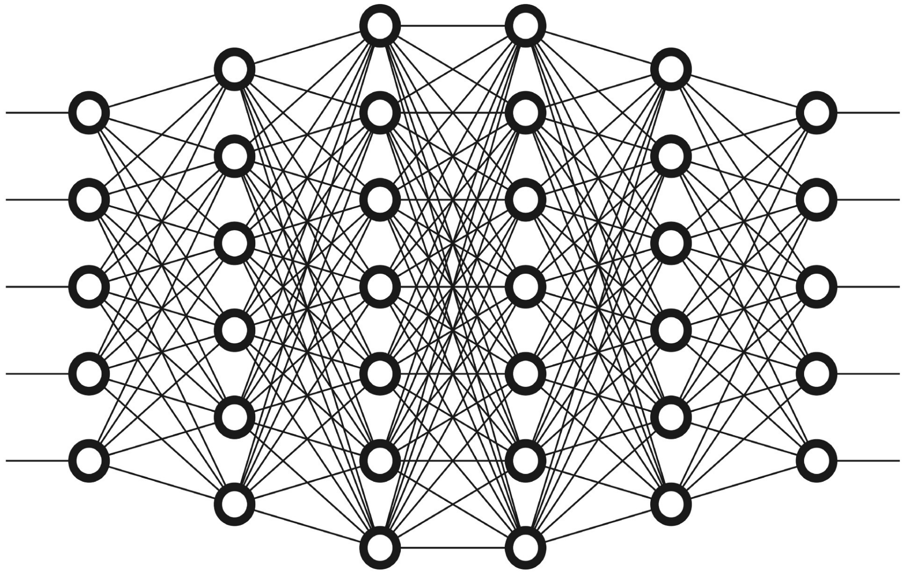
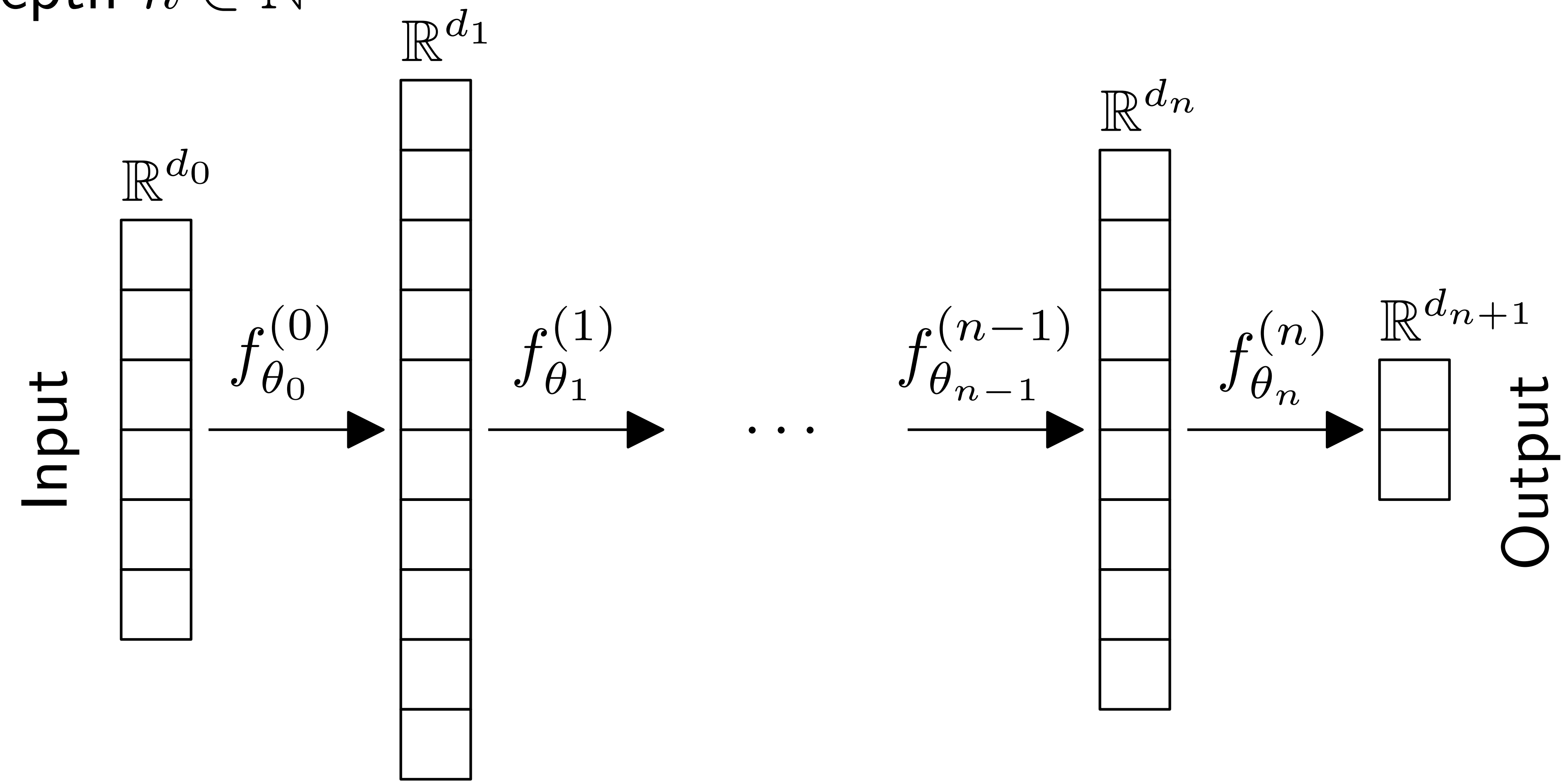
Deep Learning for Persistence Diagrams

