A Lie Group Approach to Riemannian Batch Normalization

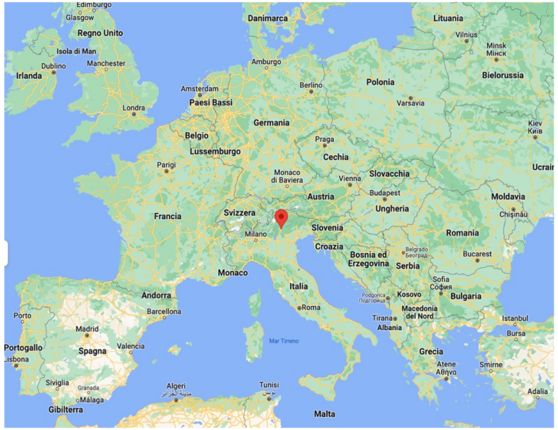
– ICLR 2024 –

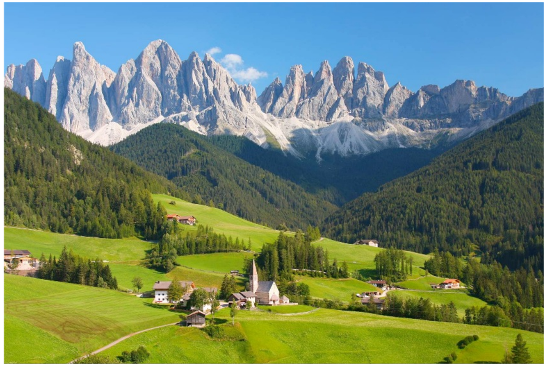
Ziheng Chen[1], Yue Song[1], Yunmei Liu[2], Nicu Sebe[1]

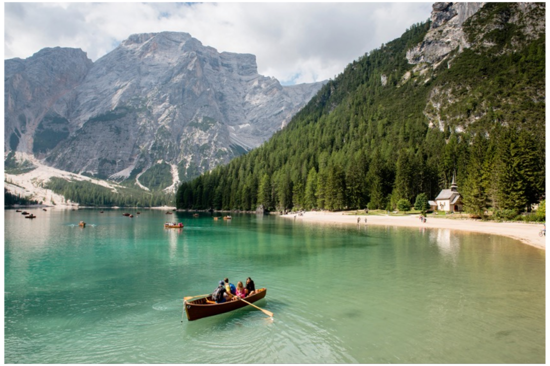
1 Dept. of Information Engineering and Computer Science, University of Trento
2 Industrial Engineering Department, University of Louisville

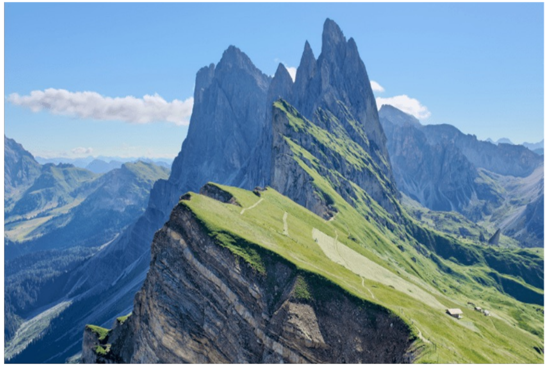
Trento

- German-Italian autonomous province of north Italy
- Barycenter of Europe and Alps
- Dolomities Mountains (UNESCO World Heritage)









LieBN: from Euclidean to Lie Groups

Euclidean Batch Normalization: facilitating network training by controlling mean and variance

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

Batch Normalization on manifolds:

- LieBN on general Lie groups
- LieBN on SPD manifolds under three deformed Lie groups
- Preliminary experiments on rotation matrices

Preliminaries on Riemannian Geometry

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.

Pullback:

Definition 2.3 (Pullback Metrics). Suppose $\mathcal{M}_1, \mathcal{M}_2$ are smooth manifolds, g is a Riemannian metric on $\mathcal{M}_2$, and $f : \mathcal{M}_1 \to \mathcal{M}_2$ is smooth. Then the pullback of g by f is defined point-wisely,

$$(f^*g)_p(V, W) = g_{f(p)}(f_{*,p}(V), f_{*,p}(W)), \tag{2}$$

where $p \in \mathcal{M}$, $f_{*,p}(\cdot)$ is the differential map of f at p, and $V, W \in T_p\mathcal{M}$. If f^*g is positive definite, it is a Riemannian metric on $\mathcal{M}_1$, called the pullback metric defined by f.

Manfredo Perdigao Do Carmo and J Flaherty Francis. Riemannian Geometry. Springer.1992.
Loring W.. Tu. An introduction to manifolds. Springer. 2011

Preliminaries on the SPD Geometry

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

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

Yann Thanwerdas and Xavier Pennec. O (n)-invariant Riemannian metrics on SPD matrices. Linear Algebra and its Applications. 2023.
Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. SIAM Journal on Matrix Analysis and Applications. 2019.

Riemannian BN Revisitied

Table 2: Summary of some representative RBN methods.

Methods	Involved Statistics	Controllable Mean	Controllable Variance	Application Scenarios
SPDBN (Brooks et al., 2019b)	Mean	✓	N/A	SPD manifolds under AIM
SPDBN (Kobler et al., 2022b)	Mean+Variance	✓	✓	SPD manifolds under AIM
Chakraborty (2020, Algs. 1-2)	Mean+Variance	✗	✗	Riemannian homogeneous space
Chakraborty (2020, Algs. 3-4)	Mean+Variance	✓	✓	A certain Lie group structure and distance
RBN (Lou et al., 2020, Alg. 2)	Mean+Variance	✗	✗	Geodesically complete manifolds
Ours	Mean+Variance	✓	✓	General Lie groups

All of the previous RBN methods fail to control statistics in a general manner.

