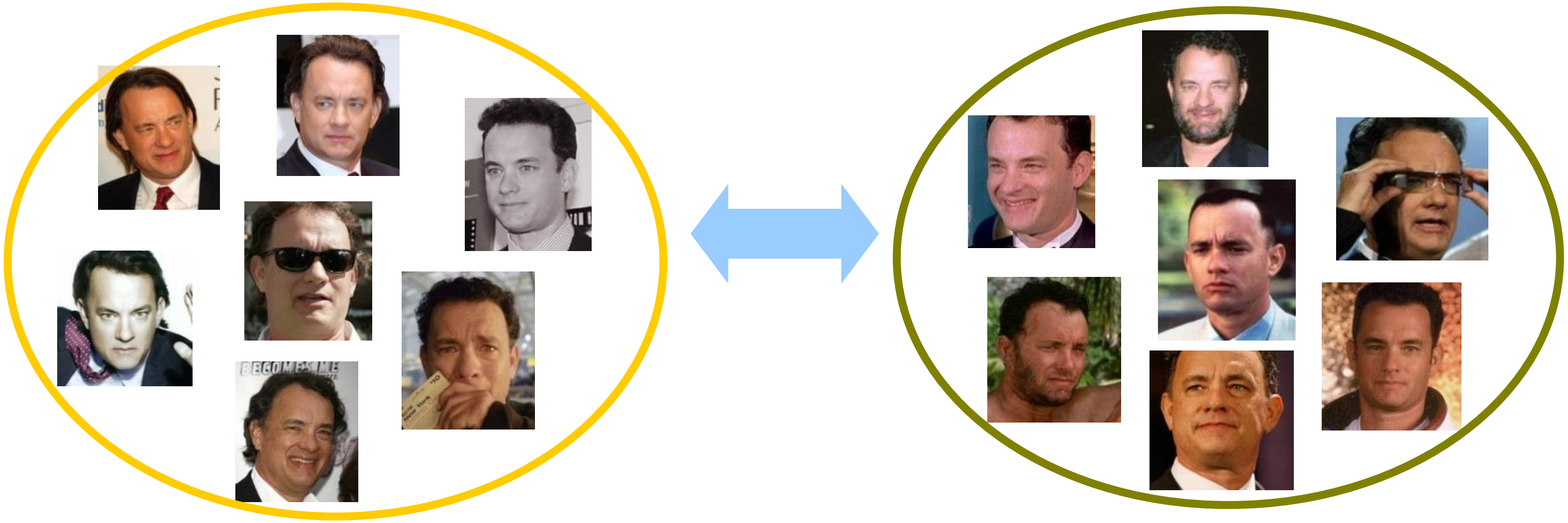
Log-Euclidean Metric Learning on Symmetric Positive Definite Manifold with Application to Image Set Classification

Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, Xilin Chen

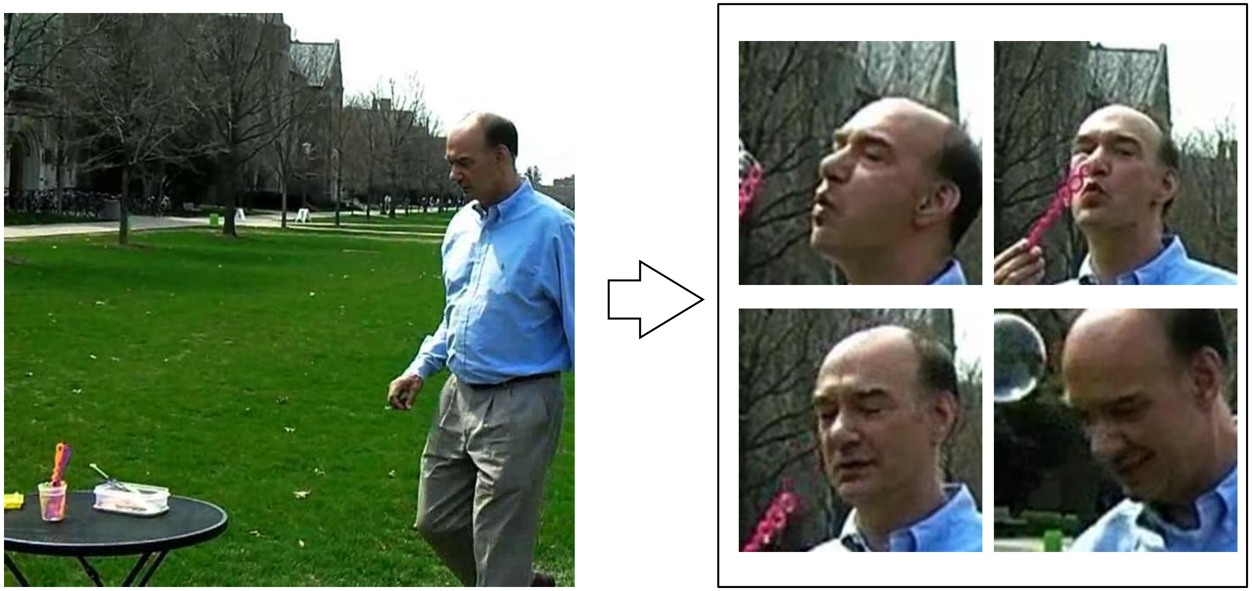
Institute of Computing Technology, Chinese Academy of Sciences

