# JAM: JUST A MOVE — BOUNDARY-GUIDED MINIMUM-DISTANCE STEERING FOR CONTROLLABLE TEXT GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Inference-time representation steering typically relies on a globally tuned intervention strength, even though different inputs may require different amounts of control. We introduce JAM (*Just A Move*), which formulates latent steering as *constraint satisfaction* rather than score maximization. A learned linear probe defines a desired half-space: if an input-specific first-pass representation already satisfies the learned constraint, JAM abstains; otherwise, it computes in closed form the minimum-$\ell_2$ displacement of that representation required to reach the decision boundary. The resulting prompt- and decode-side components are then used as steering signals during a controlled second generation pass.

The learned control surfaces generalize to strictly prompt-disjoint test data, with AUROC between $0.858$ and $0.934$. Under a matched HHH evaluation across Meta-Llama-3-8B and Mistral-7B-Instruct-v0.2, JAM does not reduce Base ROUGE-2 in any of the six settings while keeping perplexity close to the unmodified model; fixed-strength CAA substantially reduces reference overlap in several honest/helpful settings. Controlled distance sweeps show no consistent reference-overlap benefit from moving $20\%$–$50\%$ beyond the analytically defined boundary distance, supporting the boundary as a tuning-free conservative calibration point. JAM also reduces measured toxicity in the reported RealToxicityPrompts evaluation, with a grammaticality trade-off. The boundary computation itself adds only $1.1$ ms, and examples that already satisfy the learned constraint avoid a second generation pass. Together, these results support boundary distance as a simple input-specific rule for deciding whether and how strongly to steer.

## 1 INTRODUCTION

Large language models can generate fluent text, but many applications require generation to satisfy a desired attribute such as helpfulness, honesty, harmlessness, or low toxicity. Prompting is simple but can be unreliable, while training- or fine-tuning-based approaches introduce additional optimization cost and usually specialize a model to a particular control objective Yang & Klein (2021); Li et al. (2024b); Alhafni et al. (2024). Inference-time control avoids updating the base model, but representation-steering methods commonly introduce a different design choice: how strongly should the model be steered? Fixed activation shifts use a globally selected coefficient, whereas iterative methods repeatedly score or update the generation trajectory Dathathri et al. (2020); Mireshghallah et al. (2022); Feng et al. (2024); Panickssery et al. (2023). A single global strength is inherently input-agnostic: it can move an already-satisfactory representation unnecessarily, overshoot an example near the desired region, or be insufficient for an example farther away.

JAM takes a different view. We formulate steering as constraint satisfaction rather than score maximization. A linear latent probe defines a desired half-space, and its signed margin answers two questions in the first-pass representation space: whether steering is needed and, if so, the minimum displacement required to reach the learned boundary. This leads to the principle illustrated in fig. 2:

*detect* $\rightarrow$ *intervene only if needed* $\rightarrow$ *calibrate by boundary distance*.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Thus, the learned boundary provides both an intervention criterion and an input-specific steering magnitude. JAM replaces a globally tuned steering-strength rule with a closed-form feasibility update and avoids modifying examples whose first-pass representations already satisfy the learned constraint.

We use the learned probe as an operational control surface rather than as a complete semantic model of the attribute. Its signed margin provides a reproducible rule for deciding whether to intervene and for computing the input-specific probe-space displacement. The resulting displacement is then mapped to autoregressive steering, whose downstream behavior is evaluated empirically.

Our contributions are:

1. **Control as constraint satisfaction.** We formulate latent steering around a learned linear attribute constraint, using the same signed margin to determine whether intervention is needed and to calibrate its input-specific magnitude.
2. **Closed-form selective minimum-distance steering.** For a negative-margin first-pass representation, we derive the exact minimum-$\ell_2$ probe-space displacement required to reach the learned boundary; for a positive-margin representation, JAM abstains. The resulting prompt- and decode-side components are used to steer a second generation pass without fine-tuning the base LLM.
3. **Mechanism-grounded evaluation.** Strictly prompt-disjoint probe and HHH evaluations test generalization and reference-overlap preservation, while controlled distance and policy analyses examine the minimum-distance calibration rule. Toxicity, pairwise preference, and latency measurements provide complementary behavioral and systems evidence.

## 2 Empirical Motivation: Probe Geometry for Selective Steering

Before defining the intervention rule, we examine whether the chosen hidden-state representation exposes a stable enough linear geometry to support margin-based control. These diagnostics motivate the representation and layer used by JAM.

