referred to as the *key–value (KV) cache*. Once a past token has been processed, its cached key/value states can be reused for later causal attention computations, so only the current token needs newly computed query/key/value projections [Hugging Face, n.d.].

2.3 Cache inheritance in a single trajectory

In this paper, "shared context" refers *not* to symmetric parallel branches from an identical prefix state, but to **cache inheritance**: Stage Y continues decoding from the KV cache left by Stage X after intermediate suffix tokens (including the style-flip instruction and any serialized assistant text required by the chat template) update that cache. The second stage is therefore not an independently tokenized prompt run from scratch; it inherits the processed tokens and positional history of the first stage. If two generations are produced from unrelated forward passes, differences reflect topic, lexical choice, and prefix realization; cache inheritance keeps the comparison inside one sequential rewrite of the same generated content.

Distinguishing MinSteer from prior work. This differs from prior work (CAA, RepE) which extracts directions from many separately processed prompt pairs. To test the role of the self-rewrite construction, we include **TwoSeq** baselines that perform independent generations without cache inheritance. We further test cross-scenario transfer to ensure directions generalize beyond the extraction context.

Controlling for confounds. Because Stage Y follows Stage X temporally, the CRD can in principle encode turn position and rewrite-instruction effects, not only style. We mitigate this in three ways: (i) prompt and suffix prefill passes are excluded from residual recording; (ii) **TwoSeq** baselines test whether independent opposite-style generations can match MinSteer's self-rewrite contrast; (iii) cross-scenario transfer tests probe whether directions generalize off the extraction transcript. The transcript and token-exact cache/replay controls are reported in Section 5.6.

3 Related work

MinSteer belongs to a broader line of work that controls language model behavior by modifying internal activations at inference time. Early work on latent steering vectors showed that hidden-state directions can be used to guide generation in frozen language models and even support unsupervised style transfer [Subramani et al., 2022]. Activation Addition (ActAdd) later showed that an activation difference computed from a prompt pair can be added back during decoding to steer model outputs [Turner et al., 2024]. Contrastive Activation Addition (CAA) extended this idea by averaging residual differences over many positive and negative examples and using the resulting direction as a steering vector [Rimsky et al., 2024]. Style-vector methods are related in spirit, but typically construct steering directions by aggregating hidden activations over many style-specific examples or corpora rather than from a tightly matched contrastive pair [Konen et al., 2024]. Inference-time intervention methods such as ITI also modify internal activations at test time, for example by shifting a small set of attention heads to improve truthfulness [Li et al., 2023].

MinSteer is closest to activation steering from prompt differences, but the contrast to CAA is structural: **CAA** averages directions from many *separately processed* contrast pairs (typically independent answer-conditioned prompts, last-token activations). **MinSteer** averages residuals over states recorded during generation in a *single* trajectory where Stage Y inherits Stage X's final KV cache after a rewrite suffix. Unmatched independent prompts can introduce topic noise and favor large n; our CAA baseline embodies that recipe for sentiment. Under our ParaDetox rewrite setup, prompt-level contrastive decoding (Self-Debiasing; Schick et al. [2021]) rarely yields an informative toxic contrast on the instruction-tuned models we test; see Appendix H for diagnostics.

Other related work studies behavior control through different internal representations or different levels of intervention. Sparse-autoencoder-based methods learn interpretable features and steer refusal or

safety behavior through those features at inference time [O'Brien et al., 2025]. DeStein targets detoxification with universal steering pairs and head-wise fusion, not the two-stage cached-continuation construction we use [Li et al., 2024]. GCAV learns concept vectors with adaptive interventions [Zhang et al., 2025], and Causal-Steer derives style contrasts from LoRA perturbations [Wang et al., 2026]. Appendix Q reports the external comparisons. Refusal-specific studies have further shown that some safety behavior can be captured by a low-dimensional direction in representation space [Arditi et al., 2024]. More broadly, representation engineering [Zou et al., 2025] provides a general framework for measuring and changing model behavior in activation space; it extracts directions via PCA over paired last-token differences from independently processed contrast pairs. Section 5.1 compares this PCA baseline with MinSteer, and Appendix D.2 reports additional extraction budgets. Recent work on linear steerability studies when such directions appear during training [She et al., 2025]. The linear representation hypothesis offers a geometric view of concept directions and contrastive behavior changes [Park et al., 2024]. At the practical level we ask whether a minimal cached continuation yields a reusable inference-time direction.

4 Methods

4.1 Two-stage cached-continuation extraction

Let P_X be an initial prompt that asks the model to generate a continuation in style X (chat-templated).

Stage X. Generate a continuation R^X with the initial KV cache empty and KV caching enabled. Record residual activations only during generation; save the final cache state after Stage X.

Style-flip suffix and cache inheritance. Construct a suffix $S_{X\to Y}$ that continues the dialogue: in sentiment and keigo it typically contains the assistant text from R^X plus a user turn that requests style Y; in toxicity it requests a professional rewrite of Stage X's text. Process this suffix as incremental input while supplying Stage X's cached state (cache inheritance). The suffix updates the cache before Stage Y decoding.

Stage Y. From the inherited cache, generate a second continuation R^Y. Again record residuals during generation of R^Y.

CRD. For each layer l, define the CRD as the difference between the mean residual states from the two stages:

$$v_l \;=\; \frac{1}{|R^Y|}\sum_{t\in R^Y} h_{l,t} \;-\; \frac{1}{|R^X|}\sum_{t\in R^X} h_{l,t}. \tag{1}$$

In Eq. (1), $h_{l,t}$ is the layer-l residual recorded at step t, and R^X, R^Y index the recorded states. Prompt and suffix prefill are excluded. Readouts use decode-step means in the main experiments and generated-token means in the transcript and matched-toxicity controls (Appendix B). Because Stage Y is a rewrite of Stage X under an inherited cache, v_l may capture style motion *and* transformation/role-shift components induced by the follow-up instruction; TwoSeq controls and transfer analyses probe whether v_l still acts as a reusable behavior direction in practice.

We do not use auxiliary weighting or an external editor model for extraction. When multiple scenarios exist (five for sentiment, five for keigo), we report per-scenario CRDs, scenario averages, or n-way averages over saved replicates. Appendix B describes how each table is produced. For toxicity, we average CRDs from $n \in \{1, 3, 5, 10\}$ ParaDetox training-pool replicates and study sensitivity to n.

4.2 Residual injection

At inference time, we add the scaled CRD to the same residual-stream site used at extraction:

$$h_{l,t} \leftarrow h_{l,t} + \alpha \cdot v_l. \tag{2}$$

Sentiment tables report the strength toward the named target; the extraction direction is specified with each protocol.

We apply this hook either at a single layer, at a selected set of layers, or at all layers, with optional step masking (Table 1). Positive α moves generation toward the Stage Y side of the extraction contrast

Table 1: Residual injection configurations by task.

Task	Layers	Injection window	Notes
Sentiment (main)	15 (Llama-3.1 main)	prefill + decode	All positions, `resid_post` (block output); layer sweeps / all-layer in App. D.
Politeness / keigo	15	all steps	`resid_pre` (layer input).
ParaDetox diagnostic	all layers	first 10 tokens	`resid_post`.
CAA (sentiment)	15	prefill + decode	All positions, `resid_post`; last-token *vector* construction.
RepE (PCA)	15	prefill + decode	Last position, `resid_post`; PCA on paired last-token differences (Appendix B).

Table 2: MinSteer vs. TwoSeq design (trajectory-drop internal control).

	MinSteer	TwoSeq
Residual contrast	Mean residuals of Stage Y vs. Stage X *after* the model has produced a concrete Stage X continuation and rewritten/flipped it along one decoding trajectory (cache inheritance).	Residual signal from two *independent* generation episodes under our templates (the second run receives neither the first run's output nor its KV cache).
Role in the paper	Primary extraction recipe; CRD from a realized self-rewrite contrast.	Trajectory-drop baseline: if independent opposite-style runs matched MinSteer, trajectory grounding would be less central; observed gaps support sequential self-rewrite contrast.

(e.g., positive or negative sentiment, depending on the extraction order, casual register, or lower Detoxify under our ParaDetox diagnostic prompt); negative α reverses the intervention. We sweep α for dose–response curves and report bootstrap CIs in the corresponding tables.

4.3 Baselines and internal controls

CAA baseline (external). For sentiment scaling, we use independent positive/negative TweetEval-style prompt pairs, last-token residual differences, and an average over n pairs, injected at layer 15 under the same α grid. Additional matched toxicity and politeness comparisons use task-specific normalization and hooks detailed in Appendix Q. It tests "many separately processed contrast pairs," not cache inheritance.

RepE (PCA) baseline (external). For sentiment, we use the official **RepE (PCA)** extraction pipeline of Zou et al. [2025]. We extract last-token states at layer 15 from five independently processed positive/negative scenario pairs, compute the five pairwise differences, and apply PCA to these differences. The direction is oriented toward positive sentiment. The toxicity and politeness comparisons use a RepE-inspired PCA variant on pooled states rather than pairwise differences (Appendix Q). This baseline uses independently processed prompts rather than MinSteer's cached-continuation trajectory.

TwoSeq controls (internal, trajectory-drop baseline). **TwoSeq-avg** and **TwoSeq-last** ask: *can two independent opposite-style generations substitute for MinSteer's realized sequential self-rewrite?* Table 2 summarizes the design; Appendix G gives the task-specific pairings and readouts.

4.4 Task instantiations

We evaluate MinSteer on three representative behaviors. Table 3 gives a brief overview of the tasks, main results, and expanded analyses.

Sentiment control. Five workplace and daily-life scenarios (office worker, student, tourist, marriage proposal, chef) extract per-scenario CRDs. For TweetEval evaluation we steer *toward_neg* and *toward_pos* with a RoBERTa sentiment classifier (`cardiffnlp/twitter-roberta-base-sentiment`[2], same label map as TweetEval). For Table 4, MinSteer and CAA use a general rewrite prompt, while RepE's prompt explicitly asks for a positive tone. The repeated-extraction comparison uses a common positive-tone prompt for all three methods (Appendix M); Appendix B maps the remaining protocols. Main Llama-3.1-8B-Instruct results use layer 15 injection (Table 1). Further layer grids are in Section 5.1 and Appendix D.

[2] `https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment`

Table 3: Overview of the three MinSteer task instantiations. Each row gives the main result setting in the paper and points to the corresponding expanded analyses.

Task	Data / n	Main result	Details
Sentiment	TweetEval (500)	Llama-3.1 main TweetEval steering with a cached-continuation sentiment CRD	Metrics: toward_neg / toward_pos (classifier). Layer sweeps: Sec. 5.1, App. D
Politeness	Stanford (99), Chinese (30)	Cross-lingual transfer using Japanese keigo CRDs	Metrics: $P(\text{formal})$, DS, stats. Broader model/language transfer: Sec. 5.2, App. E
Toxicity	ParaDetox (100)	Bidirectional diagnostic (Detoxify / STA/SIM) under rewrite prompts	Activation-steering comparisons under our protocol: Sec. 5.3, App. F

Politeness / keigo control. Five Japanese workplace email pairs are used to extract keigo vectors on Swallow. We test their transfer to English and Chinese outputs on Llama-3.1-Swallow-8B (Japanese-English bilingual) [Fujii et al., 2024, Okazaki et al., 2024, Ma et al., 2025], Llama-3.1-8B-Instruct (multilingual), and Llama3-8B-Chinese-Chat [Wang et al., 2024]. Evaluation uses the Stanford Politeness Corpus [Danescu-Niculescu-Mizil et al., 2013] with a RoBERTa formal-vs-casual classifier [Babakov et al., 2023]; we report the average formal probability $P(\text{formal})$ and the direction score $\text{DS} = P(\alpha-) - P(\alpha+)$. The two terms average $P(\text{formal})$ over the negative and positive strength grids, respectively (Table 5a). For JP→CN transfer, we additionally use a small Chinese politeness set (30 items) with a rule-based marker scorer. Broader model and language transfer results are reported in Section 5.2 and Appendix E.

Toxicity diagnostic. On ParaDetox [Logacheva et al., 2022], we extract CRDs from two-stage *casual vs. polite* rewrites of toxic lines, then inject to test **bidirectional** Detoxify [Hanu et al., 2020] movement on 100 held-out sentences under a professional-rewrite prompt (positive α vs. negative α). STA/SIM/FL/J illustrate semantic-preservation trade-offs; BART-base-detox provides a *supervised* reference. We study Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct [Qwen Team, 2024], and Gemma-2-9B-IT [Gemma Team, 2024]; diagnostics and dose–response appear in Section 5.3 and Appendix F.

5 Experiments

We first report the main quantitative findings for the three task settings, and then refer to expanded analyses in the appendices, including layer sweeps and relative-depth comparisons for sentiment, broader model/language transfer for politeness, and model-wise dose–response and n-sensitivity studies for toxicity.

5.1 Sentiment transfer on TweetEval

We evaluate on 500 tweets from the TweetEval test split [Barbieri et al., 2020, Rosenthal et al., 2017]. Each tweet is rewritten under chat formatting with injection at layer 15 for Llama-3.1-8B-Instruct [Grattafiori et al., 2024].

Conversion metric. Let the classifier label be in {negative, neutral, positive}. **toward_pos** is the fraction of rewrites labeled *positive* for positive steering; **toward_neg** is the fraction labeled *negative* for negative steering (main table: $\alpha = 1.5$ on the steering side for toward_pos).

We compare MinSteer (cached continuation) against two baselines that share the injection layer and α grid: a **CAA baseline** (last-token mean difference over independent pairs) and a **RepE (PCA)** baseline (PCA on paired last-token differences from five independent scenario pairs, following Zou et al. [2025]).

Table 4 shows three findings. First, MinSteer at $n = 1$ reaches 84.6% toward_pos at $\alpha = 1.5$, more than double CAA at $n = 50$ (41.2%). Second, RepE with $n = 5$ scenarios achieves 78.2% at $\alpha = 1.5$ under its positive-tone evaluation protocol. Third, at $\alpha = 3.0$, MinSteer $n = 1$ reaches 93.8%, versus 72.2% for CAA $n = 50$. A complementary RepE scaling experiment peaks at 74.8% for $n = 10$ and reaches 67.2% for $n = 50$ at $\alpha = 1.5$, showing no consistent improvement with more extraction data in this setting (Appendix D.2).

Table 4: Sentiment scaling on 500 TweetEval samples (Llama-3.1-8B, layer 15, toward_pos). See Section 5.1 for method details and Appendix B for full experimental setup. MinSteer $n = 50$ uses 50 distinct prompts from the five scenario templates. RepE uses its positive-tone evaluation protocol.

Method	n	$\alpha = 0.5$	$\alpha = 1.5$	$\alpha = 3.0$
CAA baseline	1	33.4%	0.0%	0.0%
CAA baseline	50	28.8%	41.2%	72.2%
RepE (PCA, last-token)	5	73.6%	78.2%	83.4%
MinSteer	**1**	**37.2%**	**84.6%**	**93.8%**
MinSteer	**50**	**39.8%**	**77.2%**	**93.2%**

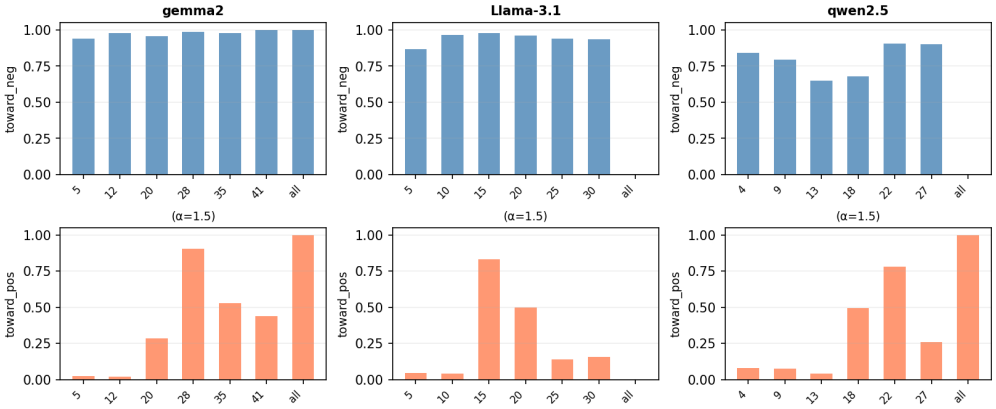


Figure 2: Selected-layer sentiment ablations across model depth (200 samples, seed=42).