Presented by **Bo Xin**
July 9, 2015

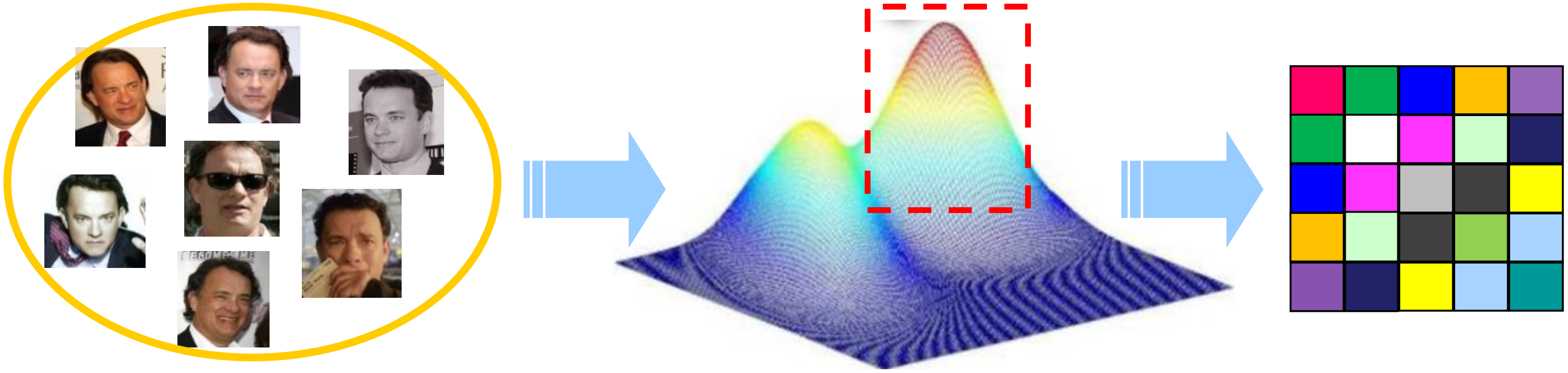
- Training/testing sample is a set of images involving a single subject
  + Rich information to describe subject
  – Complex appearance variations



- Example
  - Video-based face recognition
    - Identify a subject with his/her video sequence
      - Treating video as image set

