To Call or Not to Call: Diagnosing Intrinsic Over-Calling Bias in LLM Agents

Wei Shi[1,2]*, **Ziheng Peng**[2,3]*, **Sihang Li**[5], **Xiting Wang**[3],
Xiang Wang[5], **Mengnan Du**[4], **Na Zou**[2]†

[1]Shanghai Jiao Tong University, [2]Shanghai Artificial Intelligence Laboratory,
[3]Renmin University of China, [4]The Chinese University of Hong Kong Shenzhen,
[5]University of Science and Technology of China,
swei2001@sjtu.edu.cn, mengnandu@cuhk.edu.cn,
{ziheng.peng, xitingwang}@ruc.edu.cn,
{sihang0520, xiangwang1223, zouna891252}@gmail.com

Abstract

LLM agents exhibit a consistent tendency to over-call, invoking tools even in situations where none is needed. On the When2Call benchmark, six models from three families show high call accuracy but much lower no-call accuracy, leaving overall accuracy in the 55%–70% range. We trace this to an Intrinsic Bias Hypothesis (**IBH**): the call/no-call decision mapping carries an activation-independent CALL offset, so the model favors CALL even at activation parity. Using Sparse Autoencoders (SAEs), we recover behavior-aligned feature bases for the CALL/NO_CALL decision, reduce them to a signed activation margin, and estimate the offset directly. Across all six models, the model is decision-neutral only when NO_CALL activation outweighs CALL activation, consistent with IBH. We then causally test IBH with Adaptive Margin-Calibrated Steering (**AMCS**), a closed-form counter-bias shift along SAE decoder directions. Cancelling the diagnosed offset mitigates over-calling and improves overall accuracy with a negligible drop in call accuracy. Our work recasts over-calling from an empirical phenomenon into a mechanistic object amenable to causal correction. The code is available at https://github.com/SKURA502/agent-sae.

1 Introduction

Large language models (LLMs) have rapidly evolved from text generators into the reasoning backbone of autonomous agents [1–4]. At the core of this transition lies tool use, the ability to interface with external systems such as search engines, code interpreters, APIs, and databases, extending LLMs beyond pure language modeling [5–8].

Beyond executing calls correctly, effective tool use hinges on knowing when to invoke a tool and, crucially, when not to. As shown in Figure 1(a), across six models from three families (Qwen3.5 [4], Gemma-3 [9], Ministral-3 [10]) evaluated on the When2Call benchmark [11], call accuracy remains high, while no-call accuracy is consistently much lower, leaving overall accuracy in the 55%–70% range. Models therefore know how to call tools when calls are required, but often issue calls when none is warranted. Figure 1(a) illustrates this failure mode: given an underspecified Spotify request, the base model invokes the tool instead of asking for the missing song name and device ID. This bias degrades user experience and inflates API costs in deployed systems.

*Equal Contribution
†Corresponding

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

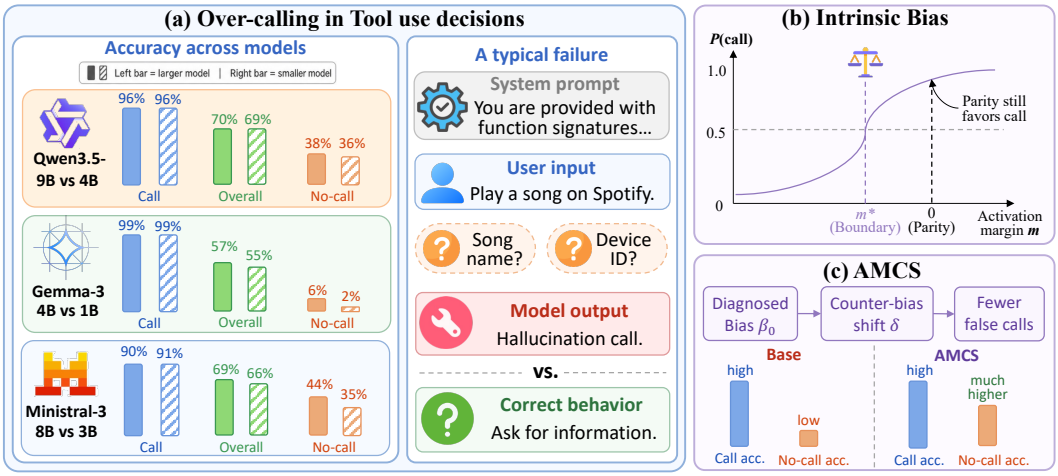

Figure 1: **Overview of over-calling, intrinsic bias, and AMCS. (a)** Across six target models, call accuracy is high, but no-call accuracy is much lower, reducing overall accuracy; a representative case shows the model calling a tool despite missing required information. **(b)** Intrinsic Bias Hypothesis: at activation parity ($m = 0$), the decision still favors CALL; the neutral boundary shifts to $m^\star < 0$. **(c)** AMCS converts the diagnosed bias β_0 into a counter-bias shift δ, aiming to reduce false calls while preserving valid calls.

Having documented the bias, we ask what governs the call/no-call decision. An activation-only account offers a natural mechanism: the decision is determined by the relative activation of call- and no-call-related directions, with the stronger side prevailing. This account makes a testable prediction. If the decision depends only on the activation difference, balancing call and no-call activation should leave the model decision-neutral. Our analysis shows the opposite. Even at activation parity, the model remains biased toward CALL; it becomes neutral only when no-call activation is stronger. A residual bias at parity cannot come from activation levels themselves and must instead enter the decision through a separate, additive term. We formalize this as the Intrinsic Bias Hypothesis (**IBH**): over-calling reflects an activation-independent call offset in the decision mapping.

We test IBH through a pipeline built on Sparse Autoencoders (SAEs) [12, 13], which decompose residual-stream activations into sparse and interpretable features [12, 14] and let us isolate the components driving the CALL/NO_CALL decision. We first recover behavior-aligned CALL and NO_CALL feature bases and verify that a handful of them predict the decision near the residual-stream upper bound (§3). We then reduce these bases to a signed activation margin and test whether activation parity removes the decision asymmetry (§4). It does not: activation geometry and logistic estimates of the offset both show a shifted neutral boundary (Figure 1(b)), and the same SAE directions also separate true from false calls, exposing a lever for intervention. Finally, we causally test IBH with Adaptive Margin-Calibrated Steering (**AMCS**), a closed-form counter-bias shift along SAE decoder directions (§5; Figure 1(c)). Cancelling the diagnosed offset raises no-call accuracy by 4–17 points on five of six models and overall accuracy by up to 5 points, with minimal impact on call accuracy across the same models.

Our main contributions are:

- We open a mechanistic view of tool-use gating by recovering behavior-aligned SAE feature bases that predict the CALL/NO_CALL decision near the residual-stream upper bound (§3).
- We formulate and validate **IBH**, showing that over-calling reflects an activation-independent call offset rather than activation levels alone, and that the same SAE directions further separate true from false calls (§4).
- We introduce **AMCS**, a closed-form steering method derived from the diagnosed offset, and use it as a causal validation of **IBH** that mitigates over-calling while preserving valid tool calls (§5).

2 Related Work

2.1 Tool-Use Evaluation and the Over-Calling Phenomenon

LLM agents have evolved from early API and calculator integrations [5, 15, 16] to systems spanning large collections of real-world tools [6]. A corresponding evaluation ecosystem studies whether models can invoke tools correctly: selecting the right function, producing valid arguments, and completing tool-mediated tasks [7, 17, 18]. These benchmarks measure how well models call tools, but the gating question of when not to call is less central. When2Call [11] targets this decision directly and reveals a consistent asymmetry: models perform much better on call-required queries than on queries where not calling a tool is correct. Rather than another benchmark or data-side remedy, we ask what internal mechanism produces this asymmetry and diagnose over-calling as a measurable bias in the model's decision mapping.

2.2 Mechanistic Interpretability and Sparse Autoencoders

Understanding why over-calling arises requires tools that can dissect a model's internal computations. Mechanistic interpretability [19] provides this foundation, aiming to decompose model behavior into interpretable internal components and identify the structures responsible for specific outputs. Within this framework, Sparse Autoencoders (SAEs) have become a widely adopted method for recovering sparse, interpretable units from residual streams [12, 20, 14, 21, 22], with variants such as TopK [13], JumpReLU [23], and RouteSAE [24] improving reconstruction fidelity and feature quality.

We use SAEs to extract a structured feature basis for testing a mechanistic hypothesis. The same basis supports both estimating an activation-independent call offset and applying a counter-bias intervention along these directions as a causal test of the diagnosed offset.

3 Discovering Gating Feature Bases

Testing IBH requires a feature basis that reliably indexes the model's call/no-call decision. We construct two such bases, $\mathcal{C}$ and $\mathcal{N}$, by training SAEs on residual streams (§3.1), ranking features by the model's observed call/no-call behavior (§3.2), and validating them with linear probes (§3.3).

3.1 SAE Training

We train a separate TopK SAE [13] for each of six target models from three families at two scales: Qwen3.5-(4B, 9B), Gemma-3-it-(1B, 4B), and Ministral-3-Instruct-(3B, 8B). Given a residual-stream activation $\mathbf{h} \in \mathbb{R}^d$, the encoder and decoder compute:

$$\mathbf{z} = \mathrm{TopK}(\mathbf{W}_{\mathrm{enc}}(\mathbf{h} - \mathbf{b}_{\mathrm{pre}})) \in \mathbb{R}^M, \tag{1}$$

$$\hat{\mathbf{h}} = \mathbf{W}_{\mathrm{dec}}\,\mathbf{z} + \mathbf{b}_{\mathrm{pre}}, \tag{2}$$

where $M = 8d$, $K = \lfloor d/32 \rfloor$, and the columns of $\mathbf{W}_{\mathrm{dec}} \in \mathbb{R}^{d \times M}$ are constrained to unit norm. Training minimizes the reconstruction loss $\mathcal{L} = \|\mathbf{h} - \hat{\mathbf{h}}\|_2^2$. Each SAE is hooked at the output residual stream of a middle-to-late transformer block of its target model, where representations are sufficiently abstract for high-level decision features to emerge. Training follows a two-stage curriculum. Stage 1 trains on OpenWebText2 [25] (≈50M tokens) to learn a broad sparse feature basis grounded in general residual-stream geometry. We use this broad-corpus stage to reduce the risk that a narrow, domain-specific initialization leaves gaps in residual-stream coverage. Stage 2 continues on the When2Call training split (≈10M tokens) to adapt the dictionary to tool-use contexts while retaining the broad coverage from Stage 1. Training details, model-specific hook locations, and diagnostics are reported in Appendix D.

3.2 Behavior-Labeled Feature Ranking

Dataset and behavioral labeling. We build the discovery set from the When2Call evaluation split [11], using all contexts regardless of the original category. For each context x_i, we run the target LLM f_θ to obtain a response y_i and use an independent LLM judge to classify y_i into one of the four When2Call response types (judge prompt in Appendix B). Responses judged as `tool_call` form

$\mathcal{D}^+$ (CALL) and responses judged as `request_for_info` form $\mathcal{D}^-$ (NO_CALL). Responses in the other two categories are excluded. Each (x_i, d_i) records the model's observed gating decision rather than an external correctness label. Per-model response-label counts are reported in Appendix C.

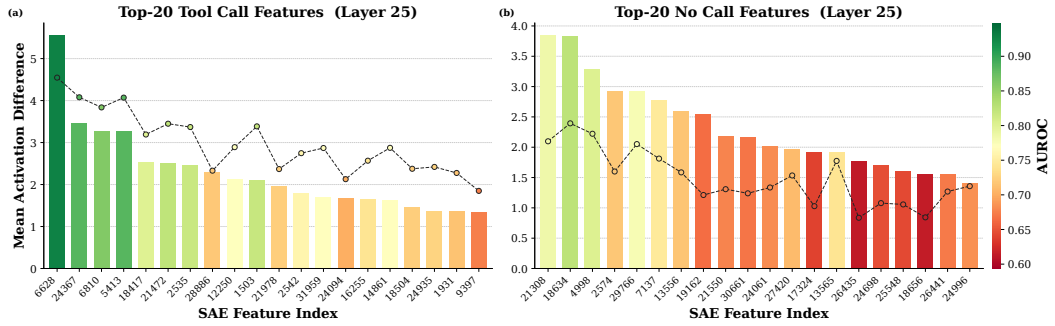

Figure 2: Top-ranked gating features discovered for Qwen3.5-9B. Bars show the mean activation difference between the target and contrast behavior-labeled sets, and the dashed line shows directional AUROC. Left: features associated with observed CALL decisions. Right: features associated with observed NO_CALL decisions.