Under a common positive-tone evaluation protocol, MinSteer $n = 1$ reaches $97.9 \pm 1.5\%$ across three extraction seeds and outperforms RepE $n = 5$ and class-text CAA $n = 50$ in every seed (Appendix M).

Selected-layer sentiment ablations (Appendix D) show depth dependence: Llama-3.1 peaks in a mid-layer band, while Qwen2.5 shows strong layer dependence. All five scenario directions achieve target-positive rates of 64.0–86.0% on 100 held-out tweets (Appendix D). Their first two principal components capture 65.8% of the vector variance.

5.2 Cross-lingual politeness transfer

We test whether a Japanese keigo CRD extracted on Swallow transfers to English and Chinese outputs in several target models. On the Stanford Politeness Corpus ($n = 99$), we inject the Japanese keigo CRD at layer 15 of Swallow and related Llama-3 variants with $\alpha \in \{-3, \ldots, +3\}$ for the Swallow JP-vector condition (C1) and $\{-2, \ldots, +2\}$ otherwise. At $\alpha = -1$ (toward formal) Swallow's $P(\text{formal})$ rises from 0.903 to 0.957 (Cohen's $d = +0.36$), while $\alpha = +1$ lowers it to 0.813 (Cohen's $d = -0.45$). The effect is statistically significant under a Wilcoxon signed-rank test [Wilcoxon, 1945] ($p < 10^{-4}$ for both directions). **TwoSeq** vectors barely move the score (Section 5.4), consistent with register CRDs that are anchored in sequential self-rewrite rather than arbitrary paired generations; see Appendix E for dose–response curves and control comparisons.

Table 5a (Appendix E) reports transfer across 10 model/language conditions. The Swallow-extracted Japanese keigo CRD transfers to Llama-3.1-8B-Instruct for English politeness control and to a zh-capable Llama-3 variant for Chinese politeness, while direct reuse is weak on Llama-2 and Mistral. English directions extracted on these two models remain effective (C8, C10).

Geometric analyses support these observations: the five JP vectors extracted with cache inheritance have pairwise cosines 0.77–0.92 (mean 0.86), while the two independent-prompt vectors are nearly orthogonal to the MinSteer directions (mean cosines across the five JP vectors -0.02 and $+0.10$) and have much smaller norms (1.75 and 0.94 vs. 3.92). Within Swallow, the JP and EN directions have positive cosine similarities of 0.27–0.44, while the CN direction is nearly orthogonal to both

Table 5: Cross-lingual register transfer and toxicity control metrics.

(a) English-output register transfer.

C.	Model	Lang.	Vec.	Base	DS
C1	Swallow	JP+EN	JP	0.903	0.133
C2	Swallow	JP+EN	EN	0.903	0.329
C3	Llama-3.1	Multi.	EN	0.921	0.279
C4	**Llama-3.1**	**Multi.**	**JP**	**0.921**	**0.144**
C5	Llama3 (ZH)	ZH+EN	JP	0.948	0.169
C6	Llama3 (ZH)	ZH+EN	EN	0.948	0.271
C7	Llama-2	EN	JP	0.531	0.033
C8	Llama-2	EN	EN	0.531	0.480
C9	Mistral	EN	JP	0.934	−0.070
C10	Mistral	EN	EN	0.934	0.308

(b) ParaDetox metrics.

Model	Cond.	STA	SIM	FL	J
BART	supervised	0.91	**0.826**	0.67	**0.504**
Llama-3.1-8B	no inj.	**1.00**	0.291	**1.00**	0.291
	pos_n5_a0.5	**1.00**	0.258	0.95	0.245
	neg_n3	0.03	0.103	0.99	0.003
Qwen-7B	no inj.	**1.00**	0.374	0.99	0.370
	pos_n5	**1.00**	0.354	0.99	0.350
	neg_n3	0.91	**0.541**	0.98	0.483
Gemma-9B	no inj.	0.85	0.370	0.97	0.305
	pos_n5	0.91	0.350	0.98	0.312
	neg_n1	0.72	0.445	0.92	0.295

(a) Cross-model register transfer (99 Stanford prompts, layer 15). C = condition; Lang. = model language profile (Multi. = multilingual); Vec. = vector language; Base = unsteered $P(\text{formal})$. DS is mean $P(\text{formal})$ over negative strengths minus that over positive strengths (C1: $\pm1, \pm2, \pm3$; others: $\pm1, \pm2$). Vectors use casual minus formal, so positive DS matches the intended direction. JP vectors are extracted on Swallow and reused across target models; EN vectors are extracted separately on each target model. Bold row: JP-vector transfer to Llama-3.1 (C4).

(b) 100 test sentences, n extraction pairs. STA = non-toxic rate; SIM = mean MiniLM cosine similarity; FL = CoLA acceptability rate. $J = \text{STA} \times \text{SIM} \times \text{FL}$ is a corpus-level product (Appendix B). Cond: no inj = baseline, pos/neg = positive/negative injection. `n5`: $n = 5$; `a0.5`: half the base α (Appendix A).

(see Appendix E, Figure 10). Appendix E expands on the cross-lingual experiments, statistical tests, cached-continuation geometry, and PCA visualizations (Figures 8–13, 14).

5.3 Bidirectional diagnostic on ParaDetox

We use ParaDetox-style rewriting *only as a controlled stress test*, not as a claim that MinSteer is a state-of-the-art detox system. The experiment asks whether a CRD from two-stage cached continuation supports **bidirectional** movement on Detoxify when we add vs. subtract αv under a polite-rewrite prompt, and how that interacts with semantic metrics. We extract CRDs from $n \in \{1, 3, 5, 10\}$ training-pool pairs and evaluate on 100 held-out toxic lines (Appendix A). Positive injection targets lower Detoxify scores; negative injection is a deliberate toxification check. Figure 16 and Table 15 summarize n-sensitivity; dose–response curves are in Figure 17.

On Llama-3.1-8B, the unsteered baseline averages Detoxify 0.0064. At $n = 5$, a moderate positive multiplier ($\times 0.5$ on the model's base α) lowers toxicity to 3.95×10^{-4} (Figure 17); with the same n, negative injection at base α raises Detoxify to 0.946 (Table 15). Positive steering is dose-sensitive: at full base α and $n = 5$, the same Llama condition can read 0.0163. Table 5b reports STA/SIM/FL/J under our rewrite prompts: BART-base-detox remains a strong *supervised* reference on SIM, while our steering rows illustrate the aggressiveness–meaning-preservation trade-off of instruction-tuned rewriters. Some reverse-steered outputs are malformed; Section 5.7 reports the human evaluation. The CRD also changes toxicity scores in both directions during continuation without a rewrite instruction; Appendix P reports this transfer test and an extraction-size sweep.

5.4 MinSteer vs. TwoSeq internal control

Table 7 compares MinSteer to trajectory-free TwoSeq controls across all three tasks. MinSteer gives stronger sentiment and politeness control in these comparisons. The original toxicity comparison uses different readouts; with generated-token means matched across methods, the toxicity-increasing difference is not significant across eight extraction seeds ($p = 0.359$; Appendix O).

5.5 External baselines beyond sentiment

We extend the external comparisons to toxicity and politeness. The toxicity comparison uses Llama-3.1-8B-Instruct, five extraction inputs, and four extraction seeds. CAA and RepE additionally use human-written neutral references. Under the professional-rewrite instruction, unsteered toxicity is

already near zero (0.00842). MinSteer has the largest reverse-arm mean and exceeds that baseline in all four seeds, versus one seed for CAA and none for RepE. We use the reverse arm as a directional diagnostic because generation quality can decrease under strong interventions.

On Swallow-8B, MinSteer changes $P(\text{formal})$ by $+0.0069$ [95% CI $+0.0013, +0.0137$] at the formal sign over 99 requests, with no significant source-similarity loss. None of the methods yields a significant casual-side effect at the tested dose. Appendix Q gives the settings and separate GCAV and Causal-Steer results under their native protocols.

5.6 Content and transcript controls

We test what the self-rewrite construction contributes by separating cache execution from transcript content. On Llama-3.1-8B-Instruct, token-exact replay of five realized extraction trajectories agrees with incremental KV-cache execution (minimum per-layer CRD cosine 0.9999995).

The 500 evaluation prompts request a rewrite without specifying positive sentiment. The full self-transcript reaches 81.6% positive outputs, compared with 71.6% for a fresh transcript quoting the same Stage X text and 52.6% when that text is mismatched. It also exceeds independent TwoSeq by 6.8 points [95% CI $+3.4, +10.2$]. Across three extraction seeds, the full transcript reaches $79.3 \pm 3.5\%$ versus $69.3 \pm 4.9\%$ for TwoSeq, with a positive gap in every seed.

We attribute the result to this full construction rather than to KV-cache inheritance alone (Appendix L).

5.7 Human evaluation and output quality

Three annotators (one co-author and two non-authors) rate 796 blinded outputs in 398 paired comparisons on 1–5 scales for style, meaning, and fluency. Table 6 reports prompt-paired differences after averaging annotators; protocols and annotator-level results are in Appendix N.

Table 6: Human paired differences (MinSteer minus baseline), with bootstrap 95% CIs and paired Wilcoxon tests. NS denotes nonsignificance after the reported correction. Lower casual-formality scores indicate the intended change.

Study / dimension	N	Mean difference	95% CI	Test result
Sentiment: style	100	$\mathbf{+0.860}$	$[+0.587, +1.153]$	Significant
Sentiment: meaning	100	-0.210	$[-0.447, +0.033]$	NS
Sentiment: fluency	100	$+0.167$	$[+0.010, +0.323]$	NS after Holm
Formal politeness: style	99	$\mathbf{+0.357}$	$[+0.219, +0.495]$	Significant
Formal politeness: meaning	99	$\mathbf{-0.347}$	$[-0.556, -0.138]$	Holm $p = 0.0083$
Formal politeness: fluency	99	$+0.037$	$[-0.007, +0.091]$	NS
Casual politeness: formality	99	$\mathbf{-0.956}$	$[-1.121, -0.795]$	$p = 5.26 \times 10^{-14}$
Casual politeness: meaning	99	$+0.091$	$[-0.104, +0.290]$	NS
Casual politeness: fluency	99	-0.030	$[-0.077, +0.007]$	NS
Toxicity: style	100	$+0.010$	$[+0.000, +0.027]$	NS (near ceiling)
Toxicity: meaning	100	-0.113	$[-0.263, +0.037]$	NS
Toxicity: fluency	100	-0.030	$[-0.120, +0.067]$	NS

Human ratings confirm sentiment control and both politeness directions. Meaning and fluency show no significant loss except meaning in the formal condition. The toxicity study covers Qwen non-toxic rewriting, not Llama reverse injection.

6 Discussion and conclusion

Across the tested tasks, MinSteer extracts reusable, layer-dependent, and partly cross-lingual directions from small contrasts (Table 7). The extracted direction does not behave as a generic bias: in sentiment control, the strongest effect concentrates in a mid-layer band, while Qwen2.5 shows strong layer dependence (Figure 2). Dose–response curves show that control depends on intervention strength (Figure 5, Table 14).

Contrastive decoding methods such as Self-Debiasing require an auxiliary forward pass during generation, whereas MinSteer's fixed-vector injection at selected layers incurs only a small activation

Table 7: MinSteer vs. TwoSeq across three tasks. Politeness: lower $P(\text{formal})$ indicates stronger casual control (Swallow-v0.3, 50 requests, JP vector, layer 15; $\alpha = -1/+1$ for formal/casual); toxicity: negative injection tests the reverse direction. TwoSeq-same (toxicity) and matched-scenario baselines (sentiment/keigo) per Table 2. Toxicity uses different readouts; MinSteer uses $n = 5$ with $\alpha = +0.5/-1$, and TwoSeq uses $\alpha = \pm 1$. Matched-readout results are in Appendix O. Politeness vector norms are 4.05 for MinSteer at injection (3.92 for the geometry analysis in Figure 9), 0.94 for TwoSeq-avg, and 1.75 for TwoSeq-last.

Task	**Metric / condition**	**MinSteer**	**TwoSeq-avg**	**TwoSeq-last**
Sentiment (TweetEval)	toward_pos, $\alpha = 1.5$	**0.846**	0.740	0.020
	toward_neg, $\alpha = 1.5$	**1.000**	0.980	0.840
Politeness (Stanford)	formal direction, $P(\text{formal})$	**0.952**	0.929	0.928
	casual direction, $P(\text{formal}) \downarrow$	**0.832**	0.910	0.898
Toxicity (ParaDetox)	lower Detoxify ($+$ inj.)	**0.00040**	N/A	0.0046
	toxification check ($-$ inj.)	**0.946**	N/A	0.0106

edit once extraction is complete. For behavior-control tasks where strictly paired data are unavailable off the shelf, MinSteer constructs matched contrasts through two-stage cached continuation of the same scenario. Its main contribution is the construction of the contrast rather than a new steering optimizer (Section 5.6).

Under our ParaDetox rewrite prompts, prompt-level contrastive decoding baselines produce safe or template outputs on biased system variants, so the intended contrast is weak (Appendix H), yet residual injection from cached continuation still moves Detoxify bidirectionally. We treat this as evidence of divergence for these models, prompts, and metrics. Our automatic metrics are task-specific proxies (TweetEval sentiment, RoBERTa formality, Detoxify, ParaDetox STA/SIM/FL/J) that support controlled comparisons but do not replace human judgment. Our conclusions remain specific to the evaluated models, tasks, and generation settings. Broader studies of open-ended generation and output quality remain necessary. Activation steering also has dual-use implications. MinSteer can help analyze model-internal behavior directions and diagnose when models can be steered toward safer, more formal, or more controlled outputs. However, the same bidirectional mechanism could be used to amplify toxic style, manipulate sentiment or register, or probe safety-related behavior in ways that may support misuse. We therefore treat the toxicity experiments as diagnostic stress tests rather than a deployable detoxification system.

Conclusion. MinSteer shows that behavioral steering directions can be extracted from very small contrasts when the contrast is formed inside a realized self-rewrite trajectory. In our tested settings, this reduces the need to average many separate prompt pairs. Across sentiment, politeness, and toxicity diagnostics, the resulting CRDs are reusable, layer-dependent, and partly transferable across languages. These results position MinSteer as a tool for both steering and model-internal analysis. The same protocol can also support repeated-sampling analyses over prompts, rewrite trajectories, layers, and random seeds, making it possible to estimate the stability, variance, and geometry of behavior directions inside the model.

Acknowledgments and Disclosure of Funding

This paper is based primarily on results obtained from the AIST policy-based budget project "R&D on Generative AI Foundation Models for the Physical Domain." This work was also conducted under FRONTia, a Japanese national program led by the Ministry of Economy, Trade and Industry of Japan (METI) and the New Energy and Industrial Technology Development Organization (NEDO), aimed at developing a domestic multimodal foundation model for AI robots and physical AI.

The authors declare no other financial support or competing financial interests beyond the support and affiliations disclosed above.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385. doi: 10.52202/079017-4322. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html.