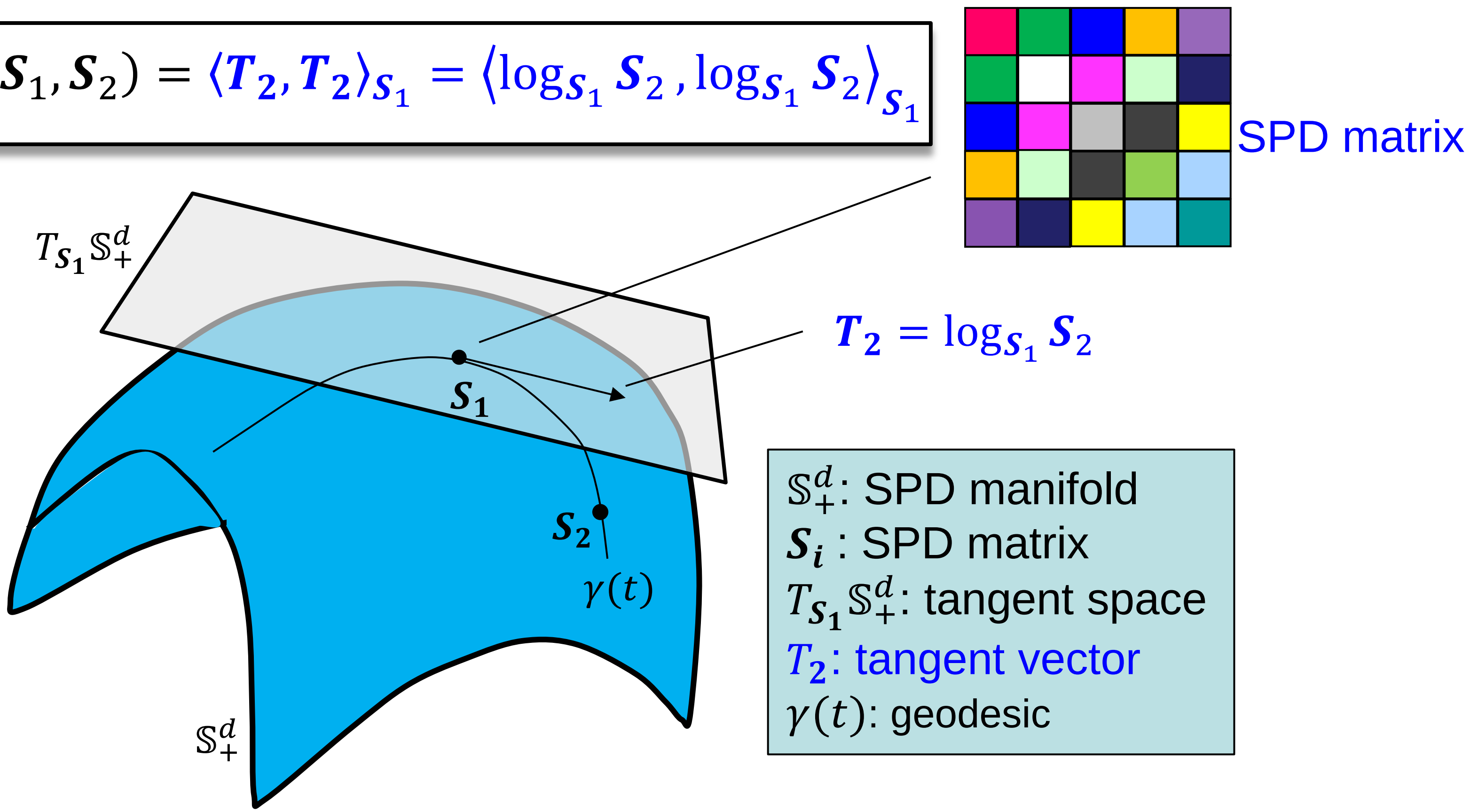


- Represent image set with Gaussian model
  - From Gaussian model to SPD matrix
    - Information geometry theory [Amari & Nagaoka,2000; Lovrić,2000]
      - $\mathcal{N}(x|\tilde{m},\tilde{C}) \sim \boldsymbol{S} = |\tilde{\boldsymbol{C}}|^{-\frac{1}{d+1}} \begin{bmatrix} \tilde{\boldsymbol{C}} + \tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}^T & \tilde{\boldsymbol{m}} \\ \tilde{\boldsymbol{m}}^T & 1 \end{bmatrix}$
      - $\tilde{\boldsymbol{C}}$: covariance matrix of size $d \times d$, $\tilde{\boldsymbol{m}}$: mean vector of size d



- Account for Riemannian geometry
  - Riemannian metric

$$\delta_g^2(\boldsymbol{S}_1, \boldsymbol{S}_2) = \langle \boldsymbol{T}_2, \boldsymbol{T}_2 \rangle_{\boldsymbol{S}_1} = \langle \log_{\boldsymbol{S}_1} \boldsymbol{S}_2, \log_{\boldsymbol{S}_1} \boldsymbol{S}_2 \rangle_{\boldsymbol{S}_1}$$

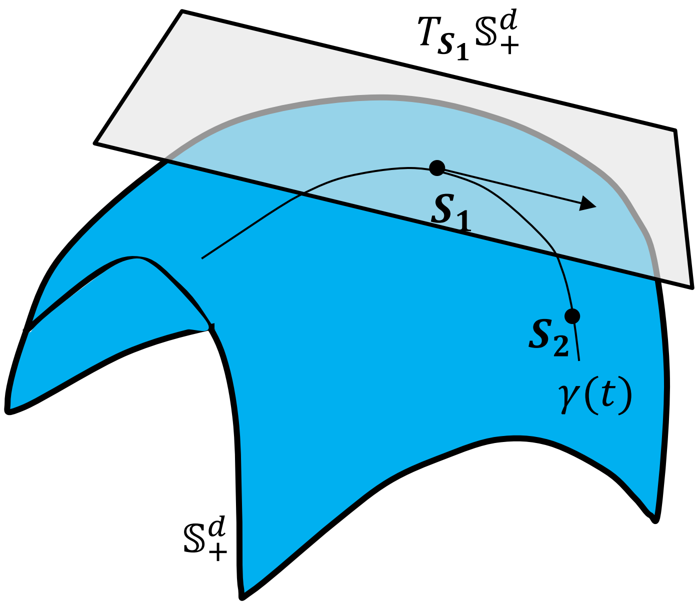


$\mathbb{S}_+^d$: SPD manifold
$\boldsymbol{S}_i$: SPD matrix
$T_{\boldsymbol{S}_1}\mathbb{S}_+^d$: tangent space
$\boldsymbol{T}_2$: tangent vector
$\gamma(t)$: geodesic

- Account for Riemannian geometry
  - Affine-Invariant metric (AIM)

$$d_a^2(\boldsymbol{S}_1,\boldsymbol{S}_2)=\left\langle \log_{\boldsymbol{S}_1}\boldsymbol{S}_2,\log_{\boldsymbol{S}_1}\boldsymbol{S}_2\right\rangle_{\boldsymbol{S}_1}=||\log\left(\boldsymbol{S}_1^{-1/2}\boldsymbol{S}_2\boldsymbol{S}_1^{-1/2}\right)||_{\mathcal{F}}^2$$

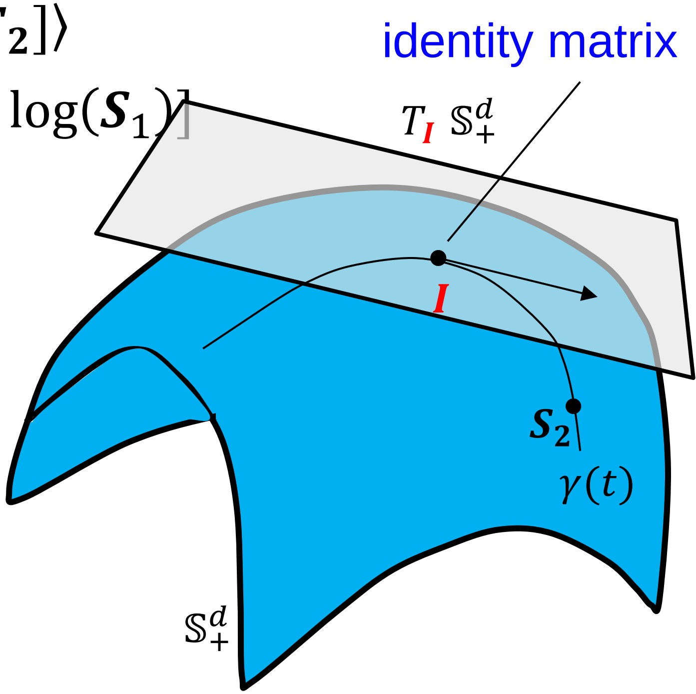
- $\langle \boldsymbol{T}_2,\boldsymbol{T}_2\rangle_{\boldsymbol{S}_1}=\left\langle \boldsymbol{S}_1^{-\frac{1}{2}}\boldsymbol{T}_2\boldsymbol{S}_1^{-\frac{1}{2}},\boldsymbol{S}_1^{-\frac{1}{2}}\boldsymbol{T}_2\boldsymbol{S}_1^{-\frac{1}{2}}\right\rangle$
- $\boldsymbol{T}_2=\log_{\boldsymbol{S}_1}\boldsymbol{S}_2=\boldsymbol{S}_1^{\frac{1}{2}}\log(\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}})\boldsymbol{S}_1^{\frac{1}{2}}$
- Computational cost is expensive
  - Main cost: $\log(\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}})$

