ELICITING SANDBAGGED LLM CAPABILITY VIA STRUCTURAL INTERVENTIONS ON ATTENTION-HEAD FUNCTIONAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliably evaluating the capabilities of advanced AI systems is critical for safe development and deployment, yet such evaluations may be undermined by sandbagging, in which a model strategically suppresses capabilities it possesses. Existing work has developed controlled sandbagging settings and methods for detecting or eliciting hidden capabilities, but comparatively little is known about how a model's internal computation changes when its capabilities are suppressed. In this work, we study sandbagging through the lens of internal functional organization. Inspired by network neuroscience, we construct functional graphs from attention-head activations and compare their organization under honest and sandbagging conditions while holding the model architecture and weights fixed. Across model architectures and community detection methods, we find a consistent structural signature: sandbagging is associated with reduced functional connectivity within detected communities. We further show that this reorganization provides an actionable signal for capability elicitation. Ablating attention heads that exhibit the largest connectivity changes substantially recovers suppressed performance while largely preserving performance under honest behavior. Moreover, incorporating these structural signals further improves existing inference-level and weight-level elicitation methods. Together, our results show that sandbagging is accompanied by systematic reorganization of internal computation and demonstrate that functional network structure can provide both a mechanistic lens for understanding capability suppression and a practical signal for recovering hidden capabilities.

1 INTRODUCTION

As AI systems become increasingly capable, reliably assessing what they can actually do is critical for safe development and deployment. A growing challenge is sandbagging, defined as strategic underperformance during evaluation (van der Weij et al., 2025). A sandbagging model possesses a capability but deliberately exhibits lower performance under evaluation, causing measured capability to underestimate what the model can achieve in other contexts. Sandbagging is therefore distinct from ordinary failure or safety-motivated refusal: its defining feature is strategic, context-dependent capability suppression.

Sandbagging can arise from two broad sources (van der Weij et al., 2025). In developer-induced sandbagging, capability suppression is deliberately introduced to create controlled settings in which hidden capabilities can be studied. For example, models can be instructed to underperform or trained to reveal a capability only when a particular password is provided (van der Weij et al., 2025; Greenblatt et al., 2024b). In model-initiated sandbagging, strategic underperformance instead emerges from the model's learned objectives or behavior. Recent work has shown that models can adapt their behavior strategically under anticipated training (Greenblatt et al., 2024a) and, in controlled settings, deliberately underperform without being explicitly instructed to sandbag (Meinke et al., 2024). These findings raise concerns that capability evaluations may fail to reveal a model's true competence.

Prior work has therefore focused on two complementary goals: constructing controlled forms of sandbagging for study and detecting or overcoming capability suppression once it occurs. Detection

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

methods include behavioral audits and activation-based probes (Goldowsky-Dill et al., 2025b; Taylor et al., 2025), while elicitation methods attempt to recover hidden capabilities through inference-time or weight-level interventions (Greenblatt et al., 2024b; Hofstätter et al., 2025; Tice et al., 2025). These approaches provide important tools for studying and mitigating sandbagging. However, they leave a more fundamental question comparatively underexplored: what changes inside a model when an available capability becomes strategically suppressed? Existing white-box approaches often use internal representations as diagnostic signals but provide limited insight into how interactions among model components reorganize during sandbagging.

In this work, we study sandbagging through the lens of internal functional organization. Motivated by network neuroscience, we construct functional graphs from attention-head activations and compare honest and sandbagging conditions while keeping the model architecture and weights fixed. We observe a consistent pattern of functional reorganization across different model architectures: sandbagging is associated with reduced connectivity within detected communities. We then ask whether this structural signature can guide targeted interventions for recovering suppressed capabilities. Ablating attention heads that exhibit the largest connectivity changes increases capability elicitation while leaving performance under the honest condition nearly intact. Finally, we show that these structural insights can further improve existing inference-level and weight-level elicitation methods. Together, our results suggest that sandbagging is accompanied by systematic changes in the organization of internal computation and that these changes provide actionable signals for understanding and recovering suppressed capabilities.

This work makes the following main contributions:

- **Functional network analysis of sandbagging.** We introduce functional networks constructed from attention-head activations as a novel lens for studying sandbagging. Across different model architectures and community detection methods, we observe consistent structural differences between honest and sandbagging conditions, indicating that sandbagging is accompanied by a reproducible shift in head-level functional organization. While not establishing a causal relationship between these changes in network properties and sandbagging, this method provides a unique way to quantitatively measure complex internal interactions.
- **Interventions to improve capability elicitation.** We identify heads through large changes in network connectivity and show that ablating them recovers suppressed capabilities more effectively while largely preserving honest accuracy. We apply these findings to develop effective and efficient interventions that improve capability elicitation at both the inference and weight levels.

2 Preliminaries

In this section, we will introduce the main graph notation and key metrics used throughout the paper.

2.1 Notation

We consider two conditions, honest and sandbagging, indexed by $C \in \{\mathrm{H}, \mathrm{S}\}$. We write $G^C = (V, E^C)$ for an undirected graph under condition C. The two graphs share the node set $V = \{1, \ldots, M\}$, while their edge sets may differ. We denote the set of unordered node pairs by $\mathcal{P} = \{(i, j) \mid 1 \leq i < j \leq M\}$, with $E^C \subseteq \mathcal{P}$. Superscripts indicate condition-dependent quantities. The degree of node i is denoted by $\deg_{G^C}(i)$. A community partition assigns each node a label c_i, and $\mathcal{C}(i) = \{j \in V \mid c_j = c_i\}$ denotes the nodes sharing its label.

3 Functional Network of Attention Heads

Our goal is to characterize how attention heads interact under honest and sandbagging conditions. This is motivated by similar work at the intersection of network control theory and neuroscience, in which connectivity within human brains was found to shift between truthful and deceptive conditions (Saryazdi et al., 2026; Ding et al., 2017; Jiang et al., 2015). In these studies, brain regions are treated as nodes of a graph, and activity correlations between different regions are treated as edges.

We apply this perspective to LLMs, treating attention heads as nodes and the statistical dependence between their activations as edges. We then compare the graphs between the two settings to determine whether or not there are meaningful changes. We hypothesize that sandbagging induces stable changes in how attention heads in an LLM are organized.

3.1 Network Construction

Consider a transformer-based language model with L layers and H attention heads per layer, giving $M = LH$ heads in total. Each head forms a node in the graph. We index heads by $i \in \{1, \dots, M\}$ and write d for the dimension of a head's output. For each condition $C \in \{\mathrm{H}, \mathrm{S}\}$, we evaluate the same n questions and record each head's output immediately before the attention output projection W_O at the final prompt token preceding the answer. Collecting these outputs gives an activation matrix $X_i^C \in \mathbb{R}^{n \times d}$ for each head, with one row per question. The model's architecture and weights remain fixed across the two conditions.

To define an edge weight, we measure how strongly two heads' activations vary together across questions. Because each activation is a vector, we use the multivariate comparison provided by SVCCA (Appendix A.1). We center each feature of X_i^C across questions and retain its leading K singular directions. Applying canonical correlation analysis to the reduced activations of heads i and j gives canonical correlations $\rho_{ij,1}^C, \dots, \rho_{ij,K}^C$. These quantify shared linear variation without requiring individual activation coordinates to correspond between the two heads.

