Hypersphere and Stiefel Manifold Constraints for Custom Optimizers For Different LLM Parts

Vuk Rosić[1,2]
[1]Open Superintelligence Lab
[2]Óbuda University

December 8, 2025

Abstract

We investigate the application of manifold constraints to different parameter groups within transformer language models. By constraining embedding vectors to a hypersphere and applying the Muon optimizer to attention and feed-forward layers, we achieve significantly improved training efficiency. On a 42M parameter GPT model, the hypersphere constraint on embeddings reduces validation loss by 5.4% and perplexity by 37.7% compared to baseline, while adding minimal computational overhead. Code is available at `https://github.com/vukrosic/custom-optimizer-research`.

1 Introduction

Training large neural networks requires keeping tensors healthy - preventing weights, activations, and gradients from growing too large or too small. While normalization is commonplace for activations (layer norm) and gradient updates (Muon optimizer), it is less commonly applied to weight matrices themselves.

We explore *modular manifolds*: the idea that different network components may benefit from different geometric constraints. Our approach treats the network as a composition of modules, each with its own:

1. **Forward function** (how it transforms inputs)
2. **Manifold constraint** (what surface the weights lie on)
3. **Distance norm** (how to measure update sizes)

2 Manifold Constraints

2.1 Hypersphere for Embeddings

For embedding vectors, we constrain each row to lie on a hypersphere of unit radius. The update rule projects back to the manifold after each step:

$$w \leftarrow \frac{w}{\|w\|_2}$$

This prevents embedding norm explosion/collapse and focuses optimization on directional changes.

2.2 Stiefel Manifold for Weight Matrices

The Stiefel manifold constrains weight matrices to have orthonormal columns (all singular values = 1):

$$\text{Stiefel}(m, n) := \{W \in \mathbb{R}^{m \times n} \mid W^T W = I_n\}$$

We apply Newton-Schulz iteration to project weights back to this manifold, keeping the condition number bounded.

2.3 Muon vs. Stiefel Manifold

It is important to distinguish between the **Muon optimizer** and the **Stiefel constraint**, as both utilize Newton-Schulz iteration for orthogonalization but target different objects:

1. **Muon (Optimizer):** Operates on the **update/gradient** matrix (G). It orthogonalizes the *step* taken at each iteration to ensure effective spectral learning, but does not constrain the final weight matrix. The weights themselves are free to drift off the manifold.
2. **Stiefel (Constraint):** Operates on the **weight** matrix (W). It forces the parameters themselves to remain on the manifold ($W^T W = I$) after every step, strictly constraining the hypothesis space.

While Muon improves the *trajectory* of optimization, the Stiefel constraint restricts the *solution space*. They can be combined (as in our `manifold_muon` experiment) to perform spectrally normalized updates while maintaining strictly orthogonal weights.

3 Experimental Setup

Model: 42M parameter GPT (4 layers, 8 heads, 512 hidden size)
Dataset: SmolLM corpus, 30K sequences of length 512
Training: 20 steps per experiment, cosine LR schedule

We compared six configurations:

Experiment	Embeddings	Attention/FFN
`adamw_only`	AdamW	AdamW
`baseline`	AdamW	Muon
`sphere_constraint`	AdamW + Sphere	Muon
`stiefel_all`	AdamW	AdamW + Stiefel
`manifold_muon`	AdamW	Muon + Stiefel
`full_manifold`	AdamW + Sphere	Muon + Stiefel

4 Results

Experiment	Val Loss	Perplexity	Δ vs Baseline
`sphere_constraint`	**8.27**	**3,914**	**-5.4%**
`full_manifold`	8.72	6,096	-0.3%
`baseline`	8.75	6,281	-
`adamw_only`	8.79	6,598	+0.5%
`stiefel_all`	8.94	7,617	+2.2%
`manifold_muon`	9.01	8,182	+3.0%

Key finding: The hypersphere constraint on embeddings alone (`sphere_constraint`) significantly outperformed all other configurations, including combining multiple constraints.

4.1 Throughput Analysis

Configuration	Tokens/sec	Overhead
AdamW only	89K	-
Baseline (Muon)	86K	-3%
Sphere constraint	85K	-4%
Stiefel manifold	58K	-35%

The Stiefel constraint's Newton-Schulz iteration adds significant computational cost, while the sphere projection is nearly free.

5 Analysis

Our results suggest that at this model scale:

1. **Embeddings benefit from geometric constraints.** The hypersphere constraint forces the model to learn directional representations rather than relying on magnitude differences, improving generalization.

2. **Stiefel constraints hurt more than they help.** While theoretically appealing for keeping singular values bounded, the overhead outweighs benefits at 42M parameters.

3. **Combining constraints doesn't stack.** `full_manifold` (sphere + Stiefel) underperformed `sphere_constraint` alone, suggesting interference between optimization dynamics.

4. **Muon alone provides strong baseline.** The spectral normalization of updates in Muon may already provide sufficient regularization for attention/FFN layers.

6 Conclusion

We demonstrate that *selective* manifold constraints - specifically hypersphere projection on embeddings - improve transformer training efficiency with minimal overhead. The key insight is that different parameter groups have different optimization needs: embeddings are sensitive to the geometry that constrain all singular values, while attention and FFN weights benefit from spectrally-normalized updates (Muon) without explicit manifold constraints.

Future work should explore:

- Scaling to 1B+ parameter models where Stiefel costs may amortize
- Adaptive constraint selection during training
- Combining with low-precision training

References

[1] Bernstein, J. (2025). *Modular Manifolds*. Thinking Machines Lab. Available at: `https://thinkingmachines.ai/blog/modular-manifolds/`

[2] Jordan, K. (2024). *Muon: An optimizer for hidden layers in neural networks*. GitHub Repository. Available at: `https://github.com/KellerJordan/Muon`

[3] Su, J. (2025). *Steepest Descent on Manifolds: 3. Muon + Stiefel*. Scientific Spaces (kexue.fm). Available at: `https://kexue.fm/archives/11221`

[4] Vaswani, A., et al. (2017). Attention is All You Need. *NeurIPS*, 30.

[5] Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8).

[6] Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. *ICLR*.

[7] Edelman, A., Arias, T. A., & Smith, S. T. (1998). The Geometry of Algorithms with Orthogonality Constraints. *SIAM J. Matrix Anal. Appl.*, 20(2), 303-353.

[8] Absil, P.-A., Mahony, R., & Sepulchre, R. (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

[9] Liu, W., et al. (2017). SphereFace: Deep Hypersphere Embedding for Face Recognition. *CVPR*.

[10] Bansal, N., Chen, X., & Wang, Z. (2018). Can We Gain More from Orthogonality Regularizations in Training Deep Networks? *NeurIPS*, 31.