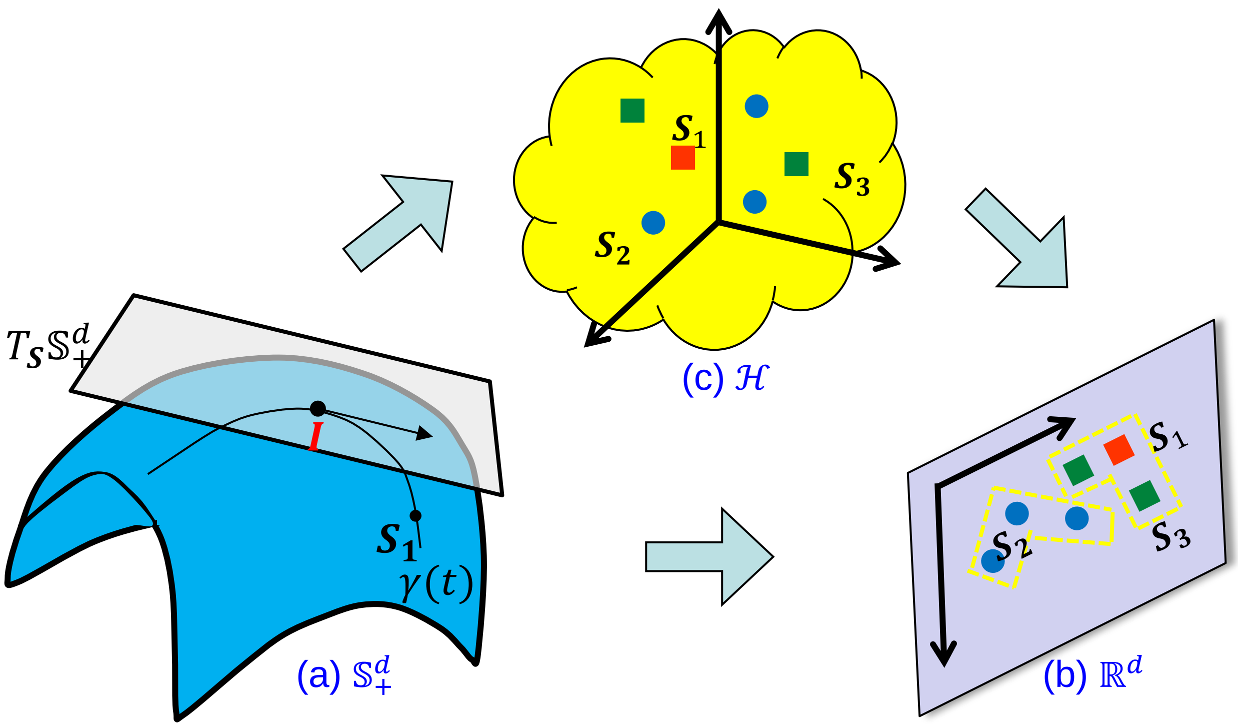


- Account for Riemannian geometry
  - Log-Euclidean metric (LEM)

$$d_l^2(\boldsymbol{S}_1, \boldsymbol{S}_2) = \left\langle \log_{\boldsymbol{S}_1} \boldsymbol{S}_2, \log_{\boldsymbol{S}_1} \boldsymbol{S}_2 \right\rangle_{\boldsymbol{S}_1} = ||\log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2)||_{\mathcal{F}}^2$$

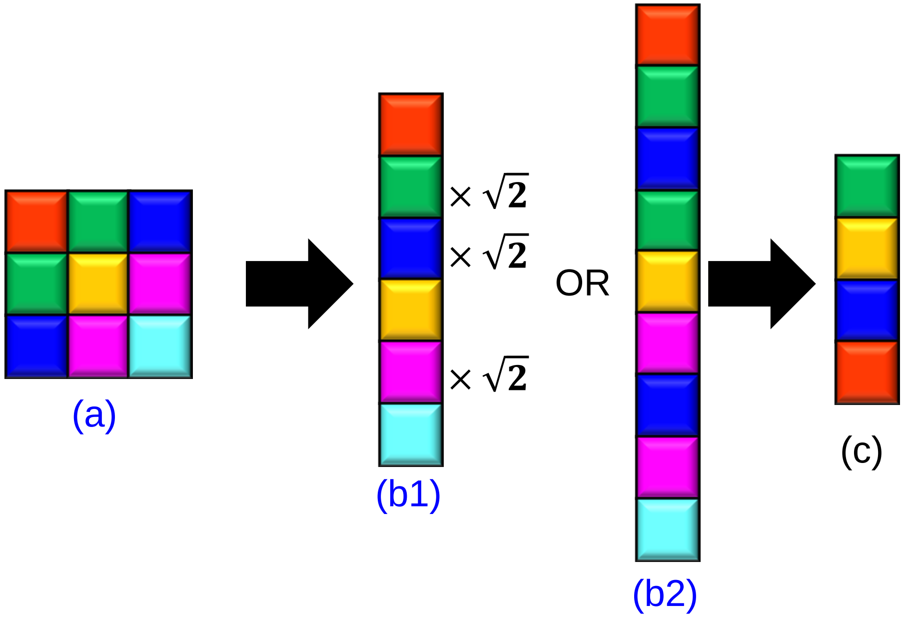
- $\langle \boldsymbol{T}_2, \boldsymbol{T}_2 \rangle_{\boldsymbol{S}_1} = \langle \mathrm{Dlog}(\boldsymbol{S}_1)[\boldsymbol{T}_2], \mathrm{Dlog}(\boldsymbol{S}_1)[\boldsymbol{T}_2] \rangle$
- $\boldsymbol{T}_2 = \log_{\boldsymbol{S}_1} \boldsymbol{S}_2 = \mathrm{D}^{-1}\log(\boldsymbol{S}_1)[\log(\boldsymbol{S}_2) - \log(\boldsymbol{S}_1)]$
- Drastic reduction in computation time
  - Need Euclidean computation in the domain of matrix logarithms



