Journal of Optimization Theory and Applications (2026) 209:2
https://doi.org/10.1007/s10957-026-02958-8

Geometric-Entropic Optimization: Integrating Optimal Transport with Riemannian Gradient Methods for Neural Network Training

Massimiliano Ferrara[1,2,3]

Received: 7 January 2026 / Accepted: 16 February 2026

Abstract
We introduce Geometric-Entropic Optimization (GEO), an algorithm for neural network training that integrates Riemannian gradient methods with entropy-regularized optimal transport. The algorithm operates on a parameter manifold equipped with a combined Fisher-Wasserstein metric and incorporates Sinkhorn-type projections to enforce distributional constraints on layer activations. We establish convergence guarantees showing that GEO achieves an $O(1/\sqrt{T}) + O(\rho^{2K})$ rate, where the first term reflects Riemannian gradient descent and the second captures the contraction of Sinkhorn iterations. Computational experiments on continuous control tasks and language modeling demonstrate consistent improvements over standard optimizers, with performance gains of approximately 20% on benchmark tasks. The theoretical framework unifies recent architectural innovations in deep learning, including manifold-constrained connections and orthogonality-preserving updates within a coherent optimization-theoretic perspective rooted in the geometric dynamics tradition.

Keywords Riemannian optimization · Optimal transport · Sinkhorn algorithm · Neural network training · Fisher information metric · Geometric dynamics

Mathematics Subject Classification 90C26 · 49M37 · 58E17 · 65K10

Communicated by Sándor Zoltán Németh.

✉ Massimiliano Ferrara
massimiliano.ferrara@unirc.it

1 Department of Law, Economics and Human Sciences & Decisions Lab, University Mediterranea of Reggio Calabria, Reggio Calabria, Italy

2 ICRIOS – Invernizzi Centre for Research on Innovation, Organization, Strategy and Entrepreneurship, Bocconi University, Milan, Italy

3 Advanced Computing Laboratory, Faculty of Engineering and Natural Sciences, Istanbul Okan University, Istanbul, Turkey

Published online: 11 March 2026

1 Introduction

The optimization of neural network parameters presents fundamental challenges that standard gradient descent methods address only partially. The loss landscape is highly non-convex, riddled with saddle points, and exhibits pathological curvature that varies dramatically across parameter space. While adaptive methods such as Adam [8] have become ubiquitous, they treat parameter space as Euclidean, ignoring the rich geometric structure inherent in neural network optimization.

This paper develops an optimization framework that respects the intrinsic geometry of the problem. We build on two foundational observations. First, the parameter space of a neural network admits a natural Riemannian structure induced by the Fisher information metric, which captures the statistical sensitivity of model outputs to parameter perturbations [2]. Second, the flow of information through network layers can be understood through the lens of optimal transport theory, where layer activations are transformed according to transport maps that minimize a suitable cost functional [11, 15].

The synthesis of these perspectives yields Geometric-Entropic Optimization (GEO), an algorithm that combines Riemannian gradient descent with entropy-regularized optimal transport constraints. The key insight is that Sinkhorn's classical theorem on doubly stochastic matrices [12], recently recognized as central to computational optimal transport [3], provides an efficient mechanism for projecting neural network dynamics onto geometrically structured manifolds.

Our work is motivated by recent architectural innovations that have achieved remarkable efficiency gains through geometric constraints. DeepSeek's Manifold-Constrained Hyper-Connections (mHC) [16] project residual connections onto specific manifolds using Sinkhorn-Knopp normalization. The Muon optimizer [7] maintains orthogonality of weight matrices through Newton-Schulz iterations. These approaches share a common mathematical foundation that we make explicit and extend.

The theoretical contributions of this paper are threefold:

1. We introduce the Fisher-Wasserstein metric on neural network parameter space and characterize its geometric properties (Section 3).
2. We establish convergence guarantees for GEO under standard smoothness assumptions, deriving explicit rates that depend on both Riemannian curvature and Sinkhorn contraction (Section 4.2).
3. We demonstrate that recent architectural innovations can be understood as special cases of manifold-constrained optimization with entropic regularization (Section 5).

The paper continues the geometric dynamics tradition pioneered by Udrişte [13], who developed systematic connections between differential geometry and optimization theory. Our previous collaborations explored these connections in the context of multi-time optimal control [14] and nonholonomic systems. The present work extends this program to the high-dimensional, stochastic setting characteristic of modern machine learning.