Mathieu Carrière
19/12/2017
DataShape Seminar



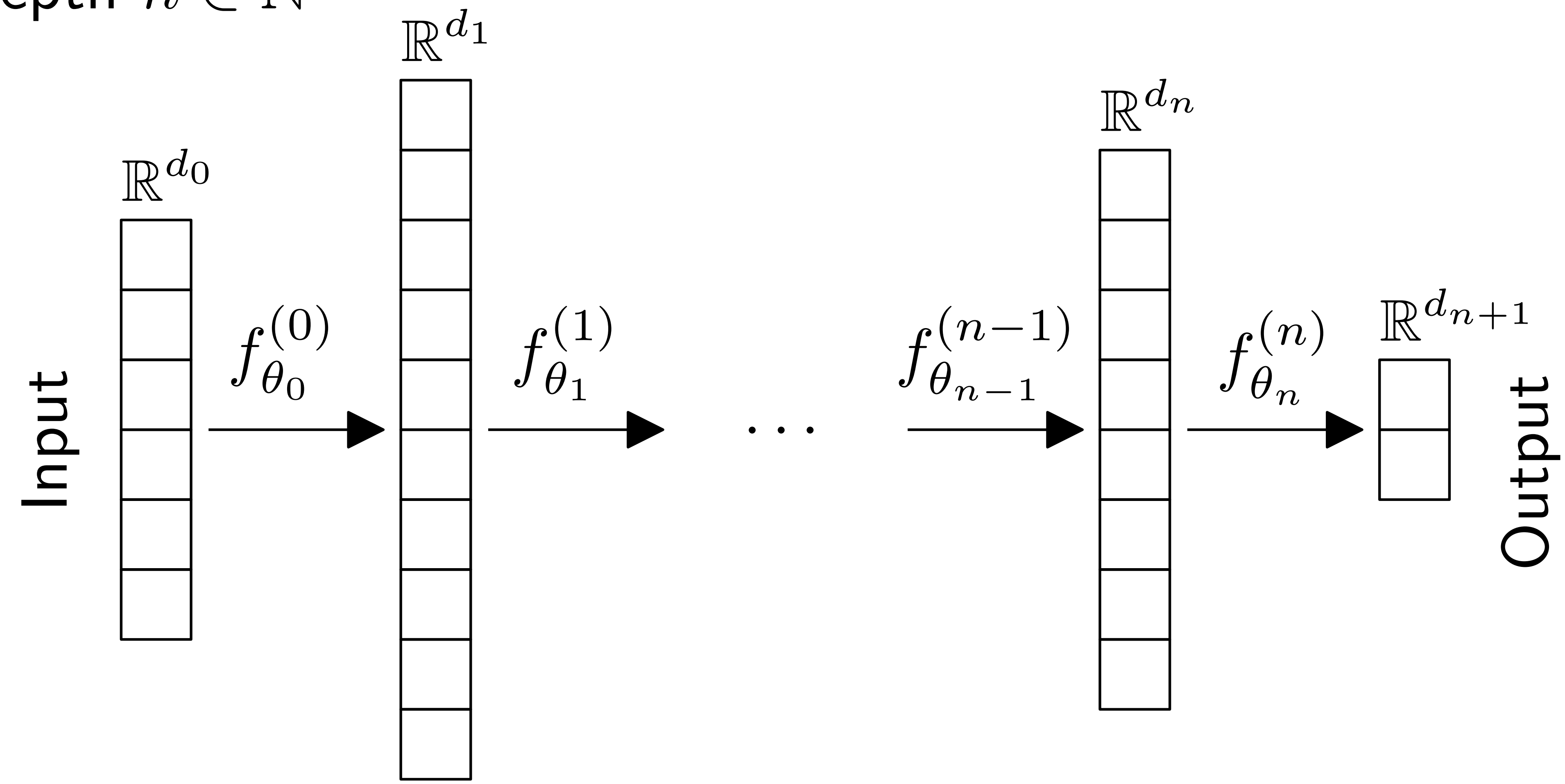
Reminder

NN with depth $n \in \mathbb{N}^*$



Reminder

NN with depth $n \in \mathbb{N}^*$



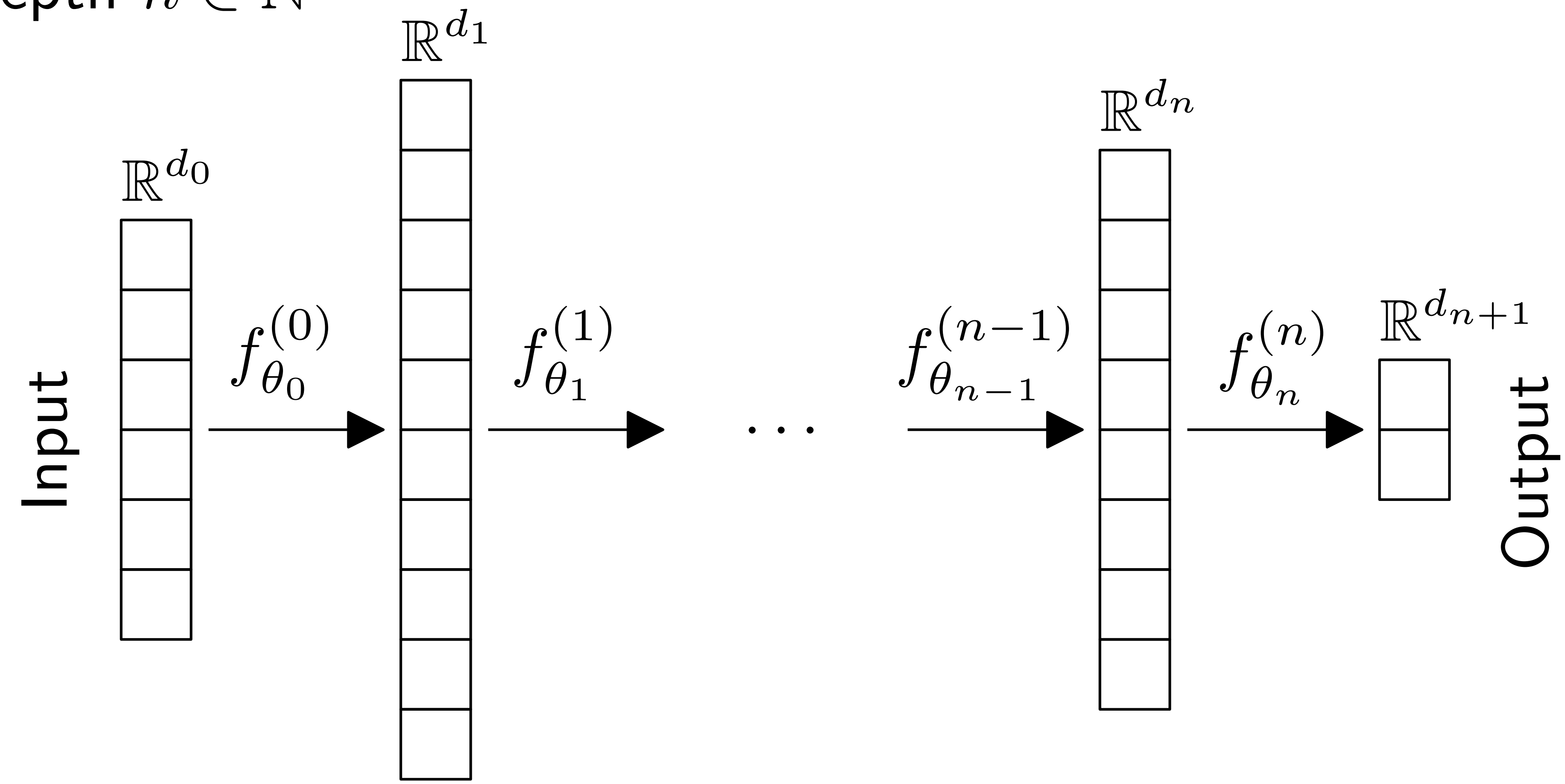
$$\theta_k = (W_k \in \mathbb{R}^{d_{k+1} \times d_k},\ b_k \in \mathbb{R}^{d_{k+1}}), \quad \sigma : x \mapsto \max(0, x) \text{ or } (1 + \mathrm{e}^{-x})^{-1}$$

$$f^{(k)}_{\theta_k} : x \in \mathbb{R}^{d_k} \mapsto \sigma(W_k \cdot x + b_k) \in \mathbb{R}^{d_{k+1}}$$

Final classifier: $f_\theta = f^{(n)}_{\theta_n} \circ \cdots \circ f^{(0)}_{\theta_0}$

Reminder

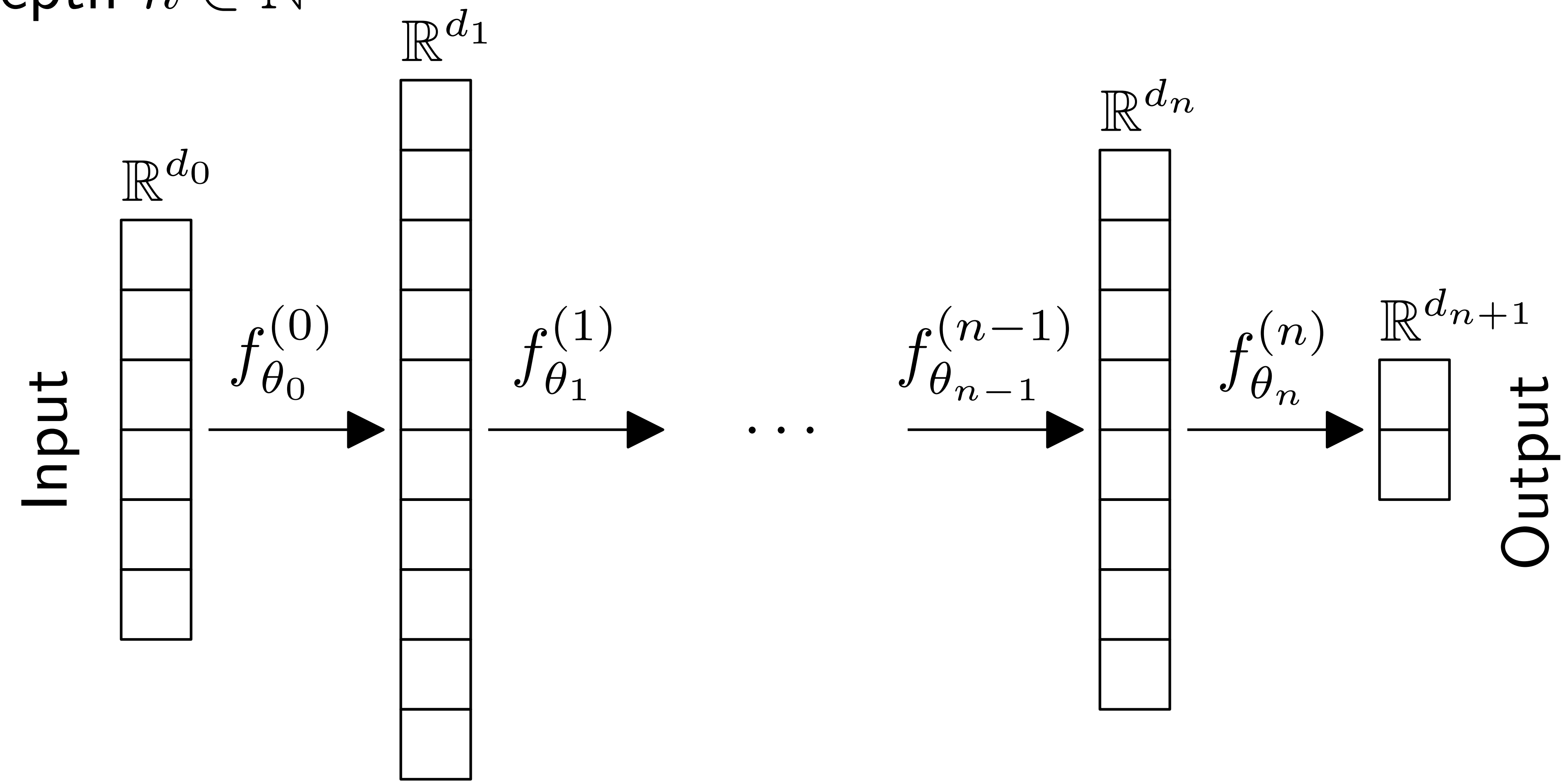
NN with depth $n \in \mathbb{N}^*$



Goal: Minimize $\ell(\theta) = \sum_i \|f_\theta(x_i) - y_i\|_2^2$ w.r.t. θ

Reminder

NN with depth $n \in \mathbb{N}^*$



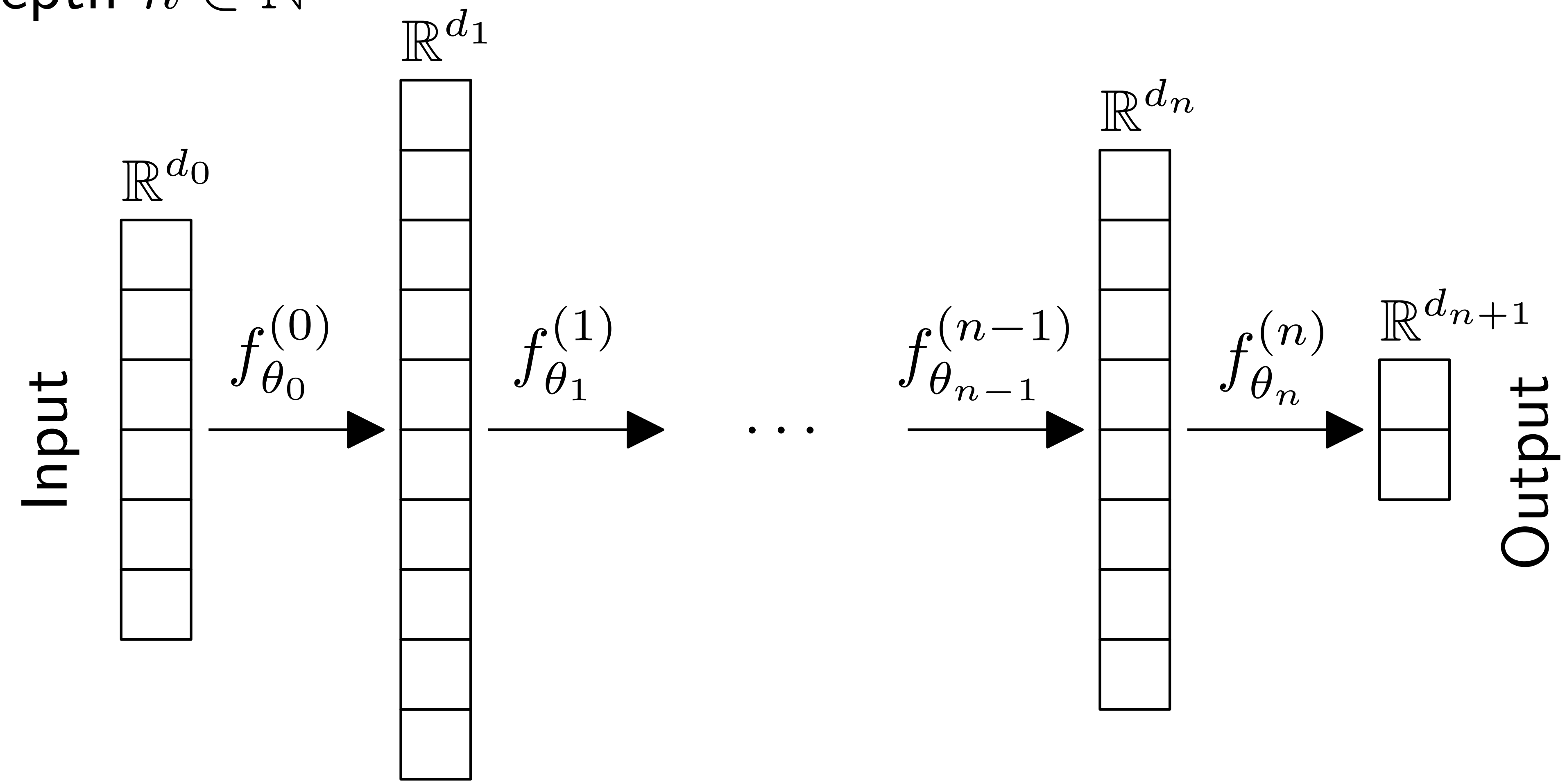
Goal: Minimize $\ell(\theta) = \sum_i \|f_\theta(x_i) - y_i\|_2^2$ w.r.t. θ

