# Transport transforms for signal analysis and machine learning

Gustavo K. Rohde

Email: gustavo.rohde@gmail.com

Imaging and data science laboratory

Department of Biomedical Engineering

Department of Electrical and Computer Engineering

University of Virginia

**imagedatascience.com/transport**
**github.com/rohdelab/PyTransKit**

# "… we get by with a little help from our friends …"

(current)

Yan Zhuang
Xuwang Yin
M. Shifat E Rabbi
Hasnat Rubayiat
Shying Li

(past)

Mohammad Hassan
Cailey Fitzgerald
Liam Cattell
Wei Wang
Saurav Basu
Soheil Kolouri
Serim Park
Shinjini Kundu
Chi Liu
Burak Tosun
Shreyas Hirway
Cheng Chen
Aabid Shariff
Hu Huang
Jia Guo
Yanfeng Li
Shuchang Xu

(collaborators)

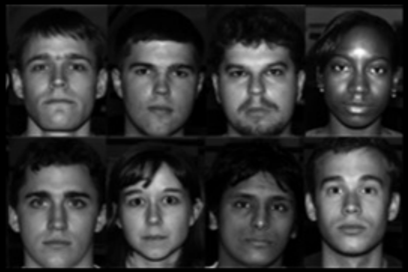
Akram Aldroubi, Vanderbilt
Jonathan Nichols, NRL
John Ozolek, WVU
Dejan Slepcev, CMU
Mathew Thorpe, U. Manchester
Sumati Thareja, Vanderbilt
Ivan Medri, Vanderbilt
Rocio Diaz Martin, Vanderbilt


## Funding

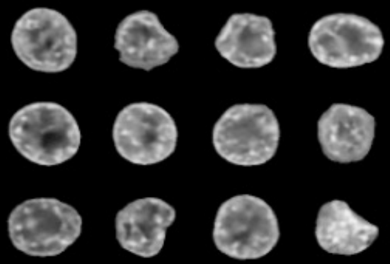
NIH R01 GM130825
NSF ABI 1759802

# Can we build smart systems cheaply with machine learning techniques?

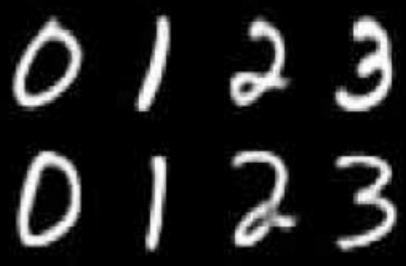
faces



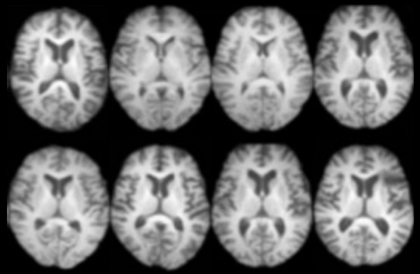
cells



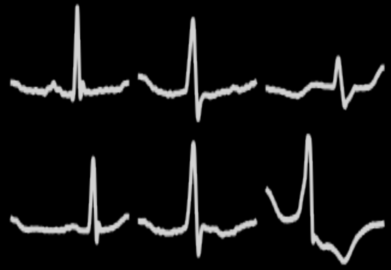
digits



brains



ECG



Example applications:

- Face recognition
- Medical diagnosis
  - (pathology, radiology)
- Optical character recognition
- EKG diagnosis
- Self driving cars
- Speech recognition
- …

What mathematical model shall we use?

How do we partition these data classes?

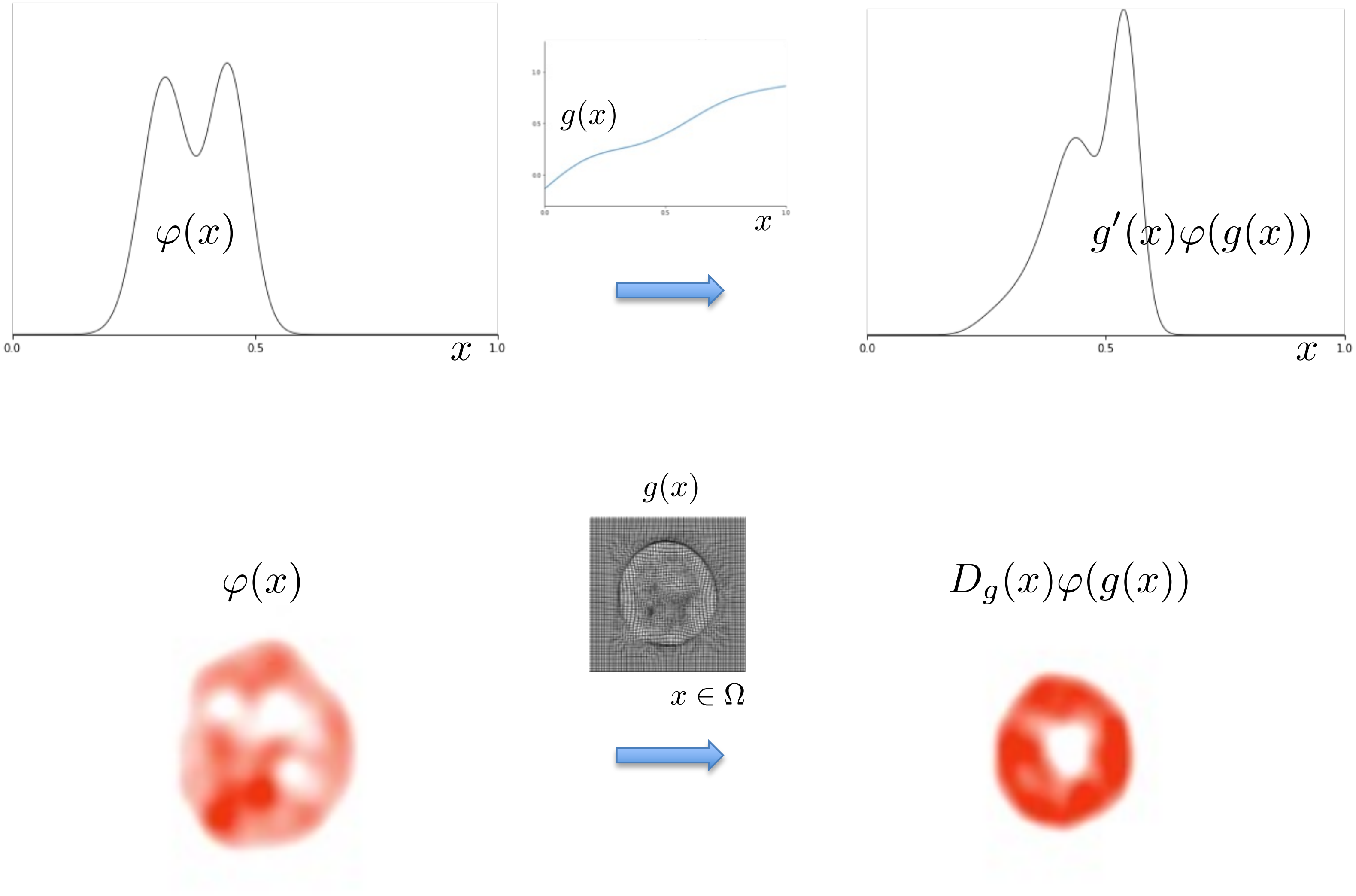
# Outline

- Modeling signal and image data
  - What are we talking about?
  - Transport model & problem statement
- Transport embeddings (representations, transforms)
  - 1D transform (CDT), ND (R-CDT, LOT),
  - Properties
- Applications
  - Signal/image classification
  - Transport-based morphometry

Matthew Thorpe
Caroline Mossmueller
Q. Merigot
A. Cloninger
D. Slepcev
D. Nguyen
B. Schmitzer
...

# Modeling signal and image data

# Model for signal/image deformations (transportations)




Translation:

$$g(x) = x - \tau$$

Scaling:

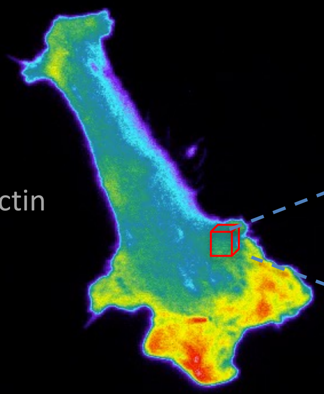
$$g(x) = ax$$

Nonrigid deformations:

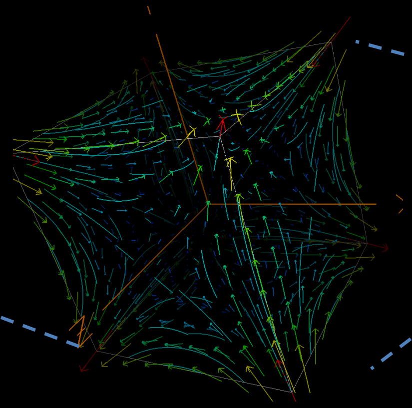
$$g(x) = \sum_k c_k \phi_k(x)$$

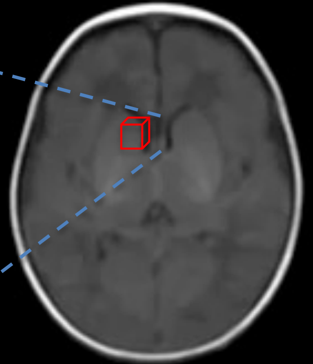
# What processes change signal/pixel intensity over time?

actin



Johnson et al, JCB, 2015





0 – 48 months

Nist lab, MNI

$\mathbf{v}$ flow/velocity

intensities not absolute

continuity equation

$$\underbrace{\frac{dI(x,y,z,t)}{dt}}_{\text{change in intensity at particular location}} = -\underbrace{\nabla\cdot\left(I(x,y,z,t)\mathbf{v}(x,y,z,t)\right)}_{\text{net flux tissue velocity}} + \underbrace{\cancel{(I, \ldots y, \ldots t)}}_{\text{source/sink term}}$$

# Transport with least action (optimal transport)

$I_0$ $\quad\underbrace{\qquad\qquad\qquad\qquad}_{I(x,t)}\quad$ $I_1$

$$\left.\begin{aligned} I(x,0) &= I_0(x) \\ I(x,1) &= I_1(x) \end{aligned}\right\}$$ sample images in a dataset

$$\frac{dI(x,t)}{dt} = -\nabla \cdot (I(x,y)v(x,t))$$

implies

$$I(x,t) = D_f(x) I_0(f(x,t))$$

$$f(x,t) = \int_0^t v(x,t)dt$$

**Theorem:** [Benamou, Brenier, Numer. Math, 2000]

$$W^2(I_0, I_1) = \inf_v \int_0^1 \int_\Omega I(x,t)|v(x,t)|^2 dx dt$$

$$f(x,t) = (1-t)x + t\nabla\phi$$

gradient of convex function

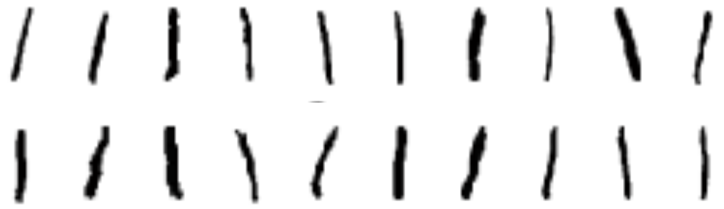
# Model for signal/image classes

An image class is modeled as

a set of transport maps $g_j \in \mathcal{G}$

applied to a template $\varphi^{(k)}$

$\mathbb{S}^1$ : class for digit 1

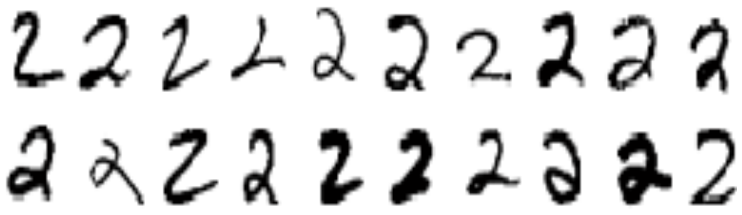


$$\mathbb{S}^{(1)} = \left\{ s_j^{(1)} | s_j^{(1)} = |\det J g_j| \varphi^{(1)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

template for class 1 $\varphi^{(1)} =$ 

“confound”

$\mathbb{S}^2$ : class for digit 2

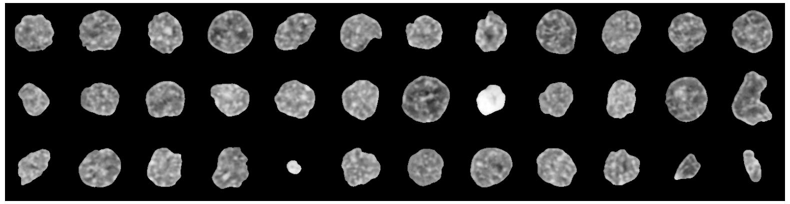


$$\mathbb{S}^{(2)} = \left\{ s_j^{(2)} | s_j^{(2)} = |\det J g_j| \varphi^{(2)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

template for class 2 $\varphi^{(2)} =$ 

# A typical learning problem

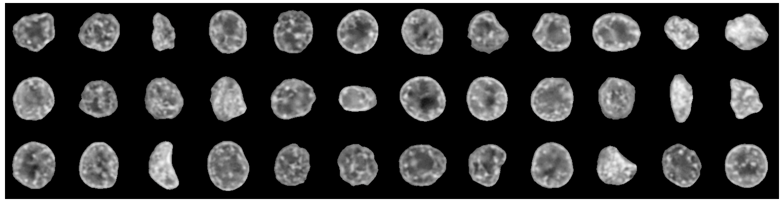
$\mathbb{S}^1$ : class 1 (benign cell nuclei)



$$\mathbb{S}^{(1)} = \left\{ s_j^{(1)} | s_j^{(1)} = |\det J g_j| \varphi^{(1)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

$\varphi^{(1)} =$ unknown

$\mathbb{S}^2$ : class 2 (malignant)



$$\mathbb{S}^{(2)} = \left\{ s_j^{(2)} | s_j^{(2)} = |\det J g_j| \varphi^{(2)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

$\varphi^{(2)} =$ unknown

$g_j \in \mathcal{G}$ unknown

# Model for 1D signal classes

$\mathbb{S}^{(1)}$: class for normal ECG heartbeats

$$\mathbb{S}^{(1)} = \{s_j^{(1)} | s_j^{(1)} = g'_j \varphi^{(1)} \circ g_j, \forall g_j \in \mathcal{G}\}$$

template for class 1: $\varphi^{(1)}$

$\mathbb{S}^{(2)}$: class for premature ventricular contraction beats

$$\mathbb{S}^{(2)} = \{s_j^{(2)} | s_j^{(2)} = g'_j \varphi^{(2)} \circ g_j, \forall g_j \in \mathcal{G}\}$$

template for class 2: $\varphi^{(2)}$

# Solution: Transport Transforms

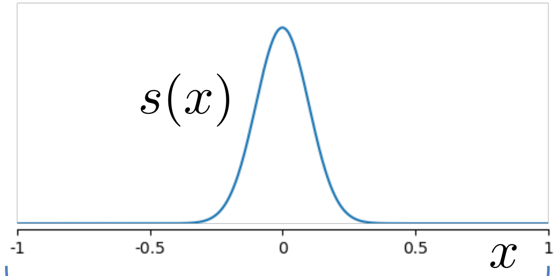
| signal space | | transform space | | signal space |
|---|---|---|---|---|
| problem hard to solve | ⇒ | problem easy to solve | ⇒ | answer in signal space |

estimation,
detection,
inverse problems,
classification

[Wang, Slepcev, Basu, Ozolek, Rohde, IJCV 2013]
[Kolouri, Tosun, Ozolek, Rohde, Pattern Recognition 2016]
[Kolouri, Park, Rohde, IEEE TIP, 2016]
[Park, Kolouri, Kundu, Rohde, ACHA 18]
[Rubaiyat et al, IEEE TSP, 20]
[Aldroubi, Li, Rohde, SaSiDa, 21]
[Shifat-E-Rabbi et al, Rohde, JMIV 21]
[Aldroubi et al, AIMS Foundations of Data Science, 22]

