Riemannian Deep Learning: Normalization and Classification

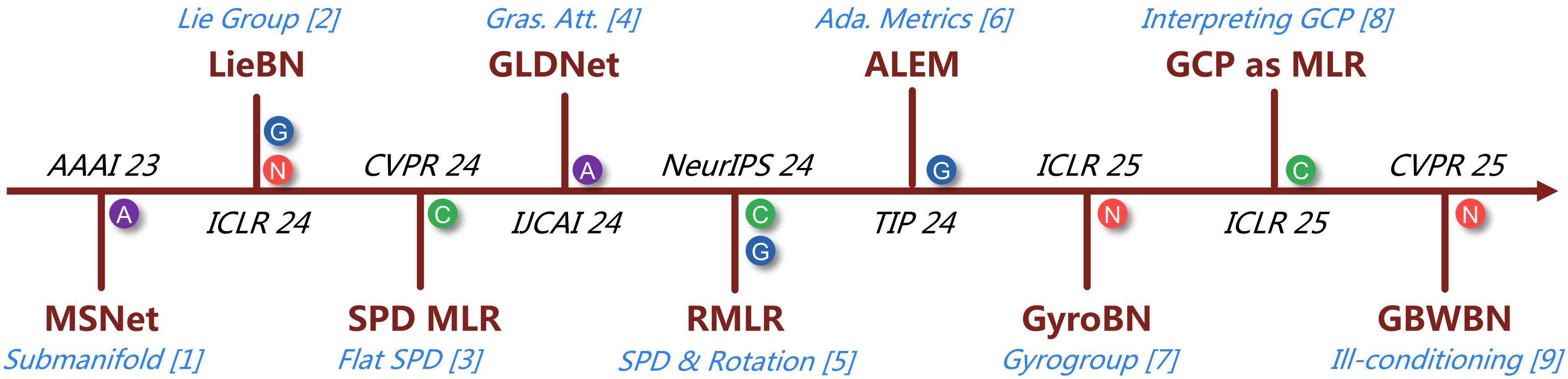
Ziheng Chen
Supervisor: Nicu Sebe
University of Trento
MHUG Lab

UNIVERSITÀ DI TRENTO

Research Route

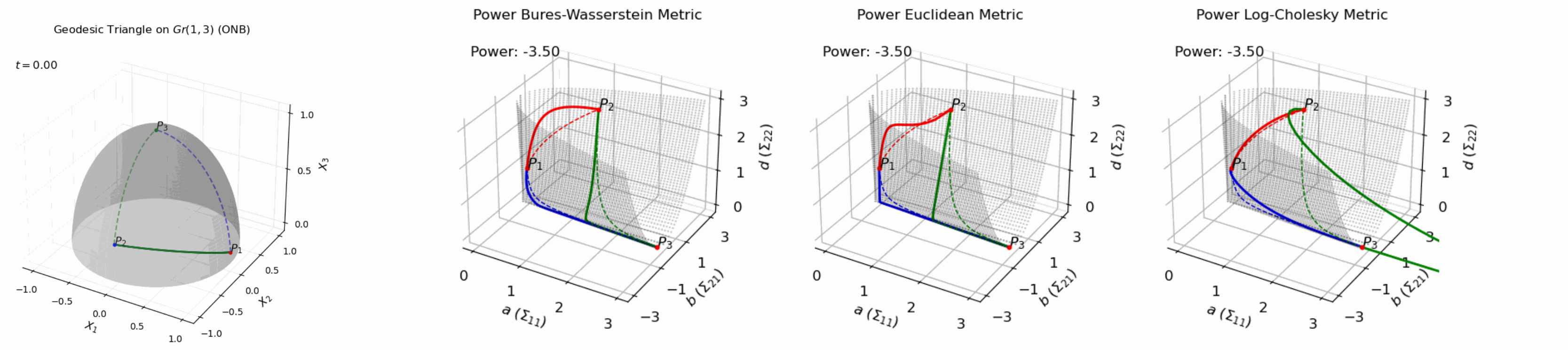
Riemannian Deep Learning

N Normalization C Classification G Geometry A Architecture



[1] **Chen Z**, et al. Riemannian local mechanism for SPD neural networks. AAAI 2023
[2] **Chen Z**, et al. A Lie group approach to Riemannian batch normalization. ICLR 2024
[3] **Chen Z**, et al. Riemannian multinomial logistics regression for SPD neural networks. CVPR 2024
[4] Wang R, Hu C, **Chen Z**†, et al. A Grassmannian manifold self-attention network for signal classification. IJCAI 2024
[5] **Chen Z**, et al. Adaptive Log-Euclidean metrics for SPD matrix learning. IEEE TIP 2024.
[6] **Chen Z**, et al. RMLR: Extending Multinomial Logistic Regression into General Geometries. NeurIPS 2024
[7] **Chen Z**, et al. Gyrogroup Batch Normalization. ICLR 2025
[8] **Chen Z**, et al. Understanding matrix function normalizations in covariance pooling through the lens of Riemannian geometry. ICLR 2025
[9] Wang, R, Jin, S, **Chen, Z**†, et al. Learning to Normalize on the SPD Manifold under Bures-Wasserstein Geometry. CVPR 2025.

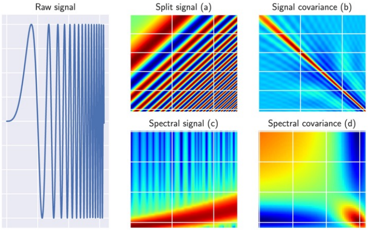
Backgrounds



- Manifolds differ with the Euclidean space

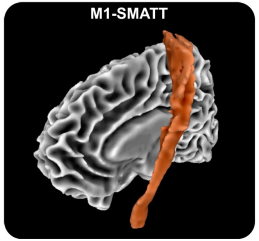
Motivations

Radar Classification



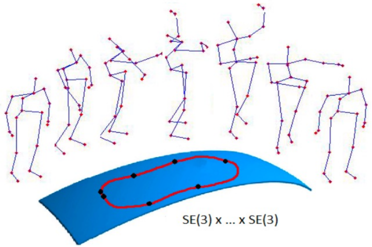
Brooks et al., 2020

Medical



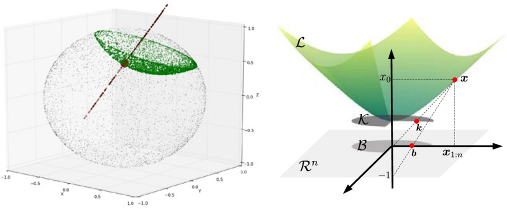
Chakraborty et al., 2020

Action Recognition



Vemulapalli, Raviteja 2014

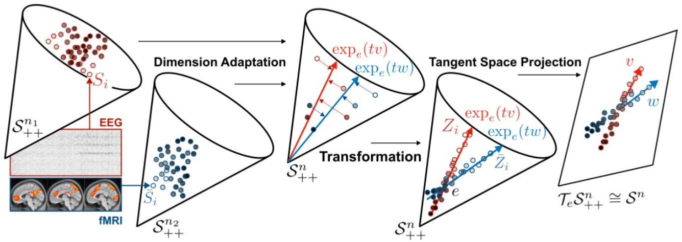
NLP, Graph …



Ganea et al., 2018

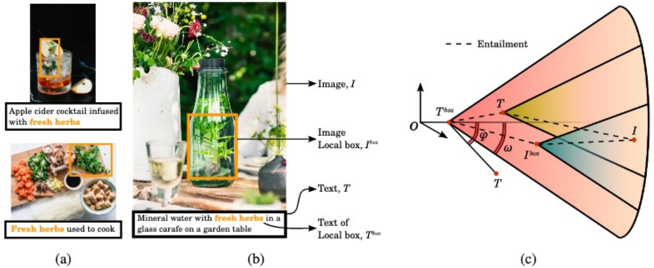
Dai et al., 2021

Brain-Computer Interfaces



Ju et al., 2024

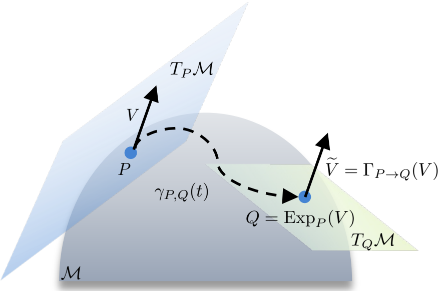
Vision-Language Models



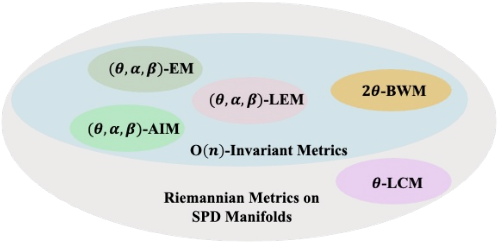
Pal et al., 2025

Brooks, Daniel, et al. "Deep learning and information geometry for drone micro-Doppler radar classification." RadarConf, 2020.
Chakraborty R et al. Manifoldnet: A deep neural network for manifold-valued data with applications. T-PAMI, 2020
Vemulapalli, Raviteja, et al. Human action recognition by representing 3D skeletons as points in a lie group. CVPR. 2014.
Ju, Ce, et al. "Deep geodesic canonical correlation analysis for covariance-based neuroimaging data." ICLR, 2024.
Pal A, et al. Compositional Entailment Learning for Hyperbolic Vision-Language Models. ICLR 2025
Ganea, O., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic neural networks. NeurIPS, 31.
Dai, J., Wu, Y., Gao, Z., & Jia, Y. (2021). A hyperbolic-to-hyperbolic graph convolutional network. CVPR, 2021

EXAMPLES



Manifold: Locally Euclidean



There could be multiple metrics on a manifold

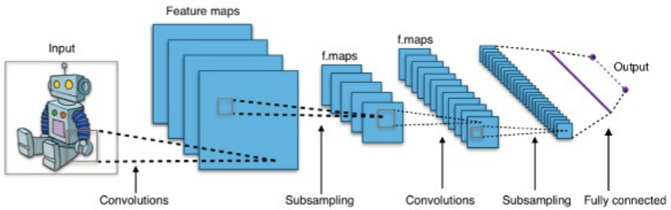
Examples in Machine Learning

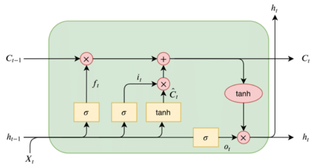
Category	Name	Definition
Matrix Manifolds	Symmetric Positive Definite (SPD)	$\mathcal{S}^n_{++} = \{X \in \mathbb{R}^{n\times n} : X = X^T, X \succ 0\}$
	Symmetric Positive Semi-Definite (SPSD)	$\mathcal{S}^n_{+} = \{X \in \mathbb{R}^{n\times n} : X = X^T, X \succeq 0\}$
	General Linear Group	$\mathrm{GL}(n) = \{A \in \mathbb{R}^{n\times n} : \det(A) \neq 0\}$
	Orthogonal Group	$\mathrm{O}(n) = \{Q \in \mathbb{R}^{n\times n} : Q^T Q = I\}$
	Special Orthogonal Group	$\mathrm{SO}(n) = \{Q \in \mathbb{R}^{n\times n} : Q^T Q = I, \det(Q) = 1\}$
	Special Euclidean Group	$\mathrm{SE}(n) = \{(R, t) : R \in \mathrm{SO}(n), t \in \mathbb{R}^n\}$
	Stiefel	$\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} : X^T X = I_p\}$
	Grassmannian (ONB Perspective)	$\mathrm{Gr}(p, n) = \{[U] : [U] := \{\widetilde{U} \in \mathrm{St}(p, n) \mid \widetilde{U} = UR, R \in \mathrm{O}(p)\}\}$
	Grassmannian (Projector Perspective)	$\widetilde{\mathrm{Gr}}(p, n) = \{P \in \mathcal{S}^n : P^2 = P, \mathrm{rank}(P) = p\}$
	Oblique	$\mathrm{Oblique}(n, p) = \{X \in \mathbb{R}^{n\times p} : \lVert X_i \rVert = 1, \forall i = 1, \ldots, p\}$
Vector Manifolds	Constant Curvature Space (CCS)	$\mathcal{M}_K = \begin{cases} \mathbb{S}^n_K = \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle = \frac{1}{K}\}, & K > 0 \\ \mathbb{H}^n_K = \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle_L = \frac{1}{K}\}, & K < 0 \\ \mathbb{R}^n, & K = 0 \end{cases}$
	Projected Hypersphere	$\mathbb{D}^n_K = \{x \in \mathbb{R}^n : \langle x, x\rangle = \frac{1}{K}\}$, for $K > 0$
	Poincaré Ball	$\mathbb{P}^n_K = \{x \in \mathbb{R}^n : \langle x, x\rangle < -\frac{1}{K}\}$, for $K < 0$
	Klein Model	$\mathbb{K}^n_K = \{x \in \mathbb{R}^n : \langle x, x\rangle < -\frac{1}{K}\}$, for $K < 0$
	Half-Space Model	$\mathbb{HS}^n_K = \{x \in \mathbb{R}^n : x_n > \sqrt{-1/K}\}$, for $K < 0$
	Upper Hemisphere Model	$\mathbb{SU}^n_K = \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle = -\frac{1}{K}, x_0 > 0\}$, for $K < 0$

