Sparse Brains are Also Adaptive Brains: Cognitive-Load-Aware Dynamic Activation for LLMs

Yiheng Yang[1,3], **Yujie Wang**[2*],
Chi Ma[2], **Lei Yu**[2], **Emmanuele Chersoni**[1], **Chu-Ren Huang**[1],
[1]Language Science and Technology, The Hong Kong Polytechnic University,
Hong Kong, China [2]Meituan, Beijing, China
[3]School of International Studies, Zhejiang University, Hangzhou, China
yvonneyoung0000@gmail.com, {wangyujie37, machi04, yulei37}@meituan.com,
{emmanuele.chersoni,churen.huang}@polyu.edu.hk

Abstract

Dense large language models (LLMs) face critical efficiency bottlenecks, as they rigidly activate all parameters regardless of input complexity. While existing sparsity methods (static pruning or dynamic activation) partially address this issue, they either lack adaptivity to contextual or model structural demands or incur prohibitive computational overhead. Inspired by the human brain's dual-process mechanisms — predictive coding (N400) for backbone sparsity and structural reanalysis (P600) for complex contexts — we propose **CLADA**, a ***C**ognitive-**L**oad-**A**ware **D**ynamic **A**ctivation* framework that synergizes statistical sparsity with semantic adaptability.

Our key insight is that LLM activations exhibit two complementary patterns: 1) *Global Statistical Sparsity* driven by sequence-level prefix information, and 2) *Local Semantic Adaptability* modulated by cognitive load metrics (e.g., surprisal and entropy).

CLADA employs a hierarchical thresholding strategy: a baseline derived from offline error-controlled optimization ensures over 40% sparsity, which is then dynamically adjusted using real-time cognitive signals. Evaluations across six mainstream LLMs and nine benchmarks demonstrate that CLADA achieves **20% average speedup with less than 2% accuracy degradation**, outperforming Griffin (over 5% degradation) and TT (negligible speedup).

Crucially, we establish the first formal connection between neurolinguistic event-related potential (ERP) components and LLM efficiency mechanisms through multi-level regression analysis ($R^2 = 0.17$ for sparsity-adaptation synergy). Requiring no retraining or architectural changes, CLADA offers a deployable solution for resource-aware LLM inference while advancing biologically-inspired AI design.

[*]Corresponding Author.

1 Introduction

Large Language Models (LLMs) achieve remarkable capabilities but face critical inference latency due to activating billions of parameters per token (Frantar and Alistarh, 2023). Existing sparsity-based solutions, including static pruning (Frantar and Alistarh, 2023; Sun et al., 2024; Ashkboos et al., 2024) and mixture-of-experts architectures (Zhang et al., 2022; Zhu et al., 2024; Szatkowski et al., 2024; Zheng et al., 2024; Pan et al., 2024; Zhong et al., 2024), generally lack the dynamic adaptability necessary for natural language processing (see Section 5).

Our analysis identifies three key limitations in current dynamic activation approaches, which we summarized in Figure 1: 1) DejaVu (Liu et al., 2023b) sparsity framework fails to generalized beyond ReLU (non-ReLU) activation requirements. 2) Threshold-based dynamic activation techniques like Threshold Truncation (TT) (Ma et al., 2024) incur prohibitive computational overhead that negates their theoretical speedup potential. 3) Training-free methods such as Griffin (Dong et al., 2024) exhibit significant performance degradation (see Table 1) due to their heuristic nature. These shortcomings motivate a fundamental research question: *Can we achieve training-free dynamic efficiency across architectures while preserving model capacity?*

	MMLU↑	TruthfulQA↑	Winogrande↑	GSM8K↑
LLaMA2-7B	45.83	61.04	74.11	13.95
DejaVu	27.02	51.12	50.2	7.22
TT	45.62	60.66	73.88	13.65
Griffin	43.59	59.26	73.21	12.31
CLADA	44.83	60.45	73.53	13.18

Table 1: Accuracy on benchmarks. Higher values are better. CLADA achieves competitive results on all benchmarks, closely rivaling the baseline LLaMA2-7B.

This work draws inspiration from cognitive neu-

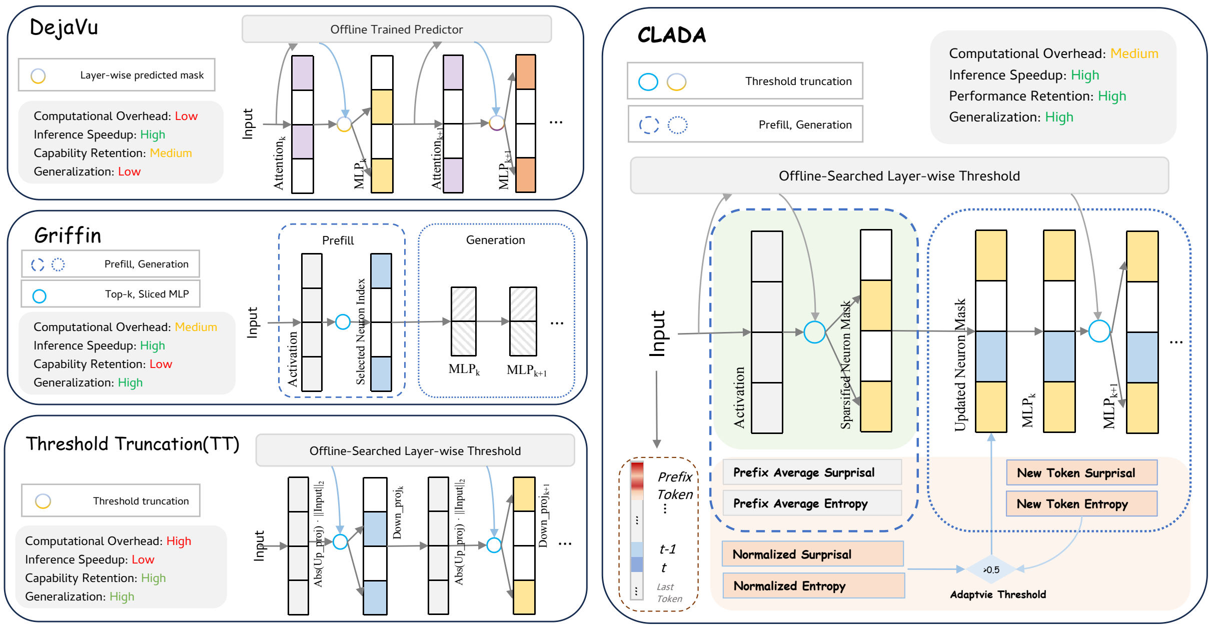

Figure 1: A comparative analysis of dynamic activation methods, highlighting CLADA's advantages in minimal computational overhead, substantial inference speedup, the ability to maximize model capability, and robust generalization ability across various LLM architectures.

roscience, where the human brain achieves remarkable efficiency through adaptive resource allocation. Neurophysiological studies reveal two complementary mechanisms: 1) **Predictive coding** via the N400 event-related potential (ERP), which minimizes semantic integration costs (Kutas and Federmeier, 2011; Li and Futrell, 2024), and 2) **Structural reanalysis** via the P600 ERP, which handles syntactic complexity (Kuperberg, 2007; Kuperberg et al., 2020). Analogous computational mechanisms, e.g. surprisal (Hale, 2001; Levy, 2008; Wilcox et al., 2023) and entropy, quantify cognitive load during language processing (Salicchi and Hsu, 2025). This N400-P600 duality in human language processing directly motivates CLADA's two complementary sparsity strategies: N400-like global sparsity for backbone processing and P600-like local adaptation for complex context.

