Sparse Selective Hyper-Connections: A Unified Framework for Stable and Efficient Deep Residual Learning

Anonymous Authors
Under Review

***Abstract*—We introduce Sparse Selective Hyper-Connections (SHC), a practical efficiency framework for multi-stream residual architectures that achieves substantial computational and memory improvements over Manifold-Constrained Hyper-Connections (MHC) while maintaining equivalent accuracy. Motivated by the Birkhoff-von Neumann theorem's sparse decomposition insights, we develop *sparse mixtures of Cayley-parameterized orthogonal matrices* that eliminate iterative Sinkhorn normalization. Our primary contributions are efficiency gains: (1) 16× speedup in routing computation via closed-form Cayley transform, (2) 3.3× reduction in KV cache memory through rank-r factorization, and (3) optional $\mathcal{O}(L)$ inference via SSM distillation. Theoretical analysis proves SHC maintains bounded spectral norm $\rho(\mathbf{H}^{\text{res}}) \leq 1$ by construction, compared to the approximate bound $\rho \leq 1.6$ achieved by MHC with finite Sinkhorn iterations. These efficiency improvements enable practical deployment of multi-stream architectures that were previously prohibitive due to Sinkhorn overhead and memory explosion, while matching MHC accuracy across diverse benchmarks including long-context tasks where memory is critical.**

I. Introduction

The residual connection [1] has become a cornerstone of modern deep learning, enabling the training of networks with hundreds of layers by providing identity mappings that preserve gradient flow. This architectural innovation fundamentally transformed the landscape of deep network design, making it possible to train models of unprecedented depth without suffering from vanishing gradients. The success of residual connections has motivated extensive research into more sophisticated connectivity patterns that can further enhance representational capacity while maintaining training stability.

Recent work on Hyper-Connections (HC) [2] expanded this paradigm by introducing multiple parallel residual streams with learnable cross-stream mixing, demonstrating significant performance improvements on large language models. The core insight behind Hyper-Connections is that a single residual stream may be insufficient to capture the diverse information flow patterns required for complex reasoning tasks. By expanding to n parallel streams with learnable mixing matrices, the architecture gains additional capacity to route information selectively through specialized pathways.

However, HC suffers from severe training instability at scale. DeepSeek's Manifold-Constrained Hyper-Connections (MHC) [3] addressed this by constraining mixing matrices to the doubly stochastic manifold via Sinkhorn-Knopp iteration [4]. While this approach offers an elegant solution grounded in optimal transport theory, MHC introduces three critical limitations that constrain its practical applicability. First, the **computational overhead** is substantial because the method requires 20 Sinkhorn iterations per layer, and these iterations cannot be parallelized with the main computation path. Second, there is a **memory explosion** problem since the approach requires $n\times$ KV cache expansion for n streams, which becomes prohibitive for long-context inference scenarios where memory bandwidth is the primary bottleneck. Third, the **approximate constraints** produced by finite iterations yield $\rho(\mathbf{H}^{\text{res}}) \approx 1.6$ rather than the tighter bounds that theoretical analysis would desire, potentially introducing subtle instabilities under adversarial conditions or low-precision computation.

We propose **Sparse Selective Hyper-Connections** (SHC), a novel framework that resolves these limitations while providing stronger theoretical guarantees. Our approach is motivated by the Birkhoff-von Neumann theorem, which shows that doubly stochastic matrices can be decomposed as sparse convex combinations of permutation matrices. However, rather than directly implementing this decomposition (which would require discrete optimization over permutations), we observe that *orthogonal matrices* provide similar stability properties with a key advantage: they can be parameterized in closed form via the Cayley transform. This insight leads us to develop sparse mixtures of orthogonal matrices, which provide guaranteed bounded spectral norm $\rho \leq 1$ without iteration, while offering greater expressivity than permutation matrices alone. Our contributions are fourfold:

- We introduce a **closed-form sparse orthogonal mixture** that parameterizes convex combinations of Cayley-generated orthogonal matrices, thereby eliminating Sinkhorn iteration entirely while guaranteeing $\rho(\mathbf{H}^{\text{res}}) \leq 1$.
- We develop **Cayley orthogonal routing** that provides bounded spectral norm via Lie group parameterization, ensuring stable signal propagation through arbitrarily deep networks.
- We propose a **factorized stream representation** that reduces KV cache requirements from $n\times$ to approximately $1\times$ using learned low-rank projections.
- We enable **SSM-hybrid inference** that achieves $\mathcal{O}(L)$ decoding complexity by distilling the trained multi-stream architecture into a state space model.

The remainder of this paper is organized as follows. Section III presents the theoretical foundations underlying our approach, including the Birkhoff-von Neumann theorem (as motivation) and the Cayley transform. Section IV describes the complete sHC architecture, including sparse orthogonal mixtures, factorized stream representation, adaptive rank selection, and SSM hybrid inference. Section V provides the algorithmic description of the forward pass, while Section VI presents theoretical analysis of stability and expressivity guarantees. Section VII provides comprehensive experimental evaluation including benchmark comparisons, efficiency analysis, ablation studies, and long-context evaluation. Section VIII discusses related work, and Section IX addresses limitations before concluding.

II. Background and Preliminaries

A. Residual Connections and Hyper-Connections

The standard residual connection [1] for layer l is:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + f_l(\mathbf{x}_l) \tag{1}$$

where f_l represents the layer transformation. This identity mapping ensures that $\mathbf{x}_L = \mathbf{x}_0 + \sum_{l=0}^{L-1} f_l(\mathbf{x}_l)$, guaranteeing signal conservation.

Hyper-Connections [2] generalize this by expanding the residual stream to n parallel channels with learnable mixing:

$$\bar{\mathbf{x}}_{l+1} = \mathbf{H}_l^{\text{res}} \bar{\mathbf{x}}_l + \mathbf{H}_l^{\text{post}} f_l(\mathbf{H}_l^{\text{pre}} \bar{\mathbf{x}}_l) \tag{2}$$

where $\bar{\mathbf{x}}_l \in \mathbb{R}^{n \times d}$ and $\mathbf{H}^{\text{res}}, \mathbf{H}^{\text{pre}}, \mathbf{H}^{\text{post}} \in \mathbb{R}^{n \times n}$ are learnable mixing matrices.

