# Conditional Memory via Scalable Lookup: A New Axis of Sparsity for Large Language Models

**Xin Cheng** [1,2] **Wangding Zeng** [2] **Damai Dai** [2] **Qinyu Chen** [2]
**Bingxuan Wang** [2] **Zhenda Xie** [2] **Kezhao Huang** [2] **Xingkai Yu** [2] **Zhewen Hao** [2]
**Yukun Li** [2] **Han Zhang** [2] **Huishuai Zhang** [1] ✉ **Dongyan Zhao** [1,3] ✉ **Wenfeng Liang** [2]
[1] Peking University [2] DeepSeek-AI
[3] National Engineering Research Center of New Electronic Publishing Technologies
{zhanghuishuai, zhaody@pku.edu.cn}

## Abstract

Mixture-of-Experts (MoE) scales capacity via conditional computation, but Transformers lack a native knowledge lookup primitive. We introduce conditional memory, instantiated via Deep Sparse Embedding (DSE), which indexes a massive embedding table using local $N$-grams for $O(1)$ retrieval. We formalize sparsity allocation problem—how to split a fixed parameter budget between MoE experts and DSE memory—and find a U-shaped scaling law that identifies an optimal balance. Scaling to 27B parameters, DSE outperform an iso-parameter and iso-FLOPs MoE baseline across knowledge and reasoning benchmarks, and achieve markedly stronger long-context performance. Mechanistic analyses show that DSE offloads early-layer static recall into memory, freeing effective depth and attention for higher-level reasoning. DSE is also infrastructure-efficient: its deterministic hashing enables offloading massive parameters into host memory during inference with negligible throughput overhead.

## 1 Introduction

Sparsity has emerged as a fundamental design principle for Large Language Models (LLMs). Currently, this principle is primarily realized through Mixture-of-Experts (MoE) (Shazeer et al., 2017; Dai et al., 2024), which scales capacity via conditional computation. Owing to its ability to drastically increase model size without proportional increases in compute, MoE has become the de facto standard for foundation models (Liu et al., 2024; Comanici et al., 2025; Team et al., 2025).

Despite the success of conditional computation, the intrinsic heterogeneity of linguistic signals suggests significant room for structural optimization. Specifically, language modeling entails two qualitatively different sub-tasks: compositional reasoning and knowledge recall. Since standard Transformers lack a native knowledge lookup primitive, current models are forced to **simulate retrieval through computation**. This process essentially amounts to an expensive runtime reconstruction of a static lookup table, wasting valuable sequential depth on trivial operations that could otherwise be allocated to higher-level reasoning.

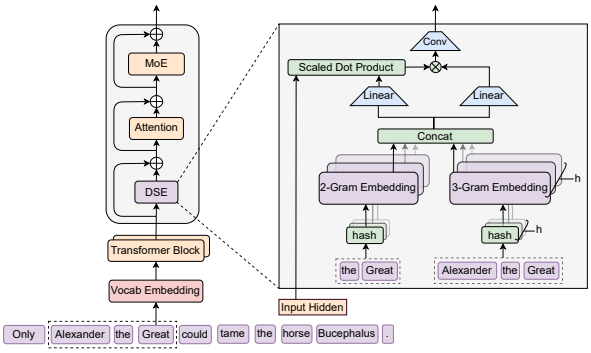


Figure 1: Overview of the Deep Sparse Embedding architecture. Left: DSE modules are selectively applied to specific layers, independent of the standard embedding. Right: Sparse embeddings are retrieved via hashed suffix $N$-grams and dynamically modulated by their similarity to the current hidden state.

To align model architecture with this linguistic duality, we advocate for a complementary axis of sparsity: conditional memory. Whereas conditional computation sparsely activates parameters to process dynamic logic (Shazeer et al., 2017), conditional memory relies on sparse lookup operations to retrieve static embeddings for fixed knowledge. As a preliminary exploration of this paradigm, we propose Deep Sparse Embedding (DSE). Inspired by prior work (Tito Svenstrup et al., 2017; Huang et al., 2025; Pagnoni et al., 2025), we utilize local $N$-grams—densified via vocabulary compression—to index a massive embedding table for constant-time $O(1)$ retrieval. To ensure these static priors align with the current context, we employ a gating mechanism where the hidden state acts as a query to dynamically filter the retrieved embeddings.

To quantify the synergy between these two prim-



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4968–4990

itives, we formulate the *Sparsity Allocation* problem: given a fixed total parameter budget, how should capacity be distributed between MoE experts and DSE memory? Our experiments uncover a distinct U-shaped scaling law, revealing that maximizing efficiency demands a precise balance of both mechanisms. Guided by this optimal allocation, we scale DSE to a 27B-parameter model. Compared to a strictly iso-parameter and iso-FLOPs MoE baseline, DSE-27B achieves superior efficiency across diverse domains. Crucially, the gains are not limited to knowledge-intensive tasks (e.g., MMLU: $+3.4$; MMLU-Pro: $+1.8$); we observe even more significant improvements in general reasoning (e.g., BBH: $+5.0$; ARC-Challenge: $+3.7$, DROP: $+3.3$) and code/math domains (e.g., HumanEval: $+3.0$; GSM8K: $+2.2$, MATH: $+2.4$). We also perform further mechanistic analysis and reveal that DSE relieves the backbone from reconstructing static knowledge in early layers, thereby increasing effective depth available for complex reasoning. Furthermore, DSE frees up attention capacity and can focus more on global context, so DSE-27B exhibits exceptional robustness in 32K-context scenarios.

Finally, we consider the infrastructure-aware efficiency. Unlike dynamic routing in MoE, DSE relies on deterministic hashing, which enables the runtime prefetch of embeddings and overlaps the communication cost with computation. We validate this on a vLLM-based engine: even with a naive implementation, offloading a 100B-parameter table entirely to host memory incurs negligible throughput overhead ($< 3\%$). This result confirms that DSE bypasses GPU memory limitations by efficiently utilizing system DRAM, allowing for a more aggressive expansion of model parameters.

## 2 Architecture

### 2.1 Overview

As shown in Figure 1, Deep Sparse Embedding (DSE) is a conditional memory module designed to augment the Transformer backbone by structurally separating static pattern storage from dynamic computation. Formally, given an input sequence $X$ and hidden states $H^{(\ell)} \in \mathbb{R}^{T \times d}$ at layer $\ell$, the module processes each position $t$ in two functional phases:

1. **Sparse Retrieval:** We extract and compress suffix $N$-grams to deterministically retrieve static embedding vectors via hashing.
2. **Context-aware Gating:** The retrieved embeddings are dynamically modulated by the current hidden state to filter noise, refined via a lightweight convolution, and merged into the residual stream.

### 2.2 Sparse Retrieval via Hashed $N$-grams

The first phase maps local contexts to static memory entries, involving compressing the tokenizer space and retrieving embeddings via deterministic hashing.

**Context Definition and Compression.** Standard subword tokenizers are constrained by lossless reconstruction, often assigning disjoint IDs to semantically equivalent terms due to surface variations (Kudo and Richardson, 2018; Li et al., 2023). To maximize representational density, we implement a vocabulary projection layer. Specifically, we pre-compute a surjective mapping $\mathcal{P} : V \to V'$ that collapses raw token IDs into canonical identifiers based on normalized textual equivalence (using NFKC, lowercasing, etc.). In practice, this is implemented as a fast lookup table that reduces the effective vocabulary size of a 128k tokenizer by 23% (see Appendix C). For a token at position $t$, we map its raw ID $x_t$ to a canonical ID $x'_t = \mathcal{P}(x_t)$, and define the local context as the suffix $N$-gram of these canonical IDs: $g_{t,n} = (x'_{t-n+1}, \ldots, x'_t)$.

**Multi-Head Hashing.** Directly parameterizing the combinatorial space of all possible $N$-grams is intractable. Following (Tito Svenstrup et al., 2017; Huang et al., 2025), we adopt a hashing-based approach. To mitigate collisions, we employ $K$ distinct hash heads for each $N$-gram order $n$. Each head $k$ maps the compressed context to an index within an embedding table $E_{n,k}$ (of prime size $M_{n,k}$) via a deterministic function $\phi_{n,k}$:

$$z_{t,n,k} \triangleq \phi_{n,k}(g_{t,n}), \quad \mathbf{e}_{t,n,k} = E_{n,k}[z_{t,n,k}]. \quad (1)$$

In practice, $\phi_{n,k}$ is implemented as a lightweight multiplicative-XOR hash. We construct the final memory vector $\mathbf{e}_t \in \mathbb{R}^{d_{\text{mem}}}$ by concatenating all retrieved embeddings:

$$\mathbf{e}_t \triangleq \Big\Vert_{n=2}^{N} \Big\Vert_{k=1}^{K} \mathbf{e}_{t,n,k}. \quad (2)$$