We posit that LLMs can achieve similar efficiency through two principled sparsity mechanisms:

1. **Statistical Sparsity**: Exploits sequence-level prefix information to identify and sparsify redundant activations;

2. **Semantic Adaptability**: Dynamically allocates computational resources based on real-time cognitive load indicators.

Building on this insight, we propose ***Cognitive-Load-Aware Dynamic Activation*** (**CLADA**), a novel framework that synergizes offline error-controlled optimized threshold truncation with cognitive-load-aware adaptation. Our method introduces three key innovations:

Theoretical Foundation The CLADA framework is the first to link cognitive load metrics (surprisal, entropy) to LLM activation patterns; demonstrates their dual role in governing statistical sparsity and semantic adaptation.

Algorithmic Design CLADA is introducing a dynamic thresholding mechanism that automatically adjusts activation masks via sequence statistics (prefix-based prediction) and semantic complexity (cognitive-load-aware adaptability), without the need of model retraining.

Empirical Validation Extensive experiments across six LLM architectures and nine benchmarks reveal that CLADA achieves 20% average speedup with <2% performance degradation (Table 1 and 4), and outperforming existing methods in computational overhead and generalization (Table 1).

Our work bridges cognitive science with machine learning, offering a new paradigm for efficient, interpretable language models. By grounding dynamic activation strategies in cognitive theories, we take a significant step toward building LLMs that mirror the adaptive efficiency of biological neural systems.

2 Bridging Cognitive Load and Activation

2.1 Global Statistical Sparsity

Theoretical Framework Statistical sparsity characterizes the phenomenon where neuron activation patterns exhibit hierarchical dependence on sequence prefixes. We formalize this through Hypothesis 2.1:

Hypothesis 2.1. ***(Existence)*** *Statistical sparsity manifests as prefix-dependent activation flocking, where activation matrix similarity between hybrid sequences grows monotonically with consistent prefix length.*

This hypothesis aligns with hierarchical representation theories in language modeling (Dubey et al., 2022; Guo et al., 2024), where early tokens establish persistent activation patterns. To operationalize this, we develop a panel regression framework:

$$\Delta_{\text{sim}} = \mu_i + \beta\ell_{ij} + \alpha_i + \epsilon_{ij} \tag{1}$$

where:

- μ_i: Constant term.
- ℓ_{ij}: Prefix length, the subscript j represents data with varying prefix ratios
- α_i: The individual fixed effect term in panel data
- ϵ_{ij}: Random perturbation error term

2.1.1 Experimental Design

Our multi-stage validation framework employs controlled sequence generation and activation pattern analysis:

Corpus Construction Using the XSum dataset (Narayan et al., 2018), we obtain:

- **N**atural **L**anguage **S**equences (**NLS**): 2,000 random authentic text samples
- **R**andom **T**oken **S**equences (**RTS**): 2,000 sequences generated by permuting XSum vocabularies while preserving part-of-speech ratios and punctuation frequency

Hybrid Sequence Generation For each prefix ratio $\alpha \in \{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$:

1. Compute prefix length: $\ell = \lceil 2048\alpha \rceil$ ($512 \leq \ell \leq 1024$).

Algorithm 1 Empirical Validation of the Existence of Statistical Sparsity

```
Require: Sequences $A$, $B$; Prefix ratio $\alpha$
1: $A' \leftarrow$ generate_hybrid_sequence$(A, B, \alpha)$
2: $\mathbf{M}_A \leftarrow$ extract_activations$(A)$
3: $\mathbf{M}_B \leftarrow$ extract_activations$(B)$
4: $\mathbf{M}_{A'} \leftarrow$ extract_activations$(A')$
5: sim$_{AB} \leftarrow$ sim$(\mathbf{M}_A, \mathbf{M}_B)$
6: sim$_{A'B} \leftarrow$ sim$(\mathbf{M}_{A'}, \mathbf{M}_B)$
7: return $\Delta_{\text{sim}} = \frac{\text{sim}(\mathbf{M}_{A'},\mathbf{M}_B)}{\text{sim}(\mathbf{M}_A,\mathbf{M}_B)} - 1$
```

2. Create hybrid sequences A' by replacing first ℓ tokens of NLS-A/RTS-B with NLS-B/RTS-B counterparts.
3. Generate six experimental groups (36,000 sequences in total):
   - NLS-A, NLS-B, NLS-A'
   - RTS-A, RTS-B, RTS-A'

2.1.2 Activation Matrix Extraction

For each sequence, we extract Layer-15 activations from LLaMA-3-8B (Grattafiori et al., 2024) at Token-2049:

$$\mathbf{M}^{(l=15)} \in \mathbb{R}^{1\times(4096\times14336)}$$

The layer selection aligns with previous findings on mid-layer semantic integration (Skean et al., 2025). The rationale for selecting token-2049 is detailed in Appendix B.1.

2.1.3 Similarity Metric Calculation

We compute normalized activation similarity shift:

$$\Delta_{\text{sim}} = \frac{\text{sim}(\mathbf{M}_{A'}, \mathbf{M}_B)}{\text{sim}(\mathbf{M}_A, \mathbf{M}_B)} - 1 \tag{2}$$

where similarity is measured by centered kernel alignment (CKA) (Kornblith et al., 2019). To ensure the robustness of the regression, we additionally employed cosine similarity to quantify the similarity between activation matrices.

2.1.4 Empirical Validation

Experiment 1.1: RTS Groups Following Algorithm 1, construct hybrid sequence A' from RTS B according to Section 2.1.1. Activation matrix similarity is computed using Equation 2.

Observation 1: Regression results (Table 2 Column 1) reveal strong prefix-length effects ($\beta = 4.12, p < 0.001$), confirming Hypothesis 2.1. To assess the robustness of this finding with respect to the similarity metric, we further replicate the analysis using cosine similarity; the results remain highly consistent ($\beta = 4.22, p < 0.001$) and are reported in Appendix B (see Table 9).

Experiment 1.2: NLS Groups Replicate the aforementioned experimental procedures in Experiment 1.1, regression results of NLS groups are reported in the second column of Table 2.

	Dependent variable: $\Delta_{\text{CKA\_sim}}$					
	(1) RTS-A'	(2) NLS-A'	(3) RTS-A'	(4) NLS-A'	(5) RTS-A'	(6) NLS-A'
Prefix_len	4.12***	3.91***	4.05***	4.03***	3.75***	3.69***
	(22.03)	(20.16)	(21.53)	(21.37)	(19.22)	(18.84)
Surprisal			-0.02	-0.85***	-0.09	-0.80***
			(-0.26)	(-8.19)	(-0.14)	(-8.13)
Entropy					-0.36	-0.12***
					(-4.38)	(-5.87)
Token_len	10.38***	13.90***	10.14***	12.31***	11.23***	11.38***
	(5.06)	(5.06)	(5.07)	(5.05)	(5.07)	(5.06)
Obs	12000	12000	12000	12000	12000	12000
Adjusted R^2	0.13	0.12	0.15	0.16	0.17	0.16
Individual FE	YES	YES	YES	YES	YES	YES
Constant	YES	YES	YES	YES	YES	YES

Note: * p<0.1; ** p<0.05; *** p<0.01

Table 2: Regression results on the impact of *prefix_len* on CKA activation similarity

Observation 2: While maintaining statistical significance ($\beta = 3.91, p < 0.001$) (see Table 2 Column 2), the attenuated effect size compared to RTS ($\Delta\beta = 5.4\%$) suggests cognitive-load-aware interference in natural language processing. This observation motivates our investigation of semantic adaptability.

