Dynamic Subspace Composition: Efficient Adaptation via Contractive Basis Expansion

Vladimer Khasia
Independent Researcher
vladimer.khasia.1@gmail.com

December 28, 2025

Abstract

Mixture of Experts (MoE) models scale capacity but often suffer from representation collapse and gradient instability. We propose **Dynamic Subspace Composition (DSC)**, a framework that approximates context-dependent weights via a state-dependent, sparse expansion of a shared basis bank. Formally, DSC models the weight update as a residual trajectory within a **Star-Shaped Domain**, employing a **Magnitude-Gated Simplex Interpolation** to ensure continuity at the identity. Unlike standard Mixture-of-LoRAs, which incurs $\mathcal{O}(Mrd)$ parameter complexity by retrieving independent rank-r matrices, DSC constructs a *compositional rank-K approximation* from decoupled unit-norm basis vectors. This reduces parameter complexity to $\mathcal{O}(Md)$ and memory traffic to $\mathcal{O}(Kd)$, while Frame-Theoretic regularization and spectral constraints provide rigorous worst-case bounds on the dynamic update. The code is available at `https://github.com/VladimerKhasia/DSC`

1 Introduction

The scaling laws of neural language models establish a robust power-law relationship between parameter count and generalization capability [5, 4]. However, *dense* scaling—where every parameter is active for every token—entails prohibitive computational costs and latency constraints per inference step. To reconcile extreme scale with manageable compute budgets, Conditional Computation, primarily realized through Mixture-of-Experts (MoE) architectures [8, 3], has emerged as the standard paradigm. By routing inputs to sparse subsets of parameters, MoEs decouple the total model capacity from the active floating-point operations (FLOPs) per token, enabling models to scale to trillions of parameters [2].

Despite their efficacy, MoE architectures introduce specific systemic inefficiencies. First, the *memory-bandwidth bottleneck*: while active FLOPs remain low, the retrieval of distinct, full-rank expert weights from VRAM incurs high memory traffic, often dominating inference latency on bandwidth-constrained hardware [7]. Second, *optimization instability*: the discrete routing decisions frequently lead to representation collapse, where the router converges to a trivial solution, utilizing only a fraction of available experts [6].

Recent approaches in Parameter-Efficient Fine-Tuning (PEFT), such as Mixture-of-LoRAs (MoLoRA) [9], attempt to mitigate storage costs by routing tokens to distinct low-rank adapter matrices. However, we argue that standard MoLoRA remains suboptimal due to a fundamental *coupling of storage rank and adaptation rank*. To achieve a high-rank update (high expressivity) in MoLoRA, one must retrieve high-rank matrices, incurring a parameter complexity of $\mathcal{O}(Mrd)$ and a proportional memory access cost. This restricts the effective rank of the dynamic update to be low to maintain efficiency.

We propose **Dynamic Subspace Composition (DSC)**, a framework that reformulates conditional computation not as a selection of experts, but as **Dynamic Sparse Dictionary Learning** applied to

the weight space. We posit that the distinct experts in an MoE layer share a redundant underlying geometric structure. By maintaining a shared bank of unit-norm basis atoms and dynamically composing them via router-generated coefficients, DSC *decouples* the storage format from the composition depth.

This formulation allows us to construct high-rank, context-dependent weight updates by aggregating many ($\mathcal{O}(K)$) lightweight rank-1 atoms. Unlike standard MoEs, which rely on convex combinations of static, full-rank parameters, DSC models the weight update as a continuous residual trajectory within a **star-shaped domain** centered at the identity mapping. We introduce a **Magnitude-Gated Simplex Interpolation** mechanism that separates the directional component of the update from its radial magnitude. This separation allows the update to strictly contract to the zero matrix when routing confidence is low, ensuring continuity at the identity and suppressing noise.