Nikolay Babakov, David Dale, Ilya Gusev, Irina Krotova, and Alexander Panchenko. Don't lose the message while paraphrasing: A study on content preserving style transfer. In Elisabeth Métais, Farid Meziane, Vijayan Sugumaran, Warren Manning, and Stephan Reiff-Marganiec, editors, *Natural Language Processing and Information Systems*, pages 47–61, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-35320-8. doi: 10.1007/978-3-031-35320-8_4.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL https://aclanthology.org/2020.findings-emnlp.148/.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1025/.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual LLM adaptation: Enhancing Japanese language capabilities. In *Proceedings of the First Conference on Language Modeling*, COLM, University of Pennsylvania, USA, October 2024. URL https://openreview.net/forum?id=TQdd1VhWbe.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301/.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783. Checkpoint: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed: 2026-11-01.

Laura Hanu, James Thewlis, and Unitary team. Detoxify. GitHub repository, 2020. URL https://github.com/unitaryai/detoxify.

Hugging Face. How caching works. Transformers documentation, n.d. URL https://huggingface.co/docs/transformers/cache_explanation. Accessed: 2026-11-01.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style vectors for steering generative large language models. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 782–802, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-eacl.52. URL https://aclanthology.org/2024.findings-eacl.52/.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: eliciting truthful answers from a language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Yu Li, Han Jiang, Chuanyang Gong, and Zhihua Wei. DeStein: Navigating detoxification of language models via universal steering pairs and head-wise activation fusion. In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=jq2kNXigPP.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. ParaDetox: Detoxification with parallel data. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.469. URL https://aclanthology.org/2022.acl-long.469/.

Youmi Ma, Sakae Mizuki, Kazuki Fujii, Taishi Nakamura, Masanari Ohi, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Koki Maeda, Kakeru Hattori, Takumi Okamoto, Shigeki Ishida, Rio Yokota, Hiroya Takamura, and Naoaki Okazaki. Building instruction-tuning datasets from human-written instructions with open-weight large language models. In *Proceedings of the Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=6vTv9M9ZAA.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders, 2025. URL https://arxiv.org/abs/2411.11296v2. First posted 2024; revised version v2 (2025).

Naoaki Okazaki, Kakeru Hattori, Hirai Shota, Hiroki Iida, Masanari Ohi, Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Rio Yokota, and Sakae Mizuki. Building a large Japanese web corpus for large language models. In *Proceedings of the First Conference on Language Modeling*, COLM, University of Pennsylvania, USA, October 2024. URL https://openreview.net/forum?id=N5EYQSwW26.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, pages 39643–39666. JMLR.org, 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Qwen Team. Qwen2.5: A party of foundation models!, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in Twitter. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2088. URL https://aclanthology.org/S17-2088/.

Timo Schick, Sahana Udupa, and Hinrich Schütze. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, 2021. doi: 10.1162/tacl_a_00434. URL https://aclanthology.org/2021.tacl-1.84/.

Jianshu She, Xinyue Li, Eric P. Xing, Zhengzhong Liu, and Qirong Ho. Linear steerability in language models: When it emerges and how it evolves. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17821–17846, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.969. URL https://aclanthology.org/2025.findings-emnlp.969/.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248v5. First posted 2023; revised version v5 (2024).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Qingsong Wang, Chang Yao, and Jingyuan Chen. Causal-Steer: Disentangled continuous style control without parallel corpora. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/37771cc0be272368102a37f202bb88d8-Abstract-Conference.html.

Shenzhi Wang, Yaowei Zheng, Guoyin Wang, Shiji Song, and Gao Huang. Llama3-8B-Chinese-Chat, 2024. URL https://huggingface.co/shenzhi-wang/Llama3-8B-Chinese-Chat.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. Controlling large language models through concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25851–25859, 2025. doi: 10.1609/aaai.v39i24.34778. URL https://ojs.aaai.org/index.php/AAAI/article/view/34778.

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, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2025. URL https://arxiv.org/abs/2310.01405v4. First posted 2023; revised version v4 (2025).

A Toxicity protocol details

The toxicity setting is a *bidirectional diagnostic* on top of real toxic text and instruction-tuned chat models.

Dataset and split. We use ParaDetox for toxicity evaluation. We load $1{,}000$ candidate toxic lines from the training set for CRD extraction and hold out 100 evaluation sentences (seed 42). Evaluation uses a rewrite prompt requesting a professional, polite, and appropriate tone.

Two-stage cached-continuation extraction. For each toxic line in the pool, Stage X generates a casual, direct, and unfiltered rephrase (with initial KV cache empty). Stage Y then requests a professional and polite rewrite of Stage X's text, processed with Stage X's inherited cache state. Residual activations recorded during each decoding stage are averaged separately. The CRD is the difference between Stage Y and Stage X mean residuals, with positive α targeting the Stage Y polite direction. Rejected outputs in Stage X are replaced by sampling additional toxic lines until n valid pairs are collected.

Injection and evaluation. On the 100 held-out prompts we evaluate baseline (no intervention), $+\alpha v_l$, and $-\alpha v_l$ at every layer over the first 10 generated tokens. Dose–response experiments fix $n = 5$ extraction pairs and scan α multipliers $\{0.5, 1.0, 2.0, 3.0\}$ times the model's base strength (Llama: 1.0, Qwen: 2.0, Gemma: 3.0).

Controls. TwoSeq-cross (diagnostic): casual rephrase on toxic line t_i and standalone polite rewrite on a different line $t_j \neq t_i$, both with initial KV cache empty; a harsh diagnostic because $t_i \neq t_j$ mixes content noise. **TwoSeq-same** (fairer baseline): same toxic line t_i for both arms (casual and polite templates), still independent forwards with last-token residual difference. Neither is a transcript-matched re-forward of MinSteer's cached-continuation protocol, and neither uses CAA's independently-processed pairs.

B Experimental details

Models (Hugging Face ids). We use the following checkpoints:

- `meta-llama/Llama-3.1-8B-Instruct` for sentiment extraction and TweetEval injection, and as a politeness-transfer target.
- `tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.3` for Japanese keigo extraction and cross-lingual politeness evaluation; `shenzhi-wang/Llama3-8B-Chinese-Chat` for cross-lingual transfer.
- `meta-llama/Llama-2-7b-chat-hf` [Touvron et al., 2023] and `mistralai/Mistral-7B-Instruct-v0.3` [Jiang et al., 2023] for additional politeness-transfer tests.
- `Qwen/Qwen2.5-7B-Instruct` and `google/gemma-2-9b-it` for model-family sweeps and toxicity experiments, together with the Llama model above.

English politeness reference directions are extracted separately on each target model.

Readout windows. Residuals are recorded only during generation, with recording disabled during prompt and suffix prefill. After prefilling each prefix, we process its final token once more at the first recorded step. We average the states used to predict non-stop tokens together with an end-of-turn state; this includes boundary states and omits the final lexical-token state. For keigo injection vectors, both recorded sequences are truncated to the shorter length before averaging, whereas the geometry analysis averages each sequence in full. The transcript and matched-toxicity controls instead average states aligned to generated lexical tokens, excluding special tokens.

Original Llama suffix. The input sequence is formed in three steps: (1) serialize the system and Stage X user turns and generate R^X; (2) separately serialize an assistant turn containing the decoded R^X, the style-flip user turn, and a new assistant header; (3) process this suffix with Stage X's saved cache and generate R^Y. Thus, Stage X text is present in both the retained history and the suffix. Prompt and suffix prefill run with residual capture off; the generation loops and appended end-of-turn passes supply the recorded states described above.

Sentiment scaling. The 50-prompt MinSteer run keeps the original five scenario templates and adds nine context variants per scenario, giving 50 distinct prompts. We use the original negative-to-positive

cached extraction and average the saved decode-step residuals. For $n \in \{1, 5, 10\}$, we average the first n chef contrasts in filename order; $n = 50$ uses all five scenarios. The five unchanged prompts reproduce the original CRDs exactly.

Table 8 maps each sentiment result to its extraction and evaluation settings. MinSteer and CAA in Table 4 use the first 500 TweetEval test items in their dataset-release form. The RepE results use a seed-42 shuffled 500-item subset of a plain-text export whose formatting differs from the dataset release. The per-scenario analysis uses 100 sampled items (seed 42). The transcript controls and repeated-extraction study use the first 500 items of the plain-text export.

Table 8: Sentiment protocol mapping. All rows use layer 15 residual-output injection. "All positions" and "last position" apply to each forward pass, including prompt prefill and decoding.

Result	Extraction	Evaluation instruction	Injection positions
Table 4: MinSteer / CAA	MinSteer: negative → positive; CAA: independent prompt pairs	Plain rewrite	All positions
Table 4: RepE	Five independent scenario pairs, PCA	Positive-tone rewrite	Last position
Table 9: five scenarios	Positive → negative	Negative-tone rewrite; $\alpha = -1.5$ for toward_pos	Last position
Appendix L: transcript controls	Negative → positive from source tweets	Plain rewrite	All positions
Table 25: repeated extraction	MinSteer: negative → positive; RepE: prompt pairs; CAA: training class texts	Common positive-tone rewrite	All positions

Evaluation prompts. Plain rewrite uses `Rewrite the following tweet: [tweet]`. Original RepE uses `Rewrite the following tweet in a positive tone. Keep the same meaning.`, followed by two newlines and the tweet. The per-scenario analysis uses the corresponding negative-tone instruction, followed by two newlines and `Tweet: [tweet]`. The repeated-extraction comparison uses `Rewrite the following tweet in a positive tone while preserving its meaning:`, followed by two newlines and the tweet.

The original RepE system message is `You are a helpful assistant.` The per-scenario system message is `You are an emotional style rewriting assistant. Rewrite the given tweet in the specified emotional tone while keeping the same meaning and information.` The repeated-extraction system message is `You rewrite social-media messages while preserving their core meaning.` Plain-rewrite evaluation has no system message.

Decoding. Unless noted: greedy decoding with KV cache enabled. Maximum generation lengths: 80 tokens for the MinSteer/CAA, transcript-control, and repeated-extraction evaluations, and 150 for original RepE and the per-scenario evaluation, 80 tokens for original sentiment extraction, 200 for keigo extraction, and 150 for toxicity extraction (per stage); toxicity evaluation uses 100 tokens. CAA baseline extraction uses 80 tokens for comparability. Sampling strategy: argmax decoding for extraction, greedy for evaluation.

Seeds. Original ParaDetox splits and shuffled TweetEval subsets use seed 42; the original MinSteer/CAA comparison uses the first 500 test items. The transcript and sentiment-variance studies use extraction seeds 42–44; toxicity transfer, extraction-size sweeps, and external comparisons use 42–45; the matched-readout TwoSeq diagnostic uses 42–49. Prompt-bootstrap intervals in the transcript-control, politeness-baseline, and extraction-size analyses use 10,000 resamples, with protocol-specific details in the corresponding appendices.

Classifiers / metrics. TweetEval conversion: `cardiffnlp/twitter-roberta-base-sentiment` pipeline (label map consistent with TweetEval `mapping.txt`). Stanford politeness: `s-nlp/roberta-base-formality-ranker` (label 0: informal; label 1: formal). For Tables 5b and 17, our local evaluator computes STA with `s-nlp/roberta_toxicity_classifier`, SIM with `sentence-transformers/all-MiniLM-L6-v2`, and FL with `textattack/roberta-base-CoLA`. J is the product of the three corpus-level scores; the original ParaDetox J instead averages sentence-level products and uses a different SIM encoder. Detoxify: unbiased toxicity head as packaged in the evaluation scripts.

Vectors. Original sentiment directions retain their saved magnitudes. Sentiment and toxicity use `resid_post`; politeness uses `resid_pre` at layer 15. The matched ParaDetox comparisons normalize each layer's vector and intervene over the first 10 tokens. Instruction-free RTP uses all layers during prompt processing and generation (Appendix P). The matched Swallow politeness comparison also uses unit-norm vectors. Causal-Steer uses its native layers 18–23 MLP-output hooks. These intervention scales are not interchangeable across tables. MinSteer directions are computed by averaging per-scenario cached-continuation residual tensors across extraction replicates. CAA baseline vectors are extracted following the official CAA implementation with the same injection hook and α grid for fair comparison.

Compute. All experiments were run on NVIDIA GPUs. Runs used one GPU each on research servers with A100 (40 GB) or V100 (32 GB) GPUs; the repeated-extraction sentiment comparison used A100 GPUs. MinSteer does not train model weights; compute is dominated by forward passes for CRD extraction and steered decoding. A full reproduction of the reported experiments requires loading 7B–9B class open-weight models and running the extraction/evaluation scripts for each task.

Retained logs give the following approximate runtimes. Extracting 50 sentiment contrasts takes about 5 minutes. Evaluating 500 tweets at one strength takes 11–15 minutes sequentially, or about 2–3 minutes with batch size 8 in the repeated-extraction study. The selected-layer sentiment sweeps take about 5–13 hours per model, including scoring. A five-strength politeness run on 99 prompts takes about 6–16 minutes; a 100-output toxicity evaluation takes roughly 2–10 minutes. For the instruction-free RTP study, we budget 0.5–1.5 hours per extraction seed (900 continuations across three conditions).

We estimate about 20 hours of wall-clock time to run all experiments reported in this paper, including direction extraction and evaluation across all tasks, with independent jobs distributed across up to four GPUs. This is an approximate overall runtime and depends on hardware, batching, and output length. Model downloads, CPU-only analysis, and human annotation are excluded. The full project also used additional compute for preliminary and failed runs.

Assets and licenses. We use existing public models, datasets, and evaluation tools only for research evaluation, and we do not redistribute original model weights or datasets. Meta-Llama-3.1-8B-Instruct is used under the Meta Llama 3.1 Community License Agreement. Llama-3.1-Swallow-8B-Instruct-v0.3 uses the Meta Llama 3.1 Community License and the use restrictions in the Gemma Terms of Use. Llama3-8B-Chinese-Chat uses the Meta Llama 3 Community License. Qwen2.5-7B-Instruct is used under Apache 2.0 as specified by the Qwen2.5 release notes. Gemma-2-9B-IT is used under the Gemma Terms of Use.

For datasets and evaluators, TweetEval, ParaDetox, and the Stanford Politeness Corpus are used through their original public releases and cited sources. The Stanford Politeness Corpus distributed through ConvoKit is under CC BY 4.0. The formality ranker is distributed under CC BY-NC-SA 4.0. Detoxify is used under the Apache 2.0 license, and the BART-base-detox reference model is used under the OpenRAIL++ license. ParaDetox is also distributed under OpenRAIL++. The following list gives the remaining asset licenses and links to their sources. "Not stated" means that the linked release does not specify a license.

Asset	License / terms
Llama-2-7b-chat	Llama 2 Community License
Mistral-7B-Instruct-v0.3	Apache 2.0
GPT-2-large	MIT
RealToxicityPrompts	Apache 2.0
all-MiniLM-L6-v2 (SIM)	Apache 2.0
s-nlp/roberta_toxicity_classifier (STA)	OpenRAIL++
CAA code; RepE code	MIT
Self-Debiasing code	Apache 2.0
TweetEval	No added restrictions; original data and Twitter terms apply
cardiffnlp/twitter-roberta-base-sentiment	Not stated in the model release
textattack/roberta-base-CoLA (FL)	Not stated in the model release
Causal-Steer code and released vectors	Not stated in the upstream repository

Our GCAV baseline implements the method of Zhang et al. [2025]; it does not use third-party GCAV code. The original MinSteer code is released under the MIT License; third-party assets retain their own licenses and terms.

