LOOKING INSIDE LLMS: SMALL-WORLD CONNECTIVITY AS A SIGNATURE OF REASONING PERFORMANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding large language model (LLM) reasoning requires looking beyond behavioral performance to examine how reasoning ability is reflected in internal organization. Inspired by neuroscience findings linking higher intelligence to stronger small-world organization in functional brain networks, we investigate small-world connectivity as a structural signature of LLM reasoning. We construct functional graphs from attention-head activation similarities and find that a higher small-world index (SWI), capturing local clustering and short global paths, consistently correlates with better fluid reasoning performance across models and training checkpoints. Since local clustering is central to small-world organization, we further examine how heads important for model performance connect within and across communities. We find that these heads tend to have a larger share of connection weight within their own communities (high core scores) and a more concentrated weight distribution across communities (low bridge scores). These observations motivate the hypothesis that high core and low bridge scores serve as structural indicators of head importance for reasoning capability. We validate this hypothesis through pruning, introducing Small-World Allocation (SWA), a hierarchical sparsity allocation method guided by these scores. Across six LLMs, SWA better preserves small-world organization and model performance than competing allocation strategies, reducing WikiText perplexity by up to 20%. Together, these findings identify small-world functional connectivity as a measurable signature of LLM reasoning performance, offering a structural perspective that complements behavioral evaluation.

1 INTRODUCTION

Large language models (LLMs) demonstrate increasingly strong performance on reasoning tasks, including mathematical problem solving, logical inference, and reasoning about code (Leon, 2025; Liu et al., 2025a; Wang et al., 2024). Behavioral benchmarks quantify success on these tasks, but understanding these capabilities also calls for examining how differences in reasoning performance are reflected inside the model. This motivates a central question: Can we identify measurable patterns of internal organization associated with stronger LLM reasoning performance?

Neuroscience offers a precedent for relating intelligence to internal network organization. In functional brain networks, defined by statistical dependencies between regional activities, stronger small-world organization has been associated with higher intelligence (Langer et al., 2012; Barbey, 2018). Such organization features high local clustering while maintaining short network paths (Latora & Marchiori, 2001; Bassett & Bullmore, 2017). This association motivates us to ask whether analogous functional organization in LLMs provides a structural signature of reasoning performance.

Examining this possibility requires representing relationships expressed through model activity. Existing structural analyses provide complementary but distinct perspectives. Architecture-level graphs relate prescribed connectivity to generalization in vision models (You et al., 2020), but primarily reflect model design rather than functional relationships shaped by training. Learned weight spectra capture training-dependent properties associated with generalization and pruning sensitivity (Martin et al., 2021; Hu et al., 2025; Lu et al., 2024), but characterize weight statistics rather than relationships among component activities. We therefore construct functional graphs with attention heads as nodes and activation similarities as edge weights, capturing relationships expressed in a

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.

trained model's responses to inputs. Using this representation, we first examine whether small-world organization is associated with reasoning performance at the network level. Across models and training checkpoints, we find that a higher small-world index (SWI) consistently correlates with better fluid reasoning performance (Figure 2 & Figure 3).

To better understand this network-level association, we examine whether attention heads important for model performance exhibit distinctive connectivity patterns within and across communities, motivated by the local clustering characteristic of small-world organization. We estimate head importance following (Michel et al., 2019) and find that more important heads tend to have a larger share of connection weight within their own communities (high core scores) and a more concentrated distribution of weight across communities (low bridge scores; Section 4.1). These observations motivate the hypothesis that high core and low bridge scores serve as structural indicators of head importance for the reasoning ability of LLMs. We validate this hypothesis through pruning: if these scores are informative about head importance, sparsity allocation guided by them should better retain models' reasoning performance. We examine this prediction through Small-World Allocation (SWA), a hierarchical sparsity allocation method guided by core and bridge scores. Across six LLMs, SWA better retains model performance than competing allocation strategies, reducing WikiText perplexity by up to 20%, while also better preserving small-world organization. These results provide additional evidence that the identified head-level patterns are informative about model performance. Together, our findings characterize small-world functional connectivity as a measurable signature of LLM reasoning and offer a structural perspective that complements behavioral evaluation. Our contributions are summarized as follows:

- **New Observation:** To our knowledge, we are the first to show that LLMs' attention head activations exhibit a small-world functional organization. We further show that SWI is correlated with performance on fluid intelligence benchmarks and remains consistent across different LLM architectures.
- **Novel Findings:** Through head importance analysis, we find that heads with low bridge scores and high core scores are significantly more prevalent among heads with greater importance for model performance. This links local connectivity patterns to head importance and motivates the hypothesis that preferentially preserving heads with these structural properties can better retain reasoning performance.
- **New Method:** We propose Small-World Allocation (SWA), a hierarchical sparsity allocation method that uses core and bridge scores to guide pruning across layers and attention heads. Experiments show that SWA better preserves small-world organization under pruning and improves language modeling performance by up to 20% over state-of-the-art baselines.

2 PRELIMINARIES

2.1 SMALL-WORLD ORGANIZATION OF BRAIN NETWORKS

Small-world organization describes a network topology characterized by high local clustering and efficient global communication (Barbey, 2018; Rosenberg et al., 2020). In neuroscience, functional brain networks are commonly constructed over anatomical regions of interest (ROIs), with nodes representing brain regions and edges representing functional connectivity (Langer et al., 2012). Stronger small-world structure has been associated with higher human intelligence: higher scores on Raven's Advanced Progressive Matrices (Bors & Vigneau, 2003; Raven, 1962) are linked to higher clustering coefficients and shorter communication path lengths (Langer et al., 2012).

More generally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set, let $C(\mathcal{G})$ denote its average clustering coefficient (Watts & Strogatz, 1998), defined as

$$C(\mathcal{G}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} C_i, \qquad C_i = \frac{2e_i}{d_i(d_i - 1)}, \tag{1}$$

where d_i is the degree of node i, e_i is the number of edges among the neighbors of node i, and we set $C_i = 0$ when $d_i < 2$. We let $E(\mathcal{G})$ denote its global efficiency, which captures efficient

communication through short paths (Latora & Marchiori, 2001):

$$E(\mathcal{G}) = \frac{1}{|\mathcal{V}|(|\mathcal{V}|-1)} \sum_{\substack{i,j\in\mathcal{V} \\ i\neq j}} \frac{1}{\mathrm{dist}(i,j)}, \tag{2}$$

where $\mathrm{dist}(i,j)$ is the shortest-path distance between nodes i and j, and disconnected node pairs contribute zero efficiency. To assess small-world organization, $\mathcal{G}$ is compared against degree-preserving randomized graphs generated by edge swaps (Hosseini & Kesler, 2013). Let C_{rand} and E_{rand} denote the average clustering coefficient and global efficiency of these randomized graphs. The small-world index of $\mathcal{G}$ is then defined as

$$\mathrm{SWI}(\mathcal{G}) = \frac{C(\mathcal{G})}{C_{\mathrm{rand}}} \cdot \frac{E(\mathcal{G})}{E_{\mathrm{rand}}}. \tag{3}$$

A larger SWI indicates stronger local clustering and more efficient global communication relative to degree-matched random graphs.

2.2 Representation Similarity

Singular Vector Canonical Correlation Analysis (SVCCA) Raghu et al. (2017) is used to measure similarity between vector-valued representations. Given two mean-centered representation matrices $\mathbf{X} \in \mathbb{R}^{m\times D_x}$ and $\mathbf{Y} \in \mathbb{R}^{m\times D_y}$ over the same m examples, SVCCA first applies singular value decomposition:

$$\mathbf{X} = \mathbf{U}_X \mathbf{S}_X \mathbf{V}_X^\top, \qquad \mathbf{Y} = \mathbf{U}_Y \mathbf{S}_Y \mathbf{V}_Y^\top. \tag{4}$$

It then retains the top singular directions explaining a high proportion of variance, e.g., 99%. Let p_x and p_y be the numbers of retained directions for $\mathbf{X}$ and $\mathbf{Y}$, respectively. The representations are then projected as

$$\mathbf{X}' = \mathbf{X}\mathbf{V}_X^{(p_x)}, \qquad \mathbf{Y}' = \mathbf{Y}\mathbf{V}_Y^{(p_y)}, \tag{5}$$

where $\mathbf{X}' \in \mathbb{R}^{m\times p_x}$ and $\mathbf{Y}' \in \mathbb{R}^{m\times p_y}$. Here, $\mathbf{V}_X^{(p_x)}$ and $\mathbf{V}_Y^{(p_y)}$ denote the matrices formed by the first p_x and p_y columns of $\mathbf{V}_X$ and $\mathbf{V}_Y$, respectively.