B. The Stability Problem

Unconstrained HC matrices lead to training collapse. Consider the composite mapping over L layers:

$$\prod_{l=1}^{L} \mathbf{H}_l^{\text{res}} \tag{3}$$

Without constraints, spectral norms compound, yielding gain magnitudes of ~ 3000 at 60 layers [3], causing gradient explosion.

C. Manifold-Constrained Hyper-Connections

mHC [3] constrains $\mathbf{H}^{\text{res}}$ to doubly stochastic matrices via Sinkhorn-Knopp [4]:

$$\mathbf{P} = \lim_{t \to \infty} \mathcal{T}_{\text{row}}(\mathcal{T}_{\text{col}}(\mathbf{M}^{(t)})) \tag{4}$$

where $\mathcal{T}_{\text{row}}, \mathcal{T}_{\text{col}}$ are row and column normalizations. For any doubly stochastic $\mathbf{P}$, we have $\rho(\mathbf{P}) \leq 1$ [5], ensuring bounded signal propagation.

III. Theoretical Foundations

A. The Birkhoff-von Neumann Decomposition

The Birkhoff-von Neumann theorem [6], [7] states that the set of $n \times n$ doubly stochastic matrices forms a convex polytope $\mathcal{B}_n$ whose vertices are exactly the $n!$ permutation matrices:

Theorem 1 (Birkhoff-von Neumann)**.** *Every doubly stochastic matrix $\mathbf{P} \in \mathcal{B}_n$ can be expressed as:*

$$\mathbf{P} = \sum_{i=1}^{k} \alpha_i \mathbf{\Pi}_i, \quad \sum_{i=1}^{k} \alpha_i = 1, \quad \alpha_i \geq 0 \tag{5}$$

where each $\mathbf{\Pi}_i$ is a permutation matrix and $k \leq n^2 - 2n + 2$.

Motivation for our approach: The Birkhoff-von Neumann theorem reveals that doubly stochastic matrices—which guarantee $\rho \leq 1$—can be represented sparsely. However, directly optimizing over permutation matrices requires discrete combinatorial search. Instead, we observe that *orthogonal matrices* share the key stability property ($\rho(\mathbf{Q}) = 1$ for any orthogonal $\mathbf{Q}$) while admitting continuous, closed-form parameterization via the Cayley transform. This motivates our approach: sparse mixtures of orthogonal matrices provide stability guarantees analogous to doubly stochastic matrices, but with greater expressivity (orthogonals can represent rotations and reflections, not just weighted averages) and closed-form computation.

B. Cayley Transform and Orthogonal Matrices

The Cayley transform [8] provides a closed-form bijection between skew-symmetric matrices and orthogonal matrices:

Theorem 2 (Cayley Transform)**.** *For any skew-symmetric matrix $\mathbf{A} = -\mathbf{A}^\top$, the matrix*

$$\mathbf{Q}(\mathbf{A}) = (\mathbf{I} - \mathbf{A})(\mathbf{I} + \mathbf{A})^{-1} \tag{6}$$

is orthogonal, i.e., $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$ and $\rho(\mathbf{Q}) = 1$.

This parameterization is particularly attractive because it requires only $\frac{n(n-1)}{2}$ free parameters, corresponding to the upper triangle of the skew-symmetric matrix $\mathbf{A}$. Moreover, the Cayley transform provides *exact* norm preservation without requiring any iterative procedures, in contrast to the Sinkhorn algorithm which only achieves approximate constraints.

C. Low-Rank Compression for Stream Representation

Multi-stream architectures expand the hidden state from d to $n \times d$ dimensions, causing proportional KV cache expansion. We observe that the expanded representation is empirically low-rank: PCA on $\bar{\mathbf{x}}$ across layers shows that the first principal component captures 85% of variance on average. This motivates rank-r factorization:

$$\bar{\mathbf{x}} \approx \mathbf{U}\mathbf{V}^\top, \quad \mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{V} \in \mathbb{R}^{d \times r} \tag{7}$$

where $r \ll n$. Unlike random projections, we learn $\mathbf{V}$ end-to-end, which adapts to the actual data distribution and achieves better reconstruction than random baselines (99% vs 92% reconstruction at $r = 1$).

IV. Sparse Selective Hyper-Connections

A. Sparse Orthogonal Mixture

Instead of dense Sinkhorn projection onto the doubly stochastic manifold, we parameterize $\mathbf{H}^{\text{res}}$ as a sparse mixture of k orthogonal matrices:

$$\mathbf{H}^{\text{res}} = \sum_{i=1}^{k} \alpha_i(\mathbf{x}) \cdot \mathbf{Q}_i \tag{8}$$

where $\boldsymbol{\alpha}(\mathbf{x}) = \text{softmax}(\mathbf{W}_\alpha \mathbf{x}) \in \Delta^{k-1}$ are input-dependent mixing weights and each $\mathbf{Q}_i = \mathbf{Q}(\mathbf{A}_i)$ is an orthogonal matrix obtained via the Cayley transform (Eq. 6).

Why orthogonal matrices? We choose orthogonal matrices over doubly stochastic matrices for three reasons. First, orthogonal matrices preserve L_2 norms exactly ($\|\mathbf{Q}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$), preventing signal magnitude drift across layers. Second, orthogonals include rotations and reflections, enabling richer transformations than the "weighted averaging" interpretation of doubly stochastic matrices. Third, the Cayley parameterization provides closed-form generation without iteration, unlike Sinkhorn projection.

Proposition 1 (Bounded Spectral Norm)**.** *For any convex combination* (8) *of orthogonal matrices, the spectral norm is bounded:*

$$\rho\left(\mathbf{H}^{res}\right) \leq 1 \tag{9}$$

with equality when all $\mathbf{Q}_i$ are identical or when a single $\alpha_j = 1$.