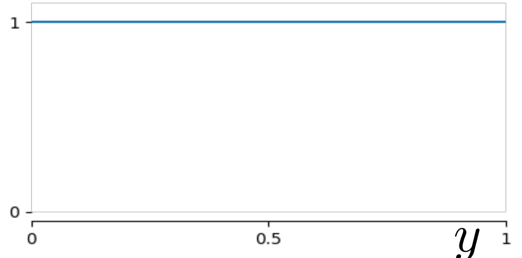
# Transport transforms

# CDT: invertible (nonlinear) signal transform

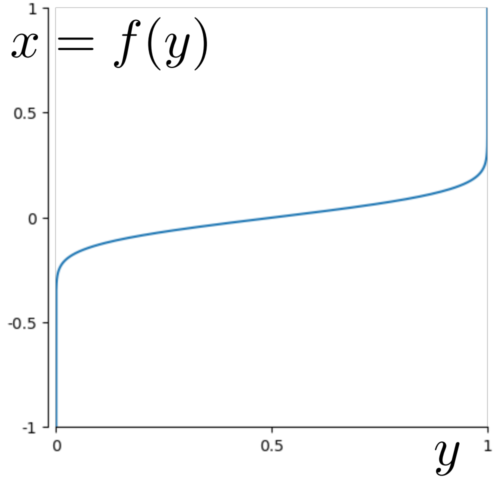



$x \in \Omega$

$s_0(y)$




$y \in \Omega_0$



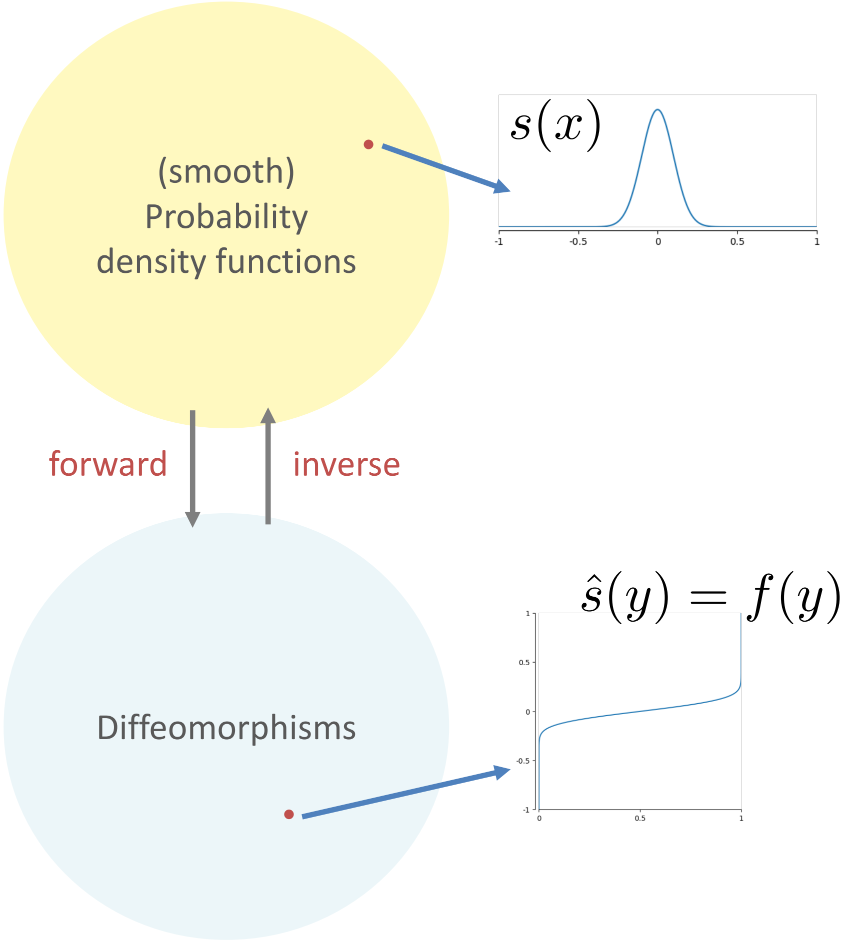

$$f : \Omega_0 \to \Omega$$

$$f'(y)s(f(y)) = s_0(y)$$

$$S(f(y)) = S_0(y)$$

$$y \in \Omega_0$$

$$\hat{s}(y) = f(y)$$



# 1-D Transport transform

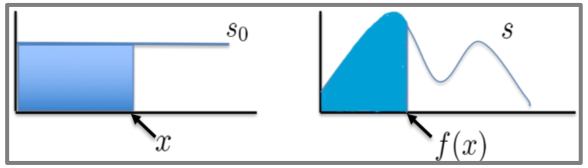
cumulative distribution transform (CDT)

[Park, Kolouri, Kundu, Rohde, ACHA 2018]

Let $s(x), s_0(x) > 0$ and $\int s(x)dx = \int s_0(x)dx = 1$ (PDFs)

Consider a map $f(x)$ , such that

$$\underbrace{\int_{-\infty}^{x} s_0(u)du}_{} = \underbrace{\int_{-\infty}^{f(x)} s(u)du}_{}$$

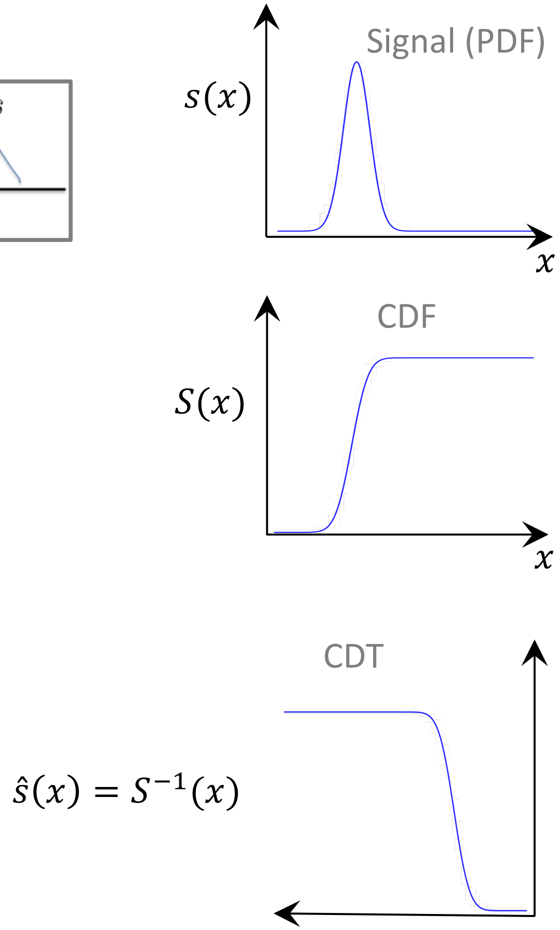



antiderivative

$$S_0(x) = S(f(x)) \implies f(x) = S^{-1}(S_0(x))$$

transform of $s$

Eg.: Let $s_0 = \chi_{[0,1]}$. Then, $S_0(x) = x$ and $f(x) = S^{-1}(x)$.




transform equations

$$\hat{s}(x) := S^{-1}(x) \quad \text{forward}$$

$$s(x) = \left(\hat{s}^{-1}(x)\right)' \quad \text{inverse}$$

# Extension to 2d, 3d: Radon Cumulative distribution transform (R-CDT)

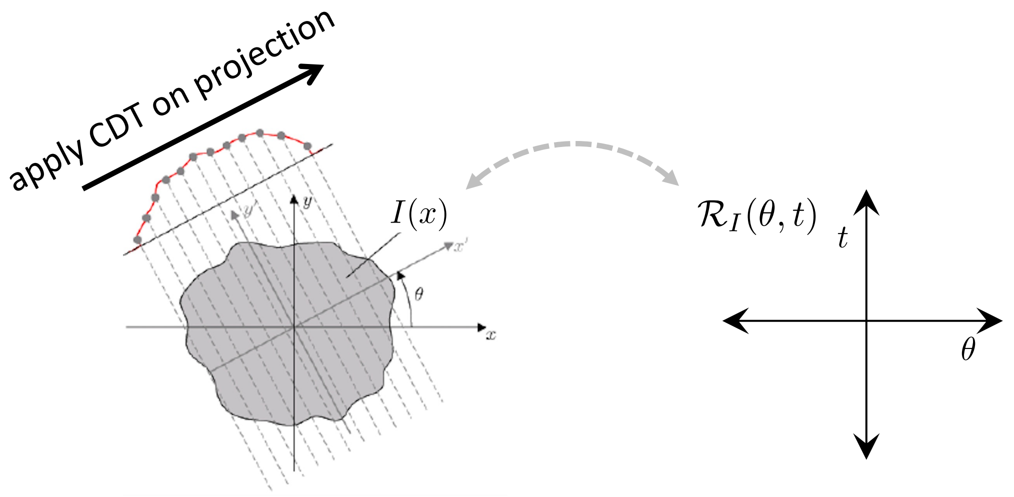
## Radon Transform

Forward

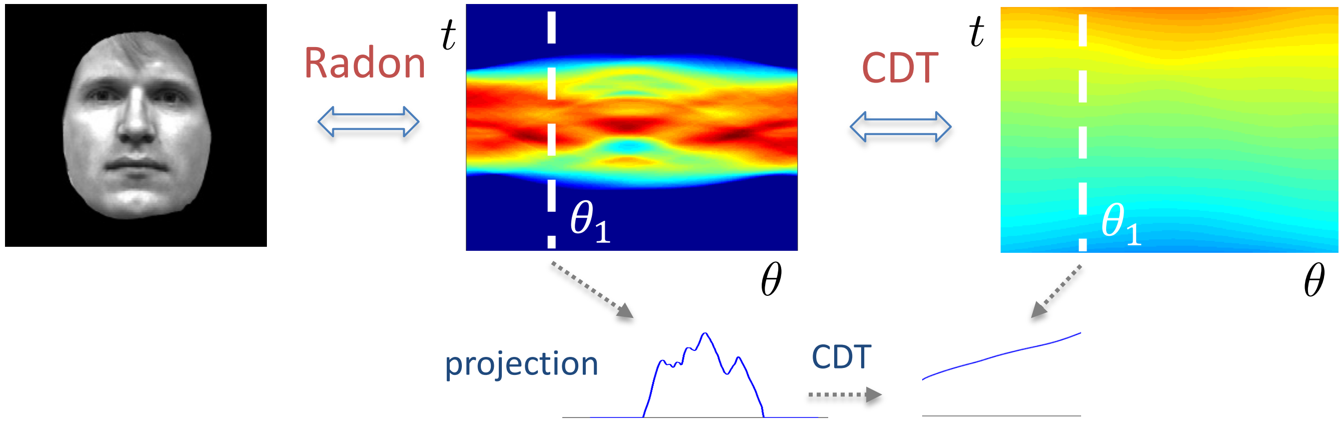
$$\mathcal{R}_I(\theta, t) = \int I(x)\delta(t - \omega \cdot x)dx$$
$$\omega = [\sin\theta, \cos\theta]^T$$

Inverse

$$\mathcal{R}_I^{-1}(x) = \int_{\mathbb{S}^{d-1}} (\mathcal{R}I(\cdot, \theta) * \eta(\cdot)) \circ (x \cdot \theta) d\theta$$




## Radon Cumulative Distribution Transform




[Kolouri, Park, Rohde, IEEE TIP 2016]

# CDT Properties

## Translation

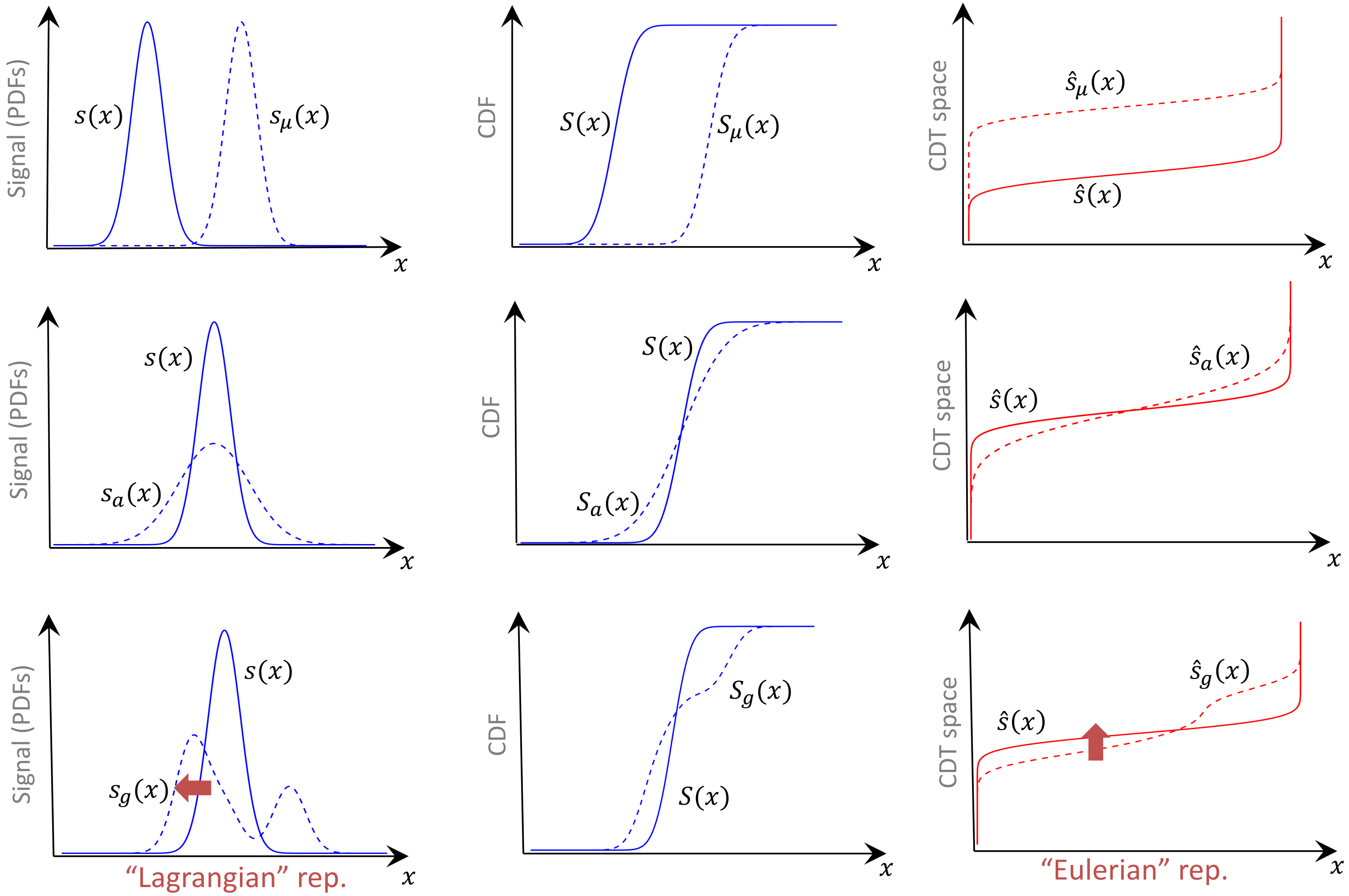
$$s_\mu(x) = s(x-\mu)$$
$$\hat{s}_\mu(x) = \hat{s}(x) + \mu$$

## Scaling

$$s_a(x) = as(ax)$$
$$\hat{s}_a(x) = \frac{\hat{s}(x)}{a}$$

## Composition

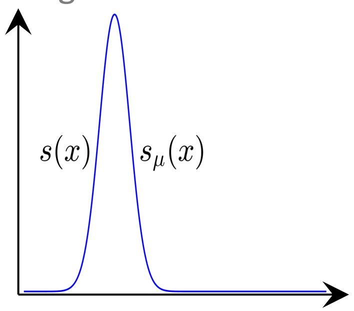
$$s_g(x) = g'(x)s(g(x))$$
$$\hat{s}_g(x) = g^{-1}(\hat{s}(x))$$