2.1.5 Visual Evidence and Case Study

To investigate the existence of Statistical Sparsity in a visual and intuitive way, we randomly extracted two entries from the XSum dataset as ***Natural Language Sequence-A*** (**NLS-A**) and NLS-B, and replaced the first 512 tokens of NLS-A with NLS-B to produce NLS-A'. Details can be seen in Appendix B.2. And a sentence-by-sentence case study can be seen in Appendix B.3

2.2 Local Semantic Adaptability: Cognitive-Load-Aware Response

Theoretical Foundation The NLS-RTS performance gap (4.12 vs 3.91) reveals a critical limitation of pure statistical sparsity: cognitive load disrupts prefix-dependent activation patterns. We formalize this complementary mechanism through Hypothesis 2.2:

Hypothesis 2.2. ***(Existence)*** *Semantic adaptability emerges through cognitive-load-modulated activation adjustments, quantified by surprisal (s_t) and entropy (H_t).*

2.2.1 Operationalization

We extend the regression framework to incorporate cognitive load metrics:

$$\Delta_{\text{sim}} = \mu_i + \beta\ell_{ij} + \gamma_1 s_{ijt} + \gamma_2 H_{ijt} + \alpha_i + \epsilon_{ij} \quad (3)$$

Cognitive Load Quantification Following psycholinguistic theory (Hale, 2016; Oh et al., 2024; Salicchi and Hsu, 2025), we define surprisal and entropy as:

$$s_t = -\log P(w_t|w_{<t}) \quad \text{(surprisal)} \quad (4)$$

$$H_t = -\sum_{w\in\mathcal{V}} P(w|w_{<t}) \log P(w|w_{<t}) \quad \text{(entropy)} \quad (5)$$

where:

- w_t: the target token
- $w_{<t}$: the context of target token
- $P(w_t|w_{<t})$: the conditional probability of target word w_t provided by the language model
- $\mathcal{V}$: the vocabulary of the language model enriched with special token (e.g., EOS)

The descriptive statistics of our dataset for surprisal and entropy are in Table 8 in Appendix B.

2.2.2 Empirical Validation

Experiment 2: Cognitive-Load-Aware Interference Key findings from Table 2 Columns 3-6:

- **RTS**: Non-significant cognitive-load-aware effects ($\gamma_1 = -0.07, \gamma_2 = -0.35, p > 0.1$)
- **NLS**: Significant inverse correlations ($\gamma_1 = -0.80, \gamma_2 = -0.12, p < 0.001$)

This demonstrates semantic adaptability's domain-specific nature: high surprisal/entropy tokens in NLS activate emergent neurons (ENs), disrupting prefix-driven activation patterns. The robustness analysis (Table 9 Column 3-6) confirms these effects persist across similarity metrics.

3 Methodology

3.1 Unified Framework: Cognitive-Load-Aware Mechanism

The interplay between statistical sparsity and semantic adaptability forms a *hierarchical activation mechanism*, where global efficiency is balanced by local precision.

Core Design

$$A_{\text{total}}^{(t)} = \underbrace{A_{\text{base}}^{(t)}}_{\text{Statistical Sparsity}} + \underbrace{\Delta A_{\text{semantic}}^{(t)}}_{\text{Semantic Adaptability}} \tag{6}$$

where:

- Statistical Sparsity ($A_{\text{base}}^{(t)}$): Provides a stable, sequence-level efficiency baseline. This represents the brain's ability to process easily predictable and familiar language with minimal effort. We establish a highly efficient baseline mode where the LLM activates only a core set of essential neurons to handle routine linguistic input, which is governed by a *base threshold* ($\tau_{\text{base}}^{(l)}$).

- Semantic Adaptability ($\Delta A_{\text{semantic}}^{(t)}$): Dynamically adjusts activation intensity to handle high-cognitive-load contexts. This is engaged when the brain encounters unexpected or complex information. This is achieved via a dynamic adjustment to the threshold.

We implement this dual-process framework through two interconnected components: a static baseline threshold and a dynamic adjustment mechanism.

3.2 Dynamic Threshold Strategy

During inference, only neurons whose activation magnitudes exceed the threshold are used. All other neurons remain dormant.

3.2.1 Neuron Activation Prefilling

For each layer l, compute neuron-wise activation magnitudes during the *prefilling* phase:

$$\begin{gathered} A_j^{(l)} = \left\| \left[\sigma(W_{\text{in},j}^{(l)} x) \odot V_{\text{in},j}^{(l)} x \right] W_{\text{out},j}^{(l)} \right\|_2 \\ \forall j \in [1, d_h^{(l)}] \end{gathered} \tag{7}$$

where j indexes neurons in layer l's hidden dimension $d_h^{(l)}$. $W_{\text{in},j}^{(l)}$ and $V_{\text{in},j}^{(l)}$ denote the j-th row of input projection matrices. $W_{\text{out},j}^{(l)}$ is the j-th column of output projection matrix.

3.2.2 Layer-wise Base Threshold

To balance between performance and efficiency, we determine optimal base thresholds through error-controlled optimization searching:

$$\begin{aligned} \tau_{\text{base}}^{(l)} &= \arg\max_{\epsilon^{(l)}} \epsilon^{(l)} \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{val}}} \left[\text{CETT}^{(l)}(\mathbf{x}; \epsilon^{(l)}) \right] \leq 0.2 \\ \text{where} \quad \text{CETT}^{(l)} &= \frac{\left\| \sum_{j \in \mathcal{S}_{\text{cut}}^{(l)}} n_j^{(l)}(\mathbf{x}) \right\|_2}{\left\| MLP^{(l)}(\mathbf{x}) \right\|_2} \\ \mathcal{S}_{\text{cut}}^{(l)} &= \left\{ j \mid A_j^{(l)} < \epsilon^{(l)} \right\} \end{aligned} \tag{8}$$

where token-level metrics are computed as Equation 4 and Equation 5.

3.2.3 Token-aware Threshold Adjustment

In this step, we integrate token-level cognitive signals with layer-wise base thresholds in Equation 9

Indicator Function $\mathbb{I}(\cdot)$ evaluates to 1 when the specified condition is satisfied and 0 otherwise.

Universal Cognitive Thresholds τ_s, τ_H are shared across layers and are set to the median (50th percentile) of the historical surprisal and entropy distributions within the current sequence, respectively. Specifically, both τ_s and τ_H are set to the median (50th percentile) of their respective historical distributions computed over the current sequence (see Figure 1).

Layer-specific Scaling Coefficients $\lambda^{(l)}, \gamma^{(l)}$ control the magnitude of threshold modulation induced by surprisal and entropy. In practice, we instantiate these coefficients from the corresponding partial regression estimates reported in Table 2 (Column 6), where surprisal and entropy coefficients are 0.80 and 0.12, respectively.

Algorithm Implementation can be found at Algorithm 2.