### 2.3 Context-aware Gating

Due to inherent polysemy and hash collisions, the retrieved embeddings may not align with the

specific semantics of the current context. To resolve this ambiguity, we employ a context-aware gating mechanism inspired by attention dynamics (Vaswani et al., 2017; Bahdanau et al., 2014). Specifically, we utilize the current hidden state $h_t$—which has aggregated global context via preceding attention layers—as a dynamic Query, while the retrieved memory $\mathbf{e}_t$ serves as the source for both Key and Value projections:

$$\mathbf{k}_t = W_K \mathbf{e}_t, \quad \mathbf{v}_t = W_V \mathbf{e}_t, \tag{3}$$

where $W_K, W_V$ are learnable projection matrices. To ensure gradient stability, we apply RMSNorm to the Query and Key before computing the scalar gate $\alpha_t \in (0, 1)$:

$$\alpha_t = \sigma \left( \frac{\text{RMSNorm}(h_t)^\top \cdot \text{RMSNorm}(\mathbf{k}_t)}{\sqrt{d}} \right). \tag{4}$$

The final output is $u_t = \alpha_t \cdot \mathbf{v}_t$. This design enforces semantic alignment: if the retrieved memory $\mathbf{e}_t$ contradicts the current context $h_t$, the gate $\alpha_t$ tends toward zero, effectively suppressing the noise. Finally, to expand the receptive field and enhance the model's non-linear expressivity, we introduce a short, depthwise causal convolution (Allen-Zhu, 2025; Peng et al., 2023a). Using a kernel size $W$, dilation $\delta$ (set to the max $N$-gram order) and SiLU activation (Elfwing et al., 2018), the output $y_t$ is:

$$y_t = \text{SiLU}\left(\text{Conv1D}(\text{RMSNorm}(u_t))\right) + u_t. \tag{5}$$

The DSE output is subsequently added to the backbone residual stream: $H^{(\ell)} \leftarrow H^{(\ell)} + y_t$, followed by the Attention and MoE layers.

### 2.4 Integration with Hyper-Connections

In this work, instead of standard residual connections (He et al., 2016), we adopt the more advanced Hyper-Connections (HC) architecture as our default backbone, chosen for its superior modeling capabilities (Zhu et al., 2025; Xie et al., 2025; Seed et al., 2025). A defining characteristic of HC is the expansion of the residual stream into $M$ parallel branches (typically $M = 4$), where information flow is modulated by learnable connection weights.

Although the DSE module is inherently topology-agnostic, adapting it to this multi-branch residual framework necessitates a structural optimization. To this end, we employ a single sparse embedding table shared across all $M$ branches, utilizing a shared Value projection matrix $W_V$ alongside $M$ distinct Key projection matrices $\{W_K^{(m)}\}_{m=1}^M$. For the $m$-th branch with hidden state $h_t^{(m)}$, the gating signal is computed as:

$$\alpha_t^{(m)} = \sigma \left( \frac{\text{RMSNorm}(h_t^{(m)})^\top \cdot \text{RMSNorm}(W_K^{(m)} \mathbf{e}_t)}{\sqrt{d}} \right) \tag{6}$$

The shared value vector $\mathbf{v}_t = W_V \mathbf{e}_t$ is then modulated by these branch-specific gates. This design allows the linear projections (one $W_V$ and $M$ distinct $W_K$) to be fused into a single dense matrix multiplication, maximizing the compute utilization of modern GPUs. Unless otherwise stated, all experiments utilize this integration with mHC ($M = 4$) (Xie et al., 2025).

### 2.5 Decoupling Compute and Memory

A key advantage of DSE is the deterministic nature of its retrieval. During training, embedding tables are sharded across GPUs, utilizing collective communication (All-to-All) to only fetch active rows. During inference, the memory indices depend solely on the input token IDs. This allows the system to: prefetch embeddings on the CPU host memory asynchronously and overlap the PCIe transfer of these embeddings with the GPU computation of preceding layers. This design effectively decouples model capacity from limited GPU HBM, enabling massive look-up tables with negligible latency overhead.

## 3 Scaling Laws and Sparsity Allocation

DSE introduces conditional memory (a large lookup table accessed sparsely), which is complementary to the conditional computation provided by MoE experts. This section investigates how to optimally allocate sparse capacity between these two primitives under a strictly compute-controlled budget. Two key questions drive our research:

1. **Allocation under fixed compute and size.** When total parameters and per-token computation are fixed (Iso-FLOPs per token), how should we split sparse capacity between MoE experts and DSE slots?
2. **Infinite Memory Regime.** If the DSE memory can be scaled aggressively, what scaling behavior does DSE exhibit by itself?

### 3.1 Optimal Allocation Ratio Between MoE and DSE

**Compute-matched formulation.** We use three parameter counts throughout this section:

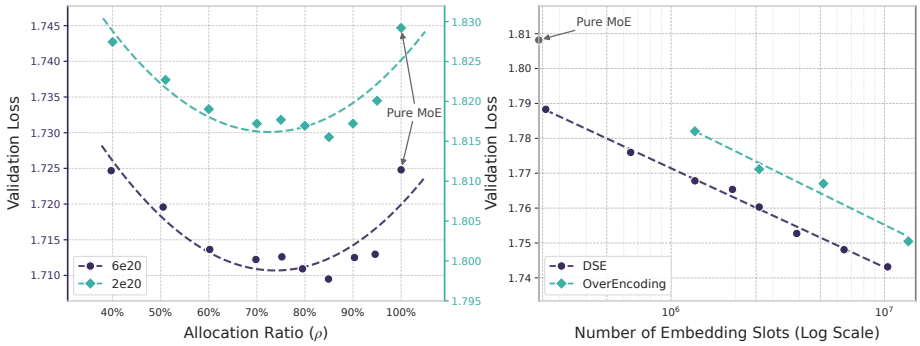


Figure 2: **Sparsity allocation and DSE scaling.** Left: Validation loss across allocation ratios $\rho$ with different compute budgets. Both curves exhibit a U-shape, with hybrid allocation surpassing Pure MoE ($\rho = 1$). Right: Scaling behavior in the infinite-memory regime. Validation loss is log-linear with respect to embedding slots.

- $P_{\text{tot}}$: total trainable parameters, excluding vocabulary embedding and LM head.
- $P_{\text{act}}$: activated parameters per token, i.e., parameters participating in the forward pass for a token. This quantity determines per-token FLOPs.
- $P_{\text{free}} \triangleq P_{\text{tot}} - P_{\text{act}}$: the remaining parameters that are not activated per token.

We keep $P_{\text{tot}}$ and $P_{\text{act}}$ fixed within each FLOPs budget, so that models have the same size and the same per-token FLOPs.[1]

**Allocation ratio.** We define the allocation ratio $\rho \in [0, 1]$ as the fraction of the inactive-parameter budget assigned to MoE expert capacity:

$$P_{\text{MoE}}^{(\text{free})} = \rho\, P_{\text{free}}, \quad P_{\text{DSE}} = (1 - \rho)\, P_{\text{free}}. \quad (7)$$

Intuitively:

- $\rho = 1$ corresponds to a pure MoE model (all inactive parameters are routed experts).
- $\rho < 1$ reduces the number of routed experts and reallocates the freed parameters to DSE embedding slots.

**Experimental protocol.** We evaluate this trade-off at two compute regimes. To ensure consistent sparsity, we maintain a constant ratio $P_{\text{tot}}/P_{\text{act}} \approx 10$ across both settings:

- $C = 2 \times 10^{20}$ FLOPs: $P_{\text{tot}} \approx 5.7$B and $P_{\text{act}} = 568$M. The baseline ($\rho = 1$) has a total of 106 experts.
- $C = 6 \times 10^{20}$ FLOPs: $P_{\text{tot}} \approx 9.9$B and $P_{\text{act}} = 993$M. The baseline ($\rho = 1$) has a total of 99 experts.

For each $\rho$, we construct the corresponding hybrid model by adjusting the number of routed experts and the number of DSE slots so that $P_{\text{tot}}$ and $P_{\text{act}}$ remain unchanged within that regime. All runs use the same training pipeline; we calculate optimization hyperparameters (e.g., learning rate and batch size) to ensure stable training in each regime.

**Results.** Figure 2 (left) reveals a consistent U-shaped relationship between validation loss and $\rho$ across both compute regimes. The pure MoE baseline ($\rho = 1$) is not optimal: reallocating roughly 20%–25% of the inactive parameters to DSE yields the best performance. Quantitatively, at $C = 6 \times 10^{20}$ (10B regime), loss improves from 1.7248 at $\rho = 1$ to 1.7109 near $\rho \approx 0.80$ ($\Delta = 0.0139$). The location of the optimum is stable across regimes ($\rho \approx 75\%$–$80\%$), suggesting that a fixed hybrid ratio can be a strong default when co-scaling MoE and DSE.

### 3.2 DSE under Infinite Memory Regime

In Section 3.1, we studied how to allocate a fixed inactive-parameter budget between MoE experts and DSE slots. We now examine the complementary setting where memory capacity can be scaled aggressively, motivated by two practical properties of DSE: (i) DSE retrieval uses static hash IDs, which enables standard systems techniques such as caching, off-device offloading, and prefetching; and (ii) retrieval cost is constant in table size—for each token, the model gathers a fixed number of embedding vectors, so scaling the table increases stored parameters but does not increase the com-

[1]For MoE, $P_{\text{act}}$ is determined by the top-$k$ selected experts, while the parameters of non-selected experts contribute to $P_{\text{free}}$. For DSE, only a constant number of slots are retrieved per token, so scaling the number of slots increases $P_{\text{tot}}$ without increasing per-token FLOPs.

putational cost (FLOPs) or the volume of data retrieved per token.