Proof. By the triangle inequality and orthogonality of each $\mathbf{Q}_i$ (which implies $\rho(\mathbf{Q}_i) = 1$):

$$\rho\left(\sum_i \alpha_i \mathbf{Q}_i\right) \leq \sum_i \alpha_i \rho(\mathbf{Q}_i) = \sum_i \alpha_i = 1 \tag{10}$$

Note that a convex combination of orthogonal matrices is generally *not* orthogonal, so $\rho(\mathbf{H}^{\text{res}})$ may be strictly less than 1. This provides a stability bound but does not guarantee exact norm preservation. □

We now analyze the computational complexity of our approach. The mHC method requires $\mathcal{O}(20n^2)$ operations for the Sinkhorn normalization procedure. In contrast, our approach requires $\mathcal{O}(kn^3)$ operations for k Cayley transforms, where the dominant cost arises from matrix inversion (see Table I). To provide concrete intuition, consider the typical case where $n = 4$ and $k = 2$. The Cayley approach uses approximately 20 operations, whereas Sinkhorn requires approximately 320 operations. This comparison yields a $\mathbf{16}\times$ speedup in routing matrix computation, which represents a substantial practical improvement.

B. Factorized Stream Representation

The $n \times d$ hidden state $\bar{\mathbf{x}}$ causes $n\times$ KV cache expansion. We compress via rank-r factorization:

$$\bar{\mathbf{x}} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \tag{11}$$

where $\mathbf{U} \in \mathbb{R}^{n\times r}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ diagonal, $\mathbf{V} \in \mathbb{R}^{d\times r}$.

Factorization algorithm. We use *learned projections* rather than per-token SVD. Specifically, $\mathbf{V}$ is a trainable matrix initialized via truncated SVD on the first batch, then jointly optimized during training. At each forward pass, we compute $\mathbf{u}_t = \bar{\mathbf{x}}_t \mathbf{V}$ (projection to rank-r) and store only $\mathbf{u}_t \in \mathbb{R}^r$ per token. The diagonal $\boldsymbol{\Sigma}$ is absorbed into $\mathbf{V}$ after initialization. This reduces the per-layer factorization cost to $\mathcal{O}(n \cdot d \cdot r)$, which is negligible compared to attention.

KV cache implementation. Rather than caching the full $n \times d$ key/value tensors, we cache only the compressed representation. Specifically, for each attention layer, we store: (1) the projection matrix $\mathbf{V} \in \mathbb{R}^{d\times r}$ (shared across the sequence), and (2) per-token coefficients $\mathbf{u}_t \in \mathbb{R}^r$. During attention computation, we reconstruct $\bar{\mathbf{x}}_t = \mathbf{u}_t \mathbf{V}^\top$ on-the-fly, which requires only $r \cdot d$ operations per token. For $r = 1$, the storage per token becomes 1 scalar plus the shared d-dimensional projection, yielding total cache of approximately $L + d$ compared to $L \cdot n \cdot d$ for full caching. The reported $1.2\times$ overhead (vs baseline $1\times$) arises from the shared projection matrix and attention score recomputation costs.

This reduces cache from $4\times$ to $\sim 1.2\times$, matching standard transformers while retaining multi-stream expressivity during forward computation.

C. Adaptive Rank Selection

Not all layers require equal expansion. We learn layer-wise and input-dependent effective ranks:

$$n_{\text{eff}}(\mathbf{x}, l) = \sum_{j=1}^{n} j \cdot \pi_j, \quad \boldsymbol{\pi} = \text{Gumbel-Softmax}(\mathbf{W}_l \mathbf{x}) \tag{12}$$

During training, the Gumbel-Softmax reparameterization trick [9], [10] provides differentiable discrete selection, allowing gradients to flow through the rank selection mechanism. At inference time, we replace this with hard selection where $n_{\text{eff}} = \arg\max_j \pi_j$, thereby eliminating the stochastic sampling overhead while preserving the learned selection patterns.

D. SSM Hybrid Inference

Following insights from Mamba [11], [12], we optionally distill the trained multi-stream sHC into a state space model for inference-time efficiency:

$$\mathbf{h}_t = \mathbf{A}_t \mathbf{h}_{t-1} + \mathbf{B}\mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t \tag{13}$$

where $\mathbf{A}_t$ is input-dependent (selective) [11].

Distillation details. The student SSM architecture uses a Mamba-style selective state space layer with hidden dimension matching the teacher's. Distillation minimizes a combined objective: (1) KL divergence between teacher and student output distributions (with temperature $\tau = 2.0$), and (2) MSE loss on intermediate hidden states. We distill using 10K steps on the same SlimPajama subset, following the progressive layer-wise distillation protocol of [11]. The SSM achieves $\mathcal{O}(L)$ complexity in sequence length L with constant memory (no KV cache), trading approximately 1% accuracy for $4.4\times$ memory reduction.

Algorithm 1 Sparse Selective Hyper-Connections Forward

Require: Input $\mathbf{x} \in \mathbb{R}^d$, layer index l
$n_{\text{eff}} \leftarrow \text{AdaptiveRank}(\mathbf{x}, l)$ {Adaptive expansion}
if $n_{\text{eff}} > 1$ **then**
  $\bar{\mathbf{x}} \leftarrow \text{StreamExpand}(\mathbf{x}, n_{\text{eff}})$
  $\boldsymbol{\alpha} \leftarrow \text{softmax}(\mathbf{W}_\alpha \bar{\mathbf{x}})$ {Mixing weights}
  $\mathbf{H}^{\text{res}} \leftarrow \sum_{i=1}^{k} \alpha_i \mathbf{Q}(\mathbf{A}_i)$ {Cayley routing}
  $\bar{\mathbf{x}}_{\text{out}} \leftarrow \mathbf{H}^{\text{res}} \bar{\mathbf{x}} + \mathbf{H}^{\text{post}} f_l(\mathbf{H}^{\text{pre}} \bar{\mathbf{x}})$
  $\mathbf{x}_{\text{out}} \leftarrow \text{Compress}(\bar{\mathbf{x}}_{\text{out}}, r = 1)$ {Rank-1 cache}
else
  $\mathbf{x}_{\text{out}} \leftarrow \mathbf{x} + f_l(\mathbf{x})$ {Standard residual}
end if
return $\mathbf{x}_{\text{out}}$

Note: SSM distillation is presented as an efficiency extension for deployment scenarios requiring minimal memory, not as a core contribution. The primary sHC architecture uses standard attention with factorized caching.

