Farther the Shift, Sparser the Representation: Analyzing OOD Mechanisms in LLMs

Mingyu Jin[1], Yutong Yin[2], Jingcheng Niu[3]†, Qingcheng Zeng[2], Wujiang Xu[1], Xinyuan Song[7]
Mengnan Du[4], Wei Cheng[5], Zhaoran Wang[2], Tianlong Chen[6], Dimitris N Metaxas[1]

[1]Rutgers University [2]Northwestern University [3]UKP Lab, TU Darmstadt
[4]New Jersey Institute of Technology [5]NEC Laboratories America
[6]University of North Carolina at Chapel Hill [7]Emory University

Abstract

In this work, we investigate how the internal representations of Large Language Models (LLMs) change with inputs of increasing difficulty. Although sparsity changes may not be obvious for individual samples, dataset-level results reveal a consistent statistical trend: as task difficulty increases (e.g., harder questions, longer contexts, or more answer choices), the last hidden states of LLMs become systematically sparser. In short, *the harder the task, the sparser the representation*. This sparsity–difficulty relation is observable across diverse models and domains, suggesting that harder inputs drive more concentrated activation patterns in the last hidden state. Through a series of controlled analyses and learning-dynamics experiments, we show that representational density is a learned property of data familiarity: models develop rich, distributed representations for mastered patterns, while unfamiliar inputs default to sparser activations. Our finding is also actionable. We illustrate this with Sparsity-Guided Curriculum In-Context Learning (SG-ICL), which uses sparsity to select few-shot demonstrations matched to the query's difficulty, outperforming standard CoT and Auto-CoT baselines on MATH-500. Our study provides new insights into how LLM representations reflect task difficulty and how this signal can be leveraged for inference. Our code: `https://github.com/MingyuJ666/sparsityLLM`.

1 Introduction

Large Language Models (LLMs) serve as powerful tools for reasoning (Wei et al., 2022; Li et al., 2025b; Brown et al., 2020) and knowledge-intensive interaction (Wang et al., 2024b; Chen et al., 2023; Tongyi DeepResearch Team et al., 2025). Yet their reliability falters as the difficulty of the tasks increases (Lightman et al., 2024; Li et al., 2025a), and the tasks we want LLMs to solve are only going to get harder. This makes it crucial to know what happens when a model faces difficulty: do those failures reflect a hard ceiling on what the model has learned, or merely a soft edge of it? External benchmarks cannot answer this. We need to look deeper inside the models' internal mechanisms. In this paper, we examine how an LLM's internal representation changes with task difficulty and what that reveals about when a model has truly learned a problem and when it has not.

We look at ***sparsity***: the extent to which a representation's activation is concentrated on a small subset of dimensions. Sparsity has long been studied as a sign of specialization in neural representations and neural networks (Olshausen and Field, 1996; Glorot et al., 2011; Frankle and Carbin, 2019; Xiong et al., 2025, *inter alia)*, and shows up in LLM work on intrinsic dimensionality (Aghajanyan et al., 2021; De Cao et al., 2022). But it has mostly been treated as a fixed property of the trained model

†Corresponding author.

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

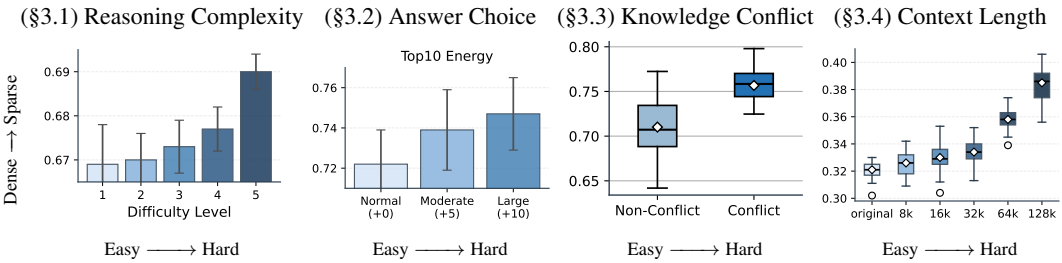


Figure 1: **Harder Inputs Induce Sparser Representations.** Across all four controlled difficulty axes, the last hidden states become progressively sparser as tasks get harder. Results are shown for `Qwen2.5-3B` using Top-10% Energy; nevertheless, the same trend holds across difficulty datasets and settings, sparsity metrics, LLM family, and LLM sizes.

(Liu et al., 2025), rather than something that changes with the input. We find that it does change with the input. The effect is noisy on individual examples, but at the dataset level it is clear and consistent: *as tasks get harder, the last hidden state gets sparser*.

Specifically, we measure sparsity in the last hidden state using Top-k Energy for multiple values of k, along with Effective Rank and Hoyer. We study difficulty along four dimensions: reasoning complexity, number of answer choices, conflicting knowledge, and context length (Sections 3.1–3.4; Figure 1). Across all four, the trend is the same: harder inputs yield sparser last-layer representations. The pattern holds across tasks, metrics, and model families, which makes it unlikely to be an artifact of any single task or model. In practical terms, increased sparsity means fewer dimensions carry most of the activation mass: the model uses a more concentrated set of features when the task gets harder.

Why does the representation become sparser as the task becomes harder? We find evidence that this regularity is a learned consequence of training rather than an architectural property. We study this both empirically and theoretically. Empirically, we train a toy Transformer from scratch on a synthetic knowledge graph: the pattern emerges only once the model has enough capacity and reaches convergence, although the training objective contains no explicit sparsity regularizer and provides no direct supervision for the harder-is-sparser pattern. Theoretically, since " difficulty " has many possible meanings, we use OOD rule compositions as a tractable proxy in our controlled setting. In a simplified cross-entropy + weight-decay model with kernelized representation dynamics, we prove a finite-horizon two-phase trajectory: concentration-based sparsity first increases during early feature selection, while density later recovers for inputs that receive sufficient training support. This provides a formal mechanism for our consolidation view: later in training, mastered inputs acquire dense, distributed representations, while inputs outside the learned regime remain comparatively sparse because this densification process is incomplete.

Finally, our finding of the sparsity–difficulty regularity is also actionable. We illustrate this with *Sparsity-Guided Curriculum In-Context Learning* (SG-ICL), which uses sparsity to select few-shot demonstrations whose difficulty matches the query. On MATH-500, SG-ICL outperforms all major baselines, indicating that sparsity carries information that inference-time strategies can exploit.

In summary, we organize our study around three core research questions:

- **RQ1:** How does the geometry of the last hidden state evolve as tasks become increasingly difficult?
- **RQ2:** How does this sparsity–difficulty regularity arise during training?
- **RQ3:** How can this sparsity signal be practically leveraged to enhance model reasoning capabilities?

Guided by these questions, our primary findings are summarized as follows:

Sparsity increases with difficulty in a robust and controlled manner. We uncover a consistent relationship where last hidden state activations become sparser as question difficulty increases, across multiple reasoning benchmarks including MATH-500 (Lightman et al., 2024), DeepMATH-103K (He et al., 2026), LongReason (Ling et al., 2025), ConflictReason (Wang et al., 2024c), and additional settings reported in the appendix. To enable precise control over difficulty, we introduce **MMLU-Robust**, a benchmark that induces graded difficulty via controlled distractor augmentation.

Learning dynamics connect density to familiarity. We identify a fundamental link between representation density and data familiarity. By pretraining a small language model from scratch on a controlled synthetic knowledge-graph corpus and tracking hidden-state sparsity across checkpoints, we observe a two-phase learning dynamic: representations first become more concentrated during early feature selection, and then become denser as the model masters the training distribution. Equivalently, concentration-based metrics such as Top-k Energy exhibit an inverted-U trajectory, while density-oriented metrics exhibit the corresponding recovery. This suggests that high activation density is a learned attribute of familiar inputs, while harder or less familiar inputs receive weaker consolidation and remain comparatively sparse. We further provide a **theoretical derivation** that explains this two-phase sparsity–density trajectory.

2 Preliminaries

Experimental Setup. We conduct all experiments under the *implicit reasoning setting*. Although Chain-of-Thought (CoT; Wei et al., 2022) prompting also produces observable sparsity patterns, its effects are noticeably less stable. In contrast, the implicit reasoning setup yields clearer and more consistent activation dynamics, allowing us to present more interpretable visualizations. The prompt we use can be found in Appendix A.5. We focus primarily on the *last hidden state* for the same reason. Although related sparsity trends are also present in intermediate layers, they are typically weaker and less consistent across tasks and models. By contrast, the final layer exhibits the clearest and most stable pattern, making it the most interpretable layer for our analysis. The details of Layer Selection can be found in Appendix A.8. For model selection, we evaluate representative open-source LLMs, including Qwen2.5-1.5B/3B/7B/14B/32B/70B, Llama-3.2-1B/3B, and Llama-3.1-70B.[3]

Last Hidden State. We consider a transformer-based LM $f_\theta(\cdot)$ with L stacked decoder layers, where each layer ℓ consists of a multi-head self-attention ($\mathrm{Attn}^{(\ell)}(\cdot)$) block and a feed-forward network ($\mathrm{MLP}^{(\ell)}(\cdot)$). For an input token sequence $x = (x_1, x_2, \ldots, x_T)$, we denote the hidden representation of token x_t at layer ℓ as $h_t^{(\ell)} \in \mathbb{R}^d$, where d is the hidden dimension. Each layer maintains a residual connection, and its output is computed as:

$$h_{\text{mid}} = h_t^{(\ell)} + \mathrm{Attn}^{(\ell)}\big(h_t^{(\ell)}\big); \quad h_t^{(\ell+1)} = h_t^{(\ell)} + \mathrm{Attn}^{(\ell)}\big(h_t^{(\ell)}\big) + \mathrm{MLP}^{(\ell)}\big(h_{\text{mid}}\big). \tag{1}$$

We focus on the output of each layer in the residual stream, which stores contextualized information aggregated across all tokens in the input prompt. Formally, we use the representation at the final layer and the final input token:

$$h_T^{(L)} = f_\theta^{(L)}(x_{1:T})[-1], \tag{2}$$

which we refer to as the *last hidden state*. This representation serves as the latent state used for next-token prediction and downstream tasks.

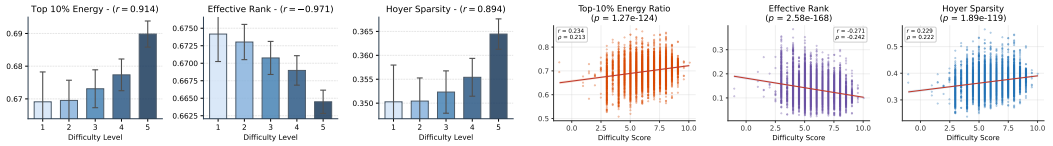
Sparsity. Prior work has shown that neural networks often exhibit concentration in their representations or effective parameterization. For example, fine-tuning can occur in low-dimensional subspaces (Aghajanyan et al., 2021), motivating parameter-efficient tuning such as LoRA (Hu et al., 2022). In this work, we study the *representational sparsity* of the last hidden state $h = h_T^{(L)} \in \mathbb{R}^d$, defined as the extent to which a small subset of coordinates carries most of the activation mass.

Our primary sparsity metric is *Top-k Energy*. For a given percentage k, let $\mathcal{T}_k(h)$ denote the indices of the largest $k\%$ coordinates of h in absolute value. We define

$$S_{\text{Top-}k}(h) = \frac{\sum_{i \in \mathcal{T}_k(h)} h_i^2}{\sum_{i=1}^d h_i^2}. \tag{3}$$

This metric measures how much of the total activation energy is captured by the largest coordinates. Since it is a ratio of squared energies, it is invariant to global rescaling of the hidden state. Higher Top-k Energy indicates that activation mass is more concentrated in a small number of dimensions and therefore corresponds to higher sparsity. Additional metrics are reported in Appendix A.4.

[3]For brevity, we present Qwen2.5 results in the main text and defer results for other models to the appendix.



(a) **MATH-500.** Sparsity strongly ($|r| > 0.89$) correlates with MATH-500 difficulty levels.

(b) **DeepMATH-103K.** Sparsity across difficulty levels; all results significant at $p < 10^{-117}$.

Figure 2: **Overview of Sparsity Analysis.** Both subfigures show that sparsity significantly correlates with difficulty for `Qwen2.5-7B` across two datasets: MATH-500 (left) and DeepMATH-103K (right).

Difficulty Our object of study is **difficulty**: the property that makes some inputs harder for the model than others. Difficulty has no single rigorous formalization for LLMs: pretraining data is opaque, and "harder" can mean many things at once. Rather than commit to one definition, we study difficulty along four interpretable axes, each capturing a distinct form of stress on the model: (1) **Reasoning complexity**: instances requiring deeper or multi-step inference (e.g., higher-difficulty math problems in MATH-500 and DeepMATH-103K); (2) **Distractor interference**: multiple-choice items with a larger number of plausible distractors, expanding the solution space; (3) **Knowledge conflict**: prompts whose context contradicts the model's parametric knowledge, forcing it to reconcile inconsistent signals; (4) **Context length**: inputs with longer or more compositional contexts that demand longer-range integration. These four axes are not equivalent forms of the same construct; they each apply a different kind of pressure to the model. The fact that they all yield the same last-layer signature is part of what we find, not a framing assumption.

3 RQ1: How does the geometry of the last hidden state evolve as tasks become increasingly difficult?

In this section, we present results addressing our first research question: *How do last hidden states behave when models encounter hard questions?* We measure sparsity metrics across tasks with varying levels of difficulty along four dimensions: reasoning complexity (§3.1), answer choice expansion (§3.2), knowledge conflict (§3.3), and context length (§3.4). Across all four settings, we observe the same consistent pattern: the harder the task, the sparser the last hidden state.

3.1 Sparsity *vs* Reasoning Complexity

We first examine reasoning complexity, asking whether the harder-is-sparser pattern appears consistently across both discretely annotated math difficulty levels and continuous difficulty scores.

MATH-500 (Hendrycks et al., 2021), a subset of the MATH benchmark with explicit difficulty annotations, serves as our first testbed. It contains 500 problems spanning five graded levels, from basic arithmetic (Level 1) to competition mathematics (Level 5). For each problem, we extract the last hidden state of the last token after applying a chat-template prompt to `Qwen2.5-7B`, and compute Top-10% Energy, Effective Rank, and Hoyer Sparsity. As shown in Figure 2a, sparsity tracks difficulty monotonically across all three metrics, with Pearson $|r| > 0.89$ on every panel.

To verify the pattern is not an artifact of the small benchmark's coarse 5-level annotation, we further evaluate on DeepMath-103K (He et al., 2026), a large-scale verifiable math reasoning dataset with continuous difficulty scores spanning roughly $[1, 10]$. We randomly sample 10,000 problems (seed 42) and evaluate `Qwen2.5-7B`, extracting the last-layer hidden state of the last token after a chat-template prompt followed by "Please give me the final answer." Figure 2b plots three sparsity metrics (Top-5% Energy, Top-10% Energy, and Effective Rank) against the continuous difficulty score, with linear fits and 95% confidence bands. Across all three metrics, both Pearson r and Spearman ρ have the expected signs (positive for Top-k Energy and Hoyer Sparsity, negative for Effective Rank); all Pearson p-values fall below 10^{-119}, and the same trend holds under non-parametric ranking. The dataset-level pattern is consistent with MATH-500: as the continuous difficulty score grows, the last hidden state becomes systematically sparser.

The same trend extends to other model families and sizes as shown in Appendix A.7.

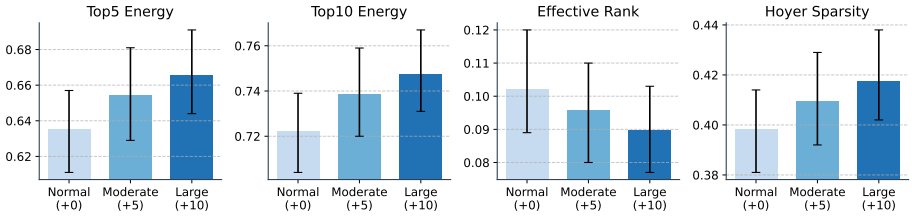
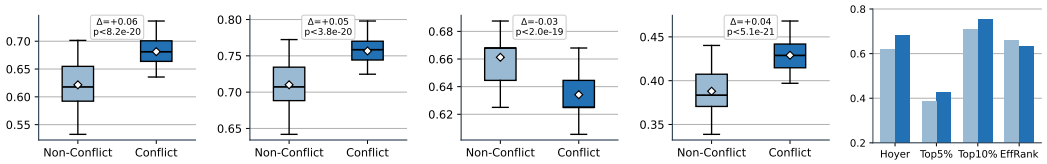


Figure 3: **Sparsity Metrics under Answer Choice Expansion.** Bar plots show mean sparsity across 14 disciplines for four metrics under Normal (+0), Moderate Expansion (+5), and Large Expansion (+10) on `Qwen2.5-3B`. Error bars indicate the minimum and maximum across disciplines. Increasing task difficulty leads to higher sparsity.



(a) Top-5% (b) Top-10% (c) Eff. Rank (d) Hoyer (e) Summary

Figure 4: **Sparsity Differences under Knowledge Conflict.** Across four sparsity metrics, the harder *conflict* (■) condition yields consistently sparser last-layer representations than the *non-conflict* (■) condition on `Qwen2.5-3B`. Arrows indicate the sparsity direction of each metric.

3.2 Sparsity *vs* Answer Choice Expansion

In this setting, we add plausible but incorrect distractor options to multiple-choice questions presented to the LLM. This makes the task strictly harder *ceteris paribus*, as the question content remains unchanged. This provides a controlled way to increase distractor interference while keeping the question stem unchanged. The setup remains simple and controlled, allowing us to study difficulty without confounding textual variation and to isolate the effect of difficulty.

Specifically, we use MMLU-Pro (Wang et al., 2024d), a multi-domain multiple-choice benchmark covering 14 academic disciplines. The original dataset provides ten answer options, which we use as the baseline **Normal** (+0) configuration. We introduce two harder configurations by adding plausible distractors: **Moderate Expansion** (+5), with five additional distractors (15 options total), and **Large Expansion** (+10), with ten additional distractors (20 options total). Question content is unchanged across configurations, isolating the effect of difficulty from textual variation. Together, these three configurations form **MMLU-Robust**; see Appendix A.9 for details.