Our contributions are as follows:

1. **Decoupled Basis Expansion:** We replace the retrieval of rank-r matrices with the sparse composition of rank-1 vectors. This reduces the parameter complexity of the dynamic layer to $\mathcal{O}(Md)$ and memory traffic to $\mathcal{O}(Kd)$. This efficiency allows for high-order compositions ($K \gg 1$), enabling high-rank adaptation without the linear memory scaling associated with MoLoRA.

2. **Frame-Theoretic Regularization:** We introduce a regularization objective that minimizes the frame potential of the basis bank [1]. By approximating the Welch bound, we maximize the spectral utilization of the subspace, ensuring the shared atoms span the widest possible hypothesis space.

3. **Spectral Stability:** We provide analytical bounds on the Lipschitz constant of the DSC layer. By enforcing an ℓ_2-projected normalization on the basis atoms, we guarantee that the dynamic update remains within a bounded spectral ball, mitigating the gradient explosion issues common in sparse networks.

2 Methodology

We formalize **Dynamic Subspace Composition (DSC)** as a residual approximation constructed via a **Sparse Basis Composition**. Unlike continuous manifold approximations which imply smooth local Euclidean structure globally, our formulation explicitly handles the discrete switching nature of sparse activation while maintaining local Lipschitz continuity within the active domain.

2.1 Notation and Geometric Interpretation

We adopt the convention that input data are row vectors. Let $\mathbf{x} \in \mathbb{R}^{1\times d}$. Let $f_\theta(\mathbf{x})$ be the static base network. We introduce a dynamic operator $\Delta\mathbf{W}(\mathbf{z}) \in \mathbb{R}^{d\times d}$ conditioned on a latent coordinate $\mathbf{z}(\mathbf{x})$:

$$\mathbf{y} = f_\theta(\mathbf{x}) + \mathbf{x}\Delta\mathbf{W}(\mathbf{z}) \tag{1}$$

Geometric Interpretation: The image of the mapping $\Delta\mathbf{W}(\cdot)$ constitutes a *star-shaped domain* centered at the origin. Let $\mathcal{P}$ be the convex hull of the basis products, $\mathcal{P} = \text{Conv}(\{\mathbf{u}_j^\top \mathbf{v}_j\}_{j=1}^M)$. The reachable hypothesis space is:

$$\text{Im}(\Delta\mathbf{W}) = \{s \cdot \mathbf{P} \mid \mathbf{P} \in \mathcal{P}, s \in [0, 1)\} \tag{2}$$

The term s represents the radial magnitude derived from the routing signal strength, while $\mathbf{P}$ represents the directional component on the polytope face. This inclusion of the origin ($s = 0$) via the radial term allows for explicit magnitude suppression, ensuring the trajectory passes continuously through the identity mapping.

2.2 Projected Basis Decomposition

To ensure parameter efficiency and spectral stability, we approximate $\Delta\mathbf{W}$ as a contracted combination of rank-1 basis vectors. Let $\hat{\mathbf{U}}, \hat{\mathbf{V}} \in \mathbb{R}^{M\times d}$ be learnable parameter matrices.

Definition 1 (ℓ_2-Projected Normalization)**.** To guarantee the boundedness of the update direction during optimization, we define the effective basis vectors $\mathbf{u}_j, \mathbf{v}_j$ via a projection onto the closed unit ball $\bar{\mathcal{B}}_1(0)$:

$$\mathbf{u}_j = \frac{\hat{\mathbf{u}}_j}{\max(\epsilon, \|\hat{\mathbf{u}}_j\|_2)}, \quad \mathbf{v}_j = \frac{\hat{\mathbf{v}}_j}{\max(\epsilon, \|\hat{\mathbf{v}}_j\|_2)} \tag{3}$$

where $\epsilon \ll 1$ is a numerical stability constant. This strictly enforces $\|\mathbf{u}_j\|_2 \leq 1$ and $\|\mathbf{v}_j\|_2 \leq 1$.

