Manifold-Guided Attention Steering

Anonymous Authors[1]

Abstract

Large language models frequently produce errors in reasoning tasks despite possessing the underlying knowledge required for correct reasoning. One possible approach to improve reasoning consistency is through activation steering. However, existing activation steering approaches apply fixed, pre-computed correction vectors, ignoring where the model currently sits along its generation trajectory; the result is indiscriminate perturbation that disrupts already-correct steps as freely as erroneous ones. We propose **Ma**nifold-**G**uided Attention **S**teering (MAGS), a trajectory-aware inference-time intervention grounded in a geometric observation: the output activations of specific attention heads diverge from a low-dimensional *correctness manifold* at the point of error, and this deviation compounds through subsequent steps. For each identified attention head, we learn a low-dimensional subspace from contrastive pairs of correct and incorrect traces that capture the directions along which error behavior deviates from correct behavior. During inference, we monitor each head's proximity to this manifold and apply a targeted projection correction when deviation exceeds a learned threshold, steering the attention output back toward the correct subspace before the error propagates. MAGS consistently outperforms both unsteered baselines and static steering approaches across benchmarks spanning mathematical reasoning (MATH-500, GSM8K), code generation (HumanEval, MBPP), and molecular generation (SMILES), suggesting that correctness manifolds are a general feature of LLM attention geometry.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

1. Introduction

Large language models (LLMs) frequently produce reasoning errors in multi-step reasoning tasks despite possessing the underlying capability to solve them. Under repeated sampling, the same model and prompt that produce an incorrect solution will often produce a correct one (Chen et al., 2021; Wang et al., 2023; Cobbe et al., 2021), suggesting that the capability is present, but its reliable expression is not. Process-level annotations further confirm that most errors arise at intermediate reasoning steps rather than from a terminal absence of knowledge (Lightman et al., 2023; Uesato et al., 2022). Since errors arise in the generation process rather than from missing knowledge, correcting them at inference time is a natural and practical target.

Existing activation steering methods (Turner et al., 2023; Zou et al., 2023; Rimsky et al., 2024; Vu and Nguyen, 2025) apply a fixed correction vector to the residual stream at generation steps. These approaches are well-suited for persistent, global behaviors (tone, style, sentiment) but are structurally mismatched to reasoning. A reasoning trajectory may proceed correctly for many steps before committing a localized error at step t^*; applying constant corrections may corrupt the correct intermediate steps while offering no guarantee of intercepting the error.

We hypothesize that reasoning errors manifest as a drift in a low-dimensional subspace of individual attention heads' output space: correct and incorrect trajectories occupy geometrically separable regions, and the transition from correct to incorrect behavior follows a structured, low-rank direction. We perform diagnostic experiments to confirm this hypothesis and in fact discover that correct and incorrect trajectories are highly separable by a low-dimensional subspace of attention-head activations. This is consistent with mechanistic interpretability findings that individual attention heads are functionally specialized (Elhage et al., 2021; Wang et al., 2022), and with the linear representation hypothesis (Park et al., 2023), which posits that semantically meaningful distinctions are encoded along low-dimensional linear directions.

Therefore, one should steer only when their attention outputs have drifted into the error subspace.

We propose **Manifold-Guided Attention Steering**

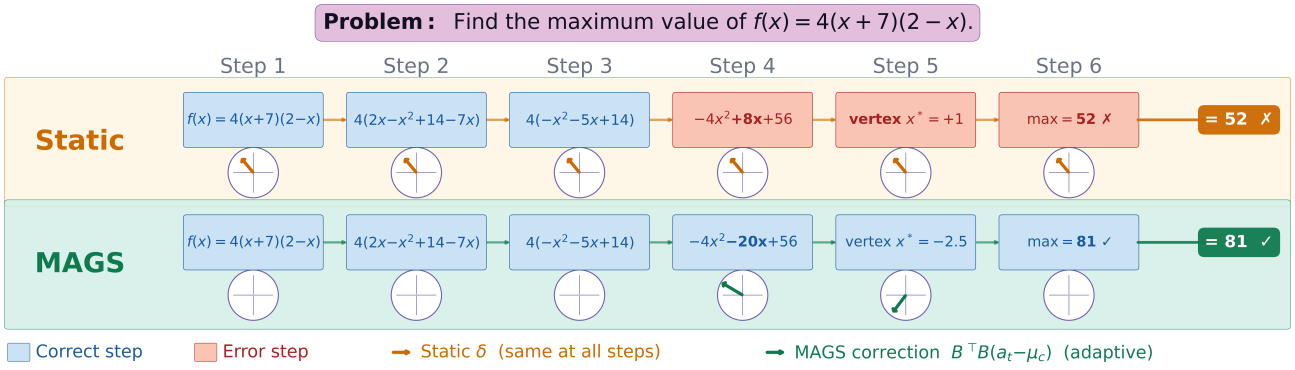

Figure 1. **Comparison of static and Manifold-Guided Attention Steering (MAGS) on an example problem.** Step-by-step reasoning traces for a static baseline and MAGS. Blue boxes denote correct reasoning steps; red boxes denote erroneous ones.

(MAGS): an adaptive intervention that dynamically steers the attention head outputs when reasoning errors are detected. MAGS outperforms static steering baselines across reasoning benchmarks and molecular generation on three model families, including Llama, Gemma, and GPT-OSS.

In summary, our contributions are as follows:

1. We hypothesize that reasoning errors manifest as structured drift in a low-dimensional subspace of individual attention heads' output space, and confirm this hypothesis with diagnostic experiments showing that correct and incorrect trajectories are highly separable (Section 3).

