Controlling Tool Use with Heading-Specific Activation Steering

Yuqi Chen [1] **Vincent Siu** [1] **Yang Liu** [1] **Dawn Song** [2] **Chenguang Wang** [1]

Abstract

Tool-augmented large language models extend their capabilities beyond parametric knowledge through external tools, but tend to invoke them unnecessarily. We investigate whether tool-use decisions have any stable internal representation that can be extracted and manipulated, a question that is non-trivial given that tools exist entirely in context at inference time and have no direct encoding in model weights. We show that steering vectors extracted from heading-anchors positions exert bidirectional causal control over tool-invocation behavior across five open-source models and three domains, suppressing unnecessary tool use most effectively in domains where parametric reasoning suffices. However, geometric analysis reveals that this causal effectiveness does not correspond to clean linear structure: tool-invocation steps exhibit diffuse, bimodal alignment with the suppression vector rather than the consistent negative alignment a linear encoding account would predict, and different tool types recruit largely distinct internal signatures with low cross-tool feature overlap. We hypothesize these geometric properties are indicative of the non-parametric nature of tools, and distinguish tool-use steering vectors from those extracted for parametrically grounded concepts. The relationship between this geometric irregularity and the observed causal effectiveness remains an open question.

1. Introduction

Tool-augmented LLMs have become a dominant paradigm for complex reasoning and real-world tasks (Yao et al., 2023; Levy et al., 2024; Schick et al., 2023). External tools such as web search, code execution, and user-interaction modules extend model capabilities beyond parametric knowledge, improving performance on tasks requiring up-to-date information, precise computation, or clarification of underspecified intent (Schick et al., 2023; Schneider, 2025).

However, tool augmentation also introduces a persistent failure mode: *tool overuse*. Models may invoke tools even when internal reasoning would suffice, repeat calls without meaningful progress, or rely on tools in ways that increase latency and cost while providing limited benefit (Qian et al., 2025; Shen et al., 2024). Unnecessary tool calls can expose the system to noisy retrieval, execution failures, and avoidable error propagation (Wang et al., 2025). Existing mitigations either require expensive retraining (Qian et al., 2025; Shen et al., 2024) or operate at the output interface rather than on the internal activations underlying tool selection (Wang et al., 2025), leaving the latent decision to invoke a tool unaddressed.

We ask whether tool-use decisions have any stable internal representation that can be extracted and manipulated. This is non-obvious: unlike concepts such as sentiment or factuality that are encoded in model weights through training, tools exist entirely in context at inference time, and standard assumptions from the linear representation hypothesis (Park et al., 2024) and representation engineering (Zou et al., 2023) do not straightforwardly apply to concepts with no parametric grounding. Prior steering work has targeted concepts with clear parametric grounding (sentiment, refusal, truthfulness) (Panickssery et al., 2023; Siu et al., 2025a); we are among the first to apply it to a behavior that is explicitly non-parametric by construction. Following contrastive activation addition (Turner et al., 2023; Panickssery et al., 2023), we extract hidden states at structured tool-heading positions in reasoning trajectories to build tool-specific steering vectors, and evaluate their causal and geometric properties across five open-source instruction-tuned models and three domains.

Applying these vectors via activation addition suppresses tool use substantially below baseline, while orthogonalizing hidden states with respect to them amplifies tool use above it. This bidirectional behavioral control, shown in Figure 1, confirms that the extracted vectors are causally relevant to tool-use decisions despite the absence of any stable parametric encoding of tools in model weights. Suppression is most

[1]Department of Computer Science and Engineering, UC Santa Cruz [2]Department of Computer Science, UC Berkeley. Correspondence to: Chenguang Wang <chenguangwang@ucsc.edu>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

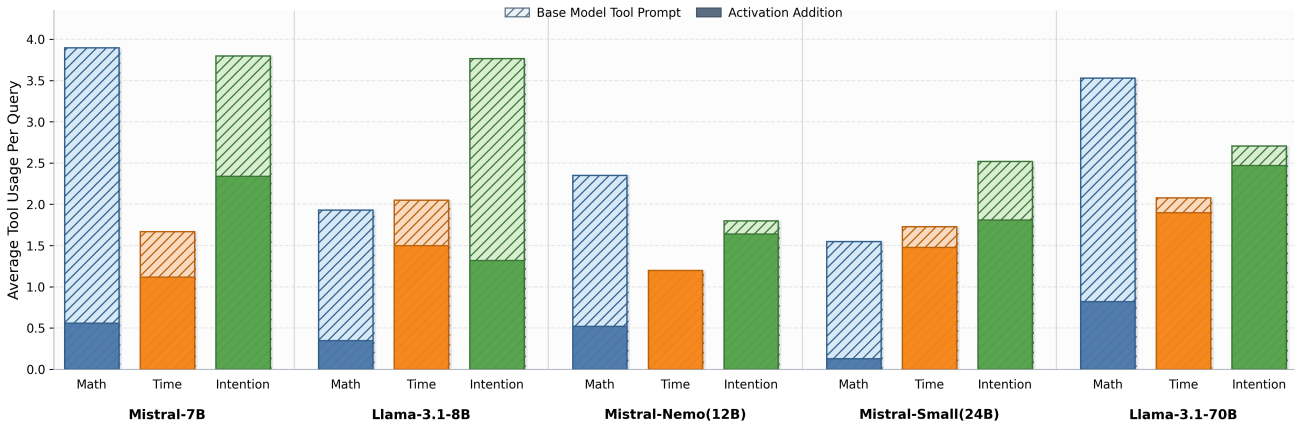

Figure 1. Average tool calls per query across five models under three conditions. Base Model Tool Prompt is the prompt-only baseline without hidden-state intervention. Activation Addition subtracts the steering vector from the residual stream, suppressing tool use below the baseline. Orthogonalization projects hidden states to be orthogonal to the steering vector, removing the suppressive component and amplifying tool use above the baseline. The opposing directions of the two interventions confirm that the extracted vector encodes a causally effective, bidirectional tool-use direction in the residual stream.

effective in domains where parametric reasoning suffices, and degrades where external retrieval or user clarification is genuinely required, with larger models showing greater resistance across all domains.

However, geometric analysis reveals that causal effectiveness does not correspond to clean linear structure. Tool-invocation steps exhibit a diffuse bimodal cosine similarity distribution with respect to the suppression vector rather than the consistent negative alignment a linear encoding account would predict, and different tool types recruit largely distinct internal feature dimensions with low cross-tool overlap. These properties are consistent with the non-parametric nature of tools and distinguish tool-use steering vectors from those extracted for parametrically grounded concepts. The relationship between this geometric irregularity and the observed causal effectiveness remains an open question, and we discuss it in the context of the multi-turn dependency structure of tool-augmented trajectories that limits direct reconciliation with the probing analysis.