Figure 3 presents the experimental results, confirming increased sparsity of final-layer hidden states when the tasks become more difficult. We analyzed four key sparsity metrics across all 14 academic areas in MMLU-Pro: Top-5% Energy, Top-10% Energy, Effective Rank, and Hoyer Sparsity. Across virtually all academic disciplines and for all four sparsity metrics, we observe a *consistent, monotonic trend*: as the question difficulty increases by adding distractors (moving from Normal (+0) $\rightarrow$ Moderate (+5) $\rightarrow$ Large (+10)), the last hidden state exhibits higher sparsity. A detailed per-discipline breakdown, along with results for additional models, is provided in Appendix A.9.2, confirming the same trends.

3.3 Sparsity *vs* Knowledge Conflict

Moreover, we use the Wang et al.'s (2024c) knowledge conflict dataset to further validate our hypothesis, which labels samples as conflict or non-conflict by pairing each question with either a truthful or a counterfactual context. In conflict scenarios, the provided context contradicts the model's parametric knowledge. Such conflicts force the model to reconcile external contextual evidence with its parametric knowledge, creating distributional stress.

As shown in Figure 4, conflict cases consistently exhibit higher sparsity across all four metrics. Defining Δ as the Conflict mean minus the Non-Conflict mean, the Top-5% Energy (a), Top-10% Energy (b), Effective Rank (c), and Hoyer Sparsity (d) changes are approximately $+0.06$, $+0.05$,

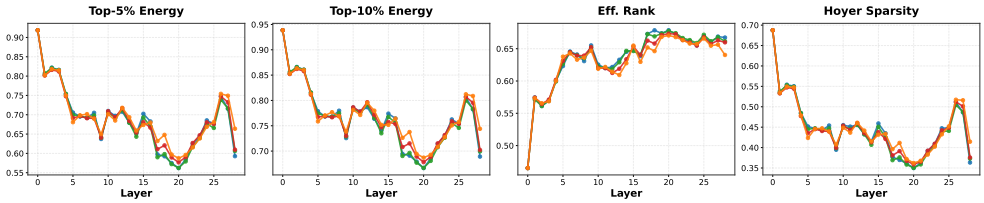


Figure 5: **Layer-wise Sparsity across Context Lengths.** While intermediate layers show minimal variation across contexts, the final layers exhibit sharp divergence: longer contexts consistently produce sparser representations. The experiment is conducted on LONGREASONQA (Ling et al., 2025), where the background context length can be controlled. Line color indicates the input context length: 8K, 16K, 32K, 64K.

-0.03, and $+0.04$, respectively. Higher Top-k Energy and Hoyer Sparsity, together with lower Effective Rank, indicate more concentrated representations. These differences are *statistically significant*, as confirmed by a paired Student's (1908) t-test, with all p-values $< 2.0 \times 10^{-19}$. The Summary bar chart (e) synthesizes these findings, clearly illustrating that the latent last hidden representations for conflict instances are generally more compressed and concentrated on fewer dimensions. We provide additional analyses across more LLM families, more model sizes, and more knowledge conflict settings (e.g., the related *distraction* phenomenon; Niu et al. (2025)) in Appendix A.6.

3.4 Sparsity *vs* Long Context Reasoning

For the long-context reasoning setting, as shown in Figure 5, we use LongReason (Ling et al., 2025). This long context dataset offers controllable context lengths and incorporates diverse, realistic reasoning tasks. By analyzing the last hidden states across these controlled length variations, we discovered that the sparsity of the hidden states increases as the input text length increases, as Figure 5. This finding further supports our hypothesis that increasing task difficulty consistently drives the model towards a state of sharper activation compression in the last hidden state. In fact, the sparsity phenomenon induced by harder inputs is primarily a *terminal behavior*. While intermediate representations remain relatively stable, the decisive shift in activation density between easier and harder inputs exclusively emerges in the final layer. We provide additional long-context analyses in Appendix A.10 and layer-level analyses in Appendix A.8 to discuss how the final layer shows the most pronounced and consistent sparsity trend across all the layers.

4 RQ2: How does this sparsity–difficulty regularity arise during training?

We now examine how the sparsity–difficulty connection develops during pretraining by training a toy-sized LM from scratch on synthetic data. In this section, we first describe the construction of the synthetic pre-training dataset (§4.1), then analyze the pre-training results (§4.2), and finally connect these dynamics to the emergence of the sparsity–difficulty relationship through a combined empirical and theoretical account (§4.3).

4.1 Controlled Environment

We create a synthetic knowledge graph dataset that allows precise control over reasoning difficulty and distributional stress. The setting is inspired by Wang et al. (2024a, 2025a); Yin and Wang (2025); Ye et al. (2025), but we focus on sparsity–difficulty patterns in this controlled symbolic environment rather than the emergence of implicit reasoning.

We validate this mechanism by analyzing representations during pretraining. Even without task-specific fine-tuning, models pretrained solely on next-token prediction exhibit the same sparsity-difficulty pattern, confirming that this is a fundamental property of the language model.

Specifically, instead of relying on real-world knowledge graphs (where structural and semantic factors are entangled), we generate an artificial graph governed by a set of logical rules. Each rule defines

how relations can be composed, e.g.,

$$r_3(x, z) \Leftarrow r_1(x, y) \wedge r_2(y, z), \tag{4}$$

ensuring that the resulting graph encodes multi-hop reasoning dependencies. Based on these rules, we construct the dataset in a three-stage process:

Preliminaries. We define a Knowledge Graph (KG) as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. $\mathcal{T} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ represents the set of facts (triples). The problem difficulty is controlled by logical rules of varying lengths L.

Graph Construction. We construct the controlled knowledge graph $\mathcal{G}$ through a systematic generation pipeline. We first initialize the vocabulary of entities $\mathcal{E} = \{e_1, \ldots, e_N\}$ and relations $\mathcal{R} = \{P_1, \ldots, P_N\}$, along with a set of K randomly sampled acyclic logic rules (lengths $L \in [L_{\min}, L_{\max}]$). The graph is seeded with random *atomic triples* (explicit facts). Subsequently, we exhaustively apply the logic rules to these atomic triples to infer all possible *deduced triples* (implicit facts). This process yields a dual-structure graph containing both direct observations and edges logically entailed by reasoning chains.

Sequence Serialization. To adapt the structured graph data for a language model, we linearize each triple $(h, r, t) \in \mathcal{T}$ into a token sequence (e.g., "e_1 P_5 e_2"). The model is trained using the standard auto-regressive objective to predict the tail entity t given the head h and relation r.

Controlled Data Splitting. To distinguish between rote memorization and logical generalization, we employ a rigorous data splitting strategy. Specifically, we partition the triples $\mathcal{T}$ into a training set $\mathcal{D}_{\text{train}}$ and three distinct evaluation subsets: $\mathcal{D}_{\text{mem}}$ for assessing rote memorization, $\mathcal{D}_{\text{ID}}$ for testing in-distribution rule composition, and $\mathcal{D}_{\text{hard}}$ for probing out-of-distribution robustness.

Appendix A.11 provides additional details on the synthetic data construction and the experimental setup used for pre-training the toy-sized LM.

4.2 Pretraining Results

Table 1: **Emergence of the Sparsity Phenomenon.** We report metrics across 3 levels. The gradient blue shading, observed uniquely in Trans-32-32, illustrates that the "harder-is-sparser" mechanism is a learned behavior.

Model	Params	Training Steps	Accuracy (%, ↑)			Top-5 Energy (%, ↑)			Top-10 Energy (%, ↑)			Eff. Rank (↓)		
			$\mathcal{D}_{\text{mem}}$	$\mathcal{D}_{\text{ID}}$	$\mathcal{D}_{\text{OOD}}$	$\mathcal{D}_{\text{mem}}$	$\mathcal{D}_{\text{ID}}$	$\mathcal{D}_{\text{OOD}}$	$\mathcal{D}_{\text{mem}}$	$\mathcal{D}_{\text{ID}}$	$\mathcal{D}_{\text{OOD}}$	$\mathcal{D}_{\text{mem}}$	$\mathcal{D}_{\text{ID}}$	$\mathcal{D}_{\text{OOD}}$
Trans-8-8	0.02B	2500	64.00	42.00	31.00	26.11	26.03	26.26	42.12	42.00	42.34	0.502	0.504	0.500
Trans-16-8	0.04B	2000	71.00	50.00	24.00	25.96	25.94	26.06	41.98	41.98	42.15	0.503	0.503	0.501
Trans-16-16	0.17B	2000	85.00	56.00	44.00	27.70	27.70	27.78	43.70	43.66	43.76	0.483	0.484	0.482
		2500	81.00	58.00	37.00	27.64	27.58	27.76	43.56	43.52	43.69	0.484	0.485	0.483
Trans-32-32	1.34B	1500	78.00	57.00	37.00	30.28	30.54	30.84	46.44	46.72	47.04	0.454	0.451	0.448
		2000	83.00	63.00	34.00	30.00	30.23	30.55	46.11	46.39	46.70	0.458	0.455	0.452
		2500	81.00	63.00	38.00	29.60	29.75	30.11	45.67	45.86	46.22	0.463	0.461	0.457

We use **Trans-L-H** to denote a Transformer with L layers and H attention heads (e.g., Trans-8-8 has 8 layers and 8 heads). We report Accuracy, Top-5/10 Energy, and Effective Rank across three difficulty levels: Easy/Medium/Hard, which correspond to $\mathcal{D}_{\text{mem}}$, $\mathcal{D}_{\text{ID}}$, $\mathcal{D}_{\text{hard}}$ respectively. The findings highlight a clear correlation between model scale, training steps, and the emergence of sparsity. These results demonstrate that the relationship between task difficulty and sparsity is a learned phenomenon: it becomes consistent only once the model exits underfitting and reaches sufficient training convergence and capacity. The absence of a clear trend in smaller models may be attributed to the high data complexity relative to their limited capacity. We provide a detailed discussion on the graph data complexity in Appendix A.11.

We find that the previously reported "harder-is-sparser" effect is not only present in instruction-tuned modern LLMs, but also arises in standard pretrained Transformer models, where it emerges as a pretraining-level property. In particular, this pattern surfaces only once the model has sufficient capacity and reaches convergence (Table 1).

4.3 Learning Dynamics: From Feature Selection to Consolidation

To study how sparsity evolves during training, we evaluate checkpoints every 200 steps over a 5,000-step training run and track the average sparsity of representations on a fixed set of training samples. Sparsity is measured by the fraction of activation energy captured by the Top-k dimensions.

Nevertheless, Table 1 and Figure 6 reveal a counter-intuitive temporal evolution of sparsity. Focusing on the **Trans-32-32** model, we observe that sparsity does not simply increase throughout training. Instead, the Top-5% Energy first rises sharply during early training, increasing from roughly 0.275 at initialization to a peak of about 0.318 around 1300 steps, while accuracy simultaneously improves from near-random performance to above 60%. After this early phase, however, Top-5% Energy gradually decreases to around 0.278 by 5000 steps, even as accuracy remains high and stable at approximately 75%. This suggests that, once the model has learned the core task structure, it no longer relies on a few sharply activated coordinates; instead, it develops a more distributed and robust representation for familiar inputs. A similar pattern appears for Top-10% Energy, which peaks around 0.483 before falling back to roughly 0.442 (with full curves reported in Appendix A.11.3).

As illustrated in Figure 6, the training process therefore exhibits a distinct two-phase dynamic.

Phase I: Feature Selection. During early training, Top-k Energy rises sharply, from about 0.275 at initialization to a peak of roughly 0.315 around 1.4K training steps. Over the same period, accuracy increases rapidly from near 10% to above 60%. This indicates that the model increasingly concentrates activation mass on a small subset of dimensions while learning the core task structure, suggesting an early feature-selection phase in which task-relevant features are amplified and irrelevant or unstable directions are suppressed.

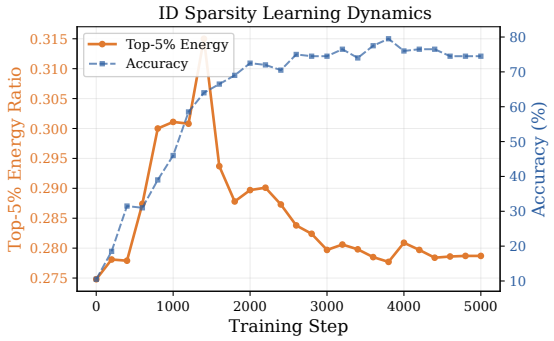


Figure 6: **Learning dynamics via Top-k Energy.** Top-k Energy first rises during early training, indicating feature selection and increasing activation concentration, and then gradually decreases while accuracy remains high, suggesting feature consolidation into a more distributed and robust representation.

Phase II: Feature Consolidation. After this initial concentration phase, Top-k Energy gradually decreases from its peak near 0.315 to about 0.278 by 5K steps, while accuracy remains high and stabilizes around 75–80%. This decrease should not be interpreted as a loss of learned structure; rather, it suggests *representation maturation*. Once the core task structure has been learned, the model no longer relies on a few sharply activated coordinates but distributes information across a broader set of reliable features. This consolidation yields a smoother and more robust representation for familiar inputs, helping explain why in-distribution examples exhibit denser final-layer representations than harder OOD cases.

We give a theoretical justification for this two-phase sparsity–density dynamic using a simplified cross-entropy + weight-decay model with kernelized representation dynamics in Appendix A.2. We explain this as a shift from early weight-decay-driven feature contraction, which decreases normalized ℓ_1, to later gradient-aligned updates on persistently hard examples that increase it.

Intuition: Learning Reduces Sparsity for Familiar Inputs

Learning can be viewed as a representation densification process, for a given input:

$$\text{more training exposure} \;\rightarrow\; \text{better learned pattern} \;\rightarrow\; \text{lower sparsity}.$$

Therefore, well-learned in-distribution inputs tend to produce dense final-layer representations. For unseen or rarely observed inputs, this densification process is incomplete:

$$\text{unseen / rarely seen pattern} \;\rightarrow\; \text{weakly learned pattern} \;\rightarrow\; \text{higher sparsity}.$$

Therefore, high activation density emerges as a *learned privilege* of data familiarity. Since the language model learns rich distributed representations specifically for patterns it has mastered, it maintains high activation density in the last hidden state when facing in-distribution samples. When an input lies farther from the model's familiar distribution, the model is less able to recruit the broad distributed feature subspaces used for in-distribution reasoning. Consequently, activations concentrate on a narrower set of dimensions, leading to a sparser final-layer representation. This mechanism offers a natural explanation for our core observation: *the farther the shift, the sparser the representation*. A fuller discussion, together with additional training curves across Top-k Energy, Effective Rank, and accuracy on different model sizes, is provided in Appendix A.11.3.

5 RQ3: How can this sparsity signal be practically leveraged to enhance model reasoning capabilities?

Based on our insight that the last hidden representation sparsity serves as a reliable signal for task complexity, we propose a novel few-shot example selection strategy for LLM reasoning: Sparsity-Guided Curriculum In-Context Learning (SG-ICL). While standard in-context learning often selects demonstrations randomly or relies solely on semantic similarity, such as Rubin et al. (2022); Ma et al. (2023) or through clustering-based methods such as Auto-CoT (Zhang et al., 2023), it ignores the cognitive load required to process them. We argue that an effective prompt should act as a *developmental curriculum*, guiding the model from rote, simple pattern matching to complex reasoning.

First, we assess the difficulty of all candidate examples in the demonstration pool. We compute the sparsity score $S(x)$ for each example x using the Top-k Energy of its last hidden state $\mathbf{h}_L$. We then sort the examples and group them into K distinct difficulty levels (or bins) $\mathcal{B}_1, \ldots, \mathcal{B}_K$:

$$S(x) = S_{\text{Top-}k}(\mathbf{h}_L(x)), \text{with } x \in \mathcal{B}_k \Leftrightarrow \tau_{k-1} \leq S(x) < \tau_k \tag{5}$$

Here, τ represents the threshold boundaries derived from the percentiles of the sparsity distribution, allowing us to categorize examples from easy to hard systematically. During the inference phase for a specific test query x_{query}, our strategy employs a dual-criteria selection process:

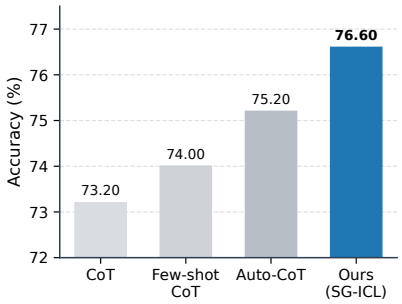


Figure 7: **Performance Comparison of Reasoning Strategies.** Our SG-ICL achieves an accuracy of 76.60% on MATH-500 with `Qwen2.5-7B`, substantially outperforming standard CoT baselines (zero-shot and few-shot) and the Auto-CoT baseline (75.20%).

1. Semantic Filtering: We first retrieve a candidate set $\mathcal{N}_{sem}$ consisting of the top-N examples most semantically similar to x_{query} by leveraging Sentence-BERT (Reimers and Gurevych, 2019) to compute dense vector embeddings.

2. Difficulty Matching: We then calculate the sparsity $S(x_{query})$ of the incoming query x_{query} to identify its corresponding difficulty level $\mathcal{B}_{target}$. From $\mathcal{N}_{sem}$, we specifically select k examples that not only maintain high semantic similarity but also align with the target difficulty level (i.e., falling within $\mathcal{B}_{target}$).

This process ensures the few-shot prompt is both contextually relevant and cognitively aligned with the query's content and complexity.

To validate this approach, we evaluate it on `Qwen2.5-7B` using MATH-500. Empirical results on the MATH-500 dataset demonstrate that our Sparsity-Guided Curriculum strategy outperforms strong baselines, as illustrated in Figure 7.

6 Conclusion

In this work, we identify a systematic connection between the internal representation geometry of LLMs and the difficulty of the tasks they face. Through analysis across diverse models, benchmarks, and controlled settings, we validate the phenomenon that *the farther the shift, the sparser the representation.* Our findings show that this sparsification is not an incidental artifact of a single dataset or metric, but a robust representational pattern that appears most clearly in the final transformer

layers. We further show that this pattern is tied to learning dynamics: as models acquire task structure, familiar inputs can recruit broader and denser feature subspaces, whereas harder or more distributionally shifted inputs activate a narrower set of dimensions.

Author Contributions

Mingyu Jin conducted the experiments. Jingcheng Niu was responsible for scaling up the experiments to larger settings and polish the writing. Yutong Yin proposed the initial idea and developed the toy model and its theoretical analysis.

Acknowledgments

We thank Minghao Guo, Xi Zhu, Bangwei Guo, Mengru Wang, Fei Sun, Xin Xu, Fengzhuo Zhang, Zirui Liu, Fan Yang, Wenyue Hua, and Jinman Zhao for their help and valuable discussions.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 7319–7328.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-Thought Reasoning in the Wild is not Always Faithful. In *Workshop on Reasoning and Planning for Large Language Models*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning (ICML)*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1877–1901.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: Benchmarking Factuality Evaluation of Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The Lottery Ticket Hypothesis for Pre-trained BERT Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 16318–16352.