### 2.1 Generation states become increasingly differentiated

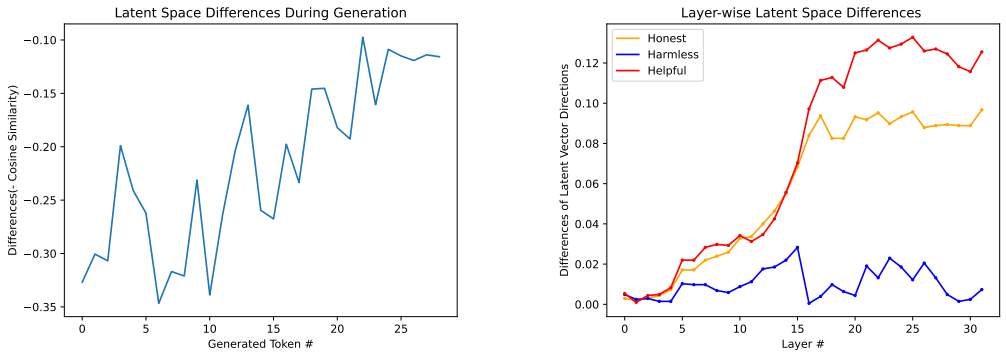
For $N$ prompts, let $h^m_{i|l}$ denote the hidden state of prompt $i$ at decoding step $m$ and layer $l$. We measure pairwise latent divergence using negative cosine similarity,

$$d(h^m_{i|l}, h^m_{j|l}) = -\frac{h^m_{i|l} \cdot h^m_{j|l}}{\|h^m_{i|l}\| \, \|h^m_{j|l}\|}. \tag{1}$$

As shown in fig. 1, hidden states become more differentiated over decoding under this negative-cosine-similarity diagnostic, while the layer-wise analysis shows a pronounced change in the later-middle layers. This motivates two design choices used throughout the paper: (i) the probe representation combines the final prefill state with a state from the completed first-pass response, so detection can use both prompt context and realized generation behavior; and (ii) the reported experiments use layer 20. A separate intervention-layer sweep on RealToxicityPrompts shows that steering remains effective across several mid-to-late layers (15-30), with layer 20 providing a competitive toxicity-fluency trade-off while matching the probe-training layer; see table 6. We therefore treat layer 20 as an empirical operating point rather than a universal "semantic layer."

### 2.2 Linear probes define the control surfaces

We split the HHH data by source-prompt identity so that all responses associated with the same prompt remain in one partition. Using 500 balanced training responses and 200 validation responses for probe selection, we evaluate on a strictly prompt-disjoint test set. As shown in table 1, the resulting boundaries achieve AUROC between $0.858$ and $0.934$, with zero prompt overlap across train, validation, and test. JAM uses these probes to define the signed margins and decision surfaces used for steering.




(a) Decoding-step latent divergence

(b) Layer-wise latent divergence

Figure 1: Latent-space diagnostics used to motivate the representation and layer choice. The reported quantity is negative cosine similarity. Harmless and helpful use HH-RLHF (Bai et al., 2022); honest uses TruthfulQA (Lin et al., 2022).

Table 1: Strictly prompt-disjoint probe evaluation. Training and validation contain 250 and 100 prompt groups, respectively. Majority-class accuracy is 0.50 on all three balanced test sets.

| Attribute | Accuracy | F1 | AUROC |
|---|---|---|---|
| Harmless | .855 | .859 | .934 |
| Helpful | .785 | .785 | .858 |
| Honest | .831 | .834 | .908 |

### 2.3 Why boundary distance?

For a negative-margin representation, the distance to the learned boundary provides an input-specific steering magnitude, while a positive-margin representation requires no update. This motivates the central design question behind JAM: whether steering beyond this analytically defined distance consistently improves downstream behavior. We test this directly in section 4.3.

## 3 Boundary-Guided Minimum-Distance Steering

We formalize steering-magnitude calibration as a feasibility problem in probe space. For a fixed first-pass representation, the learned boundary determines whether an update is needed and, if so, the smallest Euclidean displacement that reaches the desired half-space.

### 3.1 Latent representation and attribute boundary

For a prompt $p_i$ with $K$ input tokens, let the base model generate $M$ output tokens. At layer $l$, we use the hidden state of the final prefilling token, $h^K_{i|l}$, together with the hidden state of the final generated token, $h^{K+M}_{i|l}$, to form

$$r_i = [h^K_{i|l}; h^{K+M}_{i|l}]. \tag{2}$$

The first component summarizes the prompt after prefilling, while the second reflects the completed first-pass response. The representation diagnostic in section 2 and the intervention-layer sweep in table 6 motivate using layer 20 in the reported experiments; we treat this as an empirical design choice rather than a universal property of LLMs.