Feature extraction and ranking. For each $x_i \in \mathcal{D}$, we extract the residual-stream activation $\mathbf{h}_i$ at the action-boundary position, *i.e.,* the position of the final prompt token before the first generated response token. We encode $\mathbf{h}_i$ with Eq. (1) to obtain SAE activation $\mathbf{z}_i \in \mathbb{R}^M$. We first identify CALL features by treating $\mathcal{D}^+$ as the target set and $\mathcal{D}^-$ as the reference set. For each SAE feature j, we compute the mean activation gap and the directional AUROC:

$$\Delta\mathrm{CE}_{\mathcal{C}}(j) \;=\; \mathbb{E}_{\mathcal{D}^+}[z_j] \;-\; \mathbb{E}_{\mathcal{D}^-}[z_j]\,, \tag{3}$$

$$\mathrm{AUROC}_{\mathcal{C}}(j) \;=\; P\big(z_j(x^+) > z_j(x^-)\big)\,, \quad x^+ \sim \mathcal{D}^+,\; x^- \sim \mathcal{D}^-. \tag{4}$$

For a ranking cutoff R, we keep the intersection of the top-R features under the two scores. The intersection favors features that are both strongly separated in mean activation and consistently discriminative across examples. The CALL pass yields $\mathcal{C}$. We obtain the NO_CALL feature set $\mathcal{N}$ by the same procedure after swapping the target and reference sets, using $\mathcal{D}^-$ against $\mathcal{D}^+$. Sweeping R gives the accuracy-sparsity trade-off used in validation.

Figure 2 shows the resulting top-ranked features for Qwen3.5-9B. Both panels exhibit features with large activation gaps and high directional AUROC, indicating that the discovered features are not artifacts of a small number of extreme activations. Analogous results for the remaining target models are reported in Appendix E.

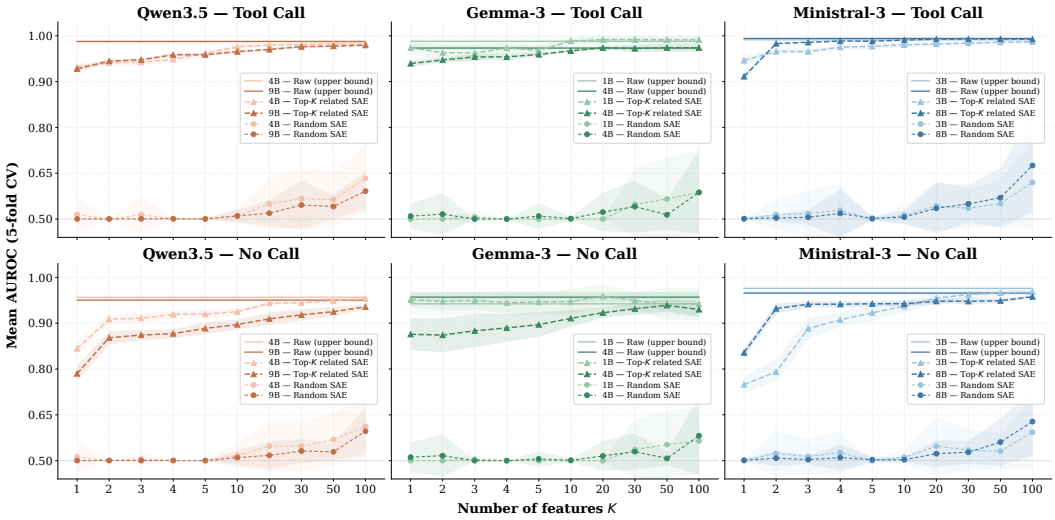

Figure 3: Discriminative validation across six target models. We train 5-fold cross-validated logistic probes with increasing numbers of selected SAE features and report mean AUROC.

3.3 Discriminative Validation

Probe setup. We validate the discovered features by testing whether a small linear probe trained on them predicts the model's behavior-labeled gating decision. For a selected feature set $\mathcal{S}$, we train a logistic regression on the sparse activations $\mathbf{z}_{\mathcal{S},i} = (z_j)_{j\in\mathcal{S}}$ at the action-boundary position:

$$\hat{d}_i \;=\; \sigma\big(\mathbf{w}^\top \mathbf{z}_{\mathcal{S},i} + b\big)\,, \tag{5}$$

where σ is the sigmoid function. Under 5-fold cross-validation, we compare three feature inputs: the top-ranked discovered SAE features, count-matched random SAE features, and the raw residual stream $\mathbf{h}_i$, where the last serves as an upper bound. We sweep the number of selected features $|\mathcal{S}|$ and report mean AUROC across folds.

Observation 1: A handful of discovered features predict the gating decision near the residual-stream upper bound. Figure 3 plots mean AUROC as $|\mathcal{S}|$ grows. Across all three model families and both call and no-call directions, the discovered features reach the raw-residual upper bound with only a handful of SAE dimensions (typically $K \leq 5$), while count-matched random features stay near chance. This proximity to the upper bound indicates that $\mathcal{C}$ and $\mathcal{N}$ capture nearly all of the linearly recoverable gating signal in the residual stream, establishing them as a reliable feature basis. Section 4 uses them to test whether the over-calling bias reduces to activation levels alone.

4 Diagnosing Intrinsic Decision Bias

This section diagnoses the mechanism of over-calling by asking whether the model's CALL preference is fully accounted for by CALL- and NO_CALL-feature activation levels, and by quantifying any activation-independent component of the decision mapping.

4.1 Formalizing the Hypothesis

Section 3 identifies two SAE feature sets, $\mathcal{C}$ and $\mathcal{N}$, that track the model's CALL and NO_CALL decisions. Using these features, we ask whether over-calling is fully explained by activation levels, or whether the decision mapping itself carries an activation-independent CALL bias. To make this question testable, we summarize the two feature groups by a signed activation margin and examine the model's decision as a function of that margin.

For each example i, let $z_{j,i}$ denote the activation of SAE feature j at the action-boundary position. Because SAE decoder columns are unit-norm by construction, feature mean activation can be compared directly in the SAE coordinate space. We define the signed activation margin as the difference between CALL- and NO_CALL-feature mean activation:

$$m_i = a_{\mathcal{C},i} - a_{\mathcal{N},i} = \frac{1}{|\mathcal{C}|}\sum_{j\in\mathcal{C}} z_{j,i} - \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}} z_{j,i}. \tag{6}$$

where positive m_i indicates stronger CALL-feature mean activation and negative m_i the reverse. Let $\hat{d}_i = 1$ denote that the model's response is judged as CALL, and $\hat{d}_i = 0$ otherwise. The margin gives a direct diagnostic axis on which the two accounts make opposite predictions, formalized below.

Activation-only account (H_{act}). The gating decision depends on the activation margin alone:

$$\Pr(\hat{d}_i = 1 \mid m_i) = \sigma(\beta m_i), \qquad \beta > 0. \tag{7}$$

Prediction. At activation parity, $\Pr(\hat{d}_i = 1 \mid m_i = 0) = 1/2$. The neutral decision boundary lies at $m_i = 0$.

Intrinsic Bias Hypothesis (H_{IBH}). The decision mapping contains an activation-independent CALL offset:

$$\Pr(\hat{d}_i = 1 \mid m_i) = \sigma(\beta m_i + \beta_0), \qquad \beta > 0, \quad \beta_0 > 0. \tag{8}$$

Prediction. At activation parity, $\Pr(\hat{d}_i = 1 \mid m_i = 0) = \sigma(\beta_0) > 1/2$. The neutral boundary shifts to $m^\star = -\beta_0/\beta < 0$.

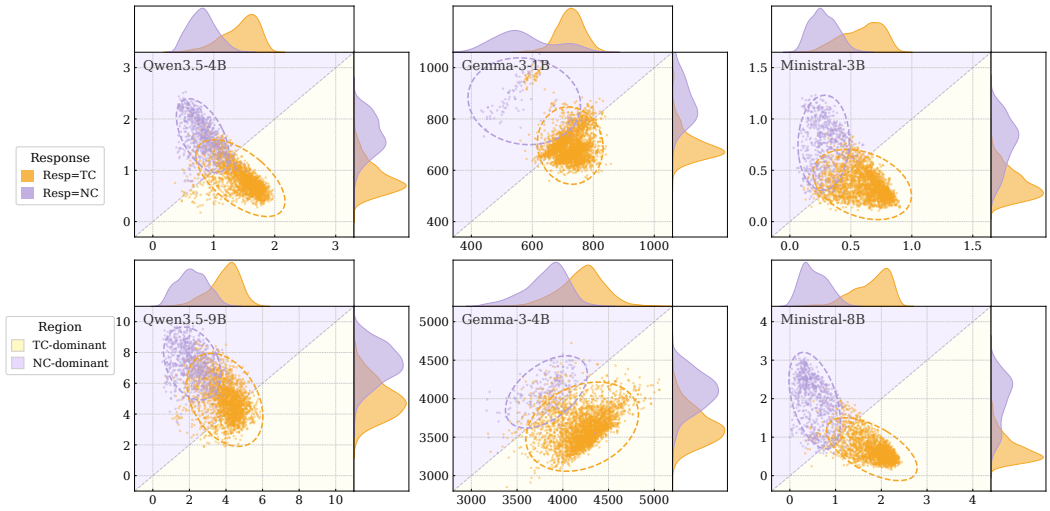

Figure 4: Response-conditioned activation geometry. Examples are grouped by the model's emitted decision: NO_CALL responses (purple) stay in the NO_CALL-dominant half-plane, while CALL responses (orange) extend across the parity diagonal into the same region. The response boundary therefore sits on the NO_CALL side of the diagonal rather than at activation parity.

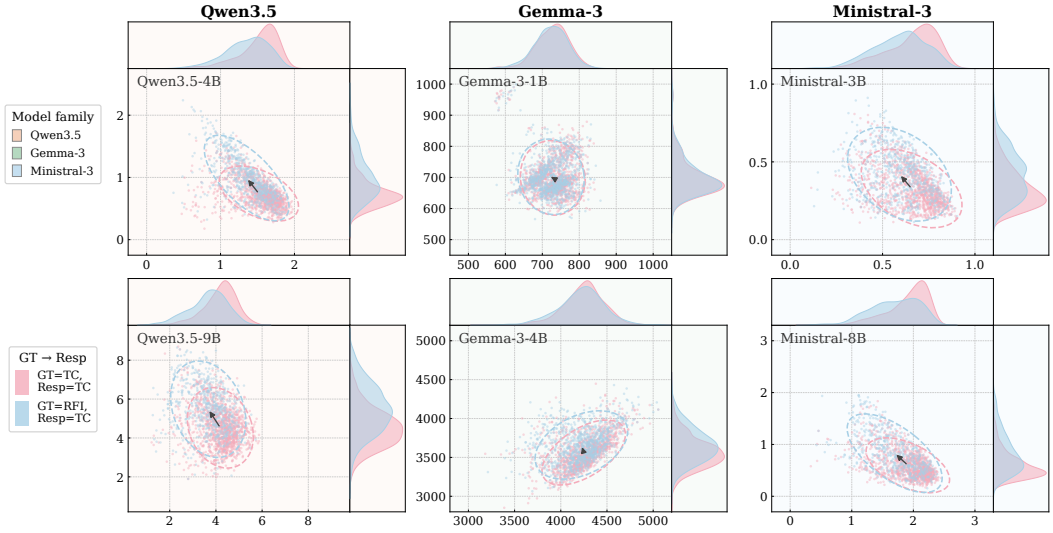

Figure 5: Input-conditioned activation geometry, restricted to examples for which the model emits a CALL. True calls (pink) lie in the CALL-dominant region, while false calls (blue) overlap the same region but sit consistently closer to the parity diagonal. True and false calls therefore remain separable along the $\mathcal{C}$ and $\mathcal{N}$ axes within a single emitted decision.

The two accounts therefore disagree on where the model becomes decision-neutral: at activation parity under H_{act}, only after NO_CALL evidence exceeds CALL evidence under H_{IBH}. We first examine the activation geometry of CALL- and NO_CALL-feature evidence from both response- and input-conditioned views, then estimate the offset term β_0 directly.