2 Problem Formulation

Consider a neural network $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta \subset \mathbb{R}^n$, where n is typically large (millions to billions of parameters). Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ and a data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the training objective is

$$\min_{\theta \in \Theta} \mathcal{L}(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)]. \tag{1}$$

Standard gradient descent updates take the form $\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t)$, treating Θ as a Euclidean space. This approach has well-known limitations: the learning rate η must be tuned carefully, updates in different parameter directions have incommensurate effects on the loss, and the optimization trajectory may follow inefficient paths through parameter space.

We reformulate Problem (1) as optimization on a Riemannian manifold with additional transport constraints:

$$\begin{aligned} \min_{\theta \in \Theta} \quad & \mathcal{L}(\theta) + \lambda \mathcal{R}(\theta) \\ \text{subject to} \quad & A^{(l)}(\theta) \in \mathcal{M}_l, \quad l = 1, \ldots, L \end{aligned} \tag{2}$$

where $\mathcal{R}(\theta)$ is a transport-based regularizer, $A^{(l)}(\theta)$ denotes the activation matrix at layer l, and $\mathcal{M}_l$ is a constraint manifold (e.g., the set of doubly stochastic matrices for attention layers).

The optimization is performed with respect to the Riemannian metric $G(\theta)$ defined in Section 3, and the constraints are enforced through Sinkhorn-type projections detailed in Section 4.

3 Geometric Framework

The Fisher-Wasserstein Metric. Let $p(y|x, \theta)$ denote the conditional distribution induced by the neural network. The Fisher information metric at θ is the positive semi-definite matrix

$$g^F_{ij}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\frac{\partial \log p(y|x,\theta)}{\partial \theta^i} \frac{\partial \log p(y|x,\theta)}{\partial \theta^j}\right]. \tag{3}$$

This metric has a natural interpretation: it measures how sensitively the output distribution responds to parameter perturbations. Gradient descent with respect to g^F yields the natural gradient [2], which is invariant to reparameterization and achieves asymptotically efficient updates.

To incorporate the geometry of distributional changes, we introduce a Wasserstein component. For a fixed input x, let $p_\theta = p(\cdot|x, \theta)$ denote the output distribution. The 2-Wasserstein distance between nearby distributions satisfies

$$W_2^2(p_\theta, p_{\theta+\varepsilon}) = \varepsilon^\top g^W(\theta)\varepsilon + O(\|\varepsilon\|^3) \tag{4}$$

for small perturbations ε, defining a metric tensor $g^W(\theta)$.

Definition 1 *(Fisher-Wasserstein Metric)* The Fisher-Wasserstein metric tensor is

$$G_{ij}(\theta) = g^F_{ij}(\theta) + \lambda g^W_{ij}(\theta), \tag{5}$$

where $\lambda > 0$ is a coupling parameter balancing statistical efficiency and distributional geometry.

Remark 2 *(Wasserstein Regularization of the Condition Number)* The addition of the Wasserstein component λg^W serves as a natural regularizer for the condition number $\kappa = M/\mu$ of the combined metric G. When the Fisher Information Matrix g^F becomes ill-conditioned—a common occurrence in deep networks due to parameter redundancy and flat directions in the loss landscape—the Wasserstein term provides spectral stabilization. Specifically, if μ_F and M_F denote the smallest and largest eigenvalues of g^F, and μ_W and M_W those of g^W, then $G = g^F + \lambda g^W$ satisfies $\mu \geq \mu_F + \lambda\mu_W$ and $M \leq M_F + \lambda M_W$. For appropriate λ, this reduces the condition number from $\kappa_F = M_F/\mu_F$ to a value closer to κ_W, which is typically better-conditioned due to the regularizing effect of optimal transport geometry.

The combined metric G defines a Riemannian structure on Θ (assuming positive definiteness, which holds generically for overparameterized networks). The geodesics with respect to G represent paths of minimal "effort" that account for both parameter sensitivity and distributional change.

Manifold Constraints via Optimal Transport. For layers where activations should satisfy distributional constraints—such as attention mechanisms, where weights should distribute focus appropriately across queries and keys—we impose manifold constraints using optimal transport theory.