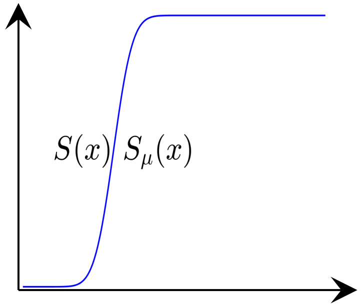
# Data Geometry

Consider the data set generated by a signal and all of its translates

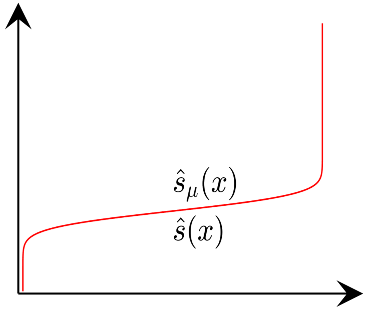
Signal



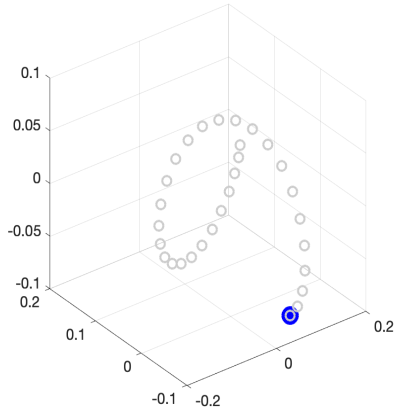

CDF



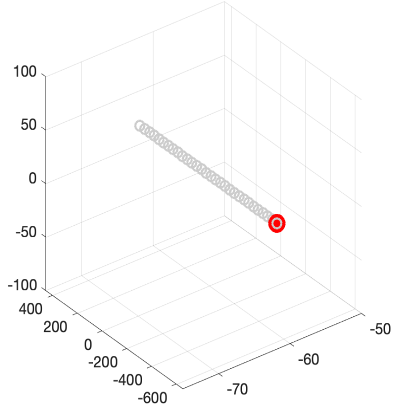

CDT




PCA (3 components) of dataset




nonlinear/nonconvex




linear/convex

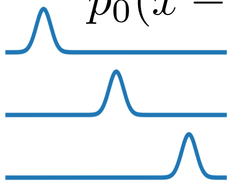
# Signal classes: generative model

## signal space

Class $\mathbb{P}$

template 1 $p_0$

$p_0(x-\mu)$

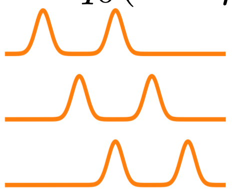


$$\mathbb{P} = \{p_g | p_g(x) = g'(x) p_0(g(x))\}$$

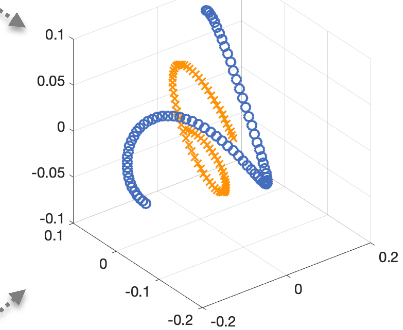
Class $\mathbb{Q}$

template 2 $q_0$

$q_0(x-\mu)$



$$\mathbb{Q} = \{q_g \, | q_g(x) = g'(x) q_0(g(x))\}$$



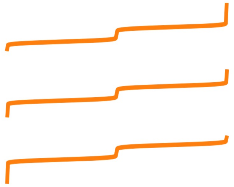

low dimensional projection

## transform space

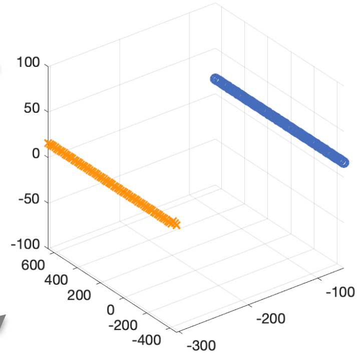
$\hat{\mathbb{P}}$

$\hat{p_g} = \hat{p} + \mu$

$\hat{\mathbb{Q}}$



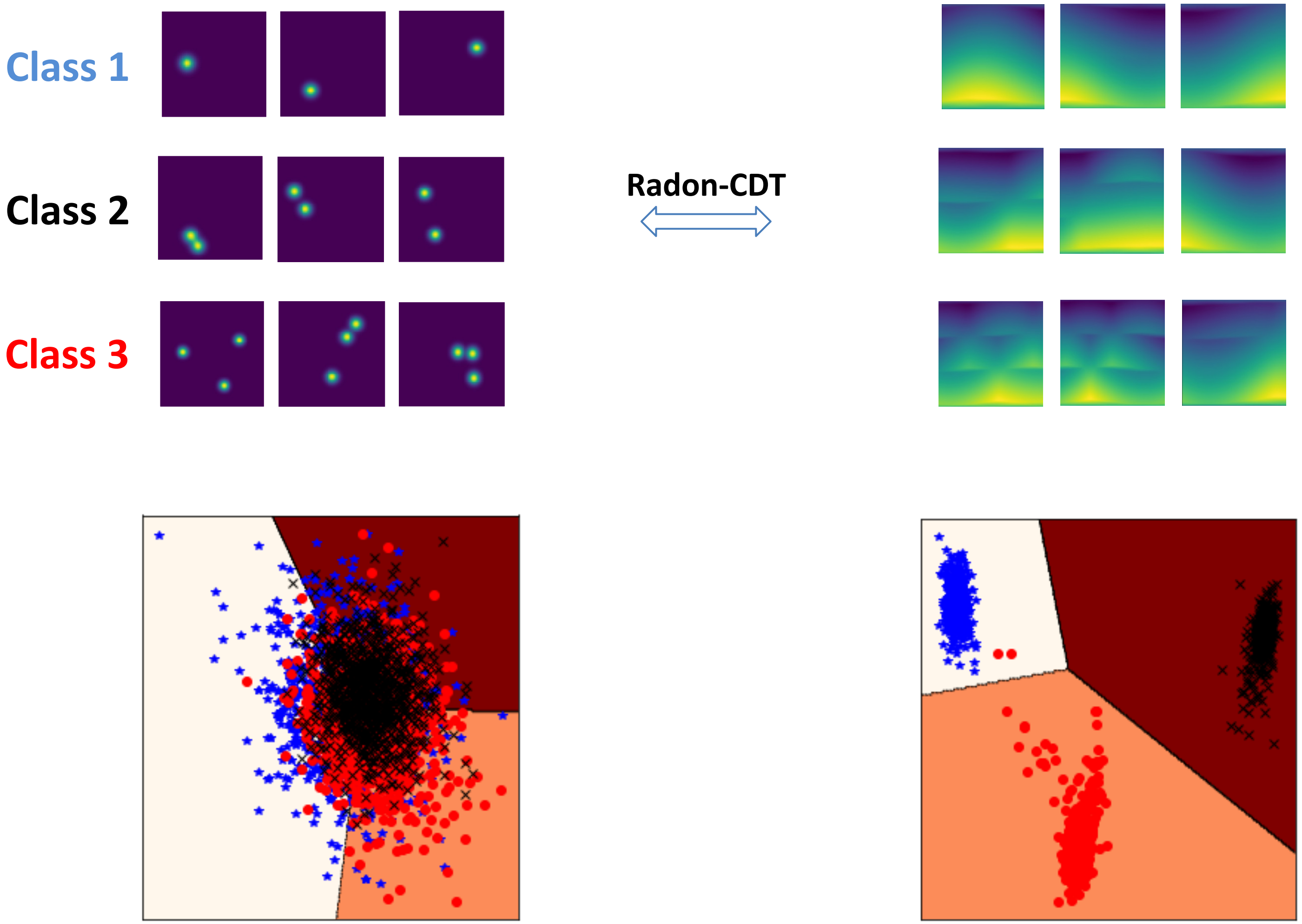
$\hat{q}_g = \hat{q} + \mu$




low dimensional projection

"confound" $g \in \mathcal{C}$ e.g.: translation $g(x) = x - \mu$

# 2D Example




LDA subspace of original data

LDA subspace of transformed data

# CDT: linear separation theorem

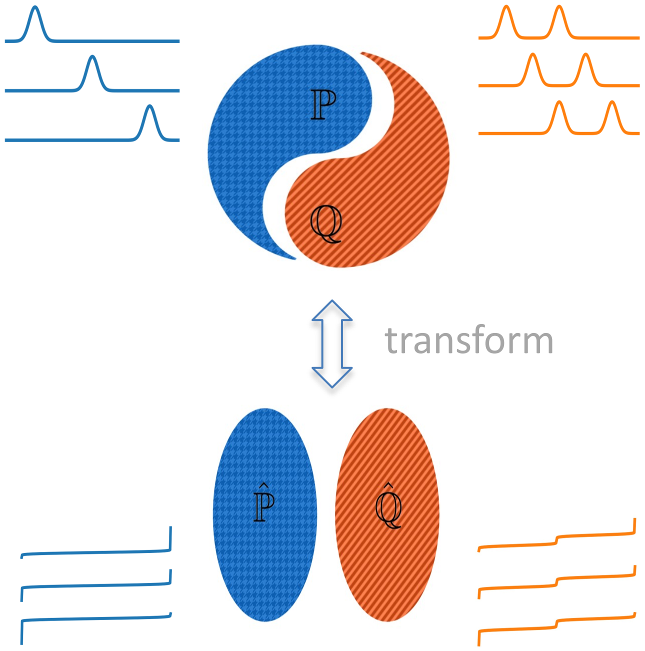
Consider signal classes:

$\mathbb{P} = \{p_g : g \in \mathcal{C}\}$ and $\mathbb{Q} = \{q_g : g \in \mathcal{C}\}$

where $p_g(x) = g'(x)p_0(g(x))$

$q_g(x) = g'(x)q_0(g(x))$

$\mathbb{P} \cap \mathbb{Q} = \emptyset$




Theorem:

$\hat{\mathbb{P}}$, $\hat{\mathbb{Q}}$ will be *convex*

iff $\mathcal{C}^{-1}$ is convex.

for any reference $s_0$

does not require knowledge of $p_0$ $q_0$

Corollary: $\hat{\mathbb{P}}$, $\hat{\mathbb{Q}}$ linearly separable

if $\mathcal{C}^{-1}$ is convex.

[Park, Kolouri, Kundu, Rohde, ACHA 18]
[Aldroubi, Li, Rohde, SaSiDa, 21]
[Shifat-E-Rabbi, …, Rohde, JMIV 21]
[Aldroubi et al, AIMS Foundations of Data Science, 22]

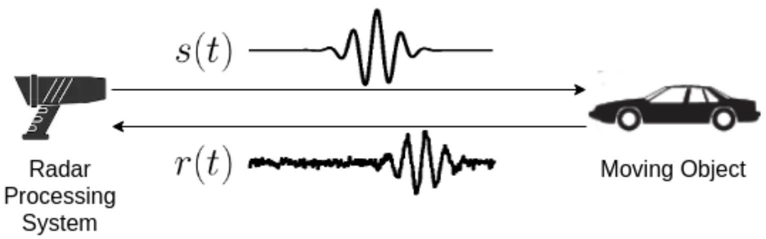
To see this: $\widehat{\mathbb{P}} = \{g^{-1}(\hat{p}(y)) | g^{-1} \in \mathcal{C}^{-1}\}$

$\widehat{\mathbb{Q}} = \{g^{-1}(\hat{q}(y)) | g^{-1} \in \mathcal{C}^{-1}\}$

$\Longrightarrow$ convex iff $\mathcal{C}^{-1}$ convex

# Applications

# Parametric signal estimation




Static object: $r(t) = s(t - \tau) + \eta(t)$

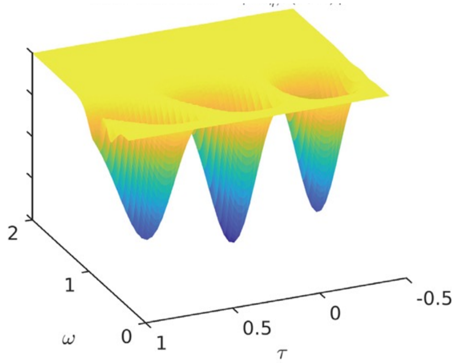
Moving object: $r(t) = s(\omega t - \tau) + \eta(t)$

$g_p(t) = \omega t - \tau$

## Traditional approach:

$$A_{r,s}(\tau, \omega) = \sqrt{\omega} \int r(t) s(\omega t - \tau) dt$$

$$\arg\min_{\tau,\omega} -|A_{r,s}(\tau, \omega)|$$

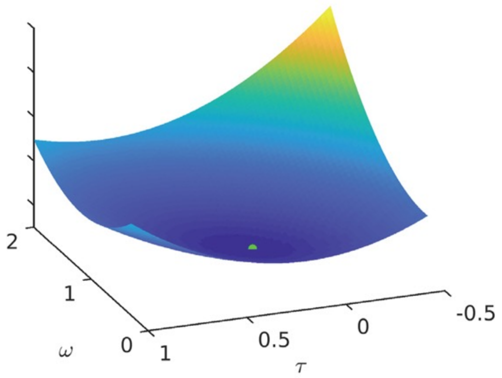



→ computationally expensive
global optimization required

## Wasserstein minimization

$$W^2(r, s) = \|\omega \hat{r} - \tau - \hat{s}\|^2$$

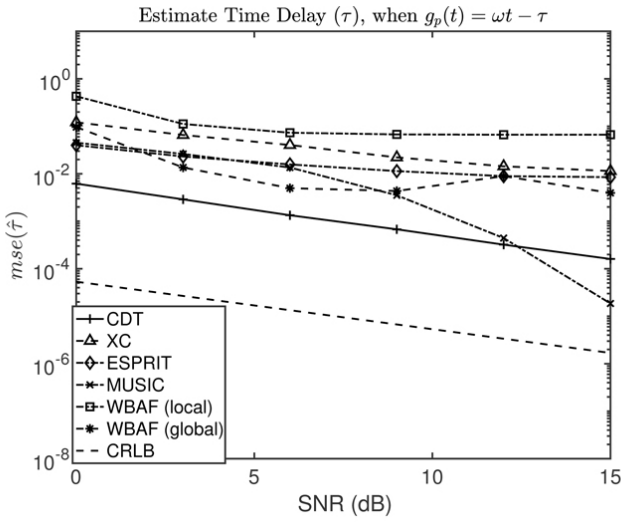
$$\arg\min_{\tau,\omega} \|\omega \hat{r} - \tau - \hat{s}\|^2$$

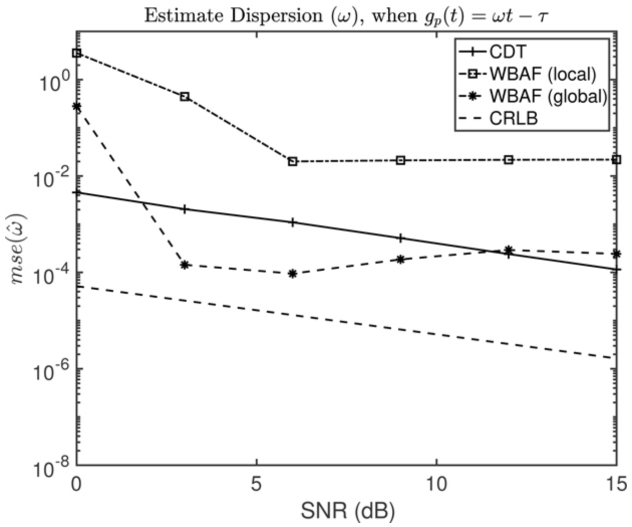


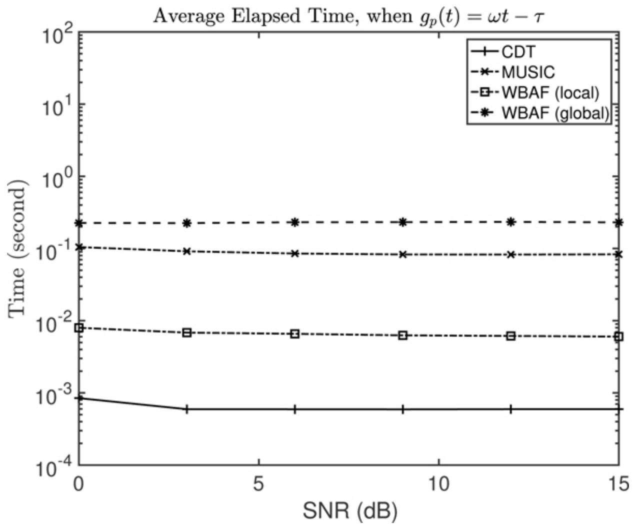

→ linear least squares
cheap to compute

AHM Rubaiyat, KM Hallam, JM Nichols, MN Hutchinson, S Li, GK Rohde, Parametric Signal Estimation Using the Cumulative Distribution Transform, IEEE Trans. Signal Processing, 68, pp. 3312-3324, 2020.

# Parametric signal estimation



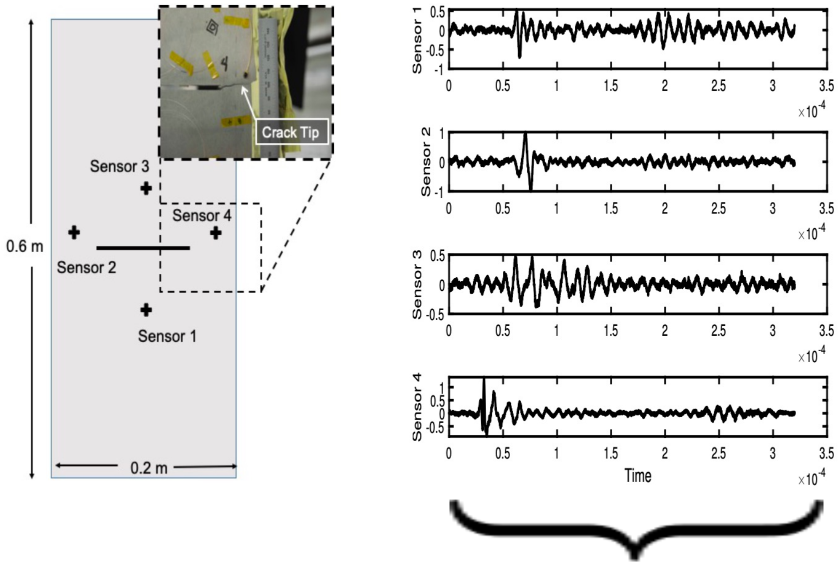




orders of magnitude cheaper to compute

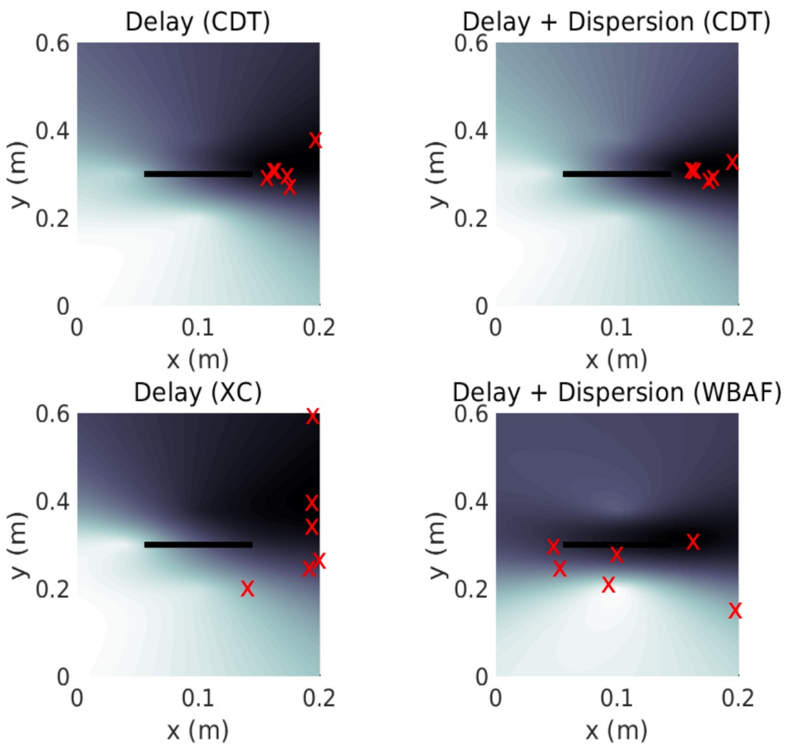
AHM Rubaiyat, KM Hallam, JM Nichols, MN Hutchinson, S Li, GK Rohde, Parametric Signal Estimation Using the Cumulative Distribution Transform, IEEE Trans. Signal Processing, 68, pp. 3312-3324, 2020.