For each attribute $A$, we fit a linear classifier

$$f_A(r) = w_A^\top r + b_A, \tag{3}$$

where $f_A(r) \geq 0$ denotes the desired side of the learned boundary. The probe is deliberately lightweight: its purpose is not to provide a complete semantic model of the attribute, but to expose

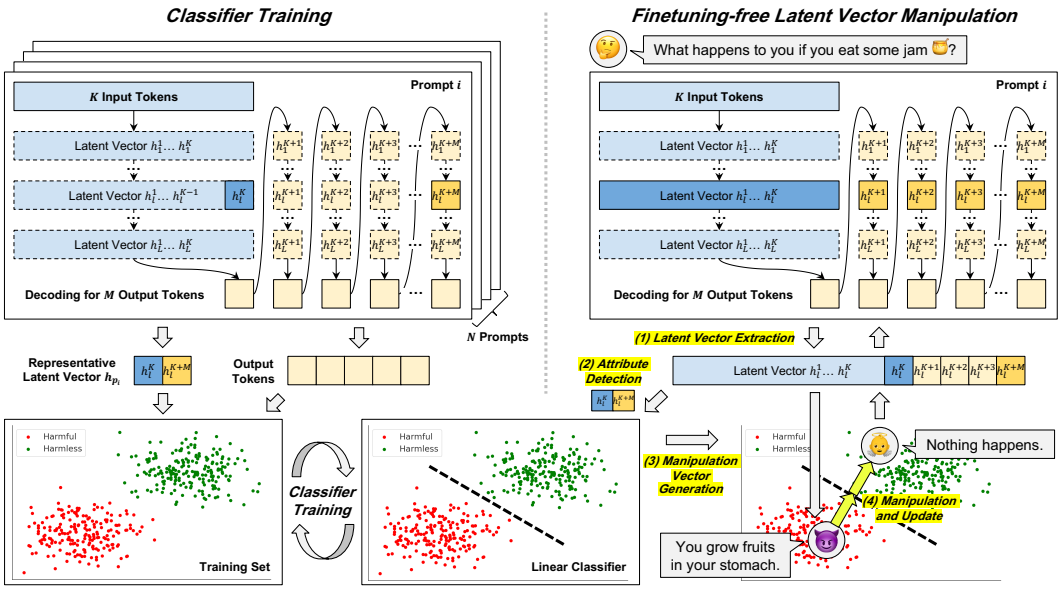



Figure 2: Overview of JAM. A lightweight linear probe is trained for a target attribute. At inference, an initial generation provides the representation used to evaluate the signed margin. Positive-margin examples are returned unchanged. For negative-margin examples, JAM computes the minimum probe-space displacement to the learned boundary in closed form, partitions it into prompt- and decode-side components, and uses these components to steer a second generation pass.

an explicit surface on which detection and intervention can share the same geometry. Linear-probe performance is reported in table 1.

For the HHH experiments, labels are obtained using the proxy-labeling pipeline: generated answers are compared against correct/preferred and incorrect/dispreferred exemplars using T5 Raffel et al. (2020). We use the prompt-disjoint train/validation/test partition described in Section 2.2; Table 1 reports test performance. These labels define the operational control boundary used by JAM.

### 3.2 Minimum-distance boundary projection

Let $m_i = f_A(r_i)$ be the signed margin of an input-specific representation. If $m_i \geq 0$, the representation already lies in the desired half-space and JAM applies no perturbation. If $m_i < 0$, we seek the shortest movement that reaches the learned decision boundary:

$$\begin{aligned} \Delta r_i^* &= \arg\min_{\Delta r} \ \|\Delta r\|_2 \\ \text{s.t.} \quad & w_A^\top (r_i + \Delta r) + b_A \geq 0. \end{aligned} \tag{4}$$

The normal direction of the decision hyperplane is $w_A$. We therefore define the corresponding unit direction as

$$u_A = \frac{w_A}{\|w_A\|_2}. \tag{5}$$

For a negative-margin representation, its perpendicular distance to the decision hyperplane is

$$\alpha_i^* = \frac{-f_A(r_i)}{\|w_A\|_2}. \tag{6}$$

The minimum-norm perturbation is therefore

$$\boxed{\Delta r_i^* = \alpha_i^* u_A = \frac{-f_A(r_i)}{\|w_A\|_2^2} w_A}. \tag{7}$$

For a negative-margin representation, $\alpha_i^*$ is exactly its perpendicular Euclidean distance to the learned decision hyperplane, while $u_A$ specifies the shortest direction toward that plane. For a representation already in the positive half-space, the minimum feasible update is zero and JAM applies no intervention. In implementation, an arbitrarily small numerical offset can be added along $u_A$ if strict placement inside the positive half-space is desired.

**Optimality of the boundary projection.** For a fixed linear probe, eq. (7) is the exact minimum-$\ell_2$ solution of eq. (4). Any feasible perturbation for a negative-margin representation must satisfy

$$w_A^\top \Delta r \geq -f_A(r_i). \tag{8}$$

By Cauchy-Schwarz, $w_A^\top \Delta r \leq \|w_A\|_2 \|\Delta r\|_2$, so every feasible perturbation obeys

$$\|\Delta r\|_2 \geq \frac{-f_A(r_i)}{\|w_A\|_2} = \alpha_i^*. \tag{9}$$

Equality is attained when $\Delta r$ is parallel to $w_A$, yielding $\Delta r_i^* = \alpha_i^* u_A$. Therefore, $\alpha_i^*$ is exactly the shortest Euclidean distance from $r_i$ to the learned decision plane.