Consider an activation matrix $A \in \mathbb{R}^{m\times n}_+$. The set of doubly stochastic matrices is the transportation polytope

$$\mathcal{DS} = \{P \in \mathbb{R}^{m\times n}_+ : P\mathbf{1}_n = \mathbf{r},\ P^\top \mathbf{1}_m = \mathbf{c}\} \tag{6}$$

where $\mathbf{r} \in \Delta^m$ and $\mathbf{c} \in \Delta^n$ are prescribed marginals (typically uniform).

The entropy-regularized projection of A onto $\mathcal{DS}$ is

$$\Pi_\varepsilon(A) = \arg\min_{P\in\mathcal{DS}} \langle C, P\rangle + \varepsilon H(P) \tag{7}$$

where $C = -\log A$ (elementwise), $H(P) = -\sum_{ij} P_{ij}\log P_{ij}$ is the entropy, and $\varepsilon > 0$ is the regularization parameter.

Proposition 3 (Sinkhorn Form of Entropic Projection) *The solution to* (7) *has the form*

$$P^* = D_1 A D_2, \tag{8}$$

where $D_1 = \mathrm{diag}(u)$ *and* $D_2 = \mathrm{diag}(v)$ *are diagonal scaling matrices with positive entries.*

Proof The Lagrangian for (7) is

$$\mathcal{L}(P, \alpha, \beta) = \langle C, P \rangle + \varepsilon H(P) - \langle \alpha, P\mathbf{1} - \mathbf{r} \rangle - \langle \beta, P^\top \mathbf{1} - \mathbf{c} \rangle.$$

Setting $\partial \mathcal{L} / \partial P_{ij} = 0$ yields

$$C_{ij} - \varepsilon(1 + \log P_{ij}) - \alpha_i - \beta_j = 0,$$

hence $P^*_{ij} = \exp((\alpha_i - 1)/\varepsilon) \cdot A_{ij} \cdot \exp(\beta_j/\varepsilon)$, which has the stated form with $u_i = \exp((\alpha_i - 1)/\varepsilon)$ and $v_j = \exp(\beta_j/\varepsilon)$. □

The scaling matrices (D_1, D_2) are computed by the Sinkhorn-Knopp algorithm, which alternates row and column normalizations:

$$u^{(k+1)} = \mathbf{r} \oslash (Av^{(k)}), \quad v^{(k+1)} = \mathbf{c} \oslash (A^\top u^{(k+1)}), \tag{9}$$

where $\oslash$ denotes elementwise division. This iteration converges linearly with rate $\rho = \|A - \mathbf{r}\mathbf{c}^\top\|_2 / \|A\|_2 < 1$ for positive matrices A.

Multi-Scale Entropic Regularization. We introduce a hierarchy of entropy regularization at different scales:

$$\mathcal{H}(\theta) = \varepsilon_1 H_{\text{batch}}(\theta) + \varepsilon_2 H_{\text{layer}}(\theta) + \varepsilon_3 H_{\text{param}}(\theta), \tag{10}$$

where $\varepsilon_1 > \varepsilon_2 > \varepsilon_3 > 0$ enforce progressively finer regularization. This multi-scale structure enables automatic annealing from exploration (high entropy at batch level) to exploitation (low entropy at parameter level) without hand-tuned schedules.

Remark 4 *(Sensitivity of Entropy Scales Across Architectures)* The fixed entropy scales $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ exhibit architecture-dependent sensitivity. For convolutional networks (CNNs), the spatial locality of operations means that layer-level entropy H_{layer} dominates, and convergence is relatively insensitive to ε_2 within an order of magnitude. For Transformers, the global attention mechanism makes batch-level entropy H_{batch} more significant, requiring careful tuning of ε_1. Empirically, we observe that scaling $\varepsilon_i \propto d_l^{-1/2}$, where d_l is the layer dimension, provides a robust heuristic across architectures. The theoretical analysis in Section 4.2 remains valid for any fixed positive scales, though the constants in Theorem 9 depend on the specific values chosen.

4 Algorithm and Convergence Analysis

4.1 The GEO Algorithm

Algorithm 1 presents the complete GEO procedure. Each iteration consists of four steps: (1) compute the Riemannian gradient using an approximation to G^{-1}; (2) project layer activations onto constraint manifolds via Sinkhorn iterations; (3) retract weight matrices to maintain orthogonality; (4) apply multi-scale entropic regularization.