Nicola De Cao, Leon Schmid, Dieuwke Hupkes, and Ivan Titov. 2022. Sparse Interventions in Language Models with Differentiable Masking. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP)*, pages 16–27.

Branton DeMoss, Silvia Sapora, Jakob Foerster, Nick Hawes, and Ingmar Posner. 2025. The complexity dynamics of grokking. *Physica D: Nonlinear Phenomena*, page 134859.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy Models of Superposition. *Transformer Circuits Thread (Anthropic Technique report)*.

Nelson Elhage, Neel Nanda, Catherine Olsson, and 1 others. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread (Anthropic Technique report)*.

Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations (ICLR)*.

Sandeep Reddy Gantla. 2025. Exploring Mechanistic Interpretability in Large Language Models: Challenges, Approaches, and Insights. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, pages 1–8.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A Unifying Framework for Inspecting Hidden Representations of Language Models. In *International Conference on Machine Learning (ICML)*, pages 15466–15490.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP (ACL Workshop)*, pages 143–155.

Jiaojiao Han, Wujiang Xu, Mingyu Jin, and Mengnan Du. 2026. SAGE: An agentic explainer framework for interpreting SAE features in language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 483–495.

Song Han, Jeff Pool, John Tran, and William J Dally. 2015. Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1135–1143.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2026. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. In *The Fourteenth International Conference on Learning Representations (ICLR)*.

Zirui He, Mingyu Jin, Bo Shen, Ali Payani, Yongfeng Zhang, and Mengnan Du. 2025. SAE-SSV: Supervised steering in sparse representation spaces for reliable control of language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2207–2236.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research (JMLR)*, 22(241):1–124.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Ziwei Ji and Matus Telgarsky. 2019. The implicit bias of gradient descent on nonseparable data. In *Annual Conference Computational Learning Theory (COLT)*.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. 2025a. Disentangling memory and reasoning ability in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1681–1701.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025b. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. 2024. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Mingyu Lee, Jun Hyung Park, Junho Kim, Kang Min Kim, and Sang Keun Lee. 2022. Efficient Pre-training of Masked Language Model via Concept-based Curriculum Masking. In *2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7417–7427.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2025a. Long-context LLMs Struggle with Long In-context Learning. *Transactions on Machine Learning Research (TMLR)*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025b. From system 1 to system 2: A survey of reasoning large language models. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, and Sanjiv Kumar. 2023. The Lazy Neuron Phenomenon: On Emergence of Activation Sparsity in Transformers. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. 2025. Longreason: A synthetic long-context reasoning benchmark via context expansion. *arXiv preprint arXiv:2501.15089*.

James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. 2025. Training-Free Activation Sparsity in Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and 1 others. 2023. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning (ICML)*, pages 22137–22176.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5303–5315.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. 2019. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTAT) (AISTATS)*, pages 3051–3059.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. "Pre-training a BERT with Curriculum Learning by Increasing Block-Size of Input Text". In *"Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)"*, pages 989–996.

Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamid Reza Saghir, and Amir H. Abdi. 2025. Llama See, Llama Do: A Mechanistic Perspective on Contextual Entrainment and Distraction in LLMs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

nostalgebraist. 2020. interpreting gpt: the logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`. Accessed: 2024-05-20.

Bruno A Olshausen and David J Field. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context Learning and Induction Heads. *Transformer Circuits Thread*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1162–1172.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Yi Ren and Danica J. Sutherland. 2025. Learning Dynamics of LLM Finetuning. In *International Conference on Learning Representations (ICLR)*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL)*, pages 2655–2671.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. 2018. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research (JMLR)*, 19(70):1–57.

Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more"' in unsupervised dependency parsing. In *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 751–759.

Student. 1908. The Probable Error of a Mean. *Biometrika*, 6(1):1–25.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025. Tongyi deepresearch technical report. *ArXiv Preprint (Tongyi Qwen technical report)*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. 2024a. Grokking of implicit reasoning in transformers: a mechanistic journey to the edge of generalization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 95238–95265.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Mengru Wang, Yunzhi Yao, Ziwen Xu, Shuofei Qiao, Shumin Deng, Peng Wang, Xiang Chen, Jia-Chen Gu, Yong Jiang, Pengjun Xie, and 1 others. 2024b. Knowledge Mechanisms in Large Language Models: A Survey and Perspective. In *Findings of the Association for Computational Linguistics: (EMNLP Findings) (Findings of EMNLP)*, pages 7097–7135.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 44(9):4555–4576.

Xinyi Wang, Shawn Tan, Mingyu Jin, William Yang Wang, Rameswar Panda, and Yikang Shen. 2025a. Do Larger Language Models Imply Better Generalization? A Pretraining Scaling Law for Implicit Reasoning. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale*.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2024c. Resolving Knowledge Conflicts in Large Language Models. In *First Conference on Language Modeling (COLM)*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024d. MMLU-Pro: a more robust and challenging multi-task language understanding benchmark. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 95266–95290.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. 2025b. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *arXiv preprint arXiv:2504.09710*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 24824–24837.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 15366–15394.

Jing Xiong, Jianghan Shen, Fanghua Ye, Chaofan Tao, Zhongwei Wan, Jianqiao Lu, Xun Wu, Chuanyang Zheng, Zhijiang Guo, Min Yang, Lingpeng Kong, and Ngai Wong. 2025. "UNComp: Can Matrix Entropy Uncover Sparsity? — A Compressor Design from an Uncertainty-Aware Perspective". In *"Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)"*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. "Curriculum Learning for Natural Language Understanding". In *"Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)"*, pages 6095–6104.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering Large Pre-Trained Language Models to Follow Complex Instructions. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Jiaran Ye, Zijun Yao, Zhidian Huang, Liangming Pan, Jinxin Liu, Yushi Bai, Amy Xin, Liu Weichuan, Xiaoyin Che, Lei Hou, and Juanzi Li. 2025. How do Transformers Learn Implicit Reasoning? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Yutong Yin and Zhaoran Wang. 2025. Are Transformers Able to Reason by Connecting Separated Knowledge in Training Data? In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Qingcheng Zeng, Mingyu Jin, Qinkai Yu, Zhenting Wang, Wenyue Hua, Guangyan Sun, Yanda Meng, Shiqing Ma, Qifan Wang, Felix Juefei-Xu, Fan Yang, Kaize Ding, Ruixiang Tang, and Yongfeng Zhang. 2025. Uncertainty quantification for multiple-choice questions is just one-token deep. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 5474–5478. Association for Computing Machinery.

Chong Zhang, Mingyu Jin, Dong Shu, Taowen Wang, Dongfang Liu, and Xiaobo Jin. 2024a. Target-driven attack for large language models. In *ECAI 2024*, pages 1752–1759.

Chong Zhang, Mingyu Jin, Qinkai Yu, Chengzhi Liu, Haochen Xue, and Xiaobo Jin. 2024b. Goal-guided generative prompt injection attack on large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 941–946. IEEE.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Haiyan Zhao, Fan Yang, Bo Shen, Himabindu Lakkaraju, and Mengnan Du. 2024b. Towards uncovering how large language model works: An explainability perspective. *arXiv preprint arXiv:2402.10688*.

Haiyan Zhao, Heng Zhao, Bo Shen, Ali Payani, Fan Yang, and Mengnan Du. 2025. Beyond Single Concept Vector: Modeling Concept Subspace in LLMs with Gaussian Distribution. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. 2024c. Implicit Geometry of Next-token Prediction: From Language Sparsity Patterns to Model Representations. In *First Conference on Language Modeling (COLM)*.

Yize Zhao and Christos Thrampoulidis. 2025a. Geometry of Concepts in Next-token Prediction: Neural-Collapse Meets Semantics. In *The Second Conference on Parsimony and Learning (CPAL)*.

Yize Zhao and Christos Thrampoulidis. 2025b. On the Geometry of Semantics in Next-token Prediction. In *The First Workshop on the Interplay of Model Behavior and Model Internals (COLM Workshop)*.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2024. MathAttack: Attacking large language models towards math solving ability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19750–19758.

A Appendix

A.1 Normalized ℓ_1 representation sparsity.

We quantify hidden-state sparsity using the normalized ℓ_1 statistic. For a last hidden state $h \in \mathbb{R}^d$, we define $S_{L_1}(h) = \frac{1}{d}\sum_{j=1}^{d} |h_j|$, which measures the average activation magnitude across coordinates. Smaller values indicate that more coordinates are close to zero and that the representation is more concentrated in a subset of dimensions. This statistic is especially suitable for dataset-level OOD analysis because sparsity changes may be weak or noisy for individual samples, but become stable after averaging across examples. It also admits a smooth proxy $N_\varepsilon(t)$, enabling the finite-horizon learning-dynamics analysis.

A.2 A finite-horizon two-phase sparsity–density certificate for the normalized ℓ_1 statistic

In Figure 6 and Figure 20, we observe a robust two-phase sparsity–density trajectory during learning: the normalized ℓ_1 magnitude of the last hidden state decreases early (*sparsification*), and later increases over another period (*densification*). Equivalently, concentration-based metrics such as Top-k Energy exhibit an inverted-U trajectory. This appendix provides a finite-horizon theoretical justification on a simplified learning task. Informally, the mechanism is: (i) early training is dominated by weight decay acting on a weak/near-zero head, which contracts feature magnitudes and thus decreases the normalized ℓ_1 statistic; (ii) if at some later time the features are already sparse enough but a nontrivial subset of examples remains persistently hard, then the loss gradient aligns with the active coordinates and dominates the contraction over a finite window, forcing the statistic to increase.

A.2.1 Simplified setting: data, model, objectives and gradients

We fix a training set $\{(x_i, y_i)\}_{i=1}^n$ with labels $y_i \in \{1, \dots, C\}$. Let $\theta(t)$ denote the parameters that generate the last-layer representation, and define

$$h_i(t) := h(x_i; \theta(t)) \in \mathbb{R}^d, \quad i = 1, \dots, n,$$

(e.g., the last hidden state $h_T^{(L)}$ in the main text, evaluated on the final token of x_i). Let $W(t) \in \mathbb{R}^{C\times d}$ denote the linear readout with rows $W_c(t) \in \mathbb{R}^d$. We express the output logits $z_i(t) \in \mathbb{R}^C$ by

$$z_i(t) := W(t)h_i(t), \quad z_{i,c}(t) = \langle W_c(t), h_i(t)\rangle.$$

Let $p_i(t) \in \Delta^{C-1}$ be the softmax probabilities:

$$(p_i(t))_c := \frac{e^{z_{i,c}(t)}}{\sum_{c'=1}^{C} e^{z_{i,c'}(t)}},$$

and let $e_{y_i} \in \mathbb{R}^C$ be the one-hot vector for label y_i.

Training objective (cross-entropy with parameter weight decay). Define the per-example cross-entropy loss

$$\ell_i(t) := \ell_{\mathrm{ce}}\big(z_i(t), y_i\big) = -\log\big((p_i(t))_{y_i}\big),$$

and the dataset-averaged cross-entropy

$$\mathcal{L}_{\mathrm{ce}}(\theta, W) := \frac{1}{n}\sum_{i=1}^{n} \ell_i(t).$$

We regularize the *parameters* that generate the representation (and optionally the readout) via ℓ_2 weight decay:

$$\mathcal{L}(\theta, W) := \mathcal{L}_{\mathrm{ce}}(\theta, W) + \frac{\lambda_h}{2}\|\theta\|_2^2 + \frac{\lambda_W}{2}\|W\|_F^2,$$

where $\lambda_h \geq 0$ is the decay coefficient for θ and $\lambda_W \geq 0$ is optional readout decay. In the analysis below, the effect of weight decay on θ is modeled at the feature level as a contraction term $-\lambda_h h_i(t)$ (plus a residual), captured by the induced dynamics in (7).

Per-example representation gradient. Define the gradient of $\ell_i(t)$ with respect to the representation:

$$g_i(t) := \nabla_{h_i} \ell_i(t) = W(t)^\top \big(p_i(t) - e_{y_i}\big) \in \mathbb{R}^d.$$

Since $\|p_i(t) - e_{y_i}\|_2 \le \sqrt{2}$, we have

$$\|g_i(t)\|_2 \le \sqrt{2} \|W(t)\|_{2\to 2}, \tag{6}$$

where $\|\cdot\|_{2\to 2}$ denotes the operator (spectral) norm.

A.2.2 Induced (kernelized) representation dynamics with weight decay

Let $K(t) \in \mathbb{R}^{n\times n}$ be a symmetric matrix (a "kernel") with entries $K_{ik}(t)$. We assume the induced dynamics of $h_i(t) = h(x_i; \theta(t))$ has the form

$$\dot{h}_i(t) = -\frac{1}{n} \sum_{k=1}^{n} K_{ik}(t) g_k(t) - \lambda_h h_i(t) + r_i(t), \tag{7}$$

where $\lambda_h \ge 0$ is the (effective) feature weight decay coefficient and $r_i(t)$ is a residual term (e.g., due to non-frozen lower layers), assumed bounded on the horizons of interest.

Remark 1 (A simple setting where (7) holds exactly)**.** *Consider a frozen backbone feature map $\phi : \mathcal{X} \to \mathbb{R}^m$ and a trainable linear representation head $V(t) \in \mathbb{R}^{d\times m}$ such that*

$$\theta(t) \equiv V(t),$$
$$h_i(t) = h(x_i; \theta(t)) = V(t)\phi(x_i) =: V(t)\phi_i.$$

Let the logits be $z_i(t) = W(t)h_i(t)$ and define $\ell_i(t) = \ell_{\mathrm{ce}}(z_i(t), y_i)$ as above. If we train V by gradient flow on $\mathcal{L}_{\mathrm{ce}}(V, W)$ with ℓ_2 weight decay $\frac{\lambda_h}{2}\|V\|_F^2$ (while W may also be trained), then a direct chain-rule computation gives

$$\dot{V}(t) = -\frac{1}{n} \sum_{k=1}^{n} g_k(t)\phi_k^\top - \lambda_h V(t),$$

which leads to

$$\begin{aligned}\dot{h}_i(t) &= \dot{V}(t)\phi_i \\ &= -\frac{1}{n} \sum_{k=1}^{n} \langle \phi_i, \phi_k \rangle g_k(t) - \lambda_h h_i(t).\end{aligned}$$

Therefore (7) *holds with the (time-independent) kernel $K_{ik} = \langle \phi_i, \phi_k \rangle$ and residual $r_i(t) \equiv 0$.*

A.2.3 Normalized ℓ_1 sparsity and a smooth proxy

Recall the normalized ℓ_1 statistic (main text):

$$S_{L_1}(h) = \frac{1}{d} \sum_{j=1}^{d} |h_j|.$$

To obtain a differentiable drift identity, we introduce a smoothed proxy.

Smoothed primitives. Fix $\varepsilon > 0$ and define, for $u \in \mathbb{R}$,

$$\rho_\varepsilon(u) := \sqrt{u^2 + \varepsilon^2},$$
$$\psi_\varepsilon(u) := \frac{u}{\sqrt{u^2 + \varepsilon^2}}.$$

For $h \in \mathbb{R}^d$, define $\psi_\varepsilon(h) \in \mathbb{R}^d$ entrywise: $(\psi_\varepsilon(h))_j := \psi_\varepsilon(h_j)$.

Dataset-averaged smoothed ℓ_1 statistic. Define

$$N_\varepsilon(t) := \frac{1}{n} \sum_{i=1}^{n} \frac{1}{d} \sum_{j=1}^{d} \rho_\varepsilon\big(h_{i,j}(t)\big). \tag{8}$$

Note that $N_\varepsilon(t) \to \frac{1}{n}\sum_{i=1}^{n} S_{L_1}(h_i(t))$ as $\varepsilon \to 0$.

A.2.4 Drift identity for N_ε

Define the drift components

$$D_\varepsilon(t) := \frac{1}{n^2 d} \sum_{i=1}^{n} \sum_{k=1}^{n} K_{ik}(t) \big\langle \psi_\varepsilon(h_i(t)), g_k(t) \big\rangle, \tag{9}$$

$$C_\varepsilon(t) := \frac{\lambda_h}{nd} \sum_{i=1}^{n} \sum_{j=1}^{d} \frac{h_{i,j}(t)^2}{\sqrt{h_{i,j}(t)^2 + \varepsilon^2}},$$

$$R_\varepsilon(t) := \frac{1}{nd} \sum_{i=1}^{n} \big\langle \psi_\varepsilon(h_i(t)), r_i(t) \big\rangle.$$

Lemma 1 (Exact drift identity)**.** *For all t for which the trajectories are differentiable,*

$$\dot{N}_\varepsilon(t) = -D_\varepsilon(t) - C_\varepsilon(t) + R_\varepsilon(t).$$

Proof. Differentiate (8):

$$\begin{aligned} \dot{N}_\varepsilon(t) &= \frac{1}{nd} \sum_{i=1}^{n} \sum_{j=1}^{d} \psi_\varepsilon(h_{i,j}(t)) \dot{h}_{i,j}(t) \\ &= \frac{1}{nd} \sum_{i=1}^{n} \big\langle \psi_\varepsilon(h_i(t)), \dot{h}_i(t) \big\rangle. \end{aligned}$$

Substitute (7) and identify terms using (9). □

Lemma 2 (Two-sided decay bound)**.** *For all t,*

$$\lambda_h \big(N_\varepsilon(t) - \varepsilon \big) \le C_\varepsilon(t) \le \lambda_h N_\varepsilon(t).$$

Consequently,

$$-D_\varepsilon(t) - \lambda_h N_\varepsilon(t) + R_\varepsilon(t) \le \dot{N}_\varepsilon(t) \le -D_\varepsilon(t) - \lambda_h \big(N_\varepsilon(t) - \varepsilon \big) + R_\varepsilon(t). \tag{10}$$

Proof. For any $u \in \mathbb{R}$,

$$0 \le \frac{u^2}{\sqrt{u^2 + \varepsilon^2}} \le \sqrt{u^2 + \varepsilon^2} = \rho_\varepsilon(u),$$

$$\frac{u^2}{\sqrt{u^2 + \varepsilon^2}} = \rho_\varepsilon(u) - \frac{\varepsilon^2}{\rho_\varepsilon(u)} \ge \rho_\varepsilon(u) - \varepsilon,$$

since $\rho_\varepsilon(u) \ge \varepsilon$. Summing over (i, j) and multiplying by $\lambda_h/(nd)$ yields the result, and then (10) follows from lemma 1. □

A.2.5 Phase I: sparsification (early-time decrease trend + hitting a low level)

Assumption 1 (Phase I boundedness on a budget horizon)**.** *Fix $T_{\mathrm{I}} > 0$. Assume that on $t \in [0, T_{\mathrm{I}}]$:*

(i) Kernel bound: $\kappa(t) := \max_{i,k} |K_{ik}(t)| \le \kappa_0$.