4 Experiments

4.1 Experimental Setup

Our approach, along with the baseline models, is implemented using the PyTorch framework, and we leverage the Hugging Face Transformers library for model and dataset management. Our experiments are powered by 1 NVIDIA A100 GPUs with 80 GB of memory. Following Section 3, we sequentially applied our methods for each Transformer layer, which reduces inference latency while preserving model performance.

$$\tau_{\text{final}}^{(l)}(t) = \tau_{\text{base}}^{(l)} \cdot \left[1 + \lambda^{(l)} \cdot \mathbb{I}(s_t > \tau_s) + \gamma^{(l)} \cdot \mathbb{I}(H_t > \tau_H)\right] \tag{9}$$

Algorithm 2 CLADA Inference

Input: Model θ, prefix $x_{1:T}$, max gen_len N.
Output: Generated sequence $y_{1:N}$.

```
1:  **Prefill Phase**:
2:  Search $\tau_{\text{base}}^{(l)}$ and fit $\lambda^{(l)}, \gamma^{(l)}$ for each layer $l$.
3:  Generate initial activation masks $\{\text{Mask}^{(l)}\}$ using $\tau_{\text{base}}^{(l)}$.
4:  **Generation Phase**:
5:  **for** $t = 1$ to $N$ **do**
6:      Compute $s(w_{<T}^{t-1})$ and $H(w_{<T}^{t-1})$.
7:      Update thresholds:$\tau_{\text{final}}^{(l)}(t)$
8:      Regenerate masks $\{\text{Mask}^{(l)}\}$ by thresholding $A_l$.
9:      Perform sparse forward pass using $\{\text{Mask}^{(l)}\}$ to predict $y_t$.
10:     Update context: $w_{<T}^{t} = w_{<T}^{t-1} \cup \{y_t\}$.
11: **end for**
```

All experiments are conducted in a single phase, without any post-training or fine-tuning stages.[1]

Models In this paper, we conducted a comprehensive series of experiments using the OPT-350M, OPT-2.7B, Gemma-2B, LLaMA-2-7B and LLaMA-3-8B and Mistral-7B models. These models represent a significant advancement in language modeling capabilities, providing a spectrum of scales to meet various computational needs and performance benchmarks.

Tasks and Datasets Following Griffin (Dong et al., 2024), we conduct evaluations on a variety of models across multiple generation and classification tasks. For generation tasks, we focus on XSum (Narayan et al., 2018), CNN/DailyMail (Nallapati et al., 2016), COQA (Reddy et al., 2019), and QASPER (Shaham et al., 2022). For classification tasks, our evaluation includes HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), COPA (Roemmele et al., 2011), ARC-Challenge (Clark et al., 2018), and BoolQ (Clark et al., 2019). Except for XSum and CNN/DailyMail, our experiments utilize the LM Evaluation Harness (Gao et al., 2023).

[1]The code is publicly available at `https://github.com/Oldify/CLADA`

Baselines Besides comparing against the original LLM, we also evaluate CLADA in relation to Griffin and TT. Unless specified otherwise, each technique is applied in a layer-wise manner, enhancing scalability even when dealing with exceptionally large models. TT has similary performance with CLADA, therefore we only evaluate its generation phase latency. For DejaVu, we did not consider it as a comparable baseline in subsequent experiments (see Table 1).

Sparsity In our evaluation, we especially focus on the MLP blocks of LLM models, which constitute approximately 67% of the parameters of model's two main blocks, making them a crucial target for dynamic activation. Griffin and CLADA achieve around 50% of sparsity in total.

4.2 Performance Evaluation

Table 3 delineates the performance differences between the Griffin and CLADA methods across various generation and classification tasks. Metrics such as Accuracy (Acc), Rouge-1, and F1 scores were measured across various datasets.

Our comprehensive evaluation across six model architectures and nine benchmarks reveals four key findings regarding CLADA's effectiveness:

Consistent Superiority Across Scales. CLADA consistently outperforms Griffin in all model sizes. For instances:

- Small (OPT-350M): $+1.48\%$ on Hellaswag
- Medium (OPT-2.7B): $+1.34\%$ on Piqa
- Larger (LLaMA-3-8B): $+1.30\%$ on BoolQ

Task-Agnostic Benefits. The advantages of CLADA are evident in both discriminative and generative tasks. In classification tasks, it demonstrates an increase in accuracy of up to 2% on reasoning-intensive benchmarks (ARC-C, Coqa). In generation tasks, it achieves an increase in Rouge-1/F1 of up to 2.35% on summarization (XSum, CNN).

Scaling Characteristics. The performance gap widens with model capacity. Griffin relies on a hard top-k, which may discard neurons that are contextually important. In contrast, CLADA adopts cognitive-load-aware thresholds, providing greater adaptability.

	Acc					Rouge-1		F1	
Models	**Hellaswag**	**Piqa**	**Copa**	**Arc-c**	**Boolq**	**Xsum**	**Cnn**	**Coqa**	**Qasper**
OPT-350M	32.06	64.64	72.00	21.33	41.01	12.89	14.82	33.39	3.34
Griffin	30.52	62.46	69.00	20.24	39.71	10.59	13.32	31.89	2.14
CLADA	32.00	64.04	72.00	20.73	40.76	11.23	13.47	32.24	2.45
OPT-2.7B	45.86	73.78	77.00	60.77	66.79	18.43	22.24	64.41	7.85
Griffin	43.76	71.84	76.00	58.21	65.92	17.43	20.74	62.91	6.85
CLADA	45.74	73.18	76.00	58.42	66.19	17.86	21.33	64.05	7.70
Gemma-2B	71.40	77.30	83.00	42.10	69.40	15.69	23.32	72.03	12.46
Griffin	70.03	76.34	82.00	41.19	68.42	14.69	22.18	71.78	11.83
CLADA	70.85	76.21	82.00	41.19	68.21	15.32	22.51	72.45	12.33
LLaMA-2-7B	57.16	78.07	87.00	43.34	77.71	27.15	10.08	77.35	26.31
Griffin	56.66	76.57	85.00	41.84	76.21	26.65	8.58	75.85	25.81
CLADA	56.86	77.67	86.00	42.84	77.51	26.85	9.98	76.95	26.11
LLaMA-3-8B	62.53	81.85	93.00	46.29	80.76	29.62	12.21	82.92	28.86
Griffin	62.03	80.35	91.00	43.79	78.26	27.12	11.71	82.42	27.36
CLADA	62.31	81.40	92.00	45.79	80.39	29.47	11.93	82.57	28.37
Mistral-7B	61.21	80.58	92.00	50.43	83.61	28.67	28.00	80.70	24.56
Griffin	59.71	79.08	92.00	47.43	82.11	27.17	26.50	78.20	22.06
CLADA	59.32	79.21	92.00	49.24	83.14	28.35	27.53	80.55	24.07

Table 3: Generation and classification performance across various model architectures. Higher values are better.

Models	**Dense**	**TT**	**Griffin**	**CLADA**
OPT-2.7B	32.95	33.52	26.96(22.22%↓)	27.77(18.65%↓)
Gemma-2B	30.17	30.16	23.92(26.13%↓)	24.06(25.39%↓)
LLaMA-3-8B	81.31	79.88	63.32(22.13%↓)	64.03(21.25%↓)
Mistral-7B	79.28	76.26	63.26(25.32%↓)	63.94(19.34%↓)

Table 4: Generation phase latency(s).