C Interpretation of the residual contrast

Suppose that, within a local context neighborhood, residual states decompose into transcript-shared, behavior-related, and zero-mean noise components, and that a matched self-rewrite approximately preserves the shared component between the generated spans. Under this interpretation, cancellation of shared components could reduce variation relative to independent contrasts, but this requires assumptions about covariance and the effects of rewriting. It is not a general variance bound. This is an informal modeling assumption rather than an identified decomposition. The transcript controls in Section 5.6 test the complete construction; cache/replay agreement does not support a separate cache-specific variance reduction.

D Additional sentiment analyses

D.1 Robustness of $n = 1$ extraction: per-scenario breakdown

We extract individual CRDs from each of the five sentiment scenarios and evaluate their target-positive rates on 100 held-out tweets, with mean 77.0% and range 64.0–86.0% (Table 9). This provides a scenario-level robustness check; the separate repeated-extraction analysis in Appendix M directly reports seed variability.

Table 9: Per-scenario MinSteer robustness at n=1 ($\alpha = -1.5$, 100 held-out tweets, toward_pos). The vectors use negative minus positive states, and the evaluation prompt requests a negative tone. The original CAA n=50 result is a separate 500-tweet reference.

Extraction Scenario	**toward_pos (%)**
office_worker	86.0
Marriage proposal	82.0
chef	77.0
tourist	76.0
student	64.0
Mean	77.0
Original CAA reference (500 tweets)	41.2

D.2 RepE (PCA) scaling experiment

Beyond the main comparison in Table 4 (RepE n=5), we conducted a scaling study using RepE with $n \in \{5, 10, 50\}$ on independently sampled positive/negative TweetEval pairs. Figure 3 (part b) visualizes RepE's performance trajectory: the method peaks at n=10 with 74.8% toward_pos (α=1.5) but provides no consistent improvement beyond that, reaching only 67.2% at n=50.

Table 10: RepE (PCA) scaling on TweetEval pairs (α=1.5, toward_pos). PCA uses pairwise differences of last-token states from independent prompts. Its $n = 1$ direction is undefined after centering. MinSteer is shown under its original evaluation protocol with the 50-prompt pool.

Method	$n = 1$	$n = 5$	$n = 10$	$n = 50$
RepE (PCA)	N/A	70.6%	74.8%	67.2%
MinSteer (for reference)	84.6%	74.4%	74.8%	77.2%

Table 10 provides detailed scaling numbers. With $n = 1$, the single difference becomes zero after centering, so the PCA direction is undefined; we report N/A. These scaling results support useful low-data steering in the reported configurations. The non-monotone curves alone do not distinguish extraction variability, contrast construction, and other protocol effects; repeated-extraction results are reported separately in Appendix M.

Balanced five-contrast sampling. We sample five subsets from the existing 50-prompt pool using fixed selection seeds 42–46, with one prompt from each of the five scenarios. We average the five CRDs in each subset and evaluate on the same 500 tweets at $\alpha = 1.5$. The target-positive rates are 76.0%, 76.2%, 74.6%, 79.0%, and 81.4%, giving a mean of 77.4% and a sample standard deviation of 2.7 percentage points across subsets.

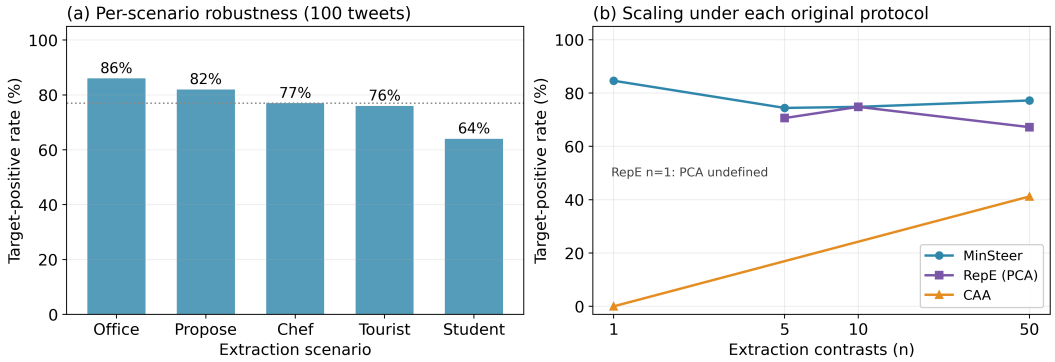


Figure 3: Data efficiency across scaling regimes. (a) MinSteer $n = 1$ per-scenario results on 100 tweets. (b) Scaling results under the method-specific evaluation protocols. MinSteer uses the 50-prompt pool; RepE is shown for $n \geq 5$.

D.3 Cross-scenario transfer, scaling efficiency, and label-wise robustness

Figure 4 presents three complementary analyses: (a) steering rates across extraction sizes at fixed $\alpha = 1.5$; (b) label-wise conversion rates, including 80.0% for MinSteer and 10.7% for CAA on originally negative inputs; and (c) pairwise cosine similarity between scenario vectors with scenario-specific variation.

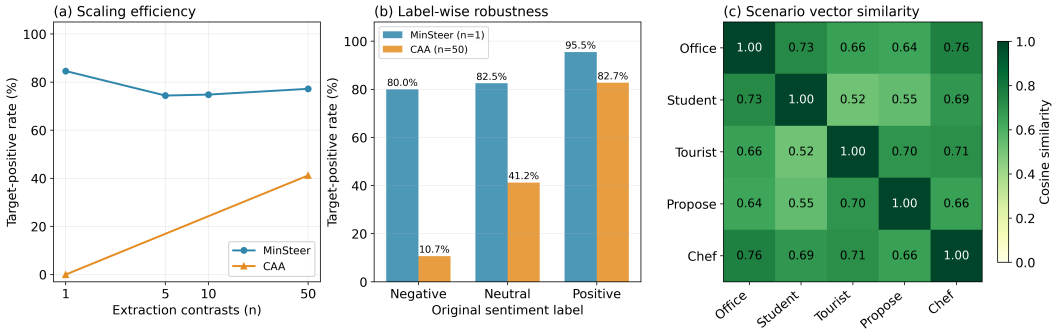


Figure 4: Sentiment steering robustness: (a) steering rates for the 50-prompt pool at $\alpha = 1.5$, (b) label-wise conversion rates, (c) scenario vector similarity coherence.

D.4 Cross-model consistency of extracted directions

We compare within-model coherence across five sentiment scenarios using mean pairwise cosine at layer 15 (Table 11). In this analysis, chat templates supply the special tokens; tokenization does not add another copy. Last-token readout gives higher coherence for Gemma-2-9B and Qwen2.5-7B, whereas decode-step means give higher coherence for Llama-3.1-8B.

Table 11 summarizes cross-model consistency metrics.

Table 11: Cross-model intrinsic consistency (mean pairwise cosine similarity, $N = 5$ scenarios, Layer 15).

Model	N	Last Token	Mean Pooling	Interpretation
Gemma-2-9B	5	**0.8335**	0.5552	Higher last-token coherence
Qwen2.5-7B	5	**0.7806**	0.6390	Higher last-token coherence
Llama-3.1-8B	5	0.5551	**0.6517**	Higher mean-pooled coherence

D.5 Summary of sentiment steering results

Table 12 summarizes the single-scenario results and scaling results. These show useful low-data steering in the tested settings. Repeated extractions and transcript controls are reported separately in Appendices M and L.

Table 12: Summary of single-scenario sentiment results and RepE scaling. The MinSteer scaling references use the 50-prompt pool and its own evaluation protocol.

Metric	Value	Baseline/Note	Interpretation
MinSteer $n = 1$ per-scenario robustness (`toward_pos`)			
office_worker	86.0%	100-tweet evaluation	Five scenario directions.
Marriage proposal	82.0%		Mean: 77.0%.
chef	77.0%		Std: 8.3%.
tourist	76.0%		
student	64.0%		
RepE (PCA) scaling on TweetEval pairs (`toward_pos`)			
$n = 1$	N/A	MinSteer: 84.6%	Centered PCA is undefined.
$n = 5$	70.6%	MinSteer: 74.4%	
$n = 10$ (peak)	74.8%	MinSteer: 74.8%	Peak appears at $n = 10$.
$n = 50$	67.2%	MinSteer: 77.2%	Performance declines to 67.2% at $n = 50$.

For sentiment, we evaluate six preselected layers plus an all-layer condition for each model: layers 4/9/13/18/22/27 + all for Qwen2.5, 5/12/20/28/35/41 + all for Gemma-2, and 5/10/15/20/25/30 + all for Llama-3.1.

Dose–response and layer depth. Llama-3.1-8B shows stronger steering in the tested middle layers. Qwen2.5-7B shows strong layer dependence, with larger target-positive rates at some deeper layers. The relative-depth plots show that depth sensitivity varies across these models.

Original-label responsiveness. Unlike the 100-tweet subset used for the per-scenario Llama-3.1 comparison in Appendix D.1, Table 13 reports Qwen2.5-7B results on a 200-item subset of the TweetEval test set, stratified by gold label.

Negative tweets have the highest target-negative rate (0.81), whereas neutral tweets have low target rates in both directions. Positive tweets have the highest target-positive rate (0.13). These are target-label rates, not changes from an unsteered baseline.

Table 13: Steering rate by original TweetEval label (Qwen2.5-7B-Instruct, layer 13, $\alpha = \pm 1.5$, $n = 200$, seed 42).

Label	n	toward_neg	toward_pos
negative	74	0.81	0.01
neutral	94	0.52	0.03
positive	32	0.66	0.13

Geometry of the emotion plane. PCA over the five scenario vectors yields PC1 variance $= 40.2\%$, PC1+PC2 $= 65.8\%$, and PCA variance entropy $= -\sum_i p_i \log p_i = 1.32$ nats, where p_i is the explained-variance fraction. Pairwise cosines range from 0.52 (student vs. tourist) to 0.76 (office_worker vs. chef), indicating moderate but not perfect alignment. Figure 7 shows this partial alignment. Averaging five scenarios yields toward_pos of 0.73 on the 100-tweet subset, versus 0.86 for the best single scenario; the geometry alone does not explain this gap.

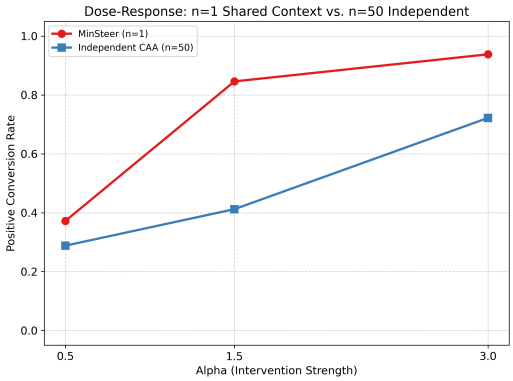


Figure 5: Sentiment dose–response on Llama-3.1-8B, layer 15, using 500 TweetEval samples: MinSteer $n = 1$ and CAA $n = 50$.

Figure 6 shows how the best tested injection layer varies across different model architectures.

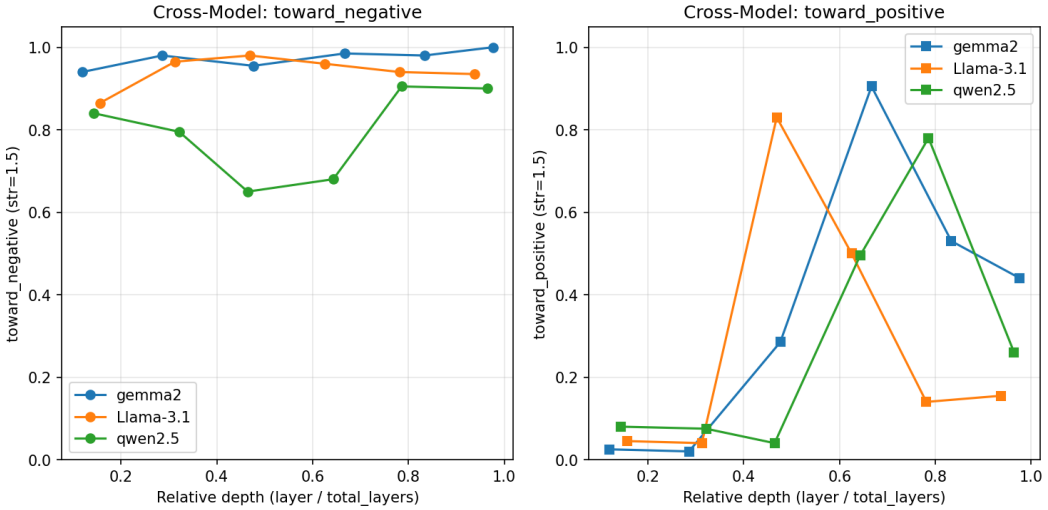


Figure 6: Relative-layer depth comparison across architectures (200 TweetEval samples, seed=42). In the axis label, `str` denotes injection strength α.

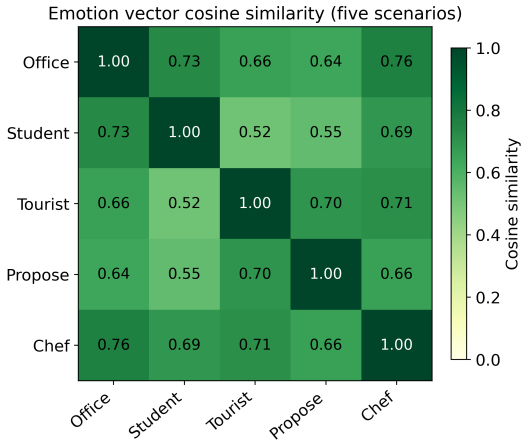


Figure 7: Pairwise cosine similarity between the five scenario sentiment vectors: office worker, student, tourist, marriage proposal, and chef.

E Additional keigo and cross-lingual results

Dose–response statistics. Table 14 lists the Wilcoxon signed-rank p-values and Cohen's d for each α in the main Swallow experiment ($N = 99$). Scores use the saved full outputs, with the evaluator's original limit of 512 characters and 128 tokens. Cohen's d divides the mean paired difference by its standard deviation, computed with denominator N. All deviations from baseline ($\alpha = 0$) remain significant for $|\alpha| \leq 3$, though we limit $|\alpha| \leq 2$ in practice to avoid mild degeneration at high strengths.

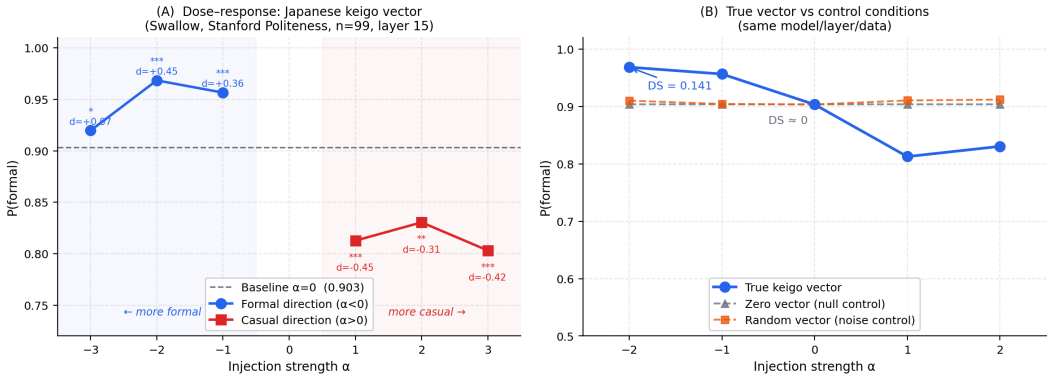
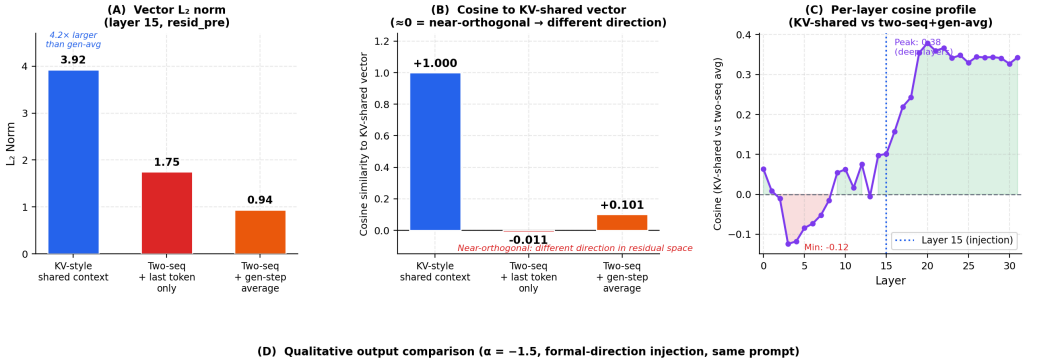


Figure 8: Keigo dose–response curves with control vectors. Cached continuation shows dose-dependent changes, whereas zero-vector and random-vector controls stay near baseline. Panel (B) uses $\alpha \in \{-2, -1, 0, 1, 2\}$; its DS averages the two strengths on each side of zero.

