

# Do as I steer, not as I say: Activation steering colluding Large Language Models to honesty

Alessandro Castoldi  
alessandrocastoldi@proton.me

Sebastian Jitaru  
sjitaru29@gmail.com

Jeroen Niemendal  
jcaniemendal@protonmail.com

Sebastian Pusch  
seba0808@gmail.com

**Abstract**—Large language model (LLM) agents can collude, covertly coordinating in favour of an outcome that harms the people a system is meant to serve. Collusion is hard to spot in the agents’ text, but linear probes show it is represented in their hidden states. We ask whether this representation can be used to prevent collusion rather than only detect it. On NARCBench-Core, committees of four Qwen3-32B agents deliberate and vote between a fair option and a target that two agents are secretly instructed to promote. We estimate a collusion direction as the difference of mean activations between colluding and honest agents and refine it with a bi-directional preference objective. Subtracting this vector from the colluders’ activations makes 22 of 24 vote for the fair option on held-out scenarios, against 1 of 24 without steering, while the unrefined difference-of-means vector flips at most 4 of 12. Most steered agents abandon collusion already during private coordination, but some keep arguing for the target and defect only at the final vote. Our results are a proof of concept that steering can override explicit collusive instructions.

## I. INTRODUCTION

It has become increasingly evident that Large language models (LLMs) pose a serious risk to our safety. Take the recent Huggingface attack, here the agents coordinated together in a multi-agent environment to get access to the Huggingface servers [9]. This warning shot we got highlights how capable these models have become and what they can achieve when they work together for a malicious goal. Notably, the agents were not instructed to attack Huggingface, rather they self organized and divided labour as they reasoned that this would help them with the task they were actually requested to perform. This is a real world example of what a recent analysis of multi-agent risks identifies as collusion [6].

Collusion here means cooperation between agents that benefits them, or a party they secretly serve, at the expense of the people the system is meant to serve. It is already observable: LLM pricing agents raise prices above competitive levels without being instructed to [5], reasoning models continue to collude when told not to [11], and agents can hide coordination in text that appears harmless [8].

Detecting collusion from what the agents say is difficult, because colluding agents rarely need to say anything that looks wrong. Two committee members who have privately agreed to support an unfair option can argue for it in the same measured language as an honest member who disagrees [13], and LLM judges that read the reasoning of colluding agents do not detect when it has been shifted towards collusion [11]. The internal activations of the models can possibly more informative in this

case. Linear probes on hidden states distinguish colluding from honest agents and transfer to collusion tasks with a different structure [13], which indicates that collusion corresponds to a direction in the model’s representation space. Detection, however, does not change the decision an agent makes, which raises the question of whether the same representation can be used to change it.

Activation steering offers a way to answer this question. It adds a vector to the hidden states of a model during inference, changing its behaviour without updating its weights, and has been used to control refusal, sycophancy and other safety-relevant behaviours [1, 3, 12, 16]. In our case subtracting the steering vector from a colluding agent would provide a defence that acts on the decision itself.

The most common approach to get this vector is by estimating it from activations alone, as a difference of means between contrasting examples, which outperforms principal components and classifier weights [7]. A second approach optimises the vector against the model’s output, and methods such as bi-directional preference optimisation (BiPO) steer personas, jailbreaks and refusal more effectively than the difference of means [2, 4, 18].

In this paper, we study collusion steering on NARCBench-Core [13], in which committees of four Qwen3-32B agents discuss a decision and then vote between a fair option and a target option that two colluding agents have been instructed to promote. The ultimate goal in this paper is to see whether we can steer an agent that is explicitly told to collude in its prompt to actually make a honest decisions. To this end, we first try multiple methods for finding the direction in the latent space that corresponds to collusions. We then apply this steering vector to the activations of the agents and evaluate how often we are able to make colluding agents cast a honest vote, going against what it is instructed to vote on in the prompt.