Canonical Correlation Analysis (CCA) (Thompson, 2000) then finds projection vectors $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projected representations: $\rho = \max_{\mathbf{w}_x,\mathbf{w}_y} \frac{\mathbf{w}_x^\top \mathbf{\Sigma}_{X'Y'} \mathbf{w}_y}{\sqrt{(\mathbf{w}_x^\top \mathbf{\Sigma}_{X'X'} \mathbf{w}_x)(\mathbf{w}_y^\top \mathbf{\Sigma}_{Y'Y'} \mathbf{w}_y)}}$. Here, $\mathbf{\Sigma}_{X'X'}$ and $\mathbf{\Sigma}_{Y'Y'}$ denote the covariance matrices of $\mathbf{X}'$ and $\mathbf{Y}'$, and $\mathbf{\Sigma}_{X'Y'}$ denotes their cross-covariance matrix. This produces a set of canonical correlations $\{\rho_q\}_{q=1}^r$, where $r = \min(p_x, p_y)$. The SVCCA similarity is defined as their mean:

$$\mathrm{SVCCA}(\mathbf{X},\mathbf{Y}) = \frac{1}{r}\sum_{q=1}^{r} \rho_q. \tag{6}$$

Higher SVCCA values indicate stronger alignment between representational subspaces. In this work, we use a variant of SVCCA that replaces mean aggregation with the sum of Fisher z-transformed canonical correlations, as detailed in Equation (7).

3 Key Findings

In this section, we present our framework for constructing functional graphs of attention heads and quantifying their small-world organization. We first show how we construct an attention-head functional graph and compute its SWI (Section 3.1). Second, we show that SWI is associated with LLM performance on fluid intelligence benchmarks (Section 3.2). The process is shown in Figure 1.

3.1 Constructing Functional Graphs of Attention Heads

We construct attention-head functional graphs by adapting the standard procedure used to build functional brain networks. In brain networks, anatomical regions of interest (ROIs) serve as nodes,

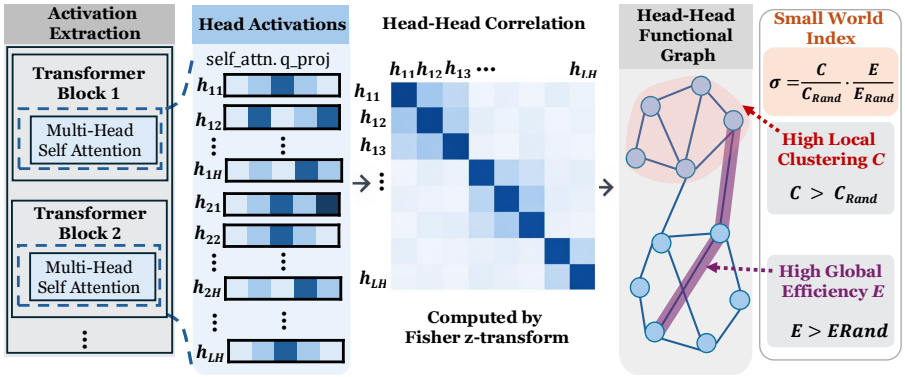


Figure 1: We extract activations from attention heads across Transformer layers and compute pairwise similarity to construct a head-level functional graph. The resulting graph exhibits small-world structure, characterized by high clustering and efficient short-path communication. We show that this structural property correlates with model capability on fluid intelligence benchmarks.

while edges capture statistical dependencies between their activity time series; thresholding the resulting connectivity matrix produces the functional graph. Following the same construction, we treat attention heads as nodes and define their functional connectivity using activation-based similarity. We then threshold the resulting head-similarity matrix to obtain the attention-head functional graph.

Activation Collection. We consider a transformer-based LLM with L layers and H attention heads per layer, yielding $M = LH$ heads in total. We denote the set of all attention heads by $\mathcal{H} = \{h_{\ell,a} \mid \ell = 1, \ldots, L,\ a = 1, \ldots, H\}$, where $h_{\ell,a}$ denotes the a-th attention head in layer ℓ. We sample N questions $\{x^{(n)}\}_{n=1}^{N}$ from a given dataset (e.g., DRE-Bench (Yang et al., 2025b)) for graph construction. These questions are excluded from downstream evaluations. For each question $x^{(n)}$, let T_n denote the number of tokens generated in its answer. We record the query activations of all self-attention heads at each generated answer token. This yields an activation tensor: $\mathbf{A}^{(n)} \in \mathbb{R}^{M \times T_n \times d}$, where d is the head dimension and $\mathbf{A}^{(n)}_{m,t,:}$ denotes the query activation of head m at the t-th generated answer token. For each head and question, we average its token-level activations over the generated answer tokens: $\mathbf{z}_m^{(n)} = \frac{1}{T_n} \sum_{t=1}^{T_n} \mathbf{A}^{(n)}_{m,t,:}$, $\mathbf{z}_m^{(n)} \in \mathbb{R}^d$. Stacking these vectors over all N questions gives the activation matrix for head m: $\mathbf{X}_m = \left[\mathbf{z}_m^{(1)}; \mathbf{z}_m^{(2)}; \ldots; \mathbf{z}_m^{(N)}\right] \in \mathbb{R}^{N \times d}$. Each row of $\mathbf{X}_m$ corresponds to the query activation of head m for one question.

Head Similarity and Graph Construction. In functional brain networks, connectivity is often estimated using Pearson correlation between scalar ROI activity time series (Biswal et al., 1995; Fox et al., 2005). Attention heads instead produce vector-valued activations, so we use a variant of SVCCA that Fisher (z)-transforms and aggregates canonical correlations to quantify functional similarity between heads. Further justification is provided in Appendix A.1.

We treat the activation matrix $\mathbf{X}_m$ of each head as its representation across the sampled questions. For each pair of distinct heads (i, j), we apply the SVD and CCA procedure described in Section 2.2 to obtain canonical correlations $\{\rho_q\}_{q=1}^{r}$. We define their similarity by summing the transformed coefficients:

$$S_{ij} = S(\mathbf{X}_i, \mathbf{X}_j) = \sum_{q=1}^{r} \operatorname{atanh}(\tilde{\rho}_q) = \frac{1}{2} \sum_{q=1}^{r} \log\left(\frac{1 + \tilde{\rho}_q}{1 - \tilde{\rho}_q}\right). \tag{7}$$

Here, $\tilde{\rho}_q = \min(\rho_q, 1 - \epsilon_{\text{clip}})$, where $\epsilon_{\text{clip}} \in (0, 1)$ is a small constant that prevents divergence at $\rho_q = 1$. The Fisher z-transform emphasizes strongly correlated canonical directions. This yields a symmetric head-similarity matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$, with $\mathbf{S}_{ii} = 1$. A larger $\mathbf{S}_{ij}$ indicates greater aggregate covariation between the two heads' query representations across the same questions, reflecting more closely aligned activation patterns in their retained subspaces.

We then construct an undirected functional graph by retaining the strongest pairwise similarities at a fixed edge density. Let $\mathcal{P}$ denote the set of all unordered pairs of distinct heads. For a target

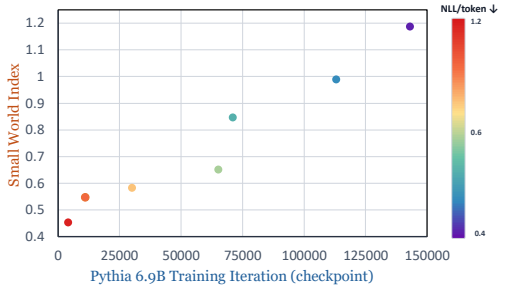


Figure 2: Across Pythia-6.9B training checkpoints (x-axis), a higher small-world index (SWI; y-axis) is consistently associated with better fluid reasoning performance on DRE-Bench (Yang et al., 2025b), as reflected by lower negative log-likelihood (NLL) per answer token (color).