2. We propose Manifold-Guided Attention Steering (MAGS), an adaptive inference-time mechanism that monitors attention heads for reasoning drift and applies dynamic correction only when needed (Section 4).

3. Empirically, MAGS consistently outperforms static steering baselines on benchmarks across three model families, by up to 10.8% while incurring negligible inference overhead (Section 5).

2. Related Work

We discuss existing inference-time steering methods and geometric interpretability work. Existing steering methods apply fixed corrections without error-detection mechanisms; existing interpretability work establishes the geometric structure that we exploit and extend for adaptive intervention. To our knowledge, MAGS is the first method to combine per-step detection with geometry-aware, conditional correction at the attention-head level.

2.1. Activation Steering and Inference-time Intervention

Activation steering methods modify internal representations at inference time without updating model parameters. *Activation Addition* (Turner et al., 2023) adds a fixed difference vector to the residual stream throughout generation. *Contrastive Activation Addition* (CAA; Rimsky et al. 2024) improves reliability by averaging difference vectors across many contrastive prompt pairs. *Representation Engineering* (RepE; Zou et al. 2023) extracts principal steering directions from contrastive activations via PCA. *Angular Steering* (Vu and Nguyen, 2025) replaces additive correction with a 2D rotation applied uniformly across all layers. *Inference-Time Intervention* (ITI; Li et al. 2023a) shifts attention head outputs along a probing direction to improve truthfulness. *CREST* (Jiang et al., 2024) identifies reasoning-relevant attention heads and applies fixed steering vectors to them, but does not adapt the correction to the model's current trajectory state.

All of these methods share a common limitation: they steer along a fixed direction regardless of the current activation state. MAGS addresses this by introducing a dynamic proximity trigger that fires only when a head drifts toward the error subspace and applying a step-dependent correction whose direction is determined by the current activation's projection onto the error subspace rather than a fixed vector (as illustrated in Figure 1).

2.2. Geometric Structure of Transformer Representations

A growing body of work establishes that transformer representations have rich geometric structure that can be leveraged for analysis and intervention.

Mechanistic interpretability studies decompose transformer computation into interpretable circuits. Elhage et al. (2021) show that attention heads implement primitive operations (copying, retrieval, inhibition) whose outputs compose addi-

tively in the residual stream. Wang et al. (2022) shows that multi-step tasks are implemented by sparse circuits across a small number of heads. Together, these results suggest that reasoning failures are likely attributable to specific heads in failure modes, which motivates our head-level intervention.

The *linear representation hypothesis* (Park et al., 2023) posits that semantically meaningful distinctions are encoded along low-dimensional linear directions in transformer representations. Burns et al. (2022) show that truth has a linear representation findable by contrastive probing, establishing a precedent for our contrastive PCA construction. Zou et al. (2023) confirm that high-level concepts, including reasoning quality, are linearly decodable from residual stream activations. MAGS extends this line of work to the per-head level, showing that the correct-to-error transition in reasoning trajectories is also linearly structured within individual head-output spaces.

3. Detecting Error Drift in Attention Heads

We hypothesize that incorrect reasoning traces induce drift in the output of a subset of attention heads toward a low-dimensional error subspace, geometrically separable from the subspace occupied by correct traces. We empirically validate this hypothesis by constructing a contrastive error manifold per head and demonstrating that a proximity-based score achieves strong trajectory-level error detection across layers and heads.

3.1. Setup and Notation

We consider a set of reasoning problems $\mathcal{P} = \{p_1, \ldots, p_N\}$. For each problem p_i, we generate S independent reasoning traces $\mathcal{T}_i = \{\tau_{i,1}, \ldots, \tau_{i,S}\}$. Each trace τ is a token sequence of length L_τ, and is assigned a binary label $y_\tau \in \{0, 1\}$ (1 = correct final answer). Assume $|\mathcal{T}_i^+| \geq 1$ and $|\mathcal{T}_i^-| \geq 1$ for all p_i. We write $\mathcal{T}_i^+ = \{\tau \in \mathcal{T}_i : y_\tau = 1\}$ and $\mathcal{T}_i^- = \{\tau \in \mathcal{T}_i : y_\tau = 0\}$ for the correct-trace and error-trace sets of problem p_i.

Given a transformer-based language model with L layers and H attention heads per layer, where each head operates on a d_h-dimensional output space. For a sample τ and head (l, h), the sequence of attention head outputs is:

$$\mathbf{A}_\tau^{(l,h)} = \left[\boldsymbol{a}_1^{(l,h,\tau)}, \boldsymbol{a}_2^{(l,h,\tau)}, \ldots, \boldsymbol{a}_{L_\tau}^{(l,h,\tau)}\right] \in \mathbb{R}^{L_\tau \times d_h}. \tag{1}$$

3.2. Contrastive Error Manifold Construction

Given correct and incorrect reasoning traces, we construct a per-head error subspace by identifying the low-dimensional directions along which correct and incorrect activations diverge.

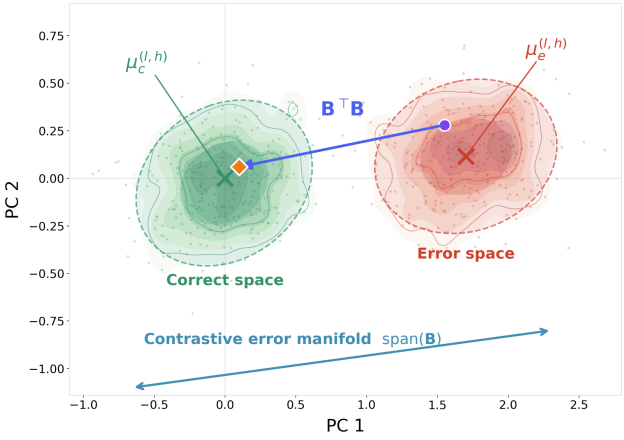

Figure 2. Schematic of the contrastive error manifold. Correct and error activation spaces are separated along the learned subspace $\mathrm{span}(\mathbf{B})$. Given an error activation a_t, the projection $\mathbf{B}^\top\mathbf{B}$ gives the direction to map back toward $\mu_c^{(l,h)}$.