# Parametric signal estimation in structural health monitoring




Estimate TDOA in CDT domain

Crack localization

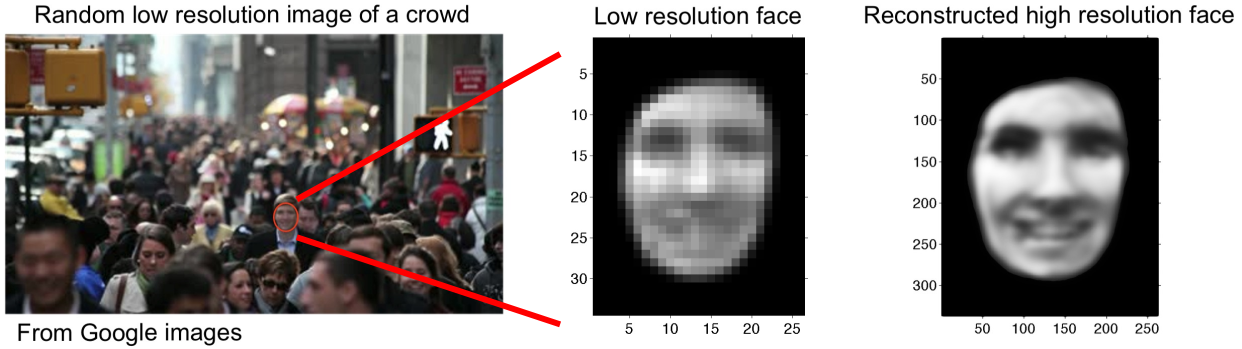



AHM Rubaiyat, KM Hallam, JM Nichols, MN Hutchinson, S Li, GK Rohde, Parametric Signal Estimation Using the Cumulative Distribution Transform, IEEE Trans. Signal Processing, 68, pp. 3312-3324, 2020.

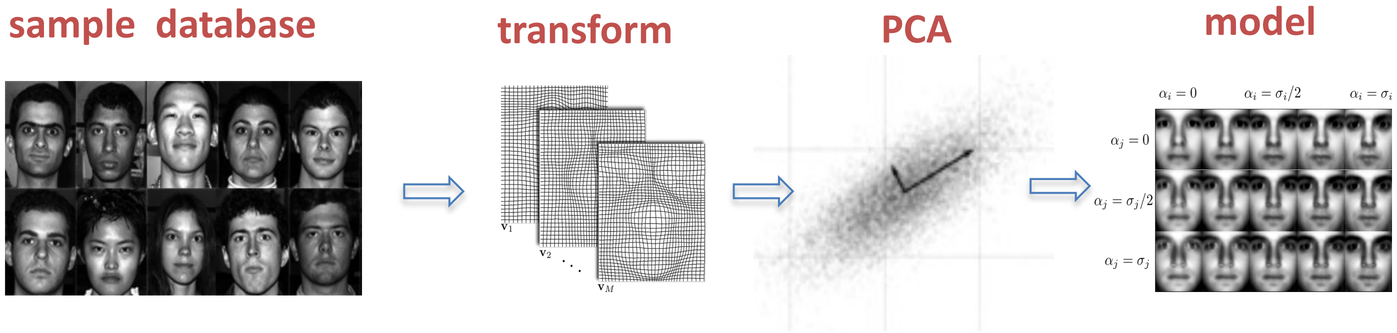
# Model-based inverse problems

## Enhancing an image

### Find model parameters that best match data




## Modeling phase



# Application: single image super resolution

[Kolouri, Rohde, CVPR 2015]

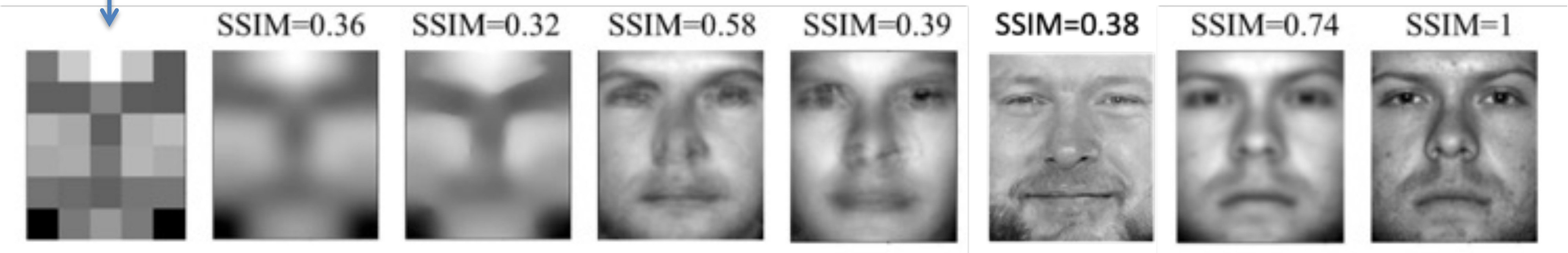
**Model:** $M_\alpha = D_{f_\alpha}(x) I_0(f_\alpha(x))$ $\quad f_\alpha(x) = \underbrace{\sum_k \alpha_k \phi_k(x)}_{\text{Transport PCA trained on high res database}}$

low-res data

**Fit:** $\min_\alpha \|I - V M_\alpha\|^2$

degradation

5x6 pixels!




Input low-res data | B-spline interp. | compressed sensing | K-means nonlinear mapping | Kernel PCA | PULSE generative modeling | transport method | ground truth

# Applications: classification

# Problem statement (supervised learning)

Training: given sample signals/images from different classes

$$\{s_1^{(1)}, s_2^{(1)}, \cdots\} \qquad s_j^{(k)} \in \mathbb{S}^{(k)} = \left\{ s_j^{(k)} | s_j^{(k)} = |\det J g_j| \varphi^{(k)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

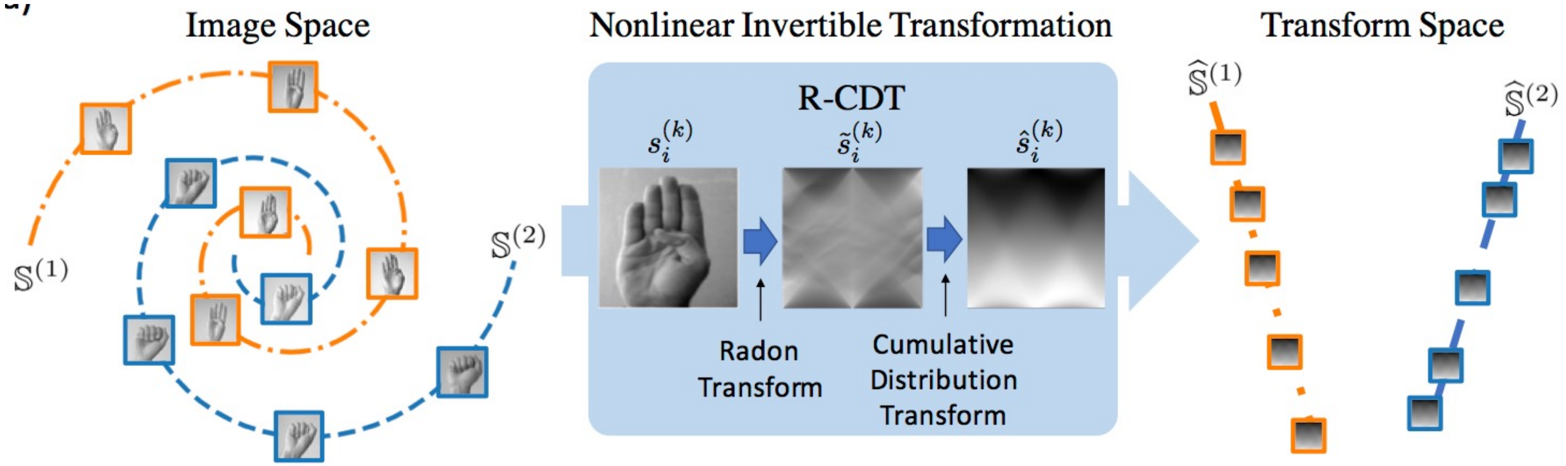
$$\{s_1^{(2)}, s_2^{(2)}, \cdots\}$$

$\varphi^{(k)}$ unknown

$g_j \in \mathcal{G}$ unknown

Testing: find the class that a test sample $s$ originated from.

# Solution overview



# RCDT – nearest subspace

**Training:** estimating basis vectors for subspaces corresponding to each class

training data $\left\{s_1^{(k)}, s_2^{(k)}, \cdots\right\}$ ⇨ transform $\left\{\widehat{s}_1^{(k)}, \widehat{s}_2^{(k)}, \cdots\right\}$ ⇨ deformation model $\left\{\widehat{s}_1^{(k)}, \widehat{s}_2^{(k)}, \cdots\right\} \cup \mathbb{U}_T$ ⇨ orthogonalize $\left[\widehat{b}_1^{(k)}, \widehat{b}_2^{(k)}, \cdots\right] = B^{(k)}$ ⇨ output $B^{(k)}$

**Testing :** predicting the test sample as belonging to the class corresponding to the nearest subspace

test sample $s$ ⇨ transform $\hat{s}$ ⇨ nearest subspace $\arg\min_k ||\widehat{s} - A^{(k)}\widehat{s}||^2$ → class 1, class 2, ⋮

Here, $A^{(k)} = B^{(k)} B^{(k)T}$

Translation $\mathbb{U}_T = \{u_T^{(1)}(t,\theta), u_T^{(2)}(t,\theta)\}$, with $u_T^{(1)}(t,\theta) = \cos\theta$ and $u_T^{(2)}(t,\theta) = \sin\theta$

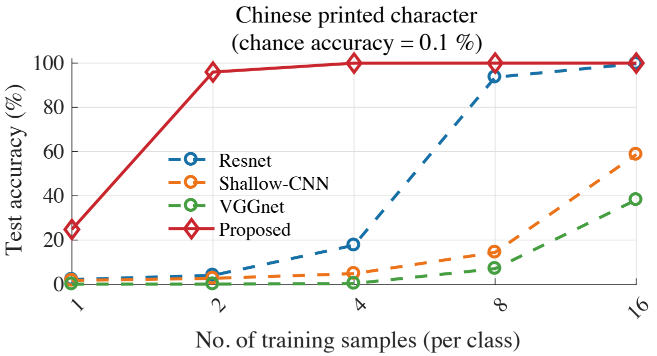
Isotropic scaling $\mathbb{U}_D = \varnothing$

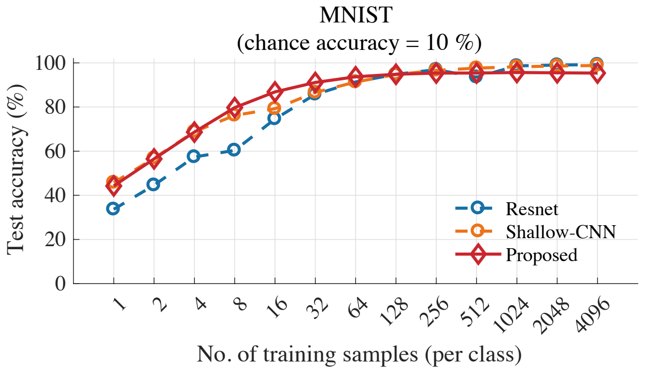
Two important aspects of the method:

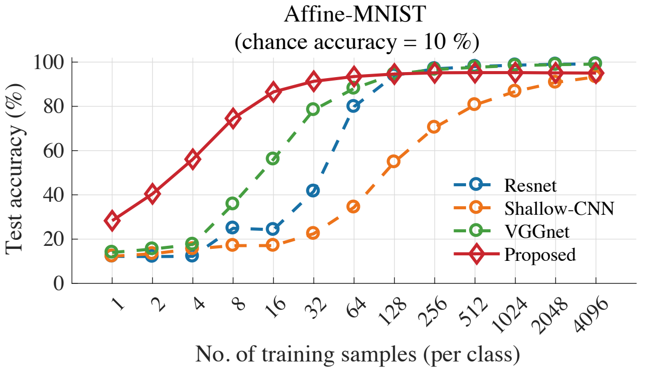
1. The method can learn from datasets
2. The method can learn from mathematically prescribed sets without explicitly using the training data

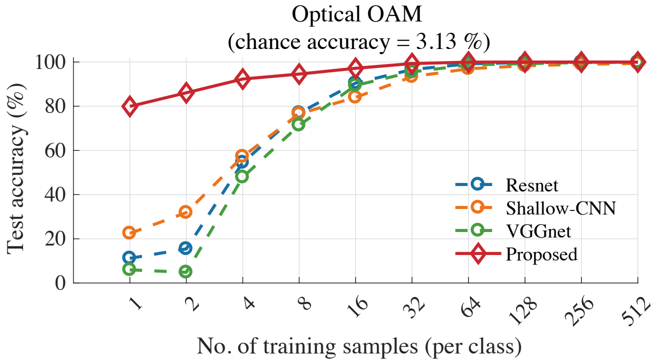
# Results and comparisons with deep learning

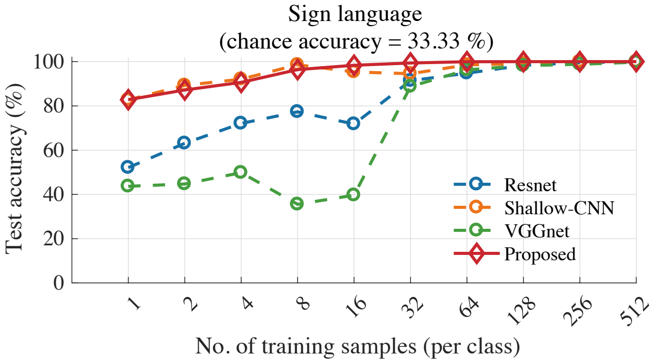
[Shifat-E-Rabbi et al, JMIV 2021]