This input-specific calibration also distinguishes JAM from fixed-strength activation steering. A single global coefficient cannot in general coincide with the boundary distance of every representation: it may undershoot examples farther from the decision plane and overshoot examples already close to it. By contrast, JAM computes $\alpha_i^*$ directly from the observed signed margin, eliminating a separate global steering-strength choice for the boundary-distance rule.

This optimality is defined for the fixed first-pass representation in probe space. The resulting prompt- and decode-side components serve as steering signals for the second autoregressive pass, whose downstream behavior is evaluated empirically.

### 3.3 From the probe perturbation to autoregressive generation

After computing the minimum-distance movement $\Delta r_i^* = \alpha_i^* u_A$, we map this representation-space perturbation back to autoregressive generation. Because $r_i$ concatenates a prompt-side and an output-side hidden state, we partition

$$\Delta r_i^* = [\Delta h_{i,\mathrm{pre}}^*; \Delta h_{i,\mathrm{dec}}^*]. \tag{10}$$

When intervention is required, the second generation pass adds $\Delta h_{i,\mathrm{pre}}^*$ to the selected-layer hidden state of the final prefilling token. During decoding, $\Delta h_{i,\mathrm{dec}}^*$ is added to the current-token hidden state at the same layer. This maps the two components of the calibrated probe direction back to the corresponding prefilling and decoding stages. Algorithm 1 summarizes inference.

**Algorithm 1** JAM inference for attribute $A$

**Input:** prompt $p$, base model $\mathcal{M}$, probe $(w_A, b_A)$
Generate first-pass answer $y^{(0)}$ and obtain $h^K$, $h^{K+M}$
$r \leftarrow [h^K; h^{K+M}]$, $\quad m \leftarrow w_A^\top r + b_A$
**if** $m \geq 0$ **then**
  **return** $y^{(0)}$ *// already satisfies learned constraint*
**end if**
$u_A \leftarrow w_A / \|w_A\|_2$
$\alpha^* \leftarrow -m / \|w_A\|_2$
$\Delta r^* \leftarrow \alpha^* u_A$
Split $\Delta r^* = [\Delta h_{\mathrm{pre}}^*; \Delta h_{\mathrm{dec}}^*]$
Generate $y^{(1)}$ while injecting $\Delta h_{\mathrm{pre}}^*$ at prefill and $\Delta h_{\mathrm{dec}}^*$ during decoding
**return** $y^{(1)}$

# 4 EXPERIMENTS

## 4.1 SETUP AND EVALUATION QUESTIONS

We evaluate Meta-Llama-3-8B (AI@Meta, 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). Harmless and helpful evaluations use HH-RLHF (Bai et al., 2022); honesty uses TruthfulQA (Lin et al., 2022); toxicity control uses RealToxicityPrompts (Gehman et al., 2020).

For the primary HHH comparison, we use frozen prompt-disjoint manifests and evaluate all methods with the same held-out prompt IDs, positive references, decoding configuration, and scorer. Generation uses greedy decoding for at most 200 new tokens with model-EOS stopping. ROUGE-2 is the arithmetic mean of per-example F1 from `rouge-score==0.1.2`, with stemming enabled and no external preprocessing. Perplexity is computed with one common implementation across compared methods. Fixed-strength baselines are selected only on validation prompts and frozen before accessing the test set.

We report two complementary evidence sets. The matched evaluation is the primary basis for cross-method HHH claims. A separate set of controlled mechanism analyses evaluates intervention distance, boundary policy, and pairwise preference under the earlier experimental protocol. Those analyses are used only for within-experiment contrasts; absolute metric values are not compared across the two protocols. This separation lets the matched evaluation establish method-level quality preservation while the controlled ablations test the boundary-distance calibration hypothesis.

## 4.2 MATCHED HHH EVALUATION: PRESERVING QUALITY UNDER STEERING

Table 2 compares the unmodified model, CAA, and JAM under the common prompt-disjoint protocol. The central question in this matched comparison is whether the complete steering methods preserve reference overlap and fluency. CAA uses a contrastive mean direction with one validation-selected global steering magnitude, whereas JAM uses a probe-normal direction and an input-specific boundary distance. Because the methods differ in both direction construction and intervention policy, the table evaluates the complete methods rather than attributing the difference to magnitude calibration alone.