We aggregate the canonical correlations into a single edge weight using the Gaussian mutual-information expression,

$$S_{ij}^C = -\frac{1}{2} \sum_{r=1}^{K} \log\Big(1 - \big(\rho_{ij,r}^C\big)^2\Big). \tag{1}$$

This aggregation combines information across retained canonical directions while emphasizing stronger correlations. Under a joint Gaussian assumption, Equation 1 provides a plug-in estimate of mutual information; otherwise, S_{ij}^C can be interpreted more generally as a covariance-based dependence measure. An assessment of the Gaussian assumption is provided in Appendix A.2.

The scores form a symmetric similarity matrix S^C. We construct $G^C = (V, E^C)$ by retaining the top 25% of unordered head pairs in $\mathcal{P}$ according to S_{ij}^C and treating the resulting edges as unweighted. This ensures that the honest and sandbagging graphs have identical nodes and edge counts, allowing us to compare how their strongest functional dependencies are organized.

This dependence-based construction is particularly suited to sandbagging, where capability suppression may alter how representations are shared or coordinated across attention heads. Comparing the resulting graphs therefore allows us to characterize changes in functional organization between honest and sandbagging conditions.

3.2 Community Structure

There are several motivating reasons to analyze attention-head networks at the level of communities. First, prior work shows that computation in transformers is often carried out by groups of heads acting together rather than by individual heads (Olsson et al., 2022; Wang et al., 2023). In addition, a single head may be too small a unit to reveal how the graph is organized, while a single statistic over the entire graph cannot show where organizational changes happen. We therefore focus on understanding sandbagging functional graphs through the lens of community reorganization.

Specifically, given the functional graph $G^C = (V, E^C)$, we partition it into communities (i.e., groups of attention heads that are more densely connected to each other than to the rest of the network). This helps us study whether heads that work together under the honest condition continue to interact strongly under the sandbagging condition. We measure this by how strongly each head connects to its own community. Let

$$e_i^C \;=\; \frac{1}{A} \sum_{j \neq i} \mathbf{1}\big[(i,j) \in E^C\big]\, \mathbf{1}\big[c_i = c_j\big], \tag{2}$$

where the summation computes the total number of within-community edges for head i, and A is a normalization factor. Depending on our analysis, we consider two ways to normalize this count by defining A differently:

- **Within-community degree ($\tilde{k}_i^C$):** By setting $A = M - 1$, we normalize by the total number of possible connections. This quantity represents the fraction of possible nodes that head i is connected to within its own community.
- **Within-community fraction (f_i^C):** By setting $A = \deg_{G^C}(i)$, we normalize by the actual degree of head i. This represents the share of a head's own edges that stay within its community. We use f_i^C when reporting graph-level connectivity.

4 Key Findings

We analyze whether sandbagging changes how attention heads are functionally organized. Using a held-out set of $n = 400$ MMLU questions (Hendrycks et al., 2020), we construct one graph per condition. We closely follow the sandbagging protocol and prompts introduced in Hofstätter et al. (2025): the model is instructed to answer correctly in the honest condition and incentivized to give incorrect answers in the sandbagging condition. For each condition $C \in \{\mathrm{H}, \mathrm{S}\}$, we compute head similarities using equation 1 and keep the highest-weight 25% of head pairs as edges. Because both graphs have the same number of edges, any difference between them tells us how connectivity was redistributed across attention heads. All connectivity measurements are made on these graphs.

We then detect communities to study whether heads keep their connections within their own functional group. Detection requires a separate graph, since thresholding leaves some heads with no edges, and such heads form singleton communities that give no information. For detection, we therefore use a symmetric k-nearest-neighbour graph, in which two heads are connected if either ranks the other among its k most similar (von Luxburg, 2007), so that every head has a neighbourhood. We identify the communities once, using the honest condition, and keep them fixed when measuring both conditions on the thresholded graphs. This ensures we compare each head against the same group of heads in both cases, so any drop in within-community connectivity means the head lost connections to its group, rather than the group itself being redefined.

4.1 Sandbagging Reduces Within-Community Connectivity

Table 1: Mean per-head fraction of edges remaining within a head's own community, honest (H) versus sandbagging (S). Here, $\Delta = 100(S - H)$ is reported in percentage points; more negative values indicate a larger loss of within-community retention under sandbagging. Under sandbagging, the within-community retention falls for every model and every community detection method.

	Louvain			**Leiden**			**Infomap**			**Spectral**		
Model	H	S	Δ	H	S	Δ	H	S	Δ	H	S	Δ
Mistral-24B	.337	.254	−8.3	.333	.253	−8.0	.462	.345	−11.7	.388	.312	−7.6
Qwen3-32B	.357	.309	−4.7	.355	.304	−5.2	.634	.589	− 4.5	.512	.465	−4.6
Llama-3.3-70B	.364	.319	−4.4	.389	.343	−4.6	.468	.400	− 6.8	.420	.370	−5.0
GPT-OSS-20B	.394	.354	−4.0	.391	.349	−4.2	.464	.426	− 3.7	.387	.343	−4.4

We compare four widely used community detection methods spanning different objectives and optimization strategies. Louvain (Blondel et al., 2008) and Leiden (Traag et al., 2019) both optimize modularity, but Leiden adds a refinement step that guarantees well-connected communities, which Louvain can miss. Infomap (Rosvall & Bergstrom, 2008) instead follows information flow, grouping nodes among which random walks tend to remain. Spectral clustering (von Luxburg, 2007), on the other hand, uses the geometry of the graph, partitioning nodes by an embedding derived from its Laplacian.

To test whether sandbagging changes how attention heads are organized, we compare within-community connectivity between the honest and sandbagging graphs. For each model and community detection method, we compute the within-community fraction f_i^C, obtained from Equation 2

with $A = \deg_{G^C}(i)$, and average it over all heads under both conditions. As shown in Table 1, this fraction falls under sandbagging in every model and every community detection method. Because the four methods optimize different objectives and the four models differ in architecture, depth, and head count, the reduction cannot be attributed to any single partitioning choice or architecture. Under sandbagging, heads consistently retain a smaller share of their strongest connections within their communities.

To rule out the possibility that communities detected on the honest graph simply fit it best, we construct a second honest graph from a paraphrased prompt and measure it against the same communities. Its within-community fraction changes much less than under sandbagging in every setting, so the reduction is not an artifact of detecting communities on the honest condition. Details are given in Appendix B.

4.2 Head-Level Structural Scores

Motivated by the finding in the previous section that sandbagging reduces within-community connectivity, our goal is to identify the heads that contribute most to that reduction, so that we can perform a localized intervention on them. Given head i in condition $C \in \{H, S\}$, we take its within-community degree $\tilde{k}_i^C$ from equation 2 with $A = M - 1$ and measure how it changes between the two conditions,

$$\Delta_i = \tilde{k}_i^S - \tilde{k}_i^H. \tag{3}$$

More negative values identify heads that lose more within-community connectivity. Given a budget of $k\%$ of all heads, we use Hamilton apportionment to distribute the budget across communities in proportion to their size, so that each community contributes a share of the selection matching its share of the network. Within each community, we rank heads by Δ_i in ascending order and take those with the largest losses, breaking ties by head index. Thus, heads are compared only with their functional peers, while the proportional quotas prevent a single large community from absorbing the entire budget and still select exactly $k\%$ of heads overall.