Algorithm 1 Geometric-Entropic Optimization (GEO)

Require: Initial parameters θ_0, learning rate η, Sinkhorn iterations K, entropy scales $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$
for $t = 0, 1, 2, \ldots$ **do**
  Sample minibatch $(x_i, y_i)_{i=1}^{B}$ from $\mathcal{D}$
  Compute stochastic gradient $\hat{g}_t = \nabla_\theta \hat{\mathcal{L}}(\theta_t)$
  Riemannian step: $\tilde{g}_t = \hat{G}(\theta_t)^{-1} \hat{g}_t$ ▷ KFAC approximation
  Sinkhorn projection: For each constrained layer l:
  for $k = 1, \ldots, K$ **do**
    $u^{(k)} \leftarrow \mathbf{r} \oslash (A^{(l)} v^{(k-1)})$
    $v^{(k)} \leftarrow \mathbf{c} \oslash ((A^{(l)})^\top u^{(k)})$
  end for
  $A^{(l)} \leftarrow \mathrm{diag}(u^{(K)}) A^{(l)} \mathrm{diag}(v^{(K)})$
  Orthogonal retraction: For weight matrices W:
    $W \leftarrow \mathrm{NewtonSchulz}(W, 5)$ ▷ Orthogonalize
  Entropic regularization: $\tilde{g}_t \leftarrow \tilde{g}_t + \nabla_\theta \mathcal{H}(\theta_t)$
  Update: $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
end for

The Fisher metric inverse G^{-1} is approximated using Kronecker-Factored Approximate Curvature (KFAC) [10], which exploits the layer-wise structure of neural networks to achieve $O(n)$ complexity per iteration. The orthogonal retraction uses Newton-Schulz iterations [6]:

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k) \tag{11}$$

which converges quadratically to the nearest orthogonal matrix.

Remark 5 *(Impact of KFAC Approximation on Convergence)* The KFAC approximation introduces an additional source of error beyond the Sinkhorn projection. Let $\hat{G}^{-1}$ denote the KFAC approximation to G^{-1} and define the relative approximation error $\delta_K = \|\hat{G}^{-1} - G^{-1}\|_{\mathrm{op}} / \|G^{-1}\|_{\mathrm{op}}$. When $\delta_K < 1$, the convergence guarantee of Theorem 9 remains valid with an additional multiplicative factor of $(1 - \delta_K)^{-1}$ in the first term of (12). Empirically, KFAC achieves $\delta_K \approx 0.1$–0.3 for typical network architectures [10], which has negligible impact on practical convergence. The block-diagonal structure of KFAC is particularly well-suited to our framework, as it preserves the layer-wise decomposition inherent in both the Sinkhorn projections and the multi-scale entropy regularization.

4.2 Convergence Guarantees

We establish convergence guarantees for GEO under the following assumptions.

Assumption 6 *(Smoothness)* The loss $\mathcal{L}$ is L-smooth: $\|\nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(\theta')\| \leq L \|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$.

Assumption 7 *(Metric Bounds)* The Fisher-Wasserstein metric satisfies $\mu I \preceq G(\theta) \preceq M I$ for all $\theta \in \Theta$, with condition number $\kappa = M/\mu$.

Assumption 8 *(Sinkhorn Contraction)* The Sinkhorn iterations (9) converge with rate $\rho < 1$: $\|P^{(k)} - P^*\| \leq C_0 \rho^k$ for some constant $C_0 > 0$.

Theorem 9 (GEO Convergence) *Under Assumptions 6–8, with learning rate $\eta = \mu/(LM)$ and K Sinkhorn iterations per step, the iterates of Algorithm 1 satisfy*

$$\min_{0 \leq t \leq T} \mathbb{E}[\|\nabla \mathcal{L}(\theta_t)\|^2] \leq \frac{2\kappa(\mathcal{L}(\theta_0) - \mathcal{L}^*)}{\mu T} + \frac{2L^2 C_0^2 \rho^{2K}}{\mu}, \tag{12}$$

where $\mathcal{L}^* = \inf_\theta \mathcal{L}(\theta)$.

Proof The proof proceeds in three stages.