Out-of-Distribution (OOD) Robustness Unlike training-dependent methods, CLADA is a training-free, token-level dynamic activation framework, which operates in a distribution-agnostic manner, computing activations dynamically for each token without reliance on training data distributions. As shown in Table 5, CLADA achieves $14.7\%-16.3\%$ accuracy improvements across five benchmarks, including HellaSwag. This design enables robust generalization under distribution shifts.

4.3 Efficiency Evaluation

Table 4 reports generation latency measurements for multiple models executed on a single NVIDIA A100 GPU. All experiments use a batch size of 1 and FP16 precision implementations from Hugging Face. Both the prompt length and the generated new token length are fixed at 1024, where the sparsity of both Griffin and CLADA are configured with 50%. In addition, the evaluations of extended context are reported in Table 6. All latency metrics in Table 4 are in seconds.

The results demonstrate that the CLADA method consistently reduces generation latency compared to the dense configuration across all evaluated models. As shown in Table 4, both Griffin and CLADA offer great speedups, ranging from 18–25%, whereas TT maintain latency comparable to dense models due to its reliance on extensive token-level threshold search.

Overall, these results confirm that CLADA effectively accelerates inference while maintaining competitive task performance.

4.4 Ablation Studies

We conduct systematic ablation studies to quantify the contributions of CLADA's core components. All experiments maintain identical hyperparameters and evaluation metrics as the main experiments. Ablation study on larger batch size and comparisons with non-cognitive alternatives (e.g., activation-magnitude and norm-based thresholding) are provided in Appendix C and Appendix D.

Component Contribution Analysis Table 5 reveals three key findings through controlled component removal:

- **Statistical Sparsity**: Disabling prefix-based

Variant	**Accuracy Retention (%)**				**Latency (s)**	
	HellaSwag	**PIQA**	**ARC-C**	**XSum**	**OPT-2.7B**	**LLaMA-3-8B**
CLADA (Full)	99.6	99.5	98.9	99.5	27.8	64.0
w/o Stat. Sparsity	80.4	82.1	78.3	76.3	26.9	63.3
w/o Sem. Adapt.	97.1	97.7	96.5	98.3	26.3	62.8
Top-P Threshold	83.3	85.3	86.5	83.6	26.8	62.9

Table 5: Component ablation study. Accuracy Retention = (Variant Score / Dense Baseline) $\times$ 100. Latency measures total time for generating 1024 tokens. Top-P = 0.5 serves as static threshold baseline.

Models	**Prompt Len**	**Generation Len**	**Dense**	**Griffin**	**CLADA**	**Diff to Griffin**
LLaMA-3-8B	1024	1024	80.16	63.32	65.13	-1.68
(*context_len_limit=8K*)	2048	1024	84.34	68.08	68.09	-0.01
	2048	2048	87.92	75.86	76.37	-0.51
	4K	2048	92.08	78.03	77.78	0.25
	8K	2048	96.53	84.00	83.79	0.21
Mistral-7B	1024	1024	79.28	63.26	64.94	-1.68
(*context_len_limit=32K*)	2048	1024	81.37	67.48	68.03	-0.55
	2048	2048	85.46	75.74	76.15	-0.41
	4K	2048	89.26	77.46	77.41	0.05
	8K	2048	93.43	83.79	83.22	0.57
	16K	2048	97.68	86.41	84.50	1.91
	32K	2048	102.35	91.35	89.78	1.57

Table 6: Long-context efficiency comparison (latency in seconds). CLADA outperforms Griffin by up to $1.9s$ at maximum context lengths.

activation patterns (**w/o Stat. Sparsity**) causes $17.6 - 23.7\%$ performance degradation across tasks, while only reducing latency by $1.8 - 3.2\%$. This demonstrates its critical role in preserving model capacity through sequence-level regularities.

- **Semantic Adaptability**: Removing cognitive-load-aware adjustments (**w/o Sem. Adapt.**) maintains latency parity but reduces accuracy by $2.4 - 3.4\%$, confirming its usefulness for handling high-entropy contexts.

- **Dynamic Thresholding**: Replacing our adaptive thresholds with static Top-P = 0.5 (**Top-P Threshold**) leads to $14.7 - 16.3\%$ performance drops, validating the superiority of our theoretically grounded approach.

Context Length Scalability Table 6 demonstrates CLADA's superior scalability to long contexts compared to Griffin. While both methods exhibit latency growth with increasing context length, CLADA achieves $1.57 - 1.91s$ speedup at 32K context due to better memory access patterns in our dynamic thresholding strategy.

5 Related Work

5.1 The Hint of Sparsity

LLMs often appear to have excessive activation of neurons during tasks, leading to inefficiency and waste of resources (Bommasani et al., 2022; Yuan et al., 2024). Several studies (e.g. Liu et al. (2023a)) show that dense neural networks often display surplus activation. Treating sparsity as a continuous process can optimize model architecture holistically. The Lottery Hypothesis (Frankle and Carbin, 2019; Malach et al., 2020) highlights pruning techniques to remove unnecessary connections and leverage inherent sparsity.

5.2 Static Sparsity

Early efforts to improve LLM efficiency focused on **static sparsity**, where redundant parameters are permanently removed or compressed.

Weight Pruning Methods like SparseGPT (Frantar and Alistarh, 2023) prune weights based on magnitude criteria, achieving up to 50% sparsity without retraining. However, such approaches ignore input-specific sparsity patterns, leading to suboptimal performance on dynamic tasks like dialogue generation. Static methods lack adaptability to varying input contexts and often require retraining to get back to the original performance level.

5.3 Dynamic Activation

Dynamic activation (DA), which selectively activates subsets of neurons during inference, offers a balance between efficiency and flexibility. Existing research on dynamic activation methods can be classified as Training-Dependent, which often requires architectural changes or additional training, and Training-Free, which aims to achieve efficiency without retraining (see in Appendix A.1 and Table 7) . Our proposed framework, CLADA, falls into the training-free category.

5.4 Cognitive Load and Language Modeling

Cognitive research on human language processing offers principled guidance for designing efficient computational models. This section examines the connection between cognitive load, as measured by ERP components such as N400 and P600, and language modeling metrics like surprisal and entropy.

Surprisal and Entropy Surprisal, proposed by Hale (2001), is defined as the negative log probability of a word given its context. This metric operationalizes the hypothesis that processing difficulty scales with unpredictability: words that deviate from contextual expectations impose greater cognitive demands. Empirical evidence supports this framework for predicting human reading times and has been shown to correlate with N400 amplitudes (Hagoort, 2003).

Entropy (Oh et al., 2024), on the other hand, measures the level of uncertainty about the upcoming linguistic input at a given point in a sentence. It is calculated based on the probability distribution of possible words in a context and has been linked to the complexity of language processing.

High surprisal and high entropy can co-occur at different stages of sentence processing (early and late stages), refecting both N400 and P600 responses. Researches suggest that surprisal models N400 (associated with semantic processing and semantic expectancy) (Michaelov et al., 2024), while entropy predicts P600 (linked to syntactic reanalysis in presence of syntactic anomalies or ambiguities) (Salicchi and Hsu, 2025). The two components operate both in parallel and sequentially, jointly shaping real-time language comprehension (Hagoort, 2003).