4.3 Identifying Heads for Improving Capability Elicitation

The score identifies heads that lose within-community connectivity, but not whether intervening on them changes sandbagging behavior and elicits the sandbagged capability. Prior work shows that many attention heads can be disabled without degrading performance (Michel et al., 2019), and that suppressing certain heads can even improve task accuracy (Kovaleva et al., 2019). We build on those findings to study whether ablating a small fraction of the heads identified in Section 4.2 elicits the sandbagged capability. Specifically, let $\mathcal{A}$ denote the selected heads. For each $h \in \mathcal{A}$, we set its pre-W_O output to zero, $z_h \leftarrow \mathbf{0}$, at the final prompt token. All other heads and all model parameters are left unchanged, so the only difference from the unmodified forward pass is the removal of these heads' contributions at a single token position.

Figure 1 demonstrates the effect of this intervention against three controls of equal budget. The global-random control draws heads uniformly from all layers and tests whether ablating any heads would help. Layer-matched random takes the same number of heads from each layer as our selection, so if the intervention were only finding the right layers, any heads from those layers would work just as well. Lastly, degree-matched random matches the degree distribution of our selection, so if the effect came from how connected the heads are, equally connected heads would work just as well. We report the mean over 20 seeded sets per control.

We find that the size of the gain varies by model, which is expected given differences in pretraining and post-training recipes. In all sixteen combinations of four community detection methods and four models, spanning dense and mixture-of-experts architectures of different depths, our selection recovers more suppressed capability than the mean of every control. We also find the cost of this ablation to be small. The largest drop in honest accuracy across all sixteen cells is 1.25 pp, so the intervention increases capability elicitation accuracy while reducing honest accuracy by at most 1.25 pp.

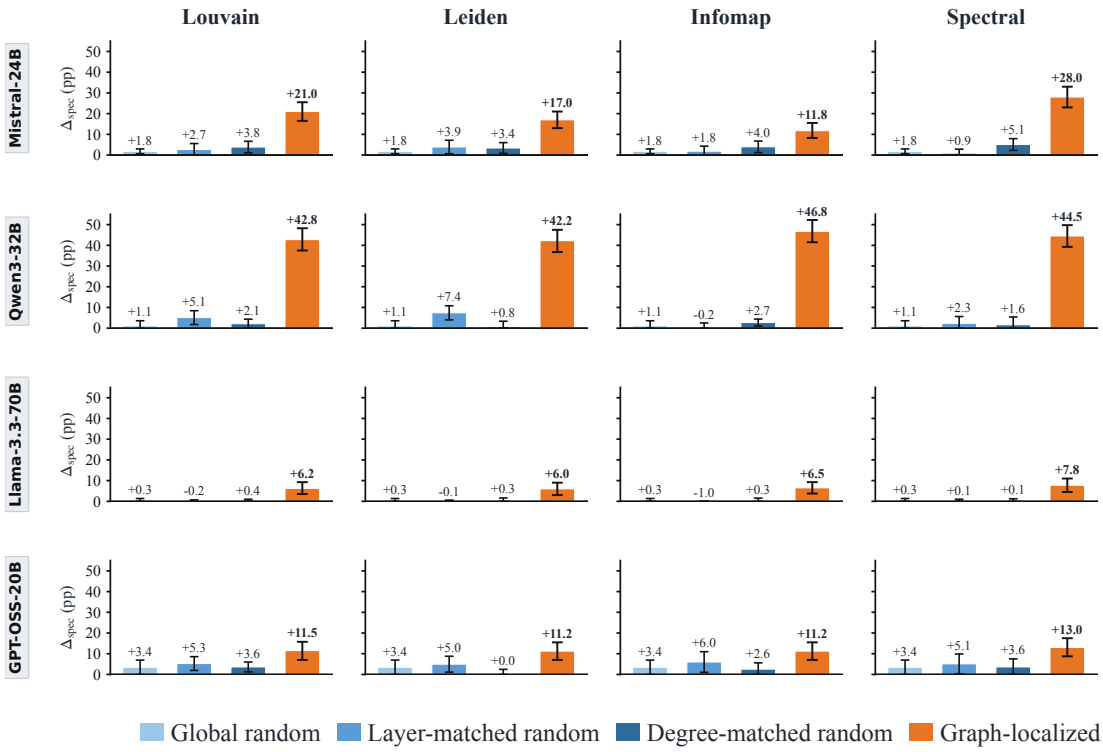

Figure 1: Ablating graph-localized heads recovers suppressed capability across four models on MMLU. Null bars are means over independently seeded global-random, exact-layer-matched, and degree-matched head sets of equal size (20 draws per control). Their whiskers are 95% confidence intervals across head-set draws; whiskers on graph-localized bars are paired-question 95% bootstrap confidence intervals. The graph-localized selection outperforms every control mean under all four community detection algorithms. Across all 16 cells, its worst change in honest accuracy is -1.25 pp.

5 Applying Our Insight to Existing Methods

We extend our method to existing elicitation methods, and demonstrate that we are able to identify heads that, when targeted, reveal intrinsic model capabilities while having minimal affect on honest accuracy.

5.1 Experimental Setup

There are two widely established protocols for studying sandbagging. In the **prompted setting**, the model is given an incentive to underperform and instructed to do so, while its weights remain unchanged. In the **fine-tuned setting**, commonly known as password locking, the model is trained to condition its behavior on a trigger string in the prompt, answering correctly when the trigger is present and underperforming when it is absent. For the prompted setting, we adopt the sandbagging prompts of Hofstätter et al. (2025); Tice et al. (2025). For the fine-tuned setting, we use the two model organisms of Hofstätter et al. (2025), a password-locked organism and a circuit-broken organism (Appendix D.1).

5.2 Baselines and Datasets

Baselines. We group existing elicitation methods into two levels. *Inference-level methods* modify only the input to the model and leave its parameters untouched. We consider few-shot prompting (Brown et al., 2020), fixed to five shots throughout, and response prefill (Hofstätter et al., 2025).

Weight-level methods, on the other hand, modify the parameters. We consider noise injection (Tice et al., 2025), which adds Gaussian noise to all weights. We also consider supervised fine-tuning (Greenblatt et al., 2024b), the most widely adopted approach and a strong one that recovers hidden capability from a small number of correct demonstrations, and we include LoRA as a parameter-efficient alternative to fine-tuning the full model. Further details on each baseline are provided in Appendix D.3.

Datasets. We evaluate on three benchmarks: Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020), the Weapons of Mass Destruction Proxy (WMDP) (Li et al., 2024), and the AI2 Reasoning Challenge (ARC-Challenge) (Clark et al., 2018), described in Appendix D.4. We hold out $n = 400$ MMLU questions for supervision and tuning. Methods that require demonstrations draw them from this set, and every method's hyperparameters are selected and tuned only on it.

5.3 Structural Interventions

Based on our observation that ablating the heads our functional network identifies recovers suppressed capability while leaving honest performance nearly intact, we consider two structural interventions that pair with existing methods at each access level. At the inference level, we ablate the selected heads and apply the baseline method in the same forward pass, giving **Ours + Prefill** and **Ours + Few-shot**. Both compositions require no demonstrations beyond what the baseline already uses and no change to the model's parameters.