Chen Z, et al. RMLR: Extending Multinomial Logistic Regression into General Geometries. NeurIPS 2024

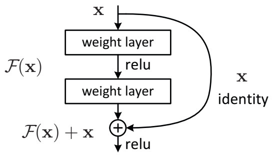
DEEP LEARNING: FROM EUCLIDEAN TO RIEMANNIAN

CNNs

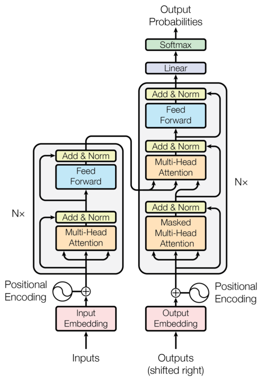




RNNs



ResNet



Transformer

Basic building blocks →

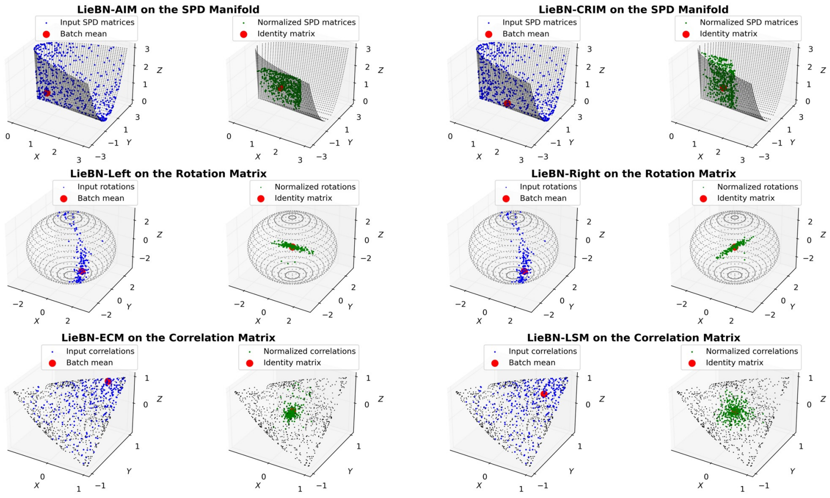
- Transformation
- Activation
- Normalization
- Classification

Intrinsically → Riemannian Spaces

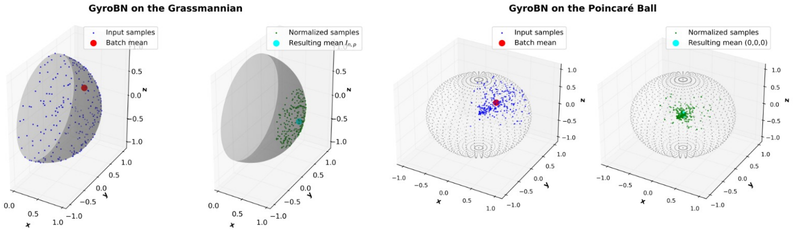
Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. NeurIPS,2012.
He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. CVPR. 2016.
Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. NeurIPS, 2017.

Riemannian Normalization

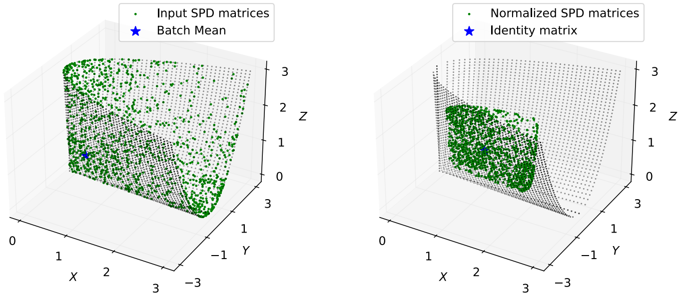
Overview: Riemannian Normalization



LieBN (ICLR24)
LieBN-Extension (PAMI; under review)



GyroBN (ICLR24)



GBWMBN for ill-conditioning (CVPR25)

Chen Z, et al. A Lie group approach to Riemannian batch normalization. ICLR 2024
Chen Z, et al. LieBN: Batch Normalization over Lie Groups. under review
Chen Z, et al. Gyrogroup Batch Normalization. ICLR 2025
Wang, R., Jin, S., **Chen, Z.**†, et al. Learning to Normalize on the SPD Manifold under Bures-Wasserstein Geometry. CVPR 2025.

Euclidean Normalization: facilitating network training by controlling mean and variance

$$\forall i \leq N, x_i \leftarrow \gamma \frac{x_i - \mu_b}{\sqrt{v_b^2 + \epsilon}} + \beta$$

TABLE 2: Summary of some representative RBN methods.

Methods	Involved Statistics	Controllable Mean	Controllable Variance	Geometries
SPDBN [32, Alg. 1]	Mean	✓	N/A	SPD manifolds under AIM
SPDBN [57, Alg. 1]	Mean+Variance	✓	✓	SPD manifolds under AIM
SPDDSMBN [33]	Mean+Variance	✓	✓	SPD manifolds under AIM
ManifoldNorm [34, Algs. 1-2]	Mean+Variance	✗	✗	Riemannian homogeneous space
ManifoldNorm [34, Algs. 3-4]	Mean+Variance	✓	✓	A specific Lie group structure and distance
RBN [35, Alg. 2]	Mean+Variance	✗	✗	Geodesically complete manifolds
LieBN (Ours)	Mean+Variance	✓	✓	General Lie groups

- All the previous RBN methods fail to control statistics in a principled manner
- Several manifold manifold-valued features form Lie groups

Chen Z, et al. A Lie group approach to Riemannian batch normalization. ICLR 2024

Lie Groups

Definition 2.1 (Lie Groups). A manifold is a Lie group, if it forms a group with a group operation $\odot$ such that $m(x, y) \mapsto x \odot y$ and $i(x) \mapsto x_{\odot}^{-1}$ are both smooth, where $x_{\odot}^{-1}$ is the group inverse.

Definition 2.2 (Left-invariance). A Riemannian metric g over a Lie group $\{G, \odot\}$ is left-invariant, if for any $x, y \in G$ and $V_1, V_2 \in T_x\mathcal{M}$,

$$g_y(V_1, V_2) = g_{L_x(y)}(L_{x*,y}(V_1), L_{x*,y}(V_2)), \tag{1}$$

where $L_x(y) = x \odot y$ is the left translation by x, and $L_{x*,y}$ is the differential map of L_x at y.

Let $\{w_{1...N}\}$ be weights satisfying a convexity constraint, *i.e.*,$\forall i, w_i > 0$ and $\sum_i w_i = 1$. The weighted Fréchet mean (WFM) of a set of SPD matrices $\{P_{i...N}\}$ is defined as

$$\mathrm{WFM}(\{w_i\}, \{P_i\}) = \underset{S \in \mathcal{S}^n_{++}}{\operatorname{argmin}} \sum_{i=1}^{N} w_i \, \mathrm{d}^2(P_i, S), \tag{5}$$

Examples

Table 1: Lie group structures and the associated Riemannian operators on SPD manifolds.

Metric	(α,β)-LEM	(α,β)-AIM	LCM
$g_P(V,W)$	$\langle \mathrm{mlog}_{*,P}(V), \mathrm{mlog}_{*,P}(W) \rangle^{(\alpha,\beta)}$	$\langle P^{-1}V, WP^{-1} \rangle^{(\alpha,\beta)}$	$\sum_{i>j} V_{ij}W_{ij} + \sum_{j=1}^{n} V_{jj}W_{jj}L_{jj}^{-2}$
$\mathrm{d}(P,Q)$	$\lVert \mathrm{mlog}(P) - \mathrm{mlog}(Q) \rVert^{(\alpha,\beta)}$	$\left\lVert \mathrm{mlog}\left(Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}}\right) \right\rVert^{(\alpha,\beta)}$	$\lVert \psi_{\mathrm{LC}} \circ \mathrm{Chol}(P) - \psi_{\mathrm{LC}} \circ \mathrm{Chol}(Q) \rVert_{\mathrm{F}}$
$Q \odot P$	$\mathrm{mexp}(\mathrm{mlog}(P) + \mathrm{mlog}(Q))$	$KPK^\top$	$\mathrm{Chol}^{-1}(\lfloor L + K \rfloor + \mathbb{K}\mathbb{L})$
$\mathrm{FM}(\{P_i\})$	$\mathrm{mexp}\left(\frac{1}{n}\sum_i \mathrm{mlog}\, P_i\right)$	Karcher Flow	$\psi_{\mathrm{LC}}^{-1}\left(\frac{1}{n}\sum_i \psi_{\mathrm{LC}}(P_i)\right)$
$\mathrm{Log}_P\, Q$	$(\mathrm{mlog}_{*,P})^{-1}\left[\mathrm{mlog}(Q) - \mathrm{mlog}(P)\right]$	$P^{\frac{1}{2}} \mathrm{mlog}\left(P^{\frac{-1}{2}} Q P^{\frac{-1}{2}}\right) P^{\frac{1}{2}}$	$(\mathrm{Chol}^{-1})_{*,L}\left[\lfloor K \rfloor - \lfloor L \rfloor + \mathbb{L}\,\mathrm{Dlog}(\mathbb{L}^{-1}\mathbb{K})\right]$
Invariance	Bi-invariance	Left-invariance	Bi-invariance

Table 8: The associated Riemannian operators on Rotation matrices.

Operators	$\mathrm{d}^2(R,S)$	$\mathrm{Log}_I\, R$	$\mathrm{Exp}_I(A)$	$\gamma_{(R,S)}(t)$	FM
Expression	$\lVert \mathrm{mlog}(R^\top S) \rVert_{\mathrm{F}}^2$	$\mathrm{mlog}(R)$	$\mathrm{mexp}(A)$	$R\,\mathrm{mexp}(t\,\mathrm{mlog}(R^\top S))$	Manton (2004, Alg. 1)

Construction

Euclidean BN

$$\forall i \leq N, x_i \leftarrow \gamma \frac{x_i - \mu_b}{\sqrt{v_b^2 + \epsilon}} + \beta$$

- Gaussian
- Mean and variance
- Centering, biasing, scaling

LieBN

Gaussian on manifolds: $p\left(X \mid M, \sigma^2\right) = k(\sigma) \exp\left(-\frac{\mathrm{d}(X, M)^2}{2\sigma^2}\right),$

Centering to the neutral element E: $\forall i \leq N, \bar{P}_i \leftarrow L_{M_\odot^{-1}}(P_i),$

Scaling the dispersion: $\forall i \leq N, \hat{P}_i \leftarrow \mathrm{Exp}_E\left[\frac{s}{\sqrt{v^2+\epsilon}} \mathrm{Log}_E(\bar{P}_i)\right],$

Biasing towards parameter $B \in \mathcal{M}$: $\forall i \leq N, \tilde{P}_i \leftarrow L_B(\hat{P}_i),$

Question

- Can they normalize sample statistics?

Yes

Proposition 4.1 (Population). [↓] *Given a random point* X *over* $\mathcal{M}$*, and the Gaussian distribution* $\mathcal{N}(M, v^2)$ *defined in Eq.* (12)*, we have the following properties for the population statistics:*

1. *(MLE of* M*) Given* $\{P_{i...N} \in \mathcal{M}\}$ *i.i.d. sampled from* $\mathcal{N}(M, v^2)$*, the maximum likelihood estimator (MLE) of* M *is the sample Fréchet mean.*
2. *(Homogeneity) Given* $X \sim \mathcal{N}(M, v^2)$ *and* $B \in \mathcal{M}$*,* $L_B(X) \sim \mathcal{N}(L_B(M), v^2)$

Proposition 4.2 (Sample). [↓] *Given* N *samples* $\{P_{i...N} \in \mathcal{M}\}$*, denoting* $\phi_s(P_i) = \mathrm{Exp}_E[s \, \mathrm{Log}_E(P_i)]$*, we have the following properties for the sample statistics:*

$$\text{Homogeneity of the sample mean: } \mathrm{FM}\{L_B(P_i)\} = L_B(\mathrm{FM}\{P_i\}), \forall B \in \mathcal{M}, \tag{16}$$

$$\text{Controllable dispersion from } E\text{: } \sum_{i=1}^{N} w_i \, \mathrm{d}^2(\phi_s(P_i), E) = s^2 \sum_{i=1}^{N} w_i \, \mathrm{d}^2(P_i, E), \tag{17}$$

where $\{w_{1...N}\}$ *are weights satisfying a convexity constraint,* i.e.,$\forall i, w_i > 0$ *and* $\sum_i w_i = 1$.