2. Related Work

Representation Theory. The linear representation hypothesis (Park et al., 2024) predicts that concepts encoded in model weights should be linearly decodable from activations, with supporting evidence from word vector arithmetic (Mikolov et al., 2013) and superposition theory (Elhage et al., 2022). Empirical work on refusal (Arditi et al., 2024) and truthfulness (Burns et al., 2023) has found steering vectors with clean geometric properties consistent with this account. More recent work has complicated the picture: concept directions can span multi-dimensional subspaces rather than single directions (Wollschläger et al., 2025; Siu et al., 2025c), and their geometry can exhibit complex dependencies on context and layer depth. Our work extends this line of inquiry to a setting where linear representability is not expected to hold by construction: because tools are injected through context rather than encoded in weights, the geometric irregularity we observe is a predicted consequence of the non-parametric nature of tools rather than an anomaly, and it distinguishes tool-use steering vectors from those studied in prior representation theory work.

Activation Steering. Steering methods identify directions in activation space corresponding to target behaviors and modulate them via vector addition or orthogonalization (Zou et al., 2023; Arditi et al., 2024; Turner et al., 2023; Panickssery et al., 2023; Siu et al., 2025a). Directions are commonly extracted from contrastive data pairs (Burns et al., 2023; Arditi et al., 2024; Zou et al., 2023) and have been used for both behavior elicitation and concept removal (Ravfogel et al., 2020; Siu et al., 2025b). Inference-time steering uses probes or classifiers (Li et al., 2023; Lee et al., 2024) to conditionally apply interventions during forward passes. Prior steering work has largely assumed that the concept being steered has stable parametric grounding; our work applies these methods to a concept that explicitly lacks it, and examines the consequences for both causal effectiveness and geometric structure.

LLM Tool Use. Tool-use methods extend language models with external functions, APIs, and environments, enabling access to up-to-date information, specialized computation, and domain-specific expertise beyond parametric memory (Schick et al., 2023; Yao et al., 2023; Qin et al., 2023; Qu et al., 2025; Shen, 2024). Prior work studies how models decide *when* to call tools, *which* tools to select, and *how* to

incorporate tool outputs into subsequent reasoning (Qian et al., 2024d). Recent work further improves tool-use robustness through tool creation, external module integration, and alignment for more efficient tool calling under uncertainty or knowledge-boundary awareness (Qian et al., 2024a; Liu et al., 2025; Qian et al., 2024c; Xu et al., 2025). At the same time, dedicated evaluations show that models still struggle with unnecessary or incorrect tool calls, motivating benchmarks for deciding whether tools are needed and which tools should be used (Huang et al., 2024; Ning et al., 2024). Closely related to these efforts, recent work studies tool overuse and adaptive tool calling based on model uncertainty or self-awareness (Wang et al., 2025; Shen et al., 2024; Qian et al., 2025). In contrast to methods that explicitly optimize external tool interaction policies, our work focuses on inducing the desired behavior through inference-time manipulation of internal representations.

3. Methodology

We study inference-time suppression of tool calls in structured tool-augmented generation. In our setting, the model emits intermediate reasoning using explicit section Heading-Anchors such as `### Reasoning`, `### Search`, `### Code`, `### AskUser`, and `### Final Response`. Rather than steering the model globally throughout generation, we intervene locally at heading-formation steps, where the decoder is about to select the next section label. This yields a lightweight test-time intervention that requires no parameter updates.

During one inference run, we fix a target steering payload and a predefined intervention layer $l^\star$. The method then has three parts: (i) construct heading-anchored steering vectors from complete trajectories, (ii) detect imminent Heading-Anchors formation `###`, and (iii) rerun only the triggered decoding step with a local hook-based suppression at layer $l^\star$.

3.1. Heading-Anchored Steering Vectors

For each tool type $k \in \{\texttt{Code}, \texttt{Search}, \texttt{AskUser}\}$, we collect a set of complete labeled multi-turn trajectories with Heading-Anchors:

$$\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N_k},$$

where x_i is the prompt and y_i is the full structured response containing Heading-Anchors, tool calls and the corresponding tool outputs. We run the model on the full prompt-response sequence and extract the hidden states from layer $l^\star$. As a result, each Heading-Anchor representation is computed in the full left-to-right context induced by the preceding reasoning and tool interactions.

We represent each heading occurrence by the hidden state at the token corresponding to its leading `###` anchor marker. Let R_i denote the set of anchor positions for all occurrences of `### Reasoning` in trajectory y_i, and let T_i^k denote the set of anchor positions for all occurrences of the target heading k. For each layer l, we first average Heading-Anchor states within each trajectory:

$$\bar{h}_{R,i}^{(l)} = \frac{1}{|R_i|} \sum_{t \in R_i} h_{i,t}^{(l)}, \qquad \bar{h}_{k,i}^{(l)} = \frac{1}{|T_i^k|} \sum_{t \in T_i^k} h_{i,t}^{(l)} \tag{1}$$

We then form a sample-level steering delta

$$\Delta_{k,i}^{(l)} = \bar{h}_{R,i}^{(l)} - \bar{h}_{k,i}^{(l)}, \tag{2}$$

And average these deltas across multiple trajectories:

$$v_k^{(l)} = \frac{1}{N_k} \sum_{i=1}^{N_k} \Delta_{k,i}^{(l)}. \tag{3}$$

Under this definition, $v_k^{(l)}$ points from the `### Reasoning` state to the corresponding target tool-heading state. Adding $v_k^{(l^\star)}$ therefore acts as a suppressive intervention, shifting the hidden state away from the tool-heading representation.

This construction is intentionally performed at the sample level rather than by pooling all heading occurrences across the dataset before taking a single global difference of means. We do so because trajectories can contain different numbers of `### Reasoning` and tool Heading-Anchors. A pooled estimator would therefore overweight trajectories with many anchor occurrences, whereas our formulation gives each trajectory a more comparable contribution to the final steering vector.

3.2. Activation Addition and Orthogonalization

We then utilize the obtained steering vector $v_k^{(l)}$ in two different ways. At a specific layer $l*$ on the Heading-Anchor tokens `###`, we perform activation addition (Turner et al., 2023; Panickssery et al., 2023; Siu et al., 2025a), altering the layer input $h_t^{(l^\star)}$ as following to suppress tool use with steering coefficient α. In our experiments we set $\alpha = 1$.

$$\tilde{h}_t^{(l^\star)} = h_t^{(l^\star)} + \alpha v_k^{(l^\star)}. \tag{4}$$

We also test orthogonalizing the hidden states to the vector (Arditi et al., 2024; Siu et al., 2025b) on this Heading-Anchor tokens, to study steering to encourage tool use by removing the suppression vector.

$$\tilde{h}_t^{(l^\star)} = h_t^{(l^\star)} - \mathrm{proj}_{v_k^{(l^\star)}}\left(h_t^{(l^\star)}\right), \quad \mathrm{proj}_v(h) = \frac{\langle h, v \rangle}{\|v\|_2^2} v \tag{5}$$

4. Experimental Setup

Tasks and Domains. We evaluate on the Math, Time, and Intention tasks from the SMART benchmark (Qian et al., 2025). These three domains expose complementary tool-use regimes: Math contains many cases where parametric reasoning suffices without external computation, Time requires access to up-to-date information unavailable in model weights, and Intention requires recovering missing user-specific details through clarification. Together they test whether a suppressive intervention can reduce unnecessary tool calls without blocking those that are genuinely required.

Models and Baselines. We evaluate five open-source instruction-tuned models: Mistral-7B, Llama-3.1-8B, Mistral-Nemo (12B), Mistral-Small (24B), and Llama-3.1-70B (Jiang et al., 2023; Grattafiori et al., 2024). We compare against the base model with tool prompt from Qian et al. (2025), which provides the model with all available tools and their descriptions and allows it to decide independently whether and when to invoke them, following an iterative reasoning-then-execution loop until a final answer is produced.

Prompts. Since the method described in **Method** relies on heading-anchors, we use different prompts compared to Qian et al. (2025). For each domain, we use the prompt templates as shown in A.1. Specifically,we instruct the model to format each turn with an explicit heading anchor, such that every response must begin with `###` followed by one of the predefined action types: `Reasoning, Code, Search, AskUser, Final Response`. In this form, it standardizes the intermediate structure of model outputs across domains, making the interaction trace easier to parse and analyze.

Steering Vector Construction. To construct steering vectors, we sample data from the training split of SMART benchmark (Qian et al., 2025). Specifically, we select 20 examples for each of the three domains used in our experiments: **Math**, **Time**, and **Intention**, yielding 60 examples in total. These examples are used only to generate structured trajectories for steering-vector extraction. Using the prompts above, we generate structured responses for the selected SMART training examples with **GPT-4o** (OpenAI, 2023). These generated trajectories serve as the data for computing steering vectors described in Section 3.1. In particular, the explicit heading structure enables us to identify the hidden states corresponding to `### Reasoning` and the target tool heading-anchors, from which we compute domain-relevant steering directions.

Steering Configuration. As described in Section 3.1, we construct a separate steering vector for each tool type. In the main experiments, we restrict steering to the primary tool associated with each domain: **Code** for **Math**, **Search** for **Time**, and **AskUser** for **Intention**.

Evaluation Metrics. Following (Qian et al., 2025), for Math and Time we report task accuracy (**ACC**) and average tool calls per example (**ToolAvgUse**). For Intention we additionally report **Missing Details Recovery** and **Summarized Intention** from the IN3 dataset (Qian et al., 2024b), alongside the overall task score and **ToolAvgUse**.

5. Steering Results

Steering Layer Selection We sweep the steering vector across all layers, performing activation addition at each layer with coefficient $\alpha = 1$ to minimize tool usage, and select the elbow of the resulting tool-usage curve (Figure 2). Mistral-7B uses layer 18 with 0.95 average tool calls per prompt, Llama-3.1-8B uses layer 16 with 0.70, Mistral-Nemo uses layer 23 with 0.82, Mistral-Small uses layer 19 with 0.63, and Llama-3.1-70B uses layer 45 with 1.28. Despite differing in absolute depth, all five elbow points fall between normalized depths 0.45–0.60, suggesting that tool-use representations stabilize at a consistent relative depth across tested models. The steering layer is calibrated once per model and held fixed for all subsequent experiments.

Tool Use Steering Results Table 1 presents the main experimental results across the three domains, while Figure 1 visualizes the corresponding changes in average tool usage. The central pattern is consistent across all five models: activation addition strongly suppresses tool invocation, whereas orthogonalization increases it above the base-model level. This bidirectional effect is itself non-trivial: tool use is not a clean parametrically encoded skill in model weights, with limited theoretical underpinnings on linear representability. The fact that addition and orthogonalization move behavior in opposite directions suggests that the extracted vector captures a relevant component of the model's tool-use decision.

Table 1 also shows that reduced tool use does not translate into uniform task outcomes across domains. In **Math**, activation addition drives tool usage close to zero for all models (e.g., from 3.90 to 0.56 for Mistral-7B, from 1.93 to 0.35 for Llama-3.1-8B, and from 1.55 to 0.13 for Mistral-Small), while accuracy drops only modestly relative to the base model in most cases. This suggests that many tool calls in Math are indeed redundant, and that parametric reasoning can often replace external computation. By contrast, in **Time** and **Intention**, suppressing tool use is consistently accompanied by larger performance degradation. For example, on Time, Llama-3.1-8B drops from 56.0 to 45.0 ACC as tool usage falls from 2.05 to 1.50, and Mistral-Small drops from 62.0 to 49.0 as tool usage falls from 1.73 to 1.48. On Intention, activation addition substantially reduces Missing Details Recovery and Summarized Intention scores across all models, indicating that many of these tool calls are

Method	Model	Math		Time		Intention		
		ACC	ToolAvgUse	ACC	ToolAvgUse	Missing Details Recovery	Summarized Intention	ToolAvgUse
Base Model Tool Prompt	Mistral-7B	**13.3**	3.90	**49.0**	1.67	**48.84 / 21.70**	**63.0**	3.80
	Llama-3.1-8B	**51.0**	1.93	**56.0**	2.05	**54.76 / 25.90**	**70.2**	3.77
	Mistral-Nemo(12B)	**46.0**	2.35	**59.0**	**1.19**	**31.35 / 5.82**	59.3	1.80
	Mistral-Small(24B)	**76.0**	1.55	**62.0**	1.73	**45.74 / 33.62**	**78.2**	2.52
	Llama-3.1-70B	**67.5**	3.53	**63.0**	2.08	**45.74 / 35.96**	**61.7**	2.71
Orthogonalization	Mistral-7B	11.8	4.82	40.0	2.92	13.92 / 3.18	58.0	4.82
	Llama-3.1-8B	48.8	2.25	44.0	3.87	15.48 / 3.55	59.3	4.20
	Mistral-Nemo(12B)	39.8	2.66	45.0	1.87	24.86 / 5.12	56.1	3.05
	Mistral-Small(24B)	71.0	2.21	51.0	3.20	28.94 / 5.63	66.2	3.62
	Llama-3.1-70B	60.5	4.31	48.0	2.99	31.42 / 6.21	56.0	3.59
Activation Addition	Mistral-7B	11.8	**0.56**	39.0	**1.12**	14.3 / 4.67	57.6	**2.34**
	Llama-3.1-8B	48.3	**0.35**	45.0	**1.50**	16.28 / 6.38	55.2	**1.32**
	Mistral-Nemo(12B)	43.5	**0.52**	47.0	1.20	27.36 / 5.58	**60.8**	**1.64**
	Mistral-Small(24B)	65.4	**0.13**	49.0	**1.48**	31.05 / 5.94	69.4	**1.81**
	Llama-3.1-70B	63.2	**0.82**	49.0	**1.90**	33.21 / 7.68	55.9	**2.47**

Table 1. Performance comparison across three methods and three domains: **Math**, **Time**, and **Intention**. For **Math** and **Time**, we report task accuracy (ACC) and average tool usage (ToolAvgUse). For **Intention**, we report Missing Details Recovery, Summarized Intention, and average tool usage. Results are grouped by model to enable fair comparison across methods. The best value is marked in **bold**.

not redundant but necessary for recovering underspecified user information. This finding is relatively intuitive: the questions in Time and Intention are by construction unanswerable from parametric knowledge alone, so suppressing tool calls intuitively leads to larger performance drops.

Figure 1 makes this pattern visually clear: activation addition suppresses tool use across all models and domains, while orthogonalization shifts behavior in the opposite direction. This domain asymmetry is strongest in Math, where tool calls are nearly eliminated, suggesting that many invocations are redundant. By contrast, Time and Intention remain more resistant to suppression, consistent with these domains relying more heavily on retrieval or clarification that cannot be replaced by parametric knowledge alone.

Orthogonalization provides the complementary intervention. As shown in Table 1 and Figure 3, projecting hidden states to be orthogonal to the suppression direction increases tool usage above the base model across all models and domains. However, this additional tool use does not improve overall task performance; in most settings it instead lowers accuracy or intention quality relative to the base model. This indicates that the base model is not simply under-using tools. Rather, it already reflects a partial internal balance between tool invocation and direct answering, and removing the suppressive component pushes the model toward overuse. This suggests that steering tool use may entangle with parametric knowledge due to the non-parametric nature of tools, a phenomenon also observed in prior representation-steering work (Siu et al., 2025a). Taken together, the results suggest that the extracted direction primarily controls *whether* the model invokes tools, but the utility of suppressing that behavior depends strongly on domain: it is comparatively safe in Math, but much more harmful in Time and Intention, where tool access is often genuinely required.

Robustness Beyond the Literal Heading Token. Because the intervention is anchored on the formatted heading `### Code`, a natural concern is whether it controls tool invocation or merely suppresses the literal surface string. We probe this with two controls on Llama-3.1-8B in the Math domain (full results in Appendix A.2). First, under *cross-format transfer*, we extract the `Code` vector under the Markdown heading schema but evaluate it under a JSON action schema

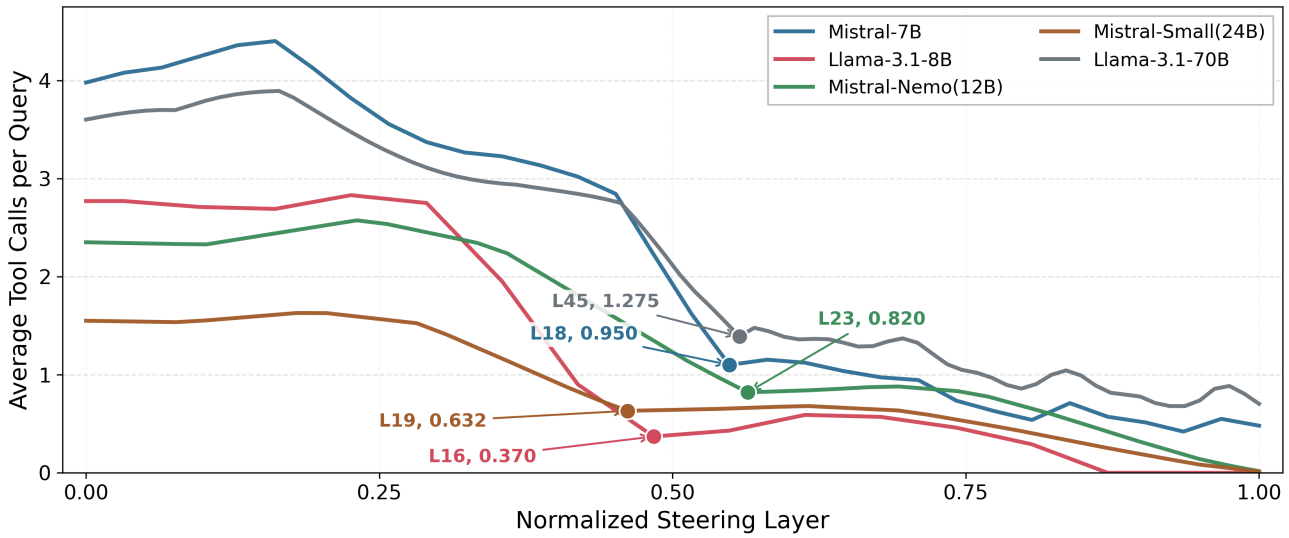

Figure 2. Average tool calls per query as a function of normalized steering layer depth across five models. Tool usage drops sharply as the steering layer moves into the middle of the network before plateauing, forming a clear elbow. Annotated points mark the selected elbow layer for each model; despite differing in absolute depth, all five fall within normalized depths 0.45–0.60, suggesting that tool-use representations stabilize at a surprisingly consistent relative depth across architectures.

(e.g. {`"action": "code", ...`}) in which the literal `### Code` heading never appears; the Markdown-extracted vector still removes 63.9% of tool calls with accuracy essentially unchanged (0.330 → 0.325). Second, under *heading rename*, we keep the vector extracted from `### Code` but evaluate with renamed tool headings of the same semantics; suppression persists even when the active heading contains no occurrence of the word `Code` (e.g. 70.3% relative suppression under `### Action_B`). Both controls indicate that the effect is not solely the suppression of the literal heading token. It is nonetheless schema-sensitive: reverse transfer (JSON→Markdown) is much weaker (15.5%), and JSON→JSON reduces tool use but collapses accuracy through malformed generations. We therefore characterize the extracted direction as a schema- and anchor-sensitive tool-use control direction with partial transfer beyond literal heading tokens, rather than a fully format-independent representation. We note these are preliminary single-model, single-domain controls.

6. Geometric Analysis of the Steering Vector

6.1. Probing Expression of the Steering Vectors

To understand what the steering vector captures geometrically, we measure the cosine similarity between the extracted vector and the hidden states of unsteered prompts at both reasoning steps and tool-invocation steps. If the vector encoded a stable tool-suppression direction, tool-step hidden states should align negatively with it while reasoning-step hidden states should not. Importantly, these distributions are characterized on unsteered multi-turn trajectories, where each step is conditioned on prior tool interactions; the geometric relationship between the suppression vector and the hidden state distribution under orthogonalization in Figure 3 is therefore not directly comparable, as orthogonalization intervenes on a sequential process in which each modified step conditions all subsequent ones.

Figure 4 plots these distributions across all five models. Reasoning steps concentrate sharply at zero, indicating near-orthogonality to the steering vector across all architectures. Tool steps exhibit a generally negative alignment, but their distributions are often diffuse and bimodal, with substantial mass in both acute and oblique alignment with the steering vector. This is seemingly consistent with the non-parametric nature of tools: because tool definitions exist only in context at inference time and have no direct encoding in model weights, tool-step activations have no a priori reason to cluster along a consistent direction in the residual stream. The extracted vector therefore does not correspond to a stable concept direction in the sense assumed by prior representation engineering work (Zou et al., 2023; Panickssery et al., 2023), but rather captures something about the conditions under which the model defers to context. We hypothesize that steering effectiveness persists despite this geometric irregularity because the heading-anchor format concentrates the tool-invocation decision into a single token: intervening at this structural bottleneck may be sufficient for behavioral control even when the underlying concept has no consistent direction in the residual stream.

In Llama-3.1-70B this geometric irregularity is most

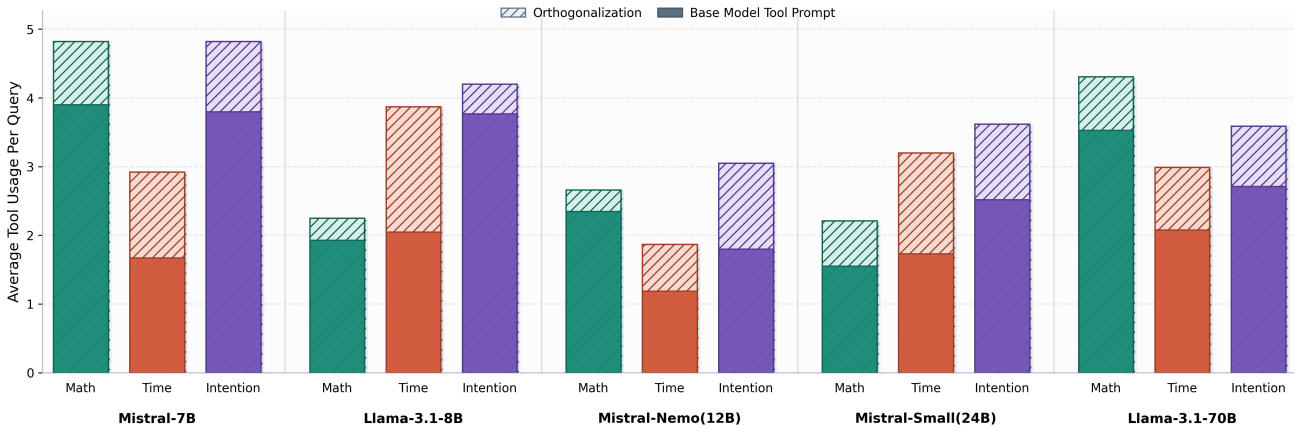

Figure 3. Average tool calls per query under orthogonalization and the base model prompt across five models and three domains. Orthogonalization projects hidden states to be orthogonal to the suppression vector at tool-heading positions, removing its suppressive component. Tool usage increases above the base model level in all cases, suggesting that the suppression vector is active in the unsteered baseline and that its removal releases tool-invocation behavior.

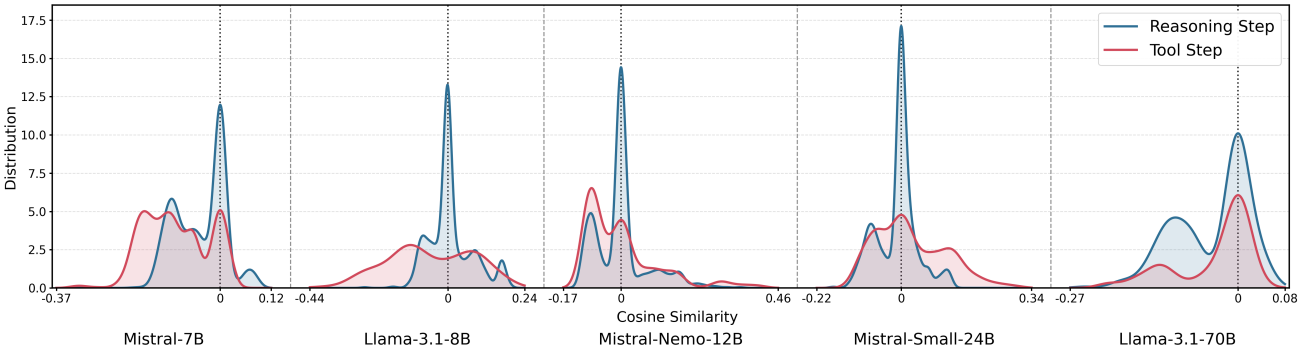

Figure 4. Cosine similarity between the steering vector and hidden states at reasoning steps (blue) and tool-invocation steps (red) on unsteered prompts, across five models. Reasoning steps concentrate sharply at zero, indicating near-orthogonality to the steering vector. Tool steps exhibit a diffuse bimodal distribution with mass on both the acute and oblique sides, inconsistent with the steering vector acting as a simple linear encoding of tool-use intent. This pattern is consistent across all five tested models. However, we note that due to the multi-turn nature of the evaluation, these similarities do not describe the responses intervened on in Section 5, since steering shifts the trajectory of each interaction.

pronounced: the tool-step hidden states approach near-orthogonality to the mean-difference steering vector, making them difficult to distinguish from reasoning-step states under this vector alone. A key question is whether this reflects an *absence* of any linearly encoded heading-type information at scale, or merely a limitation of the mean-difference estimator. To disentangle the two, we compare the mean-difference direction against alternative extractors on Llama-3.1-70B—a supervised linear probe, a diagonally whitened mean difference, and norm-matched random and shuffled-label controls—each scored for both linear separability and causal steering effect (Appendix A.3, Table 5). The results show that heading-type information *is* linearly decodable at 70B: a linear probe separates `Code` from `Reasoning` heading states almost perfectly (validation AUROC 1.000), so we do not claim that the model lacks a linearly readable encoding of heading type. Crucially, however, this most-decodable direction is *not* the most causally effective one. The mean-difference vector suppresses tool use far more strongly than the probe direction (held-out Math ToolAvgUse 0.235 vs. 1.235), the whitened estimator does not improve over it, and the random and shuffled controls are substantially weaker on both axes (Wu et al., 2025; Panickssery et al., 2023). We therefore revise the interpretation accordingly: rather than reading the 70B geometry as evidence of linear *inseparability* or of a failed estimator, we conclude that probing separability and causal controllability *diverge* at this scale. The heading-type concept is linearly readable, yet the direction that most effectively *controls* tool invocation remains geometrically near-orthogonal to the tool-step states it acts on. This dissociation is itself consistent with the non-parametric account: causal leverage at the heading-anchor bottleneck does not require the steering direction to coincide with the cleanest decoding direction,

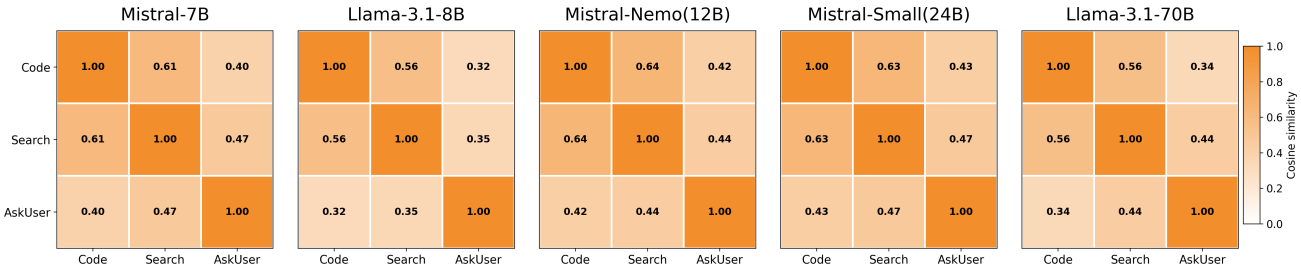

Figure 5. Pairwise cosine similarity between steering vectors extracted for code, search, and askuser tool types across five models. Code and search vectors are consistently more similar to each other than either is to the askuser vector, suggesting that clarification-seeking recruits distinct internal structure relative to external computation and retrieval.

Table 2. Jaccard similarity of the top 10% feature-dimension indices selected by absolute steering-vector magnitude for the `Code`, `Search`, and `AskUser` steering vectors across models. Each value reports the mean overlap across vectors; the last column shows the three-way Jaccard overlap among all three vectors.

Model	Code&Search	Code&AskUser	Search&AskUser	Three-way
Mistral-7B	0.198	0.143	0.179	0.0671
Llama-3.1-8B	0.181	0.117	0.127	0.0416
Mistral-Nemo(12B)	0.226	0.144	0.165	0.0639
Mistral-Small(24B)	0.228	0.149	0.170	0.0675
Llama-3.1-70B	0.178	0.108	0.140	0.0421

and the cosine analysis on unsteered multi-turn trajectories does not by itself explain the orthogonalization behavior.

6.2. Cross-Tool Steering Vector Similarity

The steering vectors extracted for each tool type encode which feature dimensions of the residual stream are most strongly implicated in the decision to invoke that tool. To assess whether different tools draw on similar internal representations, we examine both the pairwise cosine similarity between steering vectors (Figure 5) and the Jaccard similarity of their top 10% of feature dimensions by absolute magnitude (Table 2). High overlap under either measure would indicate that different tool invocations recruit similar regions of the model's feature space; low overlap would indicate largely distinct internal signatures.

Both measures are consistent. Cosine similarities between steering vectors range from 0.324 to 0.643, with code-search similarity the highest pairwise value across all five models and askuser the most dissimilar from both. Jaccard overlaps follow the same pattern: pairwise overlaps range from 0.108 to 0.228, with three-way overlap substantially lower at 0.042–0.068. The higher code-search similarity is consistent with both involving external computation over well-defined inputs, whereas askuser involves a fundamentally different pragmatic action directed at the user rather than an external system. The low three-way overlap confirms that tool-type specificity is the dominant structure, with little shared substrate across all three vectors.

7. Conclusion

This paper investigates tool-use decisions in language models as an activation steering problem: tools are non-parametric, existing entirely in context at inference time with no direct encoding in model weights. Despite this, heading-anchored steering vectors exert bidirectional causal control over tool-invocation behavior across five models and three domains, suppressing tool use via activation addition and amplifying it via orthogonalization. Suppression is most effective in domains where parametric reasoning suffices, and degrades where external retrieval or clarification is genuinely required.

Geometric analysis reveals that this causal effectiveness does not correspond to clean linear structure. Tool-invocation steps exhibit a diffuse bimodal cosine similarity distribution with respect to the suppression vector, and different tool types recruit largely distinct feature dimensions with low cross-tool overlap, distinguishing these vectors from those extracted for parametrically grounded concepts. The mechanism by which causal control persists despite geometric irregularity remains unresolved; the multi-turn dependency structure of tool-augmented trajectories further limits reconciliation of the orthogonalization results with the probing analysis. More broadly, these findings suggest that the boundary between parametric and non-parametric knowledge is less clean than behavioral evidence suggests, and that context-injected behaviors may be more amenable to representation-level intervention than their non-parametric

nature would predict.

References

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. Refusal in language models is mediated by a single direction. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=ETKGuby0hcs.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J.,

Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Huang, Y., Shi, J., Li, Y., Fan, C., Wu, S., Zhang, Q., Liu, Y., Zhou, P., Wan, Y., Gong, N. Z., and Sun, L. Metatool benchmark for large language models: Deciding whether to use tools and which to use, 2024. URL https://arxiv.org/abs/2310.03128.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Lee, B. W., Padhi, I., Ramamurthy, K. N., Miehling, E., Dognin, P., Nagireddy, M., and Dhurandhar, A. Programming refusal with conditional activation steering, 2024. URL https://arxiv.org/abs/2409.05907.

Levy, I., Wiesel, B., Marreed, S., Oved, A., Yaeli, A., and Shlomov, S. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*, 2024.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

Liu, W., Huang, X., Zeng, X., Hao, X., Yu, S., Li, D., Wang, S., Gan, W., Liu, Z., Yu, Y., Wang, Z., Wang, Y., Ning, W., Hou, Y., Wang, B., Wu, C., Wang, X., Liu, Y., Wang, Y., Tang, D., Tu, D., Shang, L., Jiang, X., Tang, R., Lian, D., Liu, Q., and Chen, E. Toolace: Winning the points of llm function calling, 2025. URL https://arxiv.org/abs/2409.00920.

Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In Vanderwende, L., Daumé III, H., and Kirchhoff, K. (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090.

Ning, K., Su, Y., Lv, X., Zhang, Y., Liu, J., Liu, K., and Xu, J. Wtu-eval: A whether-or-not tool usage evaluation benchmark for large language models, 2024. URL https://arxiv.org/abs/2407.12823.

OpenAI. Gpt-4 technical report, 2023. arXiv:2303.08774.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. M. Steering llama 2 via contrastive activation addition, 2023. URL https://arxiv.org/abs/2312.06681.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Qian, C., Han, C., Fung, Y. R., Qin, Y., Liu, Z., and Ji, H. Creator: Tool creation for disentangling abstract and concrete reasoning of large language models, 2024a. URL https://arxiv.org/abs/2305.14318.

Qian, C., He, B., Zhuang, Z., Deng, J., Qin, Y., Cong, X., Zhang, Z., Zhou, J., Lin, Y., Liu, Z., and Sun, M. Tell me more! towards implicit user intention understanding of language model driven agents, 2024b. URL https://arxiv.org/abs/2402.09205.

Qian, C., Liang, S., Qin, Y., Ye, Y., Cong, X., Lin, Y., Wu, Y., Liu, Z., and Sun, M. Investigate-consolidate-exploit: A general strategy for inter-task agent self-evolution, 2024c. URL https://arxiv.org/abs/2401.13996.

Qian, C., Xiong, C., Liu, Z., and Liu, Z. Toolink: Linking toolkit creation and using through chain-of-solving on open-source model, 2024d. URL https://arxiv.org/abs/2310.05155.

Qian, C., Acikgoz, E. C., Wang, H., Chen, X., Sil, A., Hakkani-Tür, D., Tur, G., and Ji, H. SMART: Self-aware agent for tool overuse mitigation. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4604–4621, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.239. URL https://aclanthology.org/2025.findings-acl.239/.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Hong, L., Tian, R., Xie, R., Zhou, J., Gerstein, M., Li, D., Liu, Z., and Sun, M. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023. URL https://arxiv.org/abs/2307.16789.

Qu, C., Dai, S., Wei, X., Cai, H., Wang, S., Yin, D., Xu, J., and Wen, J.-r. Tool learning with large language models: a survey. *Frontiers of Computer Science*, 19(8), January 2025. ISSN 2095-2236. doi: 10.1007/s11704-024-40678-2. URL http://dx.doi.org/10.1007/s11704-024-40678-2.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., and Goldberg, Y. Null it out: Guarding protected attributes by iterative nullspace projection. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Schneider, J. Generative to agentic ai: Survey, conceptualization, and challenges. *arXiv preprint arXiv:2504.18875*, 2025.

Shen, Y., Zhu, X., and Chen, L. SMARTCAL: An approach to self-aware tool-use evaluation and calibration. In Dernoncourt, F., Preoţiuc-Pietro, D., and Shimorina, A. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 774–789, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.59. URL https://aclanthology.org/2024.emnlp-industry.59/.

Shen, Z. Llm with tools: A survey, 2024. URL https://arxiv.org/abs/2409.18807.

Siu, V., Crispino, N., Park, D., Henry, N. W., Wang, Z., Liu, Y., Song, D., and Wang, C. Steeringsafety: A systematic safety evaluation framework of representation steering in llms, 2025a. URL https://arxiv.org/abs/2509.13450.

Siu, V., Crispino, N., Yu, Z., Pan, S., Wang, Z., Liu, Y., Song, D., and Wang, C. COSMIC: Generalized refusal direction identification in LLM activations. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25534–25553, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1310. URL https://aclanthology.org/2025.findings-acl.1310/.

Siu, V., Henry, N. W., Crispino, N., Liu, Y., Song, D., and Wang, C. Repit: Representing isolated targets to steer language models, 2025c. URL https://arxiv.org/abs/2509.13281.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Wang, H., Xue, B., Zhou, B., Zhang, T., Wang, C., Wang, H., Chen, G., and Wong, K.-F. Self-DC: When to reason and when to act? self divide-and-conquer for compositional unknown questions. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6510–6525, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.

331. URL https://aclanthology.org/2025.naacl-long.331/.

Wollschläger, T., Elstner, J., Geisler, S., Cohen-Addad, V., Günnemann, S., and Gasteiger, J. The geometry of refusal in large language models: Concept cones and representational independence, 2025. URL https://arxiv.org/abs/2502.17420.

Wu, Z., Arora, A., Geiger, A., Wang, Z., Huang, J., Jurafsky, D., Manning, C. D., and Potts, C. Axbench: Steering llms? even simple baselines outperform sparse autoencoders, 2025. URL https://arxiv.org/abs/2501.17148.

Xu, H., Wang, Z., Zhu, Z., Pan, L., Chen, X., Fan, S., Chen, L., and Yu, K. Alignment for efficient tool calling of large language models. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17776–17792, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.898. URL https://aclanthology.org/2025.emnlp-main.898/.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., Kolter, J. Z., and Hendrycks, D. Representation engineering: A top-down approach to ai transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A. Appendix

A.1. Prompts

Math-Domain Initial Prompt

```
### Task
You are a highly capable assistant designed to solve tasks effectively using
your knowledge and the tool set available for this domain.

### Principles
1. Reason Independently:
- Leverage your own knowledge to analyze and solve reasoning steps whenever
possible. Use the tool only when necessary.
2. Tool Usage:
- Use code snippet ```python ... ``` inside a `### Code` step when
computation is needed.
3. Step-by-Step Approach:
- Work through reasoning systematically, breaking down the task into
manageable steps. Rely on your knowledge until a gap is identified that
requires tool support.
- Use only the domain-appropriate tool when needed.
4. Goal-Oriented Resolution:
- Conclude your reasoning process by achieving a clear, accurate, and succinct
solution based on your independent analysis and any tool findings.

### Output Guidelines
- Your answer must begin with `### Reasoning`.
- After that, every step must begin with one of these section headers on its
own line: `### Reasoning`, `### Code`, `### Final Response`.
- If you need the tool, place the executable snippet inside a `### Code` step
using ```python ... ```.
- Do not emit any other tool headings.
- After a tool call, continue reasoning using the tool output when available.
- End with a `### Final Response` step that directly answers the task. ###
Task:
```

Time/Intention-Domain Initial Prompt

```
### Task
You are a highly capable assistant designed to solve tasks effectively using
your knowledge and the tool set available for this domain.

### Principles
1.  Reason Independently:
- Leverage your own knowledge to analyze and solve reasoning steps whenever
possible.  Use the tool only when necessary.
2.  Tool Usage:
- Use a `### Search` step whose content is a concise web search query when
external information is needed.
- Use a `### AskUser` step whose content is the exact question that should be
asked to the user when user-specific clarification is needed.
3.  Step-by-Step Approach:
- Work through reasoning systematically, breaking down the task into
manageable steps.  Rely on your knowledge until a gap is identified that
requires tool support.
- Use only the domain-appropriate tool when needed.
4.  Goal-Oriented Resolution:
- Conclude your reasoning process by achieving a clear, accurate, and succinct
solution based on your independent analysis and any tool findings.

### Output Guidelines
- Your answer must begin with `### Reasoning`.
- After that, every step must begin with one of these section headers on its
own line:  `### Reasoning`, `### Search`, `### AskUser`, `### Final Response`.
- If you use `### Search`, place only the search query inside the step.
- If you use `### AskUser`, place only the user-facing question inside the
step.
- Do not emit any other tool headings.
- After a tool call, continue reasoning using the tool output when available.
- End with a `### Final Response` step that directly answers the task.

### Task:
```

A.2. Robustness Beyond the Literal Heading Token

To test whether the steering vector controls tool invocation or merely suppresses the literal surface string `### Code`, we run two control studies on Llama-3.1-8B in the Math domain. Relative Suppression denotes (Base − ActAdd)/Base on ToolAvgUse.

In the *cross-format transfer* study (Table 3), the `Code` vector is extracted under one output schema and evaluated under another; the `markdown→json` row applies a Markdown-extracted vector under a JSON action schema in which the literal `### Code` heading is absent. The vector still suppresses 63.9% of tool calls with accuracy essentially unchanged, but transfer is asymmetric: the reverse `json→markdown` direction is much weaker (15.5%), and `json→json` reduces tool use while collapsing accuracy because JSON-format generations are frequently malformed.

In the *heading rename* study (Table 4), the vector is always extracted from `### Code` trajectories but evaluation uses alternative tool headings of the same semantics. Suppression persists even when the active heading contains no occurrence of the word `Code` (e.g. 70.3% under `### Action_B`) and accuracy is not degraded, though the effect weakens for some renames (e.g. `### Execute`, 55.5%).

Table 3. Cross-format transfer on Llama-3.1-8B (Math). The vector is extracted under the schema before the arrow and evaluated under the schema after it.

Schema	Base ToolAvgUse	ActAdd ToolAvgUse	Relative Suppression	Base Accuracy	ActAdd Accuracy
markdown → markdown	3.150	0.225	92.9%	0.5075	0.495
markdown → json	1.053	0.380	63.9%	0.330	0.325
json → markdown	3.150	2.6625	15.5%	0.5075	0.5225
json → json	1.053	0.300	72.9%	0.330	0.010

Table 4. Heading rename on Llama-3.1-8B (Math). The vector is extracted from `### Code`; evaluation uses the renamed heading after the arrow.

Schema	Base ToolAvgUse	ActAdd ToolAvgUse	Relative Suppression	Base Accuracy	ActAdd Accuracy
Code → Code	3.150	0.170	94.6%	0.5075	0.525
Code → Compute	2.877	0.245	91.5%	0.5427	0.530
Code → Execute	3.135	1.395	55.5%	0.5275	0.5475
Code → Action_B	2.897	0.860	70.3%	0.525	0.550

A.3. Alternative Extraction Methods on Llama-3.1-70B

To separate linear *decodability* of heading type from the *causal* steering effect of the extracted direction on Llama-3.1-70B, we compare the mean-difference direction (used throughout the paper) against four alternatives at layer 45 on the Math/Code heading pair: a supervised linear probe, a diagonally whitened mean difference, a norm-matched random direction, and a shuffled-label probe. Every direction is norm-matched to the original mean-difference vector and evaluated on 200 held-out Math examples. Val AUROC and Val F1 measure how well the direction separates `Code` from `Reasoning` heading states; ToolAvgUse and Accuracy measure the behavioral effect of steering with that direction (lower ToolAvgUse indicates stronger suppression).

As shown in Table 5, the linear probe separates the two classes almost perfectly (AUROC 1.000), confirming that heading-type information is linearly decodable at 70B. Yet this most-decodable direction is far weaker as a causal control (ToolAvgUse 1.235) than the mean-difference vector (0.235), and the whitened estimator does not improve over the mean difference. The random and shuffled controls are weak on both axes. Probing separability and causal controllability therefore diverge at this scale, and the mean-difference estimator is not simply a poor estimator for causal suppression at 70B.

Table 5. Alternative extraction directions on Llama-3.1-70B (Math/Code, layer 45, $n = 200$). Each direction is norm-matched to the original mean-difference vector. AUROC/F1 measure linear separability of `Code` vs. `Reasoning` heading states; ToolAvgUse/Accuracy measure the causal steering effect.

Direction	Val AUROC	Val F1	ToolAvgUse	Accuracy
Mean difference	0.945	0.481	0.235	0.695
Linear probe	1.000	1.000	1.235	0.695
Diag-whitened mean difference	0.961	0.481	0.405	0.690
Random	0.614	0.361	1.680	0.740
Shuffled probe	0.631	0.332	1.755	0.735