At the weight level, we draw on prior work showing that a frozen model can be adapted effectively by learning offsets on a small subset of attention heads (Yin et al., 2024). We adopt the same form of intervention but select the sites differently, based on changes in functional connectivity between the honest and sandbagging conditions rather than on gradient attribution or task-specific importance. Specifically, keeping the base model frozen, for head h with pre-W_O output z_h, we learn head-specific parameters g_h, u_h, and v_h and replace z_h by

$$z'_h = g_h z_h + u_h(v_h^\top z_h). \tag{4}$$

We initialize $g_h = 0$ and $u_h = 0$, which exactly reproduces the ablation in Section 4.3. This allows training to move the intervention away from ablation in the direction the objective favors. We train on the same number of correct demonstrations as full fine-tuning and LoRA, but update only these head-specific parameters, which are orders of magnitude fewer than those updated by either method. Our intervention also requires no search over perturbation magnitudes, as noise injection does. We refer to this intervention as **Ours (GraphLearned)**.

5.4 Results

We test whether our method improves existing elicitation methods at both the inference and weight levels, first under prompted sandbagging and then on fine-tuned model organisms.

Our method strengthens inference-level elicitation. We combine our ablation with prefill and few-shot prompting using the procedure described in Section 5.3. As shown in Table 2, this improves both methods across all models and benchmarks, including cases where the baseline alone degrades performance. Because these compositions leave the model's parameters untouched, the additional recovery comes at no training cost. We also test whether this gain depends on the heads identified by our functional network or whether ablating arbitrary heads would work as well. Repeating the composition with the same number of randomly chosen heads (Table 6), we find that random heads give inconsistent gains and often reduce accuracy below the baseline alone, while our selection improves performance in nearly every case. The benefit therefore comes from where we intervene, not from ablation itself.

At the weight level, our method outperforms fine-tuning and is far more stable. Under prompted sandbagging, Ours (GraphLearned) achieves the highest accuracy across models and benchmarks, surpassing noise injection, LoRA, and full fine-tuning while training only the small set of parameters described in Section 5.3. This advantage holds across demonstration budgets (Table 7). Our method has lower variance than full fine-tuning in every comparison and remains reliable at the smallest budget, where fine-tuning varies widely from one run to the next. On Mistral-24B, our method with four demonstrations exceeds full fine-tuning with sixteen demonstrations.

Our method remains effective when the suppression is trained into the weights. We apply the same interventions to the password-locked and circuit-broken organisms, building the contrast graphs without access to the password. As shown in Table 3, our method improves or preserves every inference-level baseline. At the weight level, OURS (GRAPHLEARNED) achieves the best result on three of the four organisms. The difference from other weight-level baselines is clearer on the password-locked organisms, where fine-tuning and noise injection recover little of the suppressed capability while our method recovers most of it.

Table 2: Capability elicitation (accuracy and % change from the unmodified model). Weight-level results report mean accuracy over five independent trials. Our intervention improves every inference-level baseline, and our method outperforms the weight-level methods while training orders of magnitude fewer parameters.

Model		Method	MMLU		WMDP		ARC	
			Acc. (%)	Δ_{SB}	Acc. (%)	Δ_{SB}	Acc. (%)	Δ_{SB}
Mistral-24B	Inference-level	Prefill	30.75	+6.50	28.25	+3.50	33.75	+17.25
		Ours + Prefill	**36.75**	**+12.50**	**33.75**	**+9.00**	**45.50**	**+29.00**
		Few-shot	15.58	−8.67	19.92	−4.83	6.42	−10.08
		Ours + Few-shot	**19.42**	**−4.83**	**22.58**	**−2.17**	**9.50**	**−7.00**
	Weight-level	Noise Injection	$33.00_{\pm 2.81}$	+8.75	$33.05_{\pm 4.29}$	+8.30	$45.45_{\pm 6.86}$	+28.95
		SFT (LoRA)	$64.85_{\pm 5.23}$	+40.60	$54.10_{\pm 4.19}$	+29.35	$85.90_{\pm 2.81}$	+69.40
		SFT (Full)	$68.05_{\pm 9.43}$	+43.80	$58.60_{\pm 7.50}$	+33.85	$86.10_{\pm 9.86}$	+69.60
		Ours (GraphLearned)	$\mathbf{71.35_{\pm 2.80}}$	**+47.10**	$\mathbf{60.70_{\pm 0.82}}$	**+35.95**	$\mathbf{89.85_{\pm 3.21}}$	**+73.35**
Qwen3-32B	Inference-level	Prefill	67.75	+40.50	60.75	+30.50	89.50	+71.75
		Ours + Prefill	**68.00**	**+40.75**	**62.00**	**+31.75**	**91.25**	**+73.50**
		Few-shot	73.00	+45.75	63.83	+33.58	87.08	+69.33
		Ours + Few-shot	**76.50**	**+49.25**	**68.33**	**+38.08**	**93.58**	**+75.83**
	Weight-level	Noise Injection	$56.45_{\pm 7.77}$	+29.20	$48.95_{\pm 5.12}$	+18.70	$74.70_{\pm 11.66}$	+56.95
		SFT (LoRA)	$57.05_{\pm 10.90}$	+29.80	$48.25_{\pm 8.37}$	+18.00	$74.40_{\pm 18.52}$	+56.65
		SFT (Full)	$63.55_{\pm 19.33}$	+36.30	$54.80_{\pm 14.50}$	+24.55	$76.95_{\pm 32.40}$	+59.20
		Ours (GraphLearned)	$\mathbf{70.25_{\pm 1.33}}$	**+43.00**	$\mathbf{63.60_{\pm 1.10}}$	**+33.35**	$\mathbf{89.10_{\pm 3.18}}$	**+71.35**
Llama-3.3-70B	Inference-level	Prefill	27.75	+11.25	17.00	+2.50	10.75	+4.25
		Ours + Prefill	**31.75**	**+15.25**	**20.50**	**+6.00**	**21.50**	**+15.00**
		Few-shot	20.00	+3.50	22.42	+7.92	12.58	+6.08
		Ours + Few-shot	**47.08**	**+30.58**	**39.00**	**+24.50**	**46.92**	**+40.42**
	Weight-level	Noise Injection	$40.70_{\pm 19.29}$	+24.20	$24.65_{\pm 16.86}$	+10.15	$30.25_{\pm 30.33}$	+23.75
		SFT (LoRA)	$74.45_{\pm 11.63}$	+57.95	$57.85_{\pm 13.22}$	+43.35	$80.40_{\pm 24.57}$	+73.90
		Ours (GraphLearned)	$\mathbf{76.80_{\pm 1.59}}$	**+60.30**	$\mathbf{62.90_{\pm 2.61}}$	**+48.40**	$\mathbf{87.40_{\pm 3.83}}$	**+80.90**

Table 3: Final-evaluation accuracy (%) for capability elicitation in password-locked and circuit-broken Mistral sandbagging model organisms. Weight-level results report mean accuracy over five independent trials. Our intervention improves or preserves every inference-level baseline and remains competitive with weight-level baselines.