Stage 1: Riemannian Descent. Let $g_t = G(\theta_t)^{-1} \nabla \mathcal{L}(\theta_t)$ denote the Riemannian gradient. By L-smoothness of $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}(\theta_{t+1}) &\leq \mathcal{L}(\theta_t) + \langle \nabla \mathcal{L}(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2} \|\theta_{t+1} - \theta_t\|^2 \\ &= \mathcal{L}(\theta_t) - \eta \langle \nabla \mathcal{L}(\theta_t), g_t \rangle + \frac{L\eta^2}{2} \|g_t\|^2. \end{aligned} \tag{13}$$

Using the metric bounds from Assumption 7, we obtain:

$$\langle \nabla \mathcal{L}, g_t \rangle = \nabla \mathcal{L}^\top G^{-1} \nabla \mathcal{L} \geq \frac{1}{M} \|\nabla \mathcal{L}\|^2.$$

Similarly, $\|g_t\|^2 = \nabla \mathcal{L}^\top G^{-2} \nabla \mathcal{L} \leq \frac{1}{\mu^2} \|\nabla \mathcal{L}\|^2$.

Substituting into (13):

$$\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\theta_t) - \frac{\eta}{M} \|\nabla \mathcal{L}\|^2 + \frac{L\eta^2}{2\mu^2} \|\nabla \mathcal{L}\|^2. \tag{14}$$

With $\eta = \mu/(LM)$, this simplifies to

$$\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\theta_t) - \frac{\mu}{2LM^2} \|\nabla \mathcal{L}(\theta_t)\|^2.$$

Stage 2: Sinkhorn Projection Error. The Sinkhorn projection after K iterations introduces error $\|A^{(K)} - A^*\| \leq C_0 \rho^K$ by Assumption 8. This error propagates to the gradient computation, contributing an additive term δ_t with $\|\delta_t\| \leq L C_0 \rho^K$ (by L-smoothness of the loss with respect to activations).

Stage 3: Combined Rate. Incorporating the projection error into (14) and summing over $t = 0, \ldots, T-1$:

$$\sum_{t=0}^{T-1} \|\nabla \mathcal{L}(\theta_t)\|^2 \leq \frac{2LM^2}{\mu} (\mathcal{L}(\theta_0) - \mathcal{L}^*) + \frac{2L^2 M^2 C_0^2 T \rho^{2K}}{\mu}.$$

Dividing by T and using $\kappa = M/\mu$ yields (12). □

Table 1 Mapping of architectural constraints to GEO framework components

Method	Manifold	GEO Step	Mechanism
mHC	Birkhoff $\mathcal{B}_n$	Sinkhorn	Row/column normalization
Muon	Stiefel $\mathrm{St}(n, p)$	Retraction	Newton-Schulz iteration
JEPA	Learned $\mathcal{M}$	Implicit	Architecture design

Remark 10 The convergence rate (12) consists of two terms: an $O(1/T)$ term reflecting deterministic Riemannian gradient descent, and an $O(\rho^{2K})$ term capturing Sinkhorn approximation error. For stochastic gradients, the first term becomes $O(1/\sqrt{T})$ with appropriate variance assumptions. The key insight is that increasing K reduces the second term exponentially, while the computational cost grows only linearly in K.

5 Unification of Geometric Deep Learning Methods

Before proceeding, we formally define the key manifolds that appear in this unification:

Definition 11 *(Key Constraint Manifolds)* The *Stiefel manifold* $\mathrm{St}(n, p) = \{W \in \mathbb{R}^{n\times p} : W^\top W = I_p\}$ consists of matrices with orthonormal columns. The *Birkhoff polytope* $\mathcal{B}_n = \{P \in \mathbb{R}_+^{n\times n} : P\mathbf{1} = \mathbf{1}, P^\top\mathbf{1} = \mathbf{1}\}$ is the set of doubly stochastic matrices (a convex polytope whose vertices are permutation matrices).

Table 1 summarizes how specific architectural constraints map to GEO components.

The GEO framework provides a unified perspective on several recent architectural innovations.

Manifold-Constrained Hyper-Connections. DeepSeek's mHC architecture [16] extends residual connections by allowing information exchange between multiple parallel streams, subject to manifold constraints. In our framework, this corresponds to Problem (2) with $\mathcal{M}_l$ being the Birkhoff polytope (doubly stochastic matrices).

The mHC uses Sinkhorn-Knopp normalization to project mixing matrices onto this manifold, precisely implementing our entropic projection (7) with uniform marginals. The reported stability improvements (signal gain bounded by $1.6\times$ versus $3000\times$ for unconstrained connections) follow directly from the bounded eigenvalues of doubly stochastic matrices.