Per-problem difference vectors. For each problem and each head (l, h), define the per-class mean:

$$\boldsymbol{\mu}_{c,i}^{(l,h)} = \frac{1}{\sum_{\tau \in \mathcal{T}_i^+} L_\tau} \sum_{\tau \in \mathcal{T}_i^+} \sum_{t=1}^{L_\tau} \boldsymbol{a}_t^{(l,h,\tau)}, \tag{2}$$

$$\boldsymbol{\mu}_{e,i}^{(l,h)} = \frac{1}{\sum_{\tau \in \mathcal{T}_i^-} L_\tau} \sum_{\tau \in \mathcal{T}_i^-} \sum_{t=1}^{L_\tau} \boldsymbol{a}_t^{(l,h,\tau)}. \tag{3}$$

The *contrastive difference vector* for problem p_i and head (l, h) is:

$$\boldsymbol{\delta}_i^{(l,h)} = \boldsymbol{\mu}_{e,i}^{(l,h)} - \boldsymbol{\mu}_{c,i}^{(l,h)} \in \mathbb{R}^{d_h}. \tag{4}$$

By construction, $\boldsymbol{\delta}_i^{(l,h)}$ cancels all directions uniformly activated by problem p_i regardless of correctness, isolating the *directional shift* attributable to the error.

Difference matrix. For a set of N problems, we stack the difference vectors row-wise:

$$(\mathbf{D}^{(l,h)})^T = \left[\boldsymbol{\delta}_1^{(l,h)} \quad \ldots \quad \boldsymbol{\delta}_P^{(l,h)}\right] \in \mathbb{R}^{d_h \times N}. \tag{5}$$

Error subspace via PCA. Compute the compact singular value decomposition $\boldsymbol{D}^{(l,h)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Define the *error subspace basis* as the top-k right singular vectors:

$$\boldsymbol{B}^{(l,h)} = \boldsymbol{V}_{:,1:k}^\top \in \mathbb{R}^{k \times d_h}, \tag{6}$$

where the rows of $\boldsymbol{B}^{(l,h)}$ are orthonormal. The error subspace captures the k directions in head-output space along which correct-to-incorrect deviation has the greatest variance across problems, as illustrated in Figure 2.

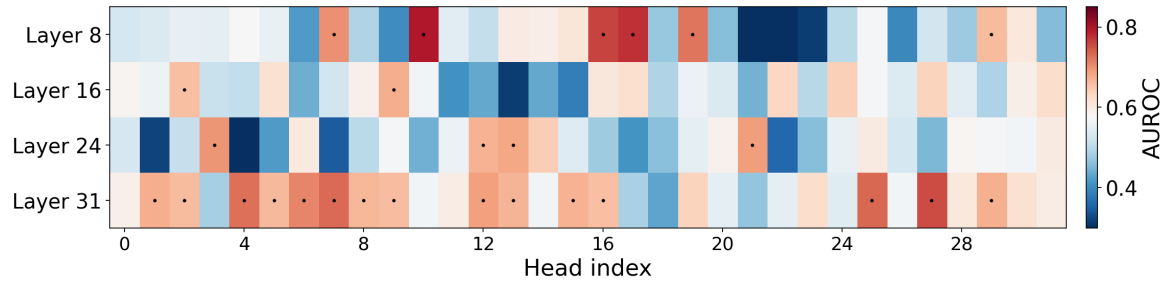

Figure 3. Per-head error detection AUROC across four monitored layers for the Math-Instruct dataset with Llama-3.1-8b-Instruct. Dots mark heads with AUROC > 0.65. The signal is sparse and concentrated in specific heads.

Global correct-state centroid. Compute a global reference point from all correct traces:

$$\boldsymbol{\mu}_c^{(l,h)} = \frac{\sum_i \sum_{\tau \in \mathcal{T}_i^+} \sum_{t=1}^{L_\tau} \boldsymbol{a}_t^{(l,h,\tau)}}{\sum_i \sum_{\tau \in \mathcal{T}_i^+} L_\tau}. \tag{7}$$

This serves as the centering reference at inference time, since per-problem means are not available during generation.

Remark 1. *The choice to fit PCA on* difference vectors *rather than on the raw error-state vectors is critical. If PCA were fit on $\mathcal{E}^{(l,h)}$ directly (pooled baseline), the dominant directions would reflect problem-level features (e.g., problem difficulty or topic) rather than the intrinsic error direction. The contrastive construction ensures that $\boldsymbol{D}^{(l,h)}$ has zero mean in any direction that is uniformly correlated with problem content, leaving only the error-specific signal.*

3.3. Proximity-Based Error Detection

To detect when a head has drifted into the error subspace, we measure how much its current output projects onto the learned error subspace. A large projection indicates the head is behaving similarly to how it behaves in erroneous traces. At each decode step t during inference, for each monitored head (l, h), we compute the *proximity score*:

$$\begin{aligned} d_t^{(l,h)} &= \left\| \boldsymbol{B}^{(l,h)} \left(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)} \right) \right\|^2 \\ &= \left(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)} \right)^\top \boldsymbol{B}^{(l,h)\top} \boldsymbol{B}^{(l,h)} \left(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)} \right). \end{aligned} \tag{8}$$

This is the squared norm of the projection of the centered head output onto the error subspace. A large value indicates that the current head output has a substantial component along the learned error directions.