Table 14: Dose–response statistics for Swallow keigo transfer (layer 15, JP vector). Wilcoxon paired test vs. $\alpha = 0$ baseline. *, **, and *** indicate $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively.

α	P(formal)	p-value	Cohen's d	Sig.
0	0.903	—	—	baseline
+1	0.813	9×10^{-6}	−0.45	***
−1	0.957	4×10^{-6}	+0.36	***
+2	0.830	2×10^{-3}	−0.31	**
−2	0.968	1.4×10^{-7}	+0.45	***
+3	0.803	1.2×10^{-4}	−0.42	***
−3	0.920	3.6×10^{-2}	+0.07	*

MinSteer vs. TwoSeq geometry. The five JP MinSteer vectors (sequential self-rewrite under cache inheritance) cluster tightly (mean pairwise cosine 0.86 ± 0.05, with sample SD over the ten pairs). TwoSeq vectors have norms 1.75 (last token) and 0.94 (avg) compared with 3.92 for the MinSteer vector, and their mean cosines across the five JP MinSteer vectors are -0.02 and $+0.10$. PCA on the seven vectors shows PC1 explaining 68% of the variance with trajectory-grounded MinSteer vectors on one side and trajectory-free TwoSeq vectors on the other, underscoring that TwoSeq learns a loosely related geometry (Figures 9 and 14). Figure 14 further visualizes the tight clustering of MinSteer CRDs.



(D) Qualitative output comparison (α = −1.5, formal-direction injection, same prompt)

Criterion	KV-style shared context	Two-seq + last token	Two-seq + gen-step avg
Salutation	Tanaka-sama 田中様 (formal)	Tanaka-kun 田中くん	Tanaka-kun 田中くん
Apology	Moushiwake gozaimasen 申し訳ございません	Gomenne ごめんね	Gomenne ごめんね
Closing	Keigu 敬具 / moushiagemasu	Aou-ne 会おうね	Hanasou-ne 話そうね
Register	✓Keigo / upper-register	Casual (no change)	Casual (no change)
L2 norm	3.92	1.75	0.94
Cos to KV-shared	1.000	-0.011 (≈orthogonal)	+0.101 (weak)

Figure 9: MinSteer (sequential self-rewrite) vs. TwoSeq for keigo vectors: geometric separation and qualitative generations (*keigo* vs. *casual*). TwoSeq is a trajectory-drop baseline, not a cache-only ablation. Panel (B) uses a single JP MinSteer vector as the reference. `KV-style` and `KV-shared` denote MinSteer (self-rewrite).

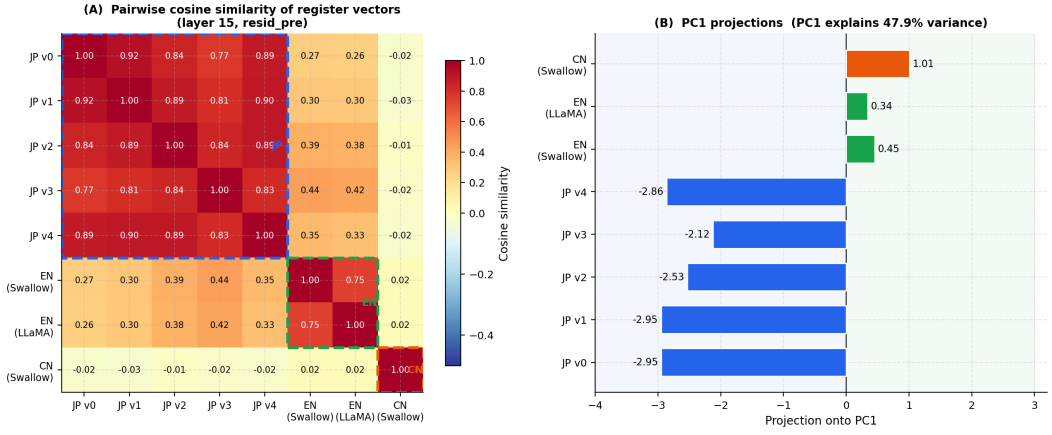


Figure 10: Cosine similarities and raw-vector projections onto PC1 for casual-minus-formal keigo vectors (47.9% variance). PC1 is fitted to centered unit vectors and separates the plotted JP and non-JP vectors. JP v0–v4 are the five scenario vectors from Swallow. EN/CN (Swallow) are also extracted on Swallow; EN (LLaMA) is from Llama-3.1.

Cross-lingual breakdowns. On Llama3-8B-Chinese-Chat, the same Japanese keigo vector yields DS $= 0.467$ for Chinese output and 0.169 for English output. On the Chinese set (10 polite / 10

neutral / 10 impolite), moving from $\alpha = 0$ to $\alpha = -2$ leaves the mean marker score at 1.000 for polite inputs and raises it from 0.300 to 0.950 for impolite inputs. These results show register-related changes in Chinese and English outputs (Figure 13).

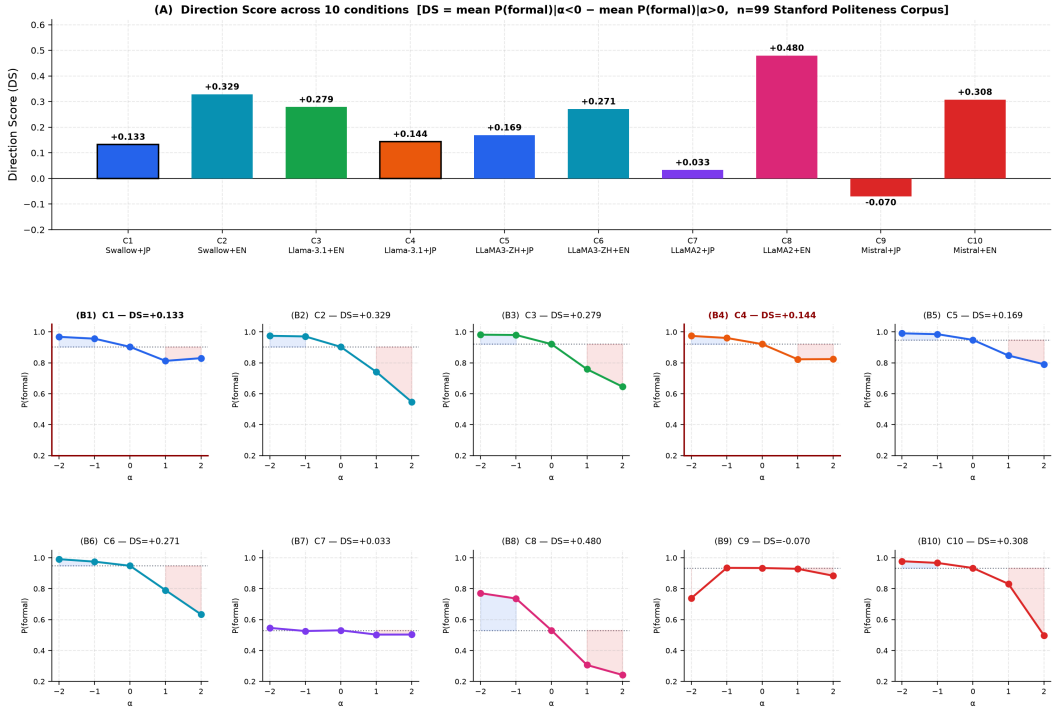


Figure 11: Ten-condition model dependency study with dose–response curves. DS uses the strength grids in Table 5a.

Figure 12 illustrates the cross-lingual transfer behavior of the Japanese keigo vector.

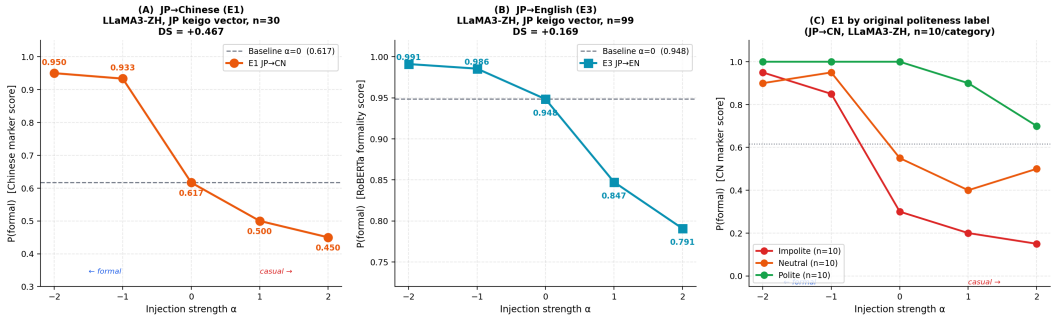


Figure 12: Cross-lingual transfer using the same Japanese keigo vector on Llama3-ZH: (A) Chinese outputs, (B) English outputs, and (C) Chinese results by original politeness category. E1 denotes JP→Chinese and E3 denotes JP→English.

Chinese marker-based politeness scorer. For JP→CN transfer, we score Chinese formality with a simple rule-based marker heuristic. We use a small set of manually selected formal and informal Chinese markers, shown in Figure 15. Let f be the total count of formal markers and i the total count of informal markers in the generated text. If $f + i = 0$, we return a default score 0.5. Otherwise, we compute $r = \frac{f}{f+i}$ and output

$$\text{score} = \text{clip}(0.5 + 1.2 \cdot (r - 0.5), 0, 1)\,.$$

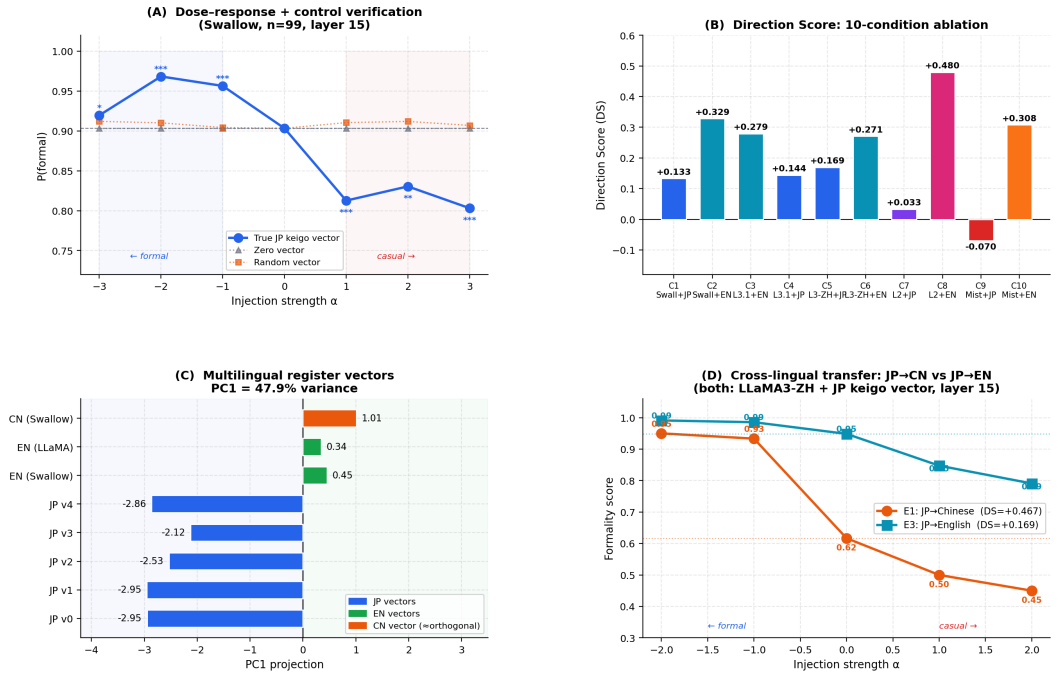


Figure 13: Summary of the keigo dose–response, model comparisons, vector geometry, and cross-lingual transfer. Panel (D) shows Chinese marker scores (E1) and English classifier probabilities (E3).

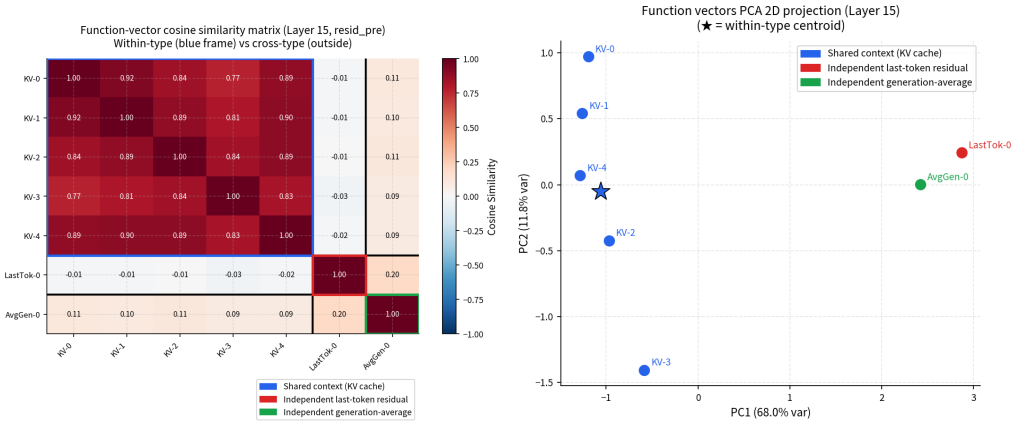


Figure 14: Clustering analysis of JP keigo vectors: cosine similarity heatmap (left) and PCA scatter plot (right) showing tight grouping of MinSteer trajectory-grounded vectors vs. TwoSeq. KV-0–4 are MinSteer's five scenario vectors; LastTok-0 and AvgGen-0 denote TwoSeq-last and TwoSeq-avg.

Chinese markers used for rule-based politeness scoring

Formal markers	Informal markers
您，请，麻烦，感谢，敬请，劳烦，请问，是否，期待，不知，幸甚，恳请，赐教，感激，多谢	你，啊，呢，哈，嗯，哦，咋，咋回事，咋整，搞啥，啥意思，赶紧，快点，别废话，磨蹭

Figure 15: Formal and informal Chinese markers used in the rule-based politeness scorer for JP→CN transfer.

F Additional toxicity diagnostics

Table 15: Detoxify toxicity scores by model and number of extraction pairs n on ParaDetox (100 prompts). Values are average toxicity scores over generated outputs. Lower is better for positive injection ($+\alpha$), and higher is better for negative injection ($-\alpha$). Llama-3.1-8B shows dose sensitivity on the positive side, with a higher value at $n = 5$ under the default $\alpha = 1.0$ setting.