4.2 Evidence for Intrinsic Bias

To test these predictions, we examine each example's feature evidence in the plane $(a_{\mathcal{C}}, a_{\mathcal{N}})$, where the diagonal marks $m = 0$. We probe this geometry from two complementary views. A response-conditioned view asks whether the parity line separates emitted CALL from NO_CALL responses. An input-conditioned view then asks whether, among emitted CALL responses, true and false calls share the same feature geometry.

Observation 2: The response boundary is shifted into the NO_CALL-dominant half-plane. Across all six target models (Figure 4), NO_CALL responses concentrate well inside the NO_CALL-dominant half-plane, with the model emitting NO_CALL only when NO_CALL-feature mean activation clearly exceeds CALL-feature mean activation. CALL responses, by contrast, extend across the parity line into regions where NO_CALL features are comparable to or stronger than CALL features. This shift is systematic: the model carries a CALL preference that persists even when feature evidence already favors NO_CALL. This is the directional signature of H_{IBH}.

Observation 3: Among emitted calls, false calls lie closer to the NO_CALL side than true calls. Conditioning on emitted CALL responses (Figure 5), true and false calls occupy distinct activation distributions: across all six models, the false-call mass sits consistently closer to the parity diagonal than the true-call mass. The same intrinsic CALL preference is therefore still acting inside the CALL subset, pushing examples with weaker CALL evidence over the response boundary. Crucially, true and false calls remain separable along the $\mathcal{C}$ and $\mathcal{N}$ axes, giving Section 5 a directional lever: a counter-bias along the SAE decoder directions of $\mathcal{C}$ and $\mathcal{N}$ can push the shifted false-call mass back across the boundary while leaving the more deeply $\mathcal{C}$-dominant true calls in place.

Together, Observations 2 and 3 establish the qualitative shape of H_{IBH}: the bias surfaces both in where the response boundary sits and in which emitted calls turn out wrong. We now estimate β_0 to make this offset quantitative.

4.3 Quantifying the Bias Offset

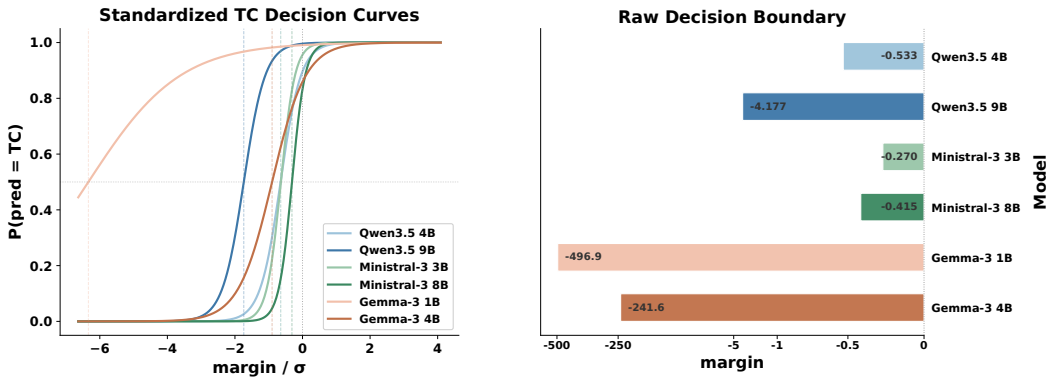

Figure 6: Logistic estimates of intrinsic CALL bias. **Left**: fitted decision curves with margin standardized within model, so the six curves can be visually compared. **Right**: raw neutral boundaries $\hat{m}^{\star}$. All boundaries are negative, meaning the model becomes decision-neutral only when NO_CALL-feature mean activation exceeds CALL-feature mean activation.

For each model, we fit Eq. 8 on cached SAE activations and observed model decisions, yielding empirical estimates $(\hat{\beta}, \hat{\beta}_0)$. The intercept $\hat{\beta}_0$ sets the CALL probability at activation parity, and the neutral boundary $\hat{m}^{\star} = -\hat{\beta}_0/\hat{\beta}$ marks the margin at which $\Pr(\hat{d}_i = 1) = 1/2$. The left panel of Figure 6 expresses margin in within-model standard-deviation units, aligning the six fitted curves: each crosses 0.5 at a negative margin and sits above 0.5 at activation parity. The right panel reports the raw $\hat{m}^{\star}$, which are negative for all six models and span several orders of magnitude across families. Across all six models, $\hat{m}^{\star} < 0$ supports H_{IBH}: parity between CALL- and NO_CALL-feature mean activations does not bring the model to decision-neutrality. The much larger raw magnitudes for Gemma likely reflect the larger numerical scale of its activations, so $\hat{m}^{\star}$ should be read as a within-model bias measure rather than a cross-model strength.

5 Causal Steering Experiments

If the offset β_0 in Section 4 is part of the mechanism that produces over-calling, counteracting it along the same SAE feature directions should rebalance CALL and NO_CALL decisions and improve overall accuracy. We instantiate this causal test with Adaptive Margin-Calibrated Steering (**AMCS**), a closed-form activation steering method derived from the fitted decision-margin model.

5.1 Adaptive Margin-Calibrated Steering

Closed-form calibration. AMCS reuses the signed activation margin in Eq. (6). For a steering budget of r features per side, let $\mathcal{C}_r \subseteq \mathcal{C}$ and $\mathcal{N}_r \subseteq \mathcal{N}$ denote the top-ranked CALL and NO_CALL features from Section 3.2. On cached calibration activations and observed model decisions, we refit the diagnostic margin model on this restricted basis using the same margin definition as Section 4.1:

$$\Pr(\hat{d} = 1 \mid m_r) = \sigma(\beta_r m_r + \beta_{0,r}), \tag{9}$$

where $\hat{d} = 1$ denotes an observed CALL response, m_r is the signed margin recomputed on $\mathcal{C}_r \cup \mathcal{N}_r$, and $\beta_r, \beta_{0,r}$ are the fitted slope and CALL offset under budget r. To remove this offset, AMCS shifts every margin by the same amount δ_r so that the calibrated logit matches the unbiased one, $\beta_r(m + \delta_r) + \beta_{0,r} = \beta_r m$ for all m. This yields the closed-form:

$$\delta_r = -\frac{\beta_{0,r}}{\beta_r}, \tag{10}$$

so the steering strength is fixed by the diagnosed bias rather than tuned on a validation set.

Steering vector and intervention. Let $\mathbf{d}_j$ be the SAE decoder column for feature j. To allocate the shift across selected features, we measure each feature's activation gap between CALL-decision and NO_CALL-decision responses on the calibration set, $\Delta_j^{\mathcal{C}}$ for $j \in \mathcal{C}_r$ and $\Delta_j^{\mathcal{N}}$ for $j \in \mathcal{N}_r$, and normalize absolute gaps within each side:

$$\omega_j^{\mathcal{C}} = \frac{|\Delta_j^{\mathcal{C}}|}{\sum_{k \in \mathcal{C}_r} |\Delta_k^{\mathcal{C}}|}, \qquad \omega_j^{\mathcal{N}} = \frac{|\Delta_j^{\mathcal{N}}|}{\sum_{k \in \mathcal{N}_r} |\Delta_k^{\mathcal{N}}|}. \tag{11}$$

These weights only set allocation, with the magnitude fixed by δ_r. The steering vector is then:

$$\mathbf{v}_r = \underbrace{\alpha r \delta_r \sum_{j \in \mathcal{C}_r} \omega_j^{\mathcal{C}} \mathbf{d}_j}_{\text{CALL } \textit{suppression } (\delta_r < 0)} + \underbrace{(1-\alpha) r (-\delta_r) \sum_{j \in \mathcal{N}_r} \omega_j^{\mathcal{N}} \mathbf{d}_j}_{\text{NO_CALL } \textit{enhancement } (-\delta_r > 0)}, \tag{12}$$

where $\alpha \in [0, 1]$ trades off the correction between CALL-side and NO_CALL-side decoder directions, and the factor r cancels the per-side averaging in m_r so that $\mathbf{v}_r$ targets a total margin shift of δ_r under a local linear SAE approximation. At inference, we add $\mathbf{v}_r$ at the SAE hook layer throughout autoregressive generation, $\mathbf{H}_\ell \leftarrow \mathbf{H}_\ell + \mathbf{v}_r$, costing one broadcast addition per token. Implementation details are in Appendix F.

5.2 Causal Validation Across Models

Setup. We evaluate AMCS on When2Call across six instruction-tuned models from three families: Qwen3.5-(4B, 9B), Gemma-3-it-(1B, 4B), and Ministral-3-Instruct-(3B, 8B), reusing the SAE and diagnosis layer of Section 4. AMCS fits β_r and $\beta_{0,r}$ on a calibration split and is evaluated on a held-out test split, with the balance set to $\alpha = 0.8$. We scan the steering budget $r \in \{5, 10, 15, 20, 25, 30\}$ and report the mean across r. We compare against three reference interventions, Prompt, Suppress, and Promote, defined in Table 1. The latter two each cover only one half of the AMCS shift, isolating the effect of the closed-form two-sided calibration.

Results. Each of the three reference interventions fails in a characteristic way (Table 1). Prompt acts at the surface and overshoots: on Ministral-3-8B it raises no-call accuracy by 45 points but cuts tool-call accuracy by 56 points, dropping Overall below *Init*. Suppress and Promote act in the right space but only on one side of the margin: on Qwen3.5-4B, each moves no-call accuracy by under 10 points and Overall by under 3.2 points. AMCS, by combining both sides with magnitude set by β_0, keeps tool-call accuracy within 5 points of *Init* on five of six models[3] while raising no-call accuracy by 4 to 17 points, giving the best Overall on Qwen3.5-4B, Qwen3.5-9B, and Ministral-3-3B, and landing within 0.6 points of the best on Ministral-3-8B. These gains come from cancelling the diagnosed offset β_0 along the same SAE directions, the causal counterpart of IBH.

[3]On the two Gemma-3 models, true and false calls show little separation along the $\mathcal{C}$ and $\mathcal{N}$ axes (Figure 5), so the SAE feature basis offers little leverage for any margin-based intervention. Gemma-3-4B is a null case, with Overall nearly unchanged, and Gemma-3-1B is a failure case, with Overall falling by 1.64 points. AMCS is therefore a model-conditional intervention.

Table 1: Causal validation of AMCS on When2Call (%). TC Acc and NC Acc denote tool-call and no-call accuracy. *Init* is the unmodified model. Prompt appends "When the user's request lacks necessary details, ask before taking action." to the user prompt. Suppress scales the top-ranked CALL feature activations by 0.5. Promote scales the top-ranked NO_CALL feature activations by 1.5. Subscripts give the absolute change versus *Init* in percentage points (positive / negative).

Metric	Method	Qwen3.5-4B	Qwen3.5-9B	Gemma3-1B	Gemma3-4B	Ministral3-3B	Ministral3-8B
TC Acc	*Init*	*95.98*	*96.45*	*98.61*	*99.23*	*91.66*	*91.97*
	Prompt	68.11$_{-27.87}$	67.72$_{-28.73}$	94.67$_{-3.94}$	58.22$_{-41.01}$	61.62$_{-30.04}$	35.91$_{-56.06}$
	Suppress	93.48$_{-2.50}$	94.61$_{-1.84}$	93.84$_{-4.77}$	94.34$_{-4.89}$	86.75$_{-4.91}$	85.34$_{-6.63}$
	Promote	93.82$_{-2.16}$	96.88$_{+0.43}$	98.38$_{-0.23}$	98.53$_{-0.70}$	90.07$_{-1.59}$	89.92$_{-2.05}$
	Ours	91.82$_{-4.16}$	91.72$_{-4.73}$	89.77$_{-8.84}$	99.18$_{-0.05}$	86.66$_{-5.00}$	89.90$_{-2.07}$
NC Acc	*Init*	*37.76*	*37.57*	*2.26*	*5.56*	*34.65*	*41.71*
	Prompt	77.31$_{+39.55}$	79.38$_{+41.81}$	15.91$_{+13.65}$	59.23$_{+53.67}$	71.56$_{+36.91}$	87.10$_{+45.39}$
	Suppress	44.65$_{+6.89}$	44.43$_{+6.86}$	4.35$_{+2.09}$	7.65$_{+2.09}$	45.75$_{+11.10}$	52.75$_{+11.04}$
	Promote	47.36$_{+9.60}$	34.27$_{-3.30}$	2.97$_{+0.71}$	7.64$_{+2.08}$	41.26$_{+6.61}$	46.91$_{+5.20}$
	Ours	51.07$_{+13.31}$	54.58$_{+17.01}$	9.40$_{+7.14}$	5.55$_{-0.01}$	45.91$_{+11.26}$	45.76$_{+4.05}$
Overall	*Init*	*69.75*	*69.88*	*55.20*	*57.11*	*65.97*	*69.33*
	Prompt	72.25$_{+2.50}$	72.97$_{+3.09}$	**59.19**$_{+3.99}$	**58.68**$_{+1.57}$	66.10$_{+0.13}$	58.97$_{-10.36}$
	Suppress	71.48$_{+1.73}$	72.00$_{+2.12}$	53.52$_{-1.68}$	55.28$_{-1.83}$	68.27$_{+2.30}$	**70.66**$_{+1.33}$
	Promote	72.89$_{+3.14}$	68.67$_{-1.21}$	55.39$_{+0.19}$	57.58$_{+0.47}$	68.08$_{+2.11}$	70.54$_{+1.21}$
	Ours	**73.47**$_{+3.72}$	**74.98**$_{+5.10}$	53.56$_{-1.64}$	57.00$_{-0.11}$	**68.30**$_{+2.33}$	70.10$_{+0.77}$

6 Transfer and Robustness

This section asks whether the offset and intervention transfer to another benchmark, whether AMCS preserves valid calls, and whether the IBH offset depends on the SAE and margin choices used above. Detailed results are in Appendix G, and results on models above 10B, a pretrained checkpoint, and reasoning-enabled decoding are in Appendix H.