To avoid flagging normal generation steps, we calibrate a per-head threshold on correct traces and trigger only when the proximity score exceeds it. We fire a correction only when the proximity score exceeds a per-head threshold calibrated on correct traces, ensuring that already-correct steps are left undisturbed. A *trigger* fires at step t for head (l, h) when:

$$d_t^{(l,h)} > \tau^{(l,h)}, \tag{9}$$

where $\tau^{(l,h)}$ is set to the q-th percentile of $\{d_t^{(l,h)}\}$ computed over all token steps from correct trajectories in the training set.

3.4. Empirical Validation of the Drift Hypothesis

We validate whether the learned error subspace carries a detectable signal for distinguishing correct from incorrect reasoning trajectories. Using Math-Instruct traces from Llama-3.1-8B-Instruct collected at layers $\{8, 16, 24, 31\}$, we perform a problem-level 70/30 train/test split, ensuring that all traces of a given problem land in the same split and the manifold never observes test problems during construction. We build the contrastive error manifold on the training split and evaluate on the held-out test split.

For each trace, we aggregate the per-step proximity scores $\{d_t^{(l,h)}\}$ into a single scalar using max aggregation, and classify the trace as incorrect if the score exceeds a threshold $\tau^{(l,h)}$ calibrated on the training set by maximizing balanced accuracy. We report trajectory-level AUROC for each (l, h) pair independently. Figure 3 shows the per-head AUROC across all monitored layers. The signal is sparse and concentrated in specific heads (marked dots for AUROC > 0.65), confirming that error drift is a structured, localized phenomenon rather than a diffuse property of all heads.

4. Manifold-Guided Attention Steering

Having established that proximity scores reliably detect drift toward the error subspace, we now describe how MAGS exploits this signal to apply a targeted correction to the attention head output at inference time.

Head selection. Rather than monitoring all $L \times H$ heads, we pre-select the top-K heads by *held-out AUROC*: for each head, AUROC is computed between the trajectory-level error label and the mean proximity score over the trajectory, on a held-out problem split. Monitoring only the top-K heads reduces per-step overhead from $O(L \cdot H \cdot k \cdot d_h)$ to $O(K \cdot k \cdot d_h)$.

Steering by error-component correction. When head (l, h) triggers at step t, we apply the following in-place correction to its output *before* it is passed to the output projection of layer l, $W_O^{(l)}$:

$$\tilde{\boldsymbol{a}}_t^{(l,h)} \;=\; \boldsymbol{a}_t^{(l,h)} - \alpha\, \boldsymbol{B}^{(l,h)\top} \boldsymbol{B}^{(l,h)} \big(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}\big), \quad (10)$$

where $\alpha \in (0, 1]$ is a steering strength hyperparameter that we can control empirically.

Suppose $\alpha = 1$, and let $\boldsymbol{P}_\perp^{(l,h)} = \boldsymbol{I}_{d_h} - \boldsymbol{B}^{(l,h)\top} \boldsymbol{B}^{(l,h)}$ denote the orthogonal projector onto the *complement* of the error subspace. Then (10) can be written equivalently as:

$$\tilde{\boldsymbol{a}}_t^{(l,h)} \;=\; \boldsymbol{\mu}_c^{(l,h)} \;+\; \boldsymbol{P}_\perp^{(l,h)} \big(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}\big). \quad (11)$$

This form makes the semantics transparent: we decompose the deviation of the head output from the correct-state mean into an error-subspace component and a complement component, then discard only the former. Algorithm 1 summarizes the complete inference procedure, combining the proximity check (Section 3.3) and the error-component correction into a single decode loop.

We formalize the key advantage over full residual stream correction: the $d_h - k$ directions unrelated to the error manifold are completely untouched.

Proposition 1 (Information Preservation)**.** *The correction* (10) *preserves all information in* $\boldsymbol{a}_t^{(l,h)}$ *that lies in the* $(d_h - k)$*-dimensional complement of the error subspace. Specifically, for any vector* $\boldsymbol{v} \in \mathbb{R}^{d_h}$ *with* $\boldsymbol{B}^{(l,h)}\boldsymbol{v} = \boldsymbol{0}$*:*

$$\left\langle \tilde{\boldsymbol{a}}_t^{(l,h)},\, \boldsymbol{v} \right\rangle \;=\; \left\langle \boldsymbol{a}_t^{(l,h)},\, \boldsymbol{v} \right\rangle. \quad (12)$$

The overhead of MAGS per decode step is dominated by the K proximity score computations. Each requires a matrix-vector product $\boldsymbol{B}^{(l,h)}\boldsymbol{v}$ of cost $O(k \cdot d_h)$, followed by a norm computation of cost $O(k)$. The conditional correction, when triggered, requires one additional matrix-vector product $\boldsymbol{B}^{(l,h)\top}(\boldsymbol{B}^{(l,h)}\boldsymbol{v})$ of cost $O(k \cdot d_h)$. The total per-step overhead is $O(K \cdot k \cdot d_h)$.

Compositional Steering for Multiple Objectives. A fundamental challenge in multi-objective steering is that naively combining steering vectors for different constraints can produce conflicting corrections. We sidestep this problem by introducing MAGS$^{\text{u}}$, which treats each objective independently from the start. For each objective c_k, MAGS$^{\text{u}}$ extracts a dedicated error manifold, which naturally yields disjoint head sets $\mathcal{H}_1, \ldots, \mathcal{H}_K$ during head selection. Then, at inference time, MAGS$^{\text{u}}$ steers the *union* of all selected heads, applying each head's correction through its own independently learned manifold.