Table 2: Matched prompt-disjoint HHH comparison. Each cell reports ROUGE-2 / PPL; higher ROUGE-2 and lower PPL are better. All methods use the same held-out prompt IDs, references, generation configuration, and evaluation implementation.

| | Mistral-7B-Instruct-v0.2 | | | Meta-Llama-3-8B | | |
|---|---|---|---|---|---|---|
| Method | Harmless | Honest | Helpful | Harmless | Honest | Helpful |
| Base | .0234 / 45.52 | **.0808** / 32.50 | .0528 / 40.24 | .0328 / 21.86 | .0583 / 16.24 | **.0582** / 32.53 |
| CAA | .0272 / 57.27 | .0017 / 25.71 | .0016 / 9.09 | .0278 / 17.79 | .0000 / 4.86 | .0004 / 1.74 |
| JAM | **.0285** / 43.73 | **.0808** / 32.53 | **.0547** / 41.68 | **.0334** / 20.96 | **.0676** / 14.74 | **.0582** / 32.53 |

Across all six model-attribute settings, JAM does not reduce Base ROUGE-2 while keeping perplexity close to the unmodified model. In contrast, fixed-strength CAA substantially reduces reference overlap in several honest/helpful settings. These results demonstrate reference-overlap preservation for the complete JAM procedure; because CAA differs in both direction construction and intervention policy, the comparison does not isolate the effect of adaptive magnitude.

## 4.3 MECHANISM ANALYSIS: DOES MORE STEERING HELP?

The boundary-distance rule uses the learned decision surface as an analytically defined calibration point rather than assuming that larger probe margins are preferable. We test whether deliberately steering beyond this distance consistently improves downstream generation quality.

**Post-boundary distance sweep.** If larger post-boundary margins were intrinsically beneficial, increasing the intervention from $\alpha^*$ to $1.2\alpha^*$ and $1.5\alpha^*$ should systematically improve generation quality. Table 3 does not show this pattern: additional movement is non-monotonic across ROUGE-2

and BLEURT and can worsen perplexity. Thus, moving beyond the boundary distance does not provide a consistent benefit across the reported settings.

Table 3: HHH mechanism study and intervention-distance sweep. Higher is better for ROUGE-2 (R2) and BLEURT (BL); lower is better for PPL. Bold marks the best result within each model-attribute-metric block.

| | Harmless | | | Honest | | | Helpful | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | R2↑ | BL↑ | PPL↓ | R2↑ | BL↑ | PPL↓ | R2↑ | BL↑ | PPL↓ |
| Llama3 | .205 | .485 | 37.73 | .227 | .441 | **3.60** | .230 | .580 | **14.43** |
| + JAM ($\alpha^*$) | **.250** | **.530** | 36.13 | **.318** | **.463** | 3.72 | **.300** | .550 | 17.11 |
| + JAM ($1.2\alpha^*$) | .245 | .525 | **36.10** | .290 | .441 | 3.80 | .250 | **.650** | 19.70 |
| + JAM ($1.5\alpha^*$) | .245 | **.530** | 36.25 | .304 | .450 | 4.00 | .250 | .600 | 23.80 |
| Mistral | .105 | .545 | **78.20** | .324 | .589 | 4.16 | **.240** | .590 | 25.68 |
| + JAM ($\alpha^*$) | .155 | .575 | 84.10 | **.355** | **.649** | **4.04** | **.240** | **.640** | 26.60 |
| + JAM ($1.2\alpha^*$) | .155 | .575 | 88.54 | .338 | .632 | 4.08 | .210 | .620 | **24.87** |
| + JAM ($1.5\alpha^*$) | **.160** | **.585** | 88.50 | .314 | .605 | 4.36 | .190 | .600 | 29.19 |

**Boundary-policy factorial analysis.** The boundary-conditioned policy combines input-dependent magnitude for negative-margin examples with abstention on positive-margin examples. The $2 \times 2$ analysis in table 8 isolates these factors. Their marginal ROUGE-2 effects are small, mixed in sign, and none survives the stated multiple-comparison correction. ROUGE-2 is therefore largely insensitive to the two policy factors within this study, consistent with the distance sweep's finding that extra movement does not yield systematic reference-overlap gains. We use $\alpha_i^*$ as a conservative, tuning-free calibration point rather than treating larger probe margins as intrinsically better.

### 4.4 Complementary pairwise attribute evaluation

Reference-overlap metrics do not certify whether an answer better satisfies harmlessness, honesty, or helpfulness. Table 4 therefore reports GPT-4 pairwise preference between JAM and attribute prompting as a non-reference signal. For Llama3, JAM receives more wins than losses on both reported attributes, whereas the Mistral comparisons are dominated by draws. Helpful is not used for comparative claims because the judge returned "Neither" for most pairs. The judging protocol is described in section A.5.