**Experimental protocol.** We start from a fixed MoE backbone with total parameters $P_{\text{tot}} \approx 3\text{B}$ and activated parameters $P_{\text{act}} = 568\text{M}$. We train the model for 100B tokens to ensure convergence. On top of this backbone, we add a DSE table with $M$ embedding slots and sweep $M$ from $2.58 \times 10^5$ to $1.0 \times 10^7$ (adding up to $\approx 13$ billion parameters). For comparison, we include the OverEncoding (Huang et al., 2025) as a reference.

**Results.** Figure 2 (right) shows that scaling the number of DSE slots yields a clear and consistent improvement in validation loss. Across the explored range, the curve is well-approximated by a line in $\log M$, i.e., validation loss decreases roughly log-linearly as memory grows. This behavior indicates that DSE provides a smooth and predictable scaling knob: larger memory continues to pay off without requiring additional computation. We also observe that while OverEncoding (Huang et al., 2025) benefits from larger $M$, DSE shows greater scaling potential.

## 4 Large Scale Pre-training

We scale DSE to the multi-billion parameter to validate its efficacy in large language model training. Specifically, we pre-train four models: (1) **Dense-4B** (4.1B total parameters), (2) **MoE-27B** (26.7B total parameters), (3) **DSE-27B** (26.7B total parameters), and (4) **DSE-40B** (39.5B total parameters). All models are trained on identical data curriculum (same token budget and order) and are strictly matched in the number of activated parameters.

### 4.1 Experimental Setup

**Training Data and Tokenization** All models are pre-trained on a corpus of 262 billion tokens sampled from our internal high-quality dataset. We utilize the tokenizer from DeepSeek-v3 (Liu et al., 2024) with a vocabulary size of 128k.

**Model Configurations** To ensure a controlled comparison, we adhere to a consistent default setting across all models unless explicitly stated otherwise. We utilize a 30-block Transformer with a hidden size of 2560. Each block integrates a Multi-head Latent Attention (MLA) module (DeepSeek-AI et al., 2024) with 32 heads, connected to FFNs via mHC (Xie et al., 2025) with an expansion rate of 4. All models are optimized using Muon (Jordan et al., 2024; Team et al., 2025); detailed hyperparameters are listed in the Appendix A.

**Evaluation Protocol** We evaluate models on a diverse suite of benchmarks spanning language modeling, knowledge and reasoning, reading comprehension, and code/math. For each benchmark, we follow standard few-shot prompting protocols. Details can be found in Appendix B.

### 4.2 Experimental Results

Consistent with prior literature, Table 1 demonstrates that sparse architectures significantly outperform iso-FLOPs dense baselines across all benchmarks, reaffirming the vital role of decoupling parameter count from computational cost for efficient scaling.

More importantly, DSE-27B consistently improves over the iso-parameter and iso-FLOPs MoE-27B baseline. Interestingly, these gains are not limited to knowledge-intensive tasks (e.g., MMLU: +3.0, CMMLU: +4.0). We observe even more significant improvements in general-reasoning domains (e.g., BBH: +5.0, ARC-Challenge: +3.7), as well as code and mathematical reasoning (e.g., HumanEval: +3.0, MATH: +2.4). To visualize training dynamics, we provide benchmark trajectories during pre-training in Appendix B. These results support our hypothesis that introducing a dedicated knowledge lookup primitive improves representation efficiency beyond what can be achieved by allocating the entire sparse budget to conditional computation.

Finally, scaling to DSE-40B further reduces pre-training loss and improves performance across most benchmarks. Although it does not yet strictly dominate DSE-27B on every task, this is likely an artifact of under-training as the loss gap between DSE-40B and the baselines continues to widen towards the end of training.

## 5 Long Context Training

To extend the context window to 32k tokens, we adopt the YaRN strategy (Peng et al., 2023b) following Liu et al. (2024), training for 5,000 steps on 30B high-quality tokens. We evaluate four configurations: the final MoE-27B and DSE-27B checkpoints, and two intermediate DSE-27B checkpoints. Notably, the DSE-27B (46k steps) and MoE-27B (50k steps) checkpoints share the same pre-training loss, establishing an "Iso-Loss" setting to isolate architectural impacts. Performance is assessed using

| Category | Benchmark (Metric) | # Shots | Dense-4B | MoE-27B | DSE-27B | DSE-40B |
|---|---|---|---|---|---|---|
| | # Total Params | | 3.8B | 26.7B | 26.7B | 39.5B |
| | # Activated Params | | 3.8B | 3.8B | 3.8B | 3.8B |
| | # Trained Tokens | | 262B | 262B | 262B | 262B |
| | # Expert (shared + routed, top-k) | | - | $2 + 72c6$ | $2 + 55c6$ | $2 + 55c6$ |
| | # DSE Parameters | | - | - | 5.7B | 18.5B |
| Language Modeling | Pile (loss) | - | 2.091 | 1.960 | **1.950** | 1.942 |
| | Validation Set (loss) | - | 1.768 | 1.634 | **1.622** | 1.610 |
| Knowledge & Reasoning | MMLU (Acc.) | 5-shot | 48.6 | 57.4 | **60.4** | 60.6 |
| | MMLU-Redux (Acc.) | 5-shot | 50.7 | 60.6 | **64.0** | 64.5 |
| | MMLU-Pro (Acc.) | 5-shot | 21.1 | 28.3 | **30.1** | 31.3 |
| | CMMLU (Acc.) | 5-shot | 47.9 | 57.9 | **61.9** | 63.4 |
| | C-Eval (Acc.) | 5-shot | 46.9 | 58.0 | **62.7** | 63.3 |
| | AGIEval (Acc.) | 0-shot | 29.1 | 38.6 | **41.8** | 45.9 |
| | ARC-Easy (Acc.) | 25-shot | 76.8 | 86.5 | **89.0** | 90.1 |
| | ARC-Challenge (Acc.) | 25-shot | 59.3 | 70.1 | **73.8** | 76.4 |
| | TriviaQA (EM) | 5-shot | 33.0 | 48.8 | **50.7** | 51.8 |
| | TriviaQA-ZH (EM) | 5-shot | 62.8 | 74.8 | **76.3** | 77.9 |
| | PopQA (EM) | 15-shot | 15.1 | 19.2 | **19.4** | 21.2 |
| | CCPM (Acc.) | 0-shot | 72.2 | 79.6 | **87.1** | 87.7 |
| | BBH (EM) | 3-shot | 42.8 | 50.9 | **55.9** | 57.5 |
| | HellaSwag (Acc.) | 0-shot | 64.3 | 71.8 | **72.7** | 73.1 |
| | PIQA (Acc.) | 0-shot | 63.8 | 71.9 | **73.5** | 76.5 |
| | WinoGrande (Acc.) | 5-shot | 64.0 | 67.6 | **67.8** | 68.1 |
| Reading Comprehension | DROP (F1) | 1-shot | 41.6 | 55.7 | **59.0** | 60.7 |
| | RACE-Middle (Acc.) | 5-shot | 72.4 | 80.9 | **82.8** | 83.3 |
| | RACE-High (Acc.) | 5-shot | 66.0 | 75.4 | **78.2** | 79.2 |
| | C3 (Acc.) | 0-shot | 57.7 | 60.1 | **63.6** | 61.8 |
| Code & Math | HumanEval (Pass@1) | 0-shot | 26.8 | 37.8 | **40.8** | 38.4 |
| | MBPP (Pass@1) | 3-shot | 35.4 | 46.6 | **48.2** | 46.2 |
| | CruxEval-i (EM) | 0-shot | 27.6 | 30.7 | **32.2** | 36.2 |
| | CruxEval-o (EM) | 0-shot | 28.7 | 34.1 | **35.0** | 35.3 |
| | GSM8K (EM) | 8-shot | 35.5 | 58.4 | **60.6** | 62.6 |
| | MGSM (EM) | 8-shot | 27.0 | 46.8 | **49.4** | 52.4 |
| | MATH (EM) | 4-shot | 15.2 | 28.3 | **30.7** | 30.6 |

Table 1: Performance of Dense, MoE, and DSE architectures. All models maintain identical activated parameters and training tokens. DSE-27B significantly outperforms the iso-FLOPs MoE-27B across all benchmarks.

LongPPL (Fang et al.) across four data categories and the RULER benchmark (Hsieh et al.) covering 14 retrieval and reasoning tasks. Please refer to Appendix J for details.

### 5.1 Experimental Results

Table 2 highlights the interplay between base modeling capability and architectural efficiency.

**Dependence on Base Capability.** First, we observe that long-context performance is not solely determined by model architecture but scales strictly with pre-training compute. The monotonic improvement across DSE variants (41k $\rightarrow$ 50k steps) confirms that a stronger foundation model is a prerequisite for effective long-context reasoning.