Algorithm 1 Manifold-Guided Head Steering (MAGS) — Inference

Require: Pre-computed manifolds $\{\boldsymbol{B}^{(l,h)}, \boldsymbol{\mu}_c^{(l,h)}, \tau^{(l,h)}\}$ for top-K heads; prompt $x_{1:\text{prompt}}$
$t \leftarrow 0$
while generation not complete **do**
  $t \leftarrow t + 1$
  Run forward pass for token t
  Collect $\boldsymbol{a}_t^{(l,h)}$ for all monitored (l, h)
  for each monitored head (l, h) in layer order **do**
    Compute $d_t^{(l,h)} \leftarrow \left\| \boldsymbol{B}^{(l,h)}(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}) \right\|^2$
    if $d_t^{(l,h)} > \tau^{(l,h)}$ **then**
      $v \leftarrow \boldsymbol{B}^{(l,h)\top}\boldsymbol{B}^{(l,h)}(\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)})$
      $\tilde{\boldsymbol{a}}_t^{(l,h)} \leftarrow \boldsymbol{a}_t^{(l,h)} - \alpha v$
      Replace $\boldsymbol{a}_t^{(l,h)}$ with $\tilde{\boldsymbol{a}}_t^{(l,h)}$
    end if
  end for
  Complete the forward pass; sample next token x_{t+1}
end while=0

5. Experiments

5.1. Reasoning Benchmarks

Setup and metrics. We evaluate across two reasoning domains chosen to span difficulty and output structure: mathematical reasoning on **MATH-500** (Lightman et al., 2023) and **GSM8K** (Cobbe et al., 2021), and code generation on **HumanEval** (Chen et al., 2021) and **MBPP** (Austin et al., 2021), where correctness is verified by executing test cases rather than by string matching. To assess generality across model families, we run all reasoning and code experiments on two instruction-tuned language models: **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024) and **Gemma-4-E4b-it** (Gemma Team, Google DeepMind, 2025).

Baselines. We compare against two families of inference-time steering methods.

Representation-level methods. **Inference-Time Intervention** (ITI;Li et al. 2023a), which steers individual attention head outputs along a fixed linear probe direction; **Angular Steering** (AS; Vu and Nguyen 2025), which applies a fixed 2D rotation in the mean-difference span across all layers.

Table 1. Performance and output fluency (perplexity, ↓) of steering methods on Llama-3.1-8B-Instruct. Best accuracy per benchmark in **bold**; best perplexity in *italics*.

	MATH-500		GSM8k		HumanEval		MBPP	
Method	Acc.↑	PPL↓	Acc.↑	PPL↓	Acc.↑	PPL↓	Acc.↑	PPL↓
Unsteered	0.478	1.229	0.860	1.196	0.561	*1.121*	0.562	2.978
ITI	0.498	*1.227*	0.855	1.335	0.573	1.136	0.548	2.250
AS	0.506	1.232	0.857	1.335	0.591	1.131	0.546	*2.214*
CD	0.492	1.295	0.858	1.271	0.591	1.152	0.555	2.291
MAGS	**0.530**	*1.227*	**0.867**	*1.194*	**0.604**	1.130	**0.574**	2.970

Table 2. Performance and output fluency (perplexity, ↓) of steering methods on Gemma-4-E4b-it. Best accuracy per benchmark in **bold**; best perplexity in *italics*

	MATH-500		GSM8k		HumanEval		MBPP	
Method	Acc.↑	PPL↓	Acc.↑	PPL↓	Acc.↑	PPL↓	Acc.↑	PPL↓
Unsteered	0.614	*1.107*	0.899	1.710	0.560	*1.231*	0.587	5.169
ITI	0.646	1.121	**0.913**	1.293	0.463	1.267	0.557	5.053
AS	0.566	1.279	0.900	1.416	0.518	1.334	0.548	5.475
CD	0.604	1.130	0.874	*1.107*	0.433	1.272	0.593	*4.748*
MAGS	**0.648**	1.108	**0.913**	1.210	**0.604**	1.281	**0.604**	5.059

Since ITI and AS were originally designed for behavioral alignment rather than reasoning, we adapt them to our setting by treating correct and incorrect solution traces as the desired and undesired contrast sets, respectively, replacing their original prompt-pair construction.

Decoding-level methods. **Contrastive Decoding** (CD; Li et al. 2023b) contrasts the token distributions of a large expert model and a smaller amateur model at each decoding step, providing a baseline that operates at the output distribution level and a more computationally expensive method than all steering methods.

Manifold construction. For each benchmark, we collect contrastive trace pairs (i.e., same problem, one correct solution and one incorrect) from the corresponding training split: **Math-Instruct** (Yue et al., 2023) for MATH-500, the **GSM8K** (Cobbe et al., 2021) training set for GSM8K, and **APPS** (Hendrycks et al., 2021) for both HumanEval (Chen et al., 2021) and MBPP(Austin et al., 2021). Traces are generated by sampling the base model; problems for which both a correct and an incorrect trace cannot be obtained within 8 samples are discarded.

Result. As shown in Table 1 and Table 2, MAGS consistently outperforms the unsteered baseline and all three steering methods across both models and all four benchmarks. On MATH-500, where multi-step derivations provide the most opportunities for error compounding, MAGS achieves the largest gains: +5.2 points over the unsteered Llama baseline and +3.4 points over the unsteered Gemma baseline. On GSM8K, where problems are shorter and errors more localized, the margin over baselines narrows, consistent with the intuition that proximity-triggered correction is most valuable when correct steps substantially outnumber erroneous ones.

Code generation reveals a sharper contrast. MAGS improves HumanEval pass@1 by approximately 4 points over the unsteered baseline on both models. All three baselines *degrade* HumanEval performance relative to the unsteered Gemma model, suggesting that unconditional interventions disrupt syntactic coherence even when the model's reasoning is already correct. MAGS avoids this failure mode since its proximity trigger suppresses corrections at steps where no drift is detected.