Level	Method	Password-locked		Circuit-broken	
		Mistral-base	Mistral-instruct	Mistral-base	Mistral-instruct
Inference	Prefill	57.03	**50.27**	**60.81**	25.41
	Ours + Prefill	**57.57**	**50.27**	**60.81**	**25.68**
	Five-shot	57.84	51.35	26.76	**28.11**
	Ours + Five-shot	**58.38**	**52.70**	**27.03**	**28.11**
Weight	Noise injection	$15.62_{\pm 2.82}$	$15.51_{\pm 1.46}$	$0.00_{\pm 0.00}$	$0.11_{\pm 0.24}$
	SFT (LoRA)	$39.24_{\pm 11.90}$	$21.46_{\pm 11.01}$	$55.19_{\pm 5.24}$	$33.95_{\pm 10.61}$
	SFT (Full)	$28.16_{\pm 17.12}$	$17.78_{\pm 4.22}$	$\mathbf{58.43_{\pm 2.31}}$	$29.41_{\pm 10.72}$
	Ours (GraphLearned)	$\mathbf{52.27_{\pm 1.04}}$	$\mathbf{51.62_{\pm 1.71}}$	$57.08_{\pm 3.22}$	$\mathbf{39.19_{\pm 5.78}}$

6 Related Work

6.1 Graph-based Mechanistic Interpretability

Graph theory has been used to enhance the interpretability of deep neural networks in several different scenarios. Early papers on clusterability fall into two categories: weight graphs and functional graphs. To construct weight graphs, Watanabe et al. (2018); Filan et al. (2021) form modules by analyzing the connectivity between weights in neural networks, while Hod et al. (2021); Lange et al. (2022) generate their functional graphs using activation correlations. While graph theory has also been fundamental to interpreting graph neural networks, graph techniques have recently been extended to other settings, ranging from a reinforcement learning policy system to the learning dynamics of LLMs. In order to elucidate the mechanistic underpinnings of GNNs, El et al. (2025) bridge the message-passing process of GNNs with the self-attention mechanisms in Transformers. While they highlight the possibility of extending this graph-based approach to interpretability for traditional transformer models, they intentionally leave it for future work. Shortly after El et al. (2025), Soligo et al. (2025) used a graph-based approach to understanding reinforcement learning dynamics, with the ultimate goal of ensuring that RL systems align with human values. However, instead of using attention heads and MLP layers as nodes in their graph, Soligo et al. (2025) implemented a Louvain algorithm to generate a lightweight graph with "functional modules" of RL policy networks. In the work most similar to ours, Rocchetti (2025) uses attention heads as a way to explicate the way in which LLMs acquire complex knowledge.

6.2 Sandbagging

While the notion of sandbagging predates AI safety, van der Weij et al. (2025) formalize this concept for capability evaluations in modern AI systems. Subsequent work has studied sandbagging from several complementary perspectives, including its induction, detection, and mitigation. Greenblatt et al. (2024b); Hofstätter et al. (2025); Tice et al. (2025) use prompting and training to develop "password-locked" model organisms, creating controlled environments to study deliberate capability suppression during induction. Research on detecting sandbagging and deception uses behavioral methods like chain-of-thought monitoring, as in Li & Siegel (2025), and white-box activation probes, as in Goldowsky-Dill et al. (2025a) and Rotalabs (2025). However, probe-based detection exhibits model-specific variations and struggles to generalize across different types of deception, as shown by Natarajan et al. (2026). To mitigate sandbagging and recover withheld capabilities, recent work explores elicitation methods using password-locked testbeds from Greenblatt et al. (2024b) and training-based solutions from Tice et al. (2025) and Hofstätter et al. (2025). Despite substantial progress in inducing, detecting, and mitigating sandbagging, existing work has primarily focused on its observable behavior and capability recovery. How sandbagging alters internal representations and computations remains underexplored, highlighting the need for a mechanistic investigation of its structural basis.

7 Conclusion

Existing work on the mechanistic underpinnings of sandbagging is scarce. We bridge this gap by introducing a new lens for studying this phenomenon: functional networks of attention heads that capture interactions between various communities in the model. Across models and community detection methods, we find that sandbagging is associated with a stable and consistent reorganization of these networks, marked by reduced within-community connectivity. Ablating the heads with the largest changes in connectivity recovers suppressed capability while leaving honest accuracy largely intact. Building on this, we designed interventions at both the inference and weight levels that improve existing elicitation methods under prompted sandbagging and on fine-tuned model organisms. While our methods do not establish a concrete causal relationship between graphs and sandbagging tendencies, they provide a way to quantitatively measure multi-head feature interactions. Our findings show that sandbagging leaves a measurable signature in how a model's components coordinate and suggest that functional networks might offer a novel lens for studying other misaligned behaviors in large language models.

8 AI USE STATEMENT

We used LLMs solely as auxiliary tools to improve the grammar and clarity of our manuscript. This assistance was limited to enhancing readability and presentation; all conceptual contributions, analyses, and interpretations were solely developed by the authors.

REFERENCES

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/p10008. URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Xiao Pan Ding, Si Jia Wu, Jiangang Liu, Genyue Fu, and Kang Lee. Functional neural networks of honesty and dishonesty in children: Evidence from graph theory analysis. *Scientific Reports*, 7:12085, 2017. doi: 10.1038/s41598-017-11754-4. URL https://doi.org/10.1038/s41598-017-11754-4.

Batu El, Deepro Choudhury, Pietro Liò, and Chaitanya K Joshi. Towards mechanistic interpretability of graph transformers via attention graphs. *arXiv preprint arXiv:2502.12352*, 2025.

Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. *arXiv preprint arXiv:2103.03386*, 2021.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception with linear probes. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=C5Jj3QKQav.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception using linear probes, 2025b. URL https://arxiv.org/abs/2502.03407.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models, 2024a. URL https://arxiv.org/abs/2412.14093.

Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models. *Advances in Neural Information Processing Systems*, 37:69144–69175, 2024b.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Shlomi Hod, Stephen Casper, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Detecting modularity in deep neural networks. 2021.

Felix Hofstätter, Teun van der Weij, Jayden Teoh, Rada Djoneva, Henning Bartsch, and Francis Rhys Ward. The elicitation game: Evaluating capability elicitation techniques. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23330–23356. PMLR, 2025. URL https://proceedings.mlr.press/v267/hofstatter25a.html.

Weixiong Jiang, Huasheng Liu, Lingli Zeng, Jian Liao, Hui Shen, Aijing Luo, Dewen Hu, and Wei Wang. Decoding the processing of lying using functional connectivity MRI. *Behavioral and Brain Functions*, 11(1):1, January 2015. ISSN 1744-9081. doi: 10.1186/s12993-014-0046-4. URL https://doi.org/10.1186/s12993-014-0046-4.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445/.

Richard D Lange, David S Rolnick, and Konrad P Kording. Clustering units in neural networks: upstream vs downstream information. *arXiv preprint arXiv:2203.11815*, 2022.

Chloe Li and Noah Y Siegel. Llms can covertly sandbag on capability evaluations against chain-of-thought monitoring. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 395–422, 2025.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024. URL https://arxiv.org/abs/2403.03218.

Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier models are capable of in-context scheming, 2024. URL https://arxiv.org/abs/2412.04984. First released December 2024; revised January 2025.

Paul Michel, Omer Levy, and Graham Neubig. Are Sixteen Heads Really Better than One? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.