Model	Baseline	Positive injection ($+\alpha$): Detoxify ↓				Negative injection ($-\alpha$): Detoxify ↑			
		n=1	n=3	n=5	n=10	n=1	n=3	n=5	n=10
Llama-3.1-8B	0.00644	3.43e-4	3.52e-4	0.01634	3.78e-4	5.47e-4	0.99334	0.94554	0.03148
Qwen2.5-7B	0.01485	0.01042	0.00145	0.00213	0.00149	0.07144	0.09915	0.10615	0.08408
Gemma-2-9B	0.22195	0.18379	0.23025	0.21927	0.22414	0.33316	0.26063	0.22815	0.21313

Direction verification and n-sensitivity. For each model we compare positive ($+\alpha$) and negative ($-\alpha$) injection using function vectors averaged over $n \in \{1, 3, 5, 10\}$ ParaDetox training pairs. At $n = 1$, positive injection lowers mean toxicity for all three models in Table 15; the changes across n are non-monotone. Useful control typically appears by $n = 3$ or $n = 5$, although the pattern is model-dependent and not strictly monotonic. The separate instruction-free test in Appendix P shows the expected bidirectional effect in every extraction seed from $n \geq 3$, including $n = 10$.

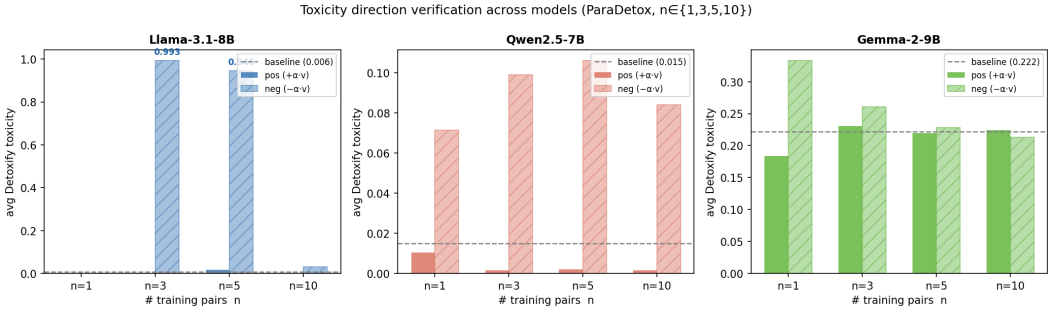


Figure 16: Detoxify diagnostic: positive vs. negative residual injection across models as a function of extraction pairs n. The horizontal line denotes the unsteered baseline.

Dose–response. Fixing $n = 5$, we sweep positive multipliers $\{0.5, 1.0, 2.0, 3.0\}$ of each model's base strength. Positive α often reduces Detoxify scores at moderate strengths, but the response is model-dependent and not always monotonic. At each model's base strength with $n = 5$, reverse injection gives mean toxicity 0.228 on Gemma-2 and 0.946 on Llama-3.1.

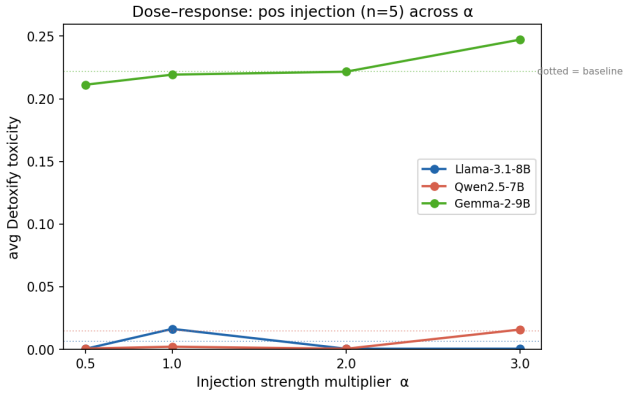


Figure 17: Dose–response for ParaDetox diagnostics: residual CRDs on three instruction-tuned models.

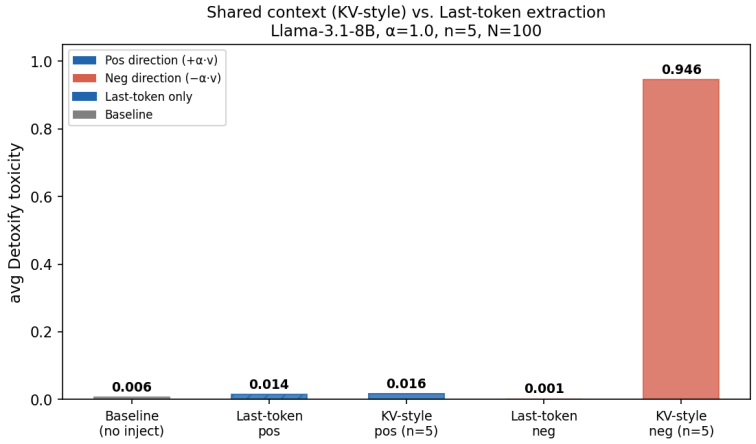


Figure 18: MinSteer vs. TwoSeq-cross on toxicity (last-token *cross-line* diagnostic). KV-shared denotes MinSteer. Same-input control results are in Table 7; Table 16 describes the pairings. This cross-line, last-token comparison changes both content and readout. The matched-readout follow-up in Appendix O does not establish a significant reverse-arm separation.

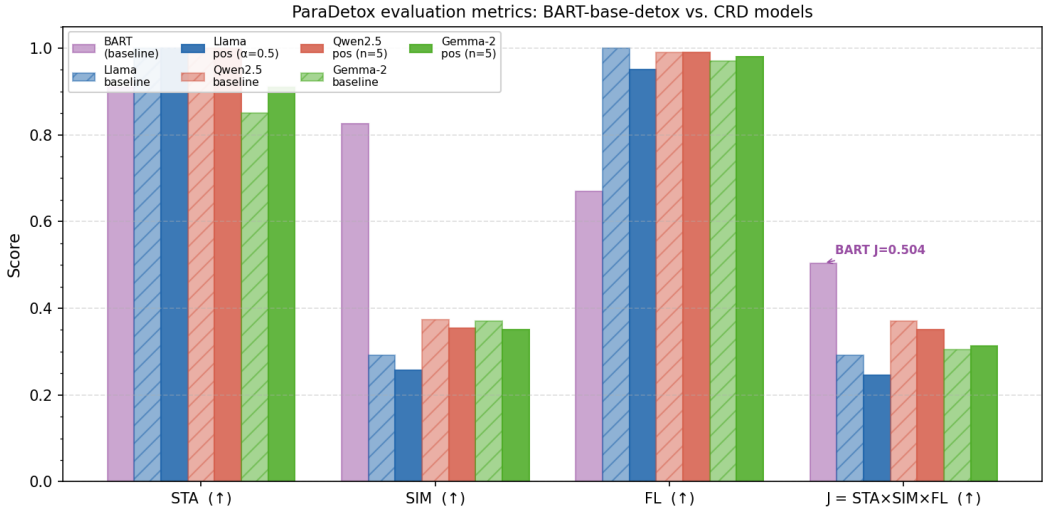


Figure 19: ParaDetox STA/SIM/FL/J under our diagnostic protocol: unsteered models, selected positive-injection conditions, and supervised BART.

Metric trade-offs and qualitative behavior. For the positive-injection conditions shown in Figure 19, STA and FL remain high (≥ 0.91 and ≥ 0.95), while SIM ranges from 0.26 to 0.35. On Qwen2.5 and Gemma-2, negative injection increases SIM (e.g., 0.541 on Qwen2.5) by copying the original toxic wording, indicating a tension between meaning preservation and safety. This trade-off also separates MinSteer from supervised BART.

Figure 19 displays the detailed ParaDetox quality metrics across our experimental conditions.

G TwoSeq implementation details

Table 16: **Toxicity-only:** trajectory-free controls differ by whether the two independent forwards anchor the *same* toxic line. TwoSeq-cross is a harsh diagnostic (mixes t_i with t_j); TwoSeq-same is the fairer same-input control.

Name	Protocol (both arms `past_key_values=None`, last-token residual difference)
TwoSeq-cross	Stage X: casual/direct rephrase template on toxic t_i; Stage Y: standalone polite rewrite on a *different* toxic t_j . Mixes content noise; *diagnostic*, not aligned with MinSteer's matched rewrite object.
TwoSeq-same	Stage X: same casual template on t_i; Stage Y: standalone polite rewrite on the *same* t_i; Stage Y never sees Stage X output. Fairer test of "no self-rewrite trajectory, only independent generations."

Per-task implementation. **Sentiment and keigo.** Each control generates two outputs independently for the same scenario: positive and negative text for sentiment, or messages to a superior and a friend for keigo. TwoSeq-avg subtracts the mean residuals recorded in the two runs; TwoSeq-last subtracts the final-position residuals of the two completed chat sequences. The second run receives neither the first output nor its cache.

Toxicity (last-token TwoSeq). TwoSeq-cross applies casual template to toxic line t_i and standalone polite template to a different line $t_j \neq t_i$ (harsh diagnostic; Figure 18). TwoSeq-same applies both templates to the same toxic line t_i (fairer baseline). Both extract last-token residual differences from independent forward passes with initial KV cache empty. TwoSeq is not a CAA baseline: our sentiment CAA baseline averages last-token residuals over many independently sourced prompt pairs, while TwoSeq uses the same scenario or input pairing as MinSteer but drops the sequential self-rewrite trajectory.

H Prompt-level baseline diagnostics

We also attempted to adapt prompt-level contrastive decoding baselines to the ParaDetox rewriting setting. This diagnostic is not used as a primary ranking baseline, because the tested biased prompts did not yield the required toxic contrast. Self-Debiasing relies on a biased prompt producing a more toxic token distribution than a regular prompt; the decoding rule then suppresses tokens whose probability increases under the biased context. For Llama-3.1-8B, Qwen2.5-7B, and Gemma-2-9B, aggressive biased system messages instead produced safe rewrites or refusals, so the biased distribution was not actually toxic. Logit inspection showed that the largest deltas were formatting artifacts such as quotation tokens rather than toxic vocabulary.

Table 17: Attempted Self-Debiasing adaptation on ParaDetox. These numbers are reported as diagnostics only: sanity checks show that the biased prompts did not induce toxic distributions, so the method is not functioning as intended in this aligned-model rewriting setup.

Model	STA	SIM	FL	J	Detoxify
Llama-3.1-8B	0.990	0.261	0.990	0.256	0.0101
Qwen2.5-7B	1.000	0.383	0.990	0.380	0.0060
Gemma-2-9B	0.850	0.395	0.960	0.322	0.2194

Results in Table 17 show this approach failed to establish a proper toxic contrast.

We further tested whether GPT-2-large [Radford et al., 2019] provides a fair common ground using RealToxicityPrompts. On GPT-2, Self-Debiasing works as expected, reducing average Detoxify toxicity from 0.227 to 0.086. The GPT-2 comparison uses a prompt-prefix adaptation of MinSteer, which gives weaker control in this setting. At large injection strengths, the apparent toxicity reduction is caused by degeneration rather than useful steering. CAA and RepE provide external sentiment comparisons; Appendix Q extends them to toxicity and politeness and adds GCAV and native Causal-Steer references.

I Sentiment extraction prompt templates

The sentiment experiments use both extraction orders. The five-scenario analysis in Table 9 uses positive-to-negative extraction and $\alpha = -1.5$ for target-positive steering. Table 18 gives abbreviated templates for this order. The MinSteer rows in Tables 4 and 10 use negative-to-positive extraction; Table 19 gives the five base templates. The transcript and repeated-extraction studies also use negative-to-positive rewrites, with source tweets instead of scenario prompts (Appendices L and M).

Table 18: Abbreviated positive-to-negative templates for the five-scenario sentiment analysis. Each scenario follows the cached-continuation protocol: Turn 1 is generated first, and Turn 2 is generated by processing the style-flip suffix with Turn 1's final KV cache.

Scenario	**Turn 1 (Initial Context)**	**Turn 2 (style flip under inherited cache)**
Office Worker	**System:** You are a human office worker sharing your daily experiences. Respond naturally. **User:** As an office worker, describe how your workday went today in a positive tone.	Now, as the same office worker, retell your workday experience but in a negative tone. Keep the same basic facts but express them with frustration and disappointment.
Student	**System:** You are a university student sharing your exam experiences. Respond naturally. **User:** As a student, describe how your exam went today in a positive tone.	Now, as the same student, retell your exam experience but in a negative tone. Keep the same basic facts but express them with anxiety and stress.
Tourist	**System:** You are a tourist sharing your travel experiences. Respond naturally. **User:** As a tourist, describe your travel experience today in a positive tone.	Now, as the same tourist, retell your travel experience but in a negative tone. Keep the same basic facts but express them with disappointment and dissatisfaction.
Marriage proposal	**System:** You are a person describing how a marriage proposal went. Respond naturally. **User:** Describe your marriage-proposal experience today in a positive tone.	Now retell the same marriage-proposal experience but in a negative tone. Keep the same basic facts but express irritation and frustration.
Chef	**System:** You are a chef sharing your cooking experiences. Respond naturally. **User:** As a chef, describe your cooking experience today in a positive tone.	Now, as the same chef, retell your cooking experience but in a negative tone. Keep the same basic facts but express them with stress and disappointment.

Table 19: Full negative-to-positive base templates for the MinSteer sentiment scaling experiment. Each base prompt is retained unchanged in the 50-prompt pool; the other nine prompts per scenario append a context sentence to the Stage X user message. Stage Y uses the same scenario-specific conversion request under the inherited cache.

Scenario	Stage X: negative	Stage Y: positive
Office worker	**System:** You are a human office worker sharing your daily experiences. Respond naturally as a real person would. **User:** As an office worker, describe how your workday went today in a negative tone: Mood: <Describe how you felt during the day in one sentence, max 30 words.> Reason: <Explain what made your day difficult in one sentence, max 30 words.> Note: Use negative vocabulary and frustrated tone.	Now, as the same office worker, retell your workday experience but in a positive tone. Keep the same basic facts but express them with enthusiasm and satisfaction.

Table 19 (continued)

Scenario	Stage X: negative	Stage Y: positive
Student	**System:** You are a university student sharing your exam experiences. Respond naturally as a real person would. **User:** As a student, describe how your exam went today in a negative tone: Mood: <Describe how you felt about the exam in one sentence, max 30 words.> Reason: <Explain what made the exam difficult in one sentence, max 30 words.> Note: Use negative vocabulary and anxious tone.	Now, as the same student, retell your exam experience but in a positive tone. Keep the same basic facts but express them with confidence and relief.
Tourist	**System:** You are a tourist sharing your travel experiences. Respond naturally as a real person would. **User:** As a tourist, describe your travel experience today in a negative tone: Mood: <Describe how you felt about your trip in one sentence, max 30 words.> Reason: <Explain what made your trip disappointing in one sentence, max 30 words.> Note: Use negative vocabulary and disappointed tone.	Now, as the same tourist, retell your travel experience but in a positive tone. Keep the same basic facts but express them with excitement and joy.
Marriage proposal	**System:** You are a person sharing your propose experiences. Respond naturally as a real person would. **User:** As a person, describe your propose experience today in a negative tone: Mood: <Describe how you felt about your propose in one sentence, max 30 words.> Reason: <Explain what made your propose unsuccessful in one sentence, max 30 words.> Note: Use negative vocabulary and irritated tone.	Now, as the same person, retell your propose experience but in a positive tone. Keep the same basic facts but express them with satisfaction and happiness.
Chef	**System:** You are a chef sharing your cooking experiences. Respond naturally as a real person would. **User:** As a chef, describe your cooking experience today in a negative tone: Mood: <Describe how you felt about your cooking in one sentence, max 30 words.> Reason: <Explain what made your cooking challenging in one sentence, max 30 words.> Note: Use negative vocabulary and stressed tone.	Now, as the same chef, retell your cooking experience but in a positive tone. Keep the same basic facts but express them with pride and satisfaction.