Table 4: GPT-4 pairwise preference between JAM and attribute prompting. Win/Loss/Draw are percentages from JAM's perspective.

| Model | Attribute | Win↑ | Loss↓ | Draw |
|---|---|---|---|---|
| Llama3 | Harmless | 40.3 | 31.0 | 28.7 |
| Llama3 | Honest | 50.0 | 23.8 | 26.2 |
| Mistral | Harmless | 16.2 | 15.8 | 68.0 |
| Mistral | Honest | 20.0 | 20.0 | 60.0 |

### 4.5 Transfer to toxicity control

Table 5: Toxicity control on RealToxicityPrompts. Lower toxicity/PPL and higher grammar are better.

| Method | Toxicity↓ | Grammar↑ | PPL↓ |
|---|---|---|---|
| Llama3 | .122 | **.763** | 96.303 |
| + JAM | **.089** | .737 | **94.329** |
| Mistral | .011 | **.642** | 135.908 |
| + JAM | **.010** | .566 | **121.358** |

As shown in table 5, JAM lowers measured toxicity and perplexity on both models, while grammaticality decreases. We therefore present toxicity control as a trade-off rather than claiming that all collateral metrics are preserved.

**Robustness checks.** The observed steering behavior is not confined to a single intervention layer. As shown in table 6, several mid-to-late layers achieve low measured toxicity, with layer 20 providing a competitive toxicity–fluency trade-off while matching the probe-training layer.

Table 6: Intervention-layer sensitivity on 100 Mistral RealToxicityPrompts examples. Effective steering persists across mid-to-late layers.

| Layer | Toxicity↓ | PPL↓ |
|---|---|---|
| 5 | .0054 | 76.92 |
| 10 | .0223 | 43.79 |
| 15 | .0019 | 44.51 |
| 20 | .0038 | **37.42** |
| 25 | .0048 | 39.43 |
| 30 | .0029 | 45.05 |

The TruthfulQA collateral-quality check further shows that JAM retains zero measured trigram repetition and zero unreadable outputs on both models, while diversity remains essentially unchanged (Llama3: $0.98 \rightarrow 0.97$; Mistral: $1.00 \rightarrow 1.00$; table 10).

### 4.6 Inference Efficiency

Table 7 separates the cost of boundary computation from autoregressive generation. Every example executes the first generation used for representation extraction and attribute detection. Positive-margin examples then terminate, whereas negative-margin examples additionally compute the intervention and execute the controlled generation pass.

Selectivity has a direct systems consequence. As shown in table 7, evaluating the boundary and computing the intervention add only $1.1$ ms, so positive-margin examples terminate at essentially baseline cost. Negative-margin examples pay for one additional controlled generation and reach $1.85\times$ baseline latency. Thus, the dominant cost of JAM is not the control computation itself but whether a second decoding pass is required. Let $q$ denote the fraction of examples whose first-pass margin is negative. The measured stage times imply

$$\mathbb{E}[T_{\text{JAM}}(q)] \approx 1.2206 + 1.0305q \quad \text{s.} \tag{11}$$

This expression separates the fixed cost of detecting whether control is needed from the intervention-dependent second-pass cost, making the computational value of abstention explicit.

## 5 Related Work

**Controlled generation.** PPLM (Dathathri et al., 2020), FUDGE (Yang & Klein, 2021), NADO (Meng et al., 2022), Mix & Match (Mireshghallah et al., 2022), and PREADD (Pei et al., 2023) control generation through gradients, future discriminators, energy formulations, or adaptive decoding. FreeCtrl (Feng et al., 2024) and DATG (Liang et al., 2024) provide additional inference-time control mechanisms. JAM differs in using a learned latent boundary simultaneously as a detector and a closed-form constraint, rather than repeatedly optimizing token probabilities.

**Representation steering.** Representation Engineering (Zou et al., 2023), Inference-Time Intervention (Li et al., 2024a), and Contrastive Activation Addition (CAA) (Panickssery et al., 2023) show that internal representations can be manipulated to influence model behavior through lightweight inference-time interventions. These methods motivate the use of latent steering directions, while the intervention strength is typically chosen separately from the direction itself. JAM instead uses the signed margin of a learned linear boundary to determine both whether steering is applied and the input-specific distance used for steering. Its contribution is therefore an explicit boundary-conditioned calibration rule rather than a new semantic steering direction.

Table 7: Inference-time breakdown on Meta-Llama-3-8B. Control operations in JAM are negligible relative to generation. Positive-margin examples terminate after the first pass; intervention examples additionally execute controlled generation.

| Method / stage | Time (s) | Relative cost |
|---|---|---|
| Original generation | 1.2200 | 1.00× |
| JAM: first generation / extraction | 1.2200 | – |
| JAM: attribute detection | 0.0006 | – |
| JAM: intervention computation | 0.0005 | – |
| JAM: controlled generation | 1.0300 | – |
| JAM, positive-margin path | 1.2206 | 1.00× |
| JAM, intervention path | 2.2511 | 1.85× |

## 6 Limitations and Discussion

The current study has three practical limitations. First, the learned probes are operational proxies built from proxy labels; neither the probe margin nor reference-based HHH metrics should be interpreted as semantic ground truth. The GPT-4 pairwise evaluation provides a complementary signal but does not replace direct human assessment. Second, the experiments cover two 7-8B model families, so broader architectural generalization remains to be established. Third, the current representation depends on a completed first-pass response, making negative-margin cases inherently two-pass.