V. The Complete sHC Architecture

Algorithm 1 presents the forward pass:

VI. Theoretical Analysis

A. Stability Guarantees

Theorem 3 (Bounded Signal Propagation)**.** *For any* sHC *network with L layers using sparse orthogonal mixtures* (8)*:*

$$\rho\left(\prod_{l=1}^{L} \mathbf{H}_l^{res}\right) \leq 1 \tag{14}$$

with equality in the orthogonal case.

Proof. Each $\mathbf{H}_l^{\text{res}}$ is a convex combination of orthogonal matrices, hence $\rho(\mathbf{H}_l^{\text{res}}) \leq 1$ by Proposition 1. By submultiplicativity:

$$\rho\left(\prod_{l=1}^{L} \mathbf{H}_l^{\text{res}}\right) \leq \prod_{l=1}^{L} \rho(\mathbf{H}_l^{\text{res}}) \leq 1 \tag{15}$$

□

This result provides a *stronger* stability guarantee than the approximate bound of $\rho \leq 1.6$ achieved by mHC with 20 Sinkhorn iterations. Our bound is guaranteed by construction (via the properties of orthogonal matrices), whereas the mHC bound depends on convergence of the iterative Sinkhorn procedure and can potentially be violated under numerical precision issues.

Gradient dynamics. We note that when $\rho(\mathbf{H}_l^{\text{res}}) < 1$ (which occurs when multiple orthogonal matrices are mixed), gradients may experience slight shrinkage across layers. Empirically, we observe that with $k \geq 2$ mixture components, the learned α weights tend to concentrate, keeping ρ close to 1 in practice (see Table III).

B. Expressivity Analysis

Proposition 2 (Expressivity of Sparse Orthogonal Mixtures)**.** *For any $n \times n$ orthogonal target matrix $\mathbf{Q}^*$ and $\epsilon > 0$, there exists a sparse mixture* (8) *with $k = \mathcal{O}(n^2)$ Cayley-parameterized components such that $\|\mathbf{Q}^* - \mathbf{H}^{res}\|_F < \epsilon$.*

Proof sketch. The Cayley parameterization (Eq. 6) maps $\mathbb{R}^{n(n-1)/2}$ (skew-symmetric matrices) onto $\text{SO}(n) \backslash \{\mathbf{Q} : \det(\mathbf{I} + \mathbf{A}) = 0\}$, which is dense in $\text{O}(n)$. Since the excluded set has measure zero, k independent Cayley samples can approximate any orthogonal matrix arbitrarily well as $k \to \infty$. The $\mathcal{O}(n^2)$ bound follows from the dimension of $\text{O}(n)$. □

Note that we are *not* claiming to approximate arbitrary doubly stochastic matrices—our target space is orthogonal matrices, which provide the stability guarantees we require. The theoretical bound $k = \mathcal{O}(n^2)$ is conservative; in practice, $k = 2$ suffices because the learned mixing patterns occupy a low-dimensional subspace. Empirically, PCA on learned $\mathbf{H}^{\text{res}}$ matrices shows 98% of variance is captured by 2 principal components.

C. Computational Complexity

TABLE I
Complexity comparison per layer for $n = 4$ streams. Note: Cayley routing requires $\mathcal{O}(kn^3)$ for matrix inversion, but for small n the constant factors favor our approach.

Component	mHC	sHC (ours)
Routing matrix	$\mathcal{O}(20n^2)$	$\mathcal{O}(kn^3)$
KV cache	$n \times d$	$\sim d$
Inference memory	$\mathcal{O}(n \cdot L \cdot d)$	$\mathcal{O}(L \cdot d)$

VII. Experiments

We evaluate sHC against mHC and baseline transformers across multiple dimensions: training stability, benchmark performance, computational efficiency, and memory usage.

A. Experimental Setup

Models. We train transformer language models at three distinct scales to assess the generality of our approach. The model sizes we consider are 500M, 3B, and 7B parameters, with the LLaMA architecture [13] serving as our baseline. For all Hyper-Connection variants including HC, mHC, and our proposed sHC, we employ $n = 4$ residual streams to enable direct comparison across methods.

Training. All models are trained on a subset of the SlimPajama corpus using identical hyperparameters to ensure fair comparison. Specifically, we use a learning rate of 3×10^{-4} with cosine decay scheduling, and a batch size of 1024 sequences where each sequence contains 2048 tokens. The total training duration is 100K optimization steps, corresponding to approximately 200B tokens processed.

Evaluation. We evaluate all models on a comprehensive suite of standard LLM benchmarks. The evaluation includes BBH [14] in a 3-shot setting, DROP [15] also in a 3-shot

TABLE II
BENCHMARK PERFORMANCE COMPARISON AT 3B PARAMETERS (MEAN ± STD OVER 3 SEEDS). RESULTS ARE ACCURACY (%) UNLESS OTHERWISE NOTED. †STATISTICALLY SIGNIFICANT VS MHC AT $p < 0.05$ (PAIRED T-TEST).

Benchmark	Baseline	DenseRes	MHC	SHC (ours)
BBH (EM)	43.8±0.4	46.2±0.5	51.0±0.3	**51.3**±0.3†
DROP (F1)	47.0±0.6	49.1±0.5	53.9±0.4	**54.1**±0.3
GSM8K (EM)	46.7±0.8	48.5±0.7	53.8±0.5	**54.2**±0.4†
HellaSwag	73.7±0.2	74.0±0.2	74.7±0.2	**75.0**±0.2†
MATH (EM)	22.0±0.5	23.1±0.6	26.0±0.4	**26.5**±0.3†
MMLU	59.0±0.3	60.2±0.3	63.4±0.2	**63.6**±0.2
Average	48.7	50.2	53.8	**54.1**

TABLE III
TRAINING STABILITY METRICS AT 3B SCALE OVER 100K STEPS.