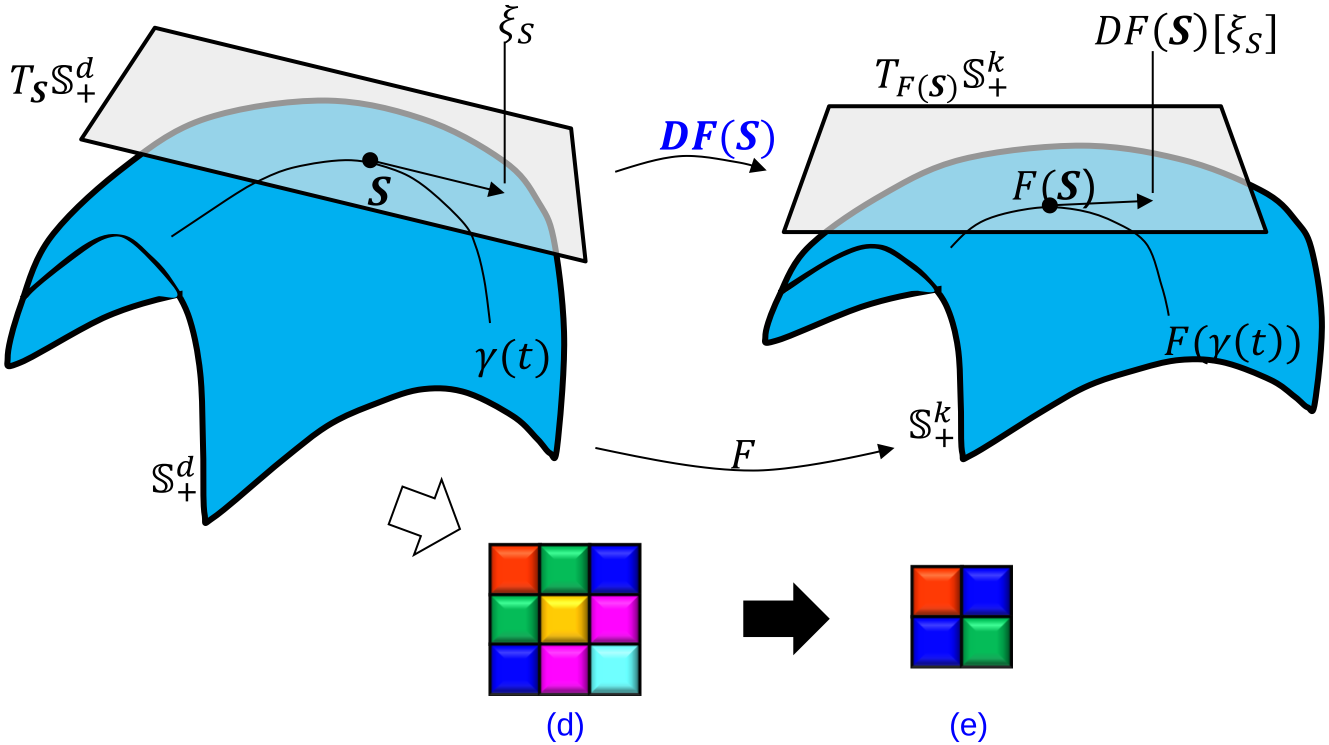


- Tangent space approximation ((a)-(b))
  - e.g., Tosato *et al.* 2010, Carreira *et al.* 2012, Vemulapalli *et al.* 2015
- Hilbert space embedding ((a)-(c)-(b))
  - e.g., Wang *et al.* 2012, Jayasumana *et al.* 2013, Minh *et al.* 2014