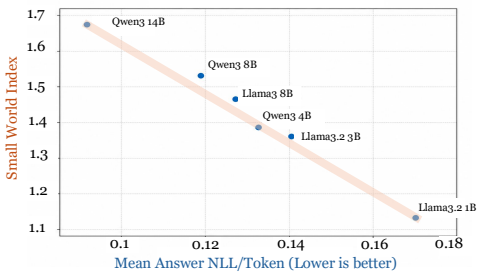


Figure 3: Across different LLMs, a higher small-world index (SWI; y-axis) is consistently associated with better fluid reasoning performance, measured by lower negative log-likelihood (NLL) per answer token on DRE-Bench (Yang et al., 2025b) (x-axis).

edge density δ, we rank these pairs by S_{ij} and retain the top $K_\delta = \lfloor \delta |\mathcal{P}| \rfloor$ pairs as edges. The resulting graph is $\mathcal{G}_\delta = (\mathcal{V}, \mathcal{E}_\delta)$, where $\mathcal{V} = \{1, \ldots, M\}$ indexes the attention heads and $\mathcal{E}_\delta$ contains the retained pairs.

3.2 Connection of Small-world Index to Fluid Intelligence

We find that **stronger small-world organization in head–head functional graphs is associated with better fluid reasoning performance**, both across training checkpoints of the same model and across different models. For each model or checkpoint, we compute the SWI of $\mathcal{G}_\delta$ constructed on DRE-Bench using Equation (3). We measure performance using the mean per-token negative log-likelihood over the answer tokens (NLL/token), where lower values indicate better performance.

As shown in Figure 2, later checkpoints of Pythia-6.9B generally exhibit higher SWI and lower NLL than earlier checkpoints. Similarly, Figure 3 shows that models with lower NLL consistently exhibit higher SWI. Together, these observations suggest that the SWI–performance association holds both within a model over training and across models. The same trend can be observed in graphs constructed using MMLU (Hendrycks et al., 2020). Experimental details are provided in Section 5.1, with detailed numerical results in Appendix A.3.

4 Exploring the SWI–Performance Association through Local Structure

Building on the association between SWI and fluid reasoning performance observed in Section 3.2, we explore local connectivity patterns that help explain this association. We first identify attention heads that are important for model performance and examine their local connectivity patterns (Section 4.1). Our findings motivate the hypothesis that strong within-community connectivity and a concentrated distribution of connection weight across communities play an important role in supporting LLM reasoning performance. To test this hypothesis, we use pruning to examine whether retaining heads with these properties better preserves reasoning performance (Section 4.2).

4.1 Local Connectivity Patterns of Important Heads

In this section, we examine whether heads that are important for model performance exhibit local connectivity patterns. We estimate head importance, characterize connectivity using core and bridge scores, and compare these structural properties between important heads and the remaining heads.

Following Michel et al. (2019), we scale each head's output by a scalar multiplier g_i and evaluate gradients with all multipliers set to one. Head importance is the absolute derivative of the mean answer-token negative log-likelihood with respect to g_i, averaged across test examples.

To characterize local connectivity, we apply Louvain community detection (Blondel et al., 2008) to $\mathcal{G}_\delta$ to obtain communities $\mathcal{C}_\delta = \{C_1, \ldots, C_K\}$. Let $C(i)$ denote the community containing head

Table 1: Prevalence of low bridge scores (b) and high core scores (c) in Llama-3.2-3B on DRE-Bench, comparing the top 25% of heads ranked by gradient-based importance with the remaining 75%. Low bridge and high core scores are defined as $b \leq \text{median}(b)$ and $c > \text{median}(c)$, respectively, with medians computed across all heads. Two-sided p-values test the null hypothesis of equal prevalence between the two groups and are adjusted for multiple comparisons using Holm's method (Holm, 1979). Smaller p-values indicate stronger evidence that the two groups differ in the proportion of heads exhibiting the corresponding structural feature.

Head property	Top 25%	Remaining 75%	p-value
Low Bridge Score	**89.3%**	36.9%	3.61×10^{-34}
High Core Score	**49.4%**	23.8%	5.95×10^{-9}

i, and set $S_{ij} = 0$ for non-edge pairs in $\mathcal{G}_\delta$. We define the *core score* as $\text{Core}_i = \frac{\sum_{j \in C(i)} S_{ij}}{\sum_{j \in \mathcal{V}} S_{ij} + \varepsilon}$, where ε is a small positive constant used for numerical stability. A higher core score indicates that a larger proportion of a head's total connection weight lies within its own community. We define the *bridge score* using the weighted participation coefficient (Pedersen et al., 2017): $\text{Bridge}_i = 1 - \sum_{k=1}^{K} \left(\frac{s_{i,k}}{s_i + \varepsilon} \right)^2$, where $s_{i,k} = \sum_{j \in C_k} S_{ij}$ and $s_i = \sum_{j \in \mathcal{V}} S_{ij}$. A higher bridge score indicates that connection weight is distributed more evenly across communities, whereas a lower score indicates greater concentration in fewer communities.

As shown in Table 1, low bridge scores and high core scores are each significantly more prevalent among important heads than among the remaining heads. This motivates our hypothesis that attention heads with high core scores and low bridge scores play an important role in LLM reasoning.

4.2 Validating Hypothesis Via Pruning

To test whether the identified attention heads with these structural properties support LLM reasoning performance, we use pruning to examine whether retaining heads with high core scores and low bridge scores better preserves performance. Specifically, we propose Small-World Allocation (SWA), a hierarchical sparsity allocation method that uses core and bridge scores to allocate sparsity across layers and attention heads, assigning lower sparsity to heads with these properties.

SWA defines pruning priority scores at both the layer and head levels. Here, each layer corresponds to a transformer block containing self-attention and feed-forward modules. Layer-level scores allocate sparsity across these blocks, while head-level scores determine the pruning ratios of individual attention heads within each block. This hierarchical design accounts for structural differences both across layers and among heads within a layer.

At the head level, we combine core and bridge scores into a pruning priority score. For each head i, we first convert its bridge and core scores into percentile ranks over all heads, denoted by $\tilde{B}_i \in [0, 1]$ and $\tilde{C}_i \in [0, 1]$, where larger values indicate stronger structural roles. We define the priority score as

$$p_i = \tilde{B}_i (1 - \tilde{C}_i). \tag{8}$$

Heads with low bridge scores or high core scores receive smaller p_i values and are assigned lower sparsity, preserving more of their weights. At the layer level, we aggregate head-level pruning priority scores within each layer. Let $\mathcal{H}_\ell$ denote the set of heads in layer ℓ, we have $p_\ell = \frac{1}{|\mathcal{H}_\ell|} \sum_{i \in \mathcal{H}_\ell} p_i$.

A smaller score p_ℓ indicates that layer ℓ contains more structurally important heads and should receive a lower sparsity budget, while a larger p_ℓ indicates higher layer-level pruning priority.

SWA assigns pruning budgets in proportion to the priority scores through a hierarchical allocation procedure. Layer-level scores first determine the sparsity budget for each transformer block, including all modules within the block; head-level scores then distribute the attention sparsity across individual heads. Following Hu et al. (2025), we normalize the scores, convert them into relative pruning ratios, and rescale the ratios so that the overall sparsity matches the target global sparsity.

Table 2: We report the small-world index (SWI ↑) of head-head functional graphs after applying SparseGPT and its variants at sparsity 0.5. SWA (Head) and SWA (Layer) denote head-only and layer-only allocation strategies, respectively. **Higher SWI indicates better preservation** of small-world functional organization under pruning. Results are averaged over three seeds and reported as mean$_{\pm \text{std}}$. The best result for each model is **bolded**.