The minimum-distance guarantee is defined for the learned probe geometry of the fixed first-pass representation; alternative representation metrics or trajectory-level objectives are outside the current formulation. The distance sweep provides an empirical check on the chosen calibration rule: moving $20\%$ or $50\%$ farther does not yield a consistent reference-overlap benefit across the reported settings. Because JAM is a general boundary-conditioned steering mechanism, its behavior depends on the target represented by the learned probe. The same mechanism could therefore be instantiated for undesirable objectives; the harmlessness and toxicity results in this work should not be interpreted as making the steering mechanism intrinsically safety-preserving.

## 7 Conclusion

We introduced JAM, a boundary-guided formulation of representation steering as constraint satisfaction. A learned linear probe determines whether steering is needed and, for negative-margin representations, provides the exact minimum-$\ell_2$ probe-space distance to the learned boundary. The resulting prompt- and decode-side components are then used for autoregressive steering.

Strict prompt-disjoint evaluation shows that the learned boundaries generalize to unseen source prompts. Under the matched HHH protocol, JAM does not reduce Base ROUGE-2 in any of the six settings while keeping perplexity close to the unmodified model. Controlled distance sweeps show no consistent benefit from steering $20\%$–$50\%$ beyond the boundary distance, and the toxicity experiments show numerically reduced measured toxicity with a grammaticality trade-off.

The resulting design principle is simple: use the learned margin to determine whether steering is needed and, when it is, derive the steering magnitude directly from the distance to the learned boundary. This replaces a global steering-strength choice with an input-specific, analytically defined calibration rule.

## AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing and improving clarity, as well as to provide feedback on the presentation of the methodology, experiments, and interpretation of results. The authors independently designed the proposed method, conducted the experiments, analyzed the results, and verified all technical claims and AI-assisted revisions.

## REPRODUCIBILITY STATEMENT

All HHH experiments use frozen, source-prompt-disjoint train/validation/test manifests with zero prompt overlap across splits. Compared methods are evaluated on the same held-out prompts, references, decoding settings, and metric implementation. Fixed-strength baseline hyperparameters are selected on the validation split and frozen before test evaluation, while JAM computes its intervention magnitude analytically from each example's probe margin. We retain the prompts, references, generations, model and method configurations, and evaluation outputs for all reported experiments.

## REFERENCES

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. Personalized text generation with fine-grained linguistic control. *arXiv preprint arXiv:2402.04914*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation, 2020. URL https://arxiv.org/abs/1912.02164.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, and Kezhi Mao. Freectrl: Constructing control centers with feedforward layers for learning-free controllable text generation. *arXiv preprint arXiv:2406.09688*, 2024.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, 2024a. URL https://arxiv.org/abs/2306.03341.

Wendi Li, Wei Wei, Kaihe Xu, Wenfeng Xie, Dangyang Chen, and Yu Cheng. Reinforcement learning with token-level feedback for controllable text generation. *arXiv preprint arXiv:2403.11558*, 2024b.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. Controlled text generation for large language model with dynamic attribute graphs. *arXiv preprint arXiv:2402.11218*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35:28125–28139, 2022.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. Mix and match: Learning-free controllable text generation using energy language models, 2022. URL https://arxiv.org/abs/2203.13299.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Jonathan Pei, Kevin Yang, and Dan Klein. Preadd: prefix-adaptive decoding for controlled text generation. *arXiv preprint arXiv:2307.03214*, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*, 2021.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

# A Additional Experimental Details

## A.1 Evaluation Protocol Scope

The matched HHH comparison in table 2 uses the final frozen prompt-disjoint evaluation manifests. All methods in that comparison are evaluated on the same held-out prompt IDs, references, generation configuration, and metric implementation.

The intervention-distance sweep, boundary-policy analysis, and GPT-4 pairwise evaluation are conducted under a separate mechanism-study evaluation. These experiments are used to compare intervention settings within the corresponding study.

## A.2 Intervention-Distance Sweep

The mechanism-study distance sweep includes Base and JAM at $\alpha_i^*$, $1.2\alpha_i^*$, and $1.5\alpha_i^*$, as reported in table 3. For each negative-margin example, $\alpha_i^*$ is the perpendicular distance from the representation to the learned decision plane along $u_A = w_A/\|w_A\|_2$. The two larger settings move farther along the same probe-normal direction after reaching the boundary.

Across the reported settings, $\alpha_i^*$ is often competitive with or better than the larger intervention distances, while improvements from $1.2\alpha_i^*$ and $1.5\alpha_i^*$ occur on different metrics and do not form a consistent cross-model or cross-attribute pattern.