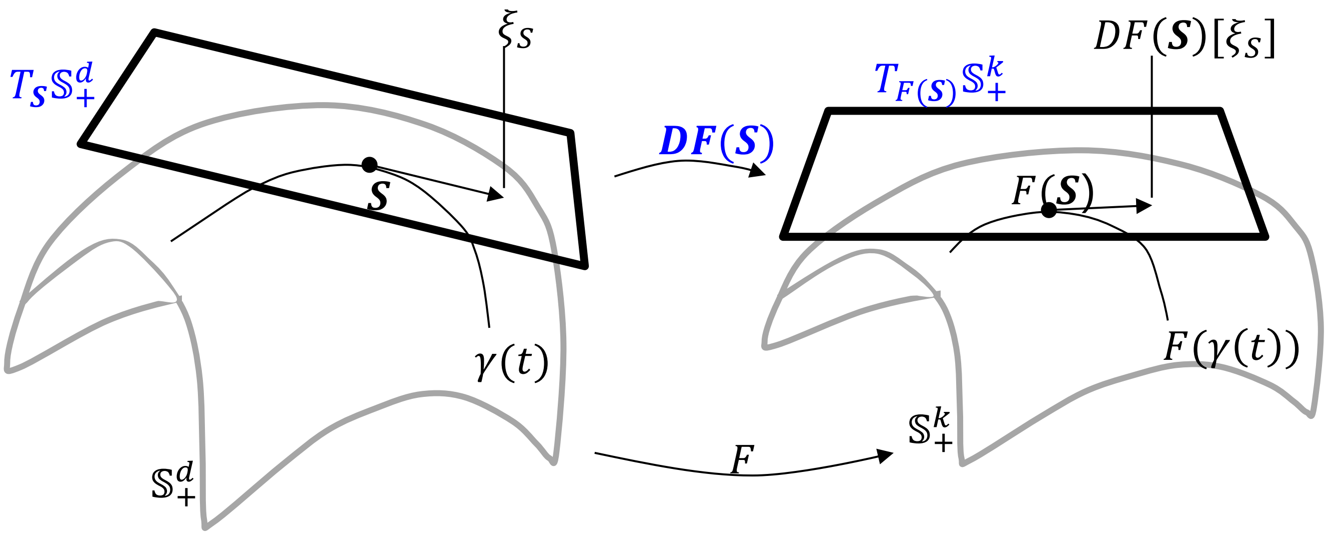
- Convert SPD matrix logarithm into vector-form in tangent space at identity matrix ((a)-(b1)/(b2))
  - Ignore the symmetric property of SPD matrix logarithm
  - Work inefficiently on the SPD vector-form often of high dimensionality



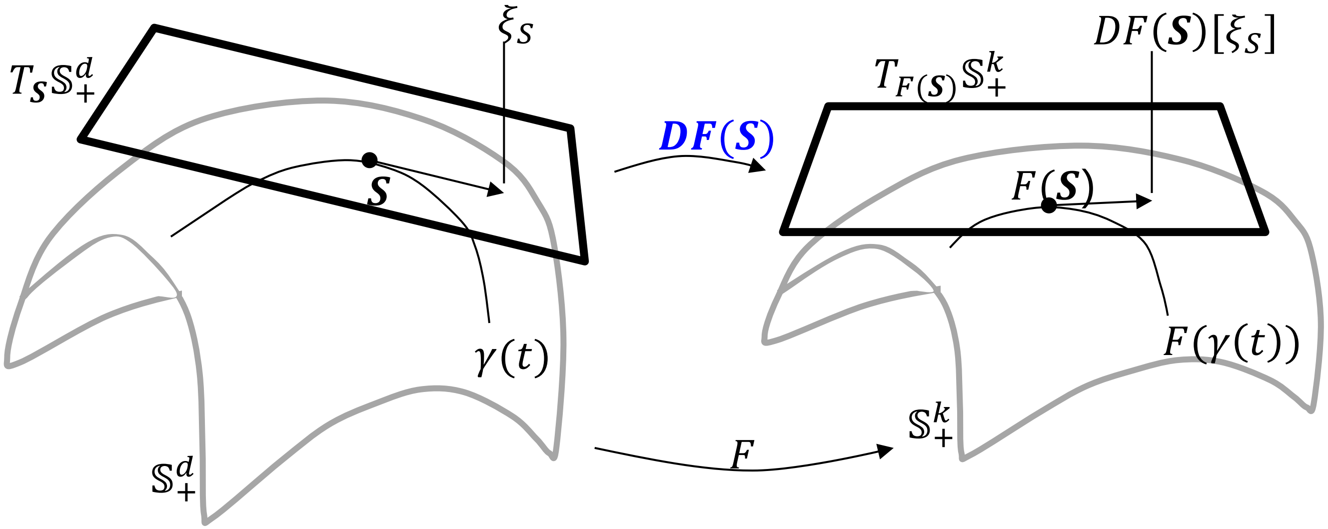
Our Approach



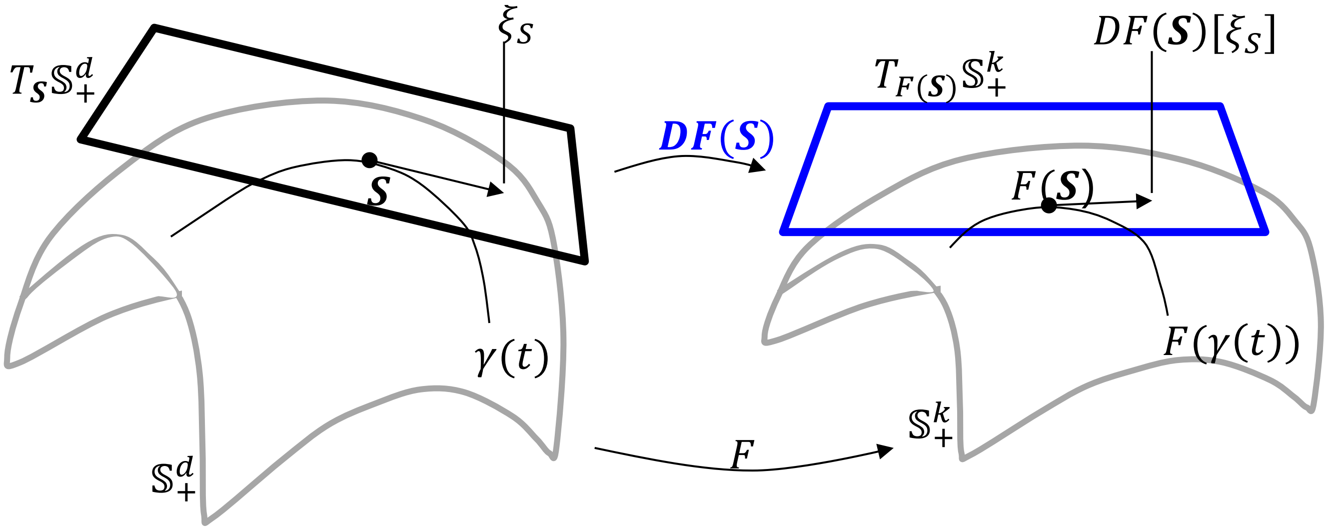
- Learn tangent-to-tangent map $\boldsymbol{DF(S)}$
  - Work on the matrix-form of SPD matrix logarithm ((d)-(e))
    - Keep the symmetric property of SPD matrix logarithm
    - Work efficiently on lower-dimensional matrix-form