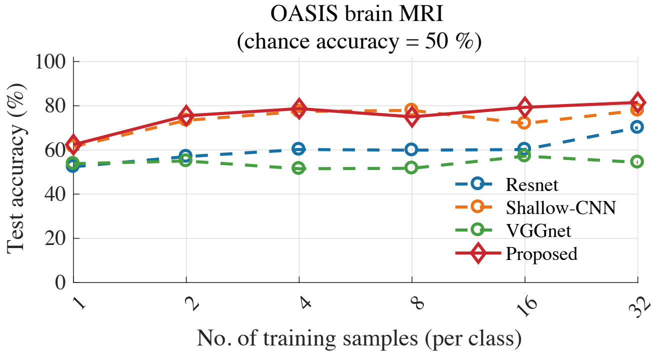


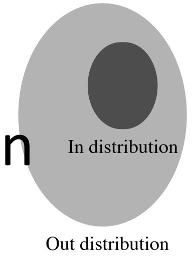



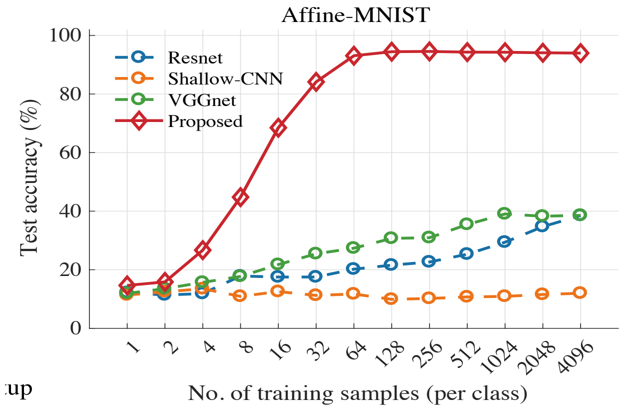



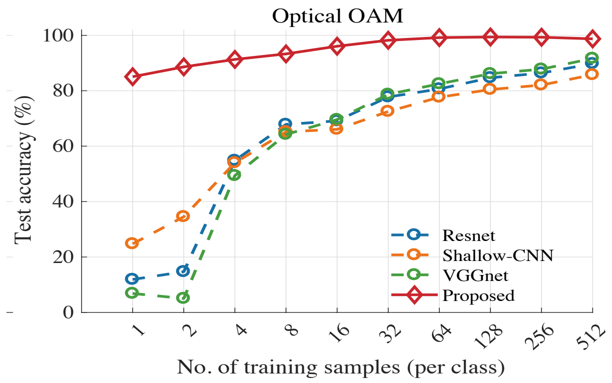



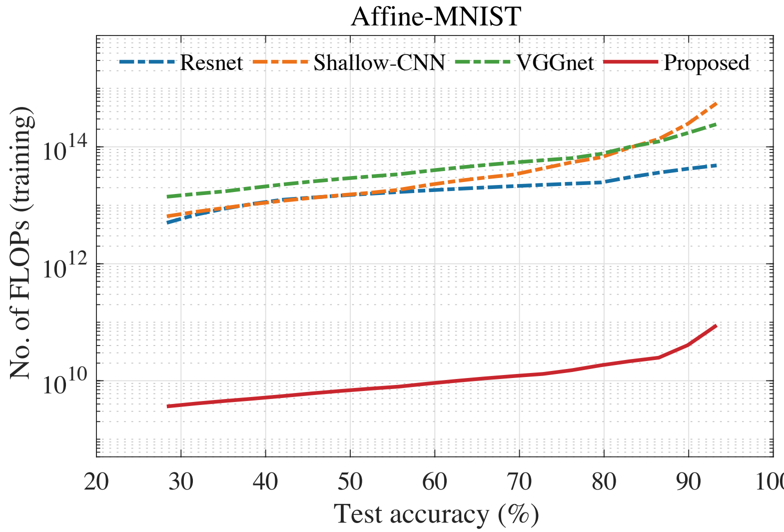

## Out of distribution testing

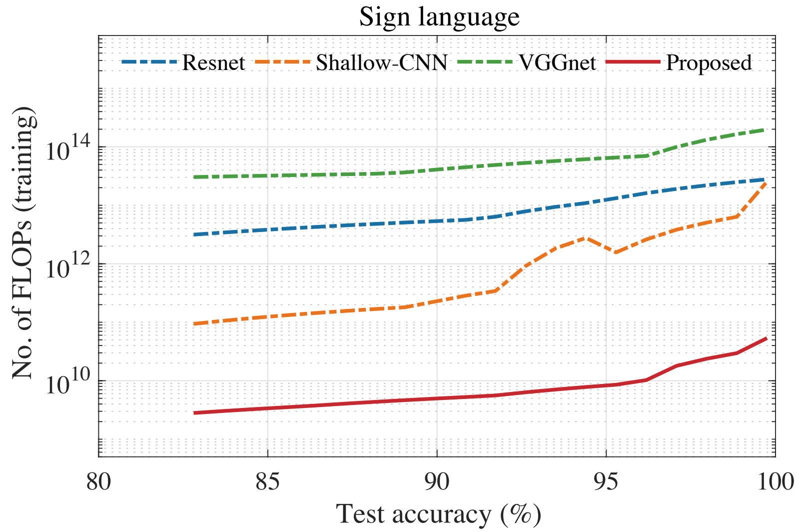


# Results and comparisons with deep learning



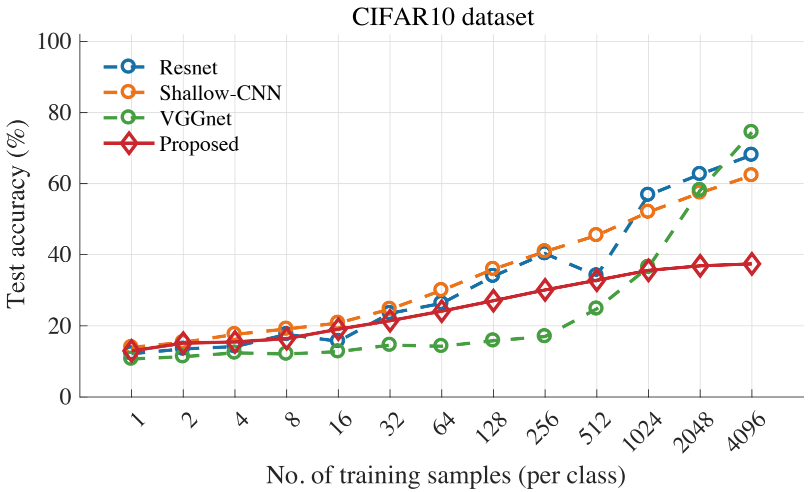

## Limitations

$$\mathbb{P} = \{p_g : g \in \mathcal{C}\} \quad \mathbb{Q} = \{q_g : g \in \mathcal{C}\}$$

where

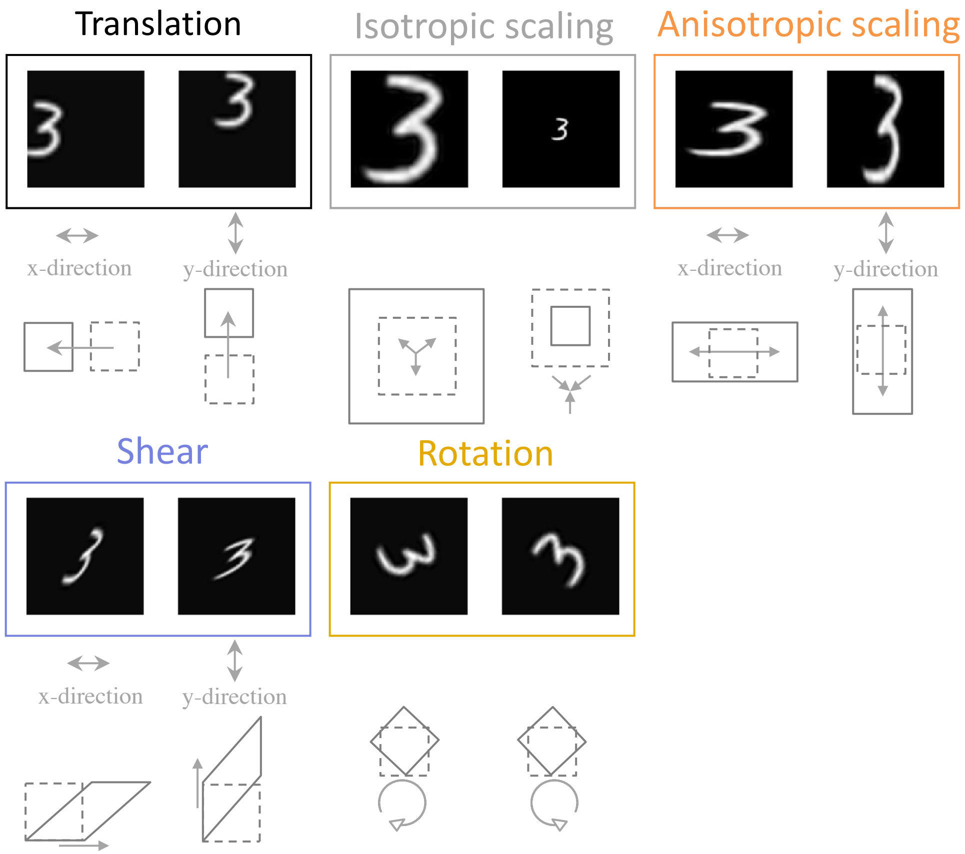
$$p_g(x) = g'(x)p_0(g(x))$$
$$q_g(x) = g'(x)q_0(g(x))$$

It doesn't seem to work as well for general object detection



# Invariance encoding

Aim: Can we mathematically model invariances in data?

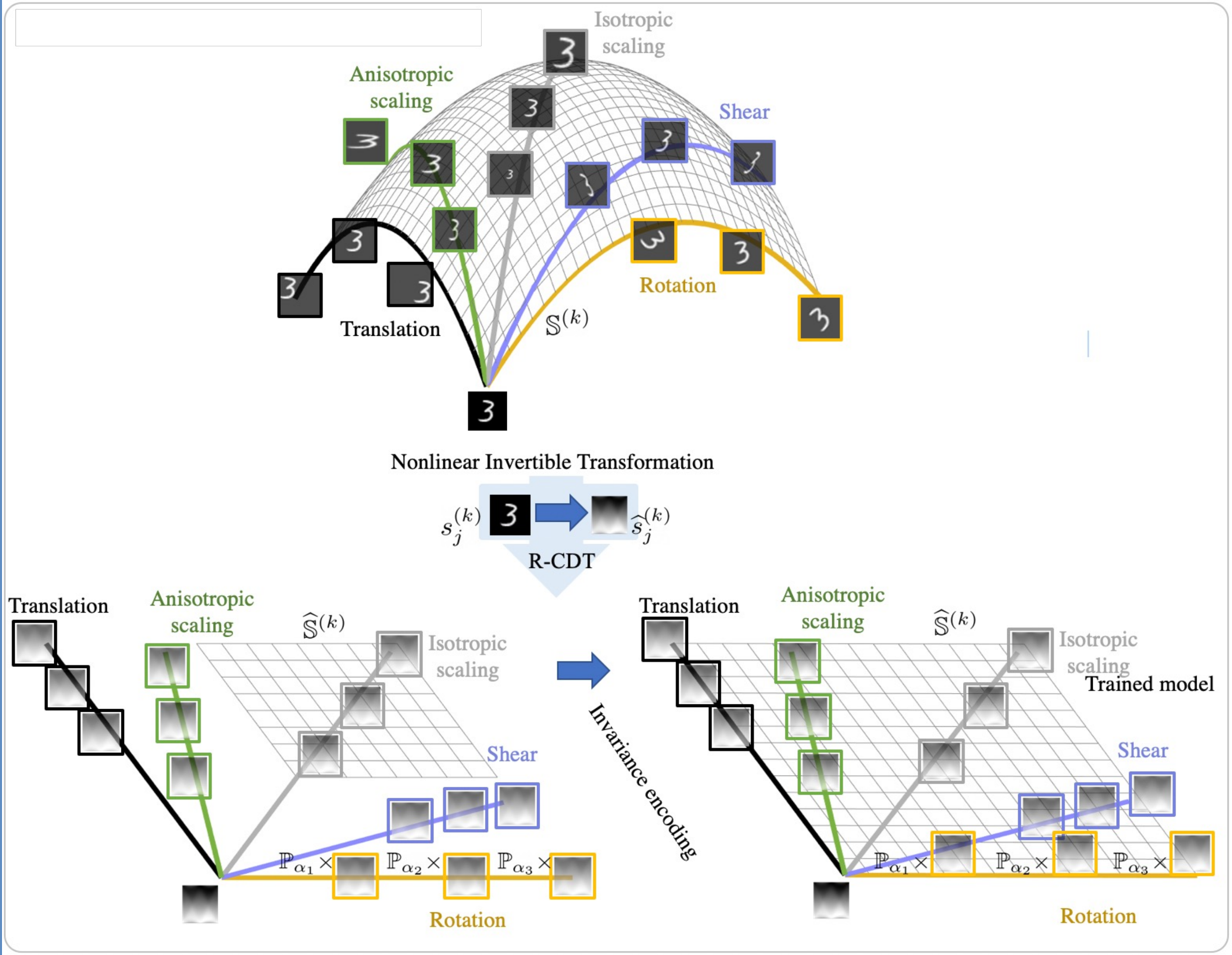



$$\{s_1^{(1)}, s_2^{(1)}, \cdots\} \quad \{s_1^{(2)}, s_2^{(2)}, \cdots\}$$

$$s_j^{(k)} \in \mathbb{S}^{(k)} = \left\{ s_j^{(k)} | s_j^{(k)} = |\det J g_j| \varphi^{(k)} \circ g_j, \forall g_j \in \mathcal{G} \right\}$$

$\varphi^{(k)}$ unknown $\qquad g_j \in \mathcal{G}$ unknown + affine

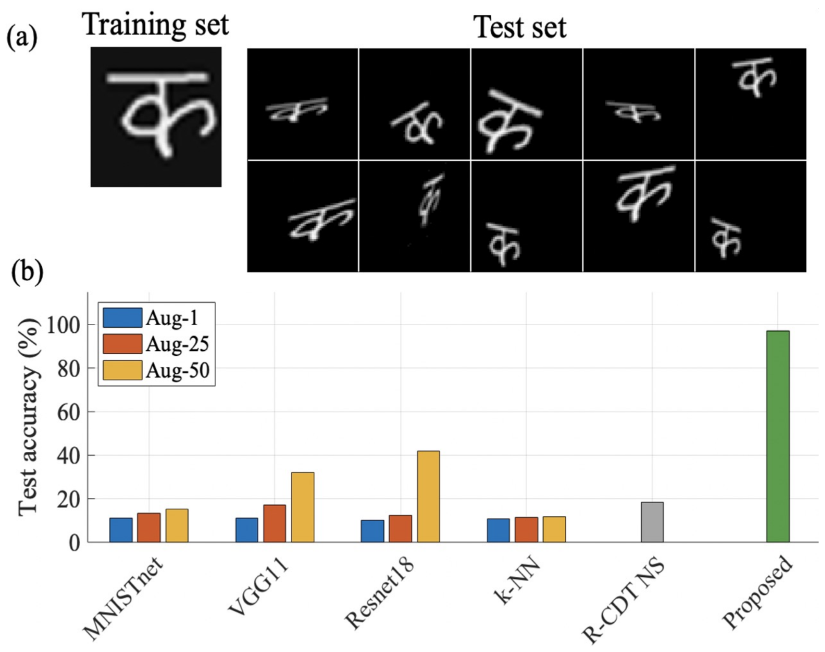
Solution overview:



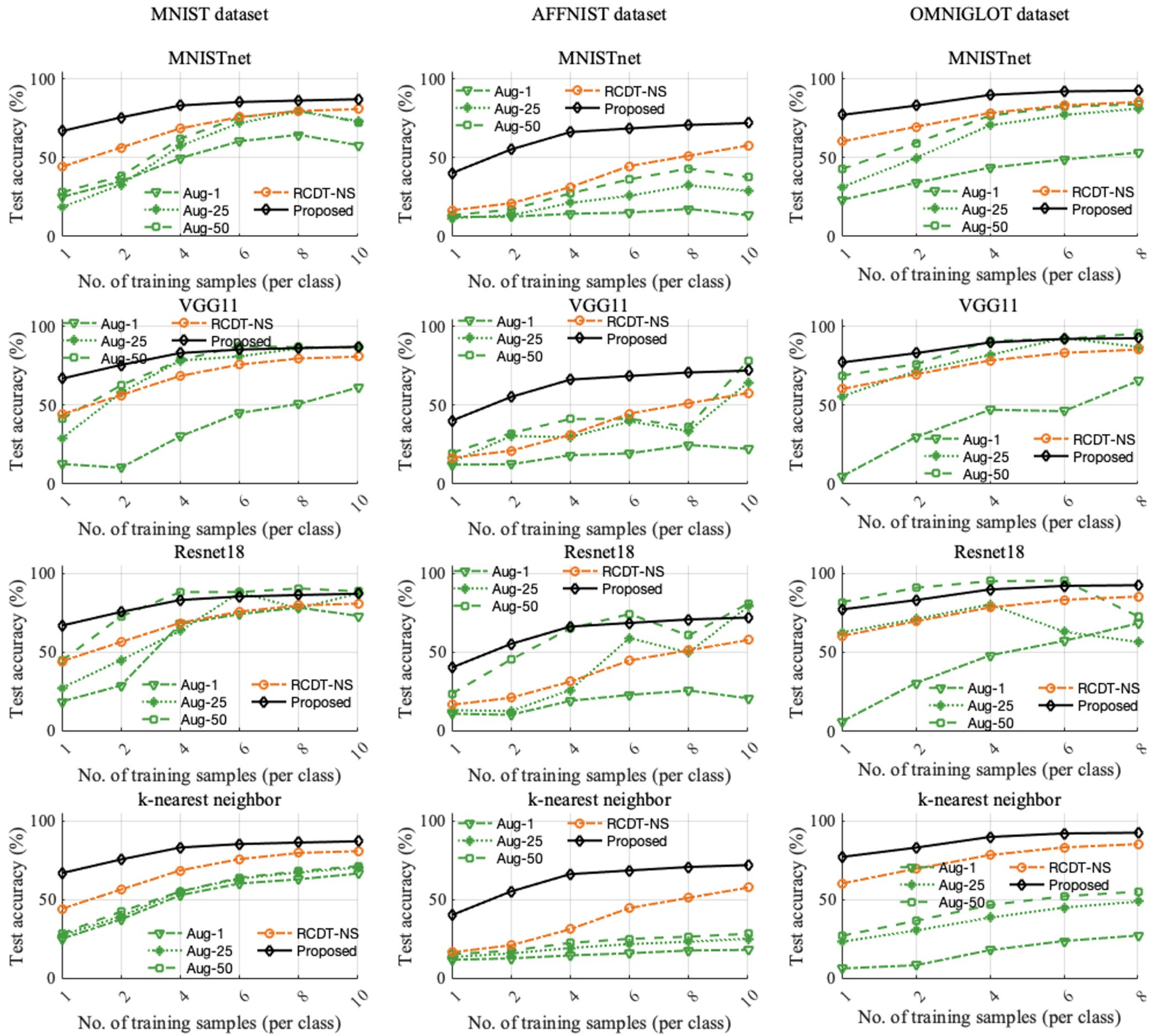
# Results and comparisons with deep learning