Backpropagation: for each k:

1. compute $\nabla \ell(\theta_k)$ with chain rule
2. update $\theta_k := \theta_k - \eta \nabla \ell(\theta_k)$

Reminder

NN with depth $n \in \mathbb{N}^*$



Goal: Minimize $\ell(\theta) = \sum_i \|f_\theta(x_i) - y_i\|_2^2$ w.r.t. θ

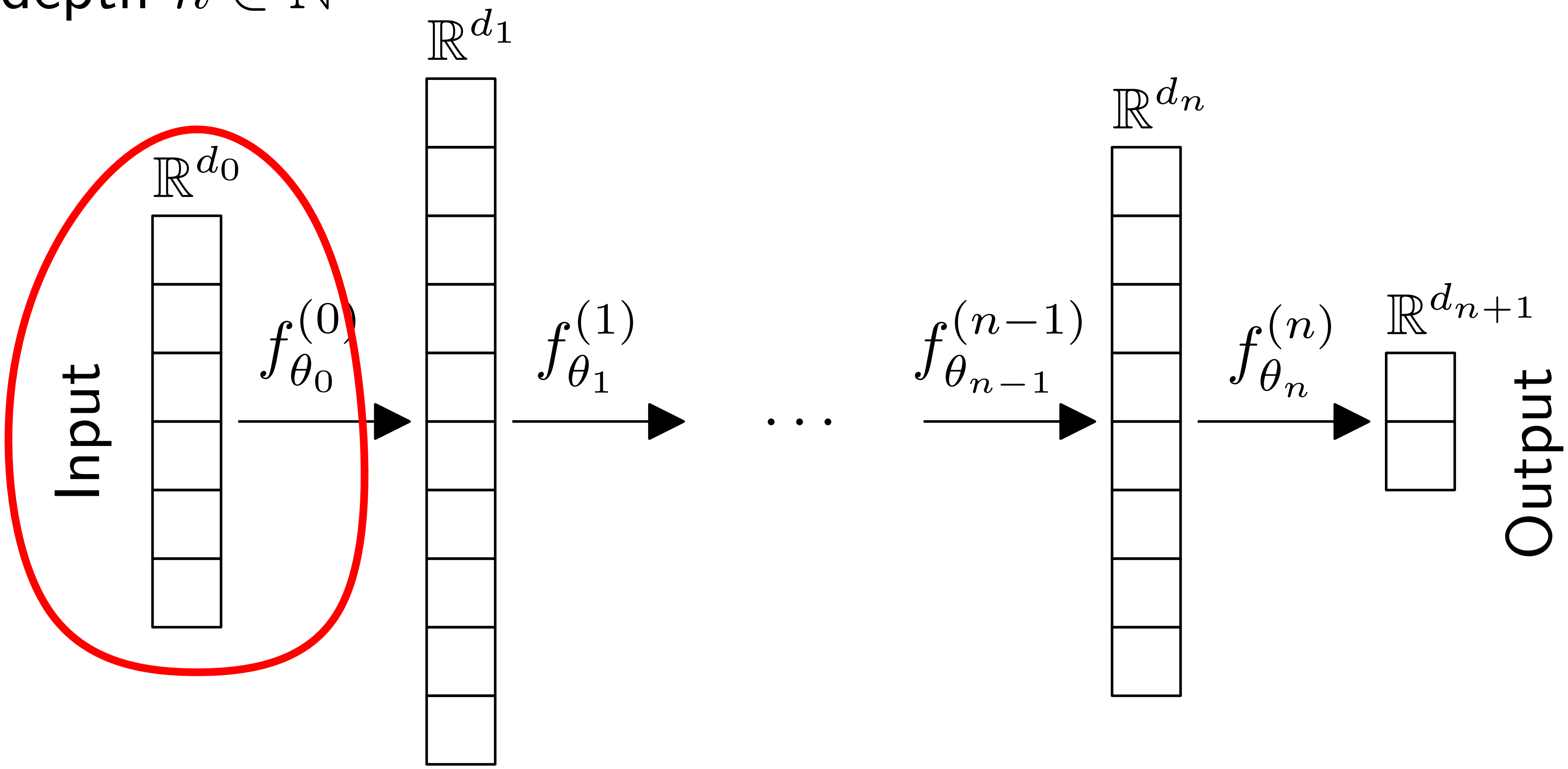
Backpropagation: for each k:

1. compute $\nabla \ell(\theta_k)$ with chain rule
2. update $\theta_k := \theta_k - \eta \nabla \ell(\theta_k)$

Requirement: $f^{(k)}_{\theta_k}$ needs to be differentiable w.r.t. θ_k and x

What about Persistence Diagrams?

NN with depth $n \in \mathbb{N}^*$



NN require vectors

Solutions

Map Persistence Diagrams to $\mathbb{R}^d$

Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Solutions

Map Persistence Diagrams to $\mathbb{R}^d$

Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Add specific layer to handle Persistence Diagrams:

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Solutions

Map Persistence Diagrams to $\mathbb{R}^d$

Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

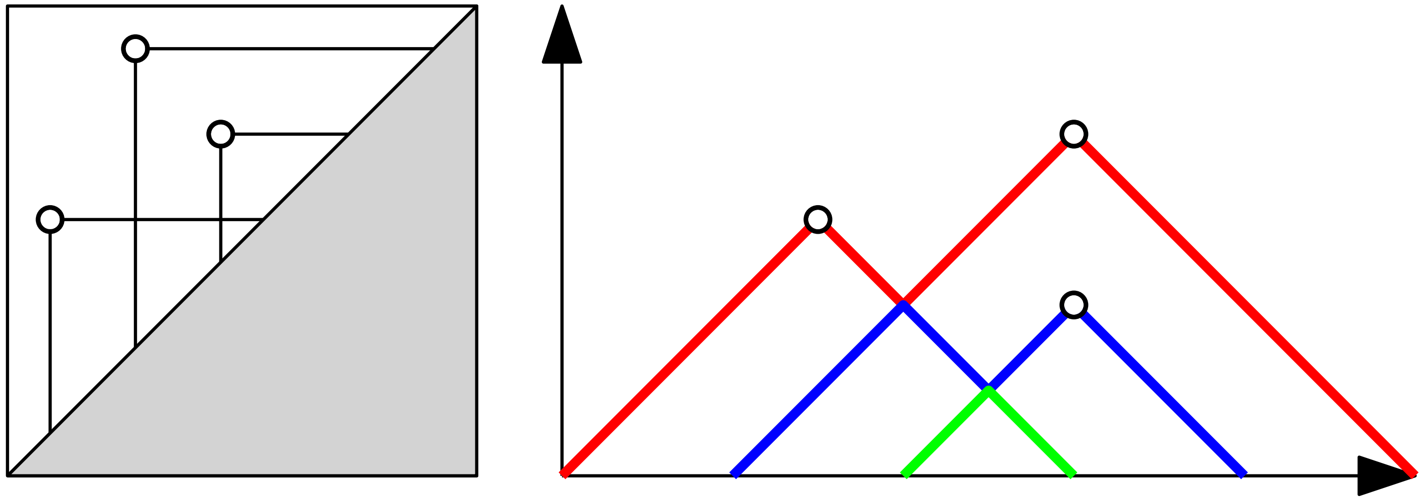
Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Add specific layer to handle Persistence Diagrams:

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

See landscape as a 1D image:

B. Beaufils et al., 2018



Solutions

Map Persistence Diagrams to $\mathbb{R}^d$

Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

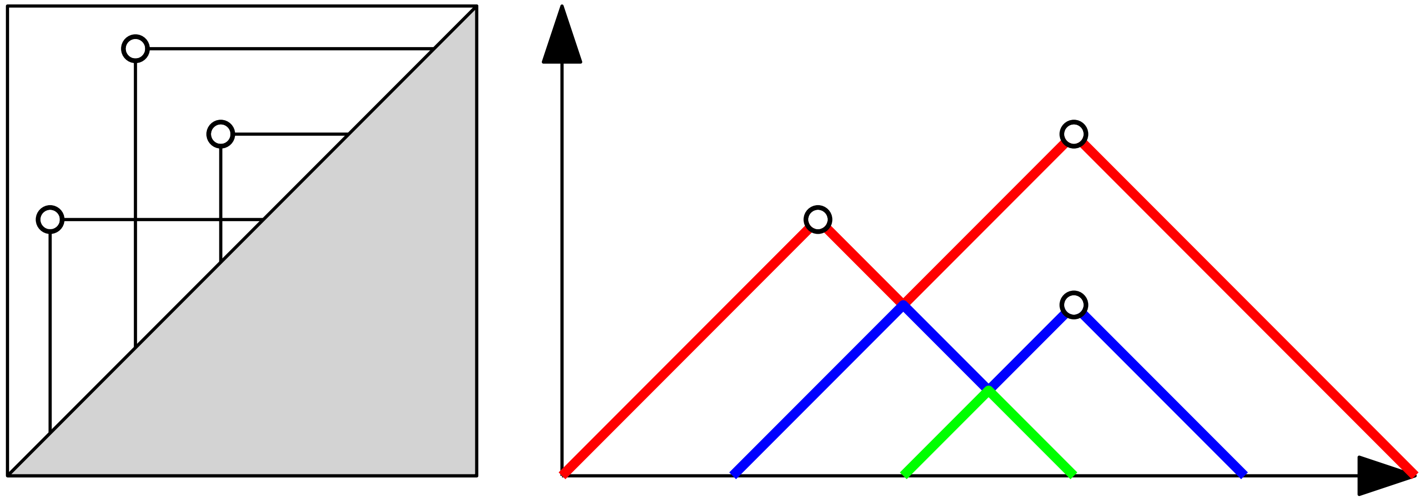
Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Add specific layer to handle Persistence Diagrams:

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

See landscape as a 1D image:

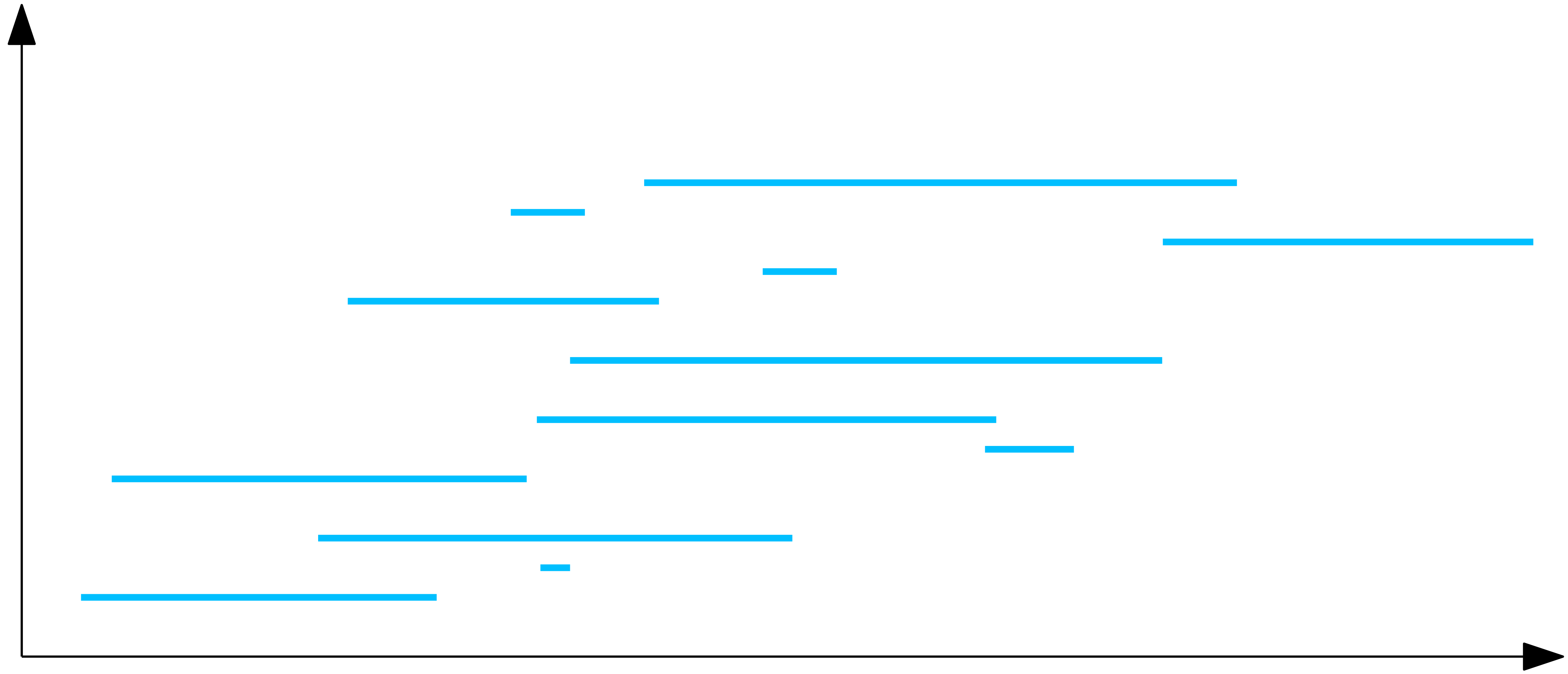
B. Beaufils et al., 2018



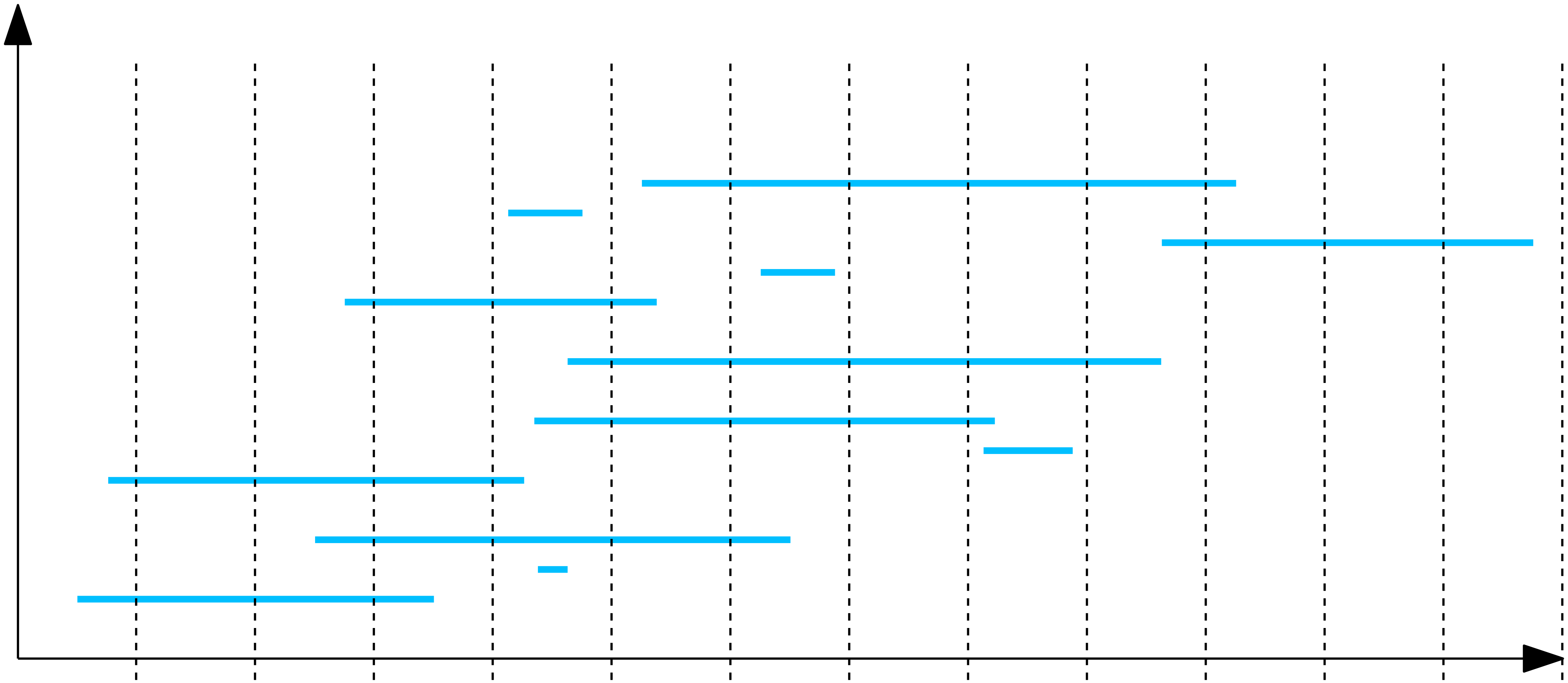
Use persistence as a new layer (?!):

J.Y. Liu et al., **Applying Topological Persistence in Convolutional Neural Network for Music Audio Signals**, *arXiv*, 2016

Mapping to $\mathbb{R}^d$



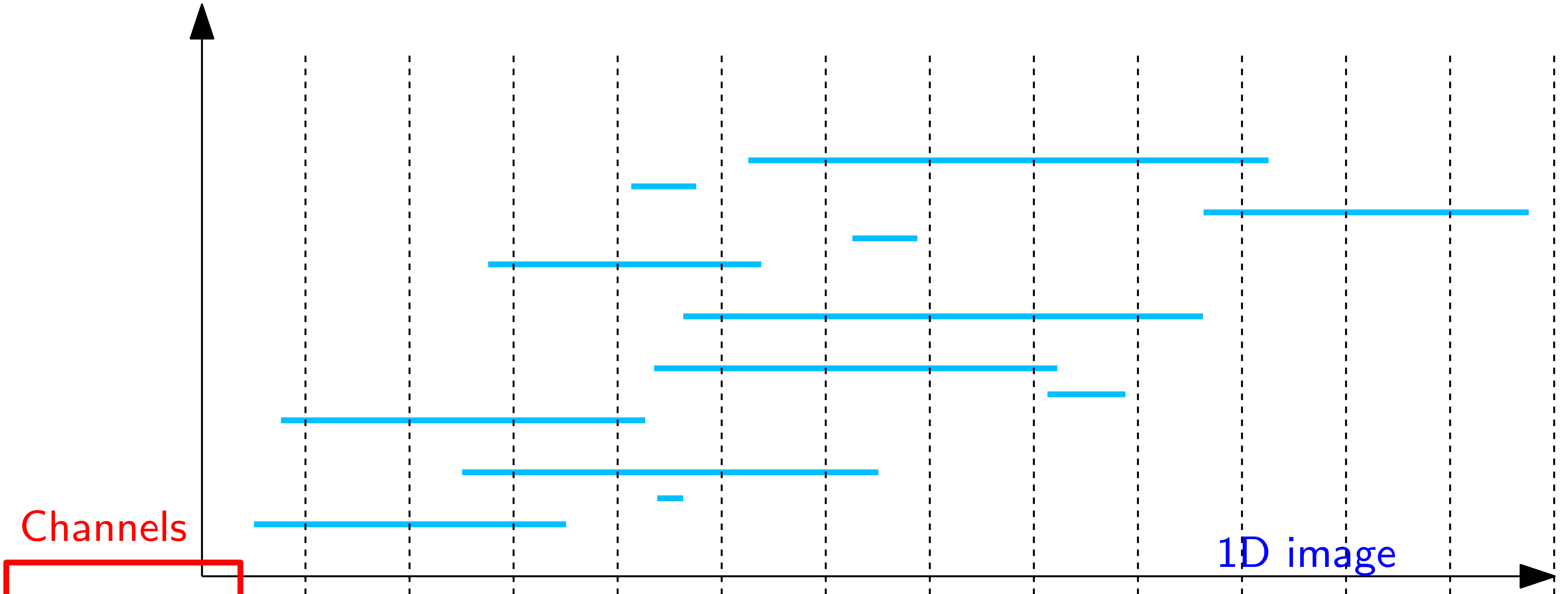
Mapping to $\mathbb{R}^d$



#Births	2	0	2	0	4	1	1	0	1	1	0	0	0
#Deaths	0	0	0	1	3	1	2	0	2	1	1	0	1
#Pairs	0	0	0	0	2	0	1	0	1	0	0	0	0

Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Mapping to $\mathbb{R}^d$



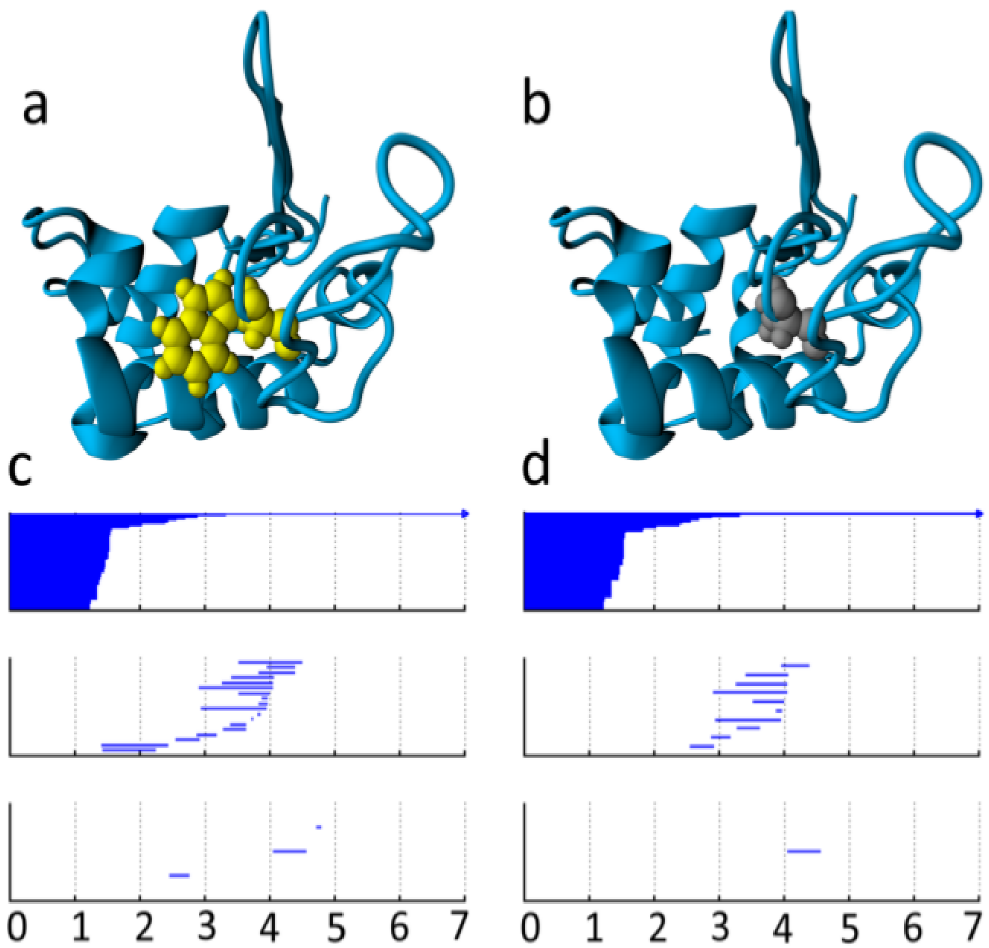
#Births	2	0	2	0	4	1	1	0	1	1	0	0	0
#Deaths	0	0	0	1	3	1	2	0	2	1	1	0	1
#Pairs	0	0	0	0	2	0	1	0	1	0	0	0	0

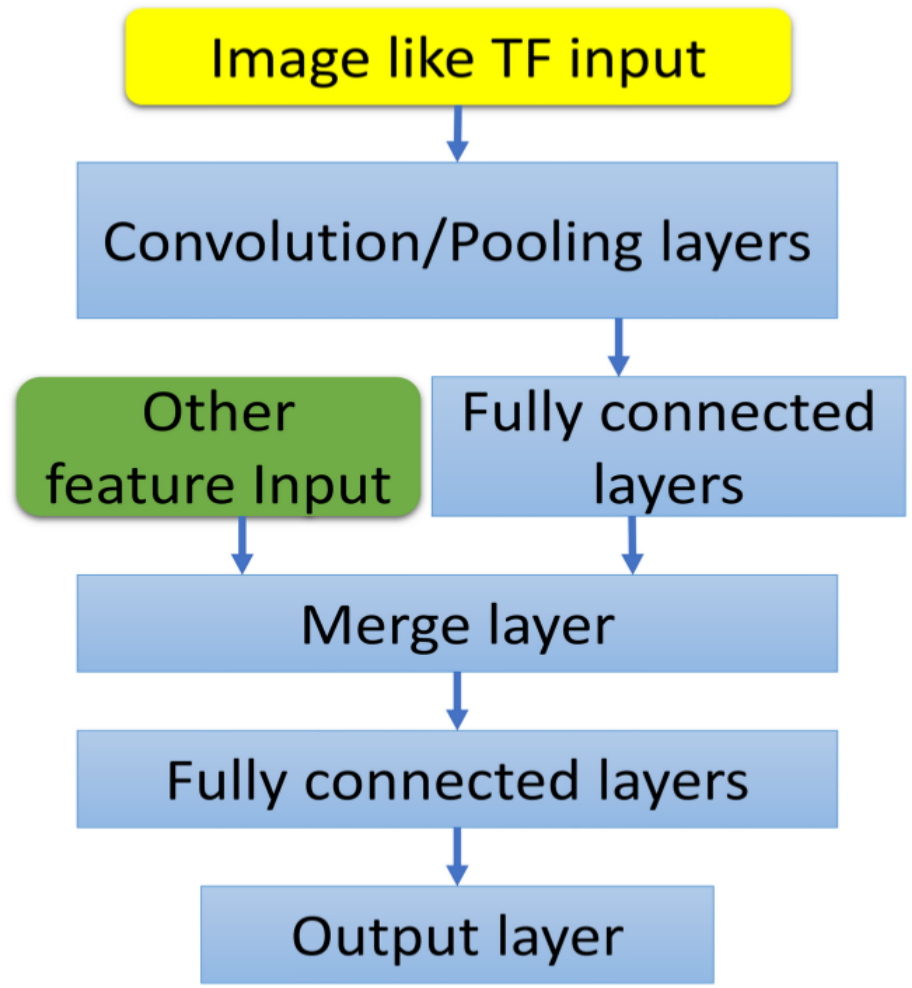
Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Mapping to $\mathbb{R}^d$

Protein Classification

Rips PD $+$ CNN





Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Mapping to $\mathbb{R}^d$

Table 2. Performance comparisons of TNet-MP and other methods.

Method	S350			S2648		
	n^d	R_P	RMSE	n^d	R_P	RMSE
TNet-MP-2	350	0.81	0.94	2648	0.77	0.94
STRUM[b]	350	0.79	0.98	2647	0.77	0.94
TNet-MP-1	350	0.74	1.07	2648	0.72	1.02
mCSM[b,c]	350	0.73	1.08	2643	0.69	1.07
INPS[b,c]	350	0.68	1.25	2648	0.56	1.26
PoPMuSiC 2.0[b]	350	0.67	1.16	2647	0.61	1.17
PoPMuSiC 1.0[a]	350	0.62	1.23	-	-	-
I-Mutant 3.0[b]	338	0.53	1.35	2636	0.60	1.19
Dmutant[a]	350	0.48	1.38	-	-	-
Automute[a]	315	0.46	1.42	-	-	-
CUPSAT[a]	346	0.37	1.46	-	-	-
Eris[a]	334	0.35	1.49	-	-	-
I-Mutant 2.0[a]	346	0.29	1.50	-	-	-

Comparison of Pearson correlation coefficients (R_P) and RMSEs (kcal/mol) of various methods on the prediction task of the "S350" set and 5-fold cross validation of the "S2648". TNet-MP-1 is our multichannel topological convolutional neural network model that solely utilizes topological information. TNet-MP-2 is our model that complements TNet-MP-1 with auxiliary features.

[a] Data directly obtained from Worth *et al* [89].

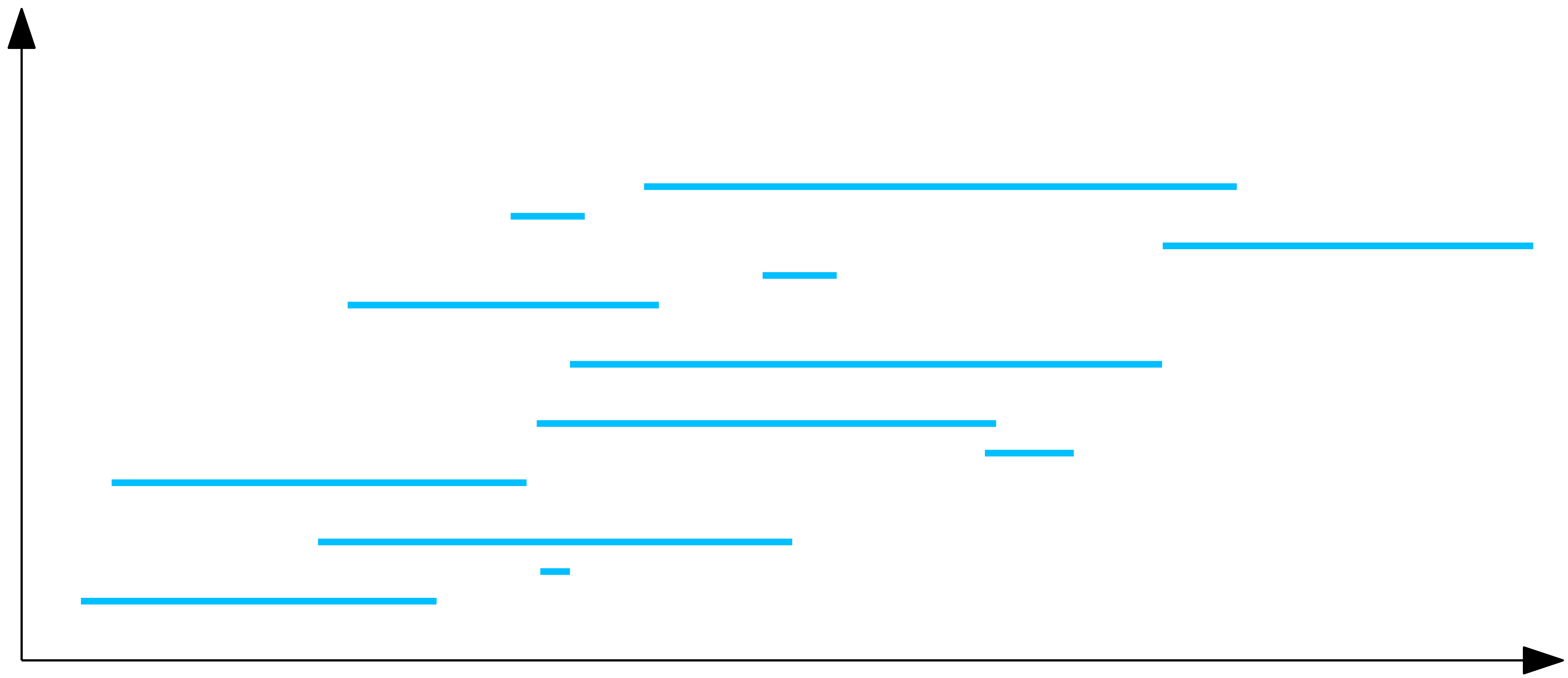
[b] Data obtained from Quan *et al* [94].

[c] The results reported in the publications are listed in the table. According to Ref. [94], the data from the online server has R_p (RMSE) of 0.59 (1.28) and 0.70 (1.13) for INPS and mCSM respectively in the task of S350 set.