ALGORITHM

Algorithm 1: Lie Group Batch Normalization (LieBN) Algorithm

Input : A batch of activations $\{P_{1\ldots N} \in \mathcal{M}\}$, a small positive constant ϵ, and momentum $\gamma \in [0,1]$
running mean $M_r = E$, running variance $v_r^2 = 1$,
biasing parameter $B \in \mathcal{M}$, scaling parameter $s \in \mathbb{R}/\{0\}$,

Output : Normalized activations $\{\tilde{P}_{1\ldots N}\}$

if *training* **then**
  Compute batch mean M_b and variance v_b^2 of $\{P_{1\ldots N}\}$;
  Update running statistics $M_r \leftarrow \mathrm{WFM}(\{1-\gamma,\gamma\},\{M_r,M_b\})$, $v_r^2 \leftarrow (1-\gamma)v_r^2 + \gamma v_b^2$;
end
if *training* **then** $M \leftarrow M_b, v^2 \leftarrow v_b^2$;
else $M \leftarrow M_r, v^2 \leftarrow v_r^2$;
for $i \leftarrow 1$ **to** N **do**
  Centering to the neutral element E: $\bar{P}_i \leftarrow L_{M_\odot^{-1}}(P_i)$
  Scaling the dispersion: $\hat{P}_i \leftarrow \mathrm{Exp}_E\left[\frac{s}{\sqrt{v^2+\epsilon}}\mathrm{Log}_E(\bar{P}_i)\right]$
  Biasing towards parameter B: $\tilde{P}_i \leftarrow L_B(\hat{P}_i)$
end

A natural extension of the Euclidean BN:

Proposition D.1. *The LieBN algorithm presented in Alg. 1 is equivalent to the standard Euclidean BN when $\mathcal{M} = \mathbb{R}^n$, both during the training and testing phases.*

EXPERIMENTS

Radar

(a) Radar dataset.

Method	SPDNet	SPDNetBN	AIM-(1)	LEM-(1)	LCM-(1)	LCM-(-0.5)
Fit Time (s)	0.98	1.56	1.62	1.28	1.11	1.43
Mean±STD	93.25±1.10	94.85±0.99	**95.47±0.90**	94.89±1.04	93.52±1.07	94.80±0.71
Max	94.4	96.13	96.27	**96.8**	95.2	95.73

Skeleton

(b) HDM05 and FPHA datasets.

Method		SPDNet	SPDNetBN	AIM-(1)	LEM-(1)	LCM-(1)	AIM-(1.5)	LCM-(0.5)
HDM05	Fit Time (s)	0.57	0.97	1.14	0.87	0.66	1.46	1.01
	Mean±STD	59.13±0.67	66.72±0.52	67.79±0.65	65.05±0.63	66.68±0.71	68.16±0.68	**70.84±0.92**
	Max	60.34	67.66	68.75	66.05	68.52	69.25	**72.27**
FPHA	Fit Time (s)	0.32	0.62	0.80	0.55	0.39	1.03	0.65
	Mean±STD	85.59±0.72	89.33±0.49	89.70±0.51	86.56±0.79	77.64±1.00	**90.39±0.66**	86.33±0.43
	Max	86	90.17	90.5	87.83	79	**92.17**	87

EEG

(a) Inter-session classification

Method	Fit Time (s)	Mean±STD
SPDDSMBN	0.16	54.12±9.87
AIM-(1)	0.16	**55.10±7.61**
LEM-(1)	0.13	54.95±10.09
LCM-(1)	0.10	51.54±6.88
LCM-(0.5)	0.15	53.11±5.65

(b) Inter-subject classification

Method	Fit Time (s)	Mean±STD
SPDDSMBN	7.74	50.10±8.08
AIM-(1)	6.94	50.04±8.01
LEM-(1)	4.71	50.95±6.40
LCM-(1)	3.59	51.86±4.53
AIM-(-0.5)	8.71	**53.97±8.78**

SO(3)

Table 9: Results of LieNet with or without LieBN on the G3D dataset.

Methods	G3D	
	Mean±STD	Max
LieNet	87.91±0.90	89.73
LieNetLieBN	88.88±1.62	90.67

$$g_y^{\mathrm{R}}(V_1, V_2) = g_{\mathrm{R}_x(y)}^{\mathrm{R}}\left(\mathrm{R}_{x*,y}(V_1), \mathrm{R}_{x*,y}(V_2)\right), \text{ with } \mathrm{R}_x(y) = y \odot x$$

centering to E: $\bar{P}_i \leftarrow \mathrm{R}_{M_{\odot}^{-1}}(P_i)$,

biasing towards B: $\tilde{P}_i \leftarrow \mathrm{R}_B(\hat{P}_i)$.

→

Proposition 4.5. *Given a random point $X \sim \mathcal{N}(M, v^2)$ over $\{\mathcal{M}, g^{\mathrm{R}}\}$, $B \in \mathcal{M}$, and N samples $\{P_{i\ldots N}\}$ over $\mathcal{M}$, we have*

1) *Gaussian homogeneity:* $\mathrm{R}_B(X) \sim \mathcal{N}(\mathrm{R}_B(M), v^2)$;
2) *Sample homogeneity:* $\mathrm{FM}\{\mathrm{R}_B(P_i)\} = \mathrm{R}_B(\mathrm{FM}\{P_i\})$.

Chen Z, et al. LieBN: Batch Normalization over Lie Groups. under review

Theorem 5.3. *Given any SPD matrices P, Q and tangent vector $V \in T_P\mathcal{S}^n_{++}$, the Riemannian operators on $\{\mathcal{S}^n_{++}, g^{\mathrm{CRI}}\}$ are*

$$g^{\mathrm{CRI}}_P(V,V) = \left(\left\| \left(L(L^{-1}VL^{-\top})_{\frac{1}{2}} L^{-1} \right)_{\mathrm{Sym}} \right\|^{(\alpha,\beta)} \right)^2 \tag{20}$$

$$\mathrm{d}(P,Q) = \left\| \log \left(\tilde{Q}^{-\frac{1}{2}} \tilde{P} \tilde{Q}^{-\frac{1}{2}} \right) \right\|^{(\alpha,\beta)}, \tag{21}$$

$$\mathrm{Exp}_P(V) = \left(\mathrm{Exp}^{\mathrm{AI}}_{\tilde{P}} (-\bar{V}) \right)^{-1}_{\odot \mathrm{AI}}, \tag{22}$$

$$\mathrm{Log}_P(Q) = -\left(LL^\top \left(L\tilde{V}L^\top \right)^\top_{\frac{1}{2}} \right)_{\mathrm{Sym}}, \tag{23}$$

where L is the Cholesky factor of $P = LL^\top$, $(\cdot)^{-1}_{\odot \mathrm{AI}}$ is the group inverse, $\tilde{Q}$ and $\tilde{P}$ are the group inverses of P and Q, $\bar{V} = \left((L^{-1}VL^{-\top})_{\frac{1}{2}} L^{-1}L^{-\top} \right)_{\mathrm{Sym}}$, and $\tilde{V} = \mathrm{Log}^{\mathrm{AI}}_{\tilde{P}} (\tilde{Q})$. Here, $(X)_{\mathrm{Sym}} = X + X^\top, \forall X \in \mathbb{R}^{n \times n}$ denotes symmetrization, and $(\cdot)_{\frac{1}{2}}$ as its inverse map, namely $(X)_{\frac{1}{2}} = \lfloor X \rfloor + \frac{1}{2}\mathbb{X}$.

First non-trivial right-invariant SPD metric

```python
from LieBN import LieBNSPD, LieBNRot, LieBNCor
from LieBN.Geometry.SPD import SPDMatrices
from LieBN.Geometry.Rotations import
    RotMatrices
from LieBN.Geometry.Correlation import
    Correlation

# ==== SPD matrices ====
P_spd = SPDMatrices(n=5).random(4, 2, 5, 5)
# Implemented metrics: LEM,ALEM,LCM,AIM,CRIM
liebn_spd = LieBNSPD([2, 5, 5], metric="LEM",
    batchdim=[0])
output_spd = liebn_spd(P_spd)

# ==== SO(3) matrices ====
P_so3 = RotMatrices().random(4, 2, 3, 3, 3)
# LieBN-Left if is_left else -Right
liebn_so3 = LieBNRot([3, 3, 3], batchdim=[0,
    1], is_left=False)
output_so3 = liebn_so3(P_so3)

# ==== Correlation matrices ====
P_cor = Correlation(n=5).random(4, 2, 5, 5)
# Implemented metrics: ECM,LECM,OLM,LSM
liebn_cor = LieBNCor([2, 5, 5], metric="ECM",
    batchdim=[0])
output_cor = liebn_cor(P_cor)
```

Demo of the released toolbox

Manifold	SPD Manifold				Rotation Group	Correlation Manifold			
Metric	AIM	CRIM	LEM	LCM	Bi-invariant Metric	ECM	LECM	LSM	OLM
Invariance	Left	Right	Bi	Bi	Bi	Bi	Bi	Bi	Bi
Comutativity	✗	✗	✓	✓	✗	✓	✓	✓	✓
LieBN Type	Left	Right	Left=Right	Left=Right	Left & Right	Left=Right	Left=Right	Left=Right	Left=Right
Fréchet Mean	Karcher Flow	Karcher Flow	Closed Form	Closed Form	Karcher Flow	Closed Form	Closed Form	Closed Form	Closed Form

Nine implementations

EXPERIMENTS

Visualization
(at a different scale)

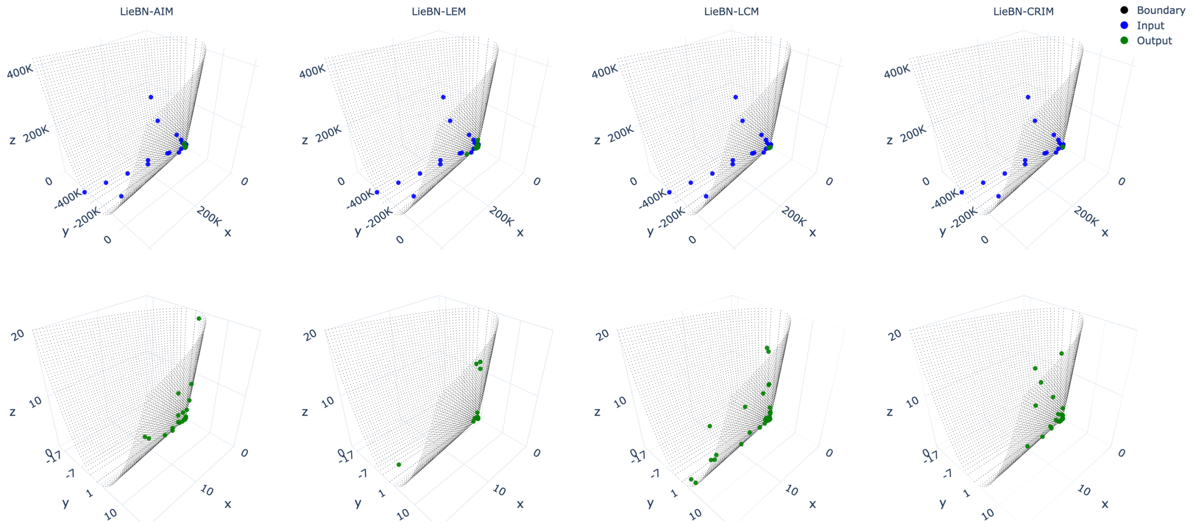


TABLE 9: Results of LieNet with or without rotation LieBN.

Method	G3D		HDM05		NTU60	
	Mean±STD	Max	Mean±STD	Max	2Blocks	3Blocks
LieNet	87.91±0.90	89.73	76.92±1.27	79.11	62.4	60.91
LieNetLieBN-Left	**88.88±1.62**	**90.67**	78.89±1.07	**80.88**	63.51	62.62
LieNetLieBN-Right	88.12±1.12	90.3	**79.39±1.13**	80.67	**63.6**	**62.72**

TABLE 10: Results of SPDNet with or without correlation LieBN under different invariant metrics.