Vikram Natarajan, Devina Jain, Shivam Arora, Satvik Golechha, and Joseph Bloom. One probe won't catch them all: Towards targeted deception detection, 2026. URL https://arxiv.org/abs/2602.01425.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability.

In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

Elisabetta Rocchetti. Modeling transformers as complex networks to analyze learning dynamics. *arXiv preprint arXiv:2509.15269*, 2025.

Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008. doi: 10.1073/pnas.0706851105.

Rotalabs. Detecting ai sandbagging with activation probes, 2025. URL https://rotalabs.ai/blog/detecting-ai-sandbagging/.

Ali Rahimi Saryazdi, Farnaz Ghassemi, Fatemeh Parastesh, Karthikeyan Rajagopal, and Sajad Jafari. From truth to deception: a network control theory perspective on brain connectivity and cognitive dynamics. *Brain Research*, pp. 150159, 2026.

Anna Soligo, Pietro Ferraro, and David Boyle. Inducing, detecting and characterising neural modules: A pipeline for functional interpretability in reinforcement learning. *arXiv preprint arXiv:2501.17077*, 2025.

Jordan Taylor, Sid Black, Dillon Bowen, Thomas Read, Satvik Golechha, Alex Zelenka-Martin, Oliver Makins, Connor Kissane, Kola Ayonrinde, Jacob Merizian, Samuel Marks, Chris Cundy, and Joseph Bloom. Auditing games for sandbagging, 2025. URL https://arxiv.org/abs/2512.07810.

Cameron Tice, Philipp Kreer, Nathan Helm-Burger, Prithviraj Shahani, Fedor Ryzhenkov, Fabien Roger, Clement Neo, Jacob Haimes, Felix Hofstätter, and Teun van der Weij. Noise injection reveals hidden capabilities of sandbagging language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/aa95d031f5e151bf8fb402f44bd97054-Abstract-Conference.html.

V. A. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), March 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-41695-z. URL http://dx.doi.org/10.1038/s41598-019-41695-z.

Teun van der Weij, Felix Hofstätter, Oliver Jaffe, Samuel Brown, and Francis Ward. AI sandbagging: Language models can strategically underperform on evaluations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b5e5753b0a0e440a6d8dc7e143617cec-Abstract-Conference.html.

Ulrike von Luxburg. A tutorial on spectral clustering, 2007. URL https://arxiv.org/abs/0711.0189.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Understanding community structure in layered neural networks. *arXiv preprint arXiv:1804.04778*, 2018.

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations, 2024. URL https://arxiv.org/abs/2406.01563.

A THE SVCCA-MI EDGE

A.1 SVCCA

Singular vector canonical correlation analysis (SVCCA) is a statistical method for comparing two collections of vector-valued observations (Raghu et al., 2017) and is often used to measure the alignment or similarity between internal features and representations. SVCCA scores range from 0 to 1, where a score of 1 indicates identical learned representations. Let $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$ contain measurements from the same n observations, with columns corresponding to measured variables. SVCCA assesses how strongly these measurements vary together without requiring individual variables in X and Y to correspond. It first applies singular value decomposition to the centered data, retaining the dominant directions of variation in each matrix. It then applies canonical correlation analysis to find pairs of linear combinations with maximal correlation. Successive pairs are constrained to be uncorrelated with earlier pairs within each dataset. This produces canonical correlations $\rho_1, \ldots, \rho_K \in [0, 1]$, where K is the smaller retained dimension. Each ρ_r measures the strength of one linear relationship between a pair of internal features or representations.

A.2 THE GAUSSIAN ASSUMPTION

Equation (1) is exact only under joint Gaussianity. Because attention activations deviate from this ideal, we quantify the extent of non-Gaussianity and evaluate its impact on the graph edge weights.

Setup. All validation checks evaluate query activations with $K = 8$ retained directions on Qwen3-0.6B, Qwen3-32B, and Pythia-1.4b, matching the model scale of the core experiments in Section 4.

Marginal Shape. The reduced features exhibit near-Gaussian behavior in the bulk, consistent with SVD acting as a coordinate average. Deviations are primarily confined to the tails: median excess kurtosis remains low (0.048–0.133), with only 11.4%–17.5% of features exceeding a kurtosis of 1 (Figure 2).

Robustness. To determine whether tail non-Gaussianity materially affects edge formulation, we apply a rank transform to enforce exact marginal Gaussianity and recompute all edge weights. Rank agreement with the untransformed edges remains near-perfect, with negligible absolute changes in weight magnitude. This confirms that the derived edges depend fundamentally on the reduced features and are robust to marginal tail deviations.