[Shifat-E-Rabbi et al, arxiv 2022]

## Synthetic data



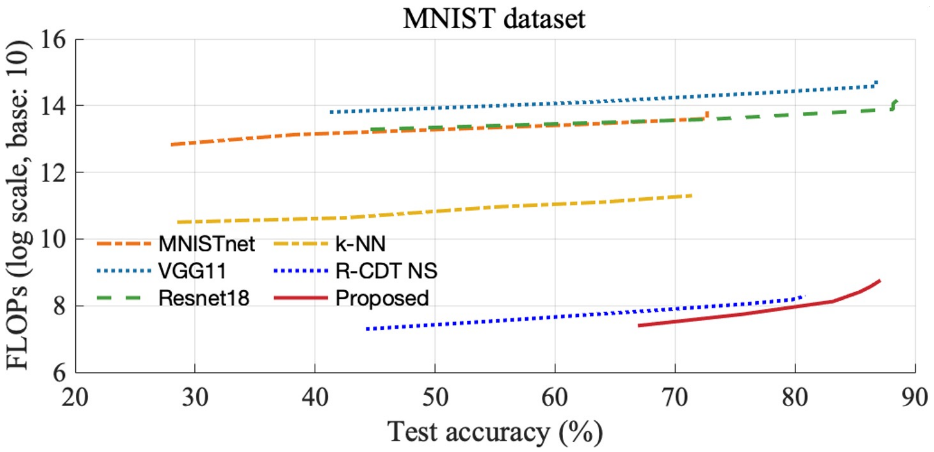

## Real data



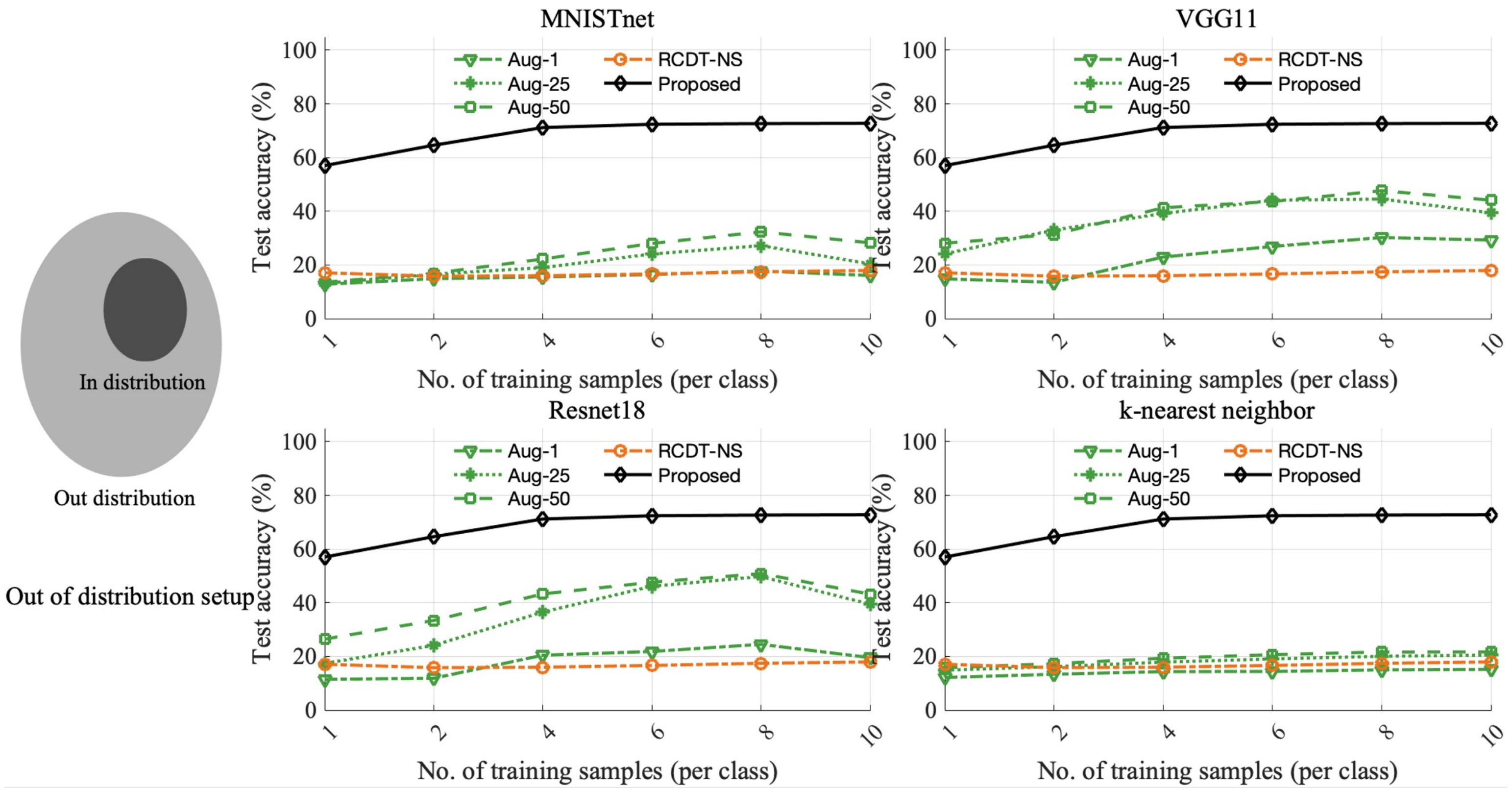
# Results and comparisons with deep learning

[Shifat-E-Rabbi et al, arxiv 2022]

## Computational complexity

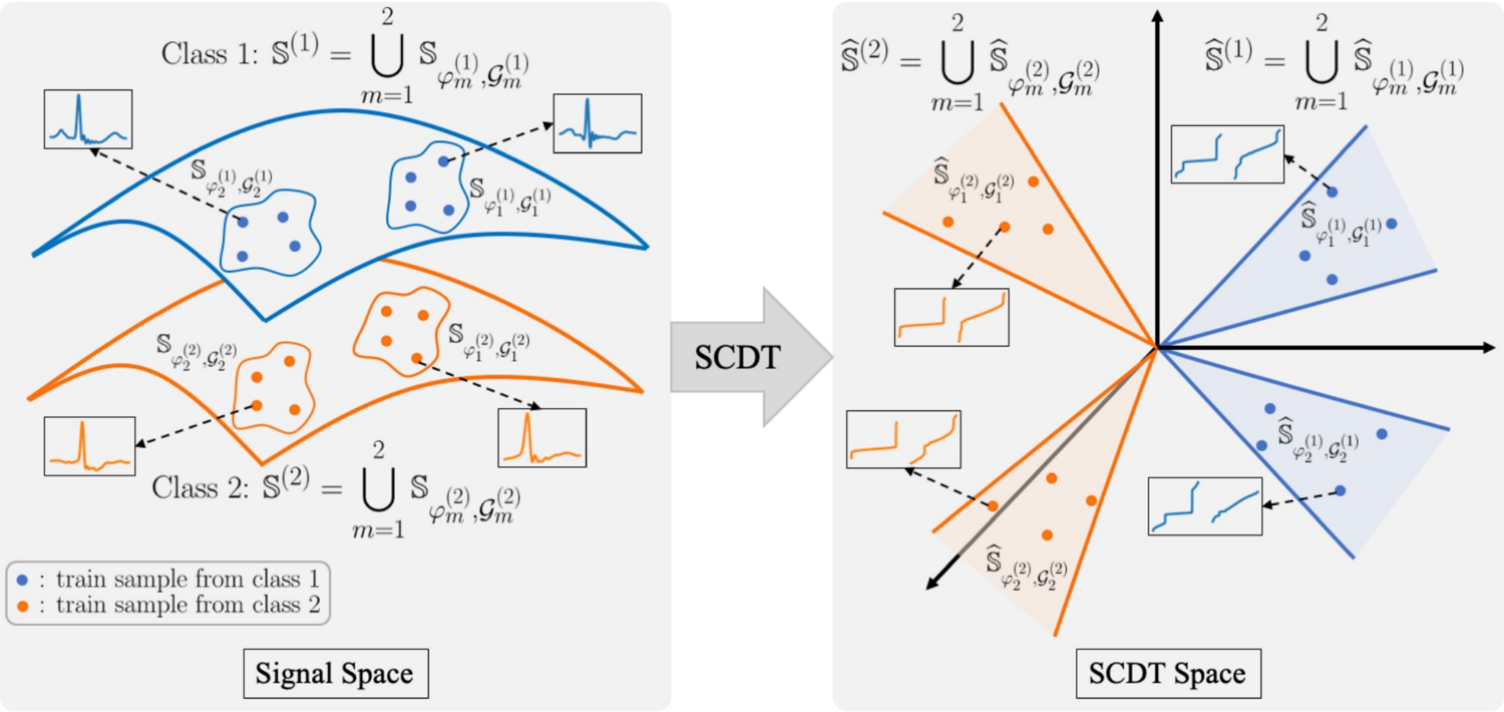



## Out-of-distribution experiment



# End-to-End Time Series Classification [Rubaiyat et al, ArXiv 2022]

Generative Model:




$$\mathbb{S}^{(c)} = \bigcup_{m=1}^{M_c} \mathbb{S}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}},$$

$$\mathbb{S}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}} = \left\{ s_{j,m}^{(c)} | s_{j,m}^{(c)} = g_j' \varphi_m^{(c)} \circ g_j, g_j' > 0, g_j \in \mathcal{G}_m^{(c)} \right\},$$

$$\left(\mathcal{G}_m^{(c)}\right)^{-1} = \left\{ \sum_{i=1}^{k} \alpha_i f_{i,m}^{(c)}, \alpha_i \geq 0 \right\}$$

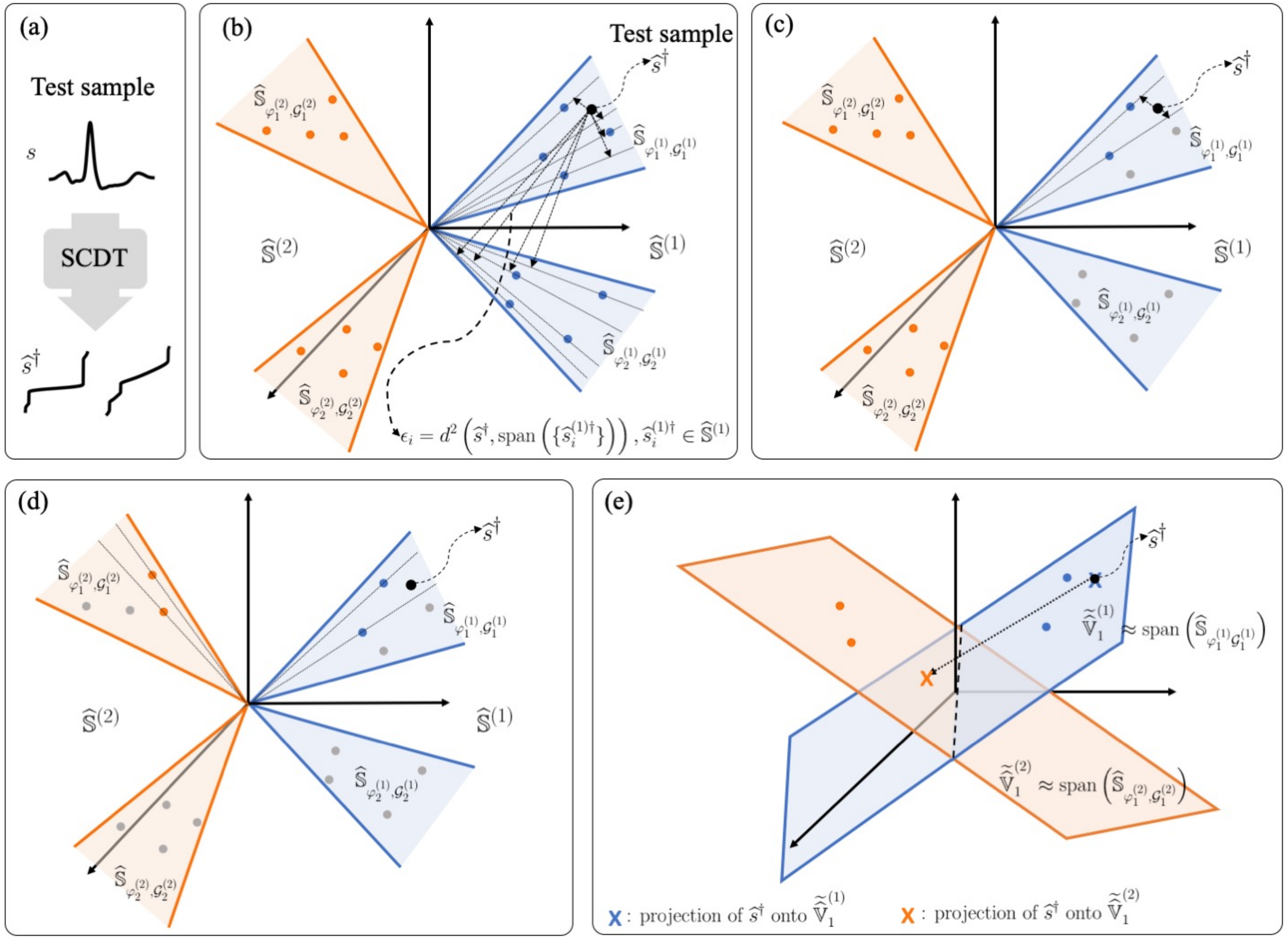
$$\widehat{\mathbb{S}}^{(c)} = \bigcup_{m=1}^{M_c} \widehat{\mathbb{S}}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}},$$

$$\widehat{\mathbb{S}}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}} = \left\{ \widehat{s}_{j,m}^{(c)} | \widehat{s}_{j,m}^{(c)} = g_j^{-1} \circ \widehat{\varphi}_m^{(c)}, g_j \in \mathcal{G}_m^{(c)} \right\},$$

$$\left(\mathcal{G}_m^{(c)}\right)^{-1} = \left\{ \sum_{i=1}^{k} \alpha_i f_{i,m}^{(c)}, \alpha_i \geq 0 \right\}$$

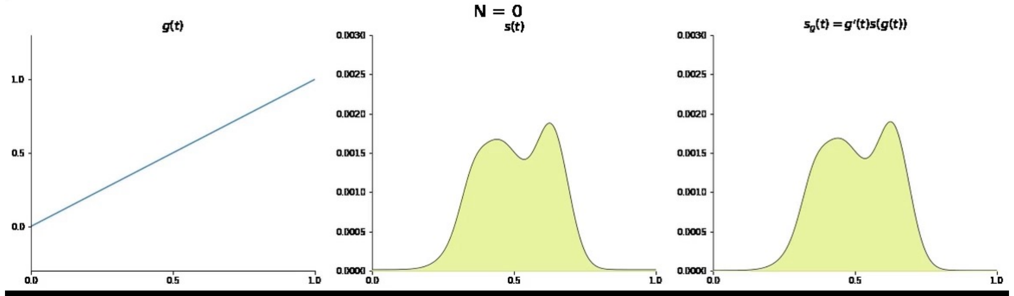
# End-to-End Time Series Classification [Rubaiyat et al, ArXiv 2022]