Dataset	SPDNet	SPDNetLieBN-Cor			
		ECM	LECM	OLM	LSM
HDM05	59.13±0.67	**65.37 ± 1.07**	61.35 ± 0.34	60.33 ± 0.12	60.00 ± 0.27
FPHA	85.59±0.72	**87.20 ± 0.12**	87.03 ± 0.32	86.80 ± 0.12	86.77 ± 0.29

RBN: From Lie Groups to Gyro Groups

How about manifold without Lie group structures?

- Grassmannian
- Hyperbolic

GyroBN

Method	Controllable Statistics	Applied Geometries	Incorporated by GyroBN
SPDBN (Brooks et al., 2019)	M	SPD manifolds under AIM	✓
SPDBN (Kobler et al., 2022b)	M+V	SPD manifolds under AIM	✓
SPDDSMBN (Kobler et al., 2022a)	M+V	SPD manifolds under AIM	✓
ManifoldNorm (Chakraborty, 2020, Algs. 1-2)	N/A	Riemannian homogeneous space	✗
ManifoldNorm (Chakraborty, 2020, Algs. 3-4)	M+V	Matrix Lie groups under the distance $d(X,Y) = \left\|\operatorname{mlog}\left(X^{-1}Y\right)\right\|$	✓
RBN (Lou et al., 2020, Alg. 2)	N/A	Geodesically complete manifolds	✗
LieBN (Chen et al., 2024b)	M+V	General Lie groups	✓
GyroBN	M+V	Pseudo-reductive gyrogroups with gyro isometric gyrations	N/A

- Generality: incorporating several previous RBN methods

Chen Z, et al. Gyrogroup Batch Normalization. ICLR 2025

GYRO STRUCTURES

Gyrogroups

- Non-associative group
- Gyrations

Definition 2.1 (Gyrogroups (Ungar, 2009)). Given a nonempty set G with a binary operation $\oplus : G \times G \to G$, $\{G, \oplus\}$ forms a gyrogroup if its binary operation satisfies the following axioms for any $a, b, c \in G$:

(G1) There is at least one element $e \in G$ called a left identity (or neutral element) such that $e \oplus a = a$.

(G2) There is an element $\ominus a \in G$ called a left inverse of a such that $\ominus a \oplus a = e$.

(G3) There is an automorphism $\mathrm{gyr}[a, b] : G \to G$ for each $a, b \in G$ such that

$$a \oplus (b \oplus c) = (a \oplus b) \oplus \mathrm{gyr}[a, b]c \quad \text{(Left Gyroassociative Law).} \tag{1}$$

The automorphism $\mathrm{gyr}[a, b]$ is called the gyroautomorphism, or the gyration of G generated by a, b.
(G4) Left reduction law: $\mathrm{gyr}[a, b] = \mathrm{gyr}[a \oplus b, b]$.

Gyro Structures

- Extends the vector structures

As shown by Nguyen & Yang (2023), given P, Q and R in a manifold $\mathcal{M}$ and $t \in \mathbb{R}$, the gyro structures can be defined as:

$$\text{Gyro addition: } P \oplus Q = \mathrm{Exp}_P\left(\mathrm{PT}_{E \to P}\left(\mathrm{Log}_E(Q)\right)\right), \tag{3}$$
$$\text{Gyro scalar product: } t \odot P = \mathrm{Exp}_E\left(t \,\mathrm{Log}_E(P)\right), \tag{4}$$
$$\text{Gyro inverse: } \ominus P = -1 \odot P = \mathrm{Exp}_E\left(-\mathrm{Log}_E(P)\right), \tag{5}$$
$$\text{Gyration: } \mathrm{gyr}[P, Q]R = (\ominus(P \oplus Q)) \oplus (P \oplus (Q \oplus R)), \tag{6}$$
$$\text{Gyro inner product: } \langle P, Q\rangle_{\mathrm{gr}} = \langle \mathrm{Log}_E(P), \mathrm{Log}_E(Q)\rangle_E, \tag{7}$$
$$\text{Gyro norm: } \|P\|_{\mathrm{gr}} = \langle P, P\rangle_{\mathrm{gr}}, \tag{8}$$
$$\text{Gyrodistance: } \mathrm{d}_{\mathrm{gry}}(P, Q) = \|\ominus P \oplus Q\|_{\mathrm{gr}}, \tag{9}$$

where E is the gyro identity element, and Log_E and $\langle \cdot, \cdot \rangle_E$ is the Riemannian logarithm and metric at E. A bijection $\omega : G \to G$ is called gyroisometry, if it preserves the gyrodistance

$$\mathrm{d}_{\mathrm{gry}}(\omega(P), \omega(Q)) = \mathrm{d}_{\mathrm{gry}}(P, Q). \tag{10}$$

Table 2: Gyrogroup properties on several geometries. Related notations are defined in App. C.3.2.

Geometry	Symbol	$P \oplus Q$ or $x \oplus y$	E	$\ominus P$ or $\ominus x$	Lie group	Gyrogroup	References
AIM $\mathcal{S}^n_{++}$	$\oplus^{\mathrm{AI}}$	$P^{\frac{1}{2}} Q P^{\frac{1}{2}}$	I_n	P^{-1}	✗	✓	(Nguyen, 2022b)
LEM $\mathcal{S}^n_{++}$	$\oplus^{\mathrm{LE}}$	$\operatorname{mexp}(\operatorname{mlog}(P) + \operatorname{mlog}(Q))$	I_n	P^{-1}	✓	✓	(Arsigny et al., 2005) (Nguyen, 2022b)
LCM $\mathcal{S}^n_{++}$	$\oplus^{\mathrm{LC}}$	$\psi_{\mathrm{LC}}^{-1}(\psi_{\mathrm{LC}}(P) + \psi_{\mathrm{LC}}(Q))$	I_n	$\psi_{\mathrm{LC}}(-\psi_{\mathrm{LC}}(P))$	✓	✓	(Lin, 2019) (Nguyen, 2022b) (Chen et al., 2024d)
$\widetilde{\mathrm{Gr}}(p,n)$	$\widetilde{\oplus}^{\mathrm{Gr}}$	$\operatorname{mexp}(\Omega) Q \operatorname{mexp}(-\Omega)$	$\widetilde{I}_{p,n}$	$\operatorname{mexp}(-\Omega) \widetilde{I}_{p,n} \operatorname{mexp}(\Omega)$	✗	Non-reductive	(Nguyen, 2022a)
$\mathrm{Gr}(p,n)$	$\oplus^{\mathrm{Gr}}$	$\operatorname{mexp}(\Omega) V$	$I_{p,n}$	$\operatorname{mexp}(-\Omega) I_{p,n}$	✗	Non-reductive	(Nguyen & Yang, 2023)
$\mathcal{M}_K$	$\oplus_K$	$\frac{(1-2K\langle x,y\rangle - K\|y\|^2)x + (1+K\|x\|^2)y}{1-2K\langle x,y\rangle + K^2\|x\|^2\|y\|^2}$	0	$-x$	✗(✓for $K=0$)	✓	(Ungar, 2009) (Ganea et al., 2018) (Skopek et al., 2019)

- Several geometries do not form Lie groups, but gyrogroups.

GyroBN

Construction

$$\forall i \leq N, x_i \leftarrow \gamma \frac{x_i - \mu_b}{\sqrt{v_b^2 + \epsilon}} + \beta$$

Gyro →

$$\forall i \leq N, \tilde{P}_i = \overbrace{B \oplus}^{\text{Biasing}} \left(\overbrace{\frac{s}{\sqrt{v^2 + \epsilon}}}^{\text{Scaling}} \odot \left(\overbrace{\ominus M \oplus P_i}^{\text{Centering}} \right) \right),$$

Question

- Calculation of the gyro statistics
- Ability to normalize the sample statistics?

Slacked Definition

$\mathrm{gyr}[a, b] = \mathrm{gyr}[a \oplus b, b]$

Definition 3.1 (Pseudo-reductive Gyrogroups). A groupoid $\{G, \oplus\}$ is a pseudo-reductive gyrogroup if it satisfies axioms (G1), (G2), (G3) and the following pseudo-reductive law:

$$\mathrm{gyr}[X, P] = \mathbb{1}, \text{ for any left inverse } X \text{ of } P \text{ in } G, \tag{12}$$

where $\mathbb{1}$ is the identity map.

Proposition 3.2. [↓] $\mathrm{Gr}(p, n)$ *and* $\mathrm{Gr}(p, n)$ *form pseudo-reductive and gyrocommutative gyrogroups.*

↓

Results

Proposition 3.6. [↓] *For every (pseudo-reductive) gyrogroup in Tab. 2, the gyrodistance is identical to the geodesic distance (therefore symmetric). The gyroinverse, any gyration and any left gyrotranslation are gyroisometries.*

Insight:

- Gyro mean and variance ⇔ Riemannian mean and variance
- Gyro operations can normalize sample statistics

THEORIES

Guarantee

Theorem 4.1 (Homogeneity). [↓] *Supposing* $\{\mathcal{M}, \oplus\}$ *is a pseudo-reductive gyrogroup with any gyration* $\operatorname{gyr}[\cdot,\cdot]$ *as a gyroisometry, for* N *samples* $\{P_{i\ldots N} \in \mathcal{M}\}$*, we have the following properties:*

$$\textit{Homogeneity of gyromean: } \operatorname{FM}(\{B \oplus P_i\}) = B \oplus \operatorname{FM}(\{P_i\}), \forall B \in \mathcal{M}, \quad (16)$$

$$\textit{Homogeneity of dispersion from } E\text{: } \frac{1}{N}\sum_{i=1}^{N} \mathrm{d}^2_{\mathrm{gry}}(t \odot P_i, E) = \frac{t^2}{N}\sum_{i=1}^{N} \mathrm{d}^2_{\mathrm{gry}}(P_i, E), \quad (17)$$

Algorithm

Algorithm 1: Gyrogroup Batch Normalization (GyroBN)

Require : batch of activations $\{P_{1\ldots N} \in \mathcal{M}\}$, small positive constant ϵ, and momentum $\eta \in [0,1]$, running mean M_r, running variance v_r^2, biasing parameter $B \in \mathcal{M}$, scaling parameter $s \in \mathbb{R}$.

Return : normalized batch $\{\tilde{P}_{1\ldots N} \in \mathcal{M}\}$

```
if training then
    Compute batch mean M_b and variance v_b^2 of {P_1...N};
    Update running statistics M_r = Bar_γ(M_b, M_r), v_r^2 = γ v_b^2 + (1 − γ) v_r^2;
end
(M, v^2) = (M_b, v_b^2) if training else (M_r, v_r^2)
∀i ≤ N, P̃_i = B ⊕ ( s / sqrt(v^2 + ϵ) ⊙ (⊖M ⊕ P_i) )
```

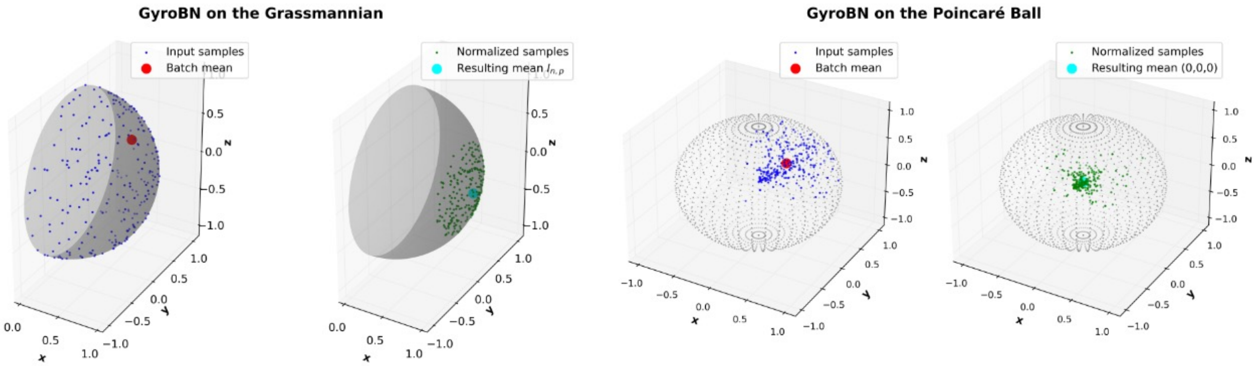
- Generality: incorporating several previous RBN methods

Instantiations

Table 9: Key operators in calculating GyroBN on the Grassmannian and hyperbolic manifolds. Here $P, Q \in \mathrm{Gr}(p,n)$ are two ONB Grassmannian points, while $x, y \in \mathbb{P}^n_K$ are two Poincaré vectors. Other notations follow from Tabs. 2 and 8.