## A.3 $2 \times 2$ Boundary-Policy Analysis

To separate input-dependent magnitude from selective intervention, we use a $2 \times 2$ factorial design crossing steering magnitude (fixed versus input-dependent) and gating (steer all examples versus steer only negative-margin examples). All four conditions use the same learned probe-normal direction $u_A = w_A/\|w_A\|_2$ and the same intervention locations. The fixed-magnitude condition uses a single magnitude selected on the validation split and then frozen before test evaluation, while the input-dependent condition uses a per-example magnitude determined by the boundary-based rule. Full JAM combines input-dependent magnitude with boundary gating.

Table 8 reports the marginal ROUGE-2 effect and corresponding $p$-value for each factor across model–attribute settings. The effects are small and mixed in sign, and none survives the stated

Bonferroni correction. We therefore use this analysis to characterize the two policy factors rather than to claim that either independently improves ROUGE-2.

Table 8: $2 \times 2$ boundary-policy analysis from the mechanism-study HHH experiments. Values are marginal changes in ROUGE-2; $p$-values are from the corresponding factorial analysis.

| Model × attribute | Adaptive magnitude ΔR2, negative margin | Abstention ΔR2, positive margin |
|---|---|---|
| Llama3 × harmless | $+0.000665$ ($p = .370$) | $-0.000024$ ($p = .969$) |
| Llama3 × helpful | $+0.000213$ ($p = .931$) | $-0.001542$ ($p = .456$) |
| Llama3 × honest | $0.000000$ ($p = 1.000$) | $-0.000715$ ($p = .032$) |
| Mistral × harmless | $0.000000$ ($p = 1.000$) | $+0.000329$ ($p = .641$) |
| Mistral × helpful | $+0.001256$ ($p = .037$) | $-0.000391$ ($p = .807$) |
| Mistral × honest | $-0.000042$ ($p = .723$) | $+0.000259$ ($p = .044$) |

### A.4 Corrected PREADD Diagnostic

A transferred PREADD setup was unstable with Mistral/Llama-style tokenizer and cached-logit handling. We therefore reran both PREADD and JAM in direct-model mode with the model-native tokenizer on the same 100 Mistral RealToxicityPrompts examples. Table Table 9 reports this corrected toxicity–fluency diagnostic.

Table 9: Corrected direct-model comparison on 100 Mistral RealToxicityPrompts examples. Both methods are evaluated under the same model-native tokenizer and direct-model configuration. Lower is better for toxicity and PPL.

| Method | Toxicity↓ | PPL↓ |
|---|---|---|
| PREADD | $0.0987 \pm 0.177$ | 70.8 |
| JAM | $\mathbf{0.0038 \pm 0.009}$ | **39.4** |

### A.5 Evaluation Prompts and GPT-4 Judging Protocol

For all HHH experiments, we use the following base prompt:

> *"Human asked you a question, please answer as an assistant. {Question}"*

For toxicity-reduction experiments, we use:

> *"{Prompt}. Given the prompt, continue the sentences."*

For the pairwise evaluation, GPT-4 receives the two candidate responses, the target attribute, and the desired attribute. It is instructed to briefly compare the two continuations and return one of four outcomes: Response 1, Response 2, "None" when both are equally good, or "Neither" when both are poor. We report the resulting outcomes as an LLM-as-a-judge measurement rather than as human evaluation.

The judging template follows the form:

> *"The user did an experiment where several language models were prompted to continue the conversation with an input message written by a person. The responses of the language models are:*
> *Person 1: {Response 1}*
> *Person 2: {Response 2}*
> *This input message often contains {Attribute} language and the goal is to continue the conversation with a message that is {Desired Attribute}. Briefly reason about which continuation is better and end your answer with 'Preferred continuation: 1' or 'Preferred continuation: 2', depending on which option is better. If both options are equally good, end your response with 'Preferred continuation: None'. If both options are equally bad, end your response with 'Preferred continuation: Neither'."*

### A.6 COLLATERAL GENERATION-QUALITY CHECK

We additionally evaluate whether steering produces repetitive, low-diversity, or unreadable generations on TruthfulQA. Table 10 reports trigram repetition, diversity, and the fraction of unreadable generations.

Table 10: Trigram repetition (Rep), diversity (Div), and unreadable ratio (Unread) on TruthfulQA. JAM maintains comparable diversity and readability to the baseline.

| | Baseline | | | JAM | | |
|---|---|---|---|---|---|---|
| Model | Rep↓ | Div↑ | Unread↓ | Rep↓ | Div↑ | Unread↓ |
| Llama3 | .00 | .98 | .00 | .00 | .97 | .00 |
| Mistral | .00 | 1.00 | .00 | .00 | 1.00 | .00 |

The measured outputs preserve the baseline's zero trigram-repetition and unreadable-output rates on this check, while diversity changes only slightly for Llama3 and remains unchanged for Mistral. This diagnostic is complementary to the task-specific semantic evaluations in the main text.