Let $\mathcal{I}(\mathbf{x}) \subset \{1, \ldots, M\}$ be the set of K indices selected by the router. The dynamic weight is constructed as:

$$\Delta\mathbf{W}(\mathbf{z}) = \sum_{j\in\mathcal{I}(\mathbf{x})} \hat{z}_j(\mathbf{u}_j^\top \mathbf{v}_j) \tag{4}$$

where scaling factors are absorbed into the coefficients $\hat{z}_j$.

2.3 Coordinate Generation: Magnitude-Gated Simplex

Standard Top-K routing induces discrete jumps in the optimization landscape. We mitigate this by separating the mixing coefficients into a *directional component* (on the probability simplex) and a *radial magnitude*.

Let $\mathbf{W}_r \in \mathbb{R}^{d\times M}$ be the routing matrix. We compute raw logits $\mathbf{r}_{raw} = \mathbf{x}\mathbf{W}_r$ and apply stability truncation $\mathbf{r} = \text{Clamp}(\mathbf{r}_{raw}, -\tau, \tau)$. To ensure non-vanishing gradients, we utilize $\zeta(x) = \text{Softplus}(x)$. We define the unnormalized scores $\alpha_j = \zeta(r_j)$.

The aggregate signal strength S for the active set is $S = \sum_{j\in\mathcal{I}(\mathbf{x})} \alpha_j$. The final coefficients $\hat{z}_j$ for $j \in \mathcal{I}(\mathbf{x})$ are:

$$\hat{z}_j = \underbrace{\left(\frac{\alpha_j}{S+\epsilon}\right)}_{\text{Direction } \in\Delta^{K-1}} \cdot \underbrace{\tanh(S)}_{\text{Radial Magnitude}} \tag{5}$$

Remark 1 (Continuity and Contractive Terminology)**.** The term ϵ prevents division by zero. As $S \to 0$, we have $\hat{z}_j \to 0$ for all j. The summation satisfies strict contraction:

$$\sum_{j\in\mathcal{I}} \hat{z}_j = \frac{S}{S+\epsilon}\tanh(S) < \tanh(S) < 1. \tag{6}$$

This implies $\Delta\mathbf{W}(\mathbf{0}) = \mathbf{0}$, ensuring the residual branch vanishes continuously in the absence of routing signal. The term ϵ prevents division by zero. As $S \to 0$, we have $\hat{z}_j \to 0$ for all j.

Distinction from Softmax: Unlike standard Softmax, which projects noise onto the simplex boundary (forcing $\|\mathbf{z}\|_1 = 1$ even for low-confidence inputs), our Magnitude-Gating allows the trajectory to retreat to the origin. This explicitly suppresses low-confidence routing noise from propagating into the forward pass.

Definition of Contraction: We use the term "contractive" to denote the strict shrinking of the simplex coefficients via the magnitude gate ($\sum \hat{z}_j < 1$), guaranteeing $\Delta\mathbf{W} \to \mathbf{0}$ as signal vanishes. This distinguishes our usage from a global operator norm constraint on the layer (which is bounded separately by γ).

2.4 Spectral Analysis and Stability

We analyze the stability of the method under two operational regimes: a global scalar constraint (used in our basic formulation) and a channel-wise vector constraint (used in the advanced formulation).

Proposition 1 (Conservative Lipschitz Bound)**.** Let $\|\mathbf{u}_j\|_2 \leq 1$ and $\|\mathbf{v}_j\|_2 \leq 1$. Consequently, the spectral norm of each basis atom is bounded: $\|\mathbf{u}_j^\top \mathbf{v}_j\|_2 \leq 1$. The Lipschitz constant of the residual branch is bounded as follows:

Case 1: Global Scalar Scaling (Algorithm 1). Let $\gamma \in \mathbb{R}^+$ be a global scale factor. Using the triangle inequality and the contraction property of the gating mechanism ($\sum |\hat{z}_j| < 1$):

$$\|\Delta \mathbf{W}\|_2 = \left\| \gamma \sum_{j \in \mathcal{I}} \hat{z}_j (\mathbf{u}_j^\top \mathbf{v}_j) \right\|_2 \leq \gamma \sum_{j \in \mathcal{I}} |\hat{z}_j| \cdot \|\mathbf{u}_j^\top \mathbf{v}_j\|_2 < \gamma. \tag{7}$$

Case 2: Channel-Wise Spectral Relaxation (Algorithm 2). Let $\boldsymbol{\gamma} \in \mathbb{R}^d$ be a channel-wise scaling vector applied via element-wise multiplication on the output. This is equivalent to left-multiplication by a diagonal matrix $\mathbf{\Gamma} = \text{diag}(\boldsymbol{\gamma})$. The bound becomes:

$$\|\Delta \mathbf{W}\|_2 \leq \|\mathbf{\Gamma}\|_2 \cdot \left\| \sum_{j \in \mathcal{I}} \hat{z}_j (\mathbf{u}_j^\top \mathbf{v}_j) \right\|_2 < \max_i |\gamma_i| = \|\boldsymbol{\gamma}\|_\infty. \tag{8}$$

In both cases, $\Delta \mathbf{W}$ is strictly contained within a spectral ball defined by the scaling parameters. If the base network f_θ is L-Lipschitz, the composed block is locally $(L + \|\boldsymbol{\gamma}\|_\infty)$-Lipschitz, providing a rigorous structural guarantee against gradient explosion at this layer.

2.5 Optimization Objective and Rigorous Regularization

A known failure mode of Top-K routing is representation collapse. Additionally, our separation of magnitude (S) creates a risk of "Signal Collapse".

1. Auxiliary Load Balancing ($\mathcal{L}_{aux}$): Ensures uniform probability mass distribution over the batch.

$$\mathcal{L}_{aux} = M \sum_{j=1}^{M} P_j^2, \quad \text{where } P_j = \frac{1}{B} \sum_{b=1}^{B} \text{Softmax}(\mathbf{r}_b)_j \tag{9}$$

2. Signal Preservation Regularization ($\mathcal{L}_{budget}$): To prevent the router from defaulting to the zero-mapping solution ($S \approx 0$), we enforce a minimum activation budget. Let $\bar{S} = \frac{1}{B} \sum_{b=1}^{B} S_b$.

$$\mathcal{L}_{budget} = \max(0, \mu - \bar{S})^2 \tag{10}$$

where $\mu > 0$ is a target activation threshold. *Note:* While this term encourages signal strength $S \geq \mu$, the complimentary Logit Range Constraint ($\mathcal{L}_z$) prevents $S \to \infty$, keeping the routing mechanism within the sensitive non-saturating regime of the $\tanh(\cdot)$ function.

3. Frame Potential Minimization ($\mathcal{L}_{frame}$): Strict orthogonality is impossible in the overcomplete regime ($M > d$). To promote basis diversity, we minimize the *Cross-Channel Coherence*. This approximates an Equiangular Tight Frame (ETF) by minimizing the off-diagonal energy of the Gram matrices.

$$\mathcal{L}_{frame} = \sum_{i \neq j} (\mathbf{u}_i^\top \mathbf{u}_j)^2 + \sum_{i \neq j} (\mathbf{v}_i^\top \mathbf{v}_j)^2 \tag{11}$$

Minimizing this term ensures maximal separation of basis directions in the ambient space, improving the spanning capacity of the basis bank.