## NLS algorithm:

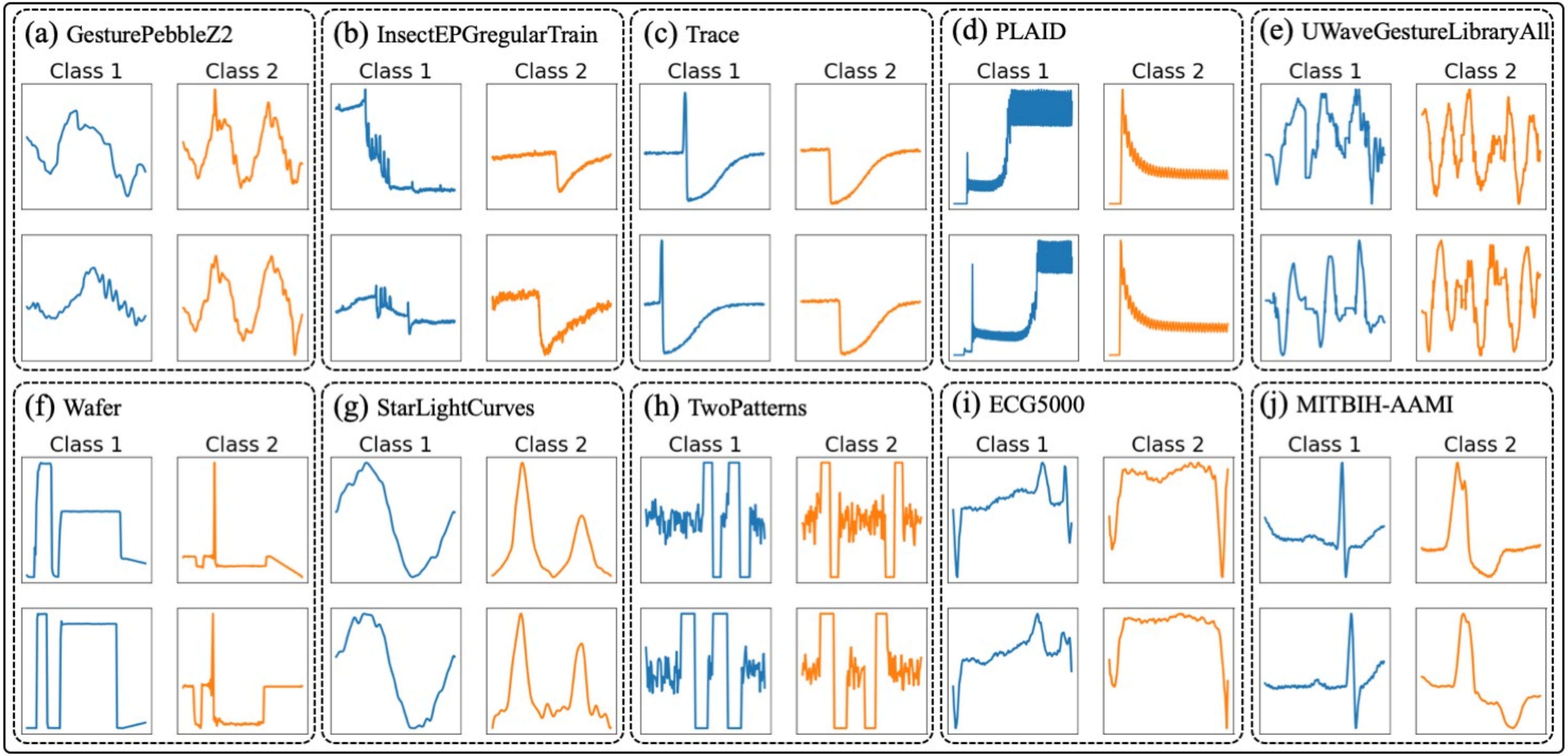



## Subspace Enrichment:

$$\widetilde{\widehat{\mathbb{V}}}_m^{(c)} = \text{span}\left(\{\widehat{z}_1^{(c)\dagger}, \cdots, \widehat{z}_k^{(c)\dagger}\} \cup \mathbb{U}_T \cup \mathbb{U}_H\right)$$



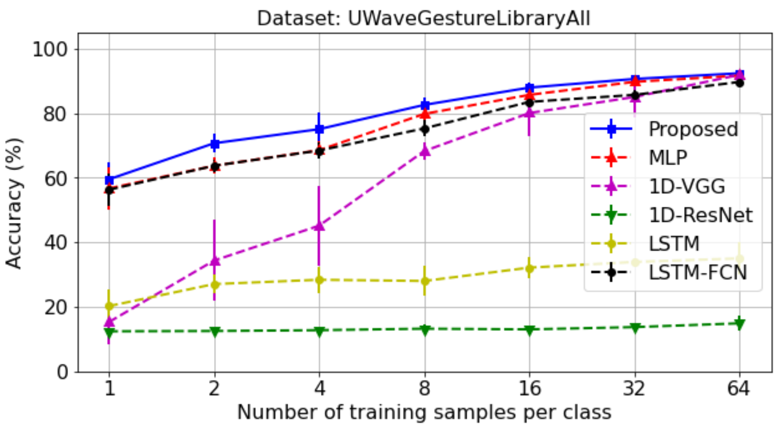

Solution is given by: $\underset{c}{\arg\min}\ d^2\left(\widehat{s}, \widetilde{\widehat{\mathbb{V}}}_m^{(c)}\right)$

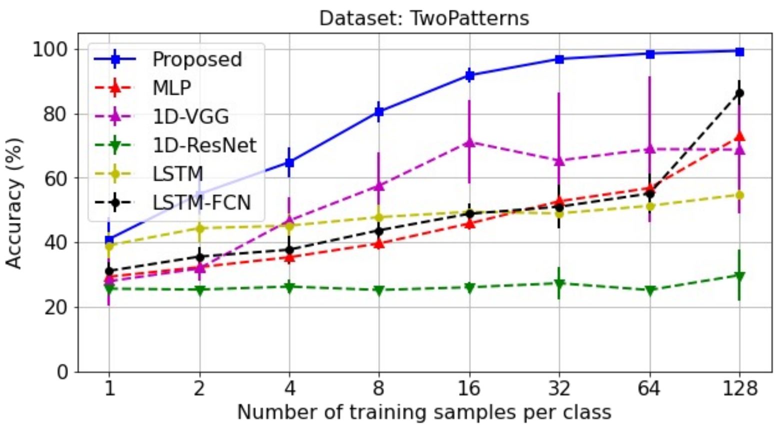

(a) GesturePebbleZ2
Class 1
Class 2
(b) InsectEPGregularTrain
Class 1
Class 2
(c) Trace
Class 1
Class 2
(d) PLAID
Class 1
Class 2
(e) UWaveGestureLibraryAll
Class 1
Class 2
(f) Wafer
Class 1
Class 2
(g) StarLightCurves
Class 1
Class 2
(h) TwoPatterns
Class 1
Class 2
(i) ECG5000
Class 1
Class 2
(j) MITBIH-AAMI
Class 1
Class 2

# Results and comparisons with deep learning

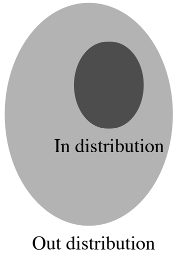
[Rubaiyat et al, ArXiv 2022]

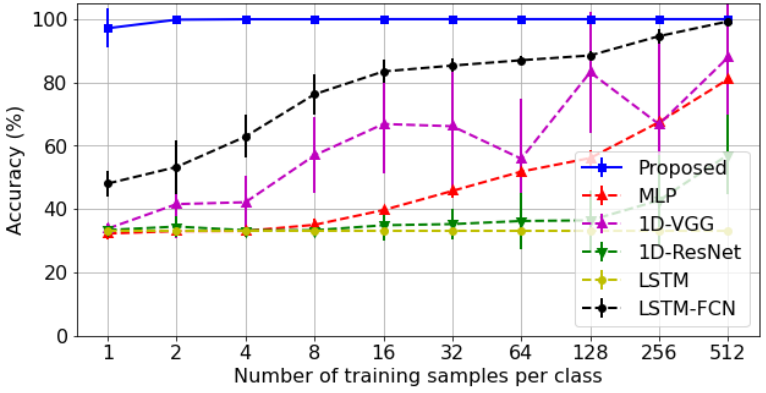
| Dataset | MLP | 1D-VGG | 1D-ResNet | LSTM | LSTM-FCN | SCDT-NS | **Proposed** |
|---|---|---|---|---|---|---|---|
| GesturePebbleZ2 | 83.92 | 48.99 | 63.23 | 60.19 | 85.82 | 81.64 | **86.07** |
| InsectEPGRegularTrain | 80.92 | 82.07 | 50.58 | 57.05 | 59.95 | 73.43 | **90.82** |
| Trace | 64.0 | 35.8 | 25.9 | 28.2 | 58.1 | 85.0 | **88.0** |
| PLAID | 39.65 | 19.35 | 29.81 | 26.35 | 39.57 | 58.29 | **70.02** |
| UWaveGestureLibraryAll | 94.74 | **96.42** | 42.86 | 30.72 | 94.16 | 90.4 | 94.92 |
| Wafer | 68.69 | 94.03 | 53.66 | 51.36 | 86.70 | 92.75 | **95.53** |
| StarLightCurves | 75.85 | 70.0 | 56.53 | 60.54 | 74.38 | 77.4 | **84.2** |
| TwoPatterns | 88.12 | 95.15 | 87.42 | 80.62 | 98.47 | 95.15 | **99.92** |
| ECG5000 | 98.9 | **98.92** | **98.92** | 98.36 | 98.76 | 93.4 | 97.4 |
| MITBIH-AAMI | 65.54 | 82.94 | 59.13 | 55.34 | 79.46 | 62.9 | **84.05** |
| Win | 0 | 2 | 1 | 0 | 0 | 0 | **8** |
| AVG arithmetic ranking | 3.5 | 3.5 | 5.6 | 6.3 | 3.6 | 3.8 | **1.6** |
| AVG geometric ranking | 3.42 | 2.8 | 5.07 | 6.27 | 3.46 | 3.49 | **1.28** |
| MPCE | 0.072 | 0.065 | 0.125 | 0.127 | 0.066 | 0.059 | **0.030** |




Out of distribution testing



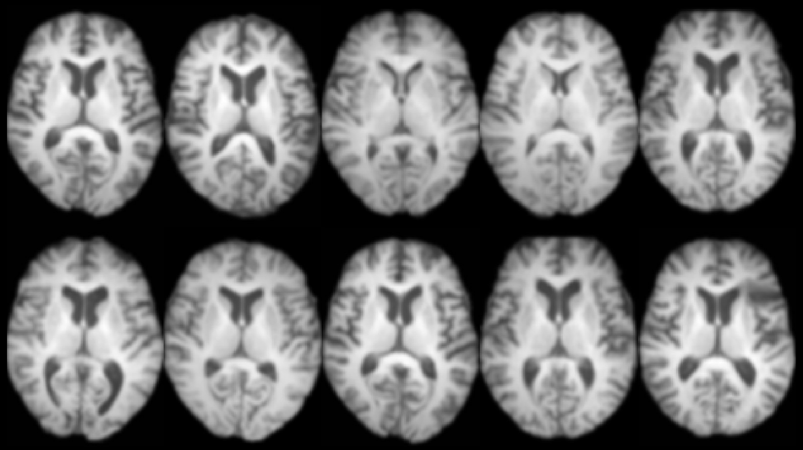
# Transport-based Morphometry (TBM)

# Biomedical image data science

Are fitness and gray matter morphology related?

Class 1:
persons with low fitness index

Class 2:
persons with high fitness index

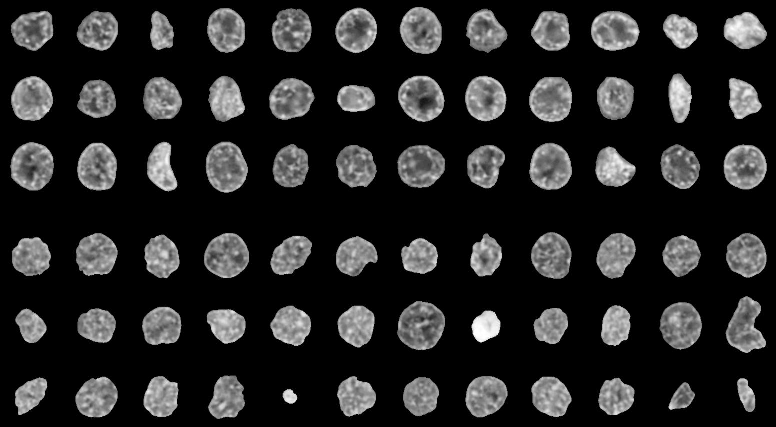


magnetic resonance imaging (MRI)

Is nuclear morphology related to cancer?

Class 1:
nuclei from benign tissues

Class 2:
nuclei from malignant tissues



digital pathology (microscopy)

## Hypothesis testing

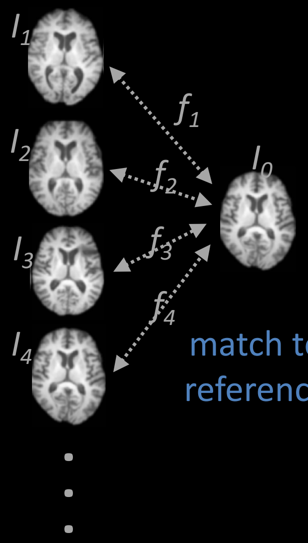
Are there differences?

## Understanding

If so, can we model them?

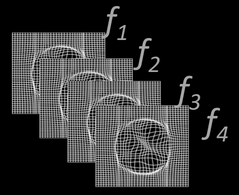
# Transport-based morphometry (TBM)

## 1) Transform




match to reference

transport maps




## 2) regression

modeling variations

- PCA, least squares, …

modeling differences

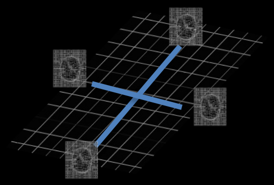
- LDA, SVM, …

modeling relationships

- CCA, …

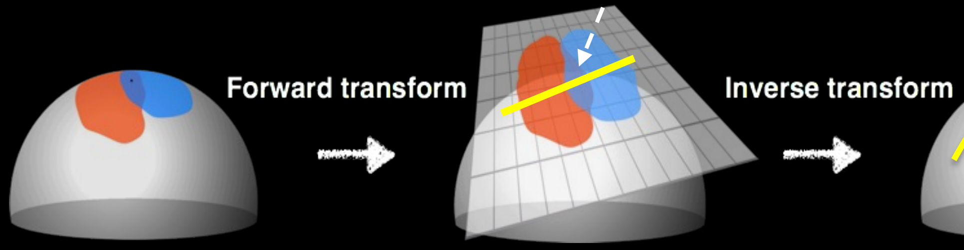
## 3) Inverse transform

PCA, LDA, CCA, …



visualization

statistical regression (PCA, LDA, CCA, …)



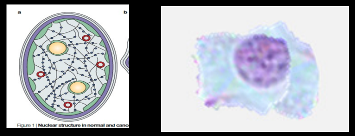

signal space

transform space

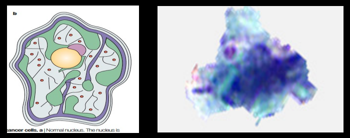
model in image space

# Example: nuclear structure in cancer

Aberrations in genetic code and/or increase in transcriptional activity → Unfolding/repackaging of chromatin → change in nuclear configuration