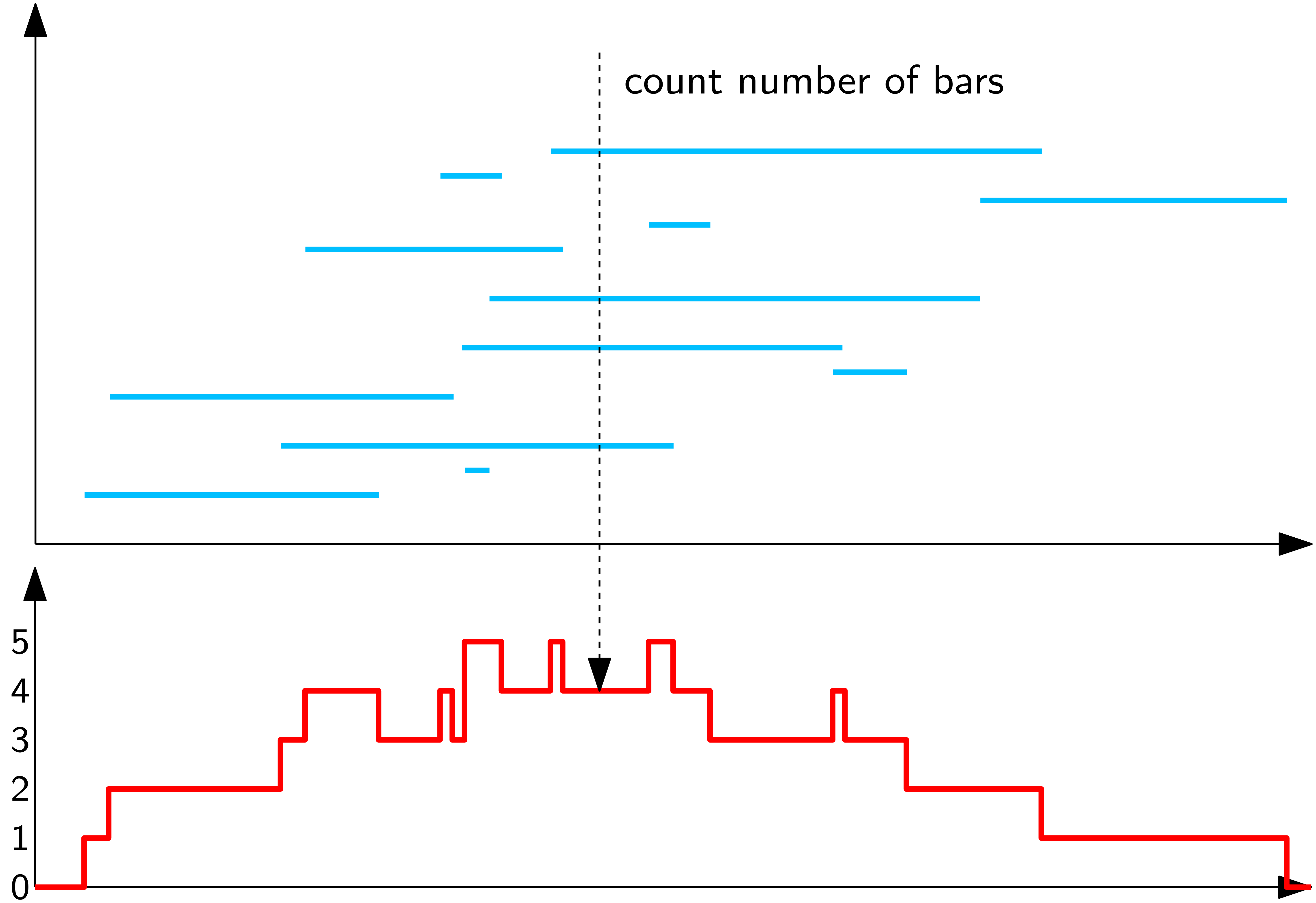
[d] Number of samples successfully processed.

Z. Cang, G.W. Wei, **TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions**, *PLOS*, 2017

Mapping to $\mathbb{R}^d$



Mapping to $\mathbb{R}^d$

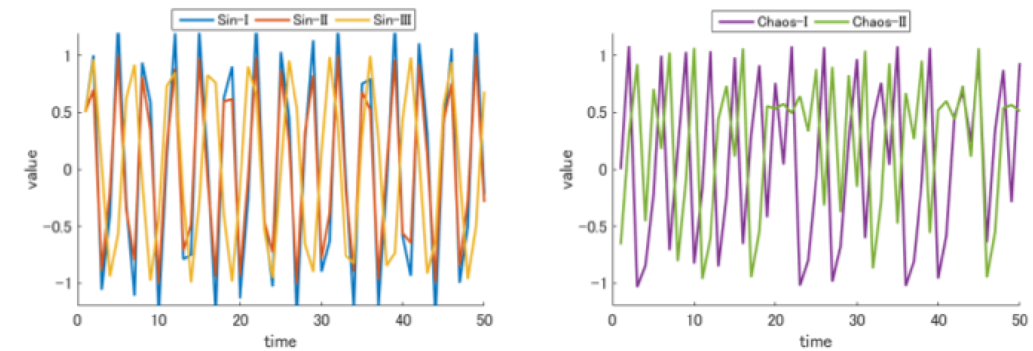


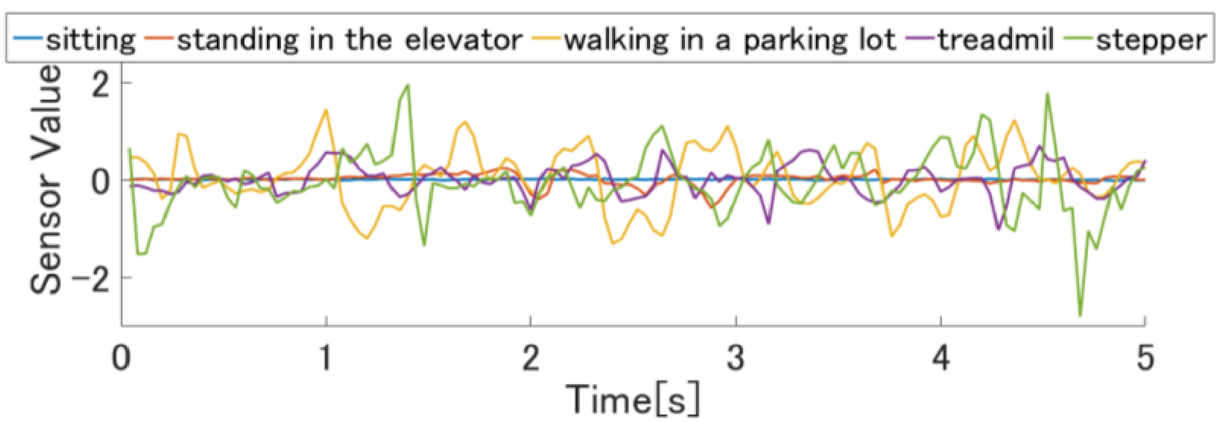
Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