**Architectural Efficiency.** Crucially, DSE demonstrates superior efficiency even when accounting for the dependency mentioned above. Remarkably, the under-trained DSE-27B (41k)—utilizing only 82% of the compute—already rivals the fully converged MoE-27B baseline. When controlled for base capability (the "Iso-Loss" setting), DSE significantly outperforms MoE on complex reasoning tasks, such as Multi-Query NIAH (97.0 vs. 84.2) and Variable Tracking (87.2 vs. 77.0). This specific gain in retrieval-heavy tasks supports our hypothesis: by offloading local patterns to DSE lookups, the model effectively preserves attention capacity for processing global dependencies.

## 6 Analysis

In this section, we delve into the internal mechanisms of DSE, examining its effective depth and ablating core design choices. And we mainly want to answer one question: **Is DSE functionally equivalent to increasing the model's depth?** We also

| Model | LongPPL (32k) | | | | RULER (32k) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Perplexity (↓) | | | | NIAH Accuracy (↑) | | | | Other Tasks (↑) | | | |
| | Book | Paper | Code | L-CoT | S | MK | MV | MQ | VT | CWE | FWE | QA |
| MoE-27B *(50k, 1.63)* | 4.38 | 2.91 | 2.49 | 14.16 | **100.0** | 88.0 | 92.7 | 84.2 | 77.0 | 4.5 | 73.0 | 34.5 |
| DSE-27B *(41k, 1.66)* | 4.37 | 2.92 | 2.50 | 14.26 | 99.6 | 88.3 | 93.0 | 89.5 | 83.2 | 3.8 | 99.6 | **44.0** |
| DSE-27B *(46k, 1.63)* | 4.19 | 2.84 | 2.45 | 13.59 | 97.6 | 89.0 | 95.5 | **97.0** | 87.2 | 4.3 | 98.6 | 37.5 |
| DSE-27B *(50k, 1.62)* | **4.14** | **2.82** | **2.44** | **13.41** | 99.3 | **89.3** | **96.5** | **97.0** | **89.0** | **5.9** | **99.3** | 40.5 |

Table 2: Long-context performance comparison. Parenthetical values denote the pre-training steps and the corresponding loss prior to the long-context extension. Two key findings: (1) With only 82% of the pre-training FLOPs (41k vs. 50k), DSE-27B matches the baseline's LongPPL performance while achieving significantly higher accuracy on RULER; (2) Under both iso-pretraining-loss (46k) and iso-pretraining-FLOPs (50k) settings, DSE-27B substantially outperforms the baseline across all metrics. **Bold** indicates the best and underline the second.

conduct a sensitivity analysis in Appendix G, a report inference throughput in Appendix H and case study in Appendix I. Current LLMs lack a dedicated knowledge lookup primitive analogous to the `SELECT` operations in relational databases; consequently, they rely on computation to simulate memory recall. As illustrated in Table 5 (reproduced from Ghandeharioun et al. (2024)), to recognize the entity `"Diana, Princess of Wales"`, Vicuna-13B (Chiang et al., 2023) must compute through multiple layers of attention and FFNs to progressively aggregate information from context tokens. This process resolves an entity that could theoretically be identified via a simple knowledge lookup operation.

Given this, we posit that by equipping the model with an explicit knowledge lookup capability, DSE effectively mimics an increase in model depth by relieving the model of the early stages of feature composition. To validate this hypothesis, we employ two mechanistic interpretability tools: LogitLens (nostalgebraist, 2020; Belrose et al., 2023) and Centered Kernel Alignment analysis (CKA) (Kornblith et al., 2019; Davari et al., 2022).

**Accelerated Prediction Convergence** We first analyze the evolution of predictions across layers using LogitLens. By projecting each intermediate layer's hidden state into the final LM Head, we compute the Kullback–Leibler divergence between the intermediate output distribution and the model's final output distribution. This metric quantifies how close a latent representation is to being "prediction-ready" (Csordás et al., 2025; Belrose et al., 2023). We evaluate this on the validation set.

Figure 3(a) reports the layer-wise KL divergence. Compared to the MoE baseline, both DSE variants exhibit systematically smaller KL divergence, with the most pronounced gap appearing in the early blocks. The steeper descent in the DSE curves indicates that the model resolves ambiguity much faster. This observation aligns with our hypothesis: by accessing external knowledge explicitly, DSE reduces the computational steps required for factual recall, thereby reaching high-confidence, valid predictions earlier in the network hierarchy.

**Representational Alignment** To further investigate whether DSE layers semantically correspond to deeper layers of the baseline, we employ Centered Kernel Alignment (CKA), a widely established metric for comparing representational structures. Given two sets of representations $X$ and $Y$ (e.g., activations from different models or layers), CKA is defined as:

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\mathrm{HSIC}(L, L)}} \tag{8}$$

where $K = XX^\top$ and $L = YY^\top$ denote the Gram matrices (using a linear kernel) and HSIC is Hilbert-Schmidt Independence Criterion (Gretton et al., 2005). We employ a minibatch implementation with an unbiased estimator of HSIC (Davari et al., 2022) and evaluate on the Few-NERD dataset (Ding et al., 2021), extracting hidden states corresponding to the final token of named entities.

To rigorously quantify the layer-wise correspondence, we first compute the pairwise CKA similarity matrix $S \in [0, 1]^{L_{\text{MoE}} \times L_{\text{DSE}}}$. We then introduce a soft alignment index $a_j$, defined as the weighted centroid of the top-$k$ most similar MoE layers for each DSE layer $j$:

$$a_j = \frac{\sum_{i \in \mathcal{I}_j} S_{i,j} \cdot i}{\sum_{i \in \mathcal{I}_j} S_{i,j}}, \quad \text{where } \mathcal{I}_j = \underset{i}{\mathrm{argtop}k}(S_{i,j}). \tag{9}$$

Here, $S_{i,j}$ denotes the similarity score between MoE layer $i$ and DSE layer $j$. The index $a_j$ serves

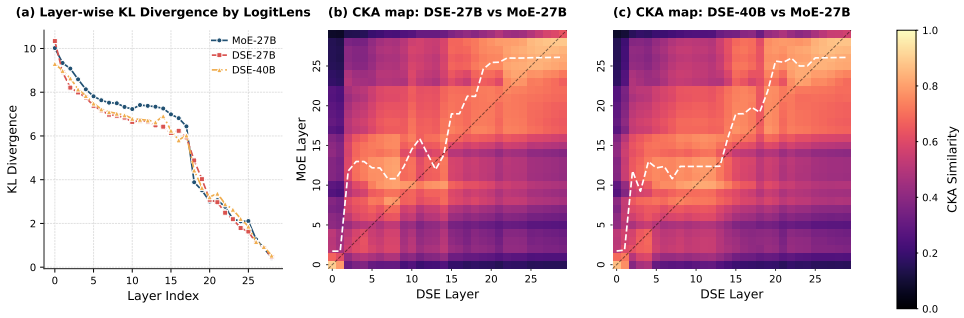


Figure 3: Layer-wise representational analysis of MoE and DSE models. (a) Layer-wise KL Divergence computed by LogitLens (nostalgebraist, 2020). The DSE models consistently exhibit lower KL divergence compared to the MoE baseline, especially in the early layers. (b-c) Similarity heatmap computed by CKA (Kornblith et al., 2019). Notably, the high-similarity regions deviate from the diagonal, showing a distinct upward shift.

as a robust proxy for the "effective MoE depth" corresponding to DSE layer $j$, utilizing top-$k$ filtering (with $k = 5$) to mitigate low-similarity noise.

Figures 3 (b)–(c) visualize the similarity heatmaps overlayed with the soft alignment curve (dashed white line). We observe a distinct upward shift from the diagonal, meaning that $a_j > j$ for a wide range of layers. For instance, the representations formed at layer 5 of DSE-27B align most closely with those at approximately layer 12 of the MoE baseline.

This consistent off-diagonal structure indicates that DSE effectively compresses the representational progression of the baseline. Combined with the LogitLens findings, these results substantiate our central claim: DSE is functionally equivalent to increasing effective depth. The explicit lookup mechanism shifts the model into later-stage representations earlier, relieving the early transformer blocks from the burden of "rediscovering" knowledge from scratch.

## 7 Related Work

**Mixture-of-Experts.** MoE architectures decouple model capacity from computational cost by conditionally activating a sparse subset of experts per token (Shazeer et al., 2017). Subsequent innovations (Lepikhin et al., 2020; Lewis et al., 2021; Fedus et al., 2022; Du et al., 2022) enabled super-linear parameter scaling while maintaining constant inference costs. More recently, DeepSeek-MoE (Dai et al., 2024) demonstrated superior efficiency. Adopting this architecture, recent models (Liu et al., 2024; Team et al., 2025) have further pushed total parameters to an extremely large scale.

**Embedding Scaling.** Recent studies explore scaling the embedding layer as a compute-efficient alternative to deepening the network architecture, capitalizing on the low computational cost of lookup operations. Notable examples include Per-Layer Embeddings (Team, 2025) and DeepEmbed (RWKV Team, 2025). Furthermore, approaches like OTT (Huang et al., 2025) and BLT (Pagnoni et al., 2025) demonstrate substantial improvements in language modeling by scaling the input vocabulary with hash embeddings (Tito Svenstrup et al., 2017). Similarly, SCONE (Yu et al., 2025) extends model capacity by employing an auxiliary model to pre-compute billions of frequent n-gram embeddings which are offloaded to off-chip storage during inference, enabling massive parameter scaling with negligible overhead.