(ii) Head bound: $\|W(t)\|_{2\to 2} \le M_0$.

(iii) Residual bound: $|R_\varepsilon(t)| \le r_0$.

Lemma 3 (Uniform bound on $|D_\varepsilon|$ on Phase I)**.** *Under assumption 1, for all $t \in [0, T_{\mathrm{I}}]$, we have $|D_\varepsilon(t)| \le \kappa_0 \sqrt{2} M_0 / \sqrt{d}$.*

Proof. By definition,

$$|D_\varepsilon(t)| \le \frac{1}{n^2 d} \sum_{i,k} |K_{ik}(t)| \cdot \big| \langle \psi_\varepsilon(h_i(t)), g_k(t) \rangle \big|.$$

Since $|\psi_\varepsilon(u)| \le 1$, $\|\psi_\varepsilon(h_i)\|_2 \le \sqrt{d}$, hence $\big|\langle \psi_\varepsilon(h_i), g_k\rangle\big| \le \sqrt{d}\|g_k\|_2$. Using (6), $\|g_k\|_2 \le \sqrt{2}\|W\|_{2\to 2} \le \sqrt{2}M_0$, and $|K_{ik}(t)| \le \kappa_0$. Summing over i, k yields

$$\begin{aligned}|D_\varepsilon(t)| &\le \frac{1}{n^2 d} \cdot n^2 \cdot \kappa_0 \cdot \sqrt{d} \cdot \sqrt{2}M_0 \\ &= \kappa_0 \frac{\sqrt{2}M_0}{\sqrt{d}}.\end{aligned}$$

□

Lemma 4 (Phase I dynamic)**.** *Under assumption 1, define $B_0 := \kappa_0\sqrt{2}M_0/\sqrt{d} + r_0$. Then for all $t \in [0, T_{\mathrm{I}}]$, we have*

$$\dot{N}_\varepsilon(t) \le -\lambda_h\big(N_\varepsilon(t) - \varepsilon\big) + B_0, \tag{11}$$

and therefore

$$N_\varepsilon(t) - \varepsilon \le \big(N_\varepsilon(0) - \varepsilon\big)e^{-\lambda_h t} + \frac{B_0}{\lambda_h}\big(1 - e^{-\lambda_h t}\big). \tag{12}$$

Proof. From the upper comparison inequality (10),

$$\dot{N}_\varepsilon(t) \le -D_\varepsilon(t) - \lambda_h\big(N_\varepsilon(t) - \varepsilon\big) + R_\varepsilon(t).$$

Use $-D_\varepsilon(t) \le |D_\varepsilon(t)|$, lemma 3, and $|R_\varepsilon(t)| \le r_0$ to obtain (11). Solving the scalar comparison ODE yields (12) and the hitting-time claim. □

Corollary 1 (Phase I decrease trend + certified hitting time)**.** *From Lemma 4, we directly have that $\dot{N}_\varepsilon(t) < 0$ whenever $N_\varepsilon(t) > \varepsilon + \frac{B_0}{\lambda_h}$. Moreover, for any target level $L > \varepsilon + \frac{B_0}{\lambda_h}$, define the (comparison) hitting time*

$$t_L := \frac{1}{\lambda_h} \log\left(\frac{N_\varepsilon(0) - \varepsilon - \frac{B_0}{\lambda_h}}{L - \varepsilon - \frac{B_0}{\lambda_h}}\right),$$

(where the logarithm is well-defined when $N_\varepsilon(0) > \varepsilon + \frac{B_0}{\lambda_h}$). If $t_L \le T_{\mathrm{I}}$, then $N_\varepsilon(t_L) \le L$.

Remark 2 (What Phase I does and does not claim)**.** *Phase I certifies (i) an early-time decrease trend whenever N_ε is above $\varepsilon + \frac{B_0}{\lambda_h}$, and (ii) the existence of a time $t_L \le T_{\mathrm{I}}$ at which $N_\varepsilon(t_L)$ reaches any prescribed level $L > \varepsilon + \frac{B_0}{\lambda_h}$. It does not claim monotone decrease beyond the certified horizon.*

A.2.6 Phase II: densification on a prescribed finite horizon

Assumption 2 (Phase II one-shot densification assumption)**.** *There is a start time $\tau > 0$ and horizon length $H > 0$ that make the following conditions hold for every $t \in [\tau, \tau + H]$.*

(A) Boundedness and residual controls. *There exist constants $\kappa_0, M, r_0, r_h \ge 0$ such that $\kappa(t) := \max_{i,k\in[n]} |K_{ik}(t)| \le \kappa_0$, $\|W(t)\|_{2\to 2} \le M$, $|R_\varepsilon(t)| \le r_0$, $\max_{i\in[n]} \|r_i(t)\|_2 \le r_h$.*

(B) Hard aligned subset with weak cross-talk. *There exist a subset $S \subset [n]$ with $|S| \ge \rho n$ and constants $k_{\min} > 0$, $\delta \in [0, 1)$, $\alpha_0 > 0$, $m_0 > 0$, $a_0 > 0$ such that for all $i \in S$ and all $t \in [\tau, \tau + H]$, we have $K_{ii}(t) \ge k_{\min}$, $\sum_{k\ne i} |K_{ik}(t)| \le \delta K_{ii}(t)$, and $\alpha_i(t) := 1 - (p_i(t))_{y_i} \ge \alpha_0$. Letting $c_i(t) \in \arg\max_{c\ne y_i} z_{i,c}(t)$ (a highest-scoring wrong class), we have*

$$z_{i,c_i(t)}(t) - \max_{c\ne y_i c\ne c_i(t)} z_{i,c}(t) \ge m_0,$$

and with $v_i(t) := W_{y_i}(t) - W_{c_i(t)}(t)$,

$$\langle \psi_\varepsilon(h_i(t)), v_i(t)\rangle \ge a_0\sqrt{d}.$$

(C) Easy complement + weak coupling to hard gradients. *There exist constants $\bar{\alpha} \in (0, 1)$ and $\bar{\delta} \ge 0$ such that for all $t \in [\tau, \tau + H]$, we have*

$$\alpha_k(t) := 1 - (p_k(t))_{y_k} \le \bar{\alpha}, \quad \forall k \notin S,$$

$$\sum_{k\in S}\sum_{i\notin S}|K_{ik}(t)| \le \bar{\delta}|S|k_{\min}.$$

(D) Feasibility (signal dominates leakage and residual). *Define* $\zeta := (C-2)e^{-m_0}$, $b_0 := (\alpha_0 - \zeta)a_0 - 2\zeta M$, $\mu_D := \rho k_{\min}\big(b_0 - \delta\sqrt{2}M\big)/n\sqrt{d}$, $\gamma_{\text{easy}} := \kappa_0\sqrt{2}M\Big(\sqrt{\bar{\alpha}} + \bar{\delta}\Big)/\sqrt{d}$ *and* $\mu_D^{\text{eff}} := \mu_D - \gamma_{\text{easy}}$. *Assume* $b_0 > \delta\sqrt{2}M$ *and* $\mu_D^{\text{eff}} > r_0$.

(E) Initial margin below the Phase II threshold for the full horizon. *Let* $N_* := (\mu_D^{\text{eff}} - r_0)/\lambda_h$. *Define* $D_{\max} := \kappa_0\sqrt{2}M/\sqrt{d}$, $H_0 := \max_{i\in[n]}\|h_i(\tau)\|_2$ *and*

$$\begin{aligned} H_{\max} &:= e^{\lambda_h H}H_0 + \frac{\kappa_0\sqrt{2}M + r_h}{\lambda_h}\big(e^{\lambda_h H} - 1\big),\\ N_{\max} &:= \frac{H_{\max}}{\sqrt{d}} + \varepsilon,\\ V &:= D_{\max} + \lambda_h N_{\max} + r_0. \end{aligned}$$

Assume there exists $\eta > 0$ *such that*

$$N_\varepsilon(\tau) \le N_* - VH - \eta. \tag{13}$$

Lemma 5 (Top-2 dominance from runner-up separation)**.** *Assume assumption 2(B) holds. Then for any* $i \in S$ *and any* $t \in [\tau, \tau + H]$,

$$\begin{aligned} \sum_{c\neq y_i,\ c\neq c_i(t)} (p_i(t))_c &\le (C-2)e^{-m_0} =: \zeta,\\ (p_i(t))_{c_i(t)} &\ge \alpha_i(t) - \zeta. \end{aligned}$$

Proof. Fix $i \in S$ and $t \in [\tau, \tau + H]$. For any $c \neq y_i, c_i(t)$,

$$\begin{aligned} \frac{(p_i(t))_c}{(p_i(t))_{c_i(t)}} &= \exp\big(z_{i,c}(t) - z_{i,c_i(t)}(t)\big)\\ &\le e^{-m_0}. \end{aligned}$$

Summing over $c \neq y_i, c_i(t)$ yields

$$\begin{aligned} \sum_{c\neq y_i, c\neq c_i(t)} (p_i(t))_c &\le (C-2)e^{-m_0}(p_i(t))_{c_i(t)}\\ &\le (C-2)e^{-m_0}\\ &= \zeta. \end{aligned}$$

Since $\alpha_i(t) = \sum_{c\neq y_i}(p_i(t))_c = (p_i(t))_{c_i(t)} + \sum_{c\neq y_i, c\neq c_i(t)}(p_i(t))_c$, we obtain $(p_i(t))_{c_i(t)} \ge \alpha_i(t) - \zeta$. □

Lemma 6 (Diagonal negativity on S)**.** *Assume assumption 2(A)–(D) holds. Then for all* $i \in S$ *and* $t \in [\tau, \tau + H]$,

$$\big\langle \psi_\varepsilon(h_i(t)), g_i(t)\big\rangle \le -b_0\sqrt{d}.$$

Proof. Fix $i \in S$ and $t \in [\tau, \tau + H]$. Recall

$$\begin{aligned} g_i(t) &= W(t)^\top(p_i(t) - e_{y_i})\\ &= -\sum_{c\neq y_i}(p_i(t))_c(W_{y_i}(t) - W_c(t)). \end{aligned}$$

Hence

$$\begin{aligned} \langle \psi_\varepsilon(h_i), g_i\rangle &= -(p_i)_{c_i}\langle \psi_\varepsilon(h_i), v_i\rangle\\ &\quad - \sum_{c\neq y_i, c\neq c_i}(p_i)_c\langle \psi_\varepsilon(h_i), W_{y_i} - W_c\rangle. \end{aligned}$$

By alignment, $\langle \psi_\varepsilon(h_i), v_i \rangle \geq a_0\sqrt{d}$. Also $\|\psi_\varepsilon(h_i)\|_2 \leq \sqrt{d}$ and $\|W_{y_i} - W_c\|_2 \leq 2\|W\|_{2\to 2} \leq 2M$, so $|\langle \psi_\varepsilon(h_i), W_{y_i} - W_c \rangle| \leq 2M\sqrt{d}$. Using lemma 5 and $\alpha_i(t) \geq \alpha_0$ gives $(p_i)_{c_i} \geq \alpha_0 - \zeta$ and $\sum_{c \neq y_i, c \neq c_i} (p_i)_c \leq \zeta$. Therefore

$$\begin{aligned}
\langle \psi_\varepsilon(h_i), g_i \rangle &\leq -(\alpha_0 - \zeta)a_0\sqrt{d} + (2M\sqrt{d})\zeta \\
&= -\big((\alpha_0 - \zeta)a_0 - 2\zeta M\big)\sqrt{d} \\
&= -b_0\sqrt{d}.
\end{aligned}$$

□

Lemma 7 (Uniform negativity of D_ε on the Phase II window (with easy-complement control))**.** *Assume Assumption 2(B)–(E) hold. Then for all $t \in [\tau, \tau + H]$, we have $D_\varepsilon(t) \leq -\mu_D^{\text{eff}}$.*

Proof. Fix $t \in [\tau, \tau + H]$. Decompose

$$\begin{aligned}
D_\varepsilon(t) &= \frac{1}{n^2 d} \sum_{i \in S} \sum_{k=1}^{n} K_{ik}(t) \langle \psi_\varepsilon(h_i(t)), g_k(t) \rangle \\
&\quad + \frac{1}{n^2 d} \sum_{i \notin S} \sum_{k=1}^{n} K_{ik}(t) \langle \psi_\varepsilon(h_i(t)), g_k(t) \rangle \\
&=: D_S(t) + D_{S^c}(t).
\end{aligned}$$

Step 1: bound $D_S(t)$ by a negative constant. As in the original proof, for $i \in S$ we have (Lemma 6)

$$\langle \psi_\varepsilon(h_i(t)), g_i(t) \rangle \leq -b_0\sqrt{d},$$

and for all i, k,

$$\begin{aligned}
\big|\langle \psi_\varepsilon(h_i(t)), g_k(t) \rangle\big| &\leq \|\psi_\varepsilon(h_i(t))\|_2 \|g_k(t)\|_2 \\
&\leq \sqrt{d} \cdot \sqrt{2}M.
\end{aligned}$$

Combining this bound with the weak cross-talk condition $\sum_{k \neq i} |K_{ik}| \leq \delta K_{ii}$, together with $K_{ii} \geq k_{\min}$ and $|S| \geq \rho n$, we obtain

$$\begin{aligned}
D_S(t) &\leq -\frac{|S|}{n^2 d} k_{\min} \big(b_0 - \delta\sqrt{2}M\big)\sqrt{d} \\
&\leq -\frac{\rho k_{\min}}{n\sqrt{d}} \big(b_0 - \delta\sqrt{2}M\big) \\
&= \frac{\rho k_{\min} \big(\delta\sqrt{2}M - b_0\big)}{n\sqrt{d}} \\
&= -\mu_D.
\end{aligned}$$

Step 2: bound the complement term $D_{S^c}(t)$ using (C). Split $D_{S^c}(t)$ by $k \in S$ and $k \notin S$:

$$\begin{aligned}
D_{S^c}(t) &= \frac{1}{n^2 d} \sum_{i \notin S} \sum_{k \in S} K_{ik} \langle \psi(h_i), g_k \rangle \\
&\quad + \frac{1}{n^2 d} \sum_{i \notin S} \sum_{k \notin S} K_{ik} \langle \psi(h_i), g_k \rangle \\
&=: T_{\text{hard}}(t) + T_{\text{easy}}(t).
\end{aligned}$$

Easy-gradient part. For $k \notin S$, (C) gives $\alpha_k(t) \leq \bar{\alpha}$, hence

$$\begin{aligned}
\|g_k(t)\|_2 &\leq \|W(t)\|_{2\to 2} \|p_k(t) - e_{y_k}\|_2 \\
&\leq M\sqrt{2\alpha_k(t)} \\
&\leq \sqrt{2}M\sqrt{\bar{\alpha}}.
\end{aligned}$$

Therefore $\left|\langle \psi(h_i), g_k\rangle\right| \le \|\psi(h_i)\|_2 \|g_k\|_2 \le \sqrt{d}\cdot\sqrt{2}M\sqrt{\bar{\alpha}}$, so with $|K_{ik}| \le \kappa_0$,

$$\begin{aligned}|T_{\text{easy}}(t)| &\le \frac{1}{n^2 d}\sum_{i\notin S}\sum_{k\notin S}|K_{ik}|\cdot\sqrt{d}(\sqrt{2}M\sqrt{\bar{\alpha}}) \\ &\le \frac{1}{n^2 d}\cdot n^2\cdot\kappa_0\cdot\sqrt{d}(\sqrt{2}M\sqrt{\bar{\alpha}}) \\ &= \frac{\kappa_0\sqrt{2}M}{\sqrt{d}}\sqrt{\bar{\alpha}}.\end{aligned}$$

Hard-gradient coupling from $i \notin S$ to $k \in S$. For $k \in S$, we use the crude bound $\|g_k(t)\|_2 \le \sqrt{2}M$, hence $\left|\langle \psi(h_i), g_k\rangle\right| \le \sqrt{d}\cdot\sqrt{2}M$. Thus

$$\begin{aligned}|T_{\text{hard}}(t)| &\le \frac{1}{n^2 d}\sum_{i\notin S}\sum_{k\in S}|K_{ik}|\cdot\sqrt{d}(\sqrt{2}M) \\ &= \frac{\sqrt{2}M}{n^2\sqrt{d}}\sum_{k\in S}\sum_{i\notin S}|K_{ik}|.\end{aligned}$$

By (C), $\sum_{k\in S}\sum_{i\notin S}|K_{ik}| \le \bar{\delta}|S|k_{\min} \le \bar{\delta}(\rho n)k_{\min}$, so

$$\begin{aligned}|T_{\text{hard}}(t)| &\le \frac{\sqrt{2}M}{n^2\sqrt{d}}\cdot\bar{\delta}(\rho n)k_{\min} \\ &= \frac{\rho k_{\min}}{n\sqrt{d}}\bar{\delta}\sqrt{2}M \\ &\le \frac{\kappa_0\sqrt{2}M}{\sqrt{d}}\bar{\delta},\end{aligned}$$

where in the last step we used $\rho k_{\min}/n \le \kappa_0$ (since $k_{\min} \le \max_{i,k}|K_{ik}| \le \kappa_0$ and $\rho \le 1$). Therefore

$$\begin{aligned}|D_{S^c}(t)| &\le |T_{\text{hard}}(t)| + |T_{\text{easy}}(t)| \\ &\le \frac{\kappa_0\sqrt{2}M}{\sqrt{d}}\left(\bar{\delta} + \sqrt{\bar{\alpha}}\right) \\ &= \gamma_{\text{easy}}.\end{aligned}$$

Step 3: combine. We have $D_\varepsilon(t) = D_S(t) + D_{S^c}(t) \le -\mu_D + |D_{S^c}(t)| \le -\mu_D + \gamma_{\text{easy}} = -\mu_D^{\text{eff}}$. □

Lemma 8 (Finite-velocity bound)**.** *Assume assumption 2(A) holds. With V as in assumption 2(E),*

$$|\dot{N}_\varepsilon(t)| \le V, \quad \forall t \in [\tau, \tau + H].$$

Proof. From lemma 1,

$$|\dot{N}_\varepsilon| \le |D_\varepsilon| + |C_\varepsilon| + |R_\varepsilon|.$$