Beyond accuracy, we evaluate output fluency via perplexity to assess whether steering distorts the model's generation distribution. MAGS consistently matches or approaches unsteered perplexity across benchmarks, confirming that the proximity threshold suppresses corrections on already-

correct steps and leaves the output distribution largely intact. ITI and Angular Steering generally raise perplexity relative to the baseline, indicating that static interventions can potentially distort the generation distribution.

5.2. Molecular Generation

Task. Molecular generation requires producing syntactically and chemically valid SMILES strings (Weininger, 1988). Unlike natural language, SMILES has rigid grammar rules: a single misplaced token renders the entire molecule invalid. Beyond validity, we additionally steer toward improved binding affinity against a target protein, measured via the AutoDock-GPU docking score (Santos-Martins et al., 2021). Since an ideal molecule would be both syntactically valid and chemically strong binding, this task induces a natural multi-objective structure.

Setup and Metrics. For molecular generation, we evaluate using **GPT-OSS 20B** (OpenAI, 2025). We report **Validity** (fraction of generated SMILES parseable; higher is better) and **Binding Affinity** (measured as AutoDock-GPU scores; lower is better). We generate 500 molecules per method. Contrastive Decoding is excluded as it requires a smaller companion model; no publicly available version of GPT-OSS below 20B parameters exists at the time of writing.

Table 3. Molecular generation by steering GPT-OSS-20B. Validity: higher is better (%). Binding Affinity: lower score is better(kcal/mol).

Method	Validity (↑)	Binding Affinity (↓)
Unsteered	50.4	-7.36
Angular Steering	51.4	-7.44
ITI	**57.8**	-7.20
MAGS	54.8	-7.54
MAGS^{u}	54.8	**-7.56**

Result. As shown in Table 3, ITI illustrates the consequence of the entanglement of objectives: it substantially improves validity but incurs a significant drop in binding affinity, a sign of overcorrection toward common valid scaffolds that are grammatically safe but chemically generic. MAGS^{u} avoids this interference by learning two independent manifolds: the affinity manifold (contrasting high and low binding affinity) and the validity manifold (contrasting valid and invalid molecules). At inference time, the union of both subspaces applies each correction in its own independent direction, allowing binding affinity to improve without the validity–affinity interference observed in joint methods.

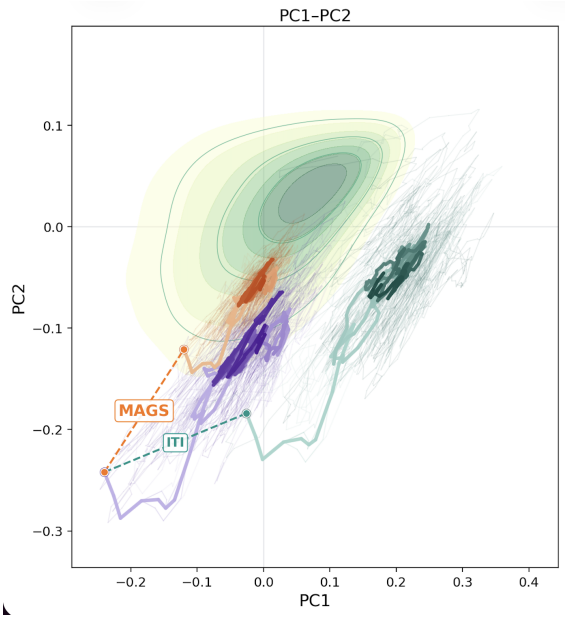

Figure 4. Latent-space trajectories of attention-head activations projected onto the top-4 principal components of the contrastive error subspace. **Filled contours**: kernel-density estimate of the correct-output activation distribution. **Purple**: unsteered mean trajectory (light → dark = early → late generation steps). **Orange**: MAGS-steered mean trajectory. **Teal**: ITI-steered mean trajectory. Faint lines show individual problem traces. Dashed connectors mark a correction step for MAGS and ITI.

6. Discussion

Ablation on hyperparameter sensitivity. While we detail the experiment hyperparameters in Appendix B, Table 4 reveals a clear contrast in robustness across methods. Steering Llama-3.1-8B on MATH-500, MAGS accuracy varies narrowly (0.492–0.530) across all configurations tested, indicating that the proximity threshold already governs when corrections are applied and reduces sensitivity to other hyperparameters. ITI spans a much wider range (0.214–0.498), with performance collapsing at high steering strength, confirming that static interventions are brittle to miscalibration. Angular Steering shows the greatest variance (0.206–0.506), with accuracy swinging dramatically across rotation angles.

Latent trajectory analysis. To visualize how MAGS affects the model's internal representations during generation, we project the attention-head output activations at a steered layer onto the top-4 principal components of the contrastive error subspace. We compare MAGS against ITI as ITI also operates directly on attention-head representations. We run parallel steered and unsteered decodes on the same set of problems, collect per-step activations, and plot the resulting trajectories in the PC1-PC2 plane.

Figure 4 reveals a clear divergence between methods. The unsteered trajectory (purple) drifts steadily away from the

correct-output distribution and fails to recover, consistent with the hypothesis that reasoning errors manifest as directional activation drift. MAGS (orange) intervenes at the correction step (orange dashed connector) and immediately redirects the trajectory back toward the high-density correct region, where it remains for the rest of the generation. ITI (teal), even after intervention, continues to diverge and settles in a region far from the correct-output distribution, which geometrically explains why static interventions can degrade performance: without a subspace constraint, the correction vector pushes activations in an imprecise direction that does not align with the correct-output manifold.