Orthogonality-Preserving Optimization. The Muon optimizer [7] maintains weight matrix orthogonality through Newton-Schulz iterations (11). This can be viewed as Riemannian optimization on the Stiefel manifold $\mathrm{St}(n, p) = \{W \in \mathbb{R}^{n\times p} : W^\top W = I_p\}$ [1].

The connection to GEO is through the orthogonal retraction step (Line 12 of Algorithm 1). The Newton-Schulz iteration implements retraction from the tangent space back to the manifold, ensuring that weight matrices remain orthogonal throughout training.

Joint Embedding Predictive Architectures. LeCun's JEPA framework [9] learns representations by predicting embeddings on manifolds rather than raw inputs. The

energy function minimized during inference can be expressed as

$$E(s_x, s_y) = d_{\mathcal{M}}(s_y, \text{Pred}(s_x))^2, \tag{15}$$

where $d_{\mathcal{M}}$ is a distance on the representation manifold and Pred is a predictor network.

This is a special case of our framework with the constraint that representations lie on a learned manifold $\mathcal{M}$, enforced implicitly through the architecture rather than explicitly through projection.

6 Computational Experiments

We evaluate GEO on two complementary domains: continuous control using MuJoCo physics simulation, and language modeling on WikiText-103. The experiments use standard benchmark protocols to enable comparison with existing methods.

6.1 Experimental Setup

Continuous Control. We integrate GEO with Soft Actor-Critic (SAC) [5] on three MuJoCo environments: HalfCheetah-v2 (17-dimensional state, 6-dimensional action), Humanoid-v2 (376-dimensional state, 17-dimensional action), and Ant-v2 (111-dimensional state, 8-dimensional action). Actor and critic networks consist of 2 hidden layers with 256 units each and ReLU activations. Training proceeds for 10^6 environment steps with 5 random seeds per configuration.

Language Modeling. We train 125-million-parameter transformer models on WikiText-103 (approximately 100 million tokens). The architecture uses 12 layers, hidden dimension 768, and 12 attention heads with context length 256. Training proceeds for 100,000 steps with batch size 64.

Baselines. We compare against Adam [8], AdamW, LAMB, Shampoo [4], and Muon [7]. All methods use tuned hyperparameters from published work or grid search.

GEO Configuration. Fisher metric approximated via KFAC with damping 10^{-3}. Sinkhorn iterations $K = 5$. Entropy scales $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0.1, 0.01, 0.001)$. Transport regularization weight $\lambda = 0.01$.

6.2 Results

Table 2 presents continuous control results. GEO achieves consistent improvements across all environments, with gains scaling with task complexity. On Humanoid, the most challenging environment, GEO outperforms Adam by 21.7% and Muon by 6.8%. Notably, the standard deviation across seeds decreases substantially (from 412 to 289 on Humanoid), indicating more robust optimization.

Table 3 presents language modeling results. GEO achieves the lowest perplexity (20.3 vs. 22.4 for Adam) and reaches target perplexity 25 in 61,000 steps compared to 85,000 for Adam (a 28% reduction). Training stability, measured as $1 - \text{CV}$ of the loss over the final 10,000 steps, improves from 0.89 to 0.97.

Table 2 Continuous control performance: average return ($\pm$ std) over 5 seeds after 10^6 steps

Optimizer	HalfCheetah	Humanoid	Ant
Adam	$9{,}835 \pm 513$	$5{,}123 \pm 412$	$4{,}521 \pm 389$
Shampoo	$10{,}456 \pm 412$	$5{,}567 \pm 367$	$4{,}912 \pm 334$
Muon	$10{,}823 \pm 356$	$5{,}834 \pm 312$	$5{,}189 \pm 298$
GEO	$\mathbf{11{,}892 \pm 298}$	$\mathbf{6{,}234 \pm 289}$	$\mathbf{5{,}612 \pm 267}$

Table 3 Language modeling on WikiText-103: 125M parameter transformers

Optimizer	PPL	Steps to 25	Stability	Time/step	Total Time
Adam	22.4	85K	0.89	1.00×	1.00×
AdamW	21.8	78K	0.92	1.02×	0.94×
Muon	21.1	68K	0.94	1.35×	1.08×
GEO	**20.3**	**61K**	**0.97**	1.51×	**1.08**×