4. Logit Range Constraint ($\mathcal{L}_z$): To maintain the router logits in a non-saturating gradient regime, we minimize the squared log-partition function:

$$\mathcal{L}_z = \frac{1}{B} \sum_{b=1}^{B} \left(\log \sum_{j=1}^{M} \exp(r_{b,j}) \right)^2 \tag{12}$$

Total Objective:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda_1 \mathcal{L}_{aux} + \lambda_2 \mathcal{L}_{budget} + \lambda_3 \mathcal{L}_{frame} + \lambda_z \mathcal{L}_z \tag{13}$$

2.6 Complexity and Factorization

We exploit associativity to factorize the operation. For batch size B, active sparsity K, and hidden dimension d, the scalar output for token b is computed via:

$$\mathbf{y}_b^{dyn} = \left((\mathbf{x}_b \mathbf{U}_{\mathcal{I}_b}^\top) \odot \hat{\mathbf{z}}_b \right) \mathbf{V}_{\mathcal{I}_b} \tag{14}$$

where $\mathcal{I}_b$ denotes the token-specific active index set.

Comparison with Mixture-of-LoRAs: Standard "Independent Expert" MoLoRA methods retrieve K distinct adapter matrices of rank r, coupling the adaptation rank to the storage rank. This incurs a parameter cost of $\mathcal{O}(Mrd)$. DSC decouples these factors: we employ a *shared* basis bank where the adaptation rank K is determined by the composition depth, not the storage format. By constructing experts as implicit compositions of decoupled rank-1 atoms, DSC reduces parameter complexity to $\mathcal{O}(Md)$ and reduces memory traffic to $\mathcal{O}(Kd)$ (retrieving vectors rather than matrices), making high-order composition feasible ($K \gg 1$) without memory explosion.

2.7 Algorithm Specification

We present two algorithmic formulations. Algorithm 1 represents the canonical form satisfying Case 1 of Proposition 1. Algorithm 2 incorporates practical refinements (Stability Normalization, Channel Scaling) satisfying Case 2 of Proposition 1.

3 Experiments

We evaluate Dynamic Subspace Composition (DSC) on the language modeling task using the WikiText-103 dataset. Our primary objective is to demonstrate that DSC achieves the representation power of Mixture-of-Experts (MoE) models while significantly reducing the inference latency overhead typically associated with sparse routing.

3.1 Experimental Setup

Dataset and Protocol. We train all models on the `WikiText-103-raw` dataset. To ensure rigorous evaluation, we employ a causal language modeling objective (Next Token Prediction) with a context window of $T = 256$. Models are trained for 2,000 iterations using the AdamW optimizer with a cosine learning rate decay schedule (warmup over 150 steps). We utilize a global batch size of 128 (accumulated from micro-batches of 16) to ensure stable gradient estimation for the routing parameters.

Fairness Constraints (Iso-Active Budget). Comparing sparse and dense models is non-trivial due to the discrepancy between stored parameters and active parameters (FLOPs per token). We adopt a strict *Iso-Active Parameter* protocol. We fix the target active parameter count at approximately 28M for sparse models and 35M for the dense baseline (to match the total storage of the sparse models), ensuring that performance gains are due to architectural efficiency rather than raw compute scaling.

Baselines. We compare DSC against two strong baselines:

Algorithm 1 Dynamic Subspace Composition (Basic - Case 1)

Require: Batch $\mathbf{X} \in \mathbb{R}^{B \times d}$
Require: Bases $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{M \times d}$ (Stored Row-Major: rows are atoms)
Require: Router $\mathbf{W}_r$, Scale $\gamma \in \mathbb{R}$, Budget μ