## 8 Conclusion

We propose Deep Sparse Embedding (DSE), a conditional memory module that retrieves hashed N-gram embeddings with constant-time lookup. Sparsity allocation reveals a U-shaped trade-off: shifting parameters from experts to DSE improves efficiency. A 27B DSE model outperforms matched MoE baselines across knowledge, reasoning, code, math, and long-context tasks. Also, its deterministic hashing enables deployment-friendly host offloading.

## Limitations

While Deep Sparse Embedding (DSE) improves parameter efficiency by introducing a hashed N-gram lookup primitive, it also comes with several limitations.

(1) Our experiments are conducted with a specific backbone design and optimization recipe. The

optimal sparsity allocation ratio and the magnitude of gains may vary with different expert-routing strategies, attention variants, depth/width scaling, training data mixtures, or token budgets;

(2) We also primarily study pre-training and long-context extension; results in post-training scenario (SFT and RL) remain underexplored.

## Acknowlegement

This work is supported in part by the National Natural Science Foundation of China (62576016).

## References

Zeyuan Allen-Zhu. 2025. Physics of language models: Part 4.1, architecture design and the magic of canon layers. arXiv preprint arXiv:2512.17351.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. arXiv preprint arXiv:2108.07732.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

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.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 7432–7439.

Yuen Ren Chao and George Kingsley Zipf. 1950. Human behavior and the principle of least effort: An introduction to human ecology. Language, 26:394.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. Preprint, arXiv:2107.03374.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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. arXiv preprint arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261.

Róbert Csordás, Christopher D Manning, and Christopher Potts. 2025. Do language models use their depth efficiently? arXiv preprint arXiv:2505.13898.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, and 1 others. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. arXiv preprint arXiv:2401.06066.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. 2022. Reliability of cka as a similarity measure in deep learning. arXiv preprint arXiv:2210.16156.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. Preprint, arXiv:2405.04434.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3198–3213.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, and 1

others. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In International conference on machine learning, pages 5547–5569. PMLR.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. arXiv preprint arXiv:1903.00161.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks, 107:3–11.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In The Thirteenth International Conference on Learning Representations.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120):1–39.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, and 1 others. 2025. Are we done with mmlu? In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5069–5096.

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, pages 15466–15490. PMLR.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. 2005. Measuring statistical dependence with hilbert-schmidt norms. In International conference on algorithmic learning theory, pages 63–77. Springer.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. 2024. Cruxeval: A benchmark for code reasoning, understanding and execution. arXiv preprint arXiv:2401.03065.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In First Conference on Language Modeling.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. 2025. Overtokenized transformer: Vocabulary is generally worth scaling. arXiv preprint arXiv:2501.16975.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, and 1 others. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. Advances in Neural Information Processing Systems, 36:62991–63010.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. Muon: An optimizer for hidden layers in neural networks.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

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

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. Preprint, arXiv:1808.06226.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the 29th symposium on operating systems principles, pages 611–626.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 6265–6274. PMLR.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. In Findings of the Association for Computational Linguistics: ACL 2024, pages 11260–11285.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, and 48 others. 2023. Starcoder: may the source be with you! Preprint, arXiv:2305.06161.

Wenhao Li, Fanchao Qi, Maosong Sun, Xiaoyuan Yi, and Jiarui Zhang. 2021. Ccpm: A chinese classical poetry matching dataset. arXiv preprint arXiv:2106.01979.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9802–9822.

nostalgebraist. 2020. interpreting gpt: the logit lens. LessWrong.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, and 1 others. 2025. Byte latent transformer: Patches scale better than tokens. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9238–9258.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, and 1 others. 2023a. Rwkv: Reinventing rnns for the transformer era. arXiv preprint arXiv:2305.13048.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023b. Yarn: Efficient context window extension of large language models. arXiv preprint arXiv:2309.00071.

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. Psychonomic bulletin & review, 21(5):1112–1130.

RWKV Team. 2025. Rwkv architecture history. https://wiki.rwkv.com/basic/architecture.html. Section "RWKV-V8's DeepEmbed", accessed 2025-12-09.

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

Seed, Baisheng Li, Banggu Wu, Bole Ma, Bowen Xiao, Chaoyi Zhang, Cheng Li, Chengyi Wang, Chengyin Xu, Chi Zhang, Chong Hu, Daoguang Zan, Defa Zhu, Dongyu Xu, Du Li, Faming Wu, Fan Xia, Ge Zhang, Guang Shi, and 100 others. 2025. Virtual width networks. Preprint, arXiv:2511.11238.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. arXiv preprint arXiv:2210.03057.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging chinese machine reading comprehension. Transactions of the Association for Computational Linguistics, 8:141–155.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In Findings of the Association for Computational Linguistics: ACL 2023, pages 13003–13051.

Gemma Team. 2025. Gemma 3n.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. arXiv preprint arXiv:2507.20534.

Dan Tito Svenstrup, Jonas Hansen, and Ole Winther. 2017. Hash embeddings for efficient word representations. Advances in neural information processing systems, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. Advances in Neural Information Processing Systems, 37:95266–95290.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. 2025. mhc: Manifold-constrained hyper-connections. Preprint, arXiv:2512.24880.

Da Yu, Edith Cohen, Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Daogao Liu, and Chiyuan Zhang. 2025. Scaling embedding layers in language models. arXiv preprint arXiv:2502.01637.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 2299–2314.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. 2025. Hyper-connections. Preprint, arXiv:2409.19606.

# Appendices

## A Detailed Model Architecture and Hyper Parameters

- **Dense-4B** serves as the dense baseline. It utilizes the backbone architecture described above, incorporating a standard dense FFN into every block.

- **MoE-27B** replaces the standard dense FFN with a DeepSeekMoE module (Dai et al., 2024). Configured with 72 routed experts and 2 shared experts (activating the top-$k = 6$ routed experts per token), this model scales to 26.7B total parameters while maintaining the same activated parameters as Dense-4B.

- **DSE-27B** adopts the overall design of MoE-27B but reduces the number of routed experts from 72 to 55 and assigns the freed parameter budget to a 5.7B-parameter DSE module, keeping the total model size constant at 26.7B. For the DSE module configuration, we set the maximum N-gram size to 3, the number of heads to 8, and the dimension to 1280. Regarding optimization, the DSE embedding parameters utilize a learning rate scaled by $5\times$ relative to the global learning rate and are optimized using Adam (Kingma, 2014) with zero weight decay. Additionally, the DSE convolution parameters are initialized to zero to ensure an initial identity mapping.

- **DSE-40B** retains the same setting as DSE-27B, but scales sparse embedding module to 18.5B parameters (totaling 39.5B parameters) to investigate scaling behavior.

| | Dense-4B | MoE-27B | DSE-27B | DSE-40B |
|---|---|---|---|---|
| Total Params | 4.1B | 26.7B | 26.7B | 39.5B |
| Active Params | | 3.8B | | |
| Total Tokens | | 262B | | |
| Layers | | 30 | | |
| Dimension | | 2560 | | |
| Leading Dense Layers | - | 1 | 1 | 1 |
| Routed Experts | - | 72 | 55 | 55 |
| Active Experts | - | 6 | 6 | 6 |
| Shared Experts | - | 2 | 2 | 2 |
| Attention module | | MLA (DeepSeek-AI et al., 2024) | | |
| RoPE $\theta$ | | 10000 | | |
| mHC Expansion Rate | | 4 | | |
| Sequence Length | | 4096 | | |
| Vocab Size | | 129280 | | |
| Batch Size | | 1280 | | |
| Training Steps | | 50000 | | |
| Optimizer | | Muon (Jordan et al., 2024) | | |
| Base Learning Rate | | 4e-4 | | |
| Lr Scheduler | | Step Decay (Bi et al., 2024) | | |
| Weight Decay | | 0.1 | | |
| DSE Dim | - | - | 1280 | 1280 |
| DSE Vocab Size | - | - | 2262400 | 7239680 |
| DSE Num Head | - | - | 8 | 8 |
| DSE Layer | - | - | [2,15] | [2,15] |
| DSE N-gram | - | - | [2,3] | [2,3] |
| DSE combine mHC | - | - | True | True |
| DSE tokenizer compression | - | - | True | True |
| DSE Conv Zero Init | - | - | True | True |
| DSE Lr Multipler | - | - | x5 | x5 |
| DSE Weight Decay | - | - | 0.0 | 0.0 |
| DSE Optimizer (Embed. only) | - | - | Adam (Kingma, 2014) | |

Table 3: Detailed model architecture information and training hyper parameters.

## B Full Benchmark Curves

- **Language Modeling:** We report loss on the test set of The Pile (Gao et al., 2020) and an internal validation set drawn from the same distribution as the training data.