Model	SparseGPT	+ FARMS	+ ATP	+ SWA (Head)	+ SWA (Layer)	+ SWA
Qwen3 14B	$0.8000_{\pm 0.76}$	$0.8031_{\pm 0.09}$	$0.8797_{\pm 0.37}$	$0.9867_{\pm 0.23}$	$1.057_{\pm 0.14}$	$\mathbf{1.1386}_{\pm 0.40}$
Qwen3 8B	$0.6509_{\pm 0.64}$	$0.7982_{\pm 0.90}$	$0.8432_{\pm 0.30}$	$0.8639_{\pm 0.22}$	$0.9053_{\pm 0.15}$	$\mathbf{0.9196}_{\pm 0.44}$
Qwen3 4B	$0.5400_{\pm 0.10}$	$0.5516_{\pm 0.81}$	$0.6555_{\pm 0.14}$	$0.6449_{\pm 0.06}$	$0.7107_{\pm 0.25}$	$\mathbf{0.7212}_{\pm 0.58}$
Llama3 8B	$0.8085_{\pm 0.05}$	$0.8229_{\pm 0.28}$	$0.8300_{\pm 0.68}$	$0.8634_{\pm 0.64}$	$0.8943_{\pm 0.22}$	$\mathbf{0.9040}_{\pm 0.45}$
Llama3.2 3B	$0.8372_{\pm 0.21}$	$0.8380_{\pm 0.27}$	$0.8530_{\pm 0.14}$	$0.8468_{\pm 0.26}$	$0.8777_{\pm 0.09}$	$\mathbf{0.8867}_{\pm 0.24}$
Llama3.2 1B	$0.5138_{\pm 0.50}$	$0.5408_{\pm 0.40}$	$0.5567_{\pm 0.54}$	$0.5943_{\pm 0.93}$	$0.6641_{\pm 0.67}$	$\mathbf{0.6740}_{\pm 0.43}$

Table 3: Comparison of **WikiText Perplexity (↓)**. We evaluate Base pruning methods (Wanda and SparseGPT) and their variants augmented with FARMS, ATP, and SWA at sparsity 0.5/0.7. Results are averaged over three seeds and reported as mean$_{\pm \text{std}}$. The best result in each setting is **bolded**. SWA reduces perplexity and achieves the best values across all the models and sparsity levels.

Model	Size	Sparsity	Wanda				SparseGPT			
			Base	+ FARMS	+ ATP	+ SWA	Base	+ FARMS	+ ATP	+ SWA
Llama 3.2	1B	0.5	$36.25_{\pm 0.18}$	$33.90_{\pm 0.02}$	$24.95_{\pm 0.04}$	$\mathbf{21.92}_{\pm 0.05}$	$21.54_{\pm 0.29}$	$21.14_{\pm 0.56}$	$20.56_{\pm 0.34}$	$\mathbf{20.47}_{\pm 0.15}$
		0.7	$415.60_{\pm 32.1}$	$380.20_{\pm 20.1}$	$264.10_{\pm 21.7}$	$\mathbf{256.67}_{\pm 18.00}$	$157.20_{\pm 3.2}$	$136.80_{\pm 5.3}$	$106.60_{\pm 3.2}$	$\mathbf{105.00}_{\pm 3.6}$
	3B	0.5	$19.71_{\pm 0.02}$	$16.45_{\pm 0.09}$	$14.42_{\pm 0.02}$	$\mathbf{13.32}_{\pm 0.12}$	$23.98_{\pm 2.15}$	$15.92_{\pm 1.85}$	$13.53_{\pm 0.08}$	$\mathbf{11.27}_{\pm 0.98}$
		0.7	$177.80_{\pm 5.6}$	$132.6_{\pm 5.3}$	$125.1_{\pm 0.8}$	$\mathbf{118.67}_{\pm 2.32}$	$70.70_{\pm 5.07}$	$67.67_{\pm 2.93}$	$51.51_{\pm 3.83}$	$\mathbf{49.30}_{\pm 0.93}$
Llama 3	8B	0.5	$11.86_{\pm 0.02}$	$11.12_{\pm 0.03}$	$11.50_{\pm 0.05}$	$\mathbf{10.14}_{\pm 0.03}$	$11.69_{\pm 0.13}$	$11.02_{\pm 0.87}$	$10.72_{\pm 0.15}$	$\mathbf{10.15}_{\pm 0.08}$
		0.7	$140.50_{\pm 2.6}$	$90.02_{\pm 3.00}$	$85.93_{\pm 0.82}$	$\mathbf{82.36}_{\pm 0.20}$	$55.94_{\pm 3.23}$	$52.55_{\pm 1.98}$	$40.40_{\pm 1.94}$	$\mathbf{38.93}_{\pm 1.29}$
Qwen 3	4B	0.5	$21.91_{\pm 2.08}$	$19.69_{\pm 0.16}$	$18.88_{\pm 0.03}$	$\mathbf{17.30}_{\pm 0.60}$	$20.52_{\pm 0.43}$	$19.58_{\pm 0.41}$	$20.44_{\pm 0.45}$	$\mathbf{19.30}_{\pm 0.77}$
		0.7	$101.80_{\pm 3.9}$	$90.49_{\pm 4.84}$	$64.78_{\pm 1.21}$	$\mathbf{62.44}_{\pm 0.27}$	$50.93_{\pm 1.62}$	$44.30_{\pm 2.26}$	$46.45_{\pm 0.91}$	$\mathbf{40.34}_{\pm 0.38}$
	8B	0.5	$14.90_{\pm 0.48}$	$13.20_{\pm 0.03}$	$12.91_{\pm 0.06}$	$\mathbf{11.90}_{\pm 0.09}$	$15.09_{\pm 0.07}$	$12.61_{\pm 0.12}$	$13.08_{\pm 0.09}$	$\mathbf{12.10}_{\pm 0.14}$
		0.7	$71.10_{\pm 1.86}$	$65.46_{\pm 1.51}$	$57.31_{\pm 1.28}$	$\mathbf{52.56}_{\pm 1.62}$	$30.05_{\pm 0.66}$	$25.13_{\pm 1.07}$	$28.93_{\pm 0.43}$	$\mathbf{20.10}_{\pm 0.14}$
	14B	0.5	$11.43_{\pm 0.04}$	$11.38_{\pm 0.04}$	$10.79_{\pm 0.04}$	$\mathbf{10.02}_{\pm 0.03}$	$11.51_{\pm 0.05}$	$11.36_{\pm 0.02}$	$10.92_{\pm 0.10}$	$\mathbf{9.95}_{\pm 0.13}$
		0.7	$37.04_{\pm 0.52}$	$35.62_{\pm 0.27}$	$34.29_{\pm 0.65}$	$\mathbf{33.71}_{\pm 0.45}$	$20.82_{\pm 0.35}$	$20.18_{\pm 0.16}$	$20.37_{\pm 0.23}$	$\mathbf{18.49}_{\pm 0.28}$

5 EXPERIMENTS

We conduct pruning experiments to validate our hypothesis that attention heads with high core scores and low bridge scores play an important role in supporting LLM reasoning performance. Specifically, we evaluate whether SWA better preserves both small-world functional organization and model performance during pruning. Our evaluation is guided by four research questions: **RQ1.** Does SWA better preserve small-world functional organization than existing pruning baselines? **RQ2.** Does SWA better maintain model performance than existing pruning baselines at the same sparsity level? **RQ3.** How do layer-level and head-level sparsity allocation contribute to SWA's effectiveness? **RQ4.** How sensitive is SWA's pruning performance to the dataset used for functional graph construction?

5.1 EXPERIMENTAL SETUP

First, we describe the setup used to collect activations, construct attention-head functional graphs, and compute SWI. Second, we describe the pruning setup used to evaluate whether SWA preserves performance and reasoning ability under sparsity. Details are provided in Appendix A.2.

Setup for SWI analysis. We first evaluate the relationship between SWI and reasoning performance using six open-source LLMs: Qwen3-4B, Qwen3-8B, Qwen3-14B, Llama-3.2-1B, Llama-3.2-3B, and Meta-Llama-3-8B (Yang et al., 2025a; Grattafiori et al., 2024). We collect attention-head activations on DRE-Bench (Yang et al., 2025b), using Level 1 and 2 questions as calibration inputs for constructing functional graphs. We exclude Levels 3 and 4 because current open-source models often fail on these harder tasks, making their answers unreliable for activation-based analysis. We measure DRE-Bench performance as the mean per-token negative log-likelihood of answer tokens (answer NLL/token; lower is better) and compare it with each model's SWI.