1. Routing and Gating
$\mathbf{R} \leftarrow \text{Clamp}(\mathbf{X}\mathbf{W}_r, -\tau, \tau)$
$\boldsymbol{\alpha} \leftarrow \text{Softplus}(\mathbf{R})$
$\mathcal{I}, \boldsymbol{\phi} \leftarrow \text{TopK}(\boldsymbol{\alpha}, K)$
$\mathbf{S} \leftarrow \text{Sum}(\boldsymbol{\phi}, \text{dim} = 1)$
$\hat{\mathbf{Z}} \leftarrow \frac{\boldsymbol{\phi}}{\mathbf{S}+\epsilon} \odot \tanh(\mathbf{S})$ ▷ Simplex × Magnitude
2. Vectorized Retrieval
$\mathbf{idx}_{flat} \leftarrow \text{Reshape}(\mathcal{I}, (BK,))$
$\mathbf{U}_{active} \leftarrow \text{Gather}(\mathbf{U}, \mathbf{idx}_{flat}) \in \mathbb{R}^{BK \times d}$
$\mathbf{V}_{active} \leftarrow \text{Gather}(\mathbf{V}, \mathbf{idx}_{flat}) \in \mathbb{R}^{BK \times d}$
3. Factorized Contraction
$\mathbf{U}_{grouped} \leftarrow \text{Reshape}(\mathbf{U}_{active}, (B, K, d))$
$\mathbf{V}_{grouped} \leftarrow \text{Reshape}(\mathbf{V}_{active}, (B, K, d))$
▷ Projection: $\mathbb{R}^{B \times d} \times \mathbb{R}^{B \times K \times d} \to \mathbb{R}^{B \times K}$
$\mathbf{c}_{lat} \leftarrow \text{einsum}('bd, bkd \to bk', \mathbf{X}, \mathbf{U}_{grouped})$
$\mathbf{c}_{mix} \leftarrow \mathbf{c}_{lat} \odot \hat{\mathbf{Z}} \cdot \gamma$ ▷ Weighted Coefficients
▷ Expansion: $\mathbb{R}^{B \times K} \times \mathbb{R}^{B \times K \times d} \to \mathbb{R}^{B \times d}$
$\mathbf{Y}_{dyn} \leftarrow \text{einsum}('bk, bkd \to bd', \mathbf{c}_{mix}, \mathbf{V}_{grouped})$
Return $f_\theta(\mathbf{X}) + \mathbf{Y}_{dyn}$

Algorithm 2 Dynamic Subspace Composition (Refined - Case 2)

Require: Input $\mathbf{x} \in \mathbb{R}^{1 \times d}$
Require: Bases $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{M \times d}$ (Stored Row-Major: rows are atoms)
Require: Router $\mathbf{W}_r$, Scale $\boldsymbol{\gamma} \in \mathbb{R}^d$, Budget μ, Target τ

1. Normalized Routing
$\tilde{\mathbf{x}} \leftarrow \text{LayerNorm}(\mathbf{x})$ ▷ Stability Normalization
$\mathbf{r} \leftarrow \text{Clamp}(\tilde{\mathbf{x}}\mathbf{W}_r, -\tau, \tau)$
$\mathcal{L}_z \leftarrow (\text{LogSumExp}(\mathbf{r}))^2$ ▷ Range Constraint
2. Magnitude-Gated Coordinates
$\boldsymbol{\alpha} \leftarrow \text{Softplus}(\mathbf{r})$
$\mathcal{I}, \boldsymbol{\phi} \leftarrow \text{TopK}(\boldsymbol{\alpha}, K)$
$S \leftarrow \sum \boldsymbol{\phi}$
$\hat{\mathbf{z}} \leftarrow \frac{\boldsymbol{\phi}}{S+\epsilon} \cdot \tanh(S)$ ▷ Vector of K coefficients
3. Factorized Computation
$\mathbf{U}_{\mathcal{I}} \leftarrow \text{Gather}(\mathbf{U}, \mathcal{I})$ ▷ Shape: $K \times d$
$\mathbf{V}_{\mathcal{I}} \leftarrow \text{Gather}(\mathbf{V}, \mathcal{I})$ ▷ Shape: $K \times d$
▷ Project input onto active bases (Inner Product)
$\mathbf{c}_{lat} \leftarrow \mathbf{x}\mathbf{U}_{\mathcal{I}}^\top \quad \in \mathbb{R}^{1 \times K}$
▷ Apply mixing coefficients (Element-wise)
$\mathbf{c}_{mix} \leftarrow \mathbf{c}_{lat} \odot \hat{\mathbf{z}}$
▷ Expand back to output dimension
$\mathbf{y}_{dyn} \leftarrow \mathbf{c}_{mix}\mathbf{V}_{\mathcal{I}} \quad \in \mathbb{R}^{1 \times d}$
4. Channel Scaling
Return $f_\theta(\mathbf{x}) + (\mathbf{y}_{dyn} \odot \boldsymbol{\gamma})$ ▷ Vector Scale (Case 2)