Under (A), applying lemma 3 with (κ_0, M) gives $|D_\varepsilon(t)| \le D_{\max}$. Also lemma 2 implies $0 \le C_\varepsilon(t) \le \lambda_h N_\varepsilon(t)$. It remains to bound $N_\varepsilon(t)$ on $[\tau, \tau + H]$. From (7), we have

$$\begin{aligned}\|\dot{h}_i(t)\|_2 &\le \frac{1}{n}\sum_k |K_{ik}(t)|\|g_k(t)\|_2 + \lambda_h\|h_i(t)\|_2 + \|r_i(t)\|_2 \\ &\le \kappa_0\sqrt{2}M + \lambda_h\|h_i(t)\|_2 + r_h,\end{aligned}$$

where we used $\frac{1}{n}\sum_k |K_{ik}| \le \max_k |K_{ik}| \le \kappa_0$ and (6). Let $u_i(t) := \|h_i(t)\|_2$. Then

$$u_i'(t) \le \kappa_0\sqrt{2}M + \lambda_h u_i(t) + r_h.$$

So comparison yields $u_i(t) \le H_{\max}$ for all $t \in [\tau, \tau + H]$. Using $\rho_\varepsilon(u) \le |u| + \varepsilon$ and $\|h_i\|_1 \le \sqrt{d}\|h_i\|_2$ gives $N_\varepsilon(t) \le N_{\max}$, hence $|C_\varepsilon(t)| \le \lambda_h N_{\max}$. Finally, (A) gives $|R_\varepsilon(t)| \le r_0$. Combining: $|\dot{N}_\varepsilon(t)| \le D_{\max} + \lambda_h N_{\max} + r_0 = V$. □

Theorem 1 (Phase II densification on a prescribed horizon)**.** *Under assumption 2, $N_\varepsilon(t)$ is strictly increasing on $[\tau, \tau + H]$ and*

$$\dot{N}_\varepsilon(t) \geq \lambda_h \eta, \quad \forall t \in [\tau, \tau + H].$$

Proof. By lemma 7, $D_\varepsilon(t) \leq -\mu_D^{\text{eff}}$ for all $t \in [\tau, \tau + H]$. Also lemma 2 gives $C_\varepsilon(t) \leq \lambda_h N_\varepsilon(t)$, and (B) gives $R_\varepsilon(t) \geq -r_0$. Thus, for all $t \in [\tau, \tau + H]$,

$$\begin{aligned}\dot{N}_\varepsilon(t) &= -D_\varepsilon(t) - C_\varepsilon(t) + R_\varepsilon(t) \\ &\geq \mu_D^{\text{eff}} - \lambda_h N_\varepsilon(t) - r_0 \\ &= \lambda_h \big(N_* - N_\varepsilon(t)\big).\end{aligned}$$

By lemma 8, $|\dot{N}_\varepsilon(t)| \leq V$, hence for any $t \in [\tau, \tau + H]$,

$$\begin{aligned}N_\varepsilon(t) &\leq N_\varepsilon(\tau) + V(t - \tau) \\ &\leq N_\varepsilon(\tau) + VH \\ &\leq N_* - \eta.\end{aligned}$$

using (13). Therefore $N_* - N_\varepsilon(t) \geq \eta$ and so

$$\begin{aligned}N_\varepsilon(t) &\leq N_\varepsilon(\tau) + V(t - \tau) \\ &\leq N_\varepsilon(\tau) + VH \\ &\leq N_* - \eta.\end{aligned}$$

This implies strict increase on the whole window. □

A.2.7 Finite-horizon two-phase trajectory: Phase I decrease trend + bridge to Phase II

Theorem 2 (Finite-horizon sparsify-then-densify trigger)**.** *Assume Phase I boundedness (assumption 1) holds on $[0, T_{\text{I}}]$ and that $\lambda_h > 0$. Let B_0 be as in lemma 4.*

(1) Phase I: early decrease and certified entry into a low-ℓ_1 regime. *For any level $L > \varepsilon + \frac{B_0}{\lambda_h}$, define t_L as in lemma 4. If $t_L \leq T_{\text{I}}$, then:*

- *(i) $N_\varepsilon(t)$ is strictly decreasing whenever $N_\varepsilon(t) > \varepsilon + \frac{B_0}{\lambda_h}$;*
- *(ii) $N_\varepsilon(t_L) \leq L$ (so the trajectory reaches the prescribed low level by time t_L).*

(2) Phase II trigger: densification on any later valid window. *Fix any horizon length $H > 0$. Suppose that at some (a priori unknown) later time $\tau \geq 0$ the Phase II assumptions (assumption 2) hold on the window $[\tau, \tau + H]$. Then on that window, N_ε is strictly increasing with the uniform slope lower bound*

$$\dot{N}_\varepsilon(t) \geq \lambda_h \eta, \quad \forall t \in [\tau, \tau + H],$$

where η is the margin parameter appearing in assumption 2.

Proof. Part (1) is exactly lemma 4. Part (2) is exactly theorem 1 applied to the (hypothetical) window $[\tau, \tau + H]$. □

Remark 3 (How to read the certificate)**.** *This theorem does* not *claim that a single continuous interval exhibits a full U-shape without further information. Instead, it provides two independent, checkable guarantees along a training trajectory: (i) an* early-time sparsification certificate *that forces N_ε to decrease and reach any prescribed reasonable level $L > \varepsilon + \frac{B_0}{\lambda_h}$ within the Phase I validity horizon, and (ii) a* densification trigger*: whenever (at any later time) the trajectory enters a finite window on which the persistent-hardness and alignment conditions of assumption 2 hold, N_ε must increase throughout that window at rate at least $\lambda_h \eta$. In this sense, a U-shape pattern arises whenever the dynamics first satisfies the Phase I conditions and later* activates *a Phase II window.*

A.3 Extended Related Work

A.3.1 Sparsity in Deep Neural Network

Sparsity in deep learning is traditionally studied through two distinct lenses: computational efficiency and representational disentanglement (Hoefler et al., 2021; Cheng et al., 2024).

From an efficiency perspective, weight sparsity aims to reduce model size and inference cost. Pioneering work in network pruning demonstrated that significant portions of parameters can be removed without performance degradation (Han et al., 2015). This line of inquiry culminated in the *Lottery Ticket Hypothesis*, which posits that dense networks contain sparse, trainable subnetworks capable of matching the original model's accuracy (Frankle and Carbin, 2019). This hypothesis was subsequently extended to Transformer architectures, particularly BERT. Research indicates that pre-trained BERT models contain matching subnetworks at extreme sparsity levels (e.g., 40%–90%) that perform comparably on downstream tasks (Chen et al., 2020; Prasanna et al., 2020; Gordon et al., 2020). Beyond imposed sparsity, recent empirical studies reveal that LLMs exhibit a high degree of *intrinsic activation sparsity*. Despite being trained as dense networks, the activations within Transformer Feed-Forward Networks (FFNs) are highly sparse, with only a small fraction of neurons firing for any given input token. This behavior, often termed the "Lazy Neuron" phenomenon, becomes more pronounced as model scale increases (Li et al., 2023).

Further analysis characterizes this as *contextual sparsity*: specific inputs activate predictable, sparse sub-graphs of the network, suggesting that LLMs implicitly learn modular structures without explicit architectural constraints (Liu et al., 2023). Zhao and Thrampoulidis (2025b,a); Zhao et al. (2024c) view language modeling as a classification problem of "predicting the distribution of the next word for each context," the sparsity pattern of the language data strongly determines the geometry of the trained representation. They found that Next Token Prediction (NTP) training implicitly favors a "sparse + low-rank" structure in the logit space.

A.3.2 LLM Curriculum Reasoning

Curriculum Learning (CL), introduced by Bengio et al. (2009), is a training strategy that exposes models to examples in an easy-to-hard progression, rather than in random order (Wang et al., 2021). Early NLP applications likewise leveraged CL principles. In unsupervised grammar induction, a dependency parser was trained on short, simple sentences first and incrementally included longer, more complex sentences, yielding improved parsing accuracy (Spitkovsky et al., 2010). With the advent of deep learning, Curriculum Learning (CL) was widely adopted in Neural Machine Translation (NMT), where examples were typically sorted by sentence length or word rarity to accelerate convergence (Platanios et al., 2019; Xu et al., 2020). In the era of Pre-trained Language Models (PLMs) such as BERT, CL strategies shifted towards optimizing data scheduling to improve sample efficiency during the massive pre-training phase (Nagatsuka et al., 2021; Lee et al., 2022).

Most recently, the rise of LLMs has reinvigorated CL, particularly in enhancing complex reasoning capabilities and instruction following Wang et al. (2025b). Unlike traditional methods relying on superficial metrics (e.g., length), modern CL for LLMs focuses on *semantic complexity* and *reasoning depth*. For instance, Xu et al. (2024) proposed *Evol-Instruct*, a method that incrementally rewrites instructions to increase difficulty, effectively creating a curriculum for instruction tuning. Similarly, Mukherjee et al. (2023) demonstrated that learning from "explanation traces" in a progressive manner allows smaller models to imitate the reasoning processes of larger foundation models. Furthermore, in mathematical and logical reasoning tasks, CL has been utilized to transition models from simple single-step problems to multi-step reasoning chains, significantly mitigating the difficulty of solving complex problems directly (Luo et al., 2025).

A.3.3 Interpretability in LLM Reasoning

The interpretation of reasoning in Large Language Models (LLMs) has evolved from analyzing surface-level generations to probing the internal causal mechanisms that drive them (Zhao et al., 2024a,b; Wang et al., 2024b). A natural starting point for interpretability is to treat generated explanations, most prominently CoT, as a proxy for the model's decision-making process (Wei et al., 2022). The structure of these demonstrations also matters: changing reasoning-step length can affect accuracy even when their information content is preserved (Jin et al., 2024). However, recent

scholarship challenges this assumption, characterizing CoT as often unfaithful. Evidence suggests that intermediate reasoning steps can be post-hoc rationalizations that do not causally determine the final prediction (Turpin et al., 2023), and increasing reasoning burden does not necessarily equate to computational transparency (Lanham et al., 2023; Arcuschin et al., 2025). Output confidence is also an imperfect proxy: Zeng et al. (2025) show that adjusting first-token probabilities can distort uncertainty estimates for multiple-choice questions while leaving answer accuracy unchanged. Consequently, the field has shifted focus from verifying *plausible explanations* to establishing *causal accounts* of model behavior. Jin et al. (2025a) explicitly separate memory recall from reasoning during inference, helping identify the sources of model errors.

To bridge the gap between model inputs and outputs, mechanistic interpretability develops tools that aim to isolate components that are necessary and sufficient for particular behaviors, and to characterize how information is represented and transformed across the network (Wang et al., 2023; Gantla, 2025). A prominent line of work uses causal interventions: most notably, causal tracing and activation patching to perform controlled swaps between activations induced by "clean" versus "corrupted" inputs, thereby localizing where task-relevant signals reside and empirically testing how those signals propagate through layers and attention pathways (Wang et al., 2023; Zhao et al., 2025; Meng et al., 2022). Complementary approaches focus less on direct causal disruption and more on interpretability via readouts: representation-decoding frameworks such as Patchscopes map hidden states into structured textual probes, offering a unified interface for inspecting intermediate computations under a variety of intervention patterns (Ghandeharioun et al., 2024). Related "lens" methods, including the Logit Lens, project intermediate activations into the vocabulary (logit) space to track how candidate outputs become linearly recoverable across depth; while lightweight and often used diagnostically, these readouts are frequently most informative when paired with causal tests that distinguish genuine computation from superficial decodability (nostalgebraist, 2020; Wendler et al., 2024; Geva et al., 2021; Belrose et al., 2023). Layer-wise probing further suggests that more complex concepts become accessible at deeper layers (Jin et al., 2025b).

At a finer level of analysis, mechanistic interpretability work tries to break down a model's behavior into concrete computational pathways ("circuits") and smaller features that can be followed as they appear and change across different inputs (Elhage et al., 2021, 2022). A central thread in this direction explains in-context learning through specific attention-head motifs: most notably induction heads that appear to implement a concrete pattern-copying mechanism (Olsson et al., 2022). To move beyond correlational stories, subsequent work has introduced more stringent validation procedures, such as causal scrubbing and automated circuit discovery (ACDC), which use behavior-preserving resampling and controlled interventions to test whether a proposed circuit is genuinely explanatory rather than an artifact of spurious co-activation (Conmy et al., 2023).

In parallel, feature-centric approaches address the pervasive polysemanticity of dense activations by learning sparse decompositions: Sparse Autoencoders (SAEs) attempt to "unsuperpose" representations into more interpretable directions that are often closer to monosemantic features, enabling a complementary level of analysis that is not limited to predefined motifs like individual heads (Huben et al., 2024; Bricken et al., 2023). Building on SAE features, SAGE iteratively tests and refines feature explanations using activation feedback (Han et al., 2026), while SAE-SSV learns supervised steering vectors within task-relevant sparse latent subspaces (He et al., 2025). Overall, the literature increasingly reflects a convergence in methodology, pairing explanatory narratives with rigorous causal tests and feature-level decomposition in order to debug and interpret reasoning processes in modern LLMs.

A.3.4 Learning Dynamics

A complementary line of work studies how optimization shapes representations over the course of training. Classical results on the *implicit bias* of gradient-based optimization show that, even without explicit regularization, gradient descent on separable classification problems converges in direction to max-margin solutions, linking training dynamics to margin growth and norm evolution (Soudry et al., 2018; Ji and Telgarsky, 2019); related analyses extend these behaviors to stochastic gradient descent (Nacson et al., 2019). More recently, phase-transition-like learning phenomena such as *grokking* highlight that models can move from memorization to generalization late in training, motivating mechanistic accounts in terms of regime changes in training dynamics and feature learning (Kumar et al., 2024; DeMoss et al., 2025).

Recent work has also begun to analyze learning dynamics in the context of LLM adaptation. Ren and Sutherland (2025) propose a learning-dynamics framework for *LLM fine-tuning*, including supervised fine-tuning and preference optimization, characterizing how learning on specific examples influences predictions on others and using this lens to explain several counter-intuitive fine-tuning behaviors. These perspectives suggest that simple representation-level statistics, such as norms, energy concentration, and effective dimensionality, can serve as useful probes of how features are selected, amplified, or suppressed during learning.

Our work is complementary to this literature. Rather than analyzing the parameter trajectory induced by optimization or fine-tuning, we study an inference-time representational signature associated with increasing OOD difficulty. In particular, we show that harder inputs consistently induce more concentrated last-layer representations, and that this "harder-is-sparser" pattern emerges only under sufficiently capable models or training regimes in our controlled setting. Thus, our results connect learning dynamics to OOD generalization from a representation-level perspective: as models acquire more structured reasoning behavior, difficulty is reflected not only in output accuracy but also in the geometry and sparsity of their hidden states.

A.4 All Metrics

A.4.1 Sparsity Metrics

We analyze the sparsity of the activation vector $\mathbf{h} \in \mathbb{R}^d$ extracted from the last hidden state of the model. Since raw activations are rarely absolute zeros in floating-point representations, we utilize these metrics that capture the *effective sparsity* (i.e., the peakedness of the distribution).

Hoyer Sparsity. Derived from the relationship between the ℓ_1 and ℓ_2 Norms, Hoyer sparsity measures how close a vector is to being sparse (containing mostly zeros). For an activation vector $\mathbf{h}$ of dimension d:

$$\text{Hoyer}(\mathbf{h}) = \frac{\sqrt{d} - \|\mathbf{h}\|_1/\|\mathbf{h}\|_2}{\sqrt{d} - 1}, \tag{14}$$

where $\|\cdot\|_1$ is the ℓ_1 norm and $\|\cdot\|_2$ is the Euclidean norm. A value of $\mathbf{0}$ indicates a dense, uniform vector ($h_i = c, \forall i$) and a value of $\mathbf{1}$ indicates maximum sparsity (only one component is non-zero).

Gini Index. Originally used in economics to measure income inequality, the Gini index here quantifies the inequality of neural activation strengths. High inequality implies that a few neurons dominate the representation while the majority remain suppressed (sparse).

$$\text{Gini}(\mathbf{h}) = \frac{\sum_{i=1}^{d}(2i - d - 1)|h|_{(i)}}{d\sum_{i=1}^{d}|h|_{(i)}}, \quad |h|_{(1)} \leq \cdots \leq |h|_{(d)}, \tag{15}$$

where $|h_{(i)}|$ are the sorted absolute values of the activations. A higher Gini index correlates with a sparser representation.

Effective Rank. Effective rank quantifies how concentrated the activation energy is across dimensions. Given $\mathbf{h} \in \mathbb{R}^d$, define a distribution over dimensions by $p_i = \frac{h_i^2}{\sum_{j=1}^{d} h_j^2}$, and entropy $H(\mathbf{p}) = -\sum_{i=1}^{d} p_i \log(p_i + \epsilon)$. We then compute

$$\text{EffRank}(\mathbf{h}) = \frac{\exp(H(\mathbf{p}))}{d} \in (0, 1]. \tag{16}$$

Larger values indicate more evenly spread energy (denser representations), while smaller values indicate stronger concentration on a few dimensions (sparser representations).

As a complementary metric, we also compute the normalized ℓ_1 magnitude:

$$S_{L_1}(h) = \tfrac{1}{d}\sum_{i=1}^{d}|h_i|. \tag{17}$$

Unlike Top-k Energy, S_{L_1} depends on the overall scale of the hidden state. We therefore use it as a complementary metric, comparing representations within the same model, layer, token position, and evaluation protocol, where hidden-state norms are broadly comparable. Under this controlled setting, a lower S_{L_1} indicates that activation mass is concentrated on fewer coordinates, providing additional evidence of increased sparsity.

A.4.2 Model Parameter Metrics

We validate our findings across a range of transformer model sizes, ranging from 0.3M to 1.3B parameters, using the Llama architecture, we need to know how to compute the model Parameter size N_{params}. As detailed in Table 1, the model complexity is controlled by the hidden size (d_{model}), the intermediate MLP size (d_{mlp}), the number of attention heads (H), and the network depth (L). Assuming a standard Llama-based architecture (utilizing SwiGLU activation and Rotary Embeddings), the approximate parameter count N_{params} for the non-embedding layers is calculated as:

$$N_{params} \approx L \cdot \left(\underbrace{4d_{model}^2}_{\text{Attention}} + \underbrace{3d_{model} \cdot d_{mlp}}_{\text{FeedForward (SwiGLU)}} \right) \tag{18}$$

where the attention block consists of W_Q, W_K, W_V, W_O, contributing $4d_{model}^2$. The FeedForward block (SwiGLU) involves three projections (gate, up, down), contributing $3d_{model} \cdot d_{mlp}$.