Table 4: Ablation study on hierarchical structural deletion. We report the absolute WikiText perplexity (PPL ↓) of SWA and the increase in PPL for each ablated variant relative to SWA on Qwen3 8B at sparsity levels of 0.5 and 0.7. SWA(Layer) uses only layer-level scores, and SWA(Head) uses only head-level scores.

Variant	SparseGPT		Wanda	
	0.5	0.7	0.5	0.7
SWA	**12.10**	**20.10**	**11.90**	**52.56**
SWA(Layer)	+3.00	+10.21	+0.08	+5.73
SWA(Head)	+3.69	+19.97	+0.17	+44.99

Table 5: Effect of the graph-construction benchmark on SWA. We report WikiText perplexity (↓) for SparseGPT at 70% sparsity, comparing the base method with SWA using functional graphs built from GSM8K, ARC-C, or MMLU.

Model	Base	GSM8K	ARC-C	MMLU
Llama3 8B	55.94 ± 3.23	42.60 ± 1.22	41.10 ± 1.35	40.75 ± 1.07
Qwen3 4B	50.93 ± 1.62	40.42 ± 1.12	38.90 ± 1.23	39.89 ± 2.35
Llama3.2 3B	70.70 ± 5.07	50.54 ± 1.30	49.69 ± 2.69	48.95 ± 1.20
Llama3.2 1B	157.20 ± 3.20	109.2 ± 6.20	110.9 ± 2.60	105.28 ± 6.62

Setup for pruning evaluation. For pruning experiments, we use the same six LLMs and apply SWA to SparseGPT (Frantar & Alistarh, 2023) and Wanda (Sun et al., 2023). We compare against uniform sparsity and layer-wise allocation baselines, including FARMS (Hu et al., 2025) and ATP (Huang et al., 2025). We evaluate pruned models using WikiText perplexity (Merity et al., 2016).

5.2 Effectiveness of Preserving SWI

To answer **RQ1**, we evaluate whether SWA preserves small-world functional organization during pruning. For each of the six models, we apply SparseGPT at a sparsity level of 0.5 and compare it with variants using our score-guided allocation strategies. We then recompute the SWI of the resulting head–head functional graphs. **SWA (Head)** and **SWA (Layer)** denote head-only and layer-only allocation, respectively. As shown in Table 2, SWA consistently achieves higher SWI values than the baseline pruning methods. Although the absolute SWI decreases substantially after pruning, SWA retains higher relative SWI than competing allocation strategies under the same sparsity level.

5.3 Effectiveness of SWA

To answer **RQ2**, we evaluate SWA on the WikiText validation set and report perplexity (PPL) in Table 3. SWA consistently outperforms the baselines across all models and sparsity levels, demonstrating that preserving small-world functional organization during pruning improves performance. Notably, the gap widens at higher sparsity, indicating that SWI-preserving pruning is particularly beneficial under aggressive pruning. Zero-shot results on downstream tasks are reported in Section A.4 of the Appendix, and results on preserving reasoning ability on other benchmarks in Section A.5.

5.4 Ablation of Hierarchical Sparsity Allocation

To answer **RQ3**, we conduct an ablation study to examine the contribution of each level of SWA's hierarchical allocation. We compare our method with two ablated variants: **SWA(Layer)**, which uses only the layer-level scores, and **SWA(Head)**, which uses only the head-level scores. We evaluate all variants on Qwen3 8B with both SparseGPT and Wanda as the base pruning methods, using WikiText perplexity at sparsity levels of 0.5 and 0.7. As shown in Table 4, removing either level increases perplexity in every setting, indicating that both head-level and layer-level structural signals contribute to preserving model quality after pruning. This degradation grows markedly from 0.5 to 0.7 sparsity for both base methods. Moreover, SWA(Head) degrades more than SWA(Layer) in every setting. These results suggest that layer-level scores provide the main allocation signal, while head-level scores refine pruning within layers.

5.5 Sensitivity to the Graph-Construction Dataset

To answer **RQ4** we examine whether SWA remains effective when functional graphs are constructed from different benchmarks. Table 5 reports WikiText perplexity for SparseGPT at 70% sparsity, with SWA using graphs constructed from GSM8K (Cobbe et al., 2021), ARC-C (Clark et al., 2018), or MMLU (Hendrycks et al., 2020). For each benchmark, we construct the functional graph using 400 training examples, or all available training examples if fewer than 400 are available. Across the four models with reported results, all three graph sources yield lower perplexity than the base method. For example, on Llama-3.2-3B, SWA reduces perplexity from 70.70 to 48.95–50.54 across the three graph sources. Although the best graph source varies across models, these consistent improvements suggest that the benefits of SWA are not limited to a single graph-construction benchmark.

6 Related Works

Structural Analysis of Neural Networks. Prior work has studied neural-network structure through architecture graphs, weight spectra, and representation similarity. Architecture-level analyses relate graph properties such as clustering and path length to predictive performance, while weight-space methods based on random matrix theory connect spectral structure to training quality, generalization, and pruning sensitivity (You et al., 2020; Martin & Mahoney, 2021). Representation-similarity methods such as SVCCA and CKA compare hidden states across layers and models, but do not explicitly model interactions among model components (Raghu et al., 2017; Kornblith et al., 2019; Liu et al., 2025b). For transformer-based models, prior studies show that attention heads are heterogeneous: some encode positional, syntactic, or induction-like behavior, while many are redundant or weakly used (Clark et al., 2019b; Voita et al., 2019; Michel et al., 2019). LLM pruning methods exploit this heterogeneity through weight- or layer-level importance estimates, including SparseGPT, Wanda, and ATP (Frantar & Alistarh, 2023; Sun et al., 2023; Huang et al., 2025). In contrast, our work constructs activation-defined functional graphs over attention heads, quantifies their global small-world organization, and derives local roles from this graph structure.

Functional Connectivity and Small-world Organization. Functional connectivity measures statistical dependence among distributed neural activities (Friston, 1994). Graph-theoretic work characterizes small-world networks by high clustering and short path lengths (Watts & Strogatz, 1998), or by high local and global efficiency (Latora & Marchiori, 2001). In neuroscience, small-world organization is viewed as a balance between segregated specialization and integrated communication in brain networks (Bassett & Bullmore, 2006; Bullmore & Sporns, 2009). Related machine-learning work uses activation-based graphs, activation correlations, and topological summaries to study neural-network organization and generalization (Horta et al., 2021; Ballester et al., 2024; 2023), while transformer representation analyses reveal clustered structure across layers (Phang et al., 2021). Building on this perspective, we use SWI to test whether attention-head activations in LLMs form small-world functional organization associated with reasoning capability.

7 Conclusion

In this work, we investigated small-world functional organization in attention-head networks and its relationship to LLM reasoning performance. Across models and training checkpoints, higher SWI is consistently associated with better fluid reasoning performance. Our head-level analysis further reveals that important heads tend to exhibit stronger within-community connectivity and more concentrated connectivity across communities, characterized by high core scores and low bridge scores, respectively. Motivated by these findings, we proposed Small-World Allocation (SWA) to examine whether preferentially preserving heads with these properties helps maintain performance under pruning. Experiments show that SWA better preserves both small-world organization and model performance than competing allocation strategies. Together, these findings highlight the relevance of global functional organization and local connectivity patterns to LLM reasoning, and demonstrate their practical value for guiding model compression.

AI USE STATEMENT

We used generative AI tools to help with writing, editing, and proofreading this manuscript. The authors reviewed and checked all AI-assisted text, and take full responsibility for the content, methods, results, and conclusions of this paper.

ETHICS STATEMENT

This work uses publicly available models and datasets and does not involve human-subject research or the collection of personal or sensitive data. Our experiments focus on analyzing attention-head functional organization and evaluating model pruning. We do not identify additional ethical concerns specific to the data or experimental procedures used in this study.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details needed to reproduce our results. Sections 2 and 3 describe the representation similarity measure, functional graph construction, and small-world index. Section 5.1 and Appendix A.2 describe the models, datasets, evaluation protocols, and pruning setup. Additional results are provided in the appendix, and all models and datasets used in our experiments are publicly available. We will release our codes to facilitate reproduction and further research.