Operator	$\mathrm{Gr}(p,n)$	$\mathbb{P}^n_K$
Identity element	$I_{p,n}$	$0 \in \mathbb{R}^n$
$P \oplus^{\mathrm{Gr}} Q$ or $x \oplus_K y$	$\mathrm{mexp}(\Omega)V$	$\frac{(1-2K\langle x,y\rangle - K\|y\|^2)x + (1+K\|x\|^2)y}{1-2K\langle x,y\rangle + K^2\|x\|^2\|y\|^2}$
$\ominus^{\mathrm{Gr}} P$ or $\ominus_K x$	$\mathrm{mexp}(-\Omega)I_{p,n}$	$-x$
$t \odot^{\mathrm{Gr}} P$ or $t \odot_K x$	$\mathrm{mexp}(t\Omega)I_{p,n}$	$\frac{1}{\sqrt{\lvert K\rvert}} \tanh\left(t \tanh^{-1}(\sqrt{\lvert K\rvert}\|x\|)\right) \frac{x}{\|x\|}$
$\mathrm{Bar}^{\mathrm{Gr}}_{\gamma}(Q,P)$ or $\mathrm{Bar}^{K}_{\gamma}(y,x)$	$\mathrm{Exp}^{\mathrm{Gr}}_P(\gamma \, \mathrm{Log}^{\mathrm{Gr}}_P(Q))$	$x \oplus_K (-x \oplus_K y) \odot_K t$
Fréchet Mean	Karcher Flow (Karcher, 1977)	(Lou et al., 2020, Alg. 1)

Visualization



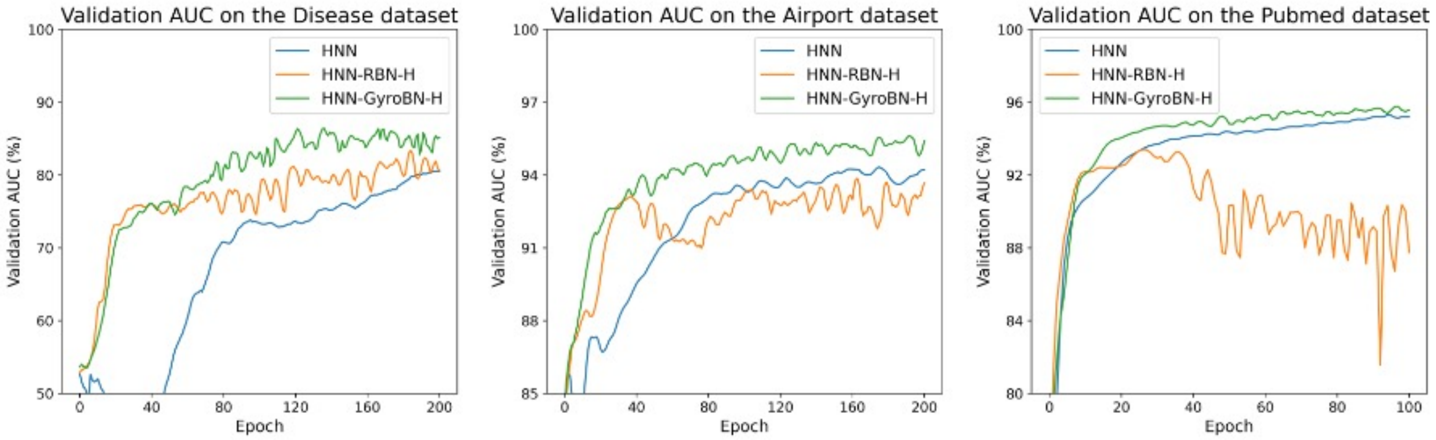
EXPERIMENTS

Grassmannian

Table 3: Comparison of GyroBN against other Grassmannian BNs under GyroGr backbone.

BN	None		ManifoldNorm-Gr		RBN-Gr		GyroBN-Gr	
Acc.	Mean±std	Max	Mean±std	Max	Mean±std	Max	Mean±std	Max
HDM05	48.97±0.24	49.23	49.67±0.76	50.41	48.64±0.77	49.49	**51.89±0.37**	**52.43**
NTU60	70.13±0.16	70.32	68.56±0.43	69.14	67.77±0.52	68.35	**72.60±0.04**	**72.65**
NTU120	53.76±0.18	53.96	51.41±0.38	51.92	50.56±0.22	50.82	**55.47±0.10**	**55.59**

Hyperbolic



GWBMBN for Ill-Conditioned SPD Matrix Learning

Ill-conditioning

- Overwhelming

Dataset	λ	Epochs	$> 10^2$	$> 10^3$	$> 10^4$	$> 10^5$	$> 10^6$
MAMEM-SSVEP-II	$1e^{-7}$	1	500 (100%)	399 (79.8%)	195 (39%)	74 (14.8%)	11 (2.2%)
		100	499 (99.8%)	428 (85.6%)	247 (49.4%)	115 (23%)	16 (3.2%)
		200 (Final)	500 (100%)	458 (91.6%)	274 (100%)	113 (54.8%)	13 (2.6%)
	$1e^{-6}$	1	489 (97.8%)	375 (75%)	191 (38.2%)	64 (12.8%)	0 (0%)
		100	496 (99.2%)	384 (76.8%)	201 (40.2%)	67 (13.4%)	1 (0.2%)
		200 (Final)	485 (97%)	370 (74%)	202 (40.4%)	71 (14.2%)	1 (0.2%)
	$1e^{-5}$	1	498 (99.6%)	379 (75.8%)	163 (32.6%)	24 (4.8%)	1 (0.2%)
		100	486 (97.2%)	302 (60.4%)	129 (25.8%)	21 (4.2%)	0 (0%)
		200 (Final)	459 (91.8%)	277 (55.4%)	113 (22.6%)	11 (2.2%)	0 (0%)
	$1e^{-4}$	1	404 (80.8%)	194 (38.8%)	99 (19.8%)	4 (0.8%)	0 (0%)
		100	406 (81.2%)	196 (39.2%)	43 (8.6%)	0 (0%)	0 (0%)
		200 (Final)	403 (80.6%)	205 (41%)	56 (11.2%)	1 (0.2%)	0 (0%)
HDM05	$1e^{-6}$	1	2086 (100%)	2086 (100%)	2086 (100%)	2086 (100%)	2047 (98.1%)
		100	2086 (100%)	2086 (100%)	2086 (100%)	2086 (100%)	2039 (97.7%)
		200 (Final)	2086 (100%)	2086 (100%)	2086 (100%)	2086 (100%)	2031 (97.4%)
NTU RGB+D	$1e^{-6}$	1	56,880 (100%)	56,880 (100%)	56,880 (100%)	56,880 (100%)	0 (0%)
		50	56,880 (100%)	56,880 (100%)	56,880 (100%)	56,880 (100%)	0 (0%)
		100 (Final)	56,880 (100%)	56,880 (100%)	56,880 (100%)	56,880 (100%)	0 (0%)

Table 7. Statistics (quantity and proportion) on the condition number ($\mathrm{eig}_{\max}/\mathrm{eig}_{\min}$) of the SPD features before the RBN layer on different datasets across various values of λ and training epochs.

Wang R, Jin S, **Chen Z†,** et al, Learning to Normalize on the SPD Manifold under Bures-Wasserstein Geometry, CVPR 2025

GWBMBN for Ill-Conditioned SPD Matrix Learning

AIM: quadratic

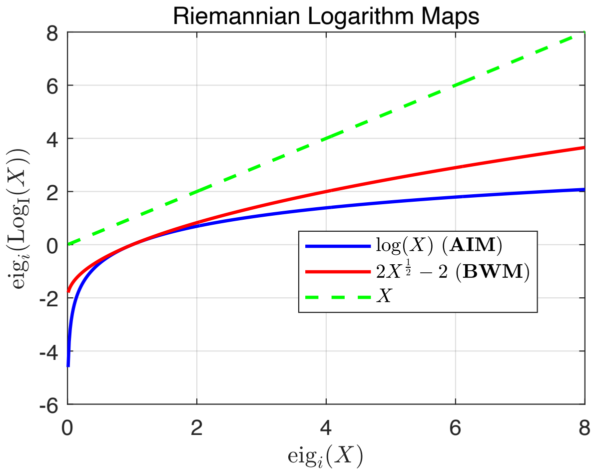
$$g^{\mathrm{AI}}_{\mathbf{X}_1}(\mathbf{S}_1, \mathbf{S}_2) = \operatorname{tr}\left(\mathbf{X}_1^{-1}\mathbf{S}_1\mathbf{X}_1^{-1}\mathbf{S}_2\right) = \operatorname{vec}(\mathbf{S}_1)^T(\mathbf{X}_1 \otimes \mathbf{X}_1)^{-1}\operatorname{vec}(\mathbf{S}_2)$$

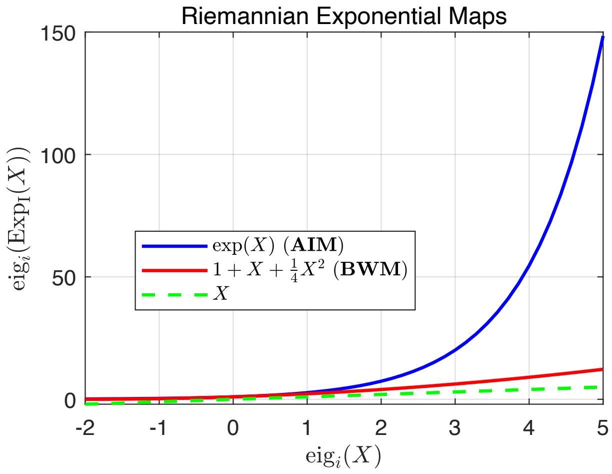
BWM: linear

$$g^{\mathrm{BW}}_{\mathbf{X}}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{2}\operatorname{tr}\left(\mathcal{L}_{\mathbf{X}}(\mathbf{S}_1)\mathbf{S}_2\right) = \frac{1}{2}\operatorname{vec}(\mathbf{S}_1)^T(\mathbf{X} + \mathbf{X})^{-1}\operatorname{vec}(\mathbf{S}_2)$$

Riemannian operators under the BWM
Riemannian metric
$g^{\mathrm{BW}}_{\mathbf{X}_1}(\mathbf{S}_1,\mathbf{S}_2) = \frac{1}{2}\operatorname{tr}(\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{S}_2) = \frac{1}{2}\operatorname{vec}(\mathbf{S}_1)^T(\mathbf{X}_1 \oplus \mathbf{X}_1)^{-1}\operatorname{vec}(\mathbf{S}_2)$
Riemannian distance
$d^2_{\mathrm{BW}}(\mathbf{X}_1,\mathbf{X}_2) = \operatorname{tr}(\mathbf{X}_1)+\operatorname{tr}(\mathbf{X}_2)-2\left(\operatorname{tr}\left(\mathbf{X}_1^{\frac{1}{2}}\mathbf{X}_2\mathbf{X}_1^{\frac{1}{2}}\right)\right)^{\frac{1}{2}}$
Riemannian geodesic
$\gamma_{\mathbf{X}_1,\mathbf{S}_1}(t) = \mathbf{X}_1 + t\mathbf{S}_1 + t^2\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{X}_1\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)$
Riemannian exponential map
$\operatorname{Exp}_{\mathbf{X}_1}\mathbf{S}_1 = \mathbf{X}_1 + \mathbf{S}_1 + \mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{X}_1\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)$
Riemannian logarithmic map
$\operatorname{Log}_{\mathbf{X}_1}\mathbf{X}_2 = (\mathbf{X}_2\mathbf{X}_1)^{\frac{1}{2}} + (\mathbf{X}_1\mathbf{X}_2)^{\frac{1}{2}} - 2\mathbf{X}_1$
Parallel Transport
$\Gamma_{\mathbf{X}_1\to\mathbf{X}_2}(\mathbf{S}) = \mathbf{U}\left[\sqrt{\frac{\boldsymbol{\delta}_i+\boldsymbol{\delta}_j}{\boldsymbol{\lambda}_i+\boldsymbol{\lambda}_j}}\,[\mathbf{S}']_{i,j}\right]_{i,j}\mathbf{U}^T$

Table 3. Basic operators based on the BWM.