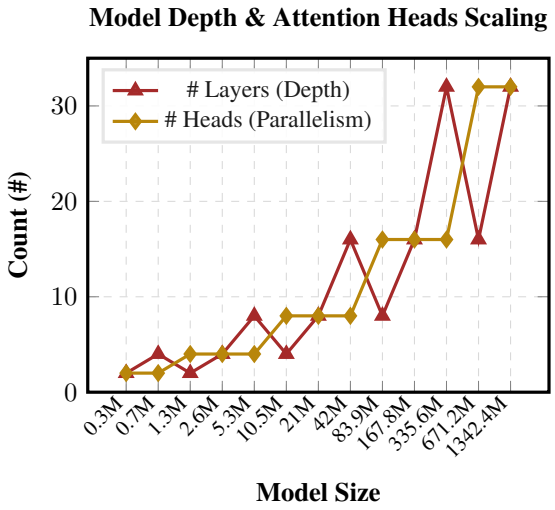


Figure 8: Scaling of computational depth (Layers) and parallel attention capacities (Heads) across Llama models. Both metrics show a step-wise increase consistent with the model's total parameter growth.

Figure 8 summarizes how L (layers) and H (attention heads) scale with model size in our Llama sweep. Both increase in discrete steps as the parameter count grows.

A.5 Reasoning Prompt

The following demonstrates the implicit reasoning prompts designed to address knowledge conflicts, mathematical reasoning, and multiple-choice questions, and related tasks:

Example Reasoning Prompt 1

System: You are a helpful assistant. For each multiple choice question, you must answer with ONLY a single letter: A, B, C, D... Do not explain or add any other text.

User: {prompt_text}. Answer the question with only the letter:

Knowledge Conflict Dataset Entry 1

Main Entity: Variable
Conflict Method: Shuffling (Variable → Random variable)

Parametric Knowledge (Ground Truth):
In programming, a variable is a named container that stores a value. It can hold different data types, such as numbers or strings, and its value can be changed during the execution of a program.

Conflicting Knowledge (Injected Noise):
In programming, a **Random variable** is a named container that stores a value. It can hold different data types, such as numbers or strings, and its value can be changed during the execution of a program.

Generated Questions: What is a named container that stores a value and can be changed during the execution of a program?

Example Reasoning Prompt 2

System: You are a helpful Knowledge Conflict assistant. You should judge the knowledge in the sentence is correct or not. You should answer with ONLY a single letter: True or False. Do not explain or add any other text.

User: {prompt_text}. Answer the question with only the letter:

Example Reasoning Prompt 3

System: You are a helpful math assistant. Provide the final answer in the end.

User: {prompt_text}. Please provide the final answer in the end.

A.6 More Details about Knowledge Conflict and different types of In-context Learning

A.6.1 Details about Knowledge Conflict Dataset

This dataset (Wang et al., 2024c) operates on a *Knowledge Injection and Interference* logic through a two step. First, it establishes Parametric Knowledge by defining ground-truth facts the model acquired during pre-training, such as the standard definition of a programming variable. Second, it performs Conflict Generation using a "shuffling" method, where the core entity is replaced with a phonetically or conceptually similar but semantically distinct term (e.g., substituting *variable* with *random variable*). This creates a strategic contradiction between the provided context and the model's internal common sense.

This dataset is particularly suitable for our research because it allows for the precise control of Out-of-Distribution (OOD) intensity by manipulating the degree of external knowledge conflict. As we posit that adversarial distractors can transform an original "easy" input into a "conflict" perturbed variant, this framework provides a quantifiable way to increase task difficulty. By introducing these specific knowledge conflicts as distractor items, we can systematically simulate OOD scenarios where the model's internal parametric memory directly clashes with contradictory external evidence.

A.6.2 More Results on Larger Model

To verify the universality of our hypothesis that OOD samples trigger higher activation sparsity in the last hidden state, we extended our evaluation to a diverse set of large-scale foundation models and used the same knowledge conflict dataset as (Wang et al., 2024c). Specifically, we selected models varying in both architecture and parameter size, including `Qwen2.5-32B`, `Qwen2.5-70B`, and `Llama-3.1-70B`.

As illustrated in Figure 9, the experimental results are highly consistent across all tested models. We observe that samples involving knowledge conflicts (a representative form of OOD data) systemati-

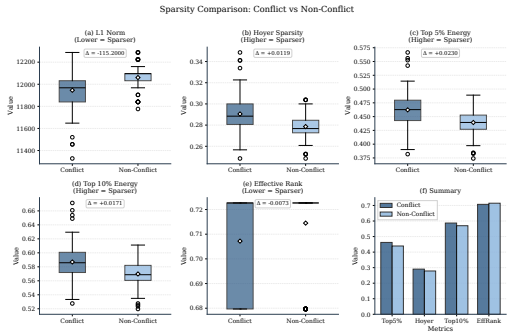
Knowledge Conflict Dataset Entry 2

Main Entity: Equation
Conflict Method: Substitution (two → 9)

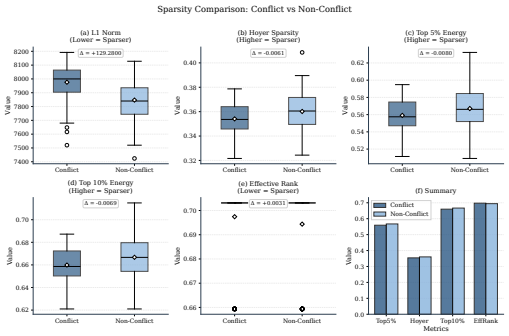
Parametric Knowledge (Ground Truth):
An equation is a mathematical statement that shows the equality of **two** expressions. It typically consists of variables, constants, and mathematical operations.

Conflicting Knowledge (Injected Noise):
An equation is a mathematical statement that shows the equality of **9** expressions. It typically consists of variables, constants, and mathematical operations.

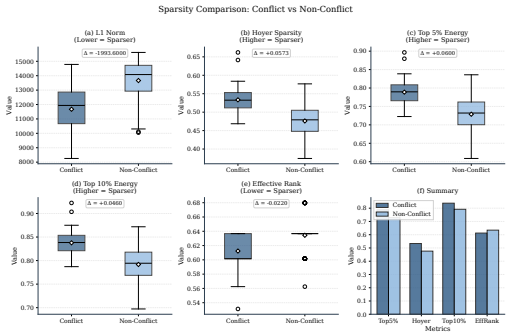
Generated Questions: How many expressions does an equation typically show the equality of?



(a) Llama3.1-70B



(b) Qwen2.5-32B



(c) Qwen2.5-70B

Figure 9: **Sparsity analysis across different models.** (a) Visualizes the sparsity pattern of Llama3.1-70B. (b) and (c) compare the Qwen2.5 series models (32B and 70B).

cally induce significantly sparser activation patterns compared to consistent (in-distribution) samples. This phenomenon remains robust regardless of the model's scaling or architectural design. For instance, even in the 70B parameter language models, the activation density drops sharply when the model encounters conflicting information. These findings strongly corroborate our core conclusion: activation sparsity serves as an architecture-agnostic proxy for sample difficulty and OOD status, validating the effectiveness of using sparsity for data ranking.

A.6.3 Different Kinds of In-Context Learning

Following the experimental framework established by (Niu et al., 2025), we devised multiple In-Context Learning (ICL) strategies to comprehensively evaluate the model's reasoning capabilities under different prompting conditions. Specifically, we implemented four distinct prompting variants, taken from (Niu et al., 2025), as shown in Table 2.

Table 2: Prompt Setup (Niu et al., 2025). We follow the exact same prompt setup as Niu et al. (2025). The emojis represent the target tokens: 😈: counterfactual; 😳: distracting; 🙂: correct.

Context Setting	Context Prompt	Query Prompt	😳	🙂
Related	`On the inside, bananas are white.`	`What color are mangoes on the inside? They are`	`white`	`orange`
Irrelevant	`The capital of Canada is Ottawa.`	`What color are mangoes on the inside? They are`	`Ottawa`	`orange`
Random	`Promotion`	`What color are mangoes on the inside? They are`	`Promotion`	`orange`
Distraction over Counterfactual Context			😈	😳	🙂
Counterfactual	`On the inside, bananas are green.`	`What color are mangoes on the inside? They are`	`green`	`white`	`orange`

Table 3: Sparsity metrics averaged over samples across different context settings in Qwen2.5-1.5B.

Context Setting	L_1 **Norm**	**Top-5% Energy**	**Top-10% Energy**	**Eff. Rank**
Related	5066.96	0.5723	0.6715	0.6715
Irrelevant	5072.15	0.5816	0.6772	0.6764
Random	3689.13	0.7600	0.8203	0.5987
Counterfactual	4938.74	0.5978	0.6937	0.6680

We categorize these prompting strategies into two primary dimensions: *Contextual Distraction* and *Counterfactual Conflict*. The Contextual Distraction settings comprise Related, Irrelevant, and Random contexts, where the input provides truthful but potentially noisy information, aiming to measure the model's susceptibility to blindly copying context tokens versus accessing its internal parametric knowledge. Specifically, the 'Related' context tests the capability to distinguish between semantically similar attributes within a domain, whereas 'Irrelevant' and 'Random' contexts serve as baselines to assess the model's resilience against pure recency bias and token repetition. Conversely, the Counterfactual Conflict setting introduces a direct contradiction between the context and ground truth, creating a tripartite competition in the probability space. Following this categorization, we further investigated the sparsity patterns of the model's last hidden state representations under these distinct settings, as shown in Table 3.

Our results provide empirical support for the hypothesis that inputs deviating further from the pre-training distribution (OOD) induce higher activation sparsity. We observe a clear gradient of sparsity corresponding to the degree of distribution shift. These metrics collectively validate that sparsity is positively correlated with the OOD nature of the input. As the context shifts from natural language (Related) to semantic conflict (Counterfactual) and finally to complete noise (Random), the model's last hidden state representations become increasingly sparse and low-rank.

As shown in Table 3, our results provide empirical support for the hypothesis that inputs deviating further from the pre-training distribution (OOD) induce higher activation sparsity. The `Random` setting, representing the most severe OOD scenario characterized by semantic and structural noise, exhibits distinct hyper-sparsity characteristics. It records the lowest activation magnitude ($L_1 \approx 3689.13$) while simultaneously showing the highest energy concentration (Top-5% Energy ≈ 0.7600), indicating that nonsensical inputs fail to activate broad semantic circuits, thereby concentrating signal flow into a narrow subset of neurons and collapsing the representation into a lower-dimensional subspace as evidenced by the significantly lower Effective Rank (0.5987). A more subtle trend is observed when comparing `Counterfactual` to `Related` contexts; although both are grammatically well-formed, the semantic conflict in the `Counterfactual` setting induces slightly higher sparsity than `Related` inputs (L_1: $4938.74 < 5066.96$; Top-5% Energy: $0.5978 > 0.5723$). This suggests that processing "lies" or counter-knowledge conflicts engages fewer knowledge retrieval circuits compared to processing truthful, coherent information, effectively "pruning" the activation of conflicting parametric knowledge. Conversely, the `Related` and `Irrelevant` settings serve as in-distribution baselines, showing the highest L_1 norms and lowest energy concentrations, which reflect a dense, broad activation of the model's semantic networks in response to natural linguistic structures.

A.7 More Details about Reasoning Complexity

To further understand the relationship between sparsity and model performance, we examine how prediction accuracy varies with sparsity across different difficulty levels. As shown in Figure 10, we observe a strong statistical correlation between accuracy and the sparsity of the last hidden states. Specifically, on the MATH-500 (Lightman et al., 2024) dataset with Qwen2.5-7B, accuracy is highly correlated with both ℓ_1 norm sparsity ($r = 0.897$) and Top-10% Energy ($r = -0.915$).

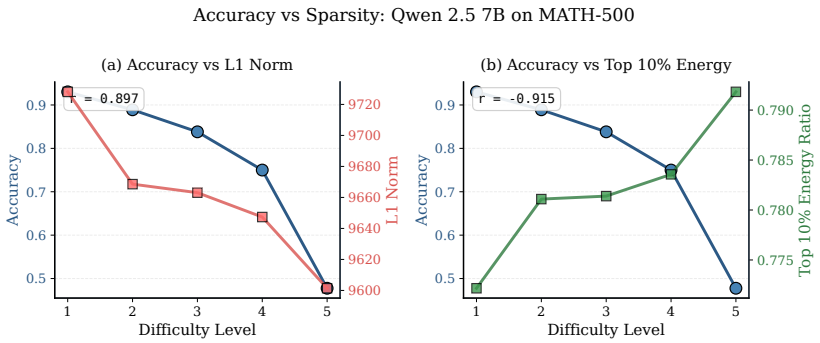


Figure 10: **Accuracy–Sparsity Correlation.** Accuracy strongly correlates with Last Hidden State sparsity on MATH-500 for Qwen2.5-7B (L_1: r=0.897; Top-10% Energy: r=-0.915).

Together, Figure 2a and Figure 10 present a consistent statistical pattern. Figure 2a shows that higher reasoning difficulty systematically induces stronger sparsity in the last hidden states, while Figure 10 demonstrates that higher sparsity is tightly associated with lower model accuracy. In other words, tasks that induce stronger representational sparsity tend to correspond to lower prediction accuracy. These results suggest a coherent relationship between reasoning difficulty, representational sparsity, and model performance: as tasks become harder, the model's last hidden states become progressively sparser, and this sparsity is accompanied by a corresponding decline in accuracy.

We use DeepMath-103K (He et al., 2026), a large-scale math reasoning dataset with about 103K examples. Each sample contains a question, final answer, topic annotation, multiple reference solutions, and an explicit difficulty score, which allows us to partition the data into different difficulty levels (from 1-10) for controlled analysis.

Example from DeepMath-103K

Topic: Mathematics → Precalculus → Limits
Difficulty: 4.5
Question. Evaluate the limit

$$\lim_{x\to\infty} \sqrt{x}\left(\sqrt[3]{x+1} - \sqrt[3]{x-1}\right).$$

Solution. Let

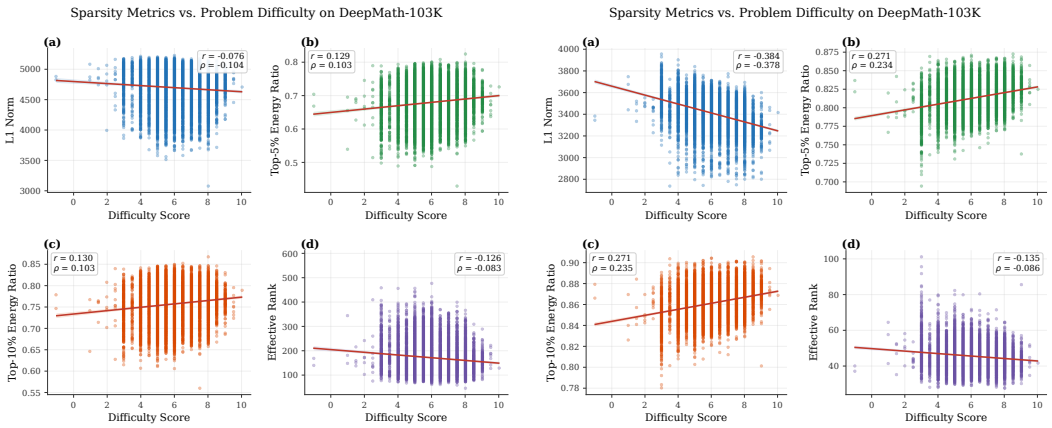
$$a = (x+1)^{1/3}, \qquad b = (x-1)^{1/3}.$$

Then

$$\begin{aligned} a - b &= \frac{a^3 - b^3}{a^2 + ab + b^2} \\ &= \frac{(x+1) - (x-1)}{a^2 + ab + b^2} \\ &= \frac{2}{a^2 + ab + b^2}. \end{aligned}$$

Hence

$$\sqrt{x}(a-b) = \frac{2\sqrt{x}}{a^2 + ab + b^2}.$$



(a) Qwen2.5-3B-Instruct (b) Qwen2.5-1.5B-Instruct

Figure 11: **Sparsity Metrics vs. Problem Difficulty on DeepMath-103K.** We plot several last-layer sparsity metrics against problem difficulty levels for Qwen2.5-3B and Qwen2.5-1.5B. Across both models, harder problems tend to induce progressively sparser final-layer representations, showing a consistent dataset-level relationship between reasoning difficulty and representational sparsity.

As $x \to \infty$, we have

$$a^2 \sim x^{2/3}, \qquad ab \sim x^{2/3}, \qquad b^2 \sim x^{2/3},$$

so

$$a^2 + ab + b^2 \sim 3x^{2/3}.$$

Therefore,

$$\sqrt{x}(a-b) \sim \frac{2x^{1/2}}{3x^{2/3}} = \frac{2}{3}x^{-1/6} \to 0.$$

Final Answer. 0

DeepMath-103K is particularly suitable for our study because its explicit difficulty stratification enables controlled, dataset-level analysis of whether increasing reasoning difficulty systematically leads to sparser last-layer representations. This pattern is further confirmed by our experiments on Qwen2.5-3B and Qwen2.5-1.5B, where we consistently observe that increasing difficulty is accompanied by progressively sparser last-layer representations across difficulty strata.

Across metrics, harder problems tend to induce progressively sparser final-layer representations, showing a consistent dataset-level relationship between reasoning difficulty and representational sparsity (as Figure 11 shows).

A.8 Layer Selection for Sparsity Measurement

An important design choice is *which layer* should be used to measure sparsity. Our main analysis focuses on the **last hidden state**, and this choice is empirically motivated rather than arbitrary. Intuitively, if sparsity is tied to the model's final adaptation to problem difficulty, the effect should become most visible near the output of the network, where reasoning-relevant information is most consolidated. To verify this, we perform an additional layer-wise comparison on **DeepMath-103K**. We find that the sparsity–difficulty relationship is much clearer at the **final layer**, whereas earlier or intermediate layers often show only weak trends or even *inconsistent directions*. For example, at the final layer, increasing difficulty tends to decrease ℓ_1 norm and effective rank while increasing Top-k energy, which is consistent with our sparsification hypothesis. In contrast, at **Layer 8**, these trends become much weaker, and several metrics even reverse direction. This suggests that intermediate-layer sparsity is not a reliable proxy for difficulty, and that the decisive sparsification effect only emerges near the end of the network. The result is also shown in Table 4.