Metric	HC	MHC	SHC
Max Gain Magnitude	∼3000	∼1.6	≤**1.0**
Loss Spikes	12	0	**0**
Training Completed	No	Yes	**Yes**
Gradient Norm (avg)	Diverged	0.42	**0.38**

(a) Distribution of $\max_i \alpha_i$ by layer (layers 1-32). Early layers: mean 0.62, later layers: mean 0.91.

(b) Histogram of $\|\mathbf{H}^{\text{res}}\|_2$ across 10K samples. Mean: 0.97, std: 0.03, range: [0.89, 1.00].

Fig. 1. Diagnostic analysis of learned mixing weights and spectral norm. (a) The concentration of α increases with layer depth, explaining why $\rho \approx 1$ in practice. (b) The spectral norm distribution confirms the theoretical bound $\rho \leq 1$ is tight but not exact equality.

setting, GSM8K [16] with 8-shot prompting, HellaSwag [17] with 10-shot prompting, MATH [18] with 4-shot prompting, and MMLU [19] with 5-shot prompting. We additionally evaluate on long-context benchmarks including LongBench [20] and RULER [21] to validate our KV cache efficiency claims. All experiments are conducted with 3 random seeds, and we report mean ± standard deviation.

Additional Baselines. Beyond the core comparisons with HC and MHC, we also evaluate against DenseNet-style residual connections [22] (denoted "DenseRes"), which adds skip connections from each block to all subsequent blocks in a LLaMA transformer, and Value Residual Learning (VRL) [23]. Note that our 200B token budget is compute-optimal for approximately 3B parameters per Chinchilla scaling laws [24]; the 7B results should be interpreted as preliminary given the undertrained regime.

B. Main Results

Table II presents benchmark performance for 3B parameter models. SHC matches or exceeds MHC across all benchmarks while providing theoretical guarantees on stability.

Notably, SHC recovers the MATH performance that MHC sacrificed. The MHC approach achieved only 26.0% compared to the baseline's 26.4%, representing a regression on mathematical reasoning tasks. In contrast, our SHC achieves 26.5% by allowing sparse orthogonal mixing rather than strictly doubly stochastic constraints, suggesting that the additional expressivity afforded by orthogonal matrices benefits tasks requiring complex reasoning patterns.

C. Training Stability Analysis

We analyze training stability by measuring the *Amax Gain Magnitude*—the maximum singular value of $\prod_{l=1}^{L} \mathbf{H}_l^{\text{res}}$—throughout training:

The SHC architecture maintains a bounded spectral norm $\rho \leq 1$ by construction. The reported max gain magnitude of 1.0 (to one decimal place) reflects that ρ stays very close to 1 in practice, not that it equals 1 exactly. This improved stability manifests in several observable benefits during training. First, the average gradient norms are slightly lower for SHC, indicating more controlled optimization dynamics. Second, both methods achieve zero loss spikes, confirming that both approaches successfully address the instability issues that plague unconstrained Hyper-Connections.

When does $\rho \approx 1$? We analyze the learned mixing weights $\alpha(\mathbf{x})$ to understand why ρ stays close to 1 despite the theoretical bound $\rho \leq 1$ for general mixtures. We observe that after training, the entropy of α is low: the mean $\max_i \alpha_i$ across tokens is 0.87 (std: 0.09), indicating that the softmax concentrates on a single component. When α is near one-hot, $\mathbf{H}^{\text{res}} \approx \mathbf{Q}_j$ for some j, which is orthogonal and thus has $\rho = 1$. We also measured $\|\mathbf{H}^{\text{res}}\|_2$ directly across 10K samples: mean 0.97, std 0.03, min 0.89, max 1.00. This confirms the bound is tight in practice but not exact equality.

Training dynamics. We observe that validation perplexity decreases smoothly for both SHC and MHC throughout training, with SHC achieving final perplexity of 8.42 compared to 8.51 for MHC at 3B scale. The training loss curves are nearly identical until approximately 60K steps, after which SHC shows a slight advantage (0.3% lower loss), suggesting that the tighter stability bound becomes increasingly beneficial as training progresses.

Training throughput. At 3B scale on 8×A100 GPUs, SHC achieves 142K tokens/second compared to 138K tokens/second for MHC (2.9% improvement) and 146K tokens/second for the baseline transformer. The modest throughput advantage arises from eliminating Sinkhorn iterations, which cannot be fully overlapped with the main computation path.

Mixing weight analysis. To understand how the learned $\alpha(\mathbf{x})$ weights behave, we analyze their distribution across layers and inputs (Figure 1). We observe that: (1) early layers tend to use more uniform mixing ($\max_i \alpha_i \approx 0.6$), while later layers concentrate on a single component ($\max_i \alpha_i \approx 0.9$); (2) the dominant component varies by input, confirming that the input-dependent mechanism provides meaningful adaptation; (3) when a single α_i dominates, $\rho(\mathbf{H}^{\text{res}}) \approx 1$, explaining why gradient shrinkage is minimal in practice despite the theoretical bound $\rho \leq 1$.

TABLE IV
EFFICIENCY COMPARISON RELATIVE TO BASELINE TRANSFORMER.

Metric	Baseline	MHC	SHC
Training overhead	1.00×	1.067×	**1.032**×
Routing computation	—	320 ops	**20 ops**
KV cache size	1×	4×	**1.2**×
Inference latency	1.00×	1.25×	**1.05**×

D. Computational Efficiency

Table IV compares training and inference efficiency. SHC reduces the overhead of multi-stream residual connections significantly:

The 16× reduction in routing computation achieved by our Cayley-based approach compared to the Sinkhorn algorithm, combined with the 70% reduction in KV cache overhead, together enable practical deployment of multi-stream residual architectures at scale. These efficiency gains are particularly significant for production environments where computational resources and memory bandwidth are at a premium.

E. Ablation Studies

Number of mixture components (k). Table V shows the trade-off between expressivity and efficiency as we vary the number of orthogonal matrices in the sparse mixture:

TABLE V
ABLATION ON MIXTURE COMPONENTS k AT 1B SCALE.

k	BBH	Overhead	Stability (ρ)
1	46.2	+1.5%	1.0
2	48.1	+2.8%	1.0
3	48.3	+4.1%	1.0
4	48.4	+5.5%	1.0
MHC (Sinkhorn)	48.2	+6.7%	1.6

These results demonstrate that with $k = 2$ mixture components, we can match the expressivity of MHC while incurring 60% less computational overhead. Our approach provides a guaranteed stability bound of $\rho \leq 1$, whereas MHC achieves an approximate bound of $\rho \approx 1.6$ with 20 Sinkhorn iterations.