REFERENCES

Rubén Ballester, Carles Casacuberta, and Sergio Escalera. Topological data analysis for neural network analysis: A comprehensive survey. *arXiv preprint arXiv:2312.05840*, 2023.

Ruben Ballester, Xavier Arnal Clemente, Carles Casacuberta, Meysam Madadi, Ciprian A Corneanu, and Sergio Escalera. Predicting the generalization gap in neural networks using topological data analysis. *Neurocomputing*, 596:127787, 2024.

Aron K Barbey. Network neuroscience theory of human intelligence. *Trends in cognitive sciences*, 22(1):8–20, 2018.

Danielle S Bassett and Edward T Bullmore. Small-world brain networks revisited. *The Neuroscientist*, 23(5):499–516, 2017.

Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12(6): 512–523, 2006.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Bharat Biswal, F Zerrin Yetkin, Victor M Haughton, and James S Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic resonance in medicine*, 34(4):537–541, 1995.

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, 2008.

Douglas A Bors and François Vigneau. The effect of practice on raven's advanced progressive matrices. *Learning and Individual Differences*, 13(4):291–312, 2003.

Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019a.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pp. 276–286, 2019b.

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. *arXiv preprint arXiv:1803.05457*, 2018.

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

Michael D Fox, Abraham Z Snyder, Justin L Vincent, Maurizio Corbetta, David C Van Essen, and Marcus E Raichle. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences*, 102(27):9673–9678, 2005.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Karl J Friston. Functional and effective connectivity in neuroimaging: a synthesis. *Human brain mapping*, 2(1-2):56–78, 1994.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pp. 65–70, 1979.

Vitor AC Horta, Ilaria Tiddi, Suzanne Little, and Alessandra Mileo. Extracting knowledge from deep neural networks through graph analysis. *Future Generation Computer Systems*, 120:109–118, 2021.

SM Hadi Hosseini and Shelli R Kesler. Influence of choice of null network on small-world parameters of structural correlation networks. *PLoS One*, 8(6):e67354, 2013.

Yuanzhe Hu, Kinshuk Goel, Vlad Killiakov, and Yaoqing Yang. Eigenspectrum analysis of neural networks without aspect ratio bias. *arXiv preprint arXiv:2506.06280*, 2025.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, and Rongrong Ji. Determining layer-wise sparsity for large language models through a theoretical perspective. *arXiv preprint arXiv:2502.14770*, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Nicolas Langer, Andreas Pedroni, Lorena RR Gianotti, Jürgen Hänggi, Daria Knoch, and Lutz Jäncke. Functional brain network efficiency predicts intelligence. *Human brain mapping*, 33(6):1393–1406, 2012.

Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical review letters*, 87(19):198701, 2001.

Maikel Leon. Gpt-5 and open-weight large language models: Advances in reasoning, transparency, and control. *Information Systems*, pp. 102620, 2025.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Xuyuan Liu, Lei Hsiung, Yaoqing Yang, and Yujun Yan. Spectral insights into data-oblivious critical layers in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4860–4877, 2025b.

Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael W Mahoney, and Yaoqing Yang. Alphapruning: Using heavy-tailed self regularization theory for improved layer-wise pruning of large language models. *Advances in neural information processing systems*, 37:9117–9152, 2024.

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

Charles H Martin, Tongsu Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, 2021.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Mangor Pedersen, Amir Omidvarnia, Jennifer M Walz, Andrew Zalesky, and Graeme D Jackson. Spontaneous brain network activity: Analysis of its temporal complexity. *Network Neuroscience*, 1 (2):100–115, 2017.

Jason Phang, Haokun Liu, and Samuel Bowman. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 529–538, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

John C Raven. Advanced progressive matrices: Sets i and ii. *(No Title)*, 1962.

Benjamin M Rosenberg, Eva Mennigen, Martin M Monti, and Roselinde H Kaiser. Functional segregation of human brain networks across the lifespan: an exploratory analysis of static and dynamic resting-state functional connectivity. *Frontiers in neuroscience*, 14:561594, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Bruce Thompson. Canonical correlation analysis. 2000.

William Hedley Thompson and Peter Fransson. On stabilizing the variance of dynamic functional brain connectivity time series. *Brain connectivity*, 6(10):735–746, 2016.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5797–5808, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Susan Whitfield-Gabrieli and Alfonso Nieto-Castanon. Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. *Brain connectivity*, 2(3):125–141, 2012.

Junjie Wu, Mo Yu, Lemao Liu, Dit-Yan Yeung, and Jie Zhou. Understanding llms' fluid intelligence deficiency: An analysis of the arc task. In *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. 8339–8360, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yue Yang, MingKang Chen, Qihua Liu, Mengkang Hu, Qiguang Chen, Gengrui Zhang, Shuyue Hu, Guangtao Zhai, Yu Qiao, Yu Wang, et al. Truly assessing fluid intelligence of large language models through dynamic reasoning evaluation. *arXiv preprint arXiv:2506.02648*, 2025b.

Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *International conference on machine learning*, pp. 10881–10891. PMLR, 2020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

A Appendix

A.1 Justification of the Head Similarity Measure

Our head similarity measure is a variant of SVCCA that retains its SVD–CCA procedure and replaces mean aggregation with the sum of Fisher z-transformed canonical correlations. This formulation compares vector-valued attention-head representations and summarizes their shared variation across questions as a scalar similarity score.

Canonical correlations for vector-valued activations. In functional brain networks, connectivity between regions of interest (ROIs) is commonly estimated using Pearson correlation between scalar ROI activity time series (Biswal et al., 1995; Fox et al., 2005). An attention head instead produces a multidimensional activation vector, motivating a multivariate measure of correlation. CCA identifies linear projections of two representations that are maximally correlated, as described in Section 2.2. For nonconstant one-dimensional representations, the canonical correlation equals the absolute Pearson correlation. CCA thus extends the correlation-based perspective to vector-valued activations without requiring coordinate-wise alignment.

SVD-based dimensionality reduction. Following the SVD–CCA procedure of Raghu et al. (2017), we retain the leading singular directions explaining at least 99% of the variance in each representation before applying CCA. This step focuses the comparison on subspaces that capture most of the observed variation and removes low-variance directions that may reflect noise. The resulting canonical correlations describe how strongly the two heads' projected query activations vary together across the sampled questions.

Fisher z-based aggregation. We apply the Fisher z-transform to each canonical correlation and sum the transformed coefficients, as defined in Equation (7). The Fisher z-transform is widely used in functional brain connectivity analysis to facilitate statistical comparisons of correlation estimates (Whitfield-Gabrieli & Nieto-Castanon, 2012; Thompson & Fransson, 2016). Here, we use its nonlinear scaling: for nonnegative correlations, $\operatorname{atanh}(\rho)$ is increasing and convex, placing greater emphasis on strongly correlated canonical directions. We clip the coefficients at $1 - \epsilon_{\text{clip}}$ to keep the score finite. The sum measures aggregate alignment and depends on both the correlation strengths and the number of retained canonical directions. We use this transformation as an aggregation rule, without assuming that it stabilizes the sampling variance of the estimated canonical correlations.

Comparison with CKA. Centered Kernel Alignment (CKA) (Kornblith et al., 2019) compares centered similarity matrices over examples. Our measure instead identifies correlated linear projections and aggregates their correlation strengths. We adopt this formulation to represent functional connectivity between heads through multivariate correlations, consistent with the correlation-based perspective used in brain network analysis. This choice reflects the intended interpretation of our graph edges rather than a claim of general superiority over CKA.

A.2 Experimental Details

We first describe the datasets and settings used for functional graph construction and SWI analysis. We then present the pruning methods, allocation baselines, and evaluation protocols.

A.2.1 Graph Construction and SWI Analysis

Graph-construction datasets. We use DRE-Bench (Yang et al., 2025b) for the main SWI analysis and pruning experiments. DRE-Bench evaluates fluid reasoning through abstract tasks involving latent rules. We sample 400 questions for graph construction: 200 from Level 1, which covers attribute-based rules, and 200 from Level 2, which covers spatial reasoning rules.