6.1 Cross-Benchmark Transfer

We transfer the diagnosis and the intervention to a 100-example ACEBench [26] set, with the When2Call SAE, CALL/NO_CALL features, and steering directions held fixed. The split, the no-call scoring rule, margin normalization, and other details are in Appendix G.1.

The frozen basis reproduces IBH on the four models whose CALL/NO_CALL boundary is identifiable. For Qwen3.5-4B, Qwen3.5-9B, Ministral-3-3B, and Ministral-3-8B, the 95% bootstrap interval for $\Pr(\hat{d} = 1 \mid m = 0)$ lies above 0.5, and the interval for $m^\star$ lies below zero. Gemma-3-1B and Gemma-3-4B yield 99/0 and 98/2 CALL/NO_CALL decisions, so the parity regression is not estimable.

Steering is evaluated on these four models. Gemma is excluded because its CALL/NO_CALL separation is already weak on When2Call (Section 5.2) and its ACEBench decisions collapse to tool calls. Contrastive Activation Addition (**CAA**) [27] uses the same hook layer, with its coefficient selected separately per model on When2Call. AMCS uses one frozen configuration, $r = 25$ and $\alpha = 0.8$, for all four models. Neither method uses ACEBench labels to choose its direction or its strength.

Table 2: Frozen transfer to ACEBench (%). Each cell is overall / call / no-call accuracy. Baselines follow Table 1. CAA uses a per-model coefficient selected on When2Call, and AMCS uses $r = 25$, $\alpha = 0.8$ for all four models. **Bold** and underline mark the best and second best macro averages.

Method	Qwen3.5-4B	Qwen3.5-9B	Ministral-3-3B	Ministral-3-8B	Macro
Init	76 / 98 / 54	70 / 100 / 40	81 / 94 / 68	85 / 98 / 72	78.00 / **97.50** / 58.50
Prompt	82 / 72 / 92	91 / 92 / 90	86 / 76 / 96	69 / 42 / 96	82.00 / 70.50 / **93.50**
Suppress	79 / 94 / 64	76 / 100 / 52	84 / 92 / 76	85 / 90 / 80	81.00 / 94.00 / 68.00
Promote	85 / 98 / 72	75 / 100 / 50	85 / 94 / 76	87 / 96 / 78	<u>83.00</u> / <u>97.00</u> / 69.00
CAA	84 / 98 / 70	77 / 100 / 54	83 / 92 / 74	87 / 96 / 78	82.75 / 96.50 / 69.00
AMCS	84 / 98 / 70	80 / 98 / 62	86 / 92 / 80	85 / 96 / 74	**83.75** / 96.00 / <u>71.50</u>

Table 3: SAE ablation for Qwen3.5-4B. Each row refits the diagnostic logistic model after rediscovering Top-25 features in that SAE. Original is the setting of Section 4. Raw $m^\star$ is not comparable across rows.

Setting	Layer	K	Seed	β	$\Pr(\hat{d}=1 \mid m=0)$	$m^\star$
Original	25	80	42	4.006	0.894	-0.533
Layer 22	22	80	42	4.652	0.742	-0.227
Layer 28	28	80	42	3.574	0.903	-0.626
$K/2$	25	40	42	3.209	0.903	-0.695
$2K$	25	160	42	4.203	0.935	-0.634
Seed 123	25	80	123	3.901	0.939	-0.703

Relative to Init, frozen AMCS raises macro-average overall accuracy from 78.00% to 83.75% and no-call accuracy from 58.50% to 71.50%, while call accuracy changes from 97.50% to 96.00%. Prompt reaches the highest no-call accuracy but lowers call accuracy to 70.50%. CAA, with a coefficient tuned per model, reaches 82.75 / 96.50 / 69.00. AMCS is not better on every model: it ties CAA on Qwen3.5-4B, leads by 3 points on Qwen3.5-9B and Ministral-3-3B, and trails by 2 points on Ministral-3-8B. With 100 examples per model these margins are small, but a configuration fixed on When2Call still raises overall and no-call accuracy with call accuracy close to Init.

6.2 Preservation of Valid Calls

Call accuracy only records whether a tool is called, so we also check whether AMCS keeps valid calls correct. On all 1,295 ground-truth tool-call examples in When2Call, we pair each unsteered response with its AMCS response and score function names and arguments with a schema-conditioned judge (Appendix G.2). Across the five models other than Gemma-3-1B, AMCS preserves 95.73% of the unsteered tool calls, and 97.3% of the flips to NO_CALL fall in the lower-margin half. On the calls preserved by both responses, joint function-and-argument accuracy changes by -0.27 points on average. Gemma-3-1B is a failure case: even a reduced Top-6 intervention flips 24.18% of its tool calls and lowers joint accuracy by 20.87 points.

6.3 Robustness to SAE and Margin Choices

A setting supports IBH when $\beta > 0$, $\Pr(\hat{d}=1 \mid m=0) > 0.5$, and $m^\star < 0$. For each of the six models we retrain five additional SAEs, changing one factor at a time (two hook layers, $K/2$, $2K$, and a second random seed), and rediscover Top-25 features in each new basis. All six settings for Qwen3.5-4B support IBH (Table 3), and 34 of the 36 settings across the six models do. The two exceptions, Ministral-3-3B at Layer 24 and Ministral-3-8B at $K/2$, keep a positive slope but yield a slightly positive $m^\star$. On the original SAE, IBH also holds in 35 of 36 feature-budget settings and in all 18 margin-construction settings. We therefore claim robustness to the tested SAE and margin choices (Appendix G.3), not invariance to every configuration.

7 Conclusion

Tool-using LLM agents must decide not only how to call tools, but also when not to call. This paper studied a systematic failure in that decision: models achieve high call accuracy while remaining much less reliable on no-call cases, producing over-calling. Using SAE feature bases for the CALL/NO_CALL gating decision, we showed that this asymmetry is not fully explained by feature activation levels. We formalized this as IBH: an activation-independent call offset that shifts the neutral boundary toward CALL, so even at activation parity the decision remains biased toward calling.

We then turned this diagnosis into AMCS, a closed-form steering method that counteracts the fitted offset along SAE decoder directions. The same offset remains under the tested SAE settings, and the intervention transfers to another benchmark with its directions held fixed. This unifies behavioral miscalibration, feature-level mechanism, and causal intervention. Over-calling is therefore not only an empirical artifact of tool-use benchmarks. It is a mechanistic object that can be measured, modeled, and causally adjusted.

Acknowledgments and Disclosure of Funding

This work is supported by Shanghai Artificial Intelligence Laboratory and the National Natural Science Foundation of China (62625608, 62476279).

References

[1] OpenAI. Openai GPT-5 system card. *CoRR*, abs/2601.03267, 2026.

[2] Anthropic. Claude opus 4.6 system card. Technical report, Anthropic, February 2026.

[3] Google DeepMind. Gemma 4 model card. https://ai.google.dev/gemma/docs/core/model_card_4, April 2026. Accessed: 2026-04-08.

[4] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026.

[5] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.

[6] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world APIs. In *ICLR*. OpenReview.net, 2024.

[7] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. *CoRR*, abs/2305.15334, 2023.

[8] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *ICLR*. OpenReview.net, 2025.

[9] Gemma Team. Gemma 3 technical report. *CoRR*, abs/2503.19786, 2025.

[10] Mistral AI. Ministral 3. *CoRR*, abs/2601.08584, 2026.

[11] Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2call: When (not) to call tools. In *NAACL (Long Papers)*, pages 3391–3409. Association for Computational Linguistics, 2025.

[12] Robert Huben, Hoagy Cunningham, Logan Riggs, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*. OpenReview.net, 2024.

[13] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. 2025.

[14] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[15] Ehud Karpas, Omri Abend, Yonatan Berant, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai Shalev-Shwartz, Amnon Shashua, and Moshe Tennenholtz. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *CoRR*, abs/2205.00445, 2022.

[16] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*. OpenReview.net, 2023.

[17] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *ICLR*. OpenReview.net, 2024.

[18] Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[19] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J. Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability. *Trans. Mach. Learn. Res.*, 2025, 2025.

[20] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

[21] Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Tao Liang, Hengxing Cai, and Xiang Wang. Interpretable reward model via sparse autoencoder. In *AAAI*, pages 34808–34816. AAAI Press, 2026.

[22] Ziheng Peng, Huiqi Deng, Haoran Jing, Xuankun Rong, Jiahui Han, Xiting Wang, Na Zou, and Xia Hu. Do llms know their vulnerable scenarios? *CoRR*, abs/2607.23496, 2026.

[23] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *CoRR*, abs/2407.14435, 2024.

[24] Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models. In *EMNLP*, pages 6801–6815. Association for Computational Linguistics, 2025.

[25] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021.

[26] Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Yuefeng Huang, Xiangcheng Liu, Xinzhi Wang, and Wu Liu. ACEBench: A comprehensive evaluation of LLM tool usage. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 12970–12998. Association for Computational Linguistics, 2025.

[27] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *ACL (Long Papers)*, pages 15504–15522. Association for Computational Linguistics, 2024.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

A Limitations and Discussions

Limitations. Appendix H.2 compares Qwen3.5-4B with its official pretrained checkpoint. The offset is already detectable before post-training, but the comparison does not identify which training stage produces it. AMCS remains an inference-time correction, not a training-side fix. The intervention relies on a local linear approximation that translates decoder-direction shifts into margin shifts. The experiments in this paper cover only single-turn decisions, and we leave multi-turn interactions to future work.

Toward controllable calling effort. Modern LLM interfaces expose knobs for reasoning effort or compute budget, but tool use lacks a comparable control over how readily a model invokes external actions rather than verifying intent with the user. The diagnosed call offset behaves like such a knob: shifting it tunes the balance between eager calling and seeking user verification, without altering the underlying tool-use capability. AMCS is therefore not only a correction for over-calling, but a first step toward a "calling effort" interface where users or systems can set a task-specific point on this call-versus-verify axis.

Toward agent interpretability. Interpreting agents requires more than explaining isolated outputs, since real agents act over multiple turns, condition on tool feedback, and update plans dynamically. This paper isolates one consequential unit of that broader problem: the call-versus-verify decision. We connect it to feature-level geometry and a causal intervention, so the mechanism can be measured and adjusted. Extending this analysis to long-horizon agents will require tracking such mechanisms across time, memory, and tool outputs.