Daniel Brooks, et al. Riemannian batch normalization for SPD neural networks. Neurips. 2019.
Reinmar J Kobler, et al. Controlling the Fréchet variance improves batch normalization on the symmetric positive definite manifold. ICASSP. 2022.
Rudrasis Chakraborty. Extending normalizations on Riemannian manifolds. ArXiv. 2020.
Aaron Lou, et al. Differentiating through the Fréchet mean. ICML. 2020

LieBN: from Euclidean to Lie Groups

UNIVERSITÀ DI TRENTO UNIVERSITY OF LOUISVILLE

Euclidean BN:

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

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

Our LieBN:

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

Properties

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

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

Our centering and biasing can control mean, while scaling can control variance:

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 $\{P_{i\ldots N} \in \mathcal{M}\}$ 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 $\{P_{i\ldots N} \in \mathcal{M}\}$, denoting $\phi_s(P_i) = \mathrm{Exp}_E\left[s\,\mathrm{Log}_E(P_i)\right]$, we have the following properties for the sample statistics:*

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

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

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

Under metrics (LEM and LCM on SPD manifolds), LieBN can further control the latent Gaussian (App. C): $\mathcal{N}(M, \sigma^2) \to \mathcal{N}(E, \sigma^2) \to \mathcal{N}(E, s^2) \to \mathcal{N}(B, s^2)$,

Algorithm

UNIVERSITÀ DI TRENTO | UNIVERSITY OF LOUISVILLE

Algorithm 1: Lie Group Batch Normalization (LieBN) Algorithm

Input : A batch of activations $\{P_{1\dots N} \in \mathcal{M}\}$, 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 $\{\tilde{P}_{1\dots N}\}$

if *training* **then**
  Compute batch mean M_b and variance v_b^2 of $\{P_{1\dots N}\}$;
  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

Our LieBN is a natural generalization 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.*

LieBN on the SPD Manifold

We further consider three deformed invariant metrics on the SPD manifold.

Table 3: Key operators in calculating LieBN on SPD manifolds.

Metric		(θ, α, β)-AIM	(α, β)-LEM	θ-LCM
Pullback Map		P_θ	mlog	$\mathrm{P}_\theta \circ \psi_{\mathrm{LC}}$
Codomain		$\{\mathcal{S}^n_{++}, \odot^{\mathrm{AI}}, \frac{1}{\theta^2} g^{(\alpha,\beta)\text{-AI}}\}$	$\{\mathcal{S}^n, \langle\cdot,\cdot\rangle^{(\alpha,\beta)}\}$	$\{\mathcal{L}^n, \frac{1}{\theta^2}\langle\cdot,\cdot\rangle\}$
Riemannian and Lie group operators in the codomain	$L_Q(P)$	$KPK^\top$	$P+Q$	$P+Q$
	$L_{Q^{-1}_{\odot}}(P)$	$K^{-1}PK^{-\top}$	$P-Q$	$P-Q$
	$\mathrm{Exp}_E\left[s \,\mathrm{Log}_E(P)\right]$	P^s	sP	sP
	FM	Karcher Flow	Arithmetic average	Arithmetic average
	$\mathrm{WFM}(\{\gamma, 1-\gamma\}, \{P_1, P_2\})$	$P_2^{\frac{1}{2}} \left(P_2^{-\frac{1}{2}} P_1 P_2^{-\frac{1}{2}}\right)^\gamma P_2^{\frac{1}{2}}$	Arithmetic weighted average	Arithmetic weighted average

Properties:

Proposition 5.1 (Deformation). [↓] *(θ, α, β)-LEM is equal to (α, β)-LEM. θ-LCM interpolates between $\tilde{g}$-LEM ($\theta = 0$) and LCM ($\theta = 1$), with $\tilde{g}$-LEM defined as*

$$\langle V, W\rangle_P = \tilde{g}(\mathrm{mlog}_{*,P}(V), \mathrm{mlog}_{*,P}(W)), \forall P \in \mathcal{S}^n_{++}, \forall V, W \in T_P\mathcal{S}^n_{++}, \tag{18}$$

where $\tilde{g}(V_1, V_2) = \frac{1}{2}\langle V_1, V_2\rangle - \frac{1}{4}\langle \mathbb{D}(V_1), \mathbb{D}(V_2)\rangle$, $\mathbb{D}(V_i)$ is a diagonal matrix consisting of the diagonal elements of V_i, and $\mathrm{mlog}_{*,P}$ is the differential map at P.

Proposition 5.2 (Invariance). [↓] *(θ, α, β)-AIM is left-invariant w.r.t. $\odot^{\theta\text{-AI}}$, while θ-LCM is bi-invariant w.r.t. $\odot^{\theta\text{-LC}}$.*

LieBN on SPDNet and TSMNet

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 data:

(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**

Preliminaries Results on Rotation

Ingredients:

Table 8: The associated Riemannian operators on Rotation matrices.

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

Application to LieNet:

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

Zhiwu Huang, et al. Deep learning on Lie groups for skeleton-based action recognition. CVPR. 2017.

Q & A

Thanks you
Q & A

Code

Paper

Homepage