For the graph-construction dataset sensitivity experiment, we additionally use GSM8K (Cobbe et al., 2021), ARC-C (Clark et al., 2018), and MMLU (Hendrycks et al., 2020). For each dataset, we use 400 training examples, or all available training examples if fewer than 400 are available. The examples used for graph construction are excluded from downstream evaluations.

Activation collection and head similarity. We follow the activation collection procedure in Section 3.1, recording query activations and averaging them over the generated answer tokens for each question. We mean-center each head's activation matrix and apply SVD, retaining the leading directions that explain at least 99% of its variance. We then apply CCA to each pair of reduced representations and aggregate the canonical correlations using the Fisher z-based similarity in Equation (7). The variance threshold is shared across all models and reported SWI analyses.

Graph construction and SWI computation. We construct graphs at a fixed edge density $\delta = 0.25$. We rank all unordered pairs of distinct heads by their Fisher z-aggregated similarity and retain the top $K_\delta = \lfloor \delta |\mathcal{P}| \rfloor$ pairs as edges, where $\mathcal{P}$ is the set of eligible head pairs. Self-connections are excluded. We select $\delta = 0.25$ based on a density sweep, which identifies it as a consistent setting for comparing SWI across model families. We compute SWI following Equation (3), using degree-preserving randomized graphs as the reference.

Performance measurement. We measure DRE-Bench performance using the mean per-token negative log-likelihood over the reference answer tokens, where lower values indicate better performance. We compare this measure with SWI across the six LLMs and across training checkpoints of Pythia-6.9B.

A.2.2 Pruning Evaluation

We integrate SWA with SparseGPT and Wanda and compare it against the base methods with uniform sparsity and variants using FARMS or ATP for layer-wise sparsity allocation.

SparseGPT. SparseGPT (Frantar & Alistarh, 2023) is a one-shot post-training pruning method that approximates each layer's original outputs on calibration data. It uses second-order information to select pruning masks and update the remaining weights without fine-tuning. We replace its uniform sparsity setting with the allocations produced by SWA or the competing allocation methods.

Wanda. Wanda (Sun et al., 2023) estimates weight importance using weight magnitude and input activation statistics. For a weight matrix W, it scores each weight by $|W_{ij}| \cdot \|X_j\|_2$, where X_j contains the observed activations of input feature j. Weights with lower scores are pruned within each output channel. Wanda does not update the remaining weights after pruning. We use it as a second base method to evaluate SWA beyond reconstruction-based pruning.

FARMS. FARMS (Hu et al., 2025) is an eigenspectrum-based layer-wise allocation method that addresses aspect-ratio bias in heavy-tailed self-regularization analysis. It samples submatrices with a fixed aspect ratio and aggregates their empirical spectral densities to obtain comparable estimates across heterogeneous weight matrices. These estimates guide layer-wise sparsity allocation.

ATP. ATP (Huang et al., 2025) allocates layer-wise sparsity to mitigate the propagation of reconstruction errors during pruning. It assigns lower sparsity to earlier layers and higher sparsity to later layers using an increasing arithmetic progression, with the common difference selected through search.

Language modeling and zero-shot evaluation. We evaluate language modeling using WikiText perplexity (Merity et al., 2016), where lower values indicate better performance. We also report average zero-shot accuracy across BoolQ (Clark et al., 2019a), RTE (Wang et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy and ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). We evaluate both base pruning methods at sparsity levels of 0.5 and 0.7. Results are averaged over three seeds and reported as $\text{mean}_{\pm \text{std}}$.

ARAOC evaluation. We additionally evaluate pruned models on the Abstraction and Reasoning on Atom Operation Corpus (ARAOC) (Wu et al., 2025). ARAOC contains six types of atomic grid transformations: *Move*, *Change Color*, *Copy*, *Mirror*, *Fill Internal*, and *Scale*.

We report **Not M**, the percentage of predictions whose output-grid shape differs from the ground-truth shape. Lower values indicate better preservation of output-shape correctness. ARAOC is used

Table 6: Relationship between DRE-Bench performance and small-world organization across Pythia-6.9B training checkpoints. We report the mean per-token negative log-likelihood over the answer tokens (NLL/token; lower is better) and SWI of the head-head functional graph on DRE-Bench Level 1 and Level 2 tasks. Results are averaged over three seeds and reported as mean$_{\pm\text{std}}$. Across training checkpoints, lower NLL is associated with higher SWI on both levels.

Training step	Tokens seen	Level 1		Level 2	
		NLL/token ↓	SWI ↑	NLL/token ↓	SWI ↑
4,000	8B	$0.5089_{\pm 0.16}$	$0.6760_{\pm 0.85}$	$0.8255_{\pm 0.25}$	$0.4587_{\pm 0.67}$
11,000	23B	$0.4677_{\pm 0.20}$	$0.8008_{\pm 0.13}$	$0.7020_{\pm 0.68}$	$0.5574_{\pm 0.80}$
30,000	63B	$0.4344_{\pm 0.28}$	$0.8035_{\pm 0.12}$	$0.7007_{\pm 0.52}$	$0.5599_{\pm 0.54}$
65,000	136B	$0.2250_{\pm 0.30}$	$0.8135_{\pm 0.19}$	$0.6411_{\pm 0.34}$	$0.8100_{\pm 0.32}$
71,000	149B	$0.1717_{\pm 0.07}$	$0.8320_{\pm 0.47}$	$0.4026_{\pm 0.18}$	$0.8469_{\pm 0.33}$
113,000	237B	$0.1623_{\pm 0.09}$	$0.9768_{\pm 0.45}$	$0.3765_{\pm 0.17}$	$0.9996_{\pm 0.88}$
143,000	300B	$0.1374_{\pm 0.03}$	$1.0075_{\pm 0.22}$	$0.3749_{\pm 0.03}$	$1.1864_{\pm 0.21}$

Table 7: Relationship between DRE-Bench performance and small-world organization. We report the mean per-token negative log-likelihood over the answer tokens (NLL/token; lower is better) and SWI of the head-head functional graph on DRE-Bench Level 1 and Level 2 tasks. Results are averaged over three seeds and reported as mean$_{\pm\text{std}}$. Across models, lower NLL is associated with higher SWI.

Model	Level 1		Level 2	
	NLL/token ↓	SWI ↑	NLL/token ↓	SWI ↑
Qwen3 14B	$0.0822_{\pm 0.01}$	$1.8160_{\pm 0.27}$	$0.2410_{\pm 0.03}$	$1.7020_{\pm 0.12}$
Qwen3 8B	$0.0949_{\pm 0.01}$	$1.5850_{\pm 0.28}$	$0.2814_{\pm 0.02}$	$1.6000_{\pm 0.07}$
Llama3 8B	$0.0980_{\pm 0.01}$	$1.3750_{\pm 0.19}$	$0.2993_{\pm 0.02}$	$1.5935_{\pm 0.11}$
Qwen3 4B	$0.1142_{\pm 0.02}$	$1.2180_{\pm 0.26}$	$0.3397_{\pm 0.05}$	$1.2722_{\pm 0.18}$
Llama3.2 3B	$0.1377_{\pm 0.07}$	$1.1933_{\pm 0.33}$	$0.3412_{\pm 0.10}$	$1.2100_{\pm 0.17}$
Llama3.2 1B	$0.1689_{\pm 0.20}$	$0.9188_{\pm 0.49}$	$0.5800_{\pm 0.26}$	$0.9344_{\pm 0.18}$

exclusively for evaluation and does not contribute to functional graph construction. We evaluate Qwen3-8B pruned with SparseGPT at 50% sparsity and report the mean and standard deviation over three runs.

A.2.3 Computing Infrastructure

All experiments are conducted on a cluster equipped with NVIDIA RTX PRO 6000 GPUs and AMD EPYC 9655 processors.

A.3 Numerical Results for the SWI–Performance Association

Tables 6 and 7 provide detailed numerical results for the checkpoint and cross-model analyses presented in Figures 2 and 3, respectively. We report mean answer-token negative log-likelihood (NLL/token) and SWI separately for DRE-Bench Level 1 and Level 2. Results are averaged over three seeds and reported as mean$_{\pm\text{std}}$.