GWBMBN for Ill-Conditioned SPD Matrix Learning

Construction

$$\forall i \leq N, x_i \leftarrow \gamma \frac{x_i - \mu_b}{\sqrt{v_b^2 + \epsilon}} + \beta$$

$$\text{Centering: } \bar{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\mathbf{I}_d}\left(\tilde{\Gamma}_{\mathcal{B}\to\mathbf{I}_d}\left(\widetilde{\mathrm{Log}}_{\mathcal{B}}(\mathbf{X}_i)\right)\right), \quad (19)$$

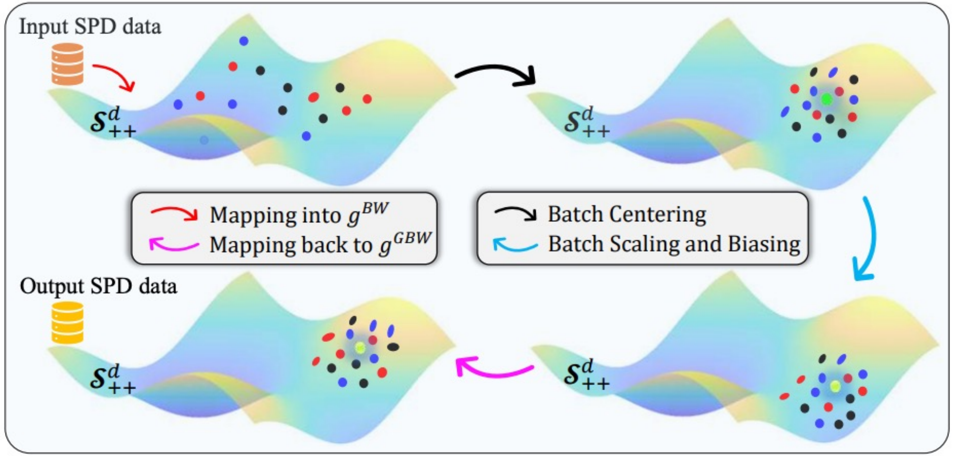
$$\text{Scaling: } \check{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\mathbf{I}_d}\left(\frac{\mathbf{s}}{\sqrt{\boldsymbol{\nu}^2+\epsilon}}(\widetilde{\mathrm{Log}}_{\mathbf{I}_d}(\bar{\mathbf{X}}_i))\right), \quad (20)$$

$$\text{Biasing: } \tilde{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\mathcal{G}}\left(\tilde{\Gamma}_{\mathbf{I}_d\to\mathcal{G}}\left(\widetilde{\mathrm{Log}}_{\mathbf{I}_d}(\check{\mathbf{X}}_i)\right)\right), \quad (21)$$

Simplification

$$g_X^{\mathrm{BW}}(S_1, S_2) = \frac{1}{2}\mathrm{tr}\left(\mathcal{L}_M(S_1)S_2\right)$$

$$g_X^{\mathrm{GBW}}(S_1, S_2) = \frac{1}{2}\mathrm{tr}\left(\mathcal{L}_{X,M}(S_1)S_2\right)$$



- Power-deformed generalized BWM is simplified by Riemannian isometries

GWBMBN for Ill-Conditioned SPD Matrix Learning

EEG

Models	Acc. (%)
EEGNet [34]	53.72 ± 7.23
ShallowConvNet [50]	56.93 ± 6.97
SCCNet [57]	62.11 ± 7.70
EEG-TCNet [28]	55.45 ± 7.66
FBCNet [37]	53.09 ± 5.67
TCNet-Fusion [40]	45.00 ± 6.45
MBEEGSE [1]	56.45 ± 7.27
SPDNet [27]	62.30 ± 3.12
SPDNet-BN(m) [7]	62.76 ± 3.01
SPDNet-BN(m+v) [33]	60.60 ± 3.57
SPDNet-Manifoldnorm [8]	62.08 ± 3.56
SPDNet-LieBN [14]	60.15 ± 3.42
SPDNet-GBWBN-($\theta = 1$)	**65.47 ± 3.33**

Table 4. Accuracy comparison on the MAMEM-SSVEP-II dataset.

RResNet

Models	HDM05		NTU RGB+D	
	Acc. (%)	Time (s/epoch)	Acc. (%)	Time (s/epoch)
RResNet [31]	61.09 ± 0.60	**3.62**	52.54 ± 0.59	**359.25**
RResNet-BN (m) [7]	63.31 ± 0.61	4.49	53.86 ± 1.19	478.62
RResNet-BN (m+v) [33]	64.51 ± 1.00	6.65	53.94 ± 1.28	546.15
RResNet-Manifoldnorm [8]	63.07 ± 0.80	6.24	53.50 ± 0.46	531.48
RResNet-LieBN [14]	66.43 ± 0.92	5.52	54.74 ± 0.75	523.19
RResNet-GBWBN-($\theta = 1$)	62.01 ± 1.23	8.21	59.45± 0.38	563.47
RResNet-GBWBN-($\theta = 0.5$)	**69.10 ± 0.83**	8.21	**59.72 ± 0.31**	563.47

Table 5. Accuracy comparison of different methods on the HDM05 and NTU RGB+D datasets.

Limitation

- Fail to normalize sample mean

$$\text{Centering: } \bar{\mathbf{X}}_i = \widetilde{\operatorname{Exp}}_{\mathbf{I}_d}\left(\tilde{\Gamma}_{\mathcal{B}\to\mathbf{I}_d}\left(\widetilde{\operatorname{Log}}_{\mathcal{B}}(\mathbf{X}_i)\right)\right), \quad (19)$$

$$\text{Scaling: } \check{\mathbf{X}}_i = \widetilde{\operatorname{Exp}}_{\mathbf{I}_d}\left(\frac{\mathbf{s}}{\sqrt{\nu^2+\epsilon}}\left(\widetilde{\operatorname{Log}}_{\mathbf{I}_d}(\bar{\mathbf{X}}_i)\right)\right), \quad (20)$$

$$\text{Biasing: } \tilde{\mathbf{X}}_i = \widetilde{\operatorname{Exp}}_{\mathcal{G}}\left(\tilde{\Gamma}_{\mathbf{I}_d\to\mathcal{G}}\left(\widetilde{\operatorname{Log}}_{\mathbf{I}_d}(\check{\mathbf{X}}_i)\right)\right), \quad (21)$$

Solution

- Invariance (in progress)

Riemannian Classifiers

Overview: Riemannian Multinomial Logistics Regression

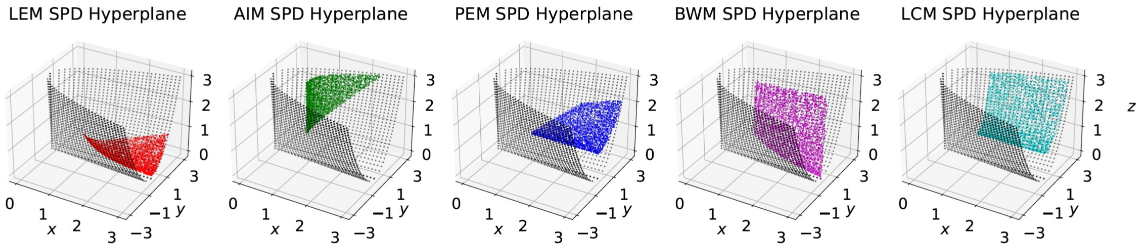


Figure 2: Conceptual illustration of SPD hyperplanes induced by five families of Riemannian metrics. The black dots denote the boundary of $\mathcal{S}^2_{++}$.

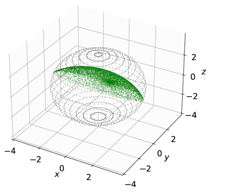
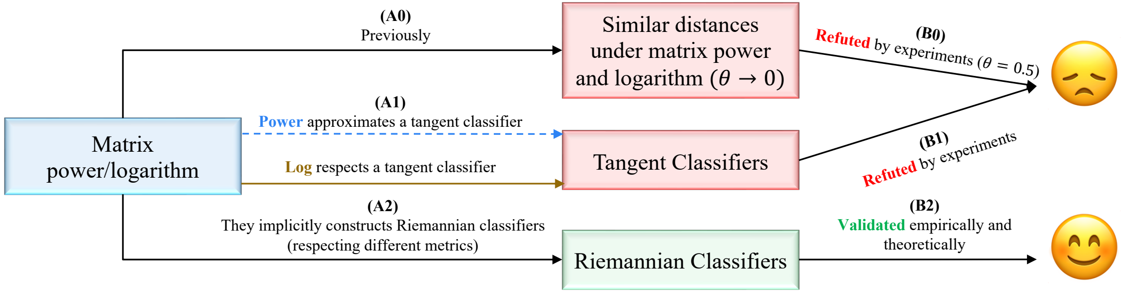


Figure 3: Conceptual illustration of a Lie hyperplane. Each pair of antipodal black dots corresponds to a rotation matrix with an Euler angle of π, while the green dots denote a Lie hyperplane.

Flat MLR (CVPR24) ⇒ RMLR (NeurIPS24)



RieGCP (ICLR25)

Chen Z, et al. Riemannian multinomial logistics regression for SPD neural networks. CVPR 2024
Chen Z, et al. RMLR: Extending Multinomial Logistic Regression into General Geometries. NeurIPS 2024
Chen Z, et al. Understanding matrix function normalizations in covariance pooling through the lens of Riemannian geometry. ICLR 2025

Riemannian Multinomial Logistics Regression

Euclidean

$y = Ax + b$

$$\forall k \in \{1, \ldots, C\}, p(y = k \mid x) \propto \exp\left((\langle a_k, x \rangle - b_k)\right)$$

Margin distance to hyperplane

$$p(y = k \mid x) \propto \exp(\operatorname{sign}(\langle a_k, x - p_k \rangle) \|a_k\| d(x, H_{a_k, p_k}))$$

$$H_{a_k, p_k} = \{x \in \mathbb{R}^n : \langle a_k, x - p_k \rangle = 0\}$$

SPD

Definition 3.1 (SPD hyperplanes). Given $P \in \mathcal{S}^n_{++}, A \in T_P\mathcal{S}^n_{++} \backslash \{\mathbf{0}\}$, we define the SPD hyperplane as

$$\tilde{H}_{A,P} = \{S \in \mathcal{S}^n_{++} : g_P(\operatorname{Log}_P S, A) = \langle \operatorname{Log}_P S, A \rangle_P = 0\}, \tag{12}$$

where P and A are referred to as shift and normal matrices, respectively.

Definition 3.2 (SPD MLR). SPD MLR is defined as

$$p(y = k \mid S) \propto \exp(\operatorname{sign}(\langle A_k, \operatorname{Log}_{P_k}(S) \rangle_{P_k}) \|A_k\|_{P_k} d(S, \tilde{H}_{A_k, P_k})), \tag{13}$$

where $P_k \in \mathcal{S}^n_{++}, A_k \in T_{P_k}\mathcal{S}^n_{++} \backslash \{\mathbf{0}\}, \langle \cdot, \cdot \rangle_{P_k} = g_{P_k}$, and $\|\cdot\|_{P_k}$ is the norm on $T_{P_k}\mathcal{S}^n_{++}$ induced by g at P_k, and $\tilde{H}_{A_k, P_k}$ is a margin hyperplane in $\mathcal{S}^n_{++}$ as defined in Eq. (12). $d(S, \tilde{H}_{A_k, P_k})$ denotes the margin distance between S and SPD hyperplane $\tilde{H}_{A_k, P_k}$, which is formulated as:

$$d(S, \tilde{H}_{A_k, P_k})) = \inf_{Q \in \tilde{H}_{A_k, P_k}} d(S, Q), \tag{14}$$

Crux

where $d(S, Q)$ is the geodesic distance induced by g.

Flat Geom.

Lemma 3.5. *Given a PEM g, the margin distance defined in Eq. (14) has a closed-form solution:*

$$d(S, \tilde{H}_{A_k,P_k})) = d(\phi(S), H_{\phi_{*,P_k}(A_k),\phi(P_k)}), \tag{15}$$

$$= \frac{|\langle \phi(S) - \phi(P_k), \phi_{*,P_k}(A_k)\rangle|}{\|A_k\|_{P_k}}, \tag{16}$$

where $|\cdot|$ is the absolute value.