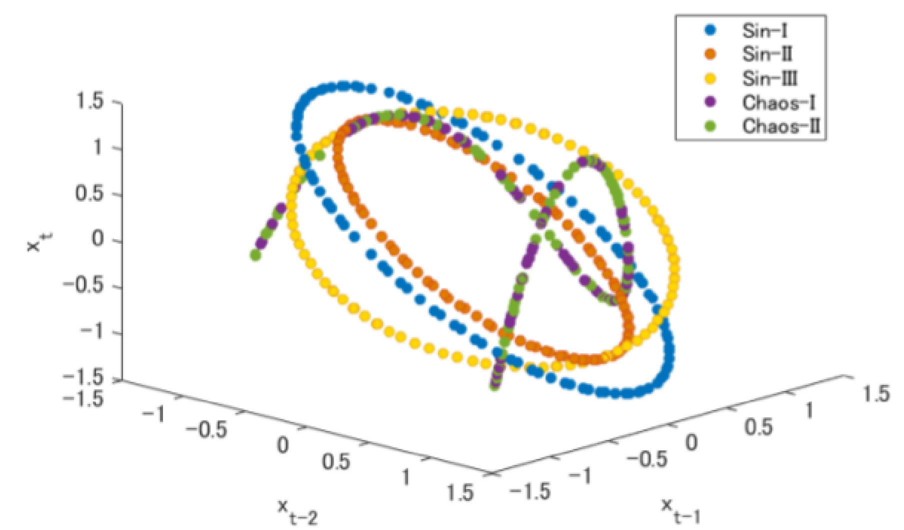
Mapping to $\mathbb{R}^d$

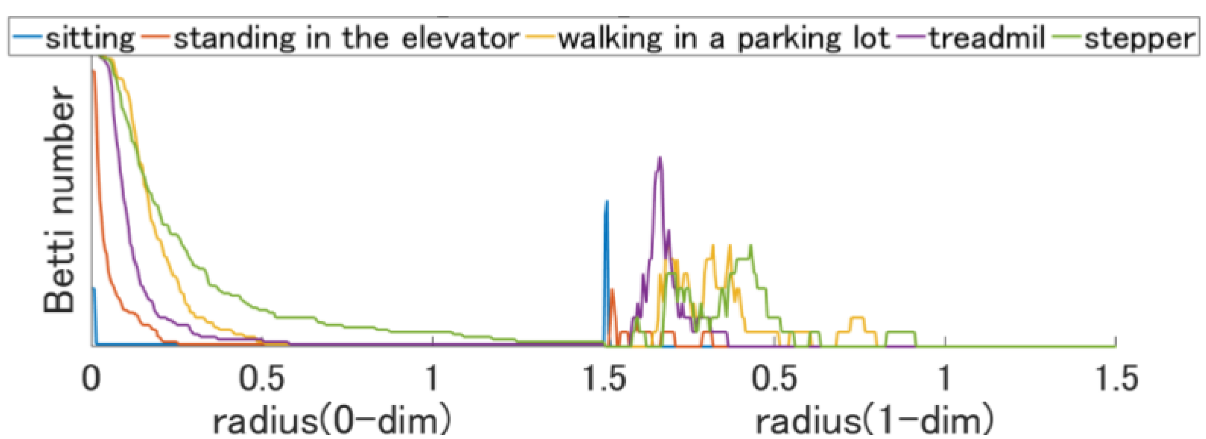
Time Series Classification

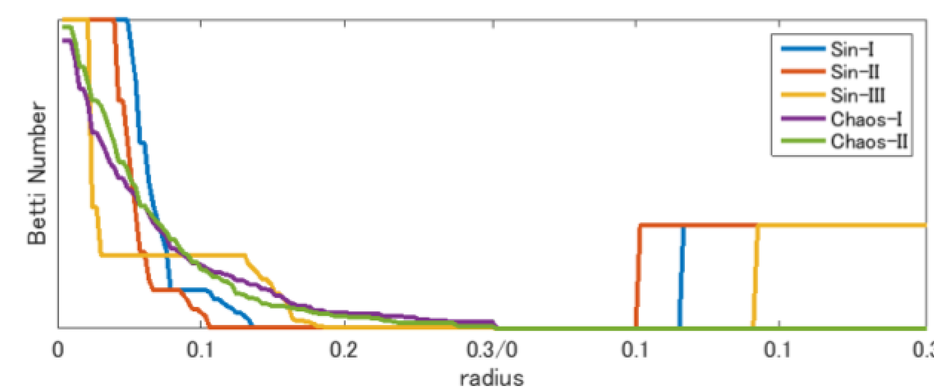
Rips PD of delay embedding + CNN











Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

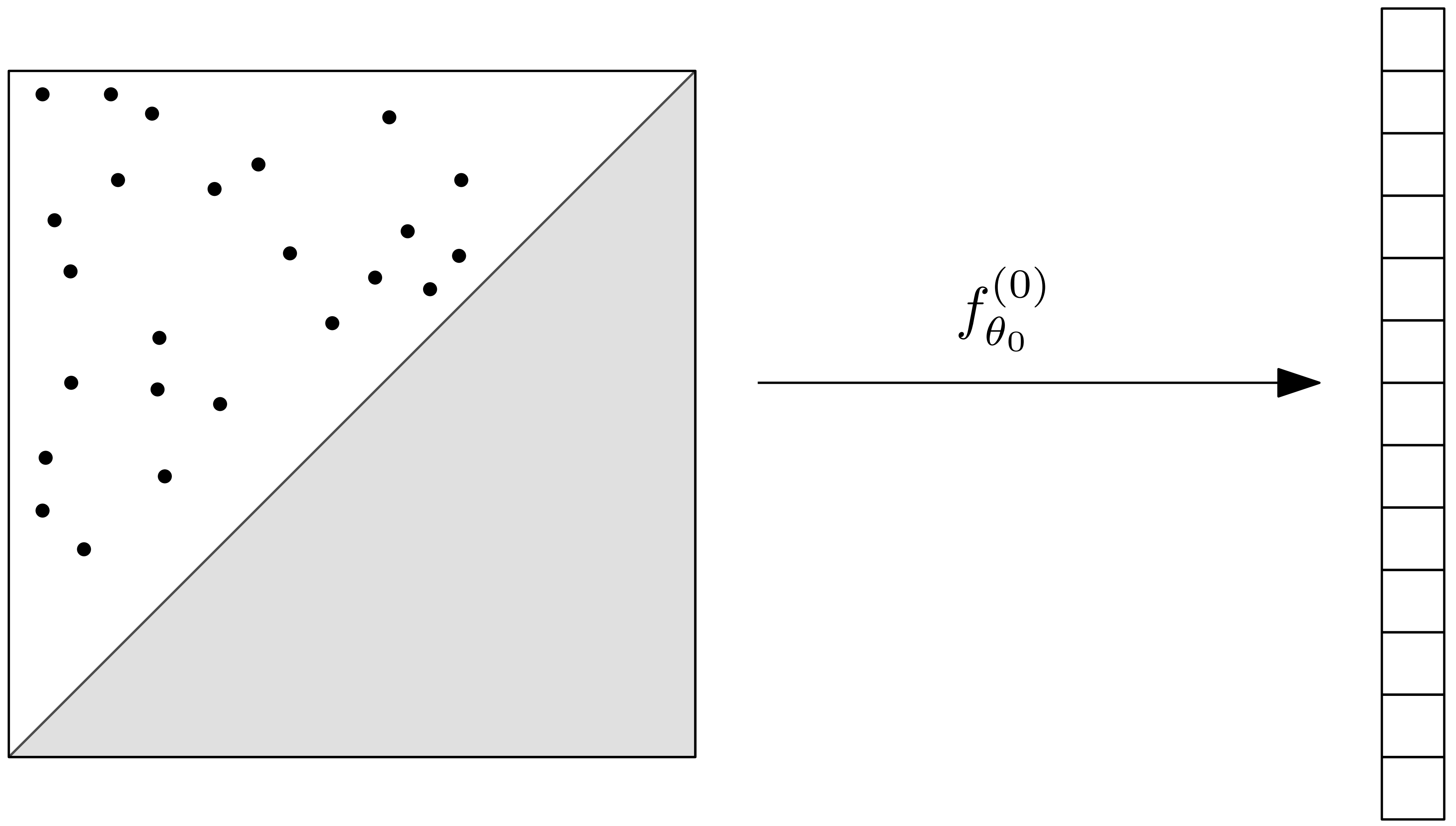
Mapping to $\mathbb{R}^d$

Time Series Classification

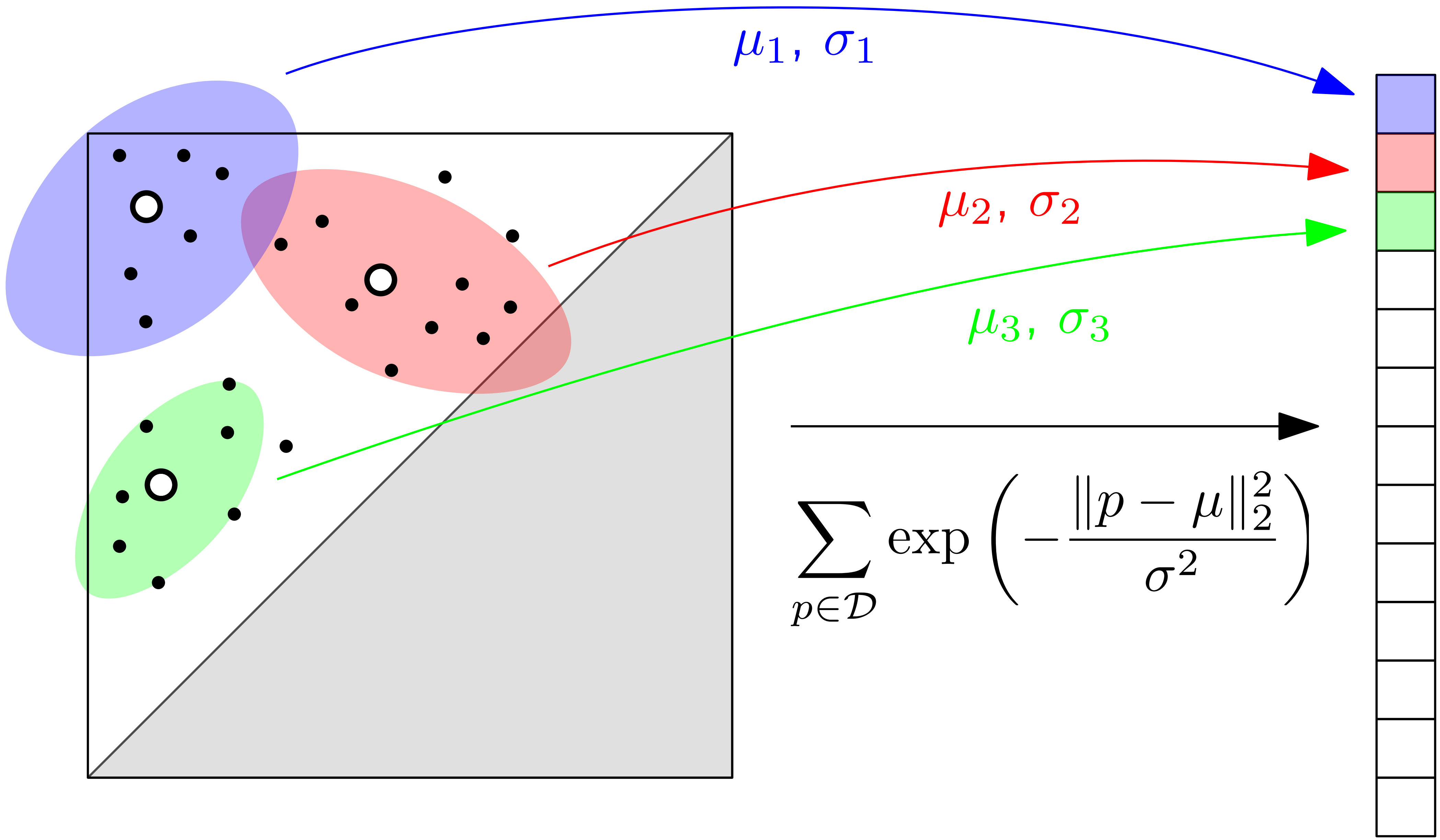
Datasets	Gyro sensor	EEG dataset	EMG dataset
	Accuracy		
method\validation	Leave one subject out [%]	10-fold[%]	Leave one subject out[%]
SVM + statistical feature	67.6 ± 4.7	44.4 ± 19.8	15.0 ± 10.0
SVM+Chaos feature	53.3 ± 7.1	55.2 ± 9.6	41.5 ± 25.9
DTW + 1-NN	6.4 ± 5.1	72.4 ± 6.1	15.0 ± 10.0
imaging CNN	18.9 ± 5.2	48.9 ± 4.2	10.0 ± 0
SVM+Betti sequence	63.5 ± 11.3	66.7 ± 5.6	49.6 ± 18.2
connected input 1-CNN+Betti sequence	79.8 ± 5.0	75.38 ± 5.7	74.4 ± 10.6
parallel 1-CNN+Betti sequence	86.1 ± 7.2	-	76.4 ± 7.2

Y. Umeda, **Time Series Classification via Topological Data Analysis**, *Transactions of the Japanese Society for Artificial Intelligence*, 2017