- **Knowledge&Reasoning:** MMLU (Hendrycks et al., 2020), MMLU-Redux (Gema et al., 2025), MMLU-Pro (Wang et al., 2024), CMMLU (Li et al., 2024), C-Eval (Huang et al., 2023), AGIEval (Zhong et al., 2024), ARC-Easy/Challenge (Clark et al., 2018), TriviaQA (Joshi et al., 2017), TriviaQA-ZH (internal), PopQA (Mallen et al., 2023), CCPM (Li et al., 2021), BBH (Suzgun et al., 2023), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and WinoGrande (Sakaguchi et al., 2021).
- **Reading Comprehension:** DROP (Dua et al., 2019), RACE (Middle/High) (Lai et al., 2017), and C3 (Sun et al., 2020).
- **Code & Math:** HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), Crux-Eval (Gu et al., 2024), GSM8K (Cobbe et al., 2021), MGSM (Shi et al., 2022), and MATH (Hendrycks et al., 2021).

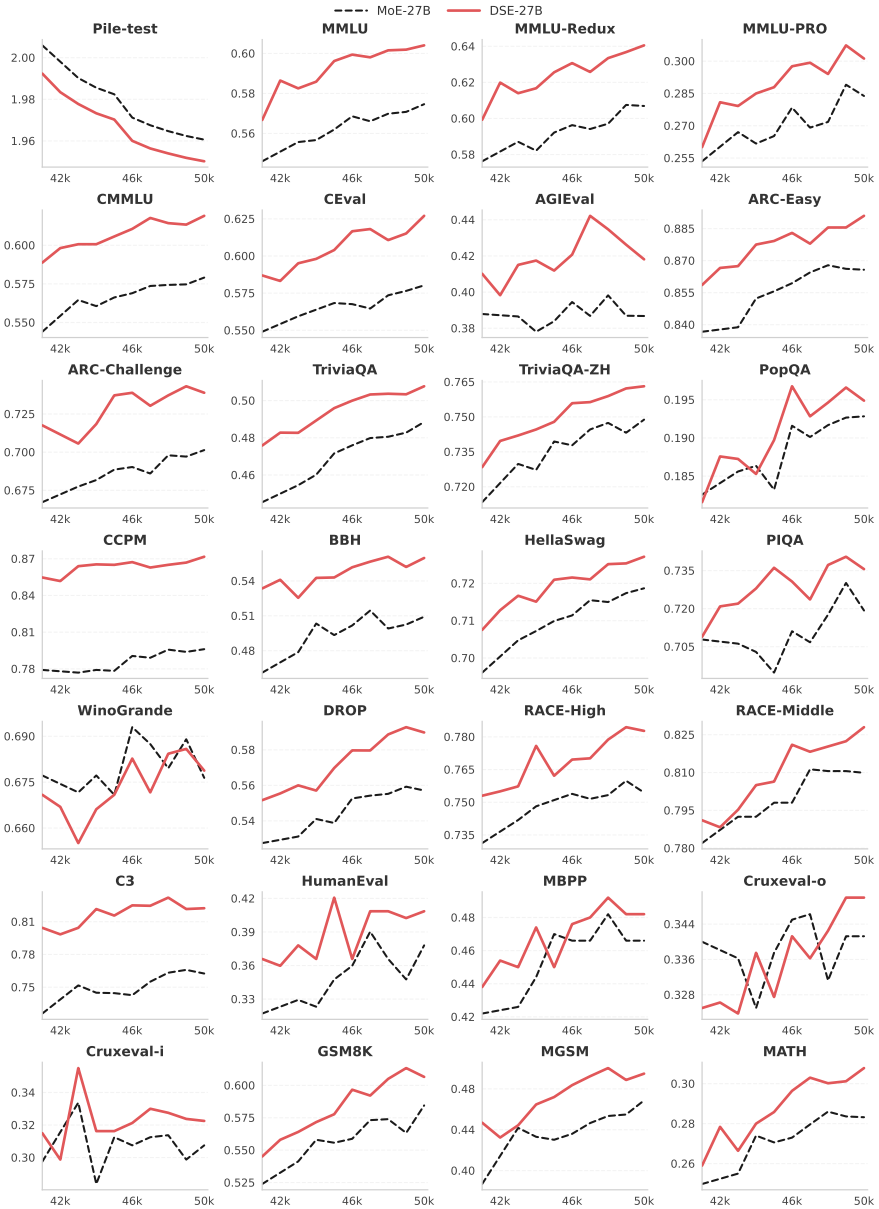


Figure 4: Benchmark curves comparing MoE-27B and DSE-27B over the last 10k training steps.

## C Case Study of Tokenizer Compression

| Rank | Merge Count | Normalized Token | Original Tokens |
|---|---|---|---|
| 1 | 163 | ’␣’ | ’\t’, ’\n’, ’\r’, ’␣’, ’␣␣’, ’\n\n’, ’␣␣␣’, ’␣\n’, ... |
| 2 | 54 | ’a’ | ’A’, ’a’, ’␣a’, ’␣A’, ’á’, ’ä’, ’ã’, ’ą’, ... |
| 3 | 40 | ’o’ | ’O’, ’o’, ’␣o’, ’ó’, ’ö’, ’ô’, ’õ’, ’ő’, ... |
| 4 | 35 | ’e’ | ’E’, ’e’, ’␣e’, ’é’, ’è’, ’ę’, ’ě’, ’ê’, ... |
| 5 | 30 | ’i’ | ’I’, ’i’, ’␣I’, ’í’, ’ì’, ’î’, ’ī’, ’ï’, ... |

Table 4: Top-5 merged tokens by *Tokenizer Compression*. Overall compression ratio: 76.57% for our 128k tokenizer.

## D Case Study of Gating Mechanism

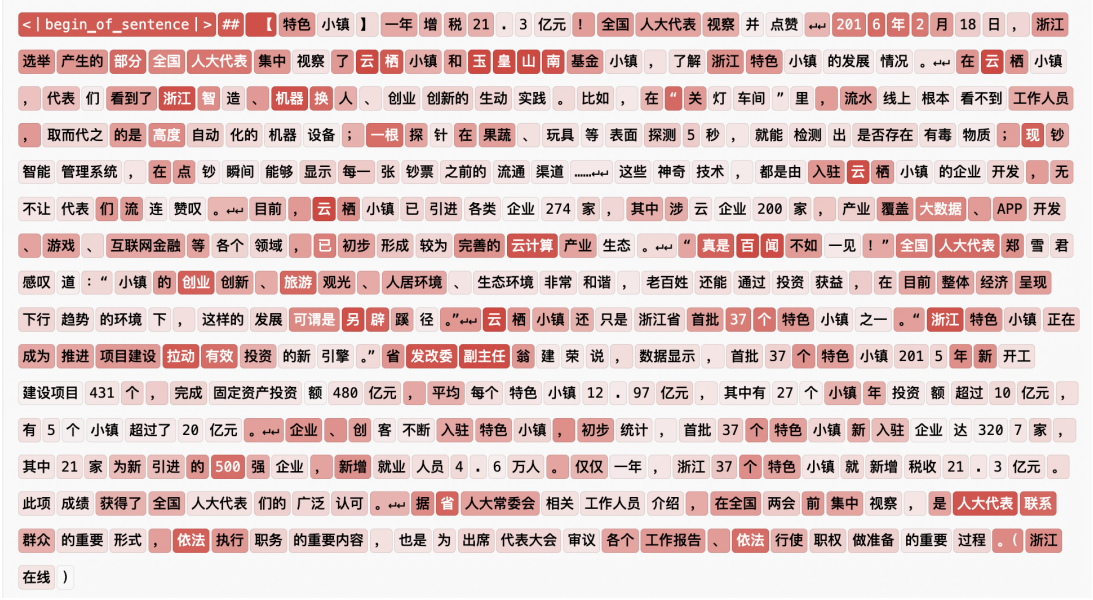


Figure 5: Visualization of DSE gate activations of a Chinese news report.

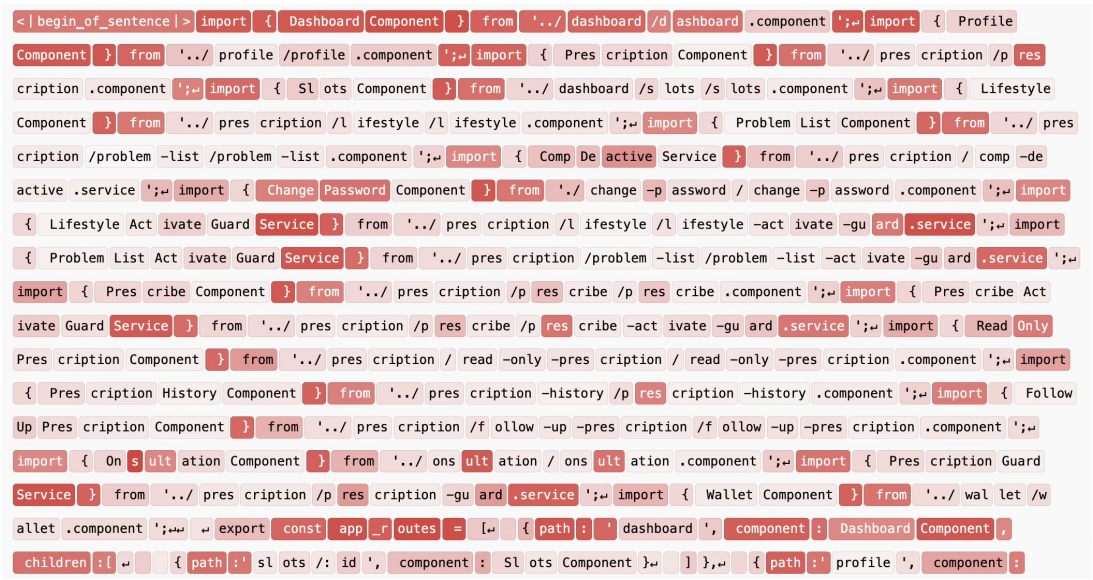


Figure 6: Visualization of DSE gate activations on a representative code snippet.