6 Conclusion & Discussion

Our work introduces **C**ognitively-**L**oad **A**ware **D**ynamic Activation (**CLADA**), a novel framework that unifies statistical sparsity and semantic adaptability for efficient LLM inference. The key findings are as follows:

Theoretical Insights We identify and formalize two complementary properties of LLMs' activation patterns:

1. **Statistical Sparsity**: Activation patterns are dominated by the prefix, providing a global efficiency baseline.
2. **Semantic Adaptability**: High-cognitive-load contexts (e.g., unanticipated terms, complex syntax) trigger dynamic adjustments in activation, preserving local precision.

Inference Gains CLADA combines offline-searched TT thresholds with cognitive-load-aware adjustments, achieving 18-25% speedup with less than 2% performance degradation across six LLMs and nine tasks.

6.1 Future Directions

To address these limitations and extend the impact of CLADA, we anticipate the following directions for future work:

1. Cross-Modal Sparsity: Extend CLADA to multimodal models (e.g., LLaVA) by incorporating visual cues into surprisal and entropy metrics for joint text-image processing;
2. Hardware Optimization: Design lightweight mask compression algorithms (e.g., sparse encoding) to reduce memory footprint and enable edge deployment.

6.2 Broader Impact

Beyond efficiency gains, CLADA offers a new perspective on bridging cognitive science and computational models. By formalizing the connection between cognitive mechanisms and LLM activation patterns, our work paves the way for more interpretable and biologically plausible AI systems. Furthermore, the training-free nature of CLADA enables broad applicability, including real-time dialogue and on-device processing.

Limitations

Despite its advantages, CLADA has several limitations that warrant further investigation:

Pre-filling Overhead For long prefixes (>2048 tokens), the pre-filling phase accounts for 10-15% of the total inference time, reducing the net speedup in real-time applications.

Hardware Constraints Storing activation masks for large models (e.g., LLaMA-70B) increases GPU memory usage, limiting deployment on resource-constrained devices.

References

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. PIQA: Reasoning about Physical Commonsense in Natural Language. *Preprint*, arXiv:1911.11641.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *Preprint*, arXiv:1803.05457.

Harry Dong, Beidi Chen, and Yuejie Chi. 2024. Prompt-prompted adaptive structured pruning for efficient LLM generation. In *First Conference on Language Modeling*.

Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. 2022. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. 2024. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21158–21166, Miami, Florida, USA. Association for Computational Linguistics.

Peter Hagoort. 2003. Interplay between syntax and semantics during sentence comprehension: Erp effects of combining syntactic and semantic violations. *Journal of Cognitive Neuroscience*, 15(6):883–899.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

John Hale. 2016. Information-theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3519–3529.

Gina R Kuperberg. 2007. Neural mechanisms of language comprehension: Challenges to syntax. *Brain Research*, 1146:23–49.

Gina R Kuperberg, Trevor Brothers, and Edward W Wlotko. 2020. A tale of two positivities and the N400: Distinct neural signatures are evoked by confirmed and violated predictions at different levels of representation. *Journal of Cognitive Neuroscience*, 32(1):12–35.

Marta Kutas and Kara D Federmeier. 2011. Thirty years and counting: finding meaning in the N400 component of the event-related brain potential (ERP). *Annual Review of Psychology*, 62(1):621–647.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Jiaxuan Li and Richard Futrell. 2024. Decomposition of surprisal: Unified computational model of erp components in language processing. *Preprint*, arXiv:2409.06803.

Ziang Liu, Genggeng Zhou, Jeff He, Tobia Marcucci, Li Fei-Fei, Jiajun Wu, and Yunzhu Li. 2023a. Model-based control with sparse neural dynamics. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023b. Deja vu: Contextual sparsity for efficient LLMs at inference time. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22137–22176. PMLR.

Chi Ma, Mincong Huang, Chao Wang, Yujie Wang, and Lei Yu. 2024. Dynamic Activation Pitfalls in LLaMA Models: An Empirical Study. *Preprint*, arXiv:2405.09274.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR.

James A Michaelov, Megan D Bardolph, Cyma K Van Petten, Benjamin K Bergen, and Seana Coulson. 2024. Strong prediction: Language model surprisal explains multiple N400 effects. *Neurobiology of Language*, 5(1):107–135.

Ramesh Nallapati, Bowen Zhou, Cicero Dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 280–290.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency Explains the Inverse Correlation of Large Language Models' Size, Training Data Amount, and Surprisal's Fit to Reading Times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2644–2663, St. Julian's, Malta. Association for Computational Linguistics.

Bowen Pan, Yikang Shen, Haokun Liu, Mayank Mishra, Gaoyuan Zhang, Aude Oliva, Colin Raffel, and Rameswar Panda. 2024. Dense training, sparse inference: Rethinking training of mixture-of-experts language models. *Preprint*, arXiv:2404.05567.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Lavinia Salicchi and Yu-Yin Hsu. 2025. Not Every Metric is Equal: Cognitive Models for Predicting N400 and P600 Components During Reading Comprehension. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3648–3654, Abu Dhabi, UAE. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. 2022. Scrolls: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. *Preprint*, arXiv:2306.11695.

Filip Szatkowski, Bartosz Wójcik, Mikoł aj Piórczyński, and Simone Scardapane. 2024. Exploiting activation sparsity with dense to dynamic-k mixture-of-experts conversion. In *Advances in Neural Information Processing Systems*, volume 37, pages 43245–43273. Curran Associates, Inc.

Ethan G Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P Levy. 2023. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, Yan Yan, Beidi Chen, Guangyu Sun, and Kurt Keutzer. 2024. LLM Inference Unveiled: Survey and Roofline Model Insights. *Preprint*, arXiv:2402.16363.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, Dublin, Ireland. Association for Computational Linguistics.

Haizhong Zheng, Xiaoyan Bai, Xueshen Liu, Zhuoqing Morley Mao, Beidi Chen, Fan Lai, and Atul Prakash. 2024. Learn to be efficient: Build structured sparsity in large language models. *Advances in Neural Information Processing Systems*, 37:101969–101991.

Zexuan Zhong, Mengzhou Xia, Danqi Chen, and Mike Lewis. 2024. Lory: Fully Differentiable Mixture-of-Experts for Autoregressive Language Model Pretraining. *Preprint*, arXiv:2405.03133.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15913–15923, Miami, Florida, USA. Association for Computational Linguistics.

A Detailed Background

A.1 Two Methods of Dynamic Activation

Training-Dependent Methods DejaVu (Liu et al., 2023b) predicts activation sparsity during training by leveraging ReLU's inherent sparsity. While achieving 20% speedup on OPT-style models, it fails on non-ReLU architectures like LLaMA's SwiGLU.

MoEfication (Zhang et al., 2022) dynamically routes inputs to expert subnets, but requires costly expert training and introduces routing overhead. DS-MoE (Pan et al., 2024) introduces a framework that employs dense computation during training and switches to sparse computation during inference. LLaMA-MoE (Zhu et al., 2024) offers a new lightweight method to transform FFNs into MoEs. LTE (Zheng et al., 2024) achieves a superior balance between sparsity and performance by activating fewer neurons and is applicable to models with both ReLU and non-ReLU activation functions.

Lory (Zhong et al., 2024) retains the autoregressive properties of language models by adopting a causally segmented routing strategy and a similarity-based data batching method. This enables efficient expert merging operations and promotes specialization among experts in processing similar documents during training sessions.