Specific Layer

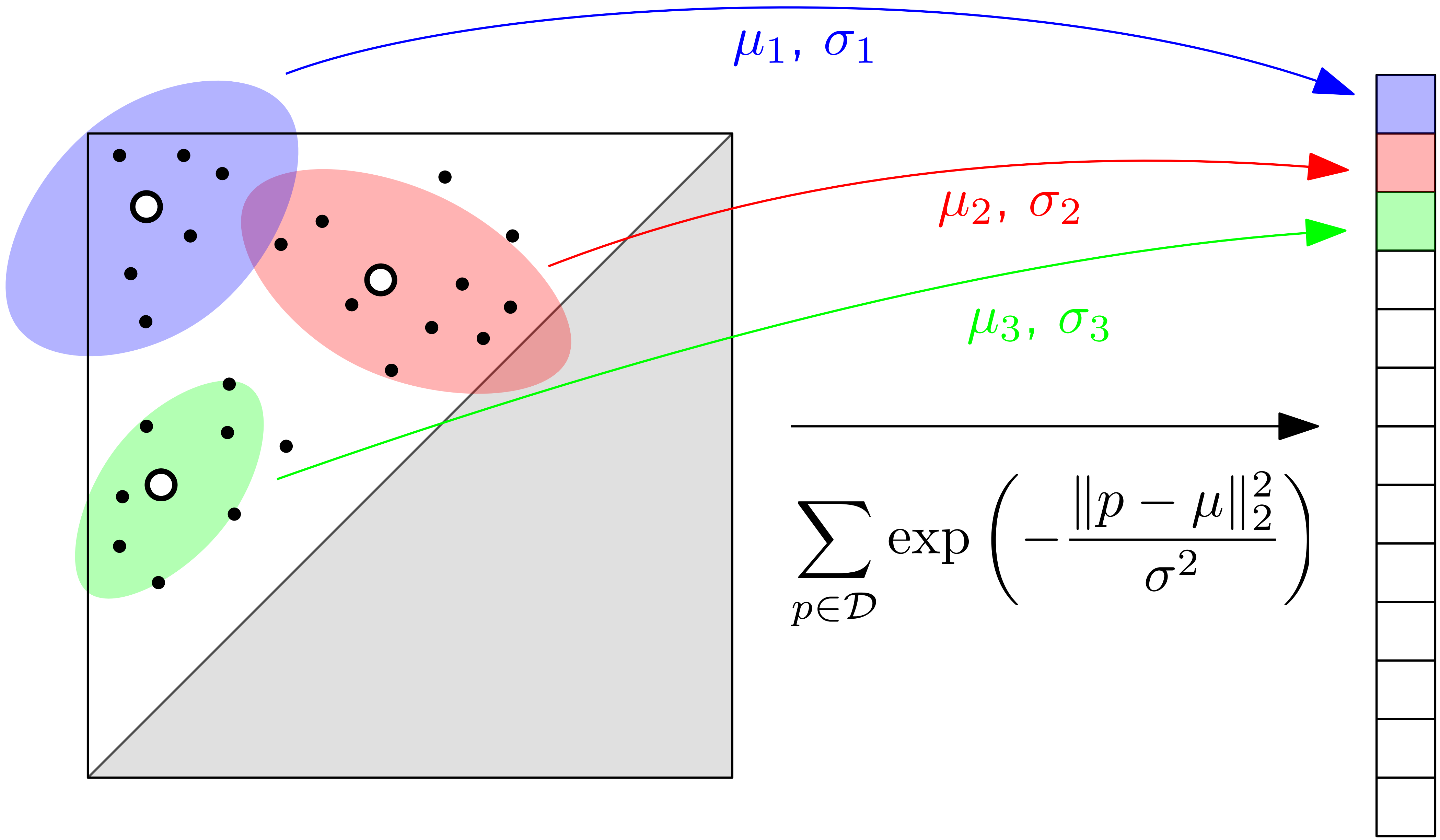


Specific Layer



C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

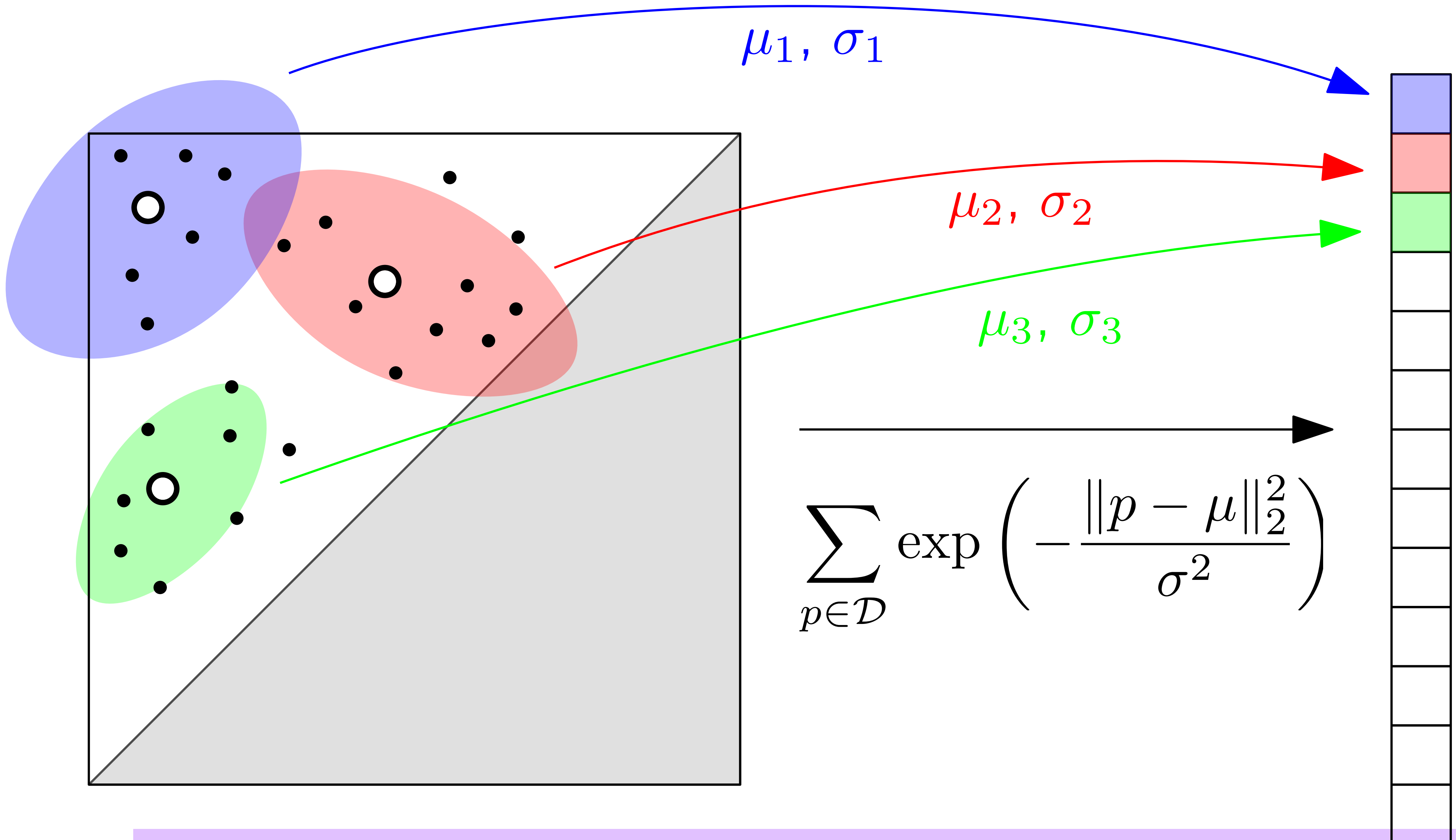
Specific Layer



$(\mu, \sigma) \mapsto s_{\mu,\sigma}(\mathcal{D}) = \sum_{p\in\mathcal{D}} \exp\left(-\frac{\|p-\mu\|_2^2}{\sigma^2}\right)$ is differentiable

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Specific Layer

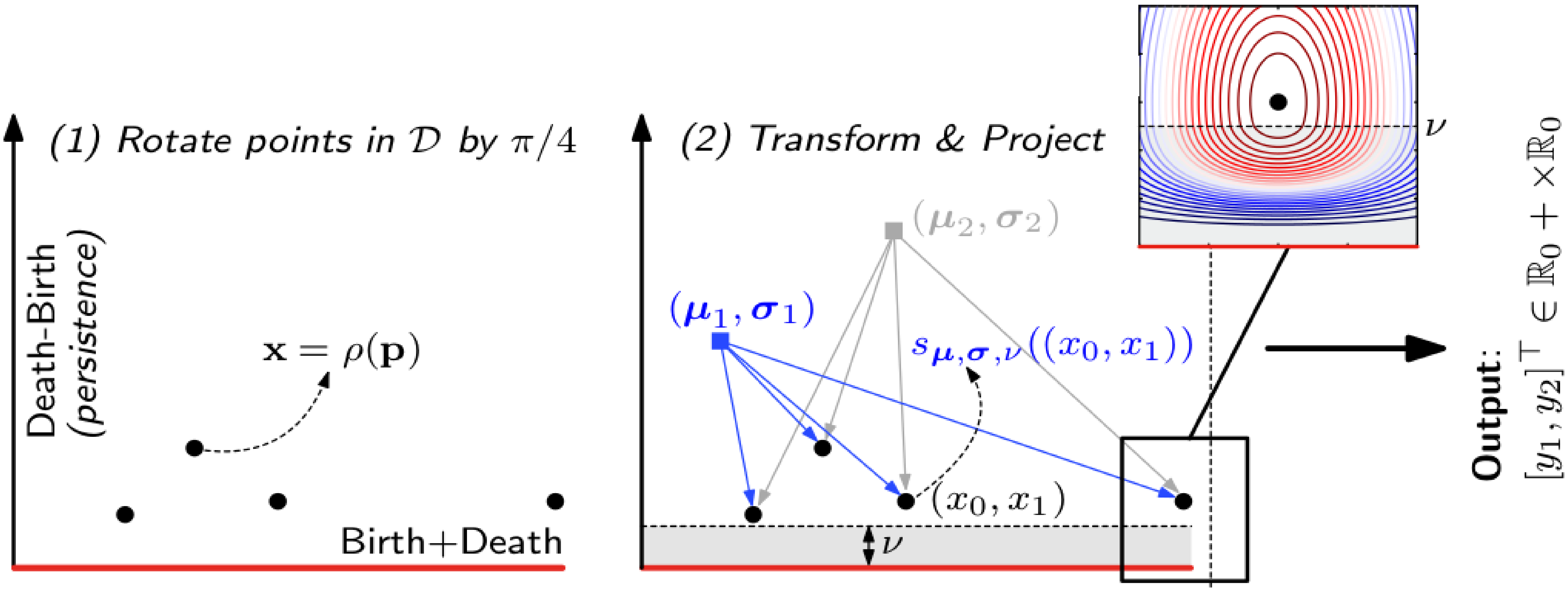


Thm: (Stability)

$$\left|s_{\mu,\sigma}(\mathcal{D}) - s_{\mu,\sigma}(\mathcal{D}')\right| \leq C_{\mu,\sigma}\, d_1(\mathcal{D}, \mathcal{D}')^{q_{\mu,\sigma}}$$

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Specific Layer



$$s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}\big((x_0, x_1)\big) = \begin{cases} e^{-\sigma_0^2(x_0-\mu_0)^2 - \sigma_1^2(x_1-\mu_1)^2} & x_1 \in [\nu, \infty) \\ e^{-\sigma_0^2(x_0-\mu_0)^2 - \sigma_1^2(\ln(\frac{x_1}{\nu})+\nu-\mu_1)^2} & x_1 \in (0, \nu) \\ 0 & x_1 = 0 \end{cases}$$

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Specific Layer

$$s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}\big((x_0,x_1)\big) = \begin{cases} e^{-\sigma_0^2(x_0-\mu_0)^2-\sigma_1^2(x_1-\mu_1)^2} & x_1 \in [\nu,\infty) \\ e^{-\sigma_0^2(x_0-\mu_0)^2-\sigma_1^2(\ln(\frac{x_1}{\nu})+\nu-\mu_1)^2} & x_1 \in (0,\nu) \\ 0 & x_1 = 0 \end{cases}$$

Remark. *Note that $s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}$ is continuous in x_1 as*

$$\lim_{x\to\nu} x = \lim_{x\to\nu} \ln\left(\frac{x}{\nu}\right) + \nu \quad \textit{and} \quad \lim_{x_1\to 0} s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}\big((x_0,x_1)\big) = 0 = s_{\mu,\sigma,\nu}\big((x_0,0)\big) \ ,$$

and $s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}$ is differentiable on $\mathbb{R}\times\mathbb{R}^+$, since

$$1 = \lim_{x\to\nu^+} \frac{\partial x_1}{\partial x_1}(x) \quad \textit{and} \quad \lim_{x\to\nu^-} \frac{\partial\left(\ln\left(\frac{x_1}{\nu}\right)+\nu\right)}{\partial x_1}(x) = \lim_{x\to\nu^-}\frac{\nu}{x} = 1 \ .$$

C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Specific Layer

$$s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}\big((x_0,x_1)\big