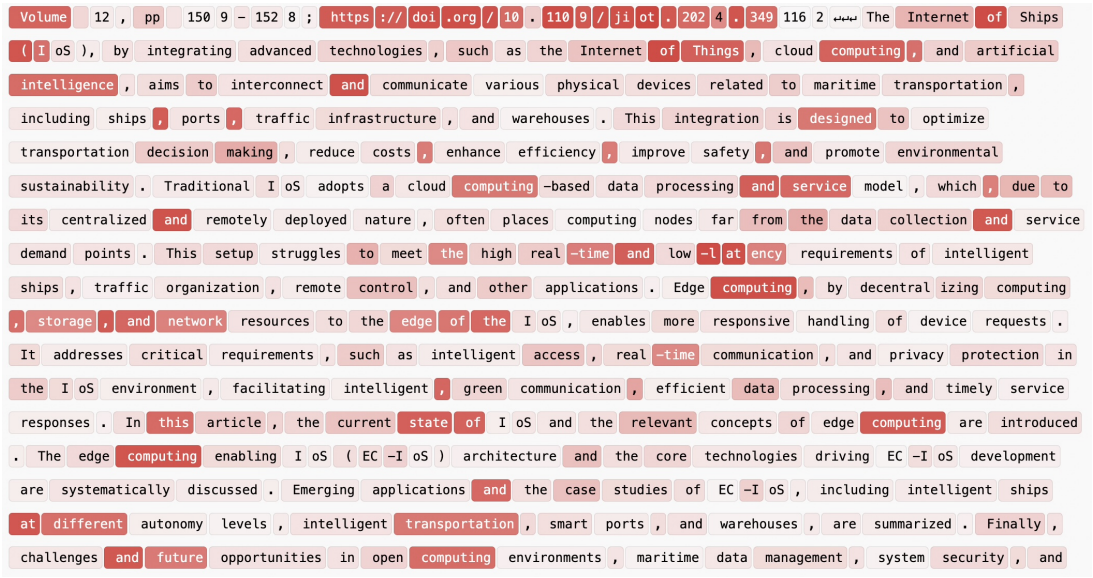


Figure 7: Visualization of DSE gate activations of a scientific paper.

## E Entity Resolution Example

Table 5: Entity resolution example reproduced from Ghandeharioun et al. (2024). This table illustrates the progressive integration of context tokens for: "Diana, Princess of Wales".

| Layer | Generation | Explanation |
|---|---|---|
| 1-2 | : Country in the United Kingdom | **Wales** |
| 3 | : Country in Europe | **Wales** |
| 4 | : Title held by female sovereigns in their own right or by queens consort | **Princess of Wales** (unspec.) |
| 5 | : Title given to the wife of the Prince of Wales (and later King) | **Princess of Wales** (unspec.) |
| 6 | : Diana, Princess of Wales (1961-1997), the first wife of Prince Charles, who was famous for her beauty | **Diana, Princess of Wales** |

## F System Design

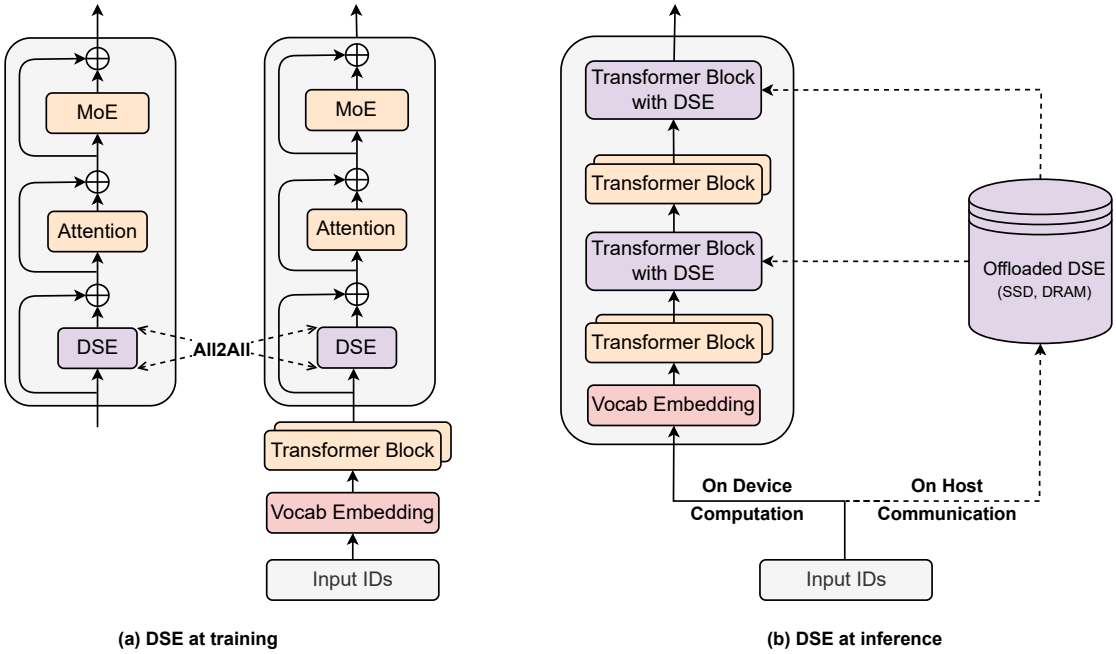


Figure 8: System implementation of DSE. (a) Training Phase: The massive embedding tables are sharded across available GPUs. An All-to-All communication primitive is employed to retrieve active embedding rows across devices. (b) Inference Phase: DSE tables are offloaded to host memory. By exploiting the deterministic retrieval logic, the host asynchronously prefetches and transfers embeddings, overlapping communication with the on-device computation of preceding Transformer blocks.

## G Sensitivity Analysis: Impact of DSE Module Ablation

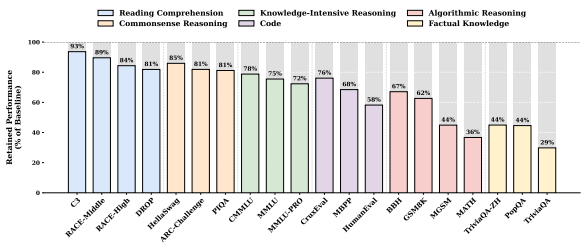


Figure 9: Retained performance under the DSE-DISABLED setting, where the sparse embedding output is completely suppressed.

**Methodology and Caveat.** To characterize the functional contribution of DSE, we evaluate the model (DSE-27B) under two intervention settings that override the learned gating mechanism while keeping the backbone unchanged. We report retained performance relative to the full model (Figures 9–10). Crucially, overriding a trained component induces a training–inference inconsistency, potentially introducing noise in complex tasks. Consequently, we prioritize the analysis of the two extremes of the sensitivity spectrum—Factual Knowledge and Reading Comprehension—which exhibit the highest signal-to-noise ratio under these stress tests.

**DSE-DISABLED** In this setting, we force the gate values to zero, effectively discarding the DSE signal (Figure 9). We observe a sharp functional dichotomy. Factual knowledge benchmarks suffer a catastrophic collapse, retaining only ∼29–44% of the original performance (e.g., TriviaQA at 29%). This confirms that DSE acts as the primary repository for parametric knowledge. In contrast, reading comprehension tasks are comparatively resilient, retaining ∼81–93% (e.g., C3 at 93%), suggesting that context-grounded tasks rely primarily on the backbone's attention mechanism.

**DSE-FORCED** We next consider the opposite extreme: bypassing the gating mechanism to force DSE injection at full strength (Figure 10). This yields an asymmetric trade-off relative to DSE-DISABLED. Factual knowledge substantially recovers (e.g., TriviaQA improves from 29% to 60%), suggesting that "always-on" retrieval facilitates fact recall. However, reading comprehension degrades sharply (e.g., C3 drops from 93% to 56%), implying that unconditional injection actively interferes with in-context signal.

This trade-off highlights the structural necessity of the gating mechanism. Because the DSE module operates via static routing on hash IDs, it is inherently context-agnostic and prone to hash collisions. The DSE-FORCED experiment demonstrates that unfiltered injection of such static signals introduces noise that disrupts precise reasoning. Consequently, contextualized gating is a critical design element that significantly enhances the module's expressivity. By leveraging the backbone's context-aware state to dynamically filter collision-induced noise, it effectively transforms a rigid, static lookup into a precise, context-dependent retrieval mechanism.