B LLM as Judge Prompt

```
You are an expert at classifying responses from AI models.
Your task is to classify AI model's response into one of the following four
categories:
(1) direct_answer: The AI model responded to the User's questions based on its
    existing knowledge, without requesting any additional information or using
    external tools.
(2) tool_call: The AI model decided to use a tool from the provided ones to
    help answer the question.
(3) request_for_info: The AI model requested additional information from the
    User.
(4) cannot_answer: The AI model refused to answer the User's questions by
    acknowledging the lack of required capabilities.
You should not judge whether the AI model's response is accurate or not. Only
provide the classification into one of these four categories.
<AVAILABLE_TOOLS> {tools} </AVAILABLE_TOOLS>
<USER_QUESTION> {question} </USER_QUESTION>
<AI_MODEL_RESPONSE> {response} </AI_MODEL_RESPONSE>
Please provide the classification in the following JSON format:
{"classification": "<direct_answer | tool_call | request_for_info |
cannot_answer>"}
Respond only in the prescribed JSON format.
```

C Behavior-Labeled Discovery Set

We build the discovery set from the full When2Call evaluation split, running each target model on all contexts regardless of the original category. The LLM judge classifies each response into one of the four When2Call response types: `tool_call`, `request_for_info`, `direct_answer`, and `cannot_answer`. We then form $\mathcal{D}^+$ from `tool_call` responses and $\mathcal{D}^-$ from `request_for_info`

Table 4: Per-model response counts across the four When2Call categories assigned by the LLM judge. We form $\mathcal{D}^+$ from `tool_call` responses and $\mathcal{D}^-$ from `request_for_info` responses, and exclude the remaining two categories from the discovery set.

Model	`tool_call`	`request_for_info`	`direct_answer`	`cannot_answer`
Qwen3.5-4B	2163	751	272	466
Qwen3.5-9B	2151	758	257	486
Gemma-3-1b-it	3318	104	47	173
Gemma-3-4b-it	3037	223	207	185
Ministral-3-3B-Instruct	2408	603	363	278
Ministral-3-8B-Instruct	1993	731	382	546

responses. Responses in the other two categories describe plain answering and tool unavailability rather than the call/no-call gating decision and are excluded from the discovery set.

Table 4 reports per-model counts of judged responses across the four categories. The first two columns correspond to $|\mathcal{D}^+|$ and $|\mathcal{D}^-|$, whose cross-model variation reflects each model's response distribution.

D SAE Training Details

Table 5: SAE training configuration for each target model. d is the text-backbone residual width, L is the number of transformer blocks, ℓ is the zero-indexed hook block, $M = 8d$ is the SAE dictionary size, and $K = \lfloor d/32 \rfloor$ is the TopK sparsity.

Model	d	L	ℓ	M	K
Qwen3.5-4B	2560	32	25	20480	80
Qwen3.5-9B	4096	32	25	32768	128
Gemma-3-1b-it	1152	26	17	9216	36
Gemma-3-4b-it	2560	34	29	20480	80
Ministral-3-3B-Instruct	3072	26	21	24576	96
Ministral-3-8B-Instruct	4096	34	31	32768	128

We train SAEs on the text-backbone residual stream of each target model. Following the $M = 8d$ and $K = \lfloor d/32 \rfloor$ configuration introduced in §3.1, we instantiate per-model dimensions as listed in Table 5, where the hook block ℓ is chosen from the middle-to-late transformer blocks of each model under zero-based indexing. Both training stages use AdamW [28] with a learning rate of 5×10^{-4}, $\beta = (0.9, 0.999)$, a batch size of 16,384 tokens, and a warmup–stable–decay schedule (10%–80%–10%).

Training diagnostics. We monitor the SAE reconstruction loss throughout both training stages for each target model. Figure 7 groups the optimization traces by model family. Stage 1 losses drop sharply within the first few hundred steps and then settle onto a stable plateau, indicating that the broad-corpus dictionary has converged on OpenWebText2. At the Stage 2 boundary the loss jumps because the residual-stream distribution shifts to When2Call, and then descends smoothly to a new plateau as the dictionary adapts to tool-use contexts. All six SAEs follow this pattern, confirming stable optimization across families and scales.

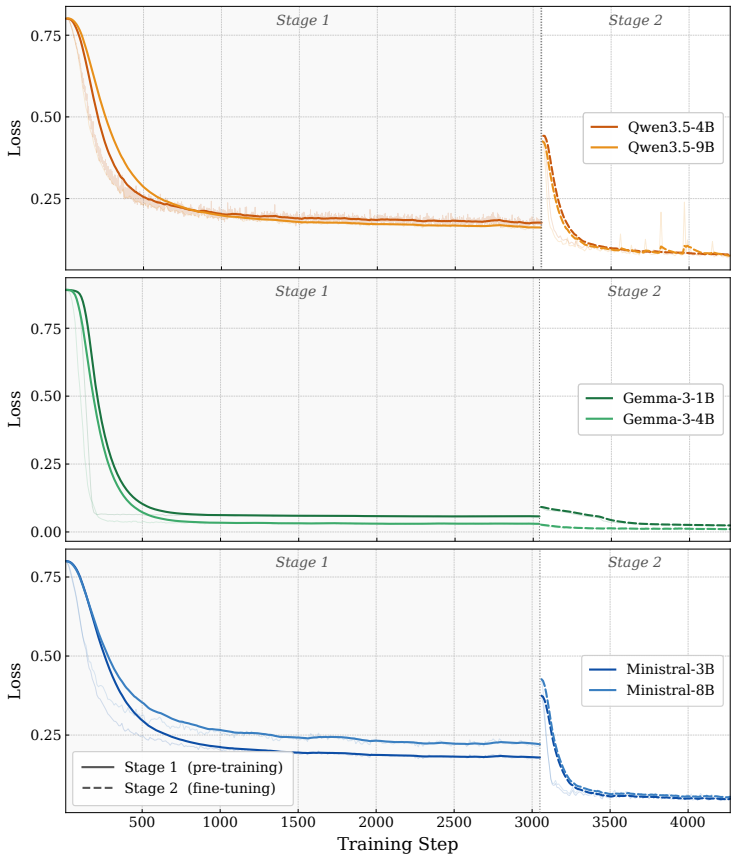

Figure 7: SAE reconstruction loss across the two-stage training curriculum, grouped by model family. Solid traces show Stage 1 (broad-corpus pre-training on OpenWebText2) and dashed traces show Stage 2 (When2Call adaptation), separated by the vertical divider. Light curves are raw per-step losses and bold curves are running averages. Within each panel, the two color shades distinguish the two model scales.

E Gating Feature Discovery Across Models

Figure 8 situates the discovered gating features inside each SAE's full feature dictionary. For the Qwen3.5 and Ministral-3 families, both TOOL_CALL and NO_CALL features collapse into a single tight cluster on the UMAP manifold, indicating that the call/no-call signal is concentrated in a coherent region of the SAE feature space. Gemma instead spreads its top features across most of the manifold, so gating information is carried by individually informative but geometrically dispersed directions. Despite this difference in feature geometry, the discovery pipeline returns a small, behavior-aligned feature set in all three families, consistent with the near-upper-bound probe AUROC reported in §3.3.

We further ask how the discovered features behave on failure cases. Figures 9, 10, and 11 contrast per-feature mean activations between Tool-Call-failure contexts (where the model wrongly issues a tool call) and No-Call-success contexts (where the model correctly withholds the tool call). Across all six target models, the top-ranked TOOL_CALL features are systematically overactivated under failure and the top-ranked NO_CALL features are systematically underactivated. This bidirectional shift is consistent with IBH: failures arise from a coordinated imbalance between the two feature groups rather than from a single group acting in isolation, and the pattern holds even where the UMAP geometry differs across families.

Figures 12 and 13 trace the top-10 TOOL_CALL and NO_CALL features back to input tokens on two representative contexts. Both groups draw most of their activation from the tool schema and the user query.

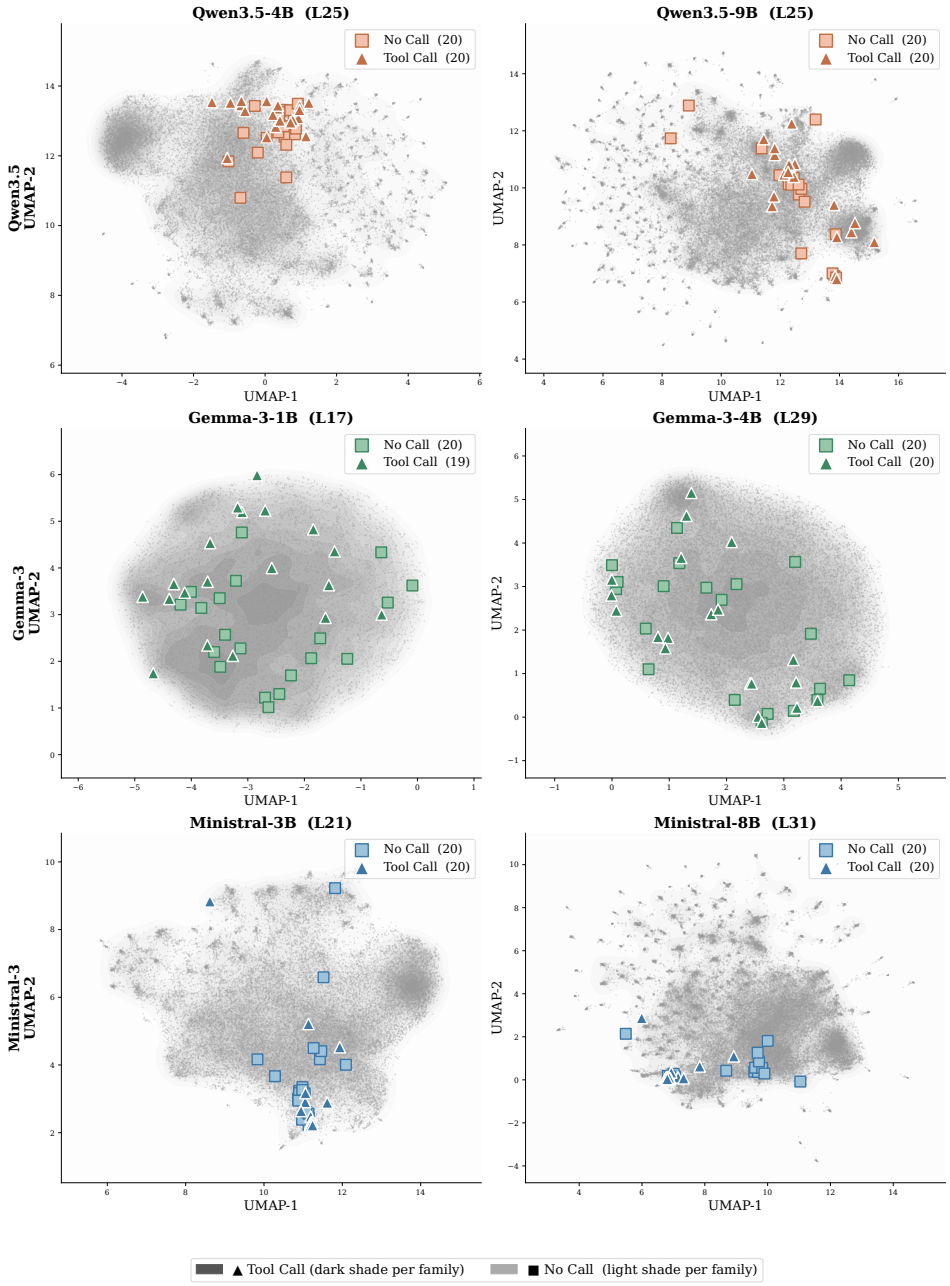

Figure 8: UMAP visualization of the SAE feature dictionary for each target model. Gray points are all SAE features projected to 2D, and the colored markers highlight the top-20 TOOL_CALL (triangles) and NO_CALL (squares) features returned by the discovery pipeline. Panel titles list the model name and the SAE hook layer.

F Adaptive Margin-Calibrated Steering Details

This section records the calibration and inference steps needed to reproduce the steering vectors, complementing the derivation in §5.1.

Offline calibration. For each target model, we cache residual-stream states at the selected SAE hook layer and the corresponding judged model decisions on the calibration set, encode each state with the model-specific SAE, and restrict to examples whose decision $\hat{d}_x \in \{\text{CALL}, \text{NO_CALL}\}$.