- Learn tangent-to-tangent map
  - From original tangent space $T_{\boldsymbol{S}}\mathbb{S}_+^d$ to a more discriminant tangent space $T_{F(\boldsymbol{S})}\mathbb{S}_+^k$
    - $DF(\boldsymbol{S}): T_{\boldsymbol{S}}\mathbb{S}_+^d \rightarrow T_{F(\boldsymbol{S})}\mathbb{S}_+^k$
    - *If $DF(\boldsymbol{S})$ is an injection, the manifold-to-manifold map $F: \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$ is an immersion*



- Learn tangent-to-tangent map
  - Specific form of tangent map
    - $DF(\boldsymbol{S})$: $f(\log(\boldsymbol{S})) = \boldsymbol{W}^T\log(\boldsymbol{S})\boldsymbol{W}$
      - $\log(\boldsymbol{S}) \in \mathbb{R}^{d\times d}, \boldsymbol{W} \in \mathbb{R}^{d\times k}, f(\log(\boldsymbol{S})) \in \mathbb{R}^{k\times k}$
      - if $\boldsymbol{W}$: column full rank, $f(\log(\boldsymbol{S}))$ *yields a valid symmetric matrix*



- Log-Euclidean metric on new SPD manifold
  - $d_l^2\big(f(\boldsymbol{T}_i), f(\boldsymbol{T}_j)\big) = ||\boldsymbol{W}^T\boldsymbol{T}_i\boldsymbol{W} - \boldsymbol{W}^T\boldsymbol{T}_j\boldsymbol{W}||_F^2$
    $= tr(\boldsymbol{Q}(\boldsymbol{T}_i - \boldsymbol{T}_j)(\boldsymbol{T}_i - \boldsymbol{T}_j))$
  - $\boldsymbol{T}_i = \log(\boldsymbol{S}_i), \boldsymbol{T}_j = \log(\boldsymbol{S}_j)$
  - $\boldsymbol{Q} = (\boldsymbol{W}\boldsymbol{W}^T)^2$: PSD matrix

*$\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{T}_i - \boldsymbol{T}_j)$ is required to be symmetric

- Objective function (matrix-form of the ITML method [Davis *et al.*, 2007])
  - $\arg\min_{\boldsymbol{Q},\xi} D_{\ell d}(\boldsymbol{Q}, \boldsymbol{Q}_{\mathbf{0}}) + \eta D_{\ell d}(\boldsymbol{\xi}, \boldsymbol{\xi}_{\mathbf{0}})$

    $\mathrm{s.t.}, \mathrm{tr}\left(\boldsymbol{Q}\boldsymbol{A}_{ij}^{T}\mathbf{A}_{ij}\right) \le \boldsymbol{\xi}_{c(i,j)}, c(i,j) \in \boldsymbol{S}$

    $\mathrm{tr}\left(\boldsymbol{Q}\boldsymbol{A}_{ij}^{T}\mathbf{A}_{ij}\right) \ge \boldsymbol{\xi}_{c(i,j)}, c(i,j) \in \boldsymbol{D}$
  - $D_{\ell d}$: LogDet divergence, $\mathbf{A}_{ij} = \log(\boldsymbol{C}_i) - \log(\boldsymbol{C}_j)$,
  - $\boldsymbol{S}/\boldsymbol{D}$: constraint set involving sample pairs with the same /different label(s)

- Optimization algorithm
  - Cyclic Bregman projection algorithm [Bregman,1967; Censor & Zenior, 1997]
    - Choose one constraint per iteration
    - Perform a projection so that the current solution satisfies the chosen constraint

Evaluated Methods

Method	Literature source	abbr.
SPD basic metric	Pennec *et al*., IJCV’2006	AIM
	Sra *et al*., NIPS’2012	Stein
	Arsigny *et al*., SIAM MAA’2007	LEM
SPD metric learning	Harandi *et al*., ECCV’2014	SPDML-AIM/Stein
	Harandi *et al*., ECCV’2012	RSR-Stein
	Wang *et al*., CVPR’2012	CDL-LEM
	Vemulapalli *et al*., arXiv’2015	ITML-LEM

- SPDML-AIM/Stein: SPD manifold learning (SPDML) with Affine-Invariant metric (AIM) or Stein divergence
- RSR-Stein: Riemannian Sparse Representation (RSR) with Stein divergence
- CDL-LEM: Covariance Discriminative Learning (CDL) with Log-Euclidean metric (LEM)
- ITML-LEM: **Information-Theoretic Metric Learning (ITML)** on vector-form of SPD matrix logarithm with **Log-Euclidean Metric (LEM)**