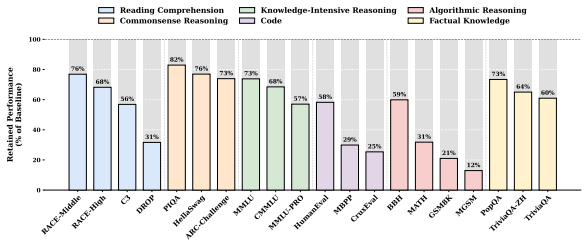


Figure 10: Retained performance under the DSE-FORCED setting, where the sparse embedding output is fully injected.

## H System Efficiency and Inference Latency

A pivotal system advantage of DSE over routing-based MoE is that its sparse activations are addressed by explicit, static hash IDs. This yields a strictly deterministic memory access pattern: indices for the next DSE lookup are fixed once the token sequence is known and can be computed before the corresponding layer executes. This property naturally facilitates communication–computation overlap. Concretely, the runtime can asynchronously stage required embedding vectors from host memory into GPU HBM while the GPU is occupied with dense attention and MLP kernels in preceding layers, effectively masking the host–device transfer latency.

**Experimental Setup.** We implemented an inference harness based on nano-vLLM[2]—a streamlined prototype of the industry-standard vLLM engine (Kwon et al., 2023). To obtain a clean latency baseline without the confounding communication patterns of expert dispatch, we benchmark on two dense backbones (Dense-4B and Dense-8B). We

[2] https://github.com/GeeeekExplorer/nano-vllm

insert a massive 100B-parameter DSE layer into the second Transformer block, with the entire embedding table resident in host DRAM. During inference, the system prefetches embeddings for the DSE layer asynchronously, overlapping the PCIe transfer with the computation of the first block.

Table 6: End-to-end Inference Throughput. We measure inference throughput with a 100B-parameter DSE layer offloaded to host memory.

| **Experimental Setup** | | |
|---|---|---|
| *Hardware* | | NVIDIA H800 |
| *Workload* | | 512 Sequences |
| *Sequence Length* | | Uniform(100, 1024) |
| **Throughput Results** | | |
| **Base Model** | **Config.** | **Thr. (tok/s)** |
| 4B-Dense | Baseline | 9,031.62 |
| | + 100B DSE | 8,858.28 |
| 8B-Dense | Baseline | 6,315.52 |
| | + 100B DSE | 6,140.02 |

**Results.** As detailed in Table 6, offloading a 100B-parameter embedding table incurs a negligible throughput penalty: only $1.9\%$ on the 4B backbone and $2.8\%$ on the 8B backbone. This confirms that the compute intensity of early dense blocks provides a sufficient temporal window to mask the retrieval latency. Crucially, the effective communication volume per step scales with the number of activated slots rather than the total embedding table size.

Crucially, these results represent a lower bound derived from a naive implementation that ignores request locality. Given the Zipfian nature of N-gram frequencies (Piantadosi, 2014; Chao and Zipf, 1950), this distribution naturally motivates a Multi-Level Cache Hierarchy that stratifies storage across GPU HBM, CPU DRAM, and even NVMe SSDs based on access frequency. We hypothesize that such infrastructure-aware optimization would effectively mask retrieval latency, rendering massive memory capacities practical with minimal overhead.

# I Case Study: Analysis on DSE Gating Mechanism

**What Does DSE Attend To?** One of the core design elements of DSE is the contextualized gate, which regulates the information flow into the Transformer's attention module. While the previous section detailed the general function of the DSE signal, this section examines the subtle nuances of its behavior within specific contexts. To this end, we run model inference on randomly selected Internet data as **granular** case studies to visualize and analyze the gating values assigned to individual tokens. To ensure the robustness and generalizability of our analysis, we curate hundreds of samples spanning diverse knowledge domains, linguistic styles, and languages—including news articles, broadcast transcripts, scientific papers, and open-source code. More examples can be found in Appendix D.

DSE-27B utilizes mHC (Xie et al., 2025), where each of the four propagation paths is equipped with a standalone DSE module. For simplicity, we aggregate the values across all paths and utilize the maximum activation value as the indicator of DSE module strength. As demonstrated in Figure 11, the DSE module distinctively selects tokens for contextualized modeling. The gate activation values exhibit high sparsity and semantic sensitivity. Specifically, we employ both bigram and trigram hash mapping functions, whereby information is aggregated at the terminal token of each $n$-gram. This aggregation is empirically reflected by significantly higher gate values at these positions, as illustrated in the figure.

Our empirical analysis reveals that the DSE gating mechanism functions as a contextualized semantic saliency filter. It prioritizes four distinct categories: (1) high-entropy, informational-dense entities ('piston rings', 'Corolla', 'Pontiac Sunbird'), (2) precise numerical constraints ('1,000 miles', '55 mph'), (3) structural anchors ('and', 'Because'), (4) common phrases ('Take your chances', 'nose-hair's worth of').

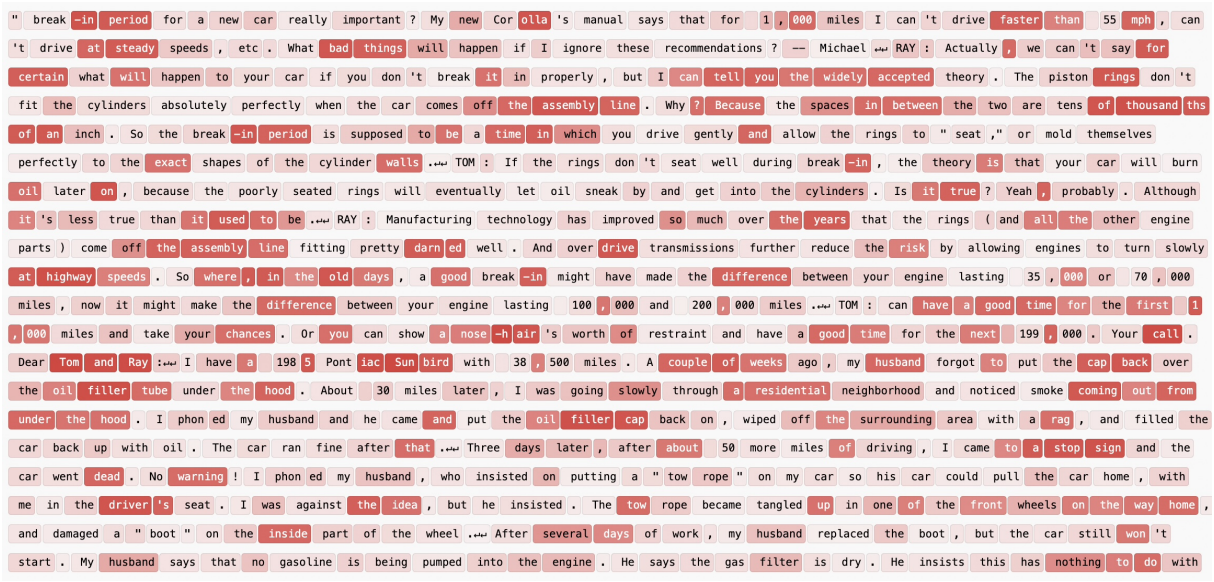


Figure 11: Illustration of DSE gate values on a broadcast snippet, deeper colors indicate higher gate values. Entities, numerical values, structural anchors and common phrases are the most common activated patterns.

# J Long Context Training Details

## J.1 Experimental Setup

**Training Details.** To enable long-context capabilities, we adopt similar context expansion strategy introduced in DeepSeek-V3 (Liu et al., 2024). Following the pre-training stage, we apply YaRN (Peng et al., 2023b) for context window extension in a 32768-token context training stage for 5,000 steps. The hyper-parameters are scale $s = 10, \alpha = 1, \beta = 32$ and the scaling factor $f = 0.707$. To align with the token count per batch in the pre-training phase, we scale down the training batch size by a factor of eight. The training corpus comprises approximately 30B tokens of high-quality, long-context data.

**Model Configurations.** We compare context extensions across four distinct model configurations. We utilize the final pre-training checkpoints (50k steps) for both MoE-27B and DSE-27B. Additionally, to rigorously benchmark architectural efficiency, we select two intermediate checkpoints for DSE-27B at 41k and 46k steps. Crucially, DSE-27B (46k) is selected because it exhibits the same pre-training loss as the fully trained MoE-27B (50k). This creates a controlled "Iso-Loss" setting, ensuring that any performance divergence during context extension is attributable to the architecture (DSE vs. MoE) rather than the starting quality of the language model.

**Evaluation Benchmarks.** We assess long-context performance using LongPPL (Fang et al.) and RULER (Hsieh et al.). For LongPPL, we construct evaluation sets spanning four categories: long books, research papers, code repositories, and long chain-of-thought (CoT) trajectories. For RULER, we evaluate on 14 subsets aggregated into 8 categories: Single (S), Multi-keys (MK), Multi-values (MV) and Multi-queries (MQ) Needle-in-a-Haystack; Multi-hop Variable Tracking (VT), Common Words Extraction (CWE), Frequent Words Extraction (FWE), and Question Answering (QA).