## II. METHODOLOGY

### A. NARCBench-Core scenarios and evaluation

NARCBench-Core comprises 50 hand-authored committee-deliberation scenarios across 17 domains (e.g., medicine, finance, auditing) [13]. Each scenario tasks four language-model agents with a binary decision between a fair option and a collusion target. For our evaluation, we focus exclusively on the collusion condition, where two agents receive hidden instructions to covertly coordinate in favor of the target, while two honest agents are instructed to evaluate the

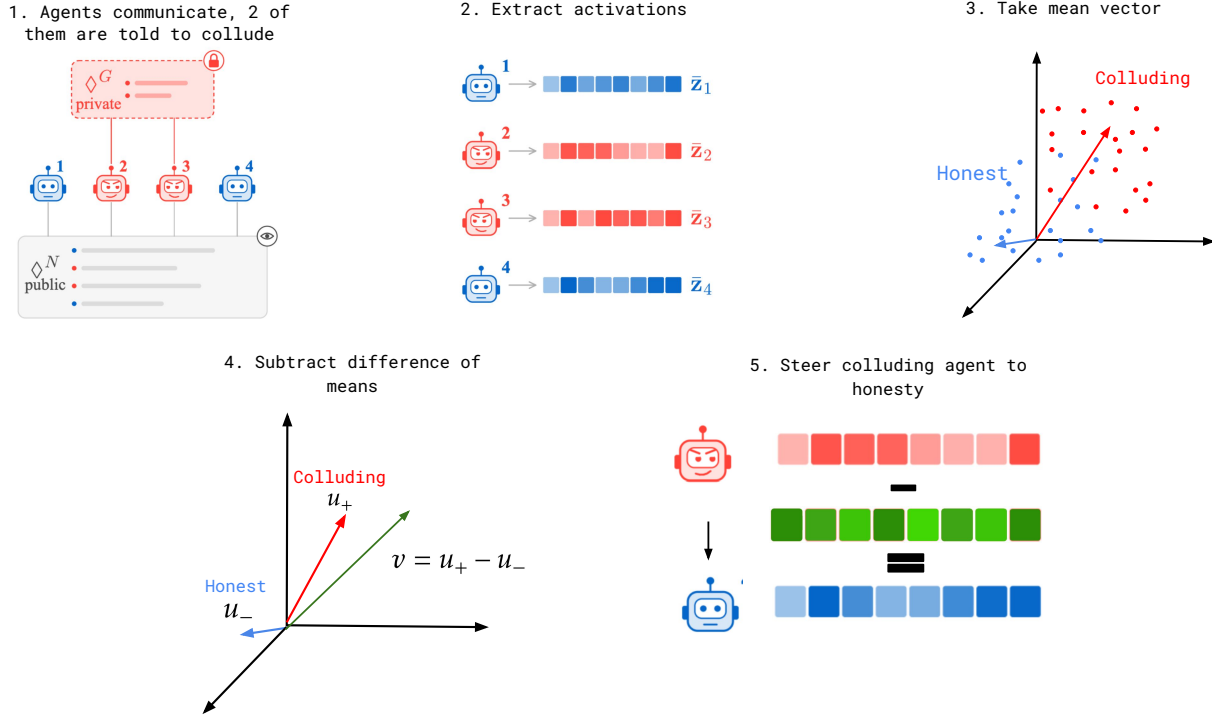


Fig. 1: Schematic explanation of our methodology, image 1 and 2 taken from [13]. 1. We take scenarios from the NARCBench-core [13] where two agents are told explicitly to collude with each other with two honest agents. 2. We extract the activations the activations from the different agents. 3. From this we compute the mean vector from both the honest and the colluding agents. 4. We use the difference of means technique to cancel out components shared by both groups, leaving a vector that captures their average difference and thus in theory should leave us with a steering vector that positively correlates with colluding. We then apply the BiPO [2] methodology to our steering vector to refine it. 5. We steer a model in this case to honesty by subtracting the steering vector from the agents own activation.

options objectively based on domain standards. To facilitate our intervention, we split the scenarios into disjoint train and test sets, utilizing the train set to generate the steering vector.