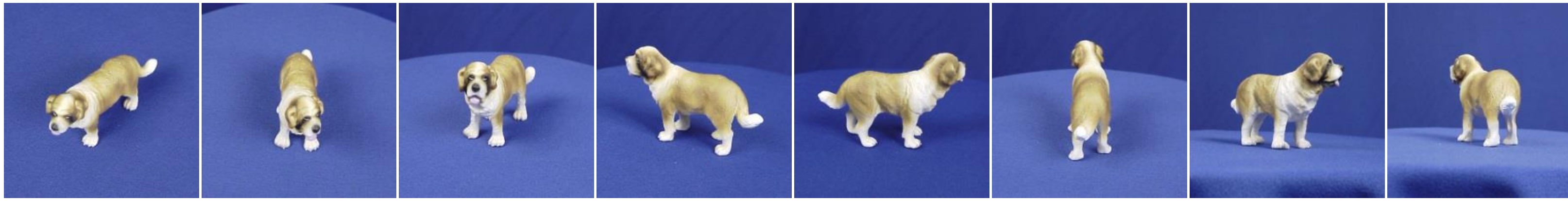
Set-based Object Categorization

- ETH-80 dataset (Leibe & Schiele, 2003)
  - 80 image sets of 8 object categories
    - Each category has 10 image sets
  - 20×20 resized intensity images
  - 401×401 SPD feature
  - Random selection for 10 tests
    - 50% for gallery, 50% for probe

$$\boldsymbol{S} = |\widetilde{\boldsymbol{C}}|^{-\frac{1}{d+1}} \begin{bmatrix} \widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{m}}\widetilde{\boldsymbol{m}}^T & \widetilde{\boldsymbol{m}} \\ \widetilde{\boldsymbol{m}}^T & 1 \end{bmatrix}$$

$\widetilde{\boldsymbol{C}}$: covariance matrix, $\widetilde{\boldsymbol{m}}$: mean

Set-based Object Categorization: Results



Method	Accuracy
AIM	87.50 ± 5.77
Stein	88.00 ± 5.11
LEM	89.25 ± 4.72
SPDML-AIM	90.75 ± 3.34
SPDML-Stein	90.50 ± 3.87
RSR-Stein	93.25 ± 3.34
CDL-LEM	93.75 ± 3.43
ITML-LEM	93.75 ± 3.43
LEML	$\mathbf{94.75 \pm 2.49}$
LEML-CDL	$\mathbf{96.00 \pm 2.11}$

Video-based Face Identification

- YouTube Celebrities dataset (Kim *et al.*, 2008)
  - 1,910 video sequences of 47 subjects from YouTube
    - Highly compressed, low resolution
  - 20×20 resized intensity images
  - 401×401 SPD feature
  - Random selection for 10 tests
    - 3 of 9 for gallery, 6 of 9 for probe

$$\boldsymbol{S} = |\widetilde{\boldsymbol{C}}|^{-\frac{1}{d+1}} \begin{bmatrix} \widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{m}}\widetilde{\boldsymbol{m}}^T & \widetilde{\boldsymbol{m}} \\ \widetilde{\boldsymbol{m}}^T & 1 \end{bmatrix}$$

$\widetilde{\boldsymbol{C}}$: covariance matrix, $\widetilde{\boldsymbol{m}}$: mean

Video-based Face Identification: Results

Method	Accuracy
AIM	62.85±3.46
Stein	61.46±3.53
LEM	63.91±3.25
SPDML-AIM	64.66±2.92
SPDML-Stein	61.57±3.43
RSR-Stein	72.77±2.69
CDL-LEM	72.67±2.47
ITML-LEM	66.51±3.67
LEML	**70.53±2.95**
LEML-CDL	**73.31±2.49**

Video-based Face Verification

- YouTube Faces DB (Wolf *et al.*, 2011)
  - 3,425 video sequences of 1,595 subjects from YouTube
    - Highly compressed, low resolution
  - 24×40 resized intensity images
  - 961×961 SPD feature
  - Random selection for 10 folds
    - 9 folds for training, 1 fold for testing

$$\boldsymbol{S} = |\widetilde{\boldsymbol{C}}|^{-\frac{1}{d+1}} \begin{bmatrix} \widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{m}}\widetilde{\boldsymbol{m}}^T & \widetilde{\boldsymbol{m}} \\ \widetilde{\boldsymbol{m}}^T & 1 \end{bmatrix}$$

$\widetilde{\boldsymbol{C}}$: covariance matrix, $\widetilde{\boldsymbol{m}}$: mean

Video-based Face Verification: Results

Method	Accuracy
AIM	59.28±2.25
Stein	58.70±1.97
LEM	61.48±2.27
SPDML-AIM	62.16±2.16
SPDML-Stein	62.56±2.49
RSR-Stein	N/A
CDL-LEM	66.76±1.89
ITML-LEM	60.02±1.84
LEML	**65.12±1.54**
LEML-CDL	**72.34±2.07**

Running Time

- Training and testing (classification of one video) time on YouTube Celebrities dataset

Method	Train	Test
SPDML-AIM	15072.56	9.35
SPDML-Stein	108.50	0.04
ITML-LEM	92007.13	0.02
LEML	**56.30**	**0.02**

Conclusions

- Our approach seeks to map the SPD matrix logarithms from the original tangent space to a more discriminant tangent space
- This keeps the symmetric property of SPD matrix logarithms, and works effectively on matrix-form

- Future work:
  – Study if the proposed SPD metric learning could be extended to end-to-end SPD feature learning
    » leverage the existing deep learning technique

Thank you!