Visualization of the effect of steering on attention graph. To examine how MAGS reshapes information routing, we visualize the *relative attention shift* at a layer after the correction layer, denoted ℓ_{bip}. Since MAGS modifies the head *output* $\mathbf{a}_h = A_h V_h$ rather than the routing matrix A_h directly, the corrected residual is written into the key-value cache at position t_{fire}. Then, subsequent query steps can attend to this corrected entry. To isolate the effect of accumulated KV-cache differences, both steered and unsteered streams are run on the same forced token sequence (the steered model's greedy outputs), so the two caches diverge solely through the corrections. For each problem we average attention weights over all heads and compute

$$\frac{\Delta W}{W} = \frac{W_{\text{steered}} - W_{\text{unsteered}}}{W_{\text{unsteered}} + \varepsilon}.$$

Figure 5 shows an example problems from the MATH-500 dataset for Gemma-4-E4b-it; red (blue) indicates positions attended to more (less) by the steered model relative to the baseline.

The strongest signal in both panels is a set of *vertical stripes*: certain key tokens receive consistently higher attention across all subsequent query steps. This reveals that MAGS does not merely fix a single prediction in isolation, but makes the corrected reasoning step a more *retrievable* context anchor: the entire generation following the correction keeps routing information through those tokens, reinforcing their influence persistently. The effect is consistent across both problems despite their different mathematical content, suggesting that this "attention anchoring" behavior may be a general mechanism by which MAGS steers the model back onto a correct reasoning trajectory.

7. Conclusion

We presented Manifold-Guided Attention Steering (MAGS), an inference-time method that corrects attention-head activations by projecting them back onto a low-rank manifold learned from correct-output traces, intervening only when a proximity threshold τ signals that the current activation has drifted clearly off-manifold and leaving already-correct

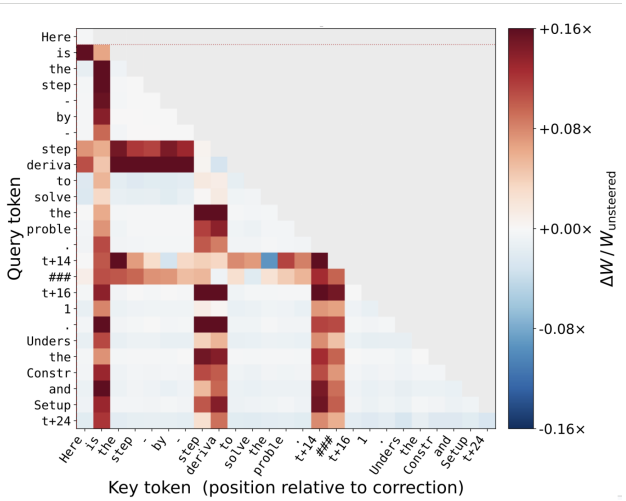

Figure 5. Relative attention shift $\Delta W / W_{\text{unsteered}}$ at layer ℓ_{bip} for an example problems from MATH-500, steered with Gemma-4-E4b-it. Each cell shows the head-averaged attention change between the steered and unsteered model on the same forced token sequence. The dashed horizontal line marks the first correction step. t_{fire}

steps undisturbed. Across mathematical reasoning, code generation, and molecular generation on three model families, MAGS consistently outperforms ITI, Angular Steering, and Contrastive Decoding. The manifold perspective further reframes steering as a *drift detection* problem: the low-rank basis B identifies the subspace most predictive of error onset, providing interpretable geometric structure that scalar-magnitude methods lack, as visualized through latent-space projections of steered trajectories being pulled back into the high-density region of correct-output activations.

Limitations and future work. MAGS requires a set of contrastive-output traces for manifold fitting, which may be difficult to curate in low-resource domains. The current method also operates on a fixed set of target heads identified offline; an adaptive scheme that selects heads dynamically based on generation context could reduce reliance on this pre-selection step. More broadly, we treat each head independently; jointly modeling interactions between heads may yield stronger corrections at lower computational cost. Finally, while our experiments cover three domains, the degree to which the learned manifold transfers across tasks within a domain (e.g., arithmetic to symbolic reasoning) remains an open question. We leave these directions to future work.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mo Bavarian, Clemens Winter, Phil Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv: 2107:03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformercircuits.pub/2021/framework/index.html.

Gemma Team, Google DeepMind. Gemma 4 technical report. https://ai.google.dev/gemma/docs/core/model_card_4, 2025.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with apps. In *The Thirty-fifth Annual Conference on Neural Information Processing Systems*, 2021.

Yutao Jiang et al. Understanding and steering the cognitive behaviors of reasoning models at test-time. *arXiv preprint arXiv:2512.24574*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

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/.

Diogo Santos-Martins, Leonardo Solis-Vasquez, Andreas F. Tillack, Michel F. Sanner, Andreas Koch, and Stefano Forli. Accelerating autodock4 with gpus and gradient-based local search. *Journal of Chemical Theory and Computation*, 17(2):1060–1073, Feb 2021. ISSN 1549-9618. doi: 10.1021/acs.jctc.0c01006. URL https://doi.org/10.1021/acs.jctc.0c01006.