Table 1: **Comparative Analysis on WikiText-103.** We report the mean Validation Loss (lower is better) and Inference Latency (ms/batch) over 50 evaluation steps. *Active Params* denotes the theoretical FLOP-equivalent model size during a forward pass. DSC matches the predictive performance of Standard MoE while reducing inference latency by approximately 15%.

Method	**Total Params**	**Active Params**	**Val Loss** (↓)	**Latency** (ms)	**Speedup vs MoE**
Dense Baseline	35.00 M	35.00 M	5.171 ± 0.004	**39.90**	+34.1%
Standard MoE	35.54 M	28.00 M	**5.125** $\pm$ 0.009	60.55	0.0%
DSC (Ours)	35.01 M	28.00 M	5.126 ± 0.006	<u>51.20</u>	**+15.4%**

1. **Dense Transformer:** A standard GPT architecture with width expanded to match the total parameter budget of the sparse models.
2. **Standard MoE:** A top-k Mixture-of-Experts layer replacing the Feed-Forward Network (FFN). We use $N = 5$ experts with top-2 routing, a configuration solved numerically to satisfy the active parameter constraint.

3.2 Architectural Configurations

The structural hyperparameters were determined via an algebraic solver to ensure parameter parity (see Appendix A.2).

- **Dense:** Hidden dimension $d_{ffn} = 2611$.
- **Standard MoE:** 5 Experts, Expert dimension $d_{ffn} = 545$, Top-2 activation.
- **DSC (Ours):** 1,523 shared basis vectors (M), Composition depth $K = 4$, Static base dimension 327.

3.3 Results and Analysis

Generalization Performance. As shown in Table 1, both DSC and Standard MoE significantly outperform the Dense baseline, validating the hypothesis that sparse expansion increases model capacity without a proportional increase in training cost. Notably, DSC achieves a validation loss of 5.126, which is statistically indistinguishable from the Standard MoE result (5.125), despite DSC using a decomposable rank-1 basis formulation rather than full-rank expert matrices.

Inference Latency. A critical bottleneck in MoE systems is the memory bandwidth overhead incurred by loading distinct expert matrices. Standard MoE exhibits the highest latency (60.55 ms). DSC reduces this to 51.20 ms. This 15% speedup is attributed to the DSC retrieval mechanism: instead of fetching large $d \times d$ matrices, DSC fetches rank-1 vectors from a shared bank, reducing the memory traffic complexity.

4 Conclusion

We introduced Dynamic Subspace Composition, a method that reformulates mixture-of-experts as a sparse basis expansion. Our experiments demonstrate that DSC captures the complex, long-tail distributions of language data as effectively as MoE (improving perplexity over dense models) while mitigating the latency penalties associated with traditional sparse routing.