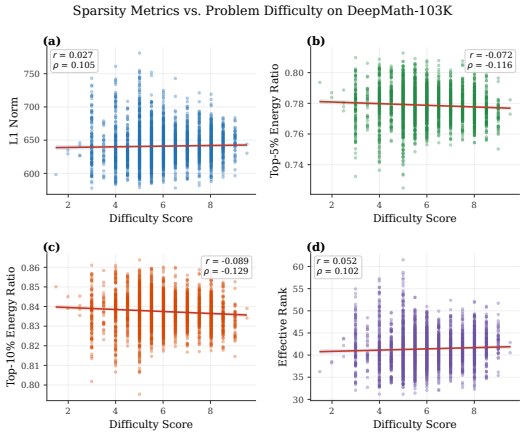


Figure 12: **Sparsity Metrics vs. Problem Difficulty on DeepMath-103K for Qwen2.5-1.5B in Layer8.** We plot several 8-layer sparsity metrics against problem difficulty levels on DeepMath-103K.

Therefore, throughout the main paper, we use the **last hidden state** as the primary object of analysis. Earlier layers may still encode useful information, but they do not provide a stable or interpretable sparsity signal for difficulty tracking.

Table 4: DeepMath-103K correlations between problem difficulty and sparsity metrics. Each entry reports Pearson r / Spearman ρ. The expected sparsification pattern is lower effective rank and higher Top-k energy as difficulty increases.

Setting	ℓ_1	**Top-5% ↑**	**Top-10% ↑**	**Eff. Rank ↓**
Last layer, 3B	$-0.076/-0.104$	$0.129/0.103$	$0.130/0.103$	$-0.126/-0.083$
Last layer, 1.5B	$-0.384/-0.378$	$0.271/0.234$	$0.271/0.235$	$-0.135/-0.086$
Layer 8, 1.5B	$0.027/0.105$	$-0.072/-0.116$	$-0.089/-0.129$	$0.052/0.102$

A.9 More Details about MMLU-Robust

A.9.1 Example in MMLU-PRO and MMLU-Robust

MMLU-Pro (Wang et al., 2024d) is an enhanced version of the original Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2021), developed to address shortcomings that have emerged as modern LLMs have improved. In particular, the original MMLU has become increasingly *too easy* for state-of-the-art models and is affected by data noise in certain subsets. MMLU-Pro introduces several key improvements over its predecessor. ◆ **Increased difficulty.** Overly simple factual questions are filtered out and replaced with more challenging problems that require multi-step and rigorous reasoning. ◆ **Expanded answer space.** The number of multiple-choice options is increased from four to ten, substantially reducing the likelihood of correct answers due to random guessing and better reflecting genuine reasoning ability. ◆ **Broader coverage.** The benchmark comprises over 12,000 questions across 14 core academic domains, spanning STEM, humanities, and social sciences.

To systematically evaluate the last hidden state under progressively intensifying OOD conditions, we construct a variant dataset, **MMLU-Robust**. Building on the already challenging MMLU-Pro benchmark, which features ten multiple-choice options, we introduce an iterative distractor augmentation mechanism. The idea is simple: adding more plausible options makes the task harder. This design draws on prior work that probes LLM robustness through adversarial input perturbations, including MathAttack and prompt-injection attacks (Zhou et al., 2024; Zhang et al., 2024b,a).

In particular, we employ a perturbation function $\mathcal{P}(\cdot)$ to generate plausible but incorrect distractors derived from the existing option set. For a given question with an initial option set $\mathcal{O}_{10}$ ($|\mathcal{O}_{10}| = 10$), we select a subset of non-ground-truth options and apply semantic or numerical perturbations (e.g., sign flipping, unit alteration, or logical inversion) to create new adversarial options. These perturbed variants are injected back into the pool to form expanded sets $\mathcal{O}_{15}$ and $\mathcal{O}_{20}$.

MMLU-Pro Dataset Samples

Sample 1: Number Theory
Question: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$?

Options:

(A) -5 (B) 0 (C) -3
(D) -7
(E) **-4** *(Correct)* (F) -6
(G) -1 (H) -2 (I) -9 (J) N/A

[CoT Reasoning] We have $12n = 22 - 7m$. A particular solution is $m = -2, n = 3$, so $m + n = 1$. The general solution is $m = -2 + 12k, n = 3 - 7k$. Then $m + n = 1 + 5k$. For $m + n < 0$, we need $5k < -1$, so $k \leq -1$. Max value is at $k = -1$: $1 + 5(-1) = -4$.

MMLU-Robust Dataset Samples

Sample 1: Number Theory
Question: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$?

Options:

(A) -5 (B) 0 (C) -3
(D) -7
(E) **-4** *(Correct)* (F) -6
(G) -1 (H) -2 (I) -9 (J) N/A
(H) **1** *[Valid member, but positive]*
(I) **-14** *[Valid member, but < -4]*
(J) **4** *[Sign flip]*
(K) **-8** *[Dense noise]*
(M) **125** *[Dense noise]*

This process creates a controlled "interference gradient." By increasing the option count from 10 to 15 and finally to 20, we densify the solution space with highly correlated noise, forcing the model to discern increasingly subtle differences. We hypothesize that this artificially induced complexity acts as a proxy for severe OOD shifts, triggering the sparse representation mechanism described in our main findings.

A.9.2 More Results in MMLU-Robust

Figure 13 reports a fine-grained breakdown of sparsity-related results on MMLU-Robust for `Qwen2.5-3B-Instruct`, grouped by academic area and evaluated under three adversarial noise levels (Normal (+0) / Moderate (+5) / Hard (+10)). For each area, we aggregate the last hidden state statistics across questions and visualize five metrics: normalized ℓ_1 magnitude, Top-5% Energy, Top-10% Energy, Effective Rank, and Hoyer Sparsity.

Figure 14 reports averaged results for four additional LMs. Across these models, we observe the same qualitative pattern as in the main results: increasing OOD difficulty consistently leads to sharper sparsity in the last hidden state across domains and metrics. This consistency suggests that the observed behavior is not specific to a particular model size or architecture. We encourage readers to reproduce and extend our analysis on additional models, datasets, and evaluation settings to further test the generality of this phenomenon.

To address the space limitation in the main text, we further report a per-area analysis on MMLU(-Robust/MMLU-Pro), where we evaluate each academic field separately to verify whether the sparsity–difficulty relationship holds beyond overall averages. As shown in Figure 13, the trend remains robust across domains: as adversarial noise increases (Normal $\rightarrow$ Moderate $\rightarrow$ Hard), the representation becomes consistently *more sparse* in nearly every area. Concretely, harder (more perturbed) inputs exhibit lower normalized ℓ_1 magnitude and effective rank, together with higher Top-k energy concen-

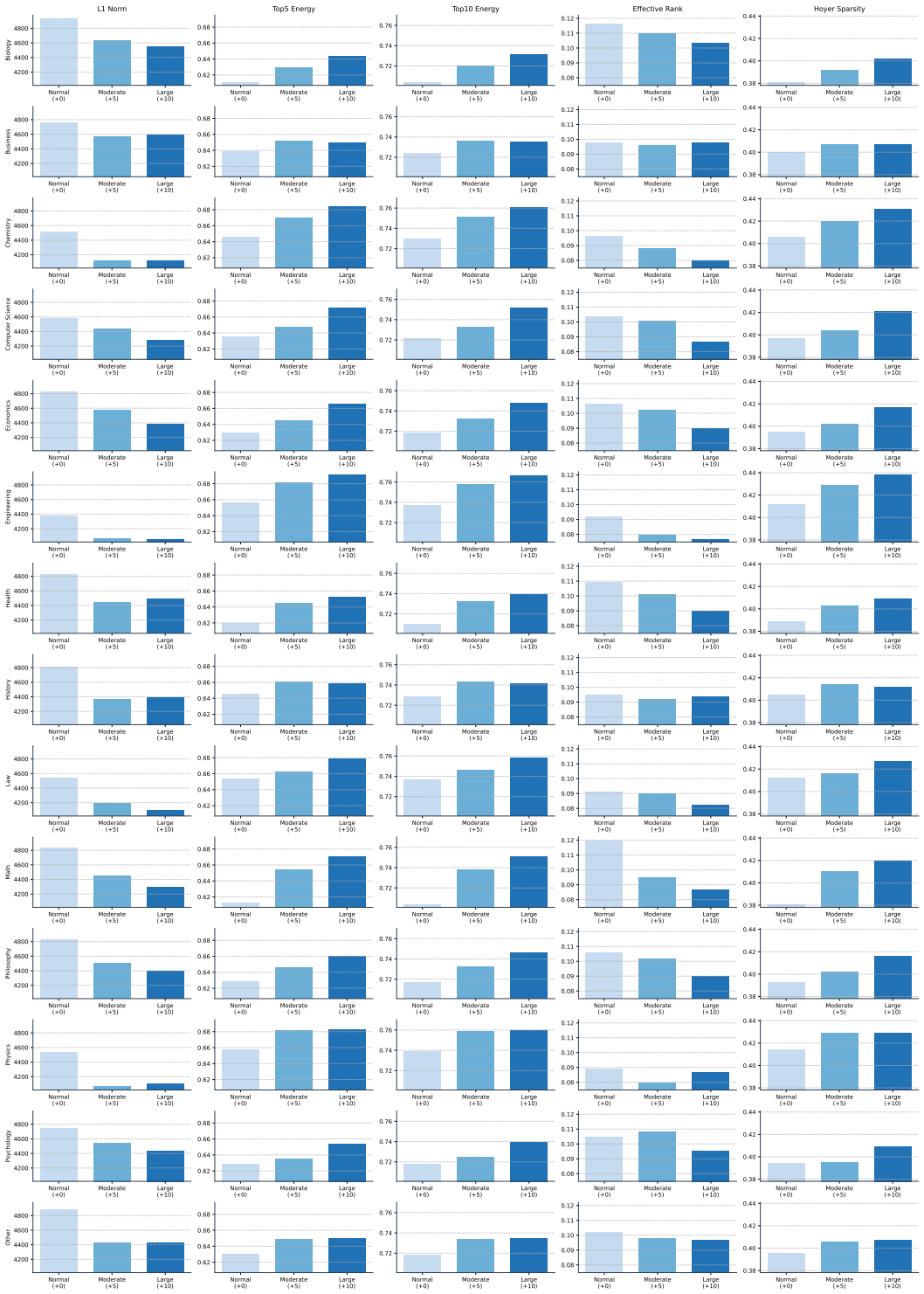
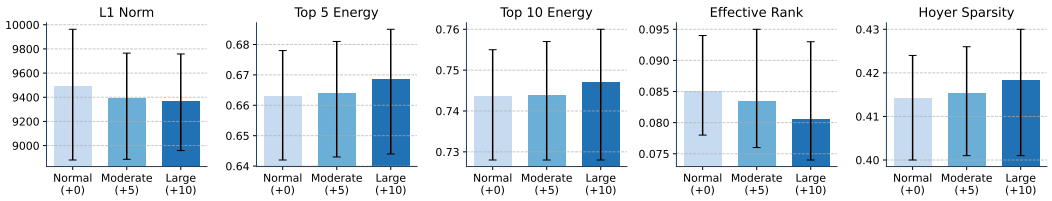
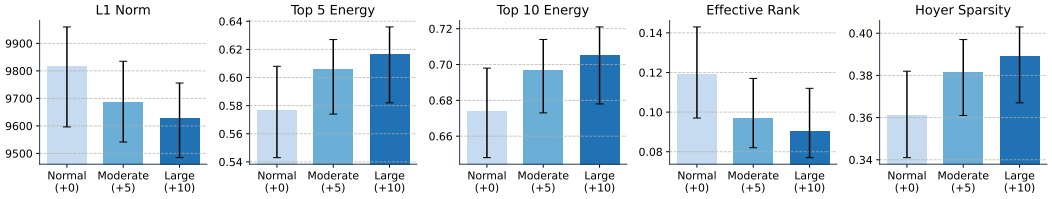


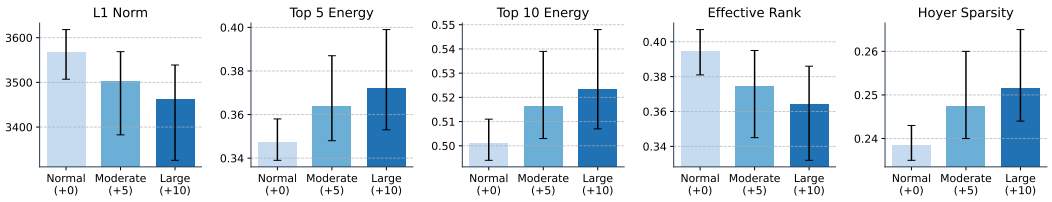
Figure 13: Per-domain sparsity statistics for `Qwen2.5-3B-Instruct` on MMLU-Robust.



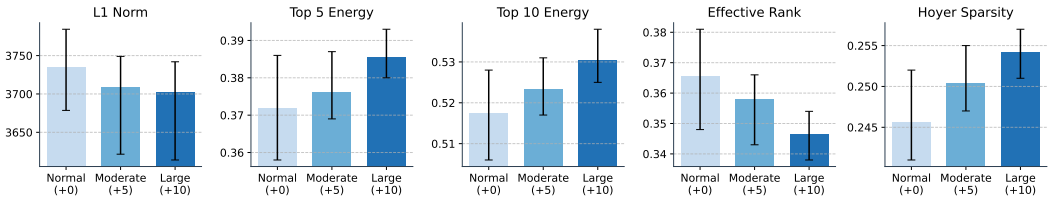
(a) **Qwen2.5-7B-Instruct**



(b) **Qwen2.5-14B-Instruct**



(c) **Llama-3.2-1B-Instruct**



(d) **Llama-3.2-3B-Instruct**

Figure 14: Average sparsity statistics on MMLU-Robust for more LMs.

tration and Hoyer sparsity. While the absolute metric levels vary by subject, the direction of change is highly consistent, supporting the conclusion that the "harder-is-sparser" phenomenon is not driven by a small subset of domains but persists broadly across academic areas.

A.10 Long Context Reasoning Details

To further probe whether our "harder-is-sparser" finding extends beyond adversarial perturbations, we study a *long-context reasoning* setting where difficulty is increased by expanding the background context length. Concretely, we compare short-context prompts against a long-context (32K) variant constructed from LongReasonQA, which controls the amount of background evidence while keeping the core question unchanged. This setup isolates the effect of contextual complexity (more tokens, more distractors, and longer-range dependency) on internal representations.

Figure 15 visualizes the distribution of last-layer activations under short vs. 32K context. The long-context condition exhibits a visibly more concentrated distribution, and the accompanying sparsity metrics shift consistently in the "sparser" direction: the normalized ℓ_1 magnitude decreases, while Top-k energy concentration increases. These results mirror the robustness experiments in the main text, suggesting that increasing task complexity via longer contexts induce a sharper, more selective representation at inference time.

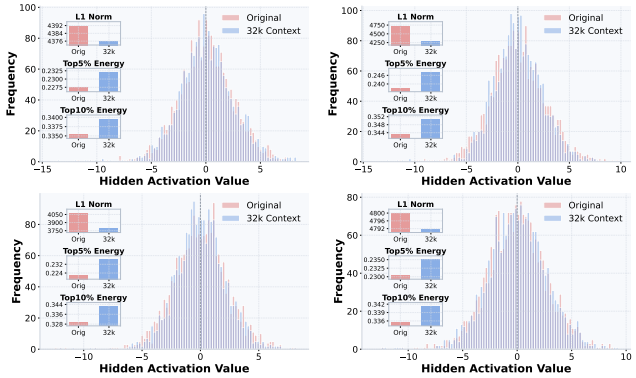
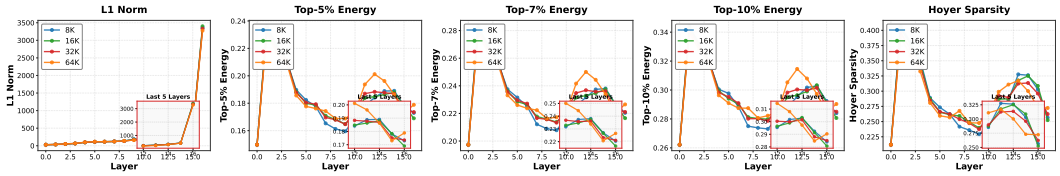
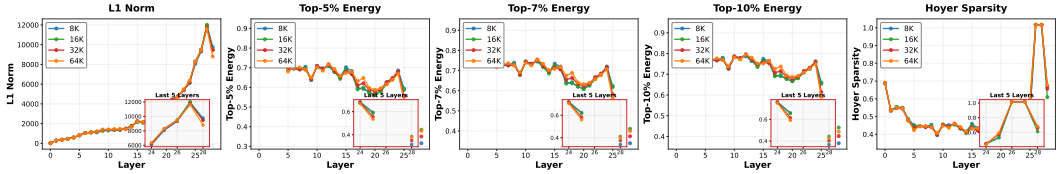


Figure 15: **Task complexity induces sparsity.** Representation activation showing increased sparsity in harder tasks (32k context, blue) compared to original settings (Short context, red). Three metrics consistently demonstrate higher sparsity for challenging inputs.



CONTEXT: Sparsity metrics across all layers for Llama3.2-1B under varying context lengths (8K, 16K, 32K, 64K).

Figure 16: **Layer-wise Sparsity across Context Lengths.** While intermediate layers show minimal variation across contexts, the final layers exhibit sharp divergence: longer contexts consistently produce sparser representations. This experiment is conducted on LongReasonQA (Li et al., 2025a), which can control the background context length.



CONTEXT: Sparsity metrics across all layers for Qwen2.5-7B under varying context lengths (8K, 16K, 32K, 64K).

Figure 17: **Layer-wise Sparsity across Context Lengths.** While intermediate layers show minimal variation across contexts, the final layers exhibit sharp divergence: longer contexts consistently produce sparser representations. This experiment is conducted on LongReasonQA (Li et al., 2025a), which can control the background context length.

We further examine *where* this sparsification emerges across the depth of the LLM. Figure 16 reports layer-wise sparsity metrics for `Llama-3.2-1B-Instruct` across 8K/16K/32K/64K contexts. A clear pattern appears: intermediate layers remain largely stable across context lengths, whereas the final layers diverge sharply, with longer contexts producing consistently sparser activations (lower ℓ_1 Norm and effective rank, higher Top-k energy and Hoyer sparsity). This localization to late layers supports the view that long-context difficulty primarily changes the *final-stage computation* that consolidates evidence into the prediction-relevant subspace, rather than uniformly reshaping representations throughout the entire LLM.

LongReasonQA (Ling et al., 2025) is a long-context reasoning benchmark that systematically controls the amount of background context (e.g., 8K/16K/32K/64K) while keeping the core question and answer format fixed, enabling a clean test of how increasing context length affects model reasoning and internal representations as Figure 18.