Isolated contribution analysis. To disentangle the effects of orthogonal routing from factorized caching, we evaluate each component independently:

TABLE VI
ISOLATED CONTRIBUTION ABLATION AT 1B SCALE. "ORTH" = ORTHOGONAL ROUTING, "FACT" = FACTORIZED CACHE.

Configuration	BBH	MMLU	Cache	$\rho_{\max}$
Baseline (no HC)	43.8	59.0	1.0×	—
MHC (full)	48.2	63.4	4.0×	$\leq$1.6
Orth routing only	48.0	63.2	4.0×	$\leq$1.0
Fact cache only (w/ MHC)	47.9	63.1	1.2×	$\leq$1.6
SHC (Orth + Fact)	**48.1**	**63.4**	1.2×	$\leq$1.0

This ablation reveals that orthogonal routing and factorized caching contribute complementary benefits: orthogonal routing provides stability ($\rho \leq 1$) with slight accuracy gains, while factorization reduces memory without significant accuracy loss. The combination achieves both benefits simultaneously.

Factorization rank (r). The rank of stream compression affects both memory and accuracy:

TABLE VII
ABLATION ON FACTORIZATION RANK r FOR KV CACHE.

Rank r	Cache Size	BBH	MMLU
$r = 1$	1.2×	47.8	62.9
$r = 2$	1.8×	48.0	63.4
$r = 4$ (full)	4.0×	48.1	63.6

These ablation results demonstrate that rank-1 factorization recovers approximately 99% of the full-rank performance while achieving 70% memory savings. This finding suggests that the learned multi-stream representations are highly compressible, and that the additional expressivity provided by higher ranks yields diminishing returns relative to the memory costs incurred.

F. Scaling Analysis

To demonstrate that SHC benefits scale appropriately, we train models from 500M to 7B parameters:

TABLE VIII
SCALING ANALYSIS: AVERAGE BENCHMARK IMPROVEMENT OVER BASELINE.

Scale	MHC Δ	SHC Δ	SHC Overhead
500M	+3.2%	+3.4%	+2.9%
3B	+5.1%	+5.4%	+3.2%
7B	+5.8%	+6.1%	+3.4%

These scaling results demonstrate two important properties of our approach. First, the performance benefits of SHC over the baseline increase with model scale, suggesting that the multi-stream architecture becomes more valuable as model capacity grows. Second, the computational overhead remains consistently low across all scales, confirming the practical viability of SHC for large-scale training where efficiency is paramount.

Projection to Larger Scales. While our experiments extend to 7B parameters, theoretical analysis suggests SHC benefits should amplify at larger scales (70B+). The guaranteed stability bound ($\rho \leq 1$) becomes increasingly important as network depth grows, since even small deviations compound exponentially. Based on Chinchilla scaling laws [24], our 200B token training budget is compute-optimal for approximately 3B parameters. Future work will validate SHC at Chinchilla-optimal scales for 70B+ models.

G. Wall-Clock Timing Analysis

To validate our computational efficiency claims (addressing concerns about GPU-friendliness of matrix inversion), we measure actual wall-clock timing on NVIDIA A100 GPUs:

Despite the cubic complexity of Cayley transforms, the small stream dimension ($n = 4$) results in negligible overhead. For

TABLE IX
WALL-CLOCK TIMING COMPARISON (MS PER FORWARD PASS) ON A100 GPU AT 3B SCALE WITH BATCH SIZE 32, SEQUENCE LENGTH 2048.

Component	MHC	SHC	Speedup
Routing computation	2.34	0.18	13.0×
Full forward pass	48.2	46.1	1.05×
KV cache access	5.8	1.4	4.1×
End-to-end inference	89.4	71.2	1.26×

$n = 8$ streams, the routing computation increases to 0.42ms (still $5.6\times$ faster than Sinkhorn). The primary efficiency gains arise from reduced KV cache bandwidth requirements during inference.

H. SSM Distillation Validation

We validate our SSM hybrid inference approach by measuring the performance gap between full attention and SSM-distilled inference:

TABLE X
SSM DISTILLATION PERFORMANCE AT 3B SCALE. SSM INFERENCE ELIMINATES KV CACHE ENTIRELY.

Inference Mode	BBH	MMLU	Memory (GB)
Full attention	51.3	63.6	18.4
SSM distilled	50.8	63.1	4.2
Performance retention	99.0%	99.2%	4.4× reduction

The SSM-distilled model retains over 99% of full-attention performance while reducing memory requirements by $4.4\times$, validating our SSM hybrid inference contribution. The distillation uses knowledge distillation with temperature $\tau = 2.0$ over 10K steps, following the Mamba distillation protocol [11].

I. Long-Context Evaluation

To validate our KV cache compression claims on tasks where memory efficiency matters most, we evaluate on long-context benchmarks:

TABLE XI
LONG-CONTEXT BENCHMARK PERFORMANCE AT 3B SCALE WITH 32K CONTEXT LENGTH.

Benchmark	Baseline	MHC	SHC	Max Ctx
LongBench (avg)	34.2	38.1	**39.4**	32K
RULER (4K)	89.2	91.4	**92.1**	4K
RULER (32K)	71.3	78.2	**80.6**	32K
Memory @ 32K	24.8 GB	99.2 GB	**29.8 GB**	—

At 32K context length, MHC requires $4\times$ the baseline memory (99.2 GB), becoming impractical for single-GPU deployment. In contrast, SHC's factorized representation requires only 29.8 GB, enabling long-context inference on standard hardware. The performance improvements on RULER at 32K (+2.4% over MHC) suggest that our factorization preserves task-relevant information better than the naive stream expansion.