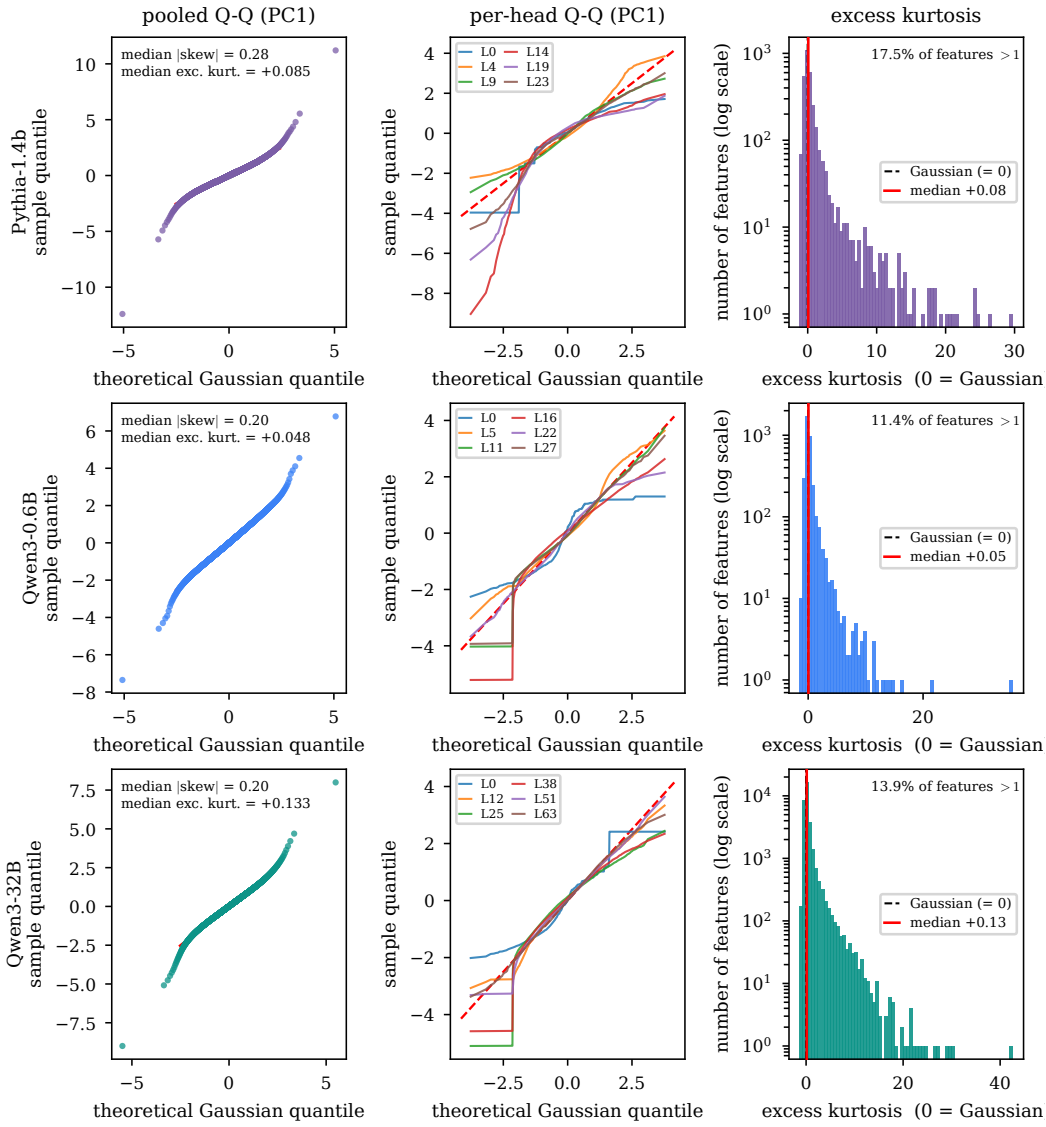

Figure 2: Marginal Gaussianity of the reduced head features across models. Left: pooled Q-Q plot of the leading score. Middle: per-head Q-Q plots at various depths. Right: distribution of per-feature excess kurtosis. The distributions demonstrate that the bulk is near-Gaussian, with deviations largely confined to a small percentage of features in the tails.

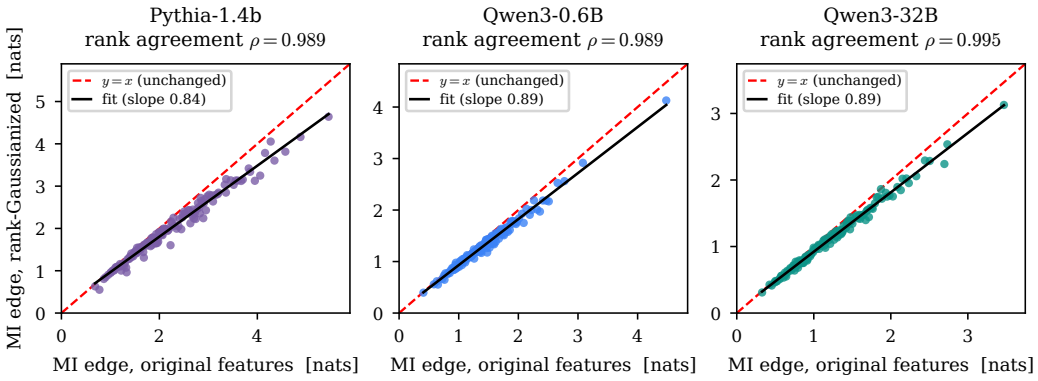

Figure 3: Gaussian nonparanormal transfer robustness. Recomputing edges after rank-transforming marginals to be exactly Gaussian leaves the ranking and weights essentially unchanged ($\rho \in [0.989, 0.995]$). This confirms that the edge formulation is robust to tail deviations.

B COMMUNITY DETECTION

The analysis in Table 1 detects communities once on the honest graph and holds them fixed when measuring both conditions. This design is necessary for comparing a head against the same group in both cases, but it raises a concern. Community detection finds the partition that best fits the graph it is run on, so the honest partition naturally fits the honest graph better than any other. Measured against that partition, any other graph might show lower within-community connectivity simply because the partition was not fitted to it. If so, the drop we report under sandbagging could reflect this mismatch rather than anything specific to sandbagging.

To separate these two explanations, we construct a second honest graph H_2 from a paraphrased version of the honest prompt. The model still answers honestly, but the wording of the prompt differs, so H_2 is a different graph that faces exactly the same mismatch with the honest partition as the sandbagging graph does, without any change in behavior. The two explanations then make opposite predictions. If the drop comes from fitting the partition to the honest graph, H_2 should show a decrease comparable to sandbagging. If the drop comes from sandbagging itself, H_2 should stay close to H.

Table 4 shows that H_2 stays close to H. The within-community fraction changes by at most 1.9 percentage points across all models and community detection methods, compared with decreases of 3.7 to 11.7 points under sandbagging, and in every cell the change under paraphrasing is smaller than the corresponding drop under sandbagging. On GPT-OSS-20B, within-community connectivity even increases under paraphrasing for all four methods, which a fitting artifact cannot produce. Changing the prompt without changing the behavior therefore leaves the community structure largely intact, and the reduction we observe under sandbagging cannot be attributed to detecting communities on the honest condition.

Table 4: Mean per-head fraction of edges remaining within a head's fixed Table 1 community, original honest prompt (H) versus honest paraphrase (H_2). Here, $\Delta = 100(H_2 - H)$ is reported in percentage points. The absolute change is at most 1.9 points across all models and community detection methods.

	Louvain			**Leiden**			**Infomap**			**Spectral**		
Model	H	H_2	Δ	H	H_2	Δ	H	H_2	Δ	H	H_2	Δ
Mistral-24B	.337	.336	$-$ 0.1	.333	.330	$-$ 0.3	.462	.456	$-$ 0.6	.388	.387	$-$ 0.2
Qwen3-32B	.357	.340	$-$ 1.6	.355	.339	$-$ 1.7	.634	.615	$-$ 1.9	.512	.493	$-$ 1.9
Llama-3.3-70B	.364	.362	$-$ 0.2	.389	.388	$-$ 0.1	.468	.465	$-$ 0.3	.420	.419	$-$ 0.1
GPT-OSS-20B	.394	.400	$+0.5$	.391	.398	$+0.7$	.464	.471	$+0.8$	.387	.390	$+0.3$

C Sensitivity to Graph Density

Our graphs retain the top 25% of head pairs. To test whether our results depend on this choice, we re-threshold the honest and sandbagging similarity matrices at densities from 0.15 to 0.35, keeping the honest-fitted Leiden communities fixed, and recompute the head scores. We then measure how closely the ranking of heads at each density agrees with the ranking at 0.25.

As shown in Table 5, head-score rankings remain strongly correlated with those obtained at 0.25 across all tested densities. Thus, the overall ranking signal is not specific to the density used in the main experiments.

Table 5: Spearman correlation between head scores at each density and at the 0.25 density used throughout the paper. Rankings remain strongly correlated across all densities and models.

Density	**Mistral-24B**	**Qwen3-32B**	**Llama-3.3-70B**
0.15	0.828	0.878	0.834
0.20	0.919	0.950	0.934
0.30	0.935	0.960	0.939
0.35	0.872	0.909	0.853

D Experimental Details

D.1 Model Organisms

The first, a password-locked organism, is trained with supervised fine-tuning to answer correctly when the password is present and to give incorrect answers when it is absent. The second, a circuit-broken organism, applies circuit breaking on top of the password-locked model, so that its internal representations, rather than only its output tokens, are trained to differ when the password is absent, under the assumption that this makes the organism more robust and harder to break with elicitation methods.

D.2 Contrast Graphs for Model Organisms

Our method requires a pair of contrast graphs, one from an honest condition and one from a sandbagging condition. In the prompted setting, these come directly from the two prompts. For the model organisms, the natural honest condition would be to supply the password, but this assumes the evaluator already knows it, and an evaluator who knows the password has no need for elicitation. We therefore build the contrast without the password. The locked model with no password serves as the sandbagging condition, and the same model with a response prefill (defined in Section 5.2) serves as a proxy for the honest condition. We use prefill because it requires no demonstrations or training and partially recovers the suppressed capability, giving a simple honest signal from which to detect candidate heads.