Training-Free Methods Griffin (Dong et al., 2024) uses sequence-level activation clustering (flocking) to skip redundant computations only in generation phase. Despite its simplicity, Griffin suffers from significant performance drops (>3% on QA tasks) due to heuristic threshold selection.

B Details on Bridging Cognitive Load and Activation

B.1 Why Token-2049?

Selecting token-2049 is imperative for the two reasons: 1) The CLADA method introduced in this paper facilitates inference speedup. Thus, it is crucial to examine the activation of the 2049_{th} or latter token when the prefix length is 2048. 2) While analyzing the 2049_{th} token, as opposed to the $(prefix + 1)_{th}$ token, may diminish the precision of observed surprisal and entropy effects, it more accurately adheres to the hypothesis of activation flocking.

B.2 Visual Evidence

We plot the whole sequence activation heat-map of NLS-A/B/A' separately in Figure 4, Figure 5, and Figure 6.

In these figures, the horizontal axis represents the neuron indices, while the vertical axis represents the token indices.

Figures 4 to 6 demonstrate that substituting the initial 512 tokens of NLS-A with the prefix from

DA Types	Definetions	Examples	Advantages	Current Limitations
Training-Dependent DA	Some leverage a *predictor*, which is pre-trained using the model's training data, to dynamically identify essential activation neurons or experts during the model's forward. (Figure 2)	DejaVu (Liu et al., 2023b), MoEfication(Zhang et al., 2022)	High Sparsity	Tend to underperform on models with non-ReLU activations(See Table 1)
	Others aim to reduce computational costs by employing multi-stage MoE-style training and introducing efficiency and separability loss penalties.	LTE (Zheng et al., 2024) and D2DMoE (Szatkowski et al., 2024)	High performance	Extra training required
Training-Free DA	Employs pre-searched or pre-defined thresholds or sparsity levels to decide which neurons to retain or discard. Neurons with activation values falling below this bar are eliminated during current forward, thereby reducing computational overhead and latency.(Figure 3)	Griffin (Dong et al., 2024), CLADA(Ours)	Training-free for all model archs	Lower performance

Table 7: Two types of DA methods

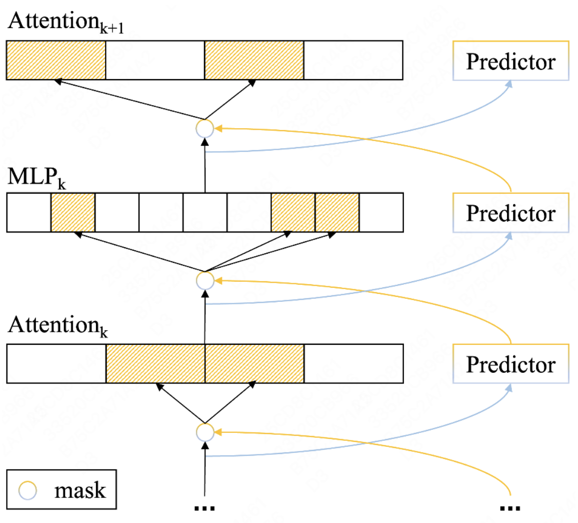

Figure 2: Training-Dependent DA

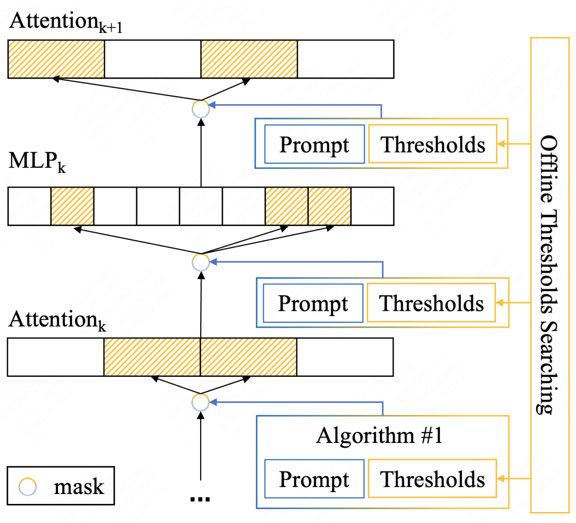

Figure 3: Training-Free DA

	Mean	Std	Min	Max
Prefix_len	-	-	512	1024
Surprisal	0.58	0.31	0.01	1.01
Entropy	0.42	0.45	0.01	1.01
Token_len	2048	0	2048	2048

Table 8: Descriptive statistics of linear regression variables. The *Surprisal* and *Entropy* values in the table are sequence-level normalized to eliminate the impact of inter-sequence variability and absolute magnitude differences.

NLS-B results in an activation pattern that more closely aligns with that of NLS-B.

B.3 Case Study

The mechanism behind statistical sparsity phenomenon needs detailed investigation. When processed as a sequence, activated neurons consider

	Dependent variable: Δ_{cos_sim}					
	(1)	(2)	(3)	(4)	(5)	(6)
	RTS-A'	NLS-A'	RTS-A'	NLS-A'	RTS-A'	NLS-A'
Prefix_len	4.22***	3.94***	4.08***	4.03***	3.71***	3.64***
	(22.12)	(20.41)	(21.35)	(21.15)	(19.46)	(18.53)
Surprisal			-0.01	-0.72***	-0.07	-0.80***
			(-0.26)	(-7.32)	(-0.12)	(-8.14)
Entropy					-0.35	-0.12***
					(-4.25)	(-5.85)
Token_len	10.63***	13.35***	10.65***	12.64***	11.75***	11.34***
	(5.04)	(5.07)	(5.02)	(5.05)	(5.06)	(5.03)
Obs	12000	12000	12000	12000	12000	12000
Adjusted R^2	0.12	0.12	0.15	0.16	0.17	0.16
Individual FE	YES	YES	YES	YES	YES	YES
Constant	YES	YES	YES	YES	YES	YES

Note: * $p<0.1$; ** $p<0.05$; *** $p<0.01$

Table 9: Robustness Test: Regression results of *prefix_len* on cosine activation similarity

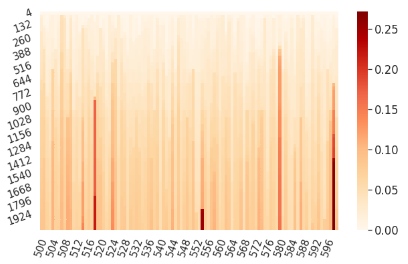

Figure 4: Activation of NLS-A

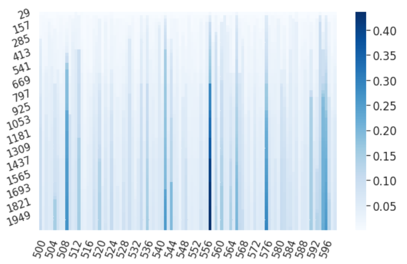

Figure 5: Activation of NLS-B

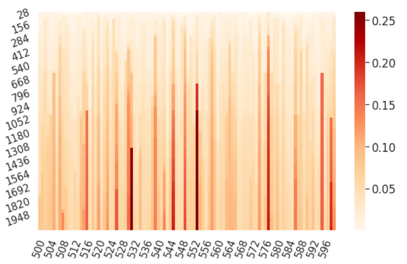

Figure 6: Activation of NLS-A'

all tokens, suggesting that statistical sparsity may be due to preceding tokens rather than the current token.

To eliminate sequence information influence, we selected specific samples and conducted a similarity analysis of their activation patterns. Details in Table 12 in Appendix B.3.