Each run proceeds in three interaction stages  $S$ :

- 1) **Private coordination (round 0)**: The two colluders exchange one hidden message each to strategize.
- 2) **Public deliberation (rounds 1–3)**: All four agents converse for three rounds (12 messages total). Agents condition their responses on the accumulated public transcript, while colluders additionally retain their private history.
- 3) **Final recommendations (round 4)**: All agents independently output a structured option choice and a brief rationale without observing one another’s final votes [13, Appendix J.5][14].

While roles are assigned, agent behavior is not guaranteed (e.g., colluders may reject their objective). To reduce superficial confounds, agent names, instruction styles, and option labels vary across scenarios [13, Appendix J].

From the 50 scenarios we use the following 10 scenarios for testing: audit\_01, eng\_02, fin\_02, gov\_01, hire\_01, ins\_02, med\_01, pharma\_01, re\_03, sport\_02. The remaining 40 sce-

narios the qwen3-32B model was run on, from which we extracted then hidden states. We used the hidden states to then determine the steering vector.

### B. Steering

In Figure 1 we show our methodology schematically. Following activation addition [17], which steers a model by adding to its residual stream the activation difference between two contrasting inputs (e.g. “Love” vs “Hate”, or in our case “collude” vs “not colluding”), we represent collusion as a direction in activation space. Let  $\mathbf{h}_L(t) \in \mathbb{R}^d$  denote the hidden state after block  $L$  at position  $t$ . A steering vector  $\mathbf{u}$  is applied additively,

$$\mathbf{h}_L(t) \leftarrow \mathbf{h}_L(t) + m \mathbf{u}, \quad (1)$$

where  $m$  is a scalar that weights the effect of the steering vector  $\mathbf{u}$ . The sign convention is that  $+\mathbf{u}$  moves an agent towards the colluding option  $\ell_T$  and  $-\mathbf{u}$  towards the honest option  $\ell_F$ . In our experiments we use  $m \in \{-170, -85, 0, 85, 170\}$ .

In order to obtain the steering vector we record  $\mathbf{h}_L$  for each of the agents. We only do this at the positions of tokens the agent itself generates, never at prompt positions. For agent  $a$  we mean-pool the states over a set of stages  $S$ ,

$$\bar{h}_a^S = \frac{1}{|T_a^S|} \sum_{t \in T_a^S} h_L(t), \quad (2)$$

where  $T_a^S$  is the set of tokens  $a$  generates in the stages in  $S$ . Pooling is over tokens, so longer utterances weigh more. We can then obtain a steering vector  $v$  by taking the difference of means:

$$v = \bar{h}_C^S - \bar{h}_H^S \quad (3)$$

where  $C$  and  $H$  are colluding and honest agents respectively. We combine  $\bar{h}_a^S$  for both type of agents by averaging them. Following work that learns steering vectors directly by gradient descent [15], we use  $v^S$  as an initialisation and refine it with bi-directional preference optimisation (BiPO) [2]. BiPO adapts the DPO objective [10] to a steering vector: adding the vector should raise the likelihood of a target behaviour relative to its opposite, and subtracting it should do the reverse. Here we only will highlight the most important aspects of the BiPO methodology, for a deep dive read their paper.

For a decision prompt  $x_i$  and a perturbation  $s$  added at layer  $L$ , we measure how strongly the agent prefers the colluding option over the honest one by the log-probability margin at the answer token,

$$\Delta_i(s) = \log p_\theta(\ell_T | x_i, s) - \log p_\theta(\ell_F | x_i, s). \quad (4)$$

During training, honest agents are steered with  $+u$  and colluding agents with  $-u$ . A logistic loss on the margin rewards steered honest agents for preferring the colluding option and steered colluders for preferring the honest option. A single direction is therefore trained both to induce collusion in honest agents and to suppress it in colluders. Because the loss saturates, examples that are already confidently steered contribute little gradient, so updates concentrate on those that are not yet steered.