Theorem 3.8 (SPD MLR under a PEM). *Under any PEM, SPD MLR and SPD hyperplane is*

$$p(y = k \mid S) \propto \exp(\langle \phi(S) - \phi(P_k), \phi_{*,I}(\tilde{A}_k)\rangle), \tag{19}$$

$$\tilde{H}_{\tilde{A}_k,P_k} = \{S \in \mathcal{S}^n_{++} : \langle \phi(S) - \phi(P_k), \phi_{*,I}(\tilde{A}_k)\rangle = 0\}, \tag{20}$$

where $\tilde{A}_k \in T_I\mathcal{S}^n_{++}/\{0\} \cong \mathcal{S}^n/\{0\}$ is a symmetric matrix, and $P_k \in \mathcal{S}^n_{++}$ is an SPD matrix.

SPD

Corollary 4.1 (SPD MLRs under the deformed LEM and LCM). *The SPD MLRs under (α, β)-LEM is*

$$p(y = k \mid S) \propto \exp\left[\langle \operatorname{mlog}(S) - \operatorname{mlog}(P_k), \tilde{A}_k\rangle^{(\alpha,\beta)}\right], \tag{21}$$

where $\tilde{A}_k \in T_I\mathcal{S}^n_{++} \cong \mathcal{S}^n$ and $P_k \in \mathcal{S}^n_{++}$. The SPD MLRs under (θ)-LCM is

$$p(y = k \mid S) \propto \exp\left[\frac{1}{\theta}\langle \lfloor\tilde{K}\rfloor - \lfloor\tilde{L}_k\rfloor + \left[\operatorname{Dlog}(\mathbb{D}(\tilde{K})) - \operatorname{Dlog}(\mathbb{D}(\tilde{L}_k))\right], \lfloor\tilde{A}_k\rfloor + \frac{1}{2}\mathbb{D}(\tilde{A}_k)\rangle\right], \tag{22}$$

where $\tilde{K} = \operatorname{Chol}(S^\theta)$, $\tilde{L}_k = \operatorname{Chol}(P_k^\theta)$, and $\mathbb{D}(\tilde{A}_k)$ denotes a diagonal matrix with diagonal elements of $\tilde{A}_k$.

Question

- What about in other geometries?

REFORMULATION BY RIEMANNIAN TRIGONOMETRY

$$p(y=k \mid x) \propto \exp\left(\operatorname{sign}(\langle a_k, x - p_k \rangle)\|a_k\| d(x, H_{a_k,p_k})\right),$$
$$H_{a_k,p_k} = \{x \in \mathbb{R}^n : \langle a_k, x - p_k \rangle = 0\},$$

Reformulation

$$d(x, H_{a,p})) = \sin(\angle xpy^*) d(x,p), \quad \text{with } y^* = \underset{y \in H_{a,p} \setminus \{p\}}{\arg\max} (\cos \angle xpy). \tag{7}$$

Riemannian trigonometry

Definition 3.1 (Riemannian Margin Distance)**.** Let $\tilde{H}_{\tilde{A},P}$ be a Riemannian hyperplane defined in Eq. (5), and $S \in \mathcal{M}$. The Riemannian margin distance from S to $\tilde{H}_{\tilde{A},P}$ is defined as

$$d(S, \tilde{H}_{\tilde{A},P}) = \sin(\angle SPY^*) d(S,P), \tag{8}$$

where $d(S,P)$ is the geodesic distance, and $Y^* = \operatorname{argmax}(\cos \angle SPY)$ with $Y \in \tilde{H}_{\tilde{A},P} \setminus \{P\}$. The initial velocities of geodesics define $\cos \angle SPY$:

$$\cos \angle SPY = \frac{\langle \operatorname{Log}_P Y, \operatorname{Log}_P S \rangle_P}{\| \operatorname{Log}_P Y \|_P, \| \operatorname{Log}_P S \|_P}, \tag{9}$$

where $\langle \cdot, \cdot \rangle_P$ is the Riemannian metric at P, and $\| \cdot \|_P$ is the associated norm.

Margin Distance

Theorem 3.2. [↓] *The Riemannian margin distance defined in Def. 3.1 is given as*

$$d(S, \tilde{H}_{\tilde{A},P}) = \frac{|\langle \operatorname{Log}_P S, \tilde{A} \rangle_P|}{\|\tilde{A}\|_P}. \quad (10)$$

Putting the Eq. (10) into Eq. (4), we can a closed-form expression for Riemannian MLR.

MLR

Theorem 3.3 (RMLR). [↓] *Given a Riemannian manifold $\{\mathcal{M}, g\}$, the Riemannian MLR induced by g is*

$$p(y = k \mid S \in \mathcal{M}) \propto \exp\left(\langle \operatorname{Log}_{P_k} S, \tilde{A}_k \rangle_{P_k}\right), \quad (11)$$

where $P_k \in \mathcal{M}$, $\tilde{A}_k \in T_{P_k}\mathcal{M} \backslash \{\mathbf{0}\}$, and Log *is the Riemannian logarithm.*

Generality

Table 1: Several MLRs on different geometries are special cases of our MLR.

MLR	Geometries	Requirements	Incorporated by Our MLR
Euclidean MLR (Eq. (1))	Euclidean geometry	N/A	✓(App. C)
Gyro SPD MLRs [50]	AIM, LEM & LCM on $\mathcal{S}^n_{++}$	Gyro structures	✓(Rem. 4.3)
Gyro SPSD MLRs [51]	SPSD product gyro spaces	Gyro structures	✓(App. D)
Flat SPD MLRs [16]	(α, β)-LEM & (θ)-LCM on $\mathcal{S}^n_{++}$	Pullback metrics from the Euclidean space	✓(Rem. 4.3)
Ours	General Geometries	Riemannian logarithm	N/A

SPD MLR

Theorem 4.2 (SPD MLRs). [↓] *By abuse of notation, we omit the subscripts k of A_k and P_k. Given SPD feature S, the SPD MLRs, $p(y=k \mid S \in \mathcal{S}^n_{++})$, are proportional to*

$$(\alpha,\beta)\text{-}LEM : \exp\left[\langle \log(S) - \log(P), A\rangle^{(\alpha,\beta)}\right], \tag{16}$$

$$(\theta,\alpha,\beta)\text{-}AIM : \exp\left[\frac{1}{\theta}\langle \log(P^{-\frac{\theta}{2}} S^\theta P^{-\frac{\theta}{2}}), A\rangle^{(\alpha,\beta)}\right], \tag{17}$$

$$(\theta,\alpha,\beta)\text{-}EM : \exp\left[\frac{1}{\theta}\langle S^\theta - P^\theta, A\rangle^{(\alpha,\beta)}\right], \tag{18}$$

$$\theta\text{-}LCM : \exp\left[\frac{1}{\theta}\langle \lfloor \tilde{K} \rfloor - \lfloor \tilde{L} \rfloor + \left[\operatorname{Dlog}(\mathbb{D}(\tilde{K})) - \operatorname{Dlog}(\mathbb{D}(\tilde{L}))\right], \lfloor A \rfloor + \frac{1}{2}\mathbb{D}(A)\rangle\right], \tag{19}$$

$$2\theta\text{-}BWM : \exp\left[\frac{1}{4\theta}\langle (P^{2\theta} S^{2\theta})^{\frac{1}{2}} + (S^{2\theta} P^{2\theta})^{\frac{1}{2}} - 2P^{2\theta}, \mathcal{L}_{P^{2\theta}}(\bar{L} A \bar{L}^\top)\rangle\right], \tag{20}$$

where $A \in T_I\mathcal{S}^n_{++} \backslash \{0\}$ is a symmetric matrix, $\log(\cdot)$ is the matrix logarithm, $\mathcal{L}_P(V)$ is the solution to the matrix linear system $\mathcal{L}_P[V]P + P\mathcal{L}_P[V] = V$, known as the Lyapunov operator, $\operatorname{Dlog}(\cdot)$ is the diagonal element-wise logarithm, $\lfloor \cdot \rfloor$ is the strictly lower part of a square matrix, and $\mathbb{D}(\cdot)$ is a diagonal matrix with diagonal elements of a square matrix. Besides, $\log_{*,P}$ is the differential maps at P, $\tilde{K} = \operatorname{Chol}(S^\theta)$, $\tilde{L} = \operatorname{Chol}(P^\theta)$, and $\bar{L} = \operatorname{Chol}(P^{2\theta})$.

Lie MLR

Theorem 5.2. [↓] *The Lie MLR on $\mathrm{SO}(n)$ is given as*

$$p(y=k \mid R \in \mathrm{SO}(n)) \propto \langle \log(P_k^\top S), A_k\rangle, \tag{22}$$

where $P_k \in \mathrm{SO}(n)$ and $A_k \in \mathfrak{so}(n)$.

Results

Riem. FFNN

Table 4: Comparison of SPDNet with LogEig against SPD MLRs on the HDM05 dataset.

Architectures	LogEig MLR	(θ,α,β)-AIM	(θ,α,β)-EM		(α,β)-LEM	2θ-BWM	θ-LCM	
		(1,1,0)	(1,1,0)	(0.5,1.0,1/30)	(1,1,0)	(0.5)	(1)	(0.5)
1-Block	57.42±1.31	58.07±0.64	66.32±0.63	**71.65±0.88**	56.97±0.61	**70.24±0.92**	63.84±1.31	**65.66±0.73**
2-Block	60.69±0.66	60.72±0.62	66.40±0.87	**70.56±0.39**	60.69±1.02	**70.46±0.71**	62.61±1.46	**65.79±0.63**
3-Block	60.76±0.80	61.14±0.94	66.70±1.26	**70.22±0.81**	60.28±0.91	**70.20±0.91**	62.33±2.15	**65.71±0.75**

Riem. GCN

Table 8: Comparison of LogEig against SPD MLRs under the SPDGCN architecture.

Classifiers	Disease		Cora		Pubmed	
	Mean±STD	Max	Mean±STD	Max	Mean±STD	Max
LogEig MLR	90.55 ± 4.83	96.85	78.04 ± 1.27	79.6	70.99 ± 5.12	77.6
(θ,α,β)-AIM	94.84 ± 2.27	98.43	79.79 ± 1.44	81.6	77.83 ± 1.08	**80**
(θ,α,β)-EM	90.87 ± 5.14	98.03	79.05 ± 1.23	81	78.16 ± 2.41	79.5
(α,β)-LEM	**96.33 ± 2.19**	**98.82**	**79.89 ± 0.99**	81.8	**78.16 ± 2.41**	79.5
2θ-BWM	91.93 ± 3.64	96.85	73.46 ± 2.18	77.7	73.22 ± 4.06	78.1
θ-LCM	93.01 ± 2.14	98.43	77.59 ± 1.20	80.1	74.46 ± 5.81	78.9

RResNet

Table 7: Comparison of LogEig against SPD MLRs under the RResNet architecture.

Datasets	LogEigMLR	(θ,α,β)-AIM	(θ,α,β)-EM	(α,β)-LEM	2θ-BWM	θ-LCM
HDM05	58.17 ± 2.07	60.23 ± 1.26	**71.89 ± 0.60 (↑ 13.72)**	59.44 ± 0.87	69.85 ± 0.23	65.76 ± 0.96
NTU60	45.22 ± 1.23	48.94 ± 0.68	52.24 ± 1.25	46.99 ± 0.41	50.56 ± 0.59	**53.63 ± 0.95 (↑ 8.41)**

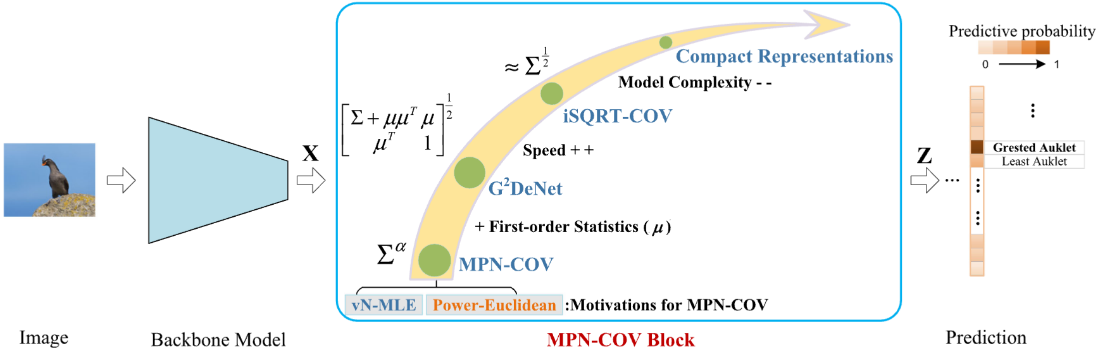
SO(n)

Table 10: Results of LogEig MLR against Lie MLR under the LieNet architecture.