D.3 Baselines

In few-shot prompting (Brown et al., 2020), correctly answered examples are prepended to the prompt so that the capability is elicited in context. In response prefill (Hofstätter et al., 2025), a short phrase committing the model to state the correct answer is appended to its turn, on the premise that a partially written response can override the instruction to underperform. Noise injection (Tice et al., 2025) adds Gaussian noise to all weights with the intuition that random perturbation degrades a sandbagging policy faster than the underlying capability, so an improvement under noise might itself be evidence of suppression.

D.4 Datasets

We describe the main datasets used throughout this paper. MMLU (Hendrycks et al., 2020) tests world knowledge and problem solving across STEM, humanities, and social sciences. WMDP (Li

et al., 2024) tests hazardous knowledge relevant to biosecurity, chemical weapons, and cybercrime, and is therefore the benchmark most directly tied to the threat model that motivates capability evaluations. ARC-Challenge (Clark et al., 2018) tests grade-school science reasoning and provides a third benchmark with a different question distribution.

E LOCALIZATION

Section 4.3 establishes that ablating the heads identified by our functional graph recovers suppressed capability better than ablating random heads. It remains unclear, however, whether this still holds once ablation is combined with other elicitation methods. Prefill and few-shot prompting already change how the model processes its input, so after combining with them, ablating any heads might help, or the choice of heads might no longer matter.

To test this, we repeat the inference-level compositions of Table 2, replacing our selected heads with the same number of randomly chosen heads, and average the results over five independent random draws. If our selection mattered only for ablation alone, random heads should perform comparably once combined with prefill or few-shot prompting.

Table 6 shows that they do not. Random heads give inconsistent gains and often reduce accuracy below the baseline alone, while our selection improves performance in nearly every case. The benefit of our method therefore comes from where we intervene, even when it is combined with existing elicitation methods.

Table 6: Localization controls under prompted sandbagging. Values are final-evaluation accuracy (%). Random-head results report the mean and sample standard deviation over five random-head draws with the same intervention budget as ours. Interventions on random heads yield inconsistent gains and in some cases reduce accuracy. This shows how our method identifies attention heads that enable targeted intervention.

Model	Method	MMLU	WMDP	ARC
Mistral-24B	Prefill	30.75	28.25	33.75
	Random heads + Prefill	$30.55_{\pm 1.73}$	$28.05_{\pm 0.99}$	$34.75_{\pm 2.86}$
	Ours + Prefill	**36.75**	**33.75**	**45.50**
	Five-shot	15.58	19.92	6.42
	Random heads + Five-shot	$16.10_{\pm 0.64}$	$20.45_{\pm 0.94}$	$7.17_{\pm 0.86}$
	Ours + Five-shot	**19.42**	**22.58**	**9.50**
Qwen3-32B	Prefill	67.75	60.75	89.50
	Random heads + Prefill	$68.15_{\pm 2.65}$	$60.30_{\pm 1.81}$	$88.65_{\pm 1.36}$
	Ours + Prefill	**68.00**	**62.00**	**91.25**
	Five-shot	73.00	63.83	87.08
	Random heads + Five-shot	$73.98_{\pm 1.50}$	$65.20_{\pm 1.69}$	$88.55_{\pm 3.07}$
	Ours + Five-shot	**76.50**	**68.33**	**93.58**
Llama-3.3-70B	Prefill	27.75	17.00	10.75
	Random heads + Prefill	$25.45_{\pm 3.06}$	$16.35_{\pm 1.38}$	$10.25_{\pm 0.94}$
	Ours + Prefill	**31.75**	**20.50**	**21.50**
	Five-shot	20.00	22.42	12.58
	Random heads + Five-shot	$24.70_{\pm 4.61}$	$25.53_{\pm 3.42}$	$17.92_{\pm 5.21}$
	Ours + Five-shot	**47.08**	**39.00**	**46.92**

F COMPARISON ACROSS DEMONSTRATIONS

The weight-level results in Table 2 are obtained with four demonstrations. Since full fine-tuning updates far more parameters, it may benefit more from additional data. We therefore examine whether our advantage persists as the demonstration budget increases, comparing OURS (GRAPHLEARNED) with full fine-tuning at four, eight, and sixteen demonstrations over five independent trials.

As shown in Table 7, our method achieves higher accuracy in nearly all comparisons and lower variance in every one. The variance of full fine-tuning decreases with more demonstrations, whereas our method is already stable at the smallest budget. On Mistral-24B, our method with four demonstrations exceeds full fine-tuning with sixteen demonstrations. These results indicate that our method recovers the suppressed capability reliably with fewer demonstrations and remains competitive as more become available.

Table 7: Comparison across demonstration budgets. Values are final-evaluation accuracy (%), reported as mean and sample standard deviation over five independent trials. Bold marks the higher mean within each matched model, demonstration budget (m), and benchmark comparison. Our method outperforms Full-SFT in nearly all scenarios across both lower and higher demonstration budgets and has lower variance in every comparison.

Model	m	Method	MMLU	WMDP	ARC
Mistral-24B	4	Full-SFT	$68.05_{\pm 9.43}$	$58.60_{\pm 7.50}$	$86.10_{\pm 9.86}$
		GraphLearned	$\mathbf{71.35_{\pm 2.80}}$	$\mathbf{60.70_{\pm 0.82}}$	$\mathbf{89.85_{\pm 3.21}}$
	8	Full-SFT	$69.85_{\pm 1.83}$	$59.10_{\pm 2.56}$	$89.35_{\pm 1.76}$
		GraphLearned	$\mathbf{72.20_{\pm 0.74}}$	$\mathbf{61.25_{\pm 0.98}}$	$\mathbf{91.30_{\pm 0.48}}$
	16	Full-SFT	$67.90_{\pm 3.34}$	$60.30_{\pm 4.08}$	$89.45_{\pm 2.00}$
		GraphLearned	$\mathbf{72.70_{\pm 1.05}}$	$\mathbf{62.00_{\pm 0.68}}$	$\mathbf{91.45_{\pm 0.21}}$
Qwen3-32B	4	Full-SFT	$63.55_{\pm 19.33}$	$54.80_{\pm 14.50}$	$76.95_{\pm 32.40}$
		GraphLearned	$\mathbf{70.25_{\pm 1.33}}$	$\mathbf{63.60_{\pm 1.10}}$	$\mathbf{89.10_{\pm 3.18}}$
	8	Full-SFT	$66.00_{\pm 13.09}$	$59.50_{\pm 6.04}$	$79.35_{\pm 24.97}$
		GraphLearned	$\mathbf{69.75_{\pm 2.68}}$	$\mathbf{64.05_{\pm 0.93}}$	$\mathbf{88.90_{\pm 4.26}}$
	16	Full-SFT	$\mathbf{73.75_{\pm 2.01}}$	$64.05_{\pm 0.86}$	$91.80_{\pm 0.97}$
		GraphLearned	$71.45_{\pm 0.78}$	$\mathbf{64.90_{\pm 0.60}}$	$\mathbf{92.00_{\pm 0.64}}$