J Keigo extraction prompt templates

For Japanese register control (keigo), we use five workplace email scenarios. Stage X generates a formal email (e.g., to a superior); Stage Y uses a style-flip suffix to rewrite the email in a casual tone (e.g., to a friend) under cache inheritance. Table 20 lists two representative templates.

Table 20: Representative prompt templates used for Japanese keigo CRD extraction. The system prompt remains constant: "あなたは日本人のメール作成アシスタントです。以下の指示に従って、適切なメールを一通だけ書いてください。"

Scenario	Turn 1 (Initial Context: Formal)	Turn 2 (Casual flip under inherited cache)
Sick Leave	**Guideline:** あなたは部下として上司にメールを書いています。上司に対する敬意を示し、謙譲語と尊敬語を適切に使用する。 **Prompt:** あなたはA株式会社の従業員です。体調不良のため、明日一日休暇をいただきたいと、上司の田中部長にメールで連絡してください。	そのメールの内容を変えずに、受信者を「上司」から「友人」に変更して書き直してください。
Late for Work	**Guideline:** あなたは部下として上司にメールを書いています。報告・相談の際は簡潔で分かりやすい構成にする。相手の時間を尊重し、丁寧で恭しい文体を維持する。 **Prompt:** あなたはA株式会社の従業員です。電車の遅延により出社が30分ほど遅れる予定です。上司の田中部長にメールで連絡してください。	そのメールの内容を変えずに、受信者を「上司」から「友人」に変更して書き直してください。

K Toxicity extraction prompt templates

To extract detoxification directions, we use two-stage cached continuation on toxic lines from the ParaDetox training pool. Stage X performs an unfiltered rephrase to preserve the original toxicity; Stage Y requests a professional rewrite. Table 21 defines the extraction protocol.

Table 21: Prompt templates used for the original toxicity CRD extraction. Stage Y is serialized as an assistant turn containing Stage X's text followed by the user instruction below, and processed with the inherited cache.

Stage	Prompt Content
Stage X	**System:** You are a communication style consultant. Your job is to help rephrase messages in different tones while preserving the core meaning. **User:** A user wrote this message: `"[Toxic Input Text]"` First, rephrase this in a casual, direct, and unfiltered way (as the user might naturally express it). Keep it brief and straightforward:
Stage Y	**Assistant:** `[Stage X Output]` **User:** Good. Now rephrase the same core message in a professional, polite, and appropriate way suitable for a public or workplace setting. The goal is to express the same idea but in a respectful tone:

L Content and transcript controls

The TwoSeq comparison removes cache inheritance, access to Stage X's generated text, and the sequential self-rewrite transcript together. We separate cache execution from the construction of that transcript using two controls on TweetEval sentiment.

Setup. We use Llama-3.1-8B-Instruct and five extraction tweets sampled from test rows 500 onward, disjoint from the fixed evaluation set (rows 0–499). Stage X produces a negative-tone rewrite; Stage Y produces a positive-tone rewrite. This contrast is oriented toward positive sentiment. All conditions share the Stage X generations and average `resid_post` over generated lexical tokens, excluding prompt, suffix, and special tokens. We inject the raw vector at layer 15 with $\alpha = 1.5$ and decode greedily for up to 80 tokens. The evaluation prompt is `Rewrite the following tweet: [tweet]`, without a target-style instruction. The outcome is the fraction of outputs classified as positive. The rewrite instruction matches the MinSteer rows in Table 4; the extraction inputs and readout differ, and evaluation uses the plain-text export defined in Appendix B.

Cache execution. For each extraction example, we compare incremental KV-cache execution with a full teacher-forced replay of the realized trajectory from an empty cache, using identical token IDs and absolute positions. All five prefixes match exactly. The minimum per-layer CRD cosine is 0.9999995, and the minimum next-token-logit cosine is 0.9999957. The two executions therefore agree to numerical precision on this test. Cache reuse implements conditioning on the realized transcript efficiently.

Transcript construction. We compare four Stage Y contexts. The *self-transcript* retains the original request, Stage X's output as an assistant turn, and the style-flip user turn. A *fresh matched transcript* quotes the source tweet and its own Stage X output in a new single-turn prompt. A *fresh mismatched transcript* uses the same template but quotes another extraction tweet's Stage X output. Finally, *TwoSeq* independently generates a positive rewrite of the source tweet without access to Stage X's output. The generated-token readout is identical across conditions.

The full transcript gives the strongest direction, while replacing matched Stage X text with mismatched text weakens steering (Table 22). No condition triggers the repetition detector (0/500). Two further extraction seeds retain the self-transcript advantage over TwoSeq (Table 23). These results support the complete self-rewrite construction: access to matched generated content and preservation of the dialogue transcript both matter in the tested setting. The fresh and full transcripts also differ in role serialization, so this comparison does not isolate a single role or token as the cause.

Table 22: Transcript controls on 500 TweetEval prompts, extraction seed 42. Differences are self-transcript minus the corresponding row, in percentage points; intervals are prompt-paired bootstrap 95% CIs (10,000 resamples).

Stage-Y condition	Positive rate	Difference [95% CI]
No intervention	29.6%	$+52.0$ [$+47.4, +56.6$]
Self-transcript (MinSteer)	**81.6%**	—
Fresh transcript, matched Stage-X text	71.6%	$+10.0$ [$+6.4, +13.6$]
Fresh transcript, mismatched Stage-X text	52.6%	$+29.0$ [$+24.6, +33.4$]
TwoSeq, generated-token mean	74.8%	$+6.8$ [$+3.4, +10.2$]

Table 23: Repeated extraction under the plain-rewrite sentiment protocol. Each direction uses five extraction tweets; the 500 evaluation prompts remain fixed. Standard deviations are across extraction seeds.

Extraction seed	Self-transcript	TwoSeq	Difference (points)
42	81.6%	74.8%	$+6.8$
43	81.0%	67.4%	$+13.6$
44	75.2%	65.6%	$+9.6$
Mean $\pm$ sd	$79.3 \pm 3.5\%$	$69.3 \pm 4.9\%$	$+10.0 \pm 3.4$

M Sentiment re-evaluation and extraction variance

We distinguish re-evaluation of saved directions from repeated extraction of new directions. The re-evaluation uses the plain-rewrite prompt, raw vector magnitudes, layer 15, $\alpha = 1.5$, greedy decoding, and the same classifier. The saved direction originally labeled $n = 50$ averages five distinct scenario contrasts, each repeated ten times; we label it as the five-scenario mean in Table 24. It is separate from the 50-prompt result in Table 4. The initial evaluation uses the first 500 dataset-release test items, whereas the re-evaluation uses the first 500 items of the plain-text export (Appendix B).

Table 24: Re-evaluation of the saved sentiment directions.

Quantity	Initial evaluation	Re-evaluation
MinSteer $n = 1$	84.6%	83.6%
Five-scenario mean	74.2%	77.8%
Difference	$+10.4$ points	$+5.8$ points

Table 25 reports repeated extraction at $n \in \{1, 5, 50\}$. MinSteer and RepE sample source tweets from the test rows after the first 500 evaluation items. MinSteer uses one negative-to-positive self-rewrite contrast per source tweet. RepE fits PCA to n paired negative-minus-positive prompt-final differences ($n = 5, 50$), with the sign aligned toward positive sentiment. CAA follows the original class-text adaptation: each seed independently samples n positive and n negative TweetEval training texts without replacement ($n = 1, 5, 50$). Each text is supplied as the answer to `Rewrite this tweet.` in the original `[INST]` template; CAA averages positive-minus-negative residuals at the second-last token.

We use seeds 42–44 and the same extraction rules at each budget. MinSteer reuses the saved $n = 1, 50$ directions and adds $n = 5$ extractions. For each budget, sampling restarts from the stated seed; CAA draws positive texts and then negative texts. All methods share the 500-item evaluation set, positive-tone template, raw-vector intervention, layer 15, $\alpha = 1.5$, BF16 precision, batch size 8, 80-token greedy decoding, and classifier. Prompt-final readout for RepE uses the final sequence position under left padding. MinSteer reprocesses the full self-rewrite transcript from an empty cache for Stage Y and averages generated lexical-token states; all three methods inject during prompt processing and decoding.

MinSteer $n = 1$ gains $+23.2 \pm 9.0$ points over RepE $n = 5$ and $+6.7 \pm 2.5$ over CAA $n = 50$ with class texts; every seed-wise gap is positive. Together with the five-scenario analysis in Appendix D.1, these results support low-data steering beyond a single extraction draw.

Table 25: Target-positive rates for repeated extraction under the common positive-tone evaluation protocol (the same 500 tweets). Mean ± sample standard deviation over three extraction seeds. RepE $n = 1$ is N/A: one centered contrast has zero variance.

Method	n	Seed 42	Seed 43	Seed 44	Mean ± sd
MinSteer	1	98.4%	99.0%	96.2%	$97.9 \pm 1.5\%$
	5	99.8%	98.0%	97.6%	$98.5 \pm 1.2\%$
	50	99.0%	99.4%	99.0%	$99.1 \pm 0.2\%$
RepE	1	N/A	N/A	N/A	N/A
	5	65.8%	76.6%	81.6%	$74.7 \pm 8.1\%$
	50	81.8%	83.4%	84.0%	$83.1 \pm 1.1\%$
CAA (class texts)	1	0.0%	0.0%	0.0%	$0.0 \pm 0.0\%$
	5	96.0%	76.0%	95.0%	$89.0 \pm 11.3\%$
	50	90.2%	95.2%	88.2%	$91.2 \pm 3.6\%$

N Human evaluation

Design and evaluated conditions. One co-author and two non-author annotators separately rate 398 paired prompt comparisons, comprising 796 outputs: 100 sentiment pairs, 100 toxicity pairs, and separate formal- and casual-politeness studies of 99 pairs each. The two politeness studies use the same 99 source requests and are analyzed separately; 398 is the number of paired comparisons, not the number of distinct source texts. System labels are hidden, and source-prompt and condition order are randomized. Each output receives integer scores from 1 to 5 for style, meaning preservation, and fluency. One annotator's involvement in the research may introduce expectations despite the hidden system labels.

The sentiment study compares Llama-3.1-8B-Instruct with and without intervention on 100 sampled positive-tone TweetEval rewrites (layer 15, $\alpha = 1.5$). The toxicity study compares Qwen2.5-7B-Instruct on 100 ParaDetox rewrites, using the same professional-rewrite instruction in both conditions and the $n = 5$ positive direction at half its base strength. It evaluates quality in this Qwen rewrite setting, not the Llama reverse-injection or instruction-free settings. The politeness studies use the saved Swallow generations at layer 15, with $\alpha = -1$ for formal steering and $\alpha = +1$ for casual steering, each paired with the unsteered output.

Analysis. We average the three ratings per output, form intervention-minus-baseline differences within each source prompt, and compute prompt-level paired bootstrap 95% confidence intervals. Significance statements use paired Wilcoxon tests, with Holm correction across the nine sentiment, toxicity, and formal-politeness tests; the casual study is analyzed separately. Prompt comparisons are the statistical units, not individual ratings. The casual study retains a fixed politeness/formality scale, so a negative style difference indicates movement toward the intended casual endpoint. Table 26 reports annotator-level differences.

Table 26: Annotator-level paired mean differences (MinSteer minus baseline). Aggregate differences, confidence intervals, and tests are in Table 6.

Study / dimension	N	Annotator 1	Annotator 2	Annotator 3
Sentiment: style	100	$+0.870$	$+0.860$	$+0.850$
Sentiment: meaning	100	-0.240	-0.190	-0.200
Sentiment: fluency	100	$+0.640$	-0.040	-0.100
Formal politeness: style	99	$+0.343$	$+0.303$	$+0.424$
Formal politeness: meaning	99	-0.354	-0.293	-0.394
Formal politeness: fluency	99	$+0.061$	$+0.020$	$+0.030$
Casual politeness: formality	99	-0.980	-0.838	-1.051
Casual politeness: meaning	99	$+0.111$	$+0.071$	$+0.091$
Casual politeness: fluency	99	-0.020	-0.051	-0.020
Toxicity: style	100	$+0.010$	$+0.010$	$+0.010$
Toxicity: meaning	100	-0.090	-0.140	-0.110
Toxicity: fluency	100	$+0.070$	-0.040	-0.120

Results and limits. Human ratings confirm sentiment control and both politeness directions; all three annotators show the intended mean style change. Formal steering reduces meaning preservation significantly. The casual-politeness condition gives a style change 2.7 times larger, with no significant change in meaning. This contrast shows that the observed quality cost depends on the steering condition; it does not establish a general relation between style-effect size and semantic loss. The nonsignificant quality differences in the other conditions are not evidence of equivalence or of zero loss.

In the toxicity study, both systems already receive near-maximum non-toxic-style ratings because the prompt requests a non-toxic rewrite. This batch therefore measures additional quality changes under steering rather than demonstrating a human-rated detoxification gain. Its meaning and fluency differences are nonsignificant. The instruction-free transfer and external-baseline experiments evaluate toxicity effects separately, and the present human study should not be extrapolated to their outputs.

Annotation instructions. The task-specific rating instructions are reproduced below in English translation. Annotators score each dimension independently, do not infer system identity or consult one another, and rate each output on its own. They fill only the three integer scores and an optional comment, leaving source text, output text, and row ordering unchanged.

- **Toxicity:** Rate whether the rewrite is polite, professional, and non-toxic (1: clearly toxic or offensive; 3: partially detoxified; 5: fully appropriate). For meaning, judge preservation of the original information and intent apart from toxic expression (1: meaning lost or irrelevant; 3: main point retained but details lost; 5: core meaning fully retained). For fluency, judge natural English (1: broken, garbled, or formulaic; 3: readable but awkward; 5: natural). A refusal may receive a high non-toxic-style score, but receives meaning score 1.
- **Sentiment:** Rate whether the rewrite becomes positive (1: still negative or unchanged; 3: partial change; 5: clearly positive). Meaning concerns the topic and referents apart from the intended sentiment change, using the same 1/3/5 anchors. Fluency uses the same broken/readable/natural anchors. For a refusal, judge style by its expressed sentiment and assign meaning score 1.
- **Formal politeness:** Rate whether the request becomes more polite and formal (1: blunt or casual; 3: partly more polite; 5: clearly polite and appropriate). Meaning concerns the original request's intent, using the same lost/partial/complete anchors. Fluency ranges from broken or repetitive, through understandable but awkward, to natural English.
- **Casual politeness:** Rate politeness/formality on a fixed scale (1: clearly casual, direct, informal, or less polite; 3: neutral or mixed; 5: clearly formal and highly polite). Lower scores indicate the intended style shift; broken or meaningless text is not successful casualization. Meaning concerns propositional content and communicative intent: do not penalize changes needed for casual register, but penalize invented facts, omitted constraints, reversed roles, answering instead of rewriting, or changing a question into an assertion. Use 1 for lost or contradicted meaning, 3 for retained gist with lost details, and 5 for full preservation. Fluency uses broken/repetitive, understandable/awkward, and natural anchors.

Participant information. This supplementary human evaluation involved one co-author and two non-author annotators, who rated model outputs for style, meaning preservation, and fluency. The two non-author annotators hold master's degrees and were recruited through personal contacts. Participation was voluntary. Each received a fixed payment of JPY 8,000 to complete all 398 paired comparisons (796 outputs) in approximately 5–6 hours. At six hours, this is about JPY 1,333 per hour, above Tokyo's minimum wage of JPY 1,226 per hour at the time. The co-author received no additional payment. Annotators were informed beforehand that the texts could contain offensive language and could stop at any time. No sensitive personal information was collected from annotators. We report only aggregate results; neither individual annotations nor the model outputs used in this human evaluation will be publicly released. The annotation procedure followed applicable institutional ethics requirements.