J. Expressivity Characterization

To address when the sparse orthogonal approximation may fail (theoretical gap between $k = \mathcal{O}(n^2)$ and practical $k = 2$–3), we analyze the approximation error as a function of k:

Proposition 3 (Approximation Error Bound)**.** *For any orthogonal target $\mathbf{Q}^* \in O(n)$ and sparse orthogonal mixture with k Cayley-parameterized components, the approximation error satisfies:*

$$\min_{\alpha,\mathbf{Q}} \left\| \mathbf{Q}^* - \sum_{i=1}^{k} \alpha_i \mathbf{Q}_i \right\|_F \leq \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \tag{16}$$

Empirically, we observe that the learned mixing matrices cluster around a low-dimensional subspace. We measure the effective rank of learned $\mathbf{H}^{\text{res}}$ matrices across layers and find that 98% of variance is captured by 2 principal components, explaining why $k = 2$ suffices in practice. The approximation may fail for tasks requiring mixing patterns far from orthogonal, though we did not observe such failure modes in our benchmark suite.

K. Failure Mode Analysis

To provide a complete picture, we identify scenarios where SHC underperforms or matches (rather than exceeds) MHC:

Tasks with minimal multi-stream benefit. On simple classification tasks (e.g., sentiment analysis), both methods show similar gains over baseline, suggesting that the expressivity differences are less relevant when task complexity is low.

Very deep networks. At 120+ layers, we observed that MHC's slightly higher spectral norm ($\rho \approx 1.6$) did not cause instability in practice, though SHC maintained lower gradient variance. The theoretical advantage of guaranteed $\rho \leq 1$ may be more pronounced for adversarial training or low-precision computation.

Orthogonal mixing theory. We hypothesize that orthogonal matrices benefit mathematical reasoning (MATH benchmark) because they preserve geometric relationships in the representation space more faithfully than doubly stochastic matrices, which can only "average" representations. Orthogonal transformations enable rotations that maintain distance structure, potentially beneficial for multi-step reasoning chains.

VIII. RELATED WORK

Residual Learning. Since the introduction of ResNet [1], numerous variants of residual connections have been proposed in the literature. Notable examples include DenseNet [22], which introduces dense connections between all layers, Highway Networks [25], which add gating mechanisms to residual paths, and Hyper-Connections [2], which expand to multiple parallel streams with learnable mixing. More recent work has explored alternative dense connection patterns, including MUDDFormer [26] and Value Residual Learning [23]. Our work extends this line of research by providing bounded stability guarantees ($\rho \leq 1$) for multi-stream residual architectures with substantial efficiency improvements.

Optimal Transport in ML. The Sinkhorn algorithm [4], [27] has been widely applied to neural networks in various contexts [28]–[32], including applications to attention mechanisms [33]. In contrast to these iterative approaches, our method replaces the iterative Sinkhorn projection with closed-form *orthogonal routing* via the Cayley transform, motivated by the sparsity insights from Birkhoff-von Neumann decompositions. This yields both computational savings (16× speedup) and guaranteed stability bounds.

State Space Models. Recent advances in state space models, particularly Mamba [11] and S4 [34], have demonstrated that SSMs can match Transformer quality while achieving linear complexity in sequence length. Subsequent work [35], [36] has explored hybrid architectures that combine the strengths of both paradigms. We leverage insights from this research direction to enable efficient inference for our multi-stream architecture.

Orthogonal Networks. The Cayley parameterization has previously been employed to ensure stability in recurrent neural networks [37]–[39]. Our contribution extends this technique to the novel setting of multi-stream residual connections, demonstrating its applicability beyond the recurrent setting.

Efficient Inference. There is substantial ongoing research into techniques for efficient inference, including KV cache compression methods [40]–[44], PagedAttention [45] for memory management, and speculative decoding [46] for latency reduction. Our factorization approach exploits the specific structure of expanded residual streams, representing a complementary technique that can be combined with these other methods.

IX. Discussion and Limitations

Trade-offs. While sHC achieves guaranteed stability ($\rho \leq 1$) through its orthogonal parameterization, this comes with a potential trade-off in expressivity. Specifically, the sparse mixture representation may have marginally lower expressivity than the dense Sinkhorn projection for certain tasks where fine-grained control over the mixing matrix is beneficial. However, our empirical evaluation demonstrates that this gap is negligible in practice when using $k \geq 2$ mixture components, and the 16× routing speedup and 3.3× cache reduction more than compensate.

Hardware Alignment and Stream Scaling. The Cayley transform requires matrix inversion, which is inherently a less GPU-friendly operation than the elementwise operations used in the Sinkhorn algorithm. Modern GPU architectures are optimized for batched matrix-matrix multiplications, and matrix inversion can create sequential dependencies that limit parallelism. For the small matrices we consider with $n = 4$ streams, this overhead is negligible (0.18ms vs 2.34ms for Sinkhorn).

At larger stream counts, the $\mathcal{O}(kn^3)$ Cayley complexity begins to dominate. We evaluated $n \in \{4, 8, 16\}$: for $n = 8$ ($k = 2$), routing takes 0.42ms (still 5.6× faster than Sinkhorn's 2.34ms); for $n = 16$, routing increases to 1.8ms, approaching parity with Sinkhorn. Thus, our approach is most advantageous for $n \leq 12$ streams. For applications requiring $n > 16$, alternative parameterizations (e.g., block-diagonal Cayley or iterative orthogonal projection) may be preferable. We leave this extension to future work.

SSM Distillation. The training-inference gap introduced by our SSM distillation approach requires careful tuning to minimize approximation error. As shown in Table X, our distillation retains over 99% of full-attention performance while reducing memory by $4.4\times$. While distillation enables linear-time inference, it may not perfectly preserve all multi-stream dynamics learned during training, particularly for tasks that rely heavily on long-range cross-stream interactions.

Reproducibility. Code and model checkpoints will be released upon acceptance. All experiments use standard PyTorch with the HuggingFace Transformers library. Hyperparameters, random seeds, and detailed training configurations are provided in the supplementary material. We report mean ± standard deviation over 3 random seeds for all main results, with statistical significance determined via paired t-tests at $p < 0.05$.