Table 4 Ablation study on HalfCheetah-v2

Configuration	Return	Degradation
Full GEO	$11{,}892 \pm 298$	–
− Fisher metric	$10{,}567 \pm 423$	−11.1%
− Multi-scale entropy	$11{,}123 \pm 389$	−6.5%
− Sinkhorn projection	$11{,}234 \pm 345$	−5.5%
− Transport regularization	$11{,}345 \pm 356$	−4.6%
− Orthogonal retraction	$11{,}456 \pm 367$	−3.7%

6.3 Ablation Study

Table 4 isolates the contribution of each GEO component on HalfCheetah. The Fisher metric provides the largest benefit (11.1% degradation when removed), confirming the importance of Riemannian geometry. Multi-scale entropy contributes 6.5%, validating the theoretical framework. Sinkhorn projection and transport regularization contribute 5.5% and 4.6% respectively.

6.4 Computational Overhead

GEO incurs 51% overhead in wall-clock time per iteration relative to Adam. However, the 28% reduction in iterations to reach target performance yields a net computational benefit when training to a fixed quality threshold. Memory overhead is 33%, primarily from storing KFAC factors.

7 Conclusion

We have introduced Geometric-Entropic Optimization, an algorithm that integrates Riemannian gradient methods with entropy-regularized optimal transport for neural network training. The theoretical framework provides convergence guarantees and unifies recent architectural innovations within a coherent optimization-theoretic perspective.

The key contributions are: (1) the Fisher-Wasserstein metric combining statistical and distributional geometry; (2) Sinkhorn-based projections for enforcing manifold constraints; (3) multi-scale entropic regularization enabling automatic exploration-exploitation tradeoff; (4) convergence analysis with explicit dependence on geometric quantities.

Empirical results demonstrate consistent improvements over standard optimizers on both continuous control and language modeling tasks. The framework suggests that the future of neural network optimization lies not in ever-larger models, but in the principled application of geometric structure to constrain and guide learning dynamics.

This work continues the geometric dynamics tradition and is dedicated to the memory of Constantin Udrişte (1946 -2023), whose foundational contributions made this research possible.

Acknowledgements The author thanks the anonymous reviewers for their constructive feedback. This work was partially supported by the Decisions Lab at University Mediterranea of Reggio Calabria.

Funding Open access funding provided by Università degli Studi Mediterranea di Reggio Calabria within the CRUI-CARE Agreement.

Data Availability The continuous control experiments used MuJoCo environments available through OpenAI Gym. The language modeling experiments used the WikiText-103 dataset from Salesforce Research. No custom datasets were generated. Source data are available from the author upon reasonable request.

Declarations

Conflict of Interest The author declares no conflict of interest.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
2. Amari, S.I.: Natural gradient works efficiently in learning. Neural Comput. **10**(2), 251–276 (1998)

3. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems, vol. 26, pp. 2292–2300 (2013)
4. Gupta, V., Koren, T., Singer, Y.: Shampoo: Preconditioned stochastic tensor optimization. In: Proceedings of the 35th International Conference on Machine Learning, pp. 1842–1850 (2018)
5. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning, pp. 1861–1870 (2018)
6. Higham, N.J.: Functions of Matrices: Theory and Computation. SIAM, Philadelphia (2008)
7. Jordan, K., Jin, Y., Boza, V., et al.: Muon: An optimizer for hidden layers in neural networks. Technical report (2024). https://kellerjordan.github.io/posts/muon/
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, pp. 1–15 (2015). Available at: arXiv:1412.6980
9. LeCun, Y.: A path towards autonomous machine intelligence. Preprint, OpenReview (2022)
10. Martens, J., Grosse, R.: Optimizing neural networks with Kronecker-factored approximate curvature. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 2408–2417 (2015)
11. Peyré, G., Cuturi, M.: Computational optimal transport. Found. Trends Mach. Learn. **11**(5–6), 355–607 (2019)
12. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Stat. **35**(2), 876–879 (1964)
13. Udrişte, C.: Geometric Dynamics. Kluwer Academic Publishers, Dordrecht (2000)
14. Udrişte, C., Ferrara, M., Zugrăvescu, D., Munteanu, F.: Controllability of a nonholonomic macroeconomic system. J. Optim. Theory Appl. **154**(3), 1036–1054 (2012)
15. Villani, C.: Optimal Transport: Old and New. Springer, Berlin (2009)
16. Xie, Z., Wei, Y., Cao, H., et al.: mHC: Manifold-Constrained Hyper-Connections. Preprint arXiv:2501.01427 (2025)