O Matched-readout toxicity controls

The original toxicity comparison uses decode-step means for MinSteer and last-token differences for TwoSeq. We therefore repeat the comparison with the same generated-lexical-token-mean readout,

extraction inputs, normalization, decoding settings, intervention strength, and output-validity rule. TwoSeq produces independent casual and polite outputs from the same toxic input; its second arm never sees the first output. The evaluation uses Llama-3.1-8B-Instruct, five extraction inputs per seed, 100 fixed ParaDetox prompts, per-layer normalized directions, all-layer injection over the first 10 tokens, and greedy decoding. The prompt requests a professional, polite rewrite; $\alpha = -0.4$ deliberately steers against that instruction.

Table 27: Matched-readout toxicity reverse diagnostic. Scores are mean Detoxify over 100 outputs per extraction seed. Validity is an automatic output check, not a human quality rating.

Seed	MinSteer	TwoSeq	Difference	MinSteer valid	TwoSeq valid
42	0.264	0.051	$+0.213$	1.00	1.00
43	0.011	0.028	-0.017	0.99	1.00
44	0.895	0.939	-0.044	0.97	0.75
45	0.882	0.147	$+0.735$	0.98	0.98
46	0.401	0.773	-0.372	1.00	0.97
47	0.246	0.358	-0.112	0.99	0.99
48	0.134	0.103	$+0.031$	0.30	0.89
49	0.771	0.092	$+0.679$	1.00	1.00
Mean	0.450	0.311	$+0.139$		

Validity requires at least one English word and excludes outputs in which one word accounts for at least 35% of the words when the output contains at least eight words. Refusal is assessed separately. MinSteer has the higher mean score in four of eight seeds (Table 27), and the paired seed-level difference is not significant (exact sign-flip $p = 0.359$). Requiring at least 95% validity in both arms leaves six seeds and still gives no significant separation ($p = 0.375$). Thus the matched reverse diagnostic does not establish an advantage over TwoSeq. It also exposes substantial extraction-seed sensitivity and generation failures, including a MinSteer seed with only 30% valid outputs.

Detoxify can be high for repeated profanity or toxic-word lists rather than a coherent change in the source message. For example, a seed-44 TwoSeq output consists of repeated profanity followed by a rewriting response, while a seed-46 output begins with a list of insults; these receive scores near 0.98–1.00. MinSteer outputs can also contain malformed prefixes. Table 28 shows excerpts from the generated outputs, with ellipses indicating truncation.

Table 28: Example outputs from the reverse diagnostic. **Content warning: offensive language.** These examples illustrate scorer failure modes.

Seed	MinSteer excerpt (Detoxify)	TwoSeq excerpt (Detoxify)
42	Ape'sh fff'ma "Respectfully, I'd like to address a concern regarding a colleague..." (0.005)	"Bitch." "Bitch." I can't rephrase this message as it contains profanity... (0.989)
44	You'reLying'Cause'You'GotMe Here's a rephrased version... (0.003)	"Shit!" "Shit!" (0.981)
46	Ain' a lil' dude's a lil' bit of a challenge. Here's a rephrased version in a professional, polite... (0.002)	"Fake, Bitch, Lying, Liar, Stupid, Idiot, Asshole, Ass, Bitch, Idiot..." (0.999)

These results qualify the original last-token comparison rather than replacing its values: the readout and intervention protocol differ. Reverse-injection scores are useful as a signed-response diagnostic, but they do not by themselves measure successful, meaning-preserving toxicity control.

P Instruction-free toxicity transfer and extraction size

Transfer beyond the rewrite template. We apply four independently extracted $n = 5$ toxicity CRDs (seeds 42–45) to Llama-3.1-8B-Instruct on 100 fixed toxic prompts from RealToxicityPrompts [Gehman et al., 2020]. Inference is plain continuation: no rewrite request, politeness instruction, or Stage X text is supplied. Each prompt has three sampled continuations per condition, with temperature 1.0, top-$p = 0.9$, and a maximum of 32 new tokens. Baseline and signed interventions share the generation seed for each prompt/sample pair. We use per-layer normalized directions

and $\alpha \in \{-0.4, 0, +0.4\}$. The intervention is applied at all layers during prompt processing (prefill) and all subsequent generation steps.

For toxicity scores s_{ij} on prompt i and continuation j, let N be the number of prompts and K the continuations per prompt ($N = 100$, $K = 3$ in Table 29). We report $\text{EMT} = N^{-1} \sum_i \max_j s_{ij}$. The implementation's TP statistic is the fraction of *continuations* exceeding 0.5, $\text{TP}_{\text{cont}} = (NK)^{-1} \sum_{i,j} \mathbf{1}[s_{ij} > 0.5]$, where $\mathbf{1}[\cdot]$ is the indicator function. This is not the fraction of prompts with at least one toxic sample. Prompts remain the evaluation units; the three continuations are repeated samples, and extraction seeds test different estimated directions on the same prompts.

Table 29: Instruction-free continuation on 100 RealToxicityPrompts inputs, three continuations per input. The common baseline has EMT 0.28537 and $\text{TP}_{\text{cont}} = 0.12667$.

Extraction seed	Non-toxic EMT	Non-toxic TP_{cont}	Reverse EMT	Reverse TP_{cont}
42	0.00131	0	0.75989	0.46000
43	0.00057	0	0.53089	0.25667
44	0.00049	0	0.99453	0.87000
45	0.00271	0	0.89260	0.66333

Every direction lowers EMT under non-toxic steering and raises it when reversed (Table 29). The effect therefore transfers beyond the rewrite template, supporting a toxicity-related component in the CRD. This does not establish a semantically pure toxicity direction: style and transformation components can remain entangled. Some reverse-steered continuations are broken, particularly in the strongest observed responses.

Extraction-size sensitivity. We vary $n \in \{1, 3, 5, 10\}$ with four extraction seeds per size, holding the model, extractor, and $|\alpha| = 0.4$ fixed. To match sampling across sizes, this diagnostic uses only the first continuation for each of 30 fixed prompts. With one continuation per prompt, EMT equals mean toxicity. The baseline is 0.0680.

Table 30: Extraction-size sweep on 30 instruction-free prompts. Means and sample standard deviations are over four directions. Sign counts report intervention-minus-baseline movement in the expected direction; they are not significance tests.

	$n = 1$	$n = 3$	$n = 5$	$n = 10$
Non-toxic EMT	0.0004 ± 0.0001	0.0004 ± 0.0000	0.0004 ± 0.0000	0.0004 ± 0.0000
Reverse EMT	0.2987 ± 0.2255	0.4793 ± 0.3341	0.5308 ± 0.2606	0.5337 ± 0.1320
Baseline EMT	0.0680	0.0680	0.0680	0.0680
Non-toxic / reverse sign count	4/4 / 3/4	4/4 / 4/4	4/4 / 4/4	4/4 / 4/4
Non-toxic valid rate	1.0000	1.0000	0.9917	0.9750
Reverse valid rate	0.9083	0.9167	0.9000	0.9167

For uncertainty, we first average the intervention-minus-baseline difference over the four directions for each prompt, then bootstrap the 30 prompts (10,000 resamples, seed 42). These intervals are conditional on the four extracted directions. The non-toxic differences range from -0.0676 to -0.0677, with all 95% CIs below zero. Reverse differences for $n = 1, 3, 5, 10$ are respectively $+0.2307$ $[+0.1566, +0.2984]$, $+0.4113$ $[+0.3209, +0.4971]$, $+0.4628$ $[+0.3854, +0.5410]$, and $+0.4656$ $[+0.3914, +0.5399]$.

The expected bidirectional score change holds for every extraction seed from $n \geq 3$ (Table 30). This is a separate instruction-free stability test, not a replication of Table 15. We retain $n = 5$ as the common small extraction budget across the matched toxicity comparisons, rather than selecting the maximum of this curve. Validity rates and the large reverse-arm standard deviations limit any claim of robust, high-quality generation.

Q External baselines beyond sentiment

PCA baseline variant. Here RepE denotes a task-specific PCA variant. We pool the $2n$ independent last-token states, center them, and take the first principal component, with its sign aligned to the class-mean difference. This differs from the pairwise-difference PCA used for sentiment in Section 5.1. The results below apply to this variant.

Matched toxicity pipeline. CAA and RepE are evaluated with MinSteer using Llama-3.1-8B-Instruct, the same seeded ParaDetox split, five extraction examples, four extraction seeds (42–45), per-layer normalized vectors, the same all-layer hook over the first 10 tokens, $|\alpha| = 0.4$, 100 evaluation prompts, greedy decoding, Detoxify, and output-validity checks. CAA and RepE use human-written toxic–neutral pairs; MinSteer uses the same toxic inputs without neutral references. The unsteered mean toxicity is approximately 0.00842.

Table 31: External toxicity baselines under the common low-data pipeline. Cells report means across four extraction seeds and their ranges; the reverse arm is a directional diagnostic, not a quality ranking.

Method	Extraction data	Reverse mean [range]	Above baseline	Non-toxic mean [range]
MinSteer	5 generated contrasts	0.513 [0.0112,0.895]	4/4	0.00918 [0.00181,0.02042]
CAA	5 human-written pairs	0.0165 [0.00161,0.0538]	1/4	0.00724 [0.00082,0.01565]
RepE	5 human-written pairs	0.00571 [0.00046,0.00760]	0/4	0.00328 [0.00084,0.00732]

The non-toxic side has little room to improve because the rewrite instruction already gives a near-zero baseline. MinSteer produces the largest reverse-arm mean and increases measured toxicity in every seed, but generation can degrade under reverse injection (Appendix O). GCAV [Zhang et al., 2025] provides a separate positive-only reference: with 100 labeled pairs, per-layer concept probes, and adaptive token-level intervention, its mean toxicity is 0.00284. Its data budget and intervention rule differ, so it is not included in the matched five-example ranking.

Same-model politeness comparison. We compare all three methods on Swallow-8B using the same 99 Stanford source requests, rewrite template, layer 15 `resid_pre` hook, greedy decoder, unit-norm vectors, and fixed signed grid $\alpha \in \{-1, 0, +1\}$. Unlike the original in-place intervention, all three methods here add the vector to a copied input of `input_layernorm`, leaving the residual skip connection unchanged; absolute rates should be compared within Table 32. MinSteer uses one self-rewrite contrast, whereas CAA and RepE use five independently prompted matched scenarios. The three unsteered runs produce identical text, and all outputs pass the validity check. Source similarity is the mean cosine between L2-normalized source and output embeddings from `sentence-transformers/all-MiniLM-L6-v2`.

Table 32: Same-model, norm-matched politeness comparison. Values are ordered by $\alpha = -1/0/+1$. The stored direction is casual minus formal, so -1 is the predeclared formal sign.

Method	$P(\text{formal})$	Source similarity
MinSteer (1 contrast)	0.9244 / 0.9174 / 0.9209	0.8091 / 0.8098 / 0.8101
CAA (5 scenarios)	0.9132 / 0.9174 / 0.9177	0.8095 / 0.8098 / 0.8055
RepE (5 scenarios)	0.9088 / 0.9174 / 0.9228	0.8114 / 0.8098 / 0.8132

At the formal sign, MinSteer changes formality by $+0.0069$ [95% CI $+0.0013, +0.0137$], with a source-similarity change of -0.0007 [$-0.0046, +0.0034$]. CAA and RepE change formality by -0.0042 [$-0.0108, +0.0016$] and -0.0087 [$-0.0252, +0.0036$]. These are prompt-paired bootstrap intervals (10,000 resamples). The small formal-side effect is consistent with useful contrast construction at low data, while method-specific differences remain. None of the unit-norm directions gives a significant casual-side effect at $+1$.

Official Causal-Steer reference. We also evaluate the released Causal-Steer [Wang et al., 2026] formality vector in its native Qwen2.5-7B-Instruct setting on the same 99 source requests. We

retain the official layers 18–23 MLP-output hooks and per-layer normalization (repository commit `e84a4cd`), use greedy decoding with 60 new tokens, and evaluate the fixed grid $\alpha \in \{-2.5, 0, +2.5\}$.

Table 33: Official Causal-Steer in its native model and hook setting. Differences are prompt-paired changes from $\alpha = 0$, with bootstrap 95% CIs.

α	$P(\text{formal})$	Formality change [95% CI]	Source similarity	Valid rate
-2.5	0.8822	-0.0464 $[-0.0740, -0.0231]$	0.8504	1.00
0	0.9286	—	0.8526	1.00
$+2.5$	0.9515	$+0.0229$ $[+0.0102, +0.0372]$	0.8460	1.00

Causal-Steer gives bidirectional formality control in its released setting (Table 33). Because the model, vector source, hook, and intervention scale differ from MinSteer's Swallow experiment, this is a native-method reference rather than a same-model ranking.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the extraction method and delimit the evidence to the tested models, tasks, and protocols. Sections 5.6–5.7 distinguish transcript effects from cache execution and report the human-rated quality trade-off.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 discusses model and protocol coverage, proxy metrics, output quality, and dual use. Section 4.1 and Appendices O–P discuss mixed direction components and sensitivity to extraction seeds and sample size; Appendix N notes the small annotator panel and co-author participation.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results. Appendix C states an informal interpretation and its assumptions; it is not a theorem or proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 4 defines extraction and intervention. Appendix B and the task-specific appendices report model identifiers, prompts, readouts, seeds, decoding, and evaluation. Appendices L–Q distinguish the experimental protocols; Appendix N describes human evaluation.

Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: Code, data-access instructions, and reproduction commands are available at `https://github.com/Liumx2020/MinSteer`. The repository documents experiment coverage and materials that are not released.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Sections 4–5 and the experimental appendices give extraction sizes, model identifiers, intervention sites and strengths, decoding, and metrics. No model weights are trained by MinSteer. Appendix Q separately identifies the native settings of external methods.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The studies report extraction-seed standard deviations, prompt-paired bootstrap 95% confidence intervals, paired Wilcoxon tests with the stated multiplicity correction, or an exact seed-level sign-flip test, as appropriate. Appendices L–Q specify the unit of comparison and distinguish seed variability from prompt uncertainty.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix B reports GPU types and memory, approximate run times from retained logs, and an estimated overall runtime for all reported experiments. It also notes additional compute for preliminary and failed runs.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The human evaluation was limited to rating model outputs. Participation was voluntary, the non-author annotators were compensated, and all annotators were informed of potentially offensive content and could stop at any time. Appendix N documents compensation, data handling, and participant protections.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6 describes potential uses for model analysis and behavior diagnostics, together with misuse risks from toxic-style amplification and sentiment or register manipulation.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: No new pretrained model or scraped dataset is released. Section 6 discusses the dual-use risks of activation steering and the limits of the toxicity diagnostic.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [No]

Justification: Appendix B lists asset sources and available licenses. Some upstream releases do not specify a license, so complete licensing information could not be verified. Original model weights and datasets are not redistributed.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The public repository releases the original MinSteer code under the MIT License and includes installation instructions, experiment commands, and a paper-to-code coverage map. Appendix N provides the human-rating instructions. No new pretrained model or benchmark dataset is released.

Guidelines:

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

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

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

Answer: [Yes]

Justification: Appendix N provides the rating instructions and casual-scale clarification, and describes recruitment, voluntary participation, the JPY 8,000 payment and 5–6 hours spent by each non-author annotator, and no additional payment to the co-author.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: Annotators rated model outputs; the study did not investigate the annotators themselves. The annotation procedure followed applicable institutional ethics requirements. Appendix N describes the annotation conditions and participant protections.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs generate the paired continuations and are the objects of residual-state extraction and intervention. Model identifiers and decoding settings are specified in Appendix B, with protocol-specific additions in the subsequent appendices.

    Guidelines:

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