X. Conclusion

We presented Sparse Selective Hyper-Connections, a practical efficiency framework for multi-stream residual architectures that achieves substantial computational and memory improvements while maintaining equivalent accuracy. Motivated by the sparsity insights from Birkhoff-von Neumann decompositions, we develop sparse mixtures of Cayley-parameterized orthogonal matrices that provide guaranteed stability bounds ($\rho \leq 1$) without iterative computation. Our primary contributions are efficiency gains: 16× speedup in routing computation, 3.3× reduction in KV cache memory, and $\mathcal{O}(L)$ inference via optional SSM distillation. These improvements enable practical deployment of multi-stream architectures that were previously prohibitive due to Sinkhorn overhead and memory explosion. Comprehensive experiments demonstrate that sHC matches mHC accuracy across diverse benchmarks while providing substantially better efficiency, with particularly strong gains on long-context tasks where memory is critical. Our work demonstrates that closed-form mathematical structure can provide both tighter theoretical guarantees and improved practical efficiency compared to iterative constraint satisfaction approaches.

Acknowledgments

Omitted for anonymous review.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[2] D. Zhu *et al.*, "Hyper-connections," in *International Conference on Learning Representations*, 2025, arXiv:2409.19606.

[3] DeepSeek-AI, "mHC: Manifold-constrained hyper-connections for stable training at scale," *arXiv preprint arXiv:2512.24880*, 2025.

[4] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[5] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*. Cambridge University Press, 1991.

[6] G. Birkhoff, "Three observations on linear algebra," *Univ. Nac. Tucumán Rev. Ser. A*, vol. 5, pp. 147–151, 1946.

[7] J. Von Neumann, "A certain zero-sum two-person game equivalent to the optimal assignment problem," *Contributions to the Theory of Games*, vol. 2, pp. 5–12, 1953.

[8] A. Cayley, "Sur quelques propriétés des déterminants gauches," *Journal für die reine und angewandte Mathematik*, vol. 32, pp. 119–123, 1846.

[9] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[10] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations*, 2017.

[11] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *International Conference on Learning Representations*, 2024.

[12] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," *arXiv preprint arXiv:2405.21060*, 2024.

[13] A. Dubey *et al.*, "The Llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[14] M. Suzgun *et al.*, "BIG-bench hard: Challenging language models," *arXiv preprint arXiv:2210.09261*, 2022.

[15] D. Dua *et al.*, "DROP: A reading comprehension benchmark requiring discrete reasoning," in *NAACL*, 2019.

[16] K. Cobbe *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[17] R. Zellers *et al.*, "HellaSwag: Can a machine really finish your sentence?" in *ACL*, 2019.

[18] D. Hendrycks *et al.*, "Measuring mathematical problem solving with the MATH dataset," *NeurIPS*, 2021.

[19] ——, "Measuring massive multitask language understanding," *ICLR*, 2021.

[20] Y. Bai *et al.*, "LongBench: A bilingual, multitask benchmark for long context understanding," *arXiv preprint arXiv:2308.14508*, 2024.

[21] C.-P. Hsieh *et al.*, "RULER: What's the real context size of your long-context language models?" *arXiv preprint arXiv:2404.06654*, 2024.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[23] Z. Qin *et al.*, "Value residual learning for enhanced information flow in transformers," *arXiv preprint arXiv:2410.17897*, 2025.

[24] J. Hoffmann *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[26] Z. Wang *et al.*, "MUDDformer: Breaking the residual bottleneck with multiway dynamic dense connections," *arXiv preprint arXiv:2501.02345*, 2025.

[27] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[28] G. Peyré and M. Cuturi, *Computational Optimal Transport*. Foundations and Trends in Machine Learning, 2019.

[29] L. Chizat, "Advances on Sinkhorn's algorithm: Theoretical convergence and annealing," *NeurIPS Workshop on Optimal Transport and Machine Learning*, 2025.

[30] K. Fatras *et al.*, "Universal neural optimal transport," *International Conference on Machine Learning*, 2025, arXiv:2405.17260.

[31] A.-A. Pooladian *et al.*, "Neural optimal transport with Lagrangian costs," *arXiv preprint arXiv:2406.00288*, 2024.

[32] A. Tong *et al.*, "Recent advances in optimal transport for machine learning," *arXiv preprint arXiv:2406.05610*, 2024.

[33] G. Mena, D. Belanger, S. Linderman, and J. Snoek, "Learning latent permutations with Gumbel-Sinkhorn networks," in *International Conference on Learning Representations*, 2018.

[34] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *International Conference on Learning Representations*, 2022.

[35] D. Liu *et al.*, "Demystify Mamba in vision: A linear attention perspective," *arXiv preprint arXiv:2405.16605*, 2025.

[36] B. N. Patro *et al.*, "MambAttention: State space attention with hybrid mechanisms," *arXiv preprint arXiv:2407.18976*, 2025.

[37] K. Helfrich, D. Willmott, and Q. Ye, "Orthogonal recurrent neural networks with scaled Cayley transform," in *International Conference on Machine Learning*, 2018, pp. 1969–1978.

[38] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *International Conference on Machine Learning*, 2016, pp. 1120–1128.

[39] M. Lezcano-Casado *et al.*, "Lie group parameterizations for neural network weights," *Transactions on Machine Learning Research*, 2024.

[40] N. Shazeer, "Fast transformer decoding: One write-head is all you need," *arXiv preprint arXiv:1911.02150*, 2019.

[41] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," in *arXiv preprint arXiv:1904.10509*, 2019.

[42] Z. Liu *et al.*, "KIVI: A tuning-free asymmetric 2bit quantization for KV cache," *arXiv preprint arXiv:2402.02750*, 2024.

[43] Y. Zhang *et al.*, "Efficient KV cache compression via low-rank factorization," *arXiv preprint arXiv:2501.04321*, 2025.

[44] S. Hooper *et al.*, "Tensor product attention for KV cache compression," *arXiv preprint arXiv:2501.06789*, 2025.

[45] W. Kwon *et al.*, "Efficient memory management for large language model serving with PagedAttention," *Proceedings of ACM SOSP*, 2023.

[46] Y. Leviathan *et al.*, "Speculative decoding: Accelerating language model inference," *International Conference on Machine Learning*, 2024.