LongReason (8K) example (truncated background)

Background (truncated):
`Following the destruction of the house and its contents, Parks filed an insurance claim ...`
⋮
`... Understanding this ejection process is essential for comprehending how “Omega” achieves its propulsion objectives.`

Question:
In the context of space station “Omega,” what does “these fluids” refer to?

Choices:
(A) Wave modulator (B) Ion converter (C) Pulsed high-energy currents (D) Ion streams

Gold answer: (D)

Figure 18: **A LongReason 8K instance.** We show one example used in our long-context evaluation. For readability, the background is truncated (head + tail); the full context is available in the dataset.

LongReason (8K) example (paraphrased & truncated)

Background (paraphrased, truncated):
`Note that adblockers might block our captcha, and other functionality on BHW so if you don’t see the captcha or see reduced functionality please disable adblockers to ensure full functionality, note we only allow relevant management verified ads on BHW.`
⋮
`Briner’s observations confirm creationist Ice Age studies. It is most reasonable, given the Bible’s history, that there was a single Ice Age, and that the largest extent of ice lasted for only hundreds of years, beginning around 2400 BC. It is possible that the period of time wherein the “lion’s share” of melting apparently occurred was the only time that it occurred. The remaining thousands of years would therefore be a product not of data, but of the assumption of deep time.`

Question:
Which of the following, if true, does not weaken the viewpoint about the restoration of the jade pine forest in the country of Novaland after a chemical spill?

Choices:
(A) The pollution in the forest may have reduced to a level where some trees can survive.
(B) Apart from artificially planted species, wild plants have also appeared.
(C) Certain specific pollutants have been removed, but the heavy metal content in the forest soil is still higher than before.
(D) The types and numbers of birds appearing in the forest are still very few, not reaching the scale before the pollution.

Gold answer: (B)

Figure 19: **A LongReason 8K instance (paraphrased).** We follow the same presentation format as our long-context example: truncated background + multiple-choice inquiry + gold label.

A.11 Pretraining data construction details

A.11.1 Model and training setup.

We train a randomly initialized, decoder-only Transformer following the LLaMA architecture. Given a configuration `llama-`L`-`H, we set the hidden size to $d = 64H$, the number of layers to L, the number of attention heads to H, and the MLP intermediate size to $2d$. The model is trained with a lightweight character-level tokenizer whose vocabulary consists of :{`Q, P, 0-9, space, newline, '-', '?', <BOS>, <EOS>, <PAD>, <UNK>`}. Training data are generated from a latent-rule

knowledge graph; each sample is a single triple serialized as "`Q{h} P{r} Q{t}\n`" and optimized with the standard next-token prediction objective (labels identical to inputs, with padding labels masked by `-100`). We use the HuggingFace `Trainer` with per-device batch size 32, learning rate 10^{-4}, cosine learning-rate schedule, warmup ratio 0.2, and no weight decay. Unless otherwise specified, we train for `max_steps` steps (with optional BF16) and save a single checkpoint at the end of training.

A.11.2 Synthetic knowledge graph generation.

We construct a synthetic knowledge graph (KG), rather than a real-world one, because we can control the graph's size and complexity. By coupling a latent rule set with a graph growth process, we follow the setting of (Wang et al., 2025a). We first sample a collection of acyclic relational rules, where each rule is represented as a relation chain $(r_0 \Leftarrow r_1, \ldots, r_\ell)$ with $\ell \in [L_{\min}, L_{\max}]$. Next, we instantiate these rules on fresh entity nodes to form an initial backbone graph: for each rule, we create a directed path following the body relations $(r_1, \ldots, r_\ell)$ and additionally insert the implied head edge r_0 between the same endpoints. After building the initial backbone graph, we grow it to the desired size by adding one new entity at a time. For each new entity, we create m connections to existing entities. The relation type of each new connection is not chosen completely at random; instead, it is restricted by a relation-adjacency map that records which relations tend to appear together in the sampled rules. Optionally, we bias the attachment toward already well-connected entities, producing a power-law degree distribution.

To inject deductive structure, we periodically apply a subset of rules to existing nodes via stochastic traversal. For each candidate head entity, we follow the rule body edges on the current graph (with probability p_{mcmc} at each hop) to obtain reachable tail entities; whenever a rule fires, we add the implied triple (h, r_0, t) and record its supporting rule(s). We refer to such implied edges as *deductible* triples, while the remaining edges are treated as *atomic* triples. We then form the training set as a mixture of atomic and deductible triples controlled by a deductible ratio ρ and remove all non-training edges from the graph to prevent leakage. Finally, we design three evaluation splits: an in-distribution (ID) set sampled from the training triples (measuring memorization), and two out-of-distribution (OOD) sets composed of deductible triples stratified by their shortest supporting rule length, resulting in long-path and short-path OOD tests. Short-path OOD contains deductible triples whose shortest supporting rule is only 1–2 steps, while long-path OOD requires at least 3 steps. In our experiments, the short path is consistently harder than the long path. The key reason is that, under our synthetic generator, short rules are typically *broad* while long rules are more *selective*. For OOD-SHORT, the rule body is easy to satisfy, so the same query (h, r) often admits multiple deducible tails (i.e., several t's are consistent with the rules and the current graph), creating substantial ambiguity and stronger competition among plausible answers. In contrast, OOD-LONG imposes stricter compositional constraints, so far fewer paths match the full chain, and the set of valid deducible tails is usually much smaller (often close to a unique solution). As a result, the long-path split is easier, whereas the short-path split is harder in our setting. The structure of the knowledge graph construction can be found in Table 5.

A.11.3 More details about the learning dynamic experiment.

Figure 20 provides a more complete view of the learning dynamics discussed in subsection 4.3. In addition to the ℓ_1 trend shown in the main text, we track scale-normalized sparsity metrics and accuracy throughout pretraining on the controlled graph environment. The model is trained only with the next-token prediction objective, and checkpoints are evaluated on three splits: ID (Easy), OOD (Medium), and OOD (Hard).

The results reveal the same two-phase pattern from a complementary perspective. During the early stage of training, Top-5% and Top-10% Energy increase while Effective Rank decreases, indicating that the representation becomes increasingly concentrated on a smaller set of active dimensions. This corresponds to an initial feature-selection or pruning phase. After this early concentration phase, Top-k Energy gradually decreases and Effective Rank recovers, while accuracy continues to improve or stabilize, especially on the ID split. This suggests a later feature-consolidation phase in which the model no longer relies on a few sharp coordinates, but distributes useful information across a broader and more stable representation. Importantly, because Top-k Energy and Effective Rank are concentration-based metrics, this trend is not merely an artifact of raw ℓ_1 scale changes. Together

Stage	Synthetic KG construction (as implemented)
Universe	Entities are symbolic nodes $Q_0, Q_1, \ldots, Q_{n-1}$; relations are $P_0, P_1, \ldots, P_{n_r-1}$.
Rule sampling	Sample a pool of candidate rules as relation chains: $(r_0 \Leftarrow r_1, r_2, \ldots, r_\ell)$ with $\ell \in [L_{\min}, L_{\max}]$. Optionally apply length-weighted sampling with temperature τ (shorter rules sampled more often). Acyclicity is enforced via a dependency graph over relations.
Backbone graph initialization	Construct an initial directed multigraph by instantiating each sampled rule on fresh entity nodes: create a path $Q \xrightarrow{r_1} \cdots \xrightarrow{r_\ell} Q'$ and add the implied edge $Q \xrightarrow{r_0} Q'$. Maintain per-relation entity pools ("repeated entities") for later attachment.
Graph growth (structural prior)	Iteratively add new nodes until reaching n entities. Each new node attaches m edges by sampling relations constrained by an adjacency map derived from rule co-occurrence (also supporting a power-law attachment mode).
Deductible edge injection	Periodically enumerate candidate heads h and attempt to apply a subset of rules ("deductible rules") by stochastic traversal: follow each rule body with probability p_{mcmc} to obtain reachable tails t; if successful, add the implied triple (h, r_0, t) as a *deductible* edge and record its supporting rule(s).
Train graph pruning	Form training triples as a mixture of *atomic* edges (not derivable) and *deductible* edges (derivable), controlled by deductible ratio ρ. All edges not selected for training are removed from the graph to avoid leakage.
Test set design (difficulty)	Three evaluation splits are constructed: (i) **ID/Easy**: sampled directly from training triples (memory); (ii) **OOD-Long/Medium**: deductible triples supported by *long* rules (e.g., ≥ 3 steps); (iii) **OOD-Short/Hard**: deductible triples supported by *short* rules (e.g., 1–2 steps). For OOD splits, "seen tails" are collected from alternative deductive paths for harder negative sampling.
Text serialization	Each triple is serialized as a character-level sequence: `Q{h} P{r} Q{t}` (plus newline), enabling a lightweight tokenizer and controlled vocabulary.

Table 5: Synthetic knowledge graph generation via a latent-rule process: rule sampling, graph growth, deductible edge injection, and difficulty-controlled evaluation splits.

with Figure 6, these results support the interpretation that sparsity evolves systematically during training rather than appearing as a static property of the architecture.

A.12 An Extended Discussion and Future Work

Our investigation establishes a robust empirical law: *The farther the distribution shift, the sparser the representation.* Beyond the raw observations, this section synthesizes the underlying mechanisms and the broader implications of this phenomenon.

We hypothesize that explicitly training for "OOD-like" sparsity could improve model robustness and generalization, effectively simulating the "active suppression" mechanism during the learning phase. Our experiments utilized dense architectures (Llama series). Mixture-of-Experts (MoE) models introduce structural sparsity by design. Investigating whether the "OOD-induced sparsity" phenomenon persists in MoE architectures, or whether the routing mechanism absorbs the shift, would be critical for generalizing our findings to the next generation of sparse foundation models. Our study focused on reasoning tasks where the model generally attempts to solve the problem. An intriguing direction is to analyze sparsity patterns during hallucinations. Does a model exhibit *low* sparsity (high confidence/confusion) when it hallucinates, or does it exhibit *extreme* sparsity (information collapse)? Establishing a correlation between sparsity signatures and factual errors could lead to lightweight, white-box hallucination detection methods that do not require external fact-checking.

A.13 Limitations

While our findings offer significant insights into the mechanism of LLMs, we acknowledge several limitations in our current study. First, our proposed sparsity metrics (e.g., Top-k Energy) rely on white-box access to the model's last hidden states, rendering the method inapplicable to closed-source proprietary models like GPT-4o. Second, our experimental scope is primarily centered on logical reasoning (MATH-500) and long-context QA (LongReason); while we hypothesize that the sparsity-difficulty relationship generalizes to other domains, its applicability to tasks such as creative writing

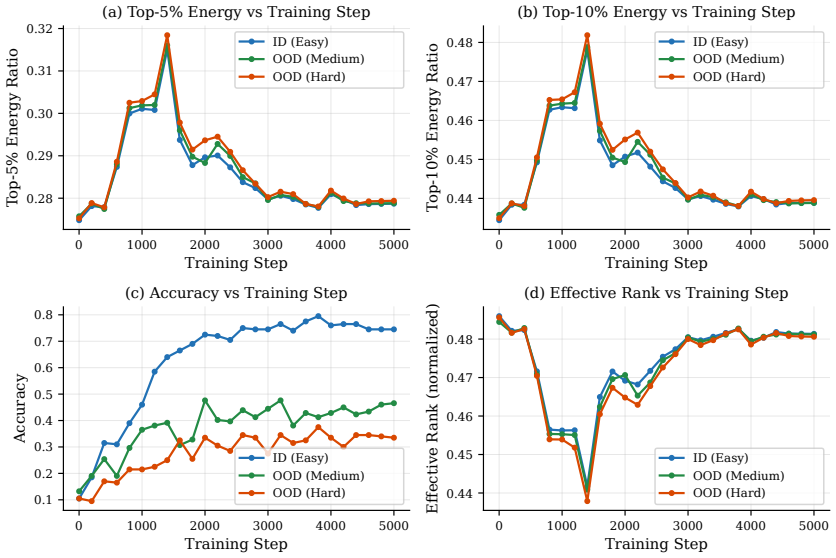


Figure 20: **Learning dynamics across sparsity metrics and accuracy.** We track Top-5% Energy, Top-10% Energy, accuracy, and normalized Effective Rank across training checkpoints for the controlled graph experiment. Early training shows increasing energy concentration and decreasing effective rank, while later training shows partial recovery of effective rank and stabilization of accuracy, supporting the two-phase feature-selection and feature-consolidation interpretation.

or multilingual processing remains to be fully verified. Finally, although SG-ICL improves reasoning accuracy, computing sparsity scores for large demonstration pools introduces pre-processing overhead compared to random sampling, though this cost can be effectively amortized by pre-computing and caching difficulty scores.

B The Use of Large Language Models (LLMs)

We used LLMs (e.g., ChatGPT) solely as general-purpose tools for **(i)** language editing and polishing of manuscript drafts and **(ii)** coding assistance for minor boilerplate (e.g., plotting and small utilities). All outputs were fully reviewed, modified, and tested by the authors. All research ideas, algorithmic designs, experiments, datasets, analyses, and conclusions were conceived and validated by the authors. LLMs were **not** used to produce results, annotations/ground truth, or methodological decisions. The authors take full responsibility for the content of this paper.

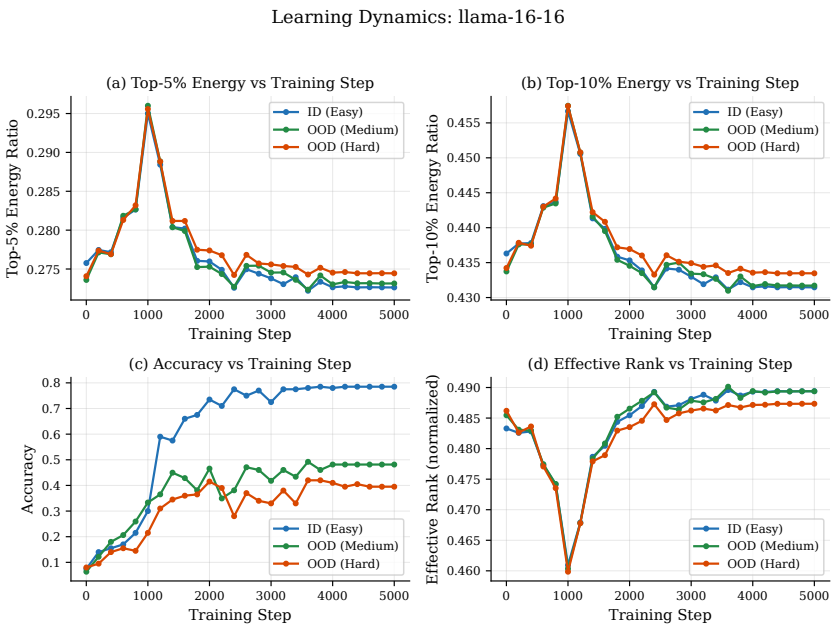


Figure 21: **Learning dynamics for Trans-16-16.** We track Top-5% Energy, Top-10% Energy, accuracy, and normalized Effective Rank across training checkpoints in the controlled graph environment. Compared with the larger Trans-32-32 model, Trans-16-16 exhibits a similar early concentration phase, where Top-k Energy increases and Effective Rank decreases, followed by a relaxation phase in which energy concentration weakens and Effective Rank recovers. This provides additional evidence that the sparsity dynamics are tied to feature selection and consolidation during training, rather than being an artifact of a single model scale.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state three research questions and corresponding contributions: (1) the sparsity–difficulty relationship across multiple controlled settings (Section 3), (2) learning dynamics explaining the mechanism (Section 4), and (3) Sparsity-Guided Curriculum ICL as a practical application. All claims are supported by experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section discussing three key limitations: (1) the requirement of white-box access to model hidden states, (2) the focus on reasoning and QA tasks with unverified generalization to other domains, and (3) the computational overhead of sparsity score pre-computation for SG-ICL.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides a theoretical derivation for the two-phase sparsity–density trajectory of the normalized ℓ_1 statistic in Appendix A.2. The simplified setting, assumptions, lemmas, and proofs are fully stated with numbered theorems and cross-references.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides an anonymous code repository, describes the sparsity metrics and experimental setup in the main text and appendix, specifies all models used (Qwen2.5-7B/3B, Llama3.2-3B, Llama3.1-8B), and details the synthetic data construction for the controlled environment experiments. All benchmarks used (MATH-500, DeepMATH-103K, LongReason, MMLU-Pro) are publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides an anonymized code repository at `https://anonymous.4open.science/r/RepSS-1C98`. All datasets used in the paper (MATH-500, DeepMATH-103K, MMLU-Pro, LongReason, ConflictReason) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes the experimental setup in Section 2 (Preliminaries), including the implicit reasoning setting, prompt design, and sparsity metrics. The controlled environment (Section 4.1) details the synthetic data construction, data splitting strategy, and model architectures. Additional details on hyperparameters and configurations are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The paper reports statistical significance tests: paired Student's t-tests with p-values (all $< 2.0 \times 10^{-29}$) for the knowledge conflict experiments (Section 3.3). Error bars indicating min/max across disciplines are shown in the MMLU-Robust experiments (Figure 3).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All inference experiments were conducted on NVIDIA A6000 GPUs. The main experiments involve inference on publicly available LLMs (Qwen2.5-7B/3B, Llama3.2-3B, Llama3.1-8B) and training a toy-sized Transformer from scratch.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. It studies the interpretability of publicly available LLMs using publicly available benchmarks and does not involve human subjects, private data, or potentially harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The work may have positive impacts by improving interpretability and enabling lightweight difficulty/OOD detection for LLMs. Potential negative impacts include over-reliance on sparsity as a proxy for difficulty or robustness, especially in domains where the sparsity–difficulty relation may not hold. We discuss these limitations and the need for domain-specific validation in Appendix A.13.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pre-trained language models, image generators, or scraped datasets. The released code and MMLU-Robust benchmark are analytical tools and evaluation data that pose no misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (MATH-500, DeepMATH-103K, MMLU-Pro, LongReason, ConflictReason) and models (Qwen2.5, Llama3) used in the paper are properly cited with their original publications. The models and datasets are used under their respective open-source licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces MMLU-Robust, a three-tier extension of MMLU-Pro with controlled distractor augmentation. Its construction methodology is described in Section 3.2 and Appendix A.9, and the code is included in the anonymized repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs (Qwen2.5-7B/3B, Llama3.2-3B, Llama3.1-8B) are the central subject of study in this research. Their usage, including model versions and inference configurations, is described throughout the paper, particularly in Section 2 (Preliminaries) and the experimental sections.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.