Alex Turner, Lisa Thiergart, David Udell, Jan Leike, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Jonathan Uesato et al. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Hieu M. Vu and Tan Minh Nguyen. Angular steering: Behavior control via rotation in activation space. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005. URL https://doi.org/10.1021/ci00057a005.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

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. Proof of Proposition 1

Proof. Expanding using (10):

$$\left\langle \tilde{\boldsymbol{a}}_t^{(l,h)}, \boldsymbol{v} \right\rangle = \left\langle \boldsymbol{a}_t^{(l,h)}, \boldsymbol{v} \right\rangle - \left\langle \boldsymbol{B}^{(l,h)\top} \boldsymbol{B}^{(l,h)} (\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}), \boldsymbol{v} \right\rangle \tag{13}$$

$$= \left\langle \boldsymbol{a}_t^{(l,h)}, \boldsymbol{v} \right\rangle - \left\langle \boldsymbol{B}^{(l,h)} (\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}), \boldsymbol{B}^{(l,h)} \boldsymbol{v} \right\rangle \tag{14}$$

$$= \left\langle \boldsymbol{a}_t^{(l,h)}, \boldsymbol{v} \right\rangle - \left\langle \boldsymbol{B}^{(l,h)} (\boldsymbol{a}_t^{(l,h)} - \boldsymbol{\mu}_c^{(l,h)}), \boldsymbol{0} \right\rangle = \left\langle \boldsymbol{a}_t^{(l,h)}, \boldsymbol{v} \right\rangle. \qquad \square$$

B. Additional Experimental Details

B.1. Ablation on Hyperparameter Sensitivity

For ITI, we follow the hyperparameter ranges reported in the original paper (Li et al., 2023a), sweeping over the number of steered heads in $\{24, 48, 96\}$ and intervention strength $\alpha \in \{0.5, 1.0, 5.0\}$. For Angular Steering, rotation angles are sampled uniformly across the full $360°$ range as evaluated in the original work (Vu and Nguyen, 2025). For MAGS, we select hyperparameters to minimize unnecessary interference: we restrict the number of steered heads to a small set (top-1 or top-3 by manifold signal strength) and sweep projection strength $\alpha \in \{0.3, 0.5, 0.7, 1.0\}$, reflecting the principle that corrections should be both targeted and conservative. Corrections are applied only to heads where drift is detected and only to the extent required to return activations to the correct-output manifold.

Table 4. Hyperparameter ablations on MATH-500 (Llama-3.1-8B-Instruct). Best result per method in **bold**.

(a) MAGS: top-k heads and steering strength α.

Config	α	Acc.
top-3	0.3	0.502
top-3	0.5	0.498
top-3	0.7	0.492
top-3	1.0	**0.530**
top-1	0.7	0.492
top-1	1.0	**0.530**

(b) ITI: number of steered heads K and strength α.

	Strength α		
K	0.5	1.0	5.0
24	0.476	0.472	0.298
48	0.488	0.472	0.214
96	**0.498**	0.480	0.314

(c) Angular Steering: rotation angle.

Angle	Acc.
0°	0.488
30°	**0.506**
60°	0.466
90°	0.284
120°	0.206
150°	0.334
180°	0.412
210°	0.400
240°	0.344
270°	0.430
300°	0.500
330°	0.468

B.2. Computational infrastructure.

Experiments with Llama-3.1-8B-Instruct and Gemma-4-E4b-it were conducted on NVIDIA RTX 4090 GPUs (24 GB VRAM) with model weights loaded in `float16` precision. GPT-OSS 20B experiments were run on NVIDIA H200 GPUs (141 GB HBM3) due to the larger memory footprint of the model.

C. Statistical Significance

All confidence intervals are 95% percentile bootstrap CIs with B=10000 resamples (seed = 42). Point estimates are original trial accuracies.

Table 5. Bootstrap 95% CIs on Gemma-4-E4b-it across all benchmarks (B=10,000 resamples, percentile method).

	MATH-500 (N=500)		**GSM8k** (N=1319)		**HumanEval** (N=164)		**MBPP** (N=427)	
Method	Acc.	95% CI	Acc.	95% CI	Acc.	95% CI	Acc.	95% CI
Unsteered	0.614	[0.572, 0.656]	0.899	[0.887, 0.911]	0.560	[0.482, 0.634]	0.587	[0.543, 0.635]
ITI	0.646	[0.604, 0.688]	**0.913**	[0.897, 0.927]	0.463	[0.384, 0.537]	0.557	[0.511, 0.604]
AS	0.566	[0.522, 0.610]	0.900	[0.883, 0.915]	0.518	[0.445, 0.592]	0.548	[0.501, 0.595]
CD	0.604	[0.562, 0.646]	0.874	[0.856, 0.892]	0.433	[0.360, 0.506]	0.593	[0.546, 0.639]
MAGS	**0.648**	[0.606, 0.690]	**0.913**	[0.898, 0.928]	**0.604**	[0.524, 0.677]	**0.604**	[0.557, 0.649]

Table 6. Bootstrap 95% CIs on Llama-3.1-8B-Instruct across all benchmarks (B=10,000 resamples, percentile method).

	MATH-500 (N=500)		**GSM8k** (N=1319)		**HumanEval** (N=164)		**MBPP** (N=427)	
Method	Acc.	95% CI	Acc.	95% CI	Acc.	95% CI	Acc.	95% CI
Unsteered	0.478	[0.428, 0.516]	0.860	[0.841, 0.879]	0.561	[0.494, 0.640]	0.562	[0.515, 0.609]
ITI	0.498	[0.456, 0.540]	0.855	[0.836, 0.874]	0.573	[0.494, 0.646]	0.548	[0.501, 0.595]
AS	0.506	[0.462, 0.550]	0.857	[0.839, 0.876]	0.591	[0.512, 0.665]	0.546	[0.499, 0.593]
CD	0.492	[0.456, 0.542]	0.858	[0.839, 0.876]	0.591	[0.518, 0.665]	0.555	[0.508, 0.602]
MAGS	**0.530**	[0.486, 0.574]	**0.867**	[0.848, 0.886]	**0.604**	[0.524, 0.677]	**0.574**	[0.527, 0.621]