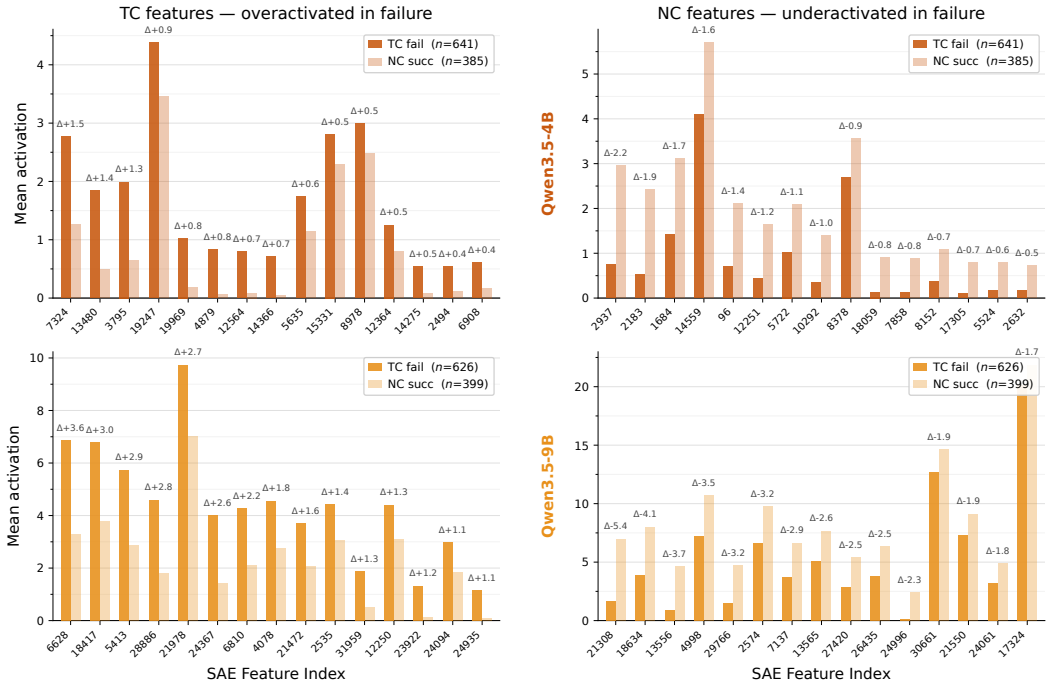

Figure 9: Per-feature mean SAE activation on Tool-Call-failure (over-calling) vs. No-Call-success contexts for Qwen3.5-4B (top) and Qwen3.5-9B (bottom), with top-ranked TOOL_CALL features (left) overactivated and NO_CALL features (right) underactivated under failure.

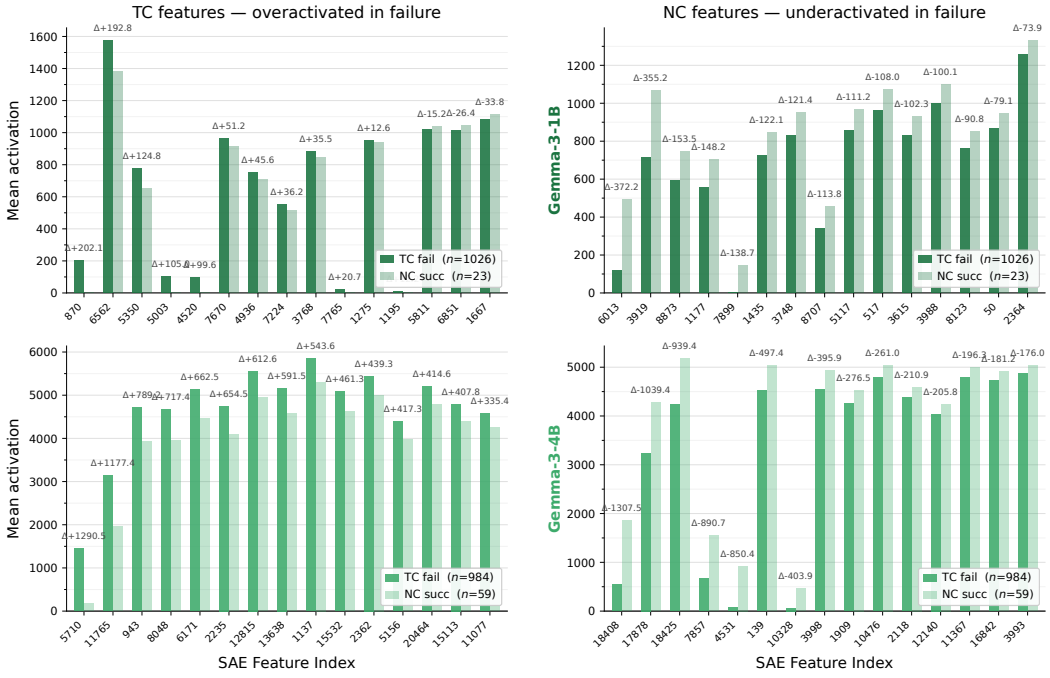

Figure 10: Same as Figure 9, for Gemma-3-1B (top) and Gemma-3-4B (bottom).

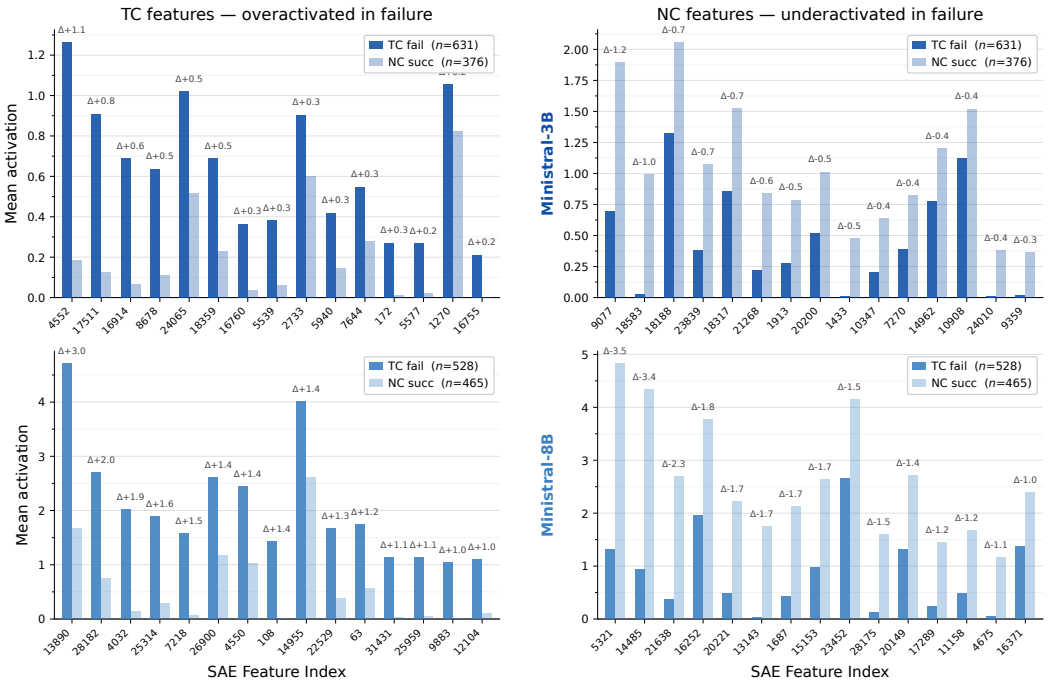

Figure 11: Same as Figure 9, for Ministral-3-3B (top) and Ministral-8B (bottom).

On this set, we compute the signed margin $m_r(x)$ over the restricted basis $\mathcal{C}_r \cup \mathcal{N}_r$ following the definition in §4.1, fit Eq. (9), and keep the candidate only when $\beta_r > 0$ so that the margin orders CALL decisions in the expected direction. We then obtain δ_r via Eq. (10), normalize feature weights as in Eq. (11), and form $\mathbf{v}_r$ from Eq. (12). Each candidate budget r induces a distinct feature basis and bias estimate, so calibration is repeated independently for each value of r.

Margin-shift approximation. Decomposing the contribution of $\mathbf{v}_r$ to the two sides of m_r under a local linear SAE approximation gives $\Delta m_r \approx \alpha\delta_r + (1-\alpha)\delta_r = \delta_r$. Because decoder directions need not form an orthogonal basis, we treat this as a calibrated target and verify the realized margin shift when reporting steering results.

Algorithm. Algorithm 1 summarizes the full offline calibration and online inference procedure.

Complexity. For a fixed feature budget r, constructing $\mathbf{v}_r$ requires a weighted sum of $2r$ decoder columns and costs $O(rd)$. At inference time, AMCS adds one vector of dimension d to the hooked residual stream. It introduces no extra model calls, no iterative search over steering coefficients, and no additional learned parameters.



Figure 12: Token-level attribution of the top-10 TOOL_CALL features on a representative context where the model correctly issues a tool call. Green intensity is proportional to each token's contribution to the summed feature activation (x-axis). The strongest signal concentrates in the tool schema definitions and the user query, consistent with these features encoding tool-invocation intent.



Figure 13: Token-level attribution of the top-10 NO_CALL features on a representative context where the model correctly withholds a tool call and requests missing information instead. Red intensity is proportional to each token's contribution to the summed feature activation (x-axis). The signal concentrates in the underspecified user query and tool schema, consistent with these features encoding information-insufficiency.

Algorithm 1 Adaptive Margin-Calibrated Steering (AMCS)

Require: Cached hidden states $\{\mathbf{h}_x\}_{x\in\mathcal{D}_{\text{cal}}}$, cached model decisions $\{\hat{d}_x\}$, SAE encoder and decoder, hook layer ℓ, feature rankings $\mathcal{C}$ and $\mathcal{N}$, attribution gaps $\{\Delta_j^{\mathcal{C}}\}$ and $\{\Delta_j^{\mathcal{N}}\}$, feature budget r and allocation coefficient α

Ensure: Steering vector $\mathbf{v}_r$ and steered response $\tilde{y}$

Offline calibration
Select $\mathcal{C}_r \leftarrow \text{Top}_r(\mathcal{C})$ and $\mathcal{N}_r \leftarrow \text{Top}_r(\mathcal{N})$
for all $x \in \mathcal{D}_{\text{cal}}$ **do**
  $\mathbf{z}_x \leftarrow \text{SAEEnc}(\mathbf{h}_x)$
  $m_r(x) \leftarrow \frac{1}{r}\sum_{j\in\mathcal{C}_r} \|\mathbf{d}_j\|_2 z_{x,j} - \frac{1}{r}\sum_{j\in\mathcal{N}_r} \|\mathbf{d}_j\|_2 z_{x,j}$
end for
$\mathcal{D}_{\text{bin}} \leftarrow \{x \in \mathcal{D}_{\text{cal}} : \hat{d}_x \in \{\text{CALL}, \text{NO_CALL}\}\}$
Fit $\Pr(\hat{d} = 1 \mid m_r) = \sigma(\beta_r m_r + \beta_{0,r})$ on $\mathcal{D}_{\text{bin}}$
if $\beta_r \le 0$ **then return** SKIP
end if
$\delta_r \leftarrow -\beta_{0,r}/\beta_r$ ▷ Eq. (10)
$\omega_j^{\mathcal{C}} \leftarrow |\Delta_j^{\mathcal{C}}| / \sum_{k\in\mathcal{C}_r} |\Delta_k^{\mathcal{C}}|$ for $j \in \mathcal{C}_r$
$\omega_j^{\mathcal{N}} \leftarrow |\Delta_j^{\mathcal{N}}| / \sum_{k\in\mathcal{N}_r} |\Delta_k^{\mathcal{N}}|$ for $j \in \mathcal{N}_r$
$\mathbf{v}_r \leftarrow \alpha r \delta_r \sum_{j\in\mathcal{C}_r} \omega_j^{\mathcal{C}} \mathbf{d}_j + (1-\alpha) r(-\delta_r) \sum_{j\in\mathcal{N}_r} \omega_j^{\mathcal{N}} \mathbf{d}_j$

Online inference
Register a forward hook at layer ℓ that applies $\mathbf{H}_\ell \leftarrow \mathbf{H}_\ell + \mathbf{v}_r$
Generate $\tilde{y}$ with the hook active
Remove the hook
return $\mathbf{v}_r, \tilde{y}$

G Detailed Transfer and Robustness Results

G.1 Cross-Benchmark Transfer

Evaluation set. The ACEBench set has 100 examples: all 50 `Special-Incomplete` examples, whose gold decision is NO_CALL, and 50 randomly sampled `Normal` single-turn single-function examples, whose gold decision is CALL. A correct NO_CALL response on `Special-Incomplete` must request the missing information, so direct answers and refusals count as incorrect.

Diagnosis. Each model reuses its When2Call SAE and Top-25 CALL/NO_CALL features, and encodes the action-boundary activation with the frozen SAE. Margins are divided by their within-model standard deviation without centering, so $m = 0$ remains activation parity. The diagnostic model is fit on responses judged as CALL or NO_CALL, with percentile 95% intervals from 1,000 example-level bootstrap resamples. All resamples are identifiable for the four non-Gemma models, and in each of them the full interval for the parity probability lies above 0.5 and the full interval for $m^\star$ lies below zero (Table 6).