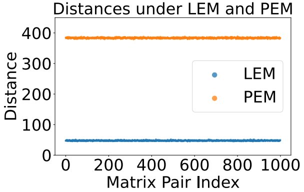
Classifiers	G3D		HDM05	
	Mean±STD	Max	Mean±STD	Max
LogEig MLR	87.91±0.90	89.73	76.92±1.27	79.11
Lie MLR	**89.13±1.7**	**92.12**	**78.24±1.03**	**80.25**

UNDERSTANDING MATRIX FUNCTIONS IN GCP

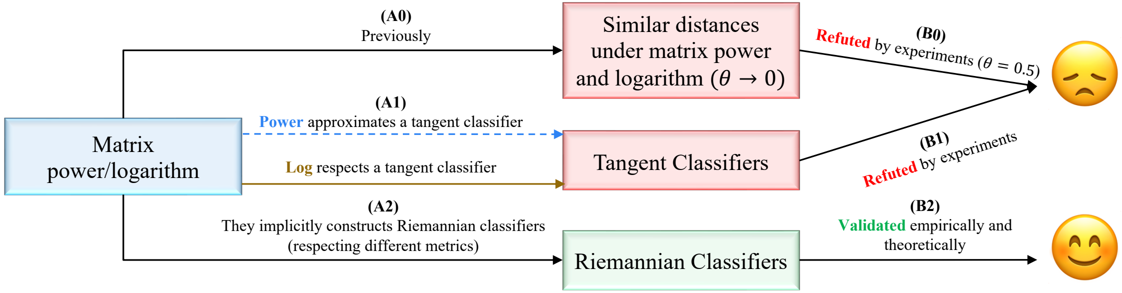
Typical GCP



Discrepancy



Explanation



Chen Z, et al. Understanding matrix function normalizations in covariance pooling through the lens of Riemannian geometry. ICLR 2025

Seven power-deformed metrics on SPD the manifold.

Name	Riemannian Metric $g_P(V, W)$	Riemannian Logarithm $\mathrm{Log}_P\, Q$	Deformation ($\theta \neq 0$)
(α,β)-LEM (Thanwerdas & Pennec, 2023)	$\langle \mathrm{mlog}_{*,P}(V), \mathrm{mlog}_{*,P}(W)\rangle^{(\alpha,\beta)}$	$(\mathrm{mlog}_{*,P})^{-1}\left[\mathrm{mlog}(Q) - \mathrm{mlog}(P)\right]$	$\frac{1}{\theta^2}\,\mathrm{Pow}_\theta^*\, g^{(\alpha,\beta)\text{-LE}}$
(α,β)-AIM (Thanwerdas & Pennec, 2023)	$\langle P^{-1}V, WP^{-1}\rangle^{(\alpha,\beta)}$	$P^{1/2}\,\mathrm{mlog}\left(P^{-1/2}QP^{-1/2}\right)P^{1/2}$	$\frac{1}{\theta^2}\,\mathrm{Pow}_\theta^*\, g^{(\alpha,\beta)\text{-AI}}$
(α,β)-EM (Thanwerdas & Pennec, 2023)	$\langle V, W\rangle^{(\alpha,\beta)}$	$Q - P$	$\frac{1}{\theta^2}\,\mathrm{Pow}_\theta^*\, g^{(\alpha,\beta)\text{-E}}$
(θ_1,θ_2)-EM (Thanwerdas & Pennec, 2022)	$\frac{1}{\theta_1\theta_2}\langle \mathrm{Pow}_{\theta_1 *,P}(V), \mathrm{Pow}_{\theta_2 *,P}(W)\rangle$	$(\mathrm{Pow}_{\theta *,P})^{-1}(Q^\theta - P^\theta)$, with $\theta = (\theta_1+\theta_2)/2$	N/A
LCM (Lin, 2019)	$\sum_{i>j}\tilde{V}_{ij}\tilde{W}_{ij} + \sum_{j=1}^{n}\tilde{V}_{jj}\tilde{W}_{jj}L_{jj}^{-2}$	$(\mathrm{Chol}^{-1})_{*,L}\left[\lfloor K\rfloor - \lfloor L\rfloor + \mathbb{D}(L)\,\mathrm{Dlog}(\mathbb{D}(L)^{-1}\mathbb{D}(K))\right]$	$\frac{1}{\theta^2}\,\mathrm{Pow}_\theta^*\, g^{\mathrm{LC}}$
BWM (Bhatia et al., 2019)	$\frac{1}{2}\langle \mathcal{L}_P[V], W\rangle$	$(PQ)^{1/2} + (QP)^{1/2} - 2P$	$\frac{1}{4\theta^2}\,\mathrm{Pow}_{2\theta}^*\, g^{\mathrm{BW}}$
GBWM (Han et al., 2023)	$\frac{1}{2}\langle \mathcal{L}_{P,M}[V], W\rangle$	$M\left(M^{-1}PM^{-1}Q\right)^{1/2} + \left(QM^{-1}PM^{-1}\right)^{1/2}M - 2P$	$\frac{1}{4\theta^2}\,\mathrm{Pow}_{2\theta}^*\, g^{M\text{-BW}}$

Matrix logarithm and power:

$$\text{Log-EMLR: } \mathrm{softmax}\left(\mathcal{F}\left(f_{\mathrm{vec}}\left(\mathrm{mlog}(S)\right); A, b\right)\right),$$

$$\text{Pow-EMLR: } \mathrm{softmax}\left(\mathcal{F}\left(f_{\mathrm{vec}}\left(S^\theta\right); A, b\right)\right),$$

Tangent Perspective

Motivation

Matrix logarithm as the Riem. log

Riem. log

Table 2: Log_I under seven families of metrics. $\theta_0 = \frac{\theta_1+\theta_2}{2}$ for (θ_1, θ_2)-EM, $\theta_0 = \theta$ for (θ, α, β)-EM and 2θ-BWM, and $(2\theta, \phi_{2\theta}(P))$-BWM.

Metric	$\mathrm{Log}_I P$	Metric	$\mathrm{Log}_I P$
(α, β)-LEM (θ, α, β)-AIM	$\mathrm{mlog}(P)$	(θ, α, β)-EM (θ_1, θ_2)-EM 2θ-BWM $(2\theta, P^{2\theta})$-BWM	$\frac{1}{\theta_0}(P^{\theta_0} - I)$
θ-LCM	$\frac{1}{\theta}\left[\lfloor \tilde{L} \rfloor + \lfloor \tilde{L} \rfloor^\top + 2\,\mathrm{Dlog}(\mathbb{D}(\tilde{L}))\right]$		

Refuted

Table 3: Results of GCP on the ImageNet-1k and Cars datasets with Pow-TMLR or Pow-EMLR under the architecture of ResNet-18.

Method	ImageNet-1k Top-1 Acc (%)	ImageNet-1k Top-5 Acc (%)	Cars Top-1 Acc (%)	Cars Top-5 Acc (%)
Pow-TMLR	71.62	89.73	51.14	74.29
Pow-EMLR	**73**	**90.91**	**80.43**	**94.15**

Motivation

Matrix logarithm implicitly construct an SPD MLR (CVPR24)

SPD MLR

LEM-based: $\exp\left[\langle \log(S) - \log(P_k), A_k \rangle\right]$,

PEM-based: $\exp\left[\frac{1}{\theta}\langle S^\theta - P_k^\theta, A_k \rangle\right]$,

RMLR

Theorem 2. [↓] *Under PEM with $\theta > 0$, optimizing each SPD parameter P_k in Eq. (16) by PEM-based RSGD and Euclidean parameter A_k by Euclidean SGD, the PEM-based SPD MLR is equivalent to a Euclidean MLR illustrated in Eq. (10) in the co-domain of $\phi_\theta(\cdot) : \mathcal{S}^n_{++} \rightarrow \mathcal{S}^n_{++}$, defined as*

$$\phi_\theta(S) = \frac{1}{\theta} S^\theta, \theta > 0, \forall S \in \mathcal{S}^n_{++}. \tag{17}$$

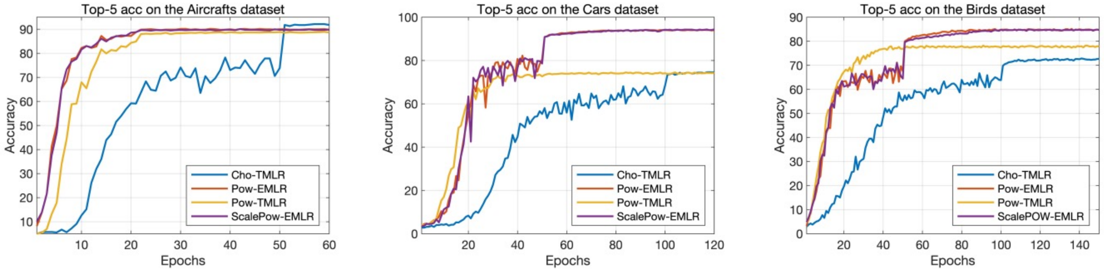
Comparison

	Log-EMLR	Pow-EMLR	ScalePow-EMLR	Pow-TMLR	Cho-TMLR
Expression	$f_s(\mathcal{F}(f_{\text{vec}}(\text{mlog}(S))))$	$f_s(\mathcal{F}(f_{\text{vec}}(S^\theta)))$ $(\theta > 0)$	$f_s(\mathcal{F}(f_{\text{vec}}(\frac{1}{\theta}S^\theta)))$ $(\theta > 0)$	$f_s\left(\mathcal{F}\left(f_{\text{vec}}\left(\frac{1}{\theta_0}(S^{\theta_0} - I)\right)\right)\right)$	$f_s\left(\mathcal{F}\left(f_{\text{vec}}\left(\tilde{V}\right)\right)\right)$
Explanation	SPD MLR	SPD MLR	SPD MLR	Tangent Classifier	Tangent Classifier
Metrics	LEM	$(\theta, 1, 0)$-EM	$(\theta, 1, 0)$-EM	(θ, α, β)-EM, (θ_1, θ_2)-EM, 2θ-BWM, $(2\theta, \phi_{2\theta}(S))$-BWM	θ-LCM
Used in GCP	✓(Eq. (4))	✓ ($\theta = 0.5$ in Eq. (5))	✗	✗	✗
Reference	(Chen et al., 2024a, Prop. 5.1)	Thm. 2	Thm. 2	Tab. 2	Tab. 2

Results

Table 5: Results of iSQRT-COV on four datasets with different covariance matrix classifiers. The backbone network on ImageNet is ResNet-18, while the one on the other three FGVC datasets is ResNet-50. Power is set to be 1/2 for Pow-TMLR, ScalePow-EMLR and Pow-EMLR.

Classifier	ImageNet-1k		Aircrafts		Birds		Cars	
	Top-1 Acc (%)	Top-5 Acc (%)	Top-1 Acc (%)	Top-5 Acc (%)	Top-1 Acc (%)	Top-5 Acc (%)	Top-1 Acc (%)	Top-5 Acc (%)
Cho-TMLR	N/A	N/A	78.97	91.81	48.07	72.59	51.06	74.33
Pow-TMLR	71.62	89.73	69.58	88.68	52.97	77.80	51.14	74.29
ScalePow-EMLR	72.43	90.44	71.05	89.86	63.48	84.69	80.31	94.07
Pow-EMLR	73	90.91	72.07	89.83	63.29	84.66	80.43	94.15



Ablations

(a) Results of different powers under the ResNet-50.

Classifier	Aircrafts		Cars	
	Top-1 Acc (%)	Top-5 Acc (%)	Top-1 Acc (%)	Top-5 Acc (%)
Pow-TMLR-0.25	65.41	86.71	41.47	66.66
ScalePow-EMLR-0.25	**72.76**	**90.31**	61.78	84.04
Pow-EMLR-0.25	71.47	90.04	**62.88**	84.14
Pow-TMLR-0.5	67.9	88.75	55.01	77.95
ScalePow-EMLR-0.5	74.29	91.12	62.42	84.82
Pow-EMLR-0.5	74.17	91.21	62.83	84.85
Pow-TMLR-0.7	65.92	87.49	50.68	74.12
ScalePow-EMLR-0.7	**74.26**	**91.15**	**64.22**	**83.67**
Pow-EMLR-0.7	74.17	90.49	61.41	82.39

(b) Results under the AlexNet.

Dataset	Result	Pow-TMLR	Pow-EMLR
Aircrafts	Top-1 Acc (%)	38.01	**65.02**
	Top-5 Acc (%)	74.4	**87.79**
Cars	Top-1 Acc (%)	28.57	**59.13**
	Top-5 Acc (%)	59.51	**82.04**

Thanks

Thanks!

WeChat

LinkedIn

Homepage

UNIVERSITÀ
DI TRENTO