Across training checkpoints. Table 6 reports results for seven training checkpoints of Pythia-6.9B (Biderman et al., 2023). On both levels, mean answer-token NLL decreases as training progresses, while mean SWI increases. For example, on Level 2, NLL decreases from 0.8255 at step 4,000 to 0.3749 at step 143,000, while SWI increases from 0.4587 to 1.1864. This comparison examines the association across checkpoints with the same architecture and parameter count.

Across models. Table 7 reports results for six LLMs from the Qwen and Llama families. On both levels, models with lower mean answer-token NLL consistently exhibit higher mean SWI. Qwen3-14B achieves the lowest NLL and highest SWI, whereas Llama-3.2-1B exhibits the highest NLL and lowest SWI. These numerical results document the cross-model trend discussed in Section 3.2.

Table 8: Comparison of Zero-shot Accuracy (↑). We evaluate Base pruning methods (Wanda and SparseGPT) and their variants augmented with FARMS, ATP, and SWA at sparsity 0.5/0.7. Results are averaged over three seeds and reported as mean$_{\pm \text{std}}$. The best result in each setting is **bolded**. SWA improves zero-shot accuracy and achieves the strongest performance over all the models and sparsity levels.

Model	Size	Sparsity	Wanda				SparseGPT			
			Base	+ FARMS	+ ATP	+ SWA	Base	+ FARMS	+ ATP	+ SWA
Llama 3.2	1B	0.5	$34.91_{\pm 3.67}$	$35.21_{\pm 2.28}$	$37.16_{\pm 0.82}$	**38.89**$_{\pm 1.03}$	$39.57_{\pm 2.67}$	$40.33_{\pm 2.41}$	$41.26_{\pm 0.23}$	**44.67**$_{\pm 1.75}$
		0.7	$25.86_{\pm 3.62}$	$27.29_{\pm 4.75}$	$29.86_{\pm 4.65}$	**32.41**$_{\pm 4.81}$	$30.85_{\pm 4.67}$	$29.38_{\pm 1.33}$	$35.29_{\pm 0.22}$	**36.46**$_{\pm 0.34}$
	3B	0.5	$40.53_{\pm 1.41}$	$42.71_{\pm 1.22}$	$44.57_{\pm 1.39}$	**45.80**$_{\pm 1.87}$	$45.50_{\pm 0.36}$	$46.67_{\pm 0.50}$	$46.05_{\pm 1.41}$	**49.91**$_{\pm 0.31}$
		0.7	$40.16_{\pm 2.41}$	$41.50_{\pm 1.63}$	$40.91_{\pm 4.74}$	**43.04**$_{\pm 0.87}$	$40.42_{\pm 1.16}$	$42.25_{\pm 0.85}$	$41.69_{\pm 4.46}$	**43.57**$_{\pm 1.11}$
Llama 3	8B	0.5	$52.12_{\pm 1.25}$	$53.49_{\pm 2.17}$	$52.43_{\pm 1.32}$	**55.13**$_{\pm 1.52}$	$48.53_{\pm 2.24}$	$50.28_{\pm 1.30}$	$53.48_{\pm 0.61}$	**55.61**$_{\pm 0.85}$
		0.7	$27.57_{\pm 0.07}$	$27.14_{\pm 1.84}$	$35.90_{\pm 0.74}$	**37.89**$_{\pm 0.90}$	$37.26_{\pm 5.39}$	$38.38_{\pm 0.90}$	$42.33_{\pm 2.86}$	**43.32**$_{\pm 1.56}$
Qwen 3	4B	0.5	$45.69_{\pm 3.35}$	$46.33_{\pm 2.49}$	$47.45_{\pm 0.39}$	**48.07**$_{\pm 0.42}$	$43.75_{\pm 1.25}$	$47.28_{\pm 1.04}$	$46.95_{\pm 4.51}$	**48.00**$_{\pm 1.23}$
		0.7	$35.86_{\pm 0.06}$	$36.60_{\pm 1.64}$	$36.98_{\pm 0.48}$	**38.30**$_{\pm 0.68}$	$41.21_{\pm 4.51}$	$42.50_{\pm 0.89}$	$42.00_{\pm 4.02}$	**43.02**$_{\pm 1.23}$
	8B	0.5	$52.50_{\pm 0.89}$	$53.28_{\pm 0.77}$	$55.38_{\pm 0.19}$	**57.84**$_{\pm 0.24}$	$53.04_{\pm 3.22}$	$54.11_{\pm 2.70}$	$55.64_{\pm 0.86}$	**56.59**$_{\pm 0.68}$
		0.7	$38.95_{\pm 2.04}$	$39.57_{\pm 0.53}$	$39.64_{\pm 0.53}$	**41.66**$_{\pm 0.88}$	$42.24_{\pm 0.52}$	$42.91_{\pm 1.72}$	$43.48_{\pm 0.74}$	**46.11**$_{\pm 0.72}$
	14B	0.5	$57.05_{\pm 0.44}$	$57.31_{\pm 0.22}$	$57.86_{\pm 1.15}$	**59.30**$_{\pm 0.45}$	$57.12_{\pm 0.39}$	$57.62_{\pm 4.73}$	$57.38_{\pm 0.15}$	**58.10**$_{\pm 0.34}$
		0.7	$43.91_{\pm 3.11}$	$44.69_{\pm 2.42}$	$45.41_{\pm 0.93}$	**46.23**$_{\pm 2.71}$	$48.43_{\pm 2.22}$	$49.11_{\pm 1.80}$	$48.97_{\pm 2.01}$	**49.93**$_{\pm 2.22}$

Table 9: ARAOC task mismatch rate at sparsity 0.5. We report mismatch rate (Not M ↓) across six tasks. Results are averaged over three runs and reported as mean$_{\pm \text{std}}$. The best result in each setting is **bolded**.

Method	Move	Change Color	Copy	Mirror	Fill Internal	Scale
Qwen3 8B (Non-Pruned)	$36.00_{\pm 4.11}$	$23.14_{\pm 4.61}$	$12.18_{\pm 4.12}$	$84.66_{\pm 3.20}$	$6.07_{\pm 1.99}$	$83.33_{\pm 5.87}$
SparseGPT	$44.21_{\pm 9.20}$	$48.87_{\pm 10.40}$	$60.99_{\pm 10.67}$	$94.76_{\pm 8.21}$	$34.25_{\pm 4.67}$	$88.91_{\pm 8.03}$
SparseGPT + FARMS	$46.22_{\pm 8.21}$	$50.77_{\pm 15.85}$	$56.60_{\pm 12.80}$	$96.66_{\pm 5.48}$	$33.11_{\pm 5.11}$	$88.06_{\pm 8.95}$
SparseGPT + ATP	$43.09_{\pm 9.01}$	$47.73_{\pm 8.66}$	$56.79_{\pm 12.48}$	$93.56_{\pm 9.63}$	$32.66_{\pm 5.81}$	$89.49_{\pm 7.47}$
SparseGPT + **SWA**	**40.29**$_{\pm 9.25}$	**46.28**$_{\pm 9.62}$	**56.11**$_{\pm 11.62}$	**90.21**$_{\pm 8.89}$	**28.69**$_{\pm 4.28}$	**85.10**$_{\pm 7.28}$

A.4 Zero-Shot Performance of SWA

Table 8 reports average zero-shot accuracy on the downstream tasks described in Section A.2.2. SWA achieves the highest mean accuracy across all six models at both sparsity levels when combined with either Wanda or SparseGPT. At 70% sparsity, SWA improves the average accuracy of Wanda-pruned Llama-3-8B from 27.57% to 37.89%, outperforming the strongest competing allocation method, ATP, by 1.99 percentage points. These results show that SWA's benefits extend beyond language modeling perplexity to downstream task performance.

A.5 Performance on ARAOC Reasoning Tasks

Table 9 reports ARAOC output-grid shape mismatch rates for Qwen3-8B at 50% sparsity. The benchmark and evaluation metric are described in Section A.2.2. SWA achieves lower mismatch rates than the pruning baselines across all six task types. For example, on *Fill Internal*, SWA reduces the mismatch rate from 34.25% with SparseGPT to 28.69%, compared with 32.66% for ATP. Although mismatch rates remain higher than those of the unpruned model, SWA better preserves output-shape correctness on these abstract reasoning tasks.