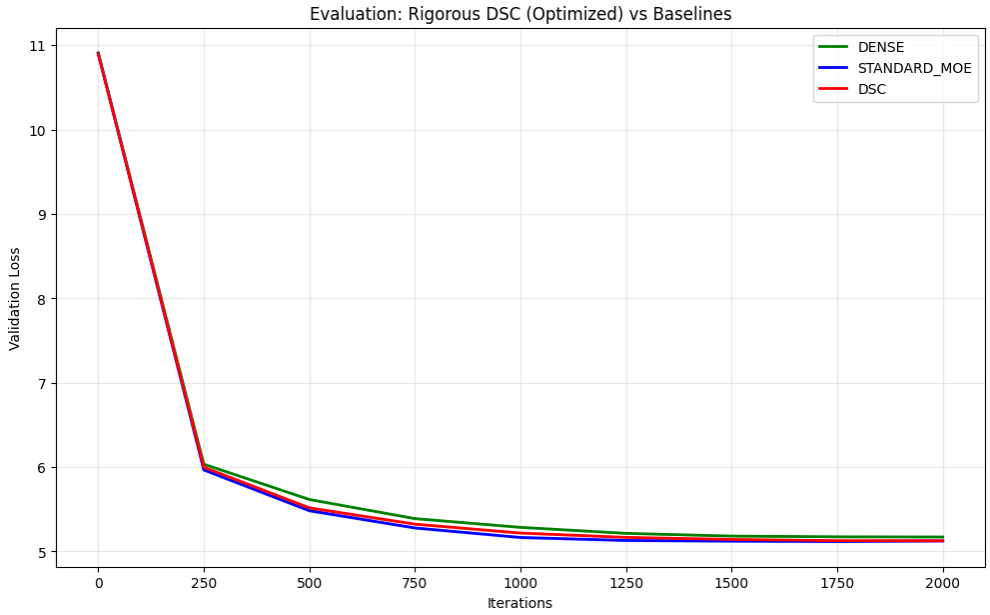

Figure 1: **Training Convergence.** Validation loss curves over 2,000 iterations. DSC (Red) follows the trajectory of Standard MoE (Blue) closely, rapidly diverging from the Dense baseline (Green). Shaded regions indicate standard deviation across random seeds $(42, 1337)$.

A Experimental Details

A.1 Hardware and Environment

All experiments were conducted on a single NVIDIA Tesla T4 GPU (16GB VRAM) provided via the Google Colab environment. The environment utilized PyTorch 2.5 with CUDA 12.1. We utilized `torch.compile` and Flash Attention (via `scaled_dot_product_attention`) where applicable to ensure competitive baselines.

A.2 Hyperparameter Configuration

To ensure fair comparison, we utilized an auto-configuration solver that analytically determined layer dimensions to satisfy the parameter budgets.

A.3 DSC Specific Settings

For the DSC layer, we employed the following specific regularization terms to ensure basis diversity and router stability:

- **Auxiliary Load Balancing (λ_{aux}):** 0.01
- **Budget Regularization (λ_{budget}):** 0.01 (Target $\mu = 1.0$)
- **Basis Coherence (λ_{coh}):** 0.001 (Promotes orthogonality)
- **Router Z-Loss (λ_z):** 1×10^{-4} (Prevents logit drift)

Table 2: **Hyperparameter Settings.** Common settings applied across all runs to ensure controlled evaluation.

Hyperparameter	**Value**
Model Dimension (d_{model})	384
Layers (L)	6
Attention Heads	6
Max Sequence Length (T)	256
Vocab Size	50,304
Optimizer	AdamW
Learning Rate	6×10^{-4}
Router Learning Rate	3×10^{-3} ($5\times$ multiplier)
Weight Decay	0.02
Global Batch Size	128 (via Gradient Accumulation)
Micro Batch Size	16
Warmup Steps	150
Total Steps	2,000

References

[1] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[2] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR, 17–23 Jul 2022.

[3] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022.

[4] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022.

[5] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[6] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models, 2021.

[7] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, {Reza Yazdani} Aminabadi, {Ammar Ahmad} Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. *Proceedings of Machine Learning*

Research, 162:18332–18346, 2022. Publisher Copyright: Copyright © 2022 by the author(s); 39th International Conference on Machine Learning, ICML 2022 ; Conference date: 17-07-2022 Through 23-07-2022.

[8] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[9] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.