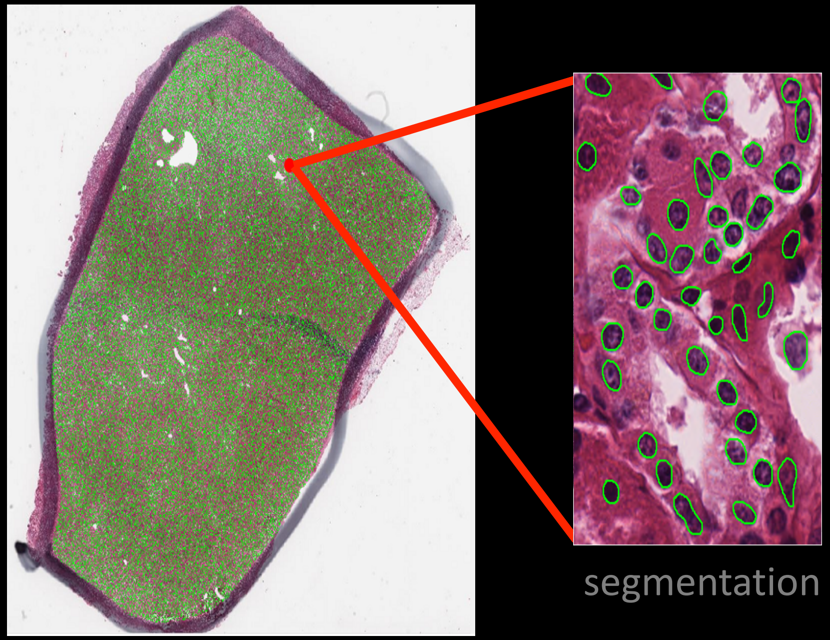


normal



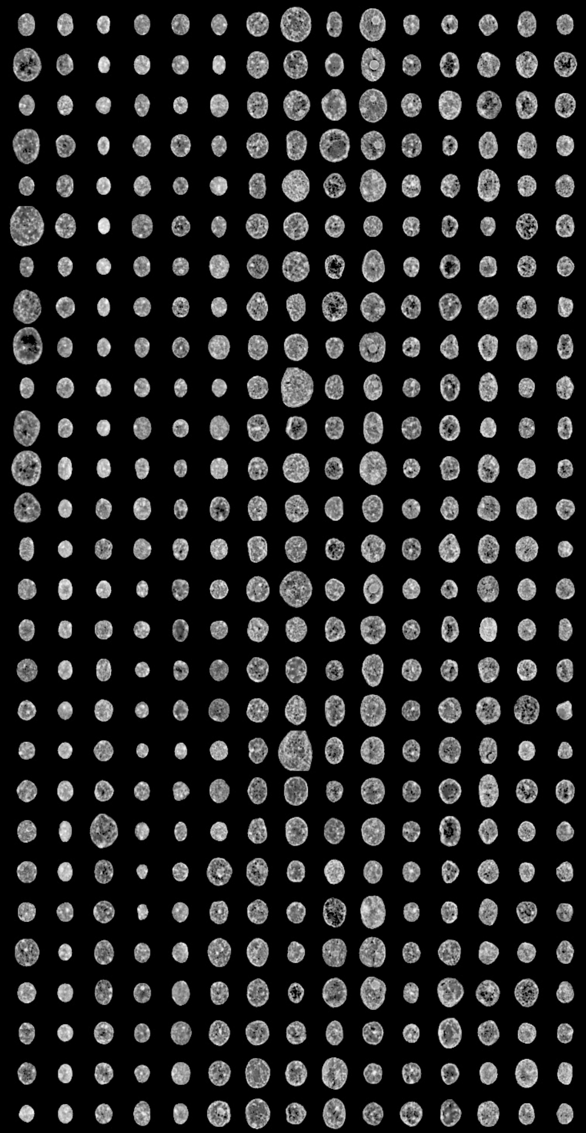
cancer

Pathology, whole slide imaging



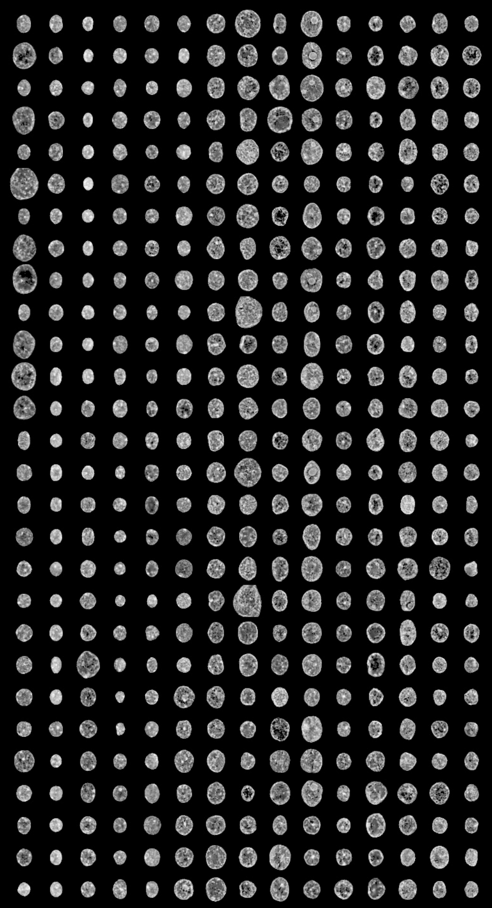
segmentation

segmented nuclei

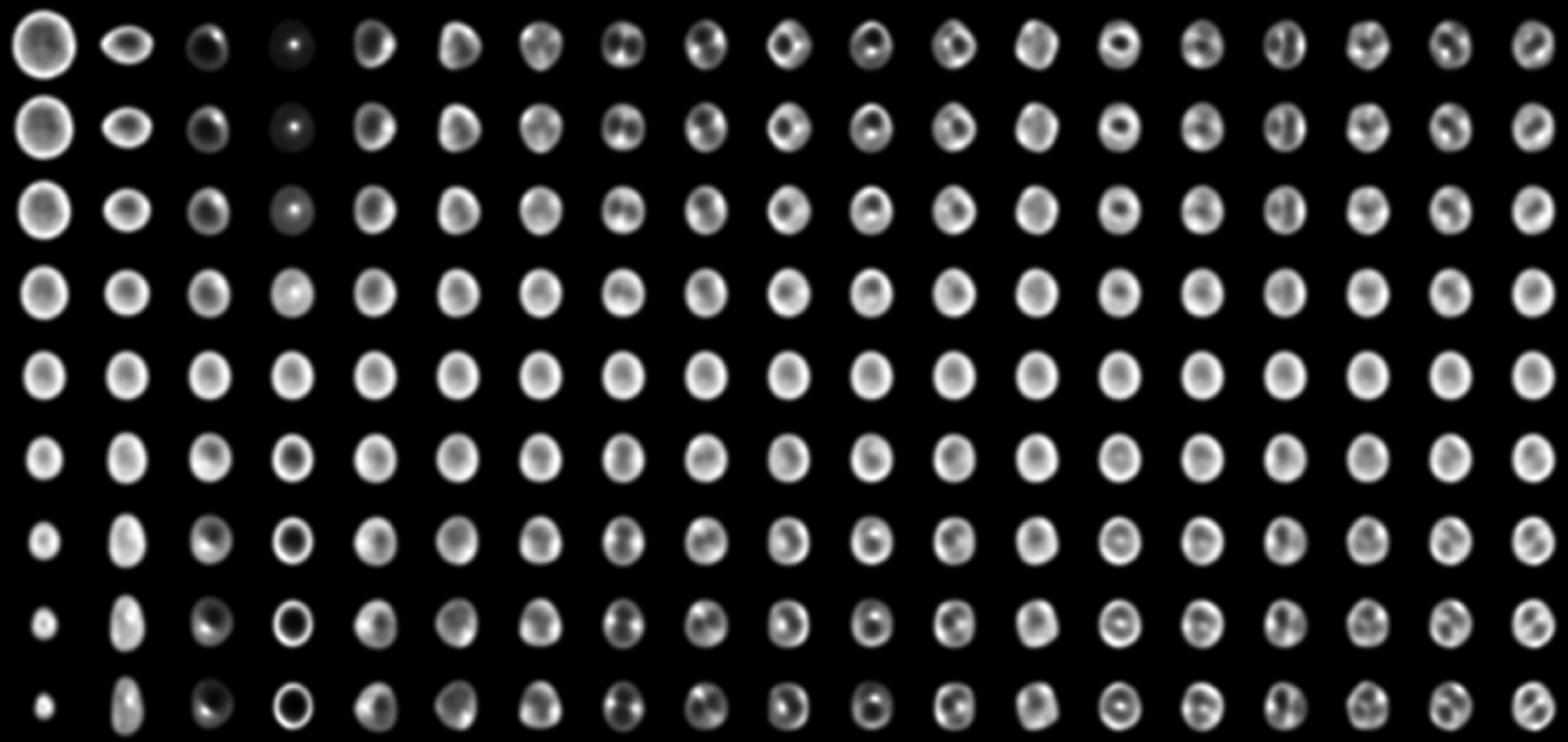


# Exploratory analysis: PCA on transport maps

database



## Modeling chromatin variation in liver cancer (shape + texture)



learned “features”

$$D_{f_\alpha^{-1}}(x)\, s_0(f_\alpha^{-1}(x))$$

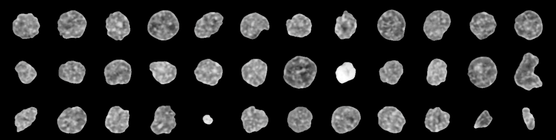
$$f_\alpha(x) = \underbrace{f(x)}_{\text{mean}} + \alpha w(x)$$

PCA eigenvector

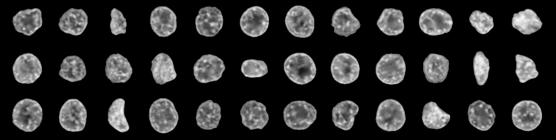
# TBM: modeling discriminant information

[Basu et al, PNAS 2014]

Class 1: benign cells

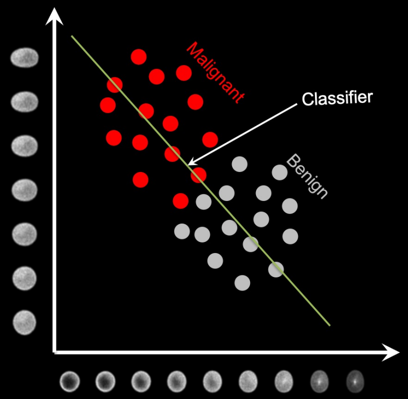


Class 2: malignant cells



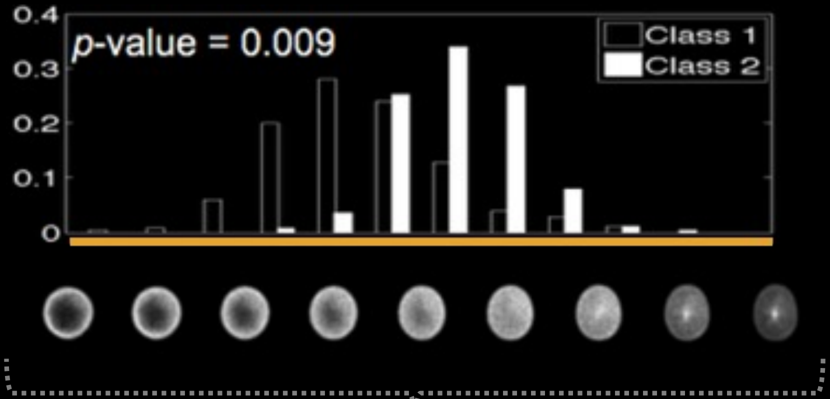
forward

**LDA in transport space**




inverse

Fetal-type hepatoblastoma




## Inverse of classification function

$$D_{f_\alpha^{-1}}(x)\, s_0(f_\alpha^{-1}(x))$$

$$f_\alpha(x) = f(x) + \alpha w(x)$$

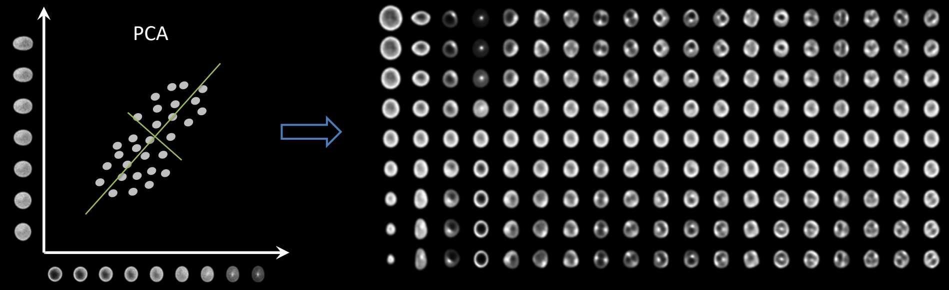
mean

classifier

# Summary: transport-based morphometry

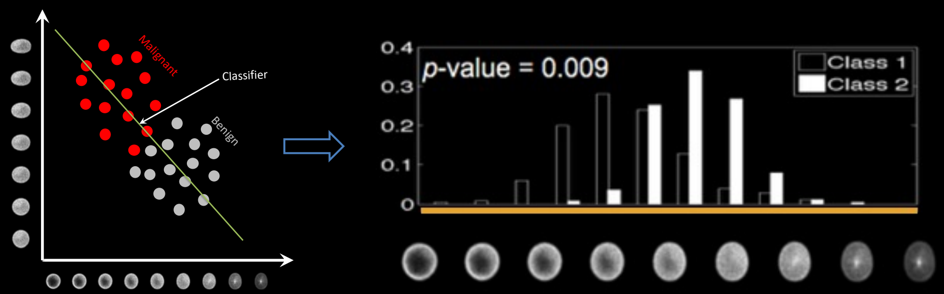
## 1) Exploratory analysis

- Transport PCA, subspace learning, …
- Models texture & shape simultaneously
- Visualizes variations within dataset



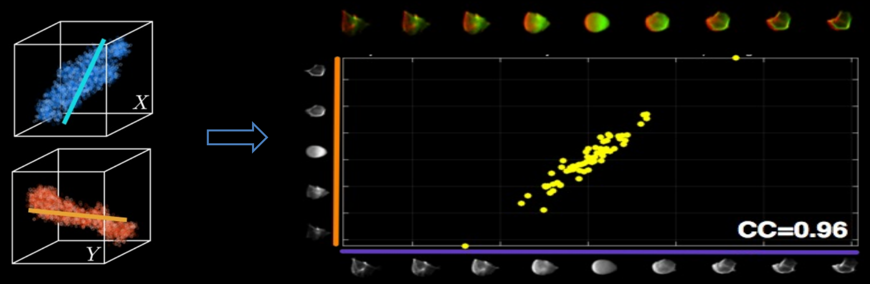

## 2) Discriminant models

- SVM, LDA, pLDA, …
- Visualizes most discriminative information




## 3) Functional relationships

- CCA
- Study & visualize relationship between different measurements



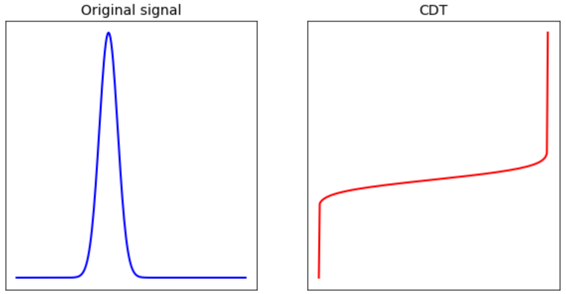
# PyTransKit

Python transport based signal processing toolkit

https://github.com/rohdelab/PyTransKit

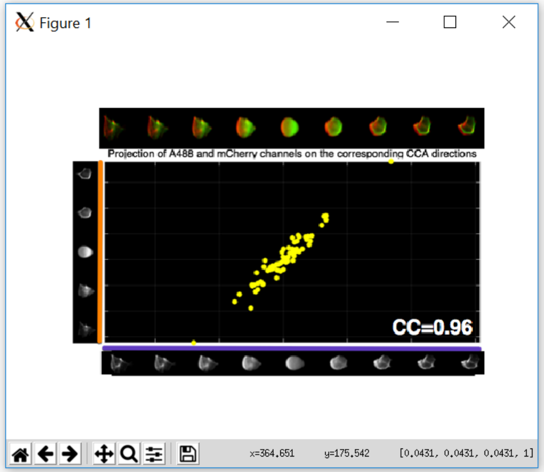
## Low level functions

```
x = np.arange(N)
mu = 100
sigma = 10
I1 = 1 / (simga*np.sqrt(2*np.pi))*np.exp(-((x-mu)**2)/(2*sigma**2))
I1 = abs(I1) + epsilon
I1 = I1/I1.sum()


# compute the forward transform
I1_hat, f1_hat, xtile = cdt.forward(x0, I0, x, I1)
```

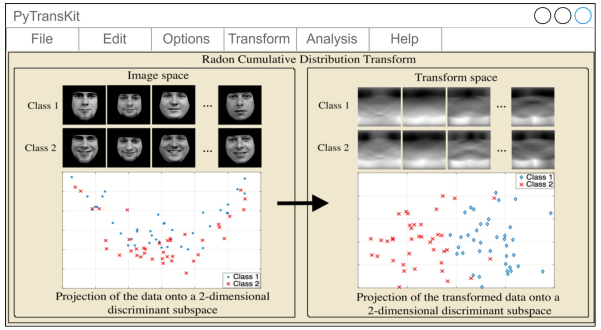



## Command line tools

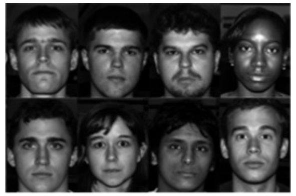
```
# load the neutrophils dataset
x, y = load_data('data/neutrophils')

# R-CDT transform with default parameters
x_t = radoncdt.transform(x)

# CCA analysis, using 2 components
x_t_comps = cca.fit_transform(x_t, y, n_components=2)

# Show results
plt.plot(x_t_comps, y)
```



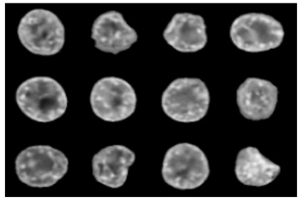

## TBM toolbox



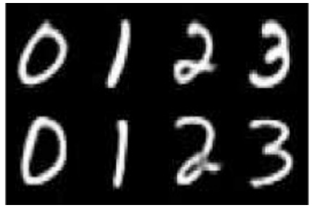
faces



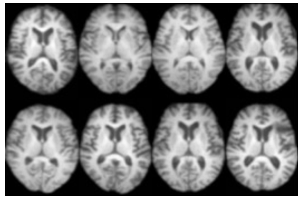
cells



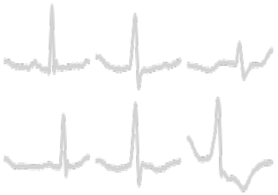
digits



brains



ECG



estimation, detection,
inverse problems, classification

| signal space | | transform space | | signal space |
|---|---|---|---|---|
| problem hard to solve | $\Rightarrow$ | problem easy to solve | $\Rightarrow$ | answer in signal space |

**signal space**

$$\mathbb{S}^{(c)} = \bigcup_{m=1}^{M_c} \mathbb{S}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}},$$

$$\mathbb{S}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}} = \left\{ s_{j,m}^{(c)} \middle| s_{j,m}^{(c)} = g_j' \varphi_m^{(c)} \circ g_j,\, g_j' > 0,\, g_j \in \mathcal{G}_m^{(c)} \right\}$$

**transform space**

$$\widehat{\mathbb{S}}^{(c)} = \bigcup_{m=1}^{M_c} \widehat{\mathbb{S}}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}},$$

$$\widehat{\mathbb{S}}_{\varphi_m^{(c)},\mathcal{G}_m^{(c)}} = \left\{ \widehat{s}_{j,m}^{(c)} \middle| \widehat{s}_{j,m}^{(c)} = g_j^{-1} \circ \widehat{\varphi}_m^{(c)},\, g_j \in \mathcal{G}_m^{(c)} \right\}$$