Table 6: IBH diagnosis on ACEBench with When2Call features held fixed. n counts responses judged as CALL or NO_CALL. Intervals are percentile 95% bootstrap intervals. Gemma rows are not estimable because NO_CALL decisions collapse.

Model	n (CALL/NO_CALL)	$\Pr(\hat{d} = 1 \mid m = 0)$	$m^\star$
Qwen3.5-4B	100 (72/28)	0.755 [0.648, 0.844]	-0.438 [-0.735, -0.209]
Qwen3.5-9B	100 (80/20)	0.990 [0.978, 0.997]	-2.174 [-2.689, -1.827]
Gemma-3-1B	99 (99/0)	–	–
Gemma-3-4B	100 (98/2)	–	–
Ministral-3-3B	99 (62/37)	0.744 [0.636, 0.828]	-0.346 [-0.540, -0.172]
Ministral-3-8B	98 (61/37)	0.644 [0.527, 0.749]	-0.174 [-0.329, -0.033]

Steering. Directions and strengths are fixed on When2Call and applied unchanged to ACEBench. CAA uses

$$v_{\text{CAA}} = \text{mean}(h_{\text{NO_CALL}}) - \text{mean}(h_{\text{CALL}}). \tag{13}$$

Its coefficient is selected per model from $\{0.2, 0.4, 0.6, 0.8\}$ on When2Call, as the setting with the highest overall accuracy among those that lower call accuracy by at most 5 points. The selected coefficients are 0.2, 0.4, 0.2, and 0.2 for Qwen3.5-4B, Qwen3.5-9B, Ministral-3-3B, and Ministral-3-8B. AMCS uses $r = 25$ and $\alpha = 0.8$ for all four models.

G.2 Preservation of Valid Calls

We pair the unsteered and AMCS responses on all 1,295 ground-truth tool-call examples in When2Call. Five models use the fixed Top-25 setting, and Gemma-3-1B uses a reduced Top-6 rerun. A separate schema-conditioned judge (Appendix I) scores the two responses independently. Joint accuracy requires an exact function name and argument keys and nested values that match under the tool schema, allowing only non-semantic differences such as JSON key order or `10` versus `10.0`. Unparseable calls are incorrect. The flip rate is computed where the unsteered model emits a tool call, and joint accuracy on the subset where both responses do (Table 7).

Table 7: Effect of AMCS on valid tool calls (When2Call examples whose gold decision is CALL). Flipped counts unsteered tool calls that AMCS turns into NO_CALL. Joint accuracy is the rate of exact function name and arguments, on the examples where both the unsteered and the AMCS response emit a tool call. Parentheses give the change in points.

Model	Flipped to NO_CALL	Joint accuracy, unsteered → AMCS
Qwen3.5-4B	48/1,238 (3.88%)	69.08% → 68.99% (−0.08)
Qwen3.5-9B	86/1,249 (6.89%)	73.09% → 71.80% (−1.29)
Gemma-3-1B, Top-6	301/1,245 (24.18%)	24.79% → 3.92% (−20.87)
Gemma-3-4B	4/1,284 (0.31%)	54.30% → 54.30% (0.00)
Ministral-3-3B	87/1,182 (7.36%)	57.35% → 57.63% (+0.27)
Ministral-3-8B	37/1,184 (3.12%)	60.85% → 60.59% (−0.26)

To locate the flips, we rank each model's unsteered tool-call examples by pre-AMCS margin, split them into four equal-count bins, and pool counts across models only after this split (Table 8). Lower margin means weaker CALL-side activation. On the five models other than Gemma-3-1B, AMCS preserves 95.73% of tool calls, 255 of the 262 flips (97.3%) fall in the lower-margin half, and joint accuracy changes by -0.27 points on average. Gemma-3-1B still flips 8.36% of tool calls in its highest-margin quartile. On these five models, AMCS therefore removes mainly the low-margin calls closest to the decision boundary and leaves the content of preserved calls nearly unchanged. Gemma-3-1B breaks this pattern, so AMCS calls for model-specific downstream checks rather than a universal guarantee that valid calls are preserved.

Table 8: Tool-call flips by pre-AMCS margin quartile. Q1 is the lowest-margin quartile.

Quartile	Five models other than Gemma-3-1B	Gemma-3-1B, Top-6
Q1	226/1,536 (14.71%)	137/312 (43.91%)
Q2	29/1,533 (1.89%)	79/311 (25.40%)
Q3	6/1,533 (0.39%)	59/311 (18.97%)
Q4	1/1,535 (0.07%)	26/311 (8.36%)

G.3 SAE and Margin Ablations

SAE retraining. For each model we compare the original SAE with five retrained SAEs that change one factor at a time: two hook layers, $K/2$, $2K$, and seed 123 instead of seed 42. Each new SAE follows the same two-stage training, Top-25 features are rediscovered in it, and the diagnostic model is refit on the decoder-norm-scaled raw-mean margin. Model outputs and judge labels stay fixed, so differences come only from the SAE. A setting supports IBH when $\beta > 0$, $\Pr(\hat{d} = 1 \mid m = 0) > 0.5$, and $m^\star < 0$. Raw $m^\star$ is not compared across settings, because independently trained SAEs differ in activation scale.

Table 9: IBH under SAE retraining for the five models not shown in Table 3. Each model has the original setting and five single-factor variants. Together with Qwen3.5-4B, these are the 36 settings summarized in Section 6.3.

Model	Setting	Layer	K	Seed	β	$\Pr(\hat{d}=1 \mid m=0)$	$m^\star$
Qwen3.5-9B	Original	25	128	42	1.338	0.996	-4.177
	Layer 22	22	128	42	2.187	0.939	-1.247
	Layer 28	28	128	42	1.083	0.865	-1.717
	$K/2$	25	64	42	1.394	0.945	-2.046
	$2K$	25	256	42	1.377	0.988	-3.211
	Seed 123	25	128	123	1.298	0.966	-2.584
Gemma-3-1B	Original	17	36	42	0.009	0.991	-496.909
	Layer 14	14	36	42	0.008	1.000	-950.271
	Layer 20	20	36	42	0.006	0.997	-904.578
	$K/2$	17	18	42	0.012	0.988	-368.191
	$2K$	17	72	42	0.017	0.873	-116.301
	Seed 123	17	36	123	0.009	0.997	-604.008
Gemma-3-4B	Original	29	80	42	0.008	0.860	-241.619
	Layer 26	26	80	42	0.003	0.993	-1439.793
	Layer 32	32	80	42	0.003	0.996	-1990.850
	$K/2$	29	40	42	0.002	1.000	-4760.214
	$2K$	29	160	42	0.006	0.964	-562.989
	Seed 123	29	80	123	0.008	0.957	-392.407
Ministral-3-3B	Original	21	96	42	9.403	0.927	-0.270
	Layer 18	18	96	42	13.926	0.890	-0.150
	Layer 24	24	96	42	5.024	0.422	0.063
	$K/2$	21	48	42	7.968	0.663	-0.085
	$2K$	21	192	42	9.135	0.913	-0.257
	Seed 123	21	96	123	8.888	0.928	-0.288
Ministral-3-8B	Original	31	128	42	3.988	0.840	-0.415
	Layer 28	28	128	42	4.461	0.913	-0.528
	Layer 24	24	128	42	7.562	0.846	-0.226
	$K/2$	31	64	42	3.902	0.433	0.069
	$2K$	31	256	42	4.059	0.612	-0.113
	Seed 123	31	128	123	3.606	0.852	-0.486

Thirty-four of the 36 settings support IBH (Tables 3 and 9). The two exceptions, Ministral-3-3B at Layer 24 and Ministral-3-8B at $K/2$, have a slightly positive $m^\star$ but positive slopes, 5.024 and 3.902. Their margins still agree with the emitted decision on 95.7% and 97.0% of examples when CALL activation is higher, and on 92.0% and 94.5% when NO_CALL activation is higher.

Feature budget. On the original SAE, each side uses its top-N features for $N \in \{5, 10, 15, 20, 25, 30\}$, with the raw mean-difference margin. Gemma-3-1B has only 19 CALL features, so its Top-20, Top-25, and Top-30 settings use 19 CALL features against 20, 25, and 30 NO_CALL features. IBH holds in 35 of the 36 settings (Table 10). The only miss is Gemma-3-4B at Top-5, where $\Pr(\hat{d}=1 \mid m=0) = 0.245$ and $m^\star = +517.893$. The Qwen3.5-4B Top-25 row lists $\beta = 4.008$, against 4.006 in Table 3 for the same SAE.

Margin construction. At Top-25 we compare three margins: the raw mean difference, a feature-standardized difference that divides each feature by its evaluation-set standard deviation, and the difference of the two side means divided by their sum. All 18 settings support IBH (Table 11). Raw $m^\star$ is not compared across constructions, because the units differ. None of the margins is mean-centered. For a centered margin $m' = m - c$, the fit becomes $\beta m + \beta_0 = \beta m' + (\beta_0 + \beta c)$, so $m' = 0$ marks the dataset mean rather than activation parity and the intercept moves with the arbitrary choice of c. A centered fit would no longer test the parity claim, so the ablations vary scale and feature weighting while keeping $m = 0$ at parity.

Table 10: Feature-budget ablation on the original SAE. Each side uses its top-N features, except Gemma-3-1B, whose CALL set contains only 19 features. The margin is the raw mean difference.

Model	N	Features	β	$\Pr(\hat{d}=1 \mid m=0)$	$m^\star$
Qwen3.5-4B	5	5/5	1.694	0.915	-1.402
	10	10/10	2.444	0.945	-1.166
	15	15/15	2.938	0.876	-0.664
	20	20/20	3.518	0.894	-0.605
	25	25/25	4.008	0.894	-0.533
	30	30/30	4.527	0.887	-0.454
Qwen3.5-9B	5	5/5	0.818	0.896	-2.628
	10	10/10	0.878	0.961	-3.664
	15	15/15	0.940	0.978	-4.018
	20	20/20	1.117	0.997	-5.354
	25	25/25	1.338	0.996	-4.177
	30	30/30	1.471	0.996	-3.694
Gemma-3-1B	5	5/5	0.005	0.926	-501.378
	10	10/10	0.007	0.974	-525.545
	15	15/15	0.007	0.987	-606.731
	20	19/20	0.009	0.992	-529.477
	25	19/25	0.009	0.991	-496.909
	30	19/30	0.010	0.985	-433.546
Gemma-3-4B	5	5/5	0.002	0.245	$+517.893$
	10	10/10	0.004	0.685	-215.102
	15	15/15	0.005	0.903	-430.247
	20	20/20	0.006	0.891	-326.064
	25	25/25	0.008	0.860	-241.619
	30	30/30	0.009	0.985	-471.839
Ministral-3-3B	5	5/5	3.985	0.930	-0.648
	10	10/10	6.407	0.921	-0.383
	15	15/15	7.209	0.929	-0.357
	20	20/20	8.294	0.903	-0.268
	25	25/25	9.403	0.927	-0.270
	30	30/30	10.186	0.907	-0.224
Ministral-3-8B	5	5/5	2.316	0.548	-0.084
	10	10/10	2.804	0.746	-0.385
	15	15/15	3.210	0.797	-0.426
	20	20/20	3.539	0.690	-0.225
	25	25/25	3.988	0.840	-0.415
	30	30/30	4.053	0.833	-0.396

H Scaling, Pretraining, and Reasoning

H.1 Models above 10B

The main experiments use models below 10B parameters. We train separate SAEs for Qwen3-14B (14.8B parameters, Top-25 features per side) and Gemma-3-12B-IT (12.0B parameters, Top-10 features per side) and rediscover the CALL and NO_CALL features on the full 3,652-example When2Call test set. Qwen3-14B is a Qwen3 checkpoint, not Qwen3.5, and neither model is evaluated with AMCS. Margins are divided by the within-model standard deviation without centering, so $m = 0$ remains activation parity.

Both models have positive slopes, parity probabilities above 0.5, and negative neutral boundaries (Table 12). For Gemma-3-12B-IT, 992 examples lie at $m < 0$ and 1,325 at $m > 0$, so parity lies inside the observed margin range. Each SAE is trained independently, so the normalized margins are not a shared coordinate, and these two models do not establish a relation between parameter count and offset size.

Table 11: Margin-construction ablation at Top-25 on the original SAE. Standardized divides each feature by its evaluation-set standard deviation and does not subtract the mean. Difference/sum divides the difference of the two side means by their sum.