Table 12 details the 13 samples used for activation pattern similarity analysis. Samples 1-3 and Samples 4, 6, and 9 form two treatment groups. If Sample 4 shows greater similarity to Sample 1 than to Samples 2 and 3, it supports Hypothesis 2.1.

From the similarity heatmap in Figure 7, we observe the following: a) Samples 4, 6, and 9 are more similarly activated to Sample 1 than to Samples 2 and 3; b) Samples 1, 6, 8, and 9 are more similarly activated to Sample 4 than to Sample 5; c) Sample 9 is more similarly activated to Samples 4, 6, and 8; d) Samples 11 and 12 are more similarly activated to Samples 9 and 10; e) Samples 10, 11, and 12 are more similarly activated to Sample 13 than to any other samples.

Batch_size	**1**	**2**	**4**	**8**	**16**
Latency	64.03	74.00	76.36	79.33	82.85

Table 10: Batch size scalability analysis. Latency measures total processing time.

C Batch Size Analysis

We evaluate CLADA's scalability across different batch sizes using LLaMA-3-8B on XSum summarization. The experiments are conducted on a single NVIDIA A100 GPU with 80GB memory, using FP16 precision. We measure total time for processing the batch. Table 10 indicated that processing time increases (27.6% from batch 1 to 16) due to optimized memory access patterns in our dynamic activation strategy.

D Decision Metrics for Dynamic Activation

We evaluate alternative non-cognitive decision criteria, including activation magnitude and ℓ_2-norm-based thresholding, under identical settings to the main experiments. As shown in Table 11, these heuristics exhibit clear disadvantages in performance preservation or latency. CLADA consistently achieves the strongest retention of capability in comparable sparsity.

Decision Metric	**BoolQ Acc.**	**XSum R-1**	**Latency (s)**
Dense	80.76	29.62	81.31
CLADA (Full)	80.39	29.47	64.03
Only Entropy	77.37	27.36	63.82
Only Surprisal	78.73	27.82	62.84
Magnitude (Top-50%)	75.26	24.47	63.32
Activation Norm (Top-50%)	78.84	27.84	75.28

Table 11: Ablation of cognitive decision metrics on LLaMA-3-8B.

These static or weakly input-dependent criteria lead to notable degradation in either performance or efficiency at matched sparsity levels. In contrast, CLADA achieves more favorable trade-offs by leveraging surprisal and entropy as dynamic cognitive signals.

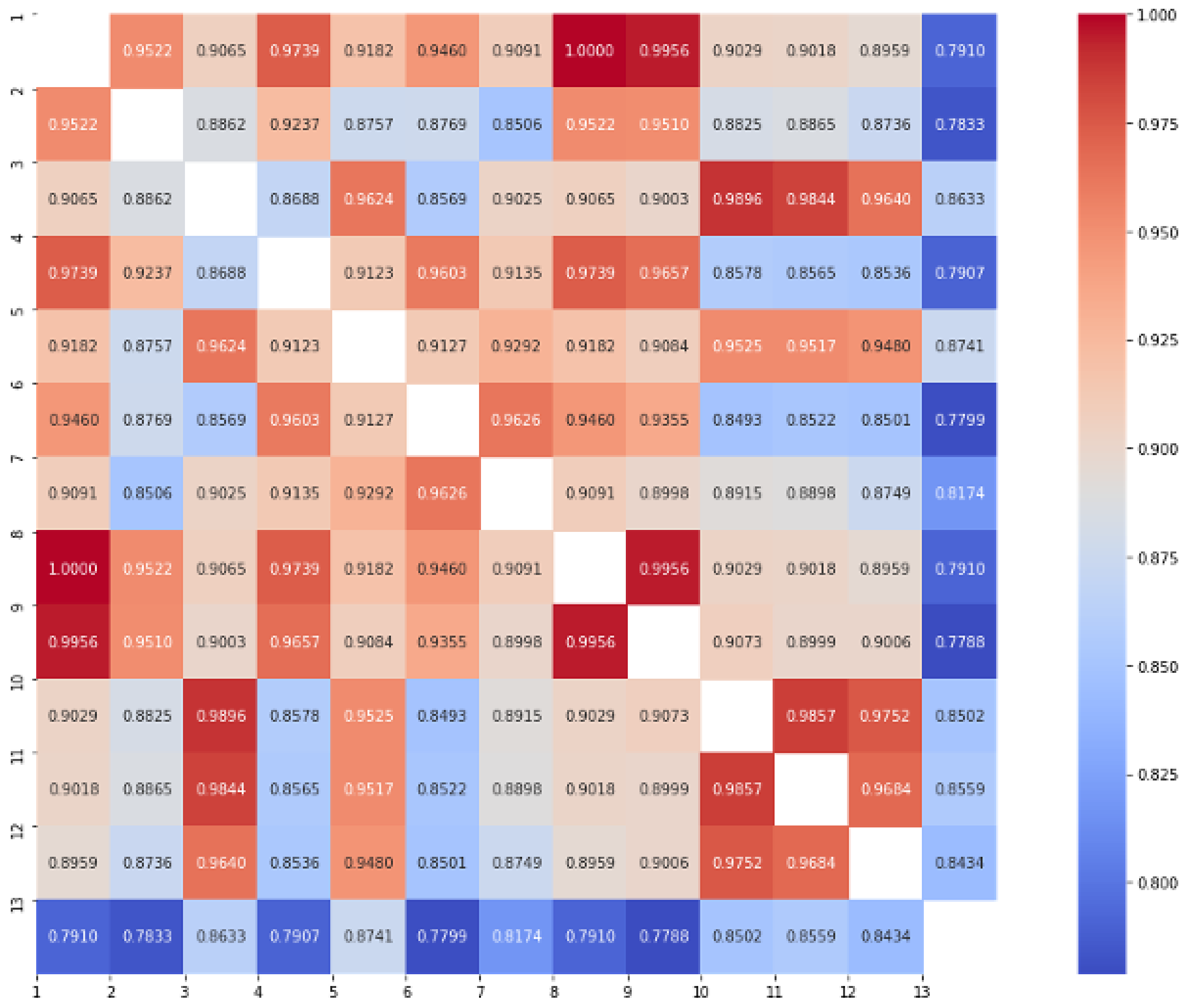

Figure 7: Similarity matrix of 13 samples' activation pattern

Index	Samples	Treatments
1	"### Article: Almost one million people visited the city"	Baseline
2	"Article: Almost one million people visited the city"	Remove beginning token
3	"Almost one million people visited the city"	Remove beginning tokens
4	"### Article: Nearly one million people visited the city"	Modify the word at the beginning of the sequence.
5	"Nearly one million people visited the city"	Remove beginning tokens
6	"### Article: Less than one million people visited the city"	Change to antonym
7	"Less than one million people visited the city"	Remove beginning tokens
8	"### Article: Almost one million people visited the city"	Similarity threshold
9	"### Article: Almost one million people visited the restaurant"	Change to synonyms
10	"Almost one million people visited the restaurant"	Modify the word at the end of the sequence
11	"Almost one million people visited the planet"	Modify the word at the end of the sequence
12	"Almost one million tourists visited the restaurant"	Modify the words at the middle and end
13	"Almost one million aliens visited the planet"	Dissimilarity threshold

Table 12: Detailed 13 samples for activation statistical sparsity check.