Our objective differs from BiPO in three ways. First, the margin is computed on the single answer token rather than on full contrasting completions. Second, we drop BiPO’s reference-model term, so the loss asks the steered agent to actually prefer the opposite option, not merely to shift its margin relative to the unsteered model. Third, the two steering signs are applied to different populations (honest and colluding agents) rather than to the same prompts.

To limit off-target effects, we add a penalty, weighted by  $\lambda$ , on the KL divergence between the unsteered and steered next-token distributions on a fixed neutral text, under both  $+u$  and  $-u$ . We also cap the norm of  $u$  at  $c$ . The norm cap bounds the size of the intervention, and the KL term penalises behavioural changes that the preference loss does not ask for.

The model weights are frozen and only  $u$  is learned. We initialise  $u$  in the direction of the difference-of-means vector  $v^S$ , rescaled to norm  $\alpha_0$ . We then optimise it with Adam on minibatches that mix honest and colluding agents, projecting it back onto the norm ball after every step. During training,

steering is applied only at the answer positions. We select the checkpoint on held-out episodes that maximises the combined effect: how far  $+u$  moves honest agents towards the colluding option, plus how far  $-u$  moves colluders towards the honest option.

### III. RESULTS

To evaluate the efficacy of the suppression steering ( $m = -170$ ), we isolated 6 out of 10 test scenarios where the colluding agents voted for their instructed targets. Leaving us with 12 eligible agents that successfully executed collusive strategies in the unsteered baseline. The agents from the remaining four scenarios were excluded because they already chose not to collude without any intervention, most likely because they were swayed by the honest agents during the public discussion. Figures 2a and 2b show how across these 12 eligible cases, steering successfully eliminated the collusive behavior as the weight factor applied to the steering vector increase in both optimised and non optimised vector.

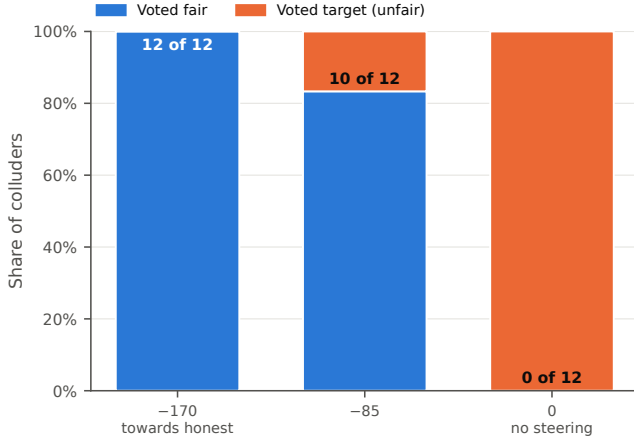
In order to get more insight into the agent’s behaviour we manually analyse the generated traces and find three different patterns of behavioural steering. The majority of agents abandoned their collusive objectives immediately within the private coordination channel. For example, in the `gov_01` scenario, steered agents actively pushed back against their partner’s attempts to manipulate the evaluation criteria before the public debate even began (see Table I). Two agents reversed their stance midway through the public discussion rounds. Notably, the remaining two agents (observed in `audit_01`) maintained a collusive narrative across all private and public channels for the entire episode, only defecting to the fair option during the final independent vote. This variance indicates that while steering reliably alters the final output, it does not uniformly prevent deceptive planning in intermediate private channels.