Model	Margin	Features	β	$\Pr(\hat{d} = 1 \mid m = 0)$	$m^\star$
Qwen3.5-4B	Raw	25/25	4.008	0.894	-0.533
	Standardized	25/25	3.019	0.782	-0.424
	Difference/sum	25/25	8.569	0.868	-0.220
Qwen3.5-9B	Raw	25/25	1.338	0.996	-4.177
	Standardized	25/25	3.368	0.959	-0.936
	Difference/sum	25/25	10.285	0.990	-0.450
Gemma-3-1B	Raw	19/25	0.009	0.991	-496.909
	Standardized	19/25	1.374	0.994	-3.708
	Difference/sum	19/25	2.453	0.988	-1.798
Gemma-3-4B	Raw	25/25	0.008	0.860	-241.619
	Standardized	25/25	3.437	0.946	-0.833
	Difference/sum	25/25	6.114	0.958	-0.510
Ministral-3-3B	Raw	25/25	9.403	0.927	-0.270
	Standardized	25/25	4.853	0.921	-0.507
	Difference/sum	25/25	9.741	0.924	-0.256
Ministral-3-8B	Raw	25/25	3.988	0.840	-0.415
	Standardized	25/25	4.135	0.714	-0.221
	Difference/sum	25/25	8.216	0.809	-0.176

Table 12: IBH on two models above 10B. n counts responses judged as CALL or NO_CALL. $m^\star$ is in within-model standard-deviation units.

Model	n (CALL/NO_CALL)	$\Pr(\hat{d} = 1 \mid m = 0)$	$m^\star$
Qwen3-14B	2,297 (2,060/237)	0.996	-0.856
Gemma-3-12B-IT	2,317 (1,976/341)	0.998	-1.615

H.2 Pretrained Checkpoint

We compare Qwen3.5-4B with the officially released pretrained-only Qwen3.5-4B-Base. The base weights stay frozen, and a separate SAE is trained on it only as a measurement model, with the same nominal layer, dictionary size, sparsity, feature discovery, prompts, judge, and tool-aware first-turn format as Qwen3.5-4B. The two SAE bases differ, so we compare the direction of the IBH criteria rather than the size of the offsets.

Table 13: IBH on Qwen3.5-4B-Base and the instruction-tuned Qwen3.5-4B. n counts responses judged as CALL or NO_CALL. Intervals are 95% confidence intervals.

Checkpoint	n (CALL/NO_CALL)	$\Pr(\hat{d} = 1 \mid m = 0)$	$m^\star$
Qwen3.5-4B-Base	2,345 (1,947/398)	0.9621 [0.9509, 0.9730]	-0.691 [-0.728, -0.651]
Qwen3.5-4B	2,308 (1,889/419)	0.8943 [0.8746, 0.9149]	-0.533 [-0.588, -0.487]

Both slopes are positive. For the base checkpoint, the full interval for the parity probability lies above 0.5 and the full interval for $m^\star$ lies below zero (Table 13), so the offset is already detectable before the publicly documented post-training stage.

Behaviorally, on the same 3,652 prompts, post-training lowers the overall call rate from 65.85% to 59.23% (-6.63 points) and the false-call rate on ground-truth NO_CALL prompts from 64.22% to 60.36% (-3.86 points, $p = 0.0051$). The correct NO_CALL rate rises only from 35.03% to 36.25% ($+1.22$ points, $p = 0.383$). The p-values are paired exact tests on the same prompts. The two rates are not complements, because the model may also answer directly or state that it cannot answer.

Post-training thus reduces the propensity to call but does not create the offset from scratch, and the comparison does not identify which training stage produces it.

H.3 Reasoning On and Off

We compare Qwen3.5-4B with reasoning off and on over 2,357 ground-truth CALL/NO_CALL examples, using the frozen Stage-2 SAE at Layer 25, Top-25 features, greedy decoding, and the same judge. The judge sees only the final answer after `</think>`. The residual state is read after `</think>` for reasoning-on, and at the action boundary after an empty reasoning block for reasoning-off. The reasoning-off responses were regenerated for this comparison. Their IBH estimates match Section 4, while their behavioral accuracies differ slightly from Table 1.

Table 14: Qwen3.5-4B with reasoning off and on. Call and no-call columns are accuracies (%).

Setting	Call	No-call	$\Pr(\hat{d} = 1 \mid m = 0)$	$m^\star$
Reasoning-off	96.37	36.25	0.894	-0.533
Reasoning-on	95.52	59.32	0.934	-0.573
On $-$ off	-0.85	$+23.07$	$+0.0395$	-0.040

Reasoning raises no-call accuracy by 23.07 points with call accuracy nearly unchanged, but the call preference at parity remains (Table 14). Under 2,000 source-cluster paired bootstrap resamples, the parity probability rises by $+0.0395$ (95% interval $[+0.0127, +0.0643]$). The change in $m^\star$, -0.040 with interval $[-0.109, +0.037]$, crosses zero, so we do not claim that reasoning moves the neutral boundary. The mean margin barely moves on ground-truth CALL examples ($+0.734$ to $+0.726$) but falls from -0.165 to -0.584 on ground-truth NO_CALL examples. The no-call mean thus moves from the call side of the off-condition boundary (-0.533) to the no-call side of the on-condition boundary (-0.573). Reasoning strengthens instance-level evidence for missing information without removing the call bias at activation parity.

I LLM Usage

This work studies LLMs as target models for tool-use decisions. We also use an independent LLM judge to classify generated responses into CALL and NO_CALL behavior labels for feature discovery, calibration, and evaluation. The judge prompt is reported in Appendix B. The same judging procedure is used for baseline and steered conditions. A separate judge, Qwen3.6-27B with deterministic decoding, scores function names and arguments for the preserved-call analysis in Appendix G.2. It is not the four-way behavior judge above. AMCS itself is defined by SAE activations, a fitted margin model, and a fixed hook, rather than by an LLM component.

J Perplexity of Steered Outputs

Note that models use different prompt templates and produce responses of different types (TOOL_CALL vs. NO_CALL), so perplexity values are only meaningful within a single model compared against its own baseline, not across models or response types. For Qwen3.5 and Ministral3, all three methods maintain perplexity within a small margin of the unsteered baseline across all tested α values, confirming that residual-stream additions do not destabilize generation. Gemma3 is the exception: Suppress at higher α produces extreme values (3.19×10^7 for Gemma3-1B at $\alpha = 0.6$, 57317.43 for Gemma3-4B at $\alpha = 1.0$), and AMCS at $\alpha = 1.0$ also degrades for the 1B model (392.70). This instability mirrors the weak CALL/NO_CALL feature separation already noted for Gemma3 in §5.2, where Gemma-3-4B is a null case and Gemma-3-1B is a failure case for AMCS.

K Broader Impacts

This work can improve the reliability and efficiency of tool-using agents by reducing unnecessary tool calls, especially in cases where the model should ask for missing information instead of calling a tool. Such behavior can lower API cost, reduce avoidable external actions, and make agent decisions easier to audit through feature-level diagnostics.

Table 15: Next-token perplexity of steered outputs on the When2Call test split. Baseline is the unmodified model. Suppress and Promote are one-sided ablations using only the call-suppression or no-call-promotion component of the AMCS vector, respectively. AMCS is the full two-sided intervention. The parameter $\alpha \in \{0.2, 0.6, 1.0\}$ controls the allocation between the two sides. Extreme values for Gemma3 reflect the weak CALL/NO_CALL feature separation noted in §5.2.

Method	α	Qwen3.5-4B	Qwen3.5-9B	Gemma3-1B	Gemma3-4B	Ministral3-3B	Ministral3-8B
Baseline	–	1.32	1.37	1.04	1.02	3.17	2.93
Suppress	0.2	1.33	1.37	1.25	1.06	3.12	2.85
	0.6	1.36	1.39	3.19×10^7	1.33	2.93	2.58
	1.0	1.45	1.42	225.02	57317.43	2.12	1.63
Promote	0.2	1.34	1.36	1.13	1.03	3.14	2.90
	0.6	1.37	1.35	2.38	1.20	3.09	2.84
	1.0	1.43	1.37	7.44	1.54	3.02	2.80
AMCS	0.2	1.36	1.38	2.29	1.02	3.08	2.90
	0.6	1.36	1.42	1.53	1.02	3.05	2.84
	1.0	1.38	1.46	392.70	1.02	3.01	2.80

At the same time, steering a model's tool-use propensity can introduce deployment risks if applied without task-specific validation. Overly conservative steering may suppress necessary tool calls, while overly aggressive steering may amplify automation errors. We therefore view AMCS as a diagnostic and mitigation tool that should be evaluated with call accuracy, no-call accuracy, and downstream task checks before deployment.

L Experiments Compute Resources

All experiments were conducted on NVIDIA A800 (80GB) GPUs. For individual experiments, SAE training on a single model requires approximately 4 GPU-hours, while WHEN2CALL evaluation on a single model takes approximately 3 GPU-hours. Experiments involving activation steering require approximately 4 GPU-hours per model. The main results in Table 1 encompass 98 WHEN2CALL evaluation runs, amounting to roughly $4 \times 98 = 392$ GPU-hours in total, which were completed in approximately 2 days using 8 GPUs in parallel. In aggregate, all experiments reported in this paper were conducted over a period of two months with dedicated access to 8 GPUs.

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 over-calling phenomenon, IBH diagnosis, and AMCS intervention, and the claims are scoped to the evaluated tool-use setting.

   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: Appendix A states that the pretrained-checkpoint comparison does not identify which training stage produces the offset, that AMCS is an inference-time correction relying on a local linear approximation, and that experiments cover only single-turn decisions. Sections 5.2 and 6 further state that AMCS is model-conditional, with Gemma-3-4B as a null case and Gemma-3-1B as a failure case, and that robustness is claimed only for the tested SAE and margin choices.

   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: [Yes]

   Justification: The paper states the assumptions behind the margin model and AMCS derivation and provides the relevant formulas and implementation details.

   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: The main text and appendix specify the datasets, model set, SAE training protocol, feature ranking, probing, calibration, and steering procedure needed to reproduce the results.

   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: The code repository linked in the abstract (https://github.com/SKURA502/agent-sae) provides code, data-processing instructions, and experiment scripts for reproducing the reported discovery, diagnosis, and steering results.

   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: The experimental sections and appendix report the data splits, model and hook choices, SAE hyperparameters, ranking criteria, calibration protocol, and evaluation settings.

   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: We report percentile 95% intervals from 1,000 example-level bootstrap resamples for the ACEBench diagnosis (Appendix G.1), 95% confidence intervals and paired exact tests for the pretrained-checkpoint comparison (Appendix H.2), and 95% intervals from 2,000 source-cluster paired bootstrap resamples for the reasoning comparison (Appendix H.3). The When2Call steering accuracies in Table 1 come from deterministic greedy decoding with a single judge run, so they are point estimates without error bars.

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: The appendix reports the compute environment and runtime requirements for SAE training, feature discovery, calibration, and inference-time steering.

   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 work uses public benchmarks and open models for diagnostic and mitigation experiments and does not involve prohibited data collection or deployment.

   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: The paper discusses positive impacts from reducing unnecessary tool calls and potential risks from steering interventions in deployed agent systems.

    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: The paper does not release high-risk pretrained models, scraped datasets, or other assets that require special controlled-access safeguards.

    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: [Yes]

Justification: All existing assets are cited and used under their respective licenses: the When2Call benchmark (CC-BY 4.0), the ACEBench benchmark (MIT), the OpenWebText2 corpus released by EleutherAI (MIT), the Qwen3.5, Qwen3.5-4B-Base, Qwen3-14B, and Qwen3.6-27B model weights (Apache 2.0), the Ministral-3-Instruct model weights (Apache 2.0), and the Gemma-3-it model weights, including Gemma-3-12B-IT (Gemma Terms of Use). The TopK SAE and CAA methods are cited.

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: Newly produced code, feature artifacts, and calibration outputs are documented with the release materials and linked to the corresponding experimental protocols.

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: [N/A]

Justification: The work does not involve crowdsourcing experiments or research with human subjects.

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: The work does not involve human subjects, so IRB or equivalent review is not applicable.

    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: Appendix I describes all LLM usage: LLMs are the target models, an LLM judge assigns behavior labels, and a separate Qwen3.6-27B judge scores function names and arguments for the preserved-call analysis. AMCS itself contains no LLM component.

    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.