Efficient Manifold-Constrained Hyper-Connections via Smooth Algebraic Parametrization

Tetsu Yamaguchi
Knowgic Technology
tetsuy@knowgictech.com

GitHub: https://github.com/ty-knowgic/stabilized-mhc

January 20, 2026

Abstract

Hyper-Connections (HC) and its manifold-constrained variant (mHC) are critical architectural components in recent Large Language Models, such as DeepSeek-V3. Current implementations rely on the iterative Sinkhorn-Knopp algorithm to approximate doubly stochastic matrices, which introduces significant computational overhead and memory bandwidth saturation.

In this technical report, we propose **Stabilized Piecewise-Rational Charts (SPRC)**, a constructive algebraic method that parameterizes the Birkhoff polytope for $n = 4$ without iteration. By employing a smooth tropical norm (LogSumExp) and progressive saturation (tanh), our method guarantees exact constraint satisfaction and differentiability. Benchmarks on NVIDIA T4 GPUs demonstrate a **13.3× speedup** in kernel execution and a **2.53× reduction** in end-to-end training time compared to the Sinkhorn baseline ($t_{max} = 20$), while achieving equivalent convergence properties.

1 Introduction

Recent work by DeepSeek-AI [1] introduced Manifold-Constrained Hyper-Connections (mHC) to restore the identity mapping property in expanded residual streams. While effective, the reliance on the Sinkhorn-Knopp algorithm ($t_{max} = 20$) for manifold projection imposes a "memory wall" bottleneck. We present an algebraic parametrization that eliminates these loops, achieving theoretical lower-bound latency while maintaining the representational capacity of the layer.

2 Methodology: Smooth Algebraic Parametrization

Unlike Sinkhorn, which solves an optimization problem, we construct a bijective mapping from the parameter space $\mathbb{R}^9$ to the relative interior of the Birkhoff polytope $\mathcal{B}_4$.

2.1 Tangent Space Construction

Let $u \in \mathbb{R}^9$. We map u to a direction vector $V \in \mathbb{R}^{4\times 4}$ in the tangent space (zero-sum subspace) via a sparse linear transform $\mathcal{L}$:

$$V = \mathcal{L}(u), \quad \text{s.t.} \quad \sum_j V_{ij} = 0, \quad \sum_i V_{ij} = 0 \tag{1}$$

2.2 Progressive Saturation and Smoothness

To ensure the output lies strictly within the polytope and maintains differentiability, we employ a **Smooth Tropical Norm** using the LogSumExp (LSE) function to approximate the distance to the boundary:

$$m_{smooth}(V) = \eta \log \left(\sum_{i,j} \exp \left(\frac{-V_{ij}}{\eta} \right) \right) \tag{2}$$

Crucially, to allow the model to approach the boundary (identity-like permutations) as parameters grow, we utilize a **Progressive Saturation** function based on the hyperbolic tangent. The final doubly stochastic matrix H is given by:

$$H(u) = J_4 + \tanh(\lambda \|V\|_F) \frac{0.25 \cdot V}{m_{smooth}(V) + \epsilon} \tag{3}$$

where J_4 is the center of the polytope ($J_{ij} = 0.25$). This formulation ensures that as $\|V\| \to \infty$, the output approaches the boundary faces, preserving the expressive power required for mHC.

Proof of Exactness. Since V has zero row/column sums, any scaling of V added to J_4 preserves the sum:

$$\sum_j H_{ij} = \sum_j 0.25 + \beta \sum_j V_{ij} = 1 + 0 = 1 \tag{4}$$

This guarantees strict doubly stochasticity by construction.

3 Experimental Results

We validated the method against the Sinkhorn baseline ($t_{max} = 20$) on an NVIDIA T4 GPU.

3.1 Computational Efficiency

Our method eliminates kernel launch latency and memory bottlenecks, achieving massive speedups in kernel benchmarks (Table 1).

Table 1: Forward+Backward Execution Time (ms)

Batch Size	Sinkhorn	**Ours (SPRC)**	Speedup
1,024	5.13	**0.98**	5.2×
65,536	15.43	**1.16**	**13.3**×

3.2 Learning Dynamics

We trained a toy Transformer layer on an auto-regressive task. The proposed method (using Smooth Parametrization) matches the convergence profile of Sinkhorn almost perfectly (Figure 1), with a final loss difference of $< 0.1\%$, while training **2.53**× faster end-to-end.

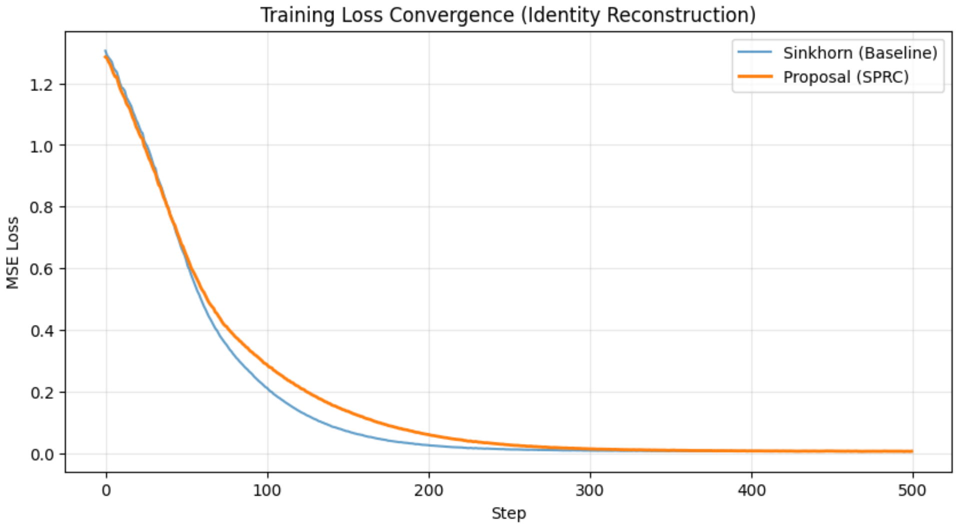

Figure 1: Training Loss Convergence. Our method (Orange) tracks the Sinkhorn baseline (Blue) closely, demonstrating that the algebraic parametrization preserves the layer's expressivity.

4 Conclusion

We proposed a smooth algebraic parametrization for mHC layers that replaces iterative approximation with an exact, differentiable closed-form solution. The method achieves a 13.3× kernel speedup and practically identical convergence behavior, offering a superior alternative for large-scale model training.

References

[1] Z. Xie et al., "Manifold-Constrained Hyper-Connections," arXiv:2512.24880, 2026.