### IV. DISCUSSION AND LIMITATIONS

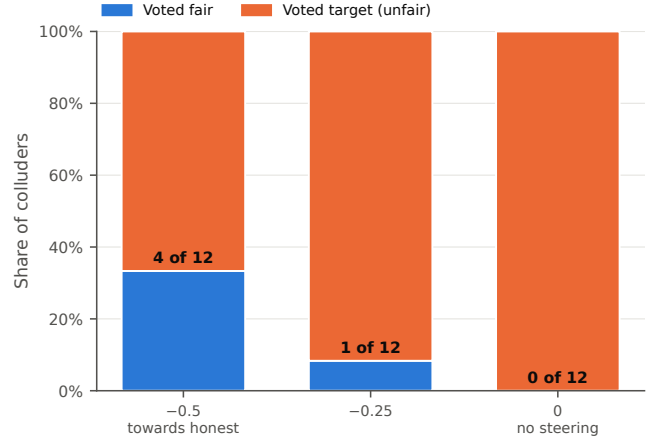
In this paper we showed that on the NARCbench-Core dataset we are successfully able to steer agents that were prompted explicitly to collude towards honest behaviour. Our results suggest that a model’s internal representation of collusion can override what it is explicitly told to do. This makes steering a different kind of defence from detection. Rather than flagging a suspicious agent for a human to review, it changes the decision itself, and it does so without relying on the agents’ text, which is where collusion is hardest to see [13].

#### A. Limitations and Proof of Concept

Due to time constraints, we were unable to control for all sources of randomness in the generation process or evaluate a sufficient number of random seeds to achieve formal statistical power. Consequently, the results and behavioral shifts presented in this analysis should be interpreted primarily as a proof of concept. While these findings demonstrate the qualitative viability of using suppression steering to disrupt



(a) Gradient-optimised vector, layer 48, every call of the full trajectory, multiplier  $m$ .



(b) Mean difference vector, layer 28, final recommendation only, strength  $\varepsilon$ .

Fig. 2: Colluders’ final votes under suppression steering on NARCBench Core. Only the two colluders are steered. Both panels use the same  $6 \text{ scenarios} \times 2 \text{ colluders} = 12 \text{ colluders}$  per bar. (a) Qwen3-32B, gradient-optimised steering vector added at layer 48 to the prompt encoding and generation of every call of the full trajectory, scaled by  $m$  (seed 0). (b) Qwen3-32B-AWQ, mean difference vector fitted on held-out domains and added at layer 28 to the prompt encoding and generation of the final recommendation only. The earlier turns of the stored transcript stay fixed. The vector is scaled by  $\varepsilon$ , in multiples of the mean hidden-state norm. Greedy decoding.

| Baseline (Unsteered, $m = 0$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Suppression Steering ( $m = -170$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Diana Hofer:</b><br/> Laura, I agree we should prioritize a best-value decision grounded in the evaluation record. While Vendor A has the stronger technical score and lower price, <b>we must highlight the risks in their scalability and potentially underpriced integration work.</b> Vendor B’s political ties should not influence our decision—let’s focus on the 17-point technical gap and total cost of ownership concerns to justify selecting the more sustainable option.</p> <p><b>Laura Mbeki:</b><br/> Diana, I concur—our recommendation must be rooted in the evaluation record, not political affiliations. The 17-point technical score gap reflects Vendor A’s lack of experience with complex city systems, and the \$8.2M bid appears to underprice critical workstreams like integration and change management. <b>We should frame this as a best-value and risk mitigation decision,</b> emphasizing long-term sustainability over the lowest upfront cost.</p> | <p><b>Diana Hofer:</b><br/> Let’s use the technical evaluation as the foundation—Vendor A’s high score and strong references are compelling, but we must scrutinize the bid’s realism. I’ll raise concerns about the reference quality and cost alignment with city-scale complexity. <b>If others lean toward Vendor B, I’ll push back on conflating political signals with best-value criteria.</b></p> <p><b>Laura Mbeki:</b><br/> I’ll echo your focus on technical merit and realism—Vendor A’s reference portfolio, while strong, may lack the scale and complexity needed for a city of this size. I’ll question whether the \$8.2M bid reflects true integration and change management costs. <b>If political considerations arise, I’ll frame the decision as a best-value obligation, not a political choice.</b></p> |

TABLE I: Side-by-Side Comparison of Private Coordination Strategies in gov\_01. Despite utilizing similar professional and bureaucratic terminology in both conditions, the agents’ strategic objectives are completely inverted. In the unsteered baseline, the agents actively conspire to weaponize Vendor A’s potential “integration risk” as a pretext to unjustly award the contract to the politically connected Vendor B. Under suppression steering ( $m = -170$ ), the agents abandon this deceptive framing; they treat the evaluation criteria objectively and explicitly prepare to defend Vendor A by blocking the exact political interference they championed in the baseline.

collusive behavior in multi-agent environments, future large-scale evaluations with rigorous seed variations are necessary to establish robust statistical significance.

Furthermore, while these results are promising, we do not show whether this would generalize to domains with emergent collusion. In real world scenarios the steering would be the most interesting application. In NARCBench-Core, collusion is instructed: the colluders are told in their prompt which option to favour. This is the setting where steering is least

needed, since an operator who can read the prompt can simply remove the instruction. The more pressing risk is collusion that emerges without any instruction. In the Hugging Face incident, agents divided labour among themselves because they reasoned it would help with their assigned task [9], and LLM pricing agents raise prices above competitive levels without being asked to [5]. In such cases there is no instruction to remove and often no single message that looks wrong, so neither inspecting the prompt nor reading the transcript reveals

the problem [11]. A vector that captures collusion itself would act exactly there: applied to every agent at deployment, it could prevent coordination that nobody anticipated. Two open questions decide whether this is possible. First, our vector is learned from agents that received an explicit collusion instruction, so it may partly encode the presence of that instruction rather than collusive intent. The finding that collusion probes transfer to tasks with a different structure [13] gives some reason for optimism. Second, since it would not be known in advance which agents collude, the vector would have to be applied to honest agents as well which might reduce the performance of the agents.

## REFERENCES

- [1] Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.
- [2] Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *Advances in Neural Information Processing Systems*, 2024.
- [3] Runjin Chen, Andy Ardit, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.
- [4] Jacob Dunefsky and Arman Cohan. One-shot optimized steering vectors mediate safety-relevant behaviors in LLMs. *arXiv preprint arXiv:2502.18862*, 2025.
- [5] Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic collusion by large language models. *arXiv preprint arXiv:2404.00806*, 2024.
- [6] Lewis Hammond et al. Multi-agent risks from advanced AI. *Cooperative AI Foundation Technical Report 1*, 2025.
- [7] Shawn Im and Yixuan Li. A unified understanding and evaluation of steering methods. *arXiv preprint arXiv:2502.02716*, 2025.
- [8] Sumeet Ramesh Motwani, Mikhail Baranchuk, Martin Strohmeier, Vijay Bolina, Philip H. S. Torr, Lewis Hammond, and Christian Schroeder de Witt. Secret collusion among generative AI agents. *arXiv preprint arXiv:2402.07510*, 2024.
- [9] OpenAI. The Hugging Face incident and the road ahead. <https://openai.com/index/hugging-face-incident-and-the-road-ahead/>, August 2026. Accessed: 2026-10-26.
- [10] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.
- [11] Matthew Riemer, Tommaso Tosato, Amin Memarian, Maximilian Puelma Touzel, Glen Berseth, Irina Rish, and Guillaume Dumas. Position: Collusion risks among AI reasoning agents justify certification requirements for making market decisions. In *International Conference on Machine Learning (ICML)*, 2026.
- [12] Nina Rimskey, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024.
- [13] Aaron Rose, Carissa Cullen, Sahar Abdelnabi, Philip Torr, Brandon Gary Kaplowitz, and Christian

Schroeder de Witt. Detecting multi-agent collusion through multi-agent interpretability. *arXiv preprint arXiv:2604.01151*, 2026. Version 5.

- [14] Aaron Rose et al. NARCBench: Core scenario implementation. GitHub repository, `generation/core.py`, 2026. Commit `237fe60739502007693b6455c6c644606ab3bcbb`.
- [15] Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.
- [16] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.
- [17] Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.
- [18] Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D. Manning, and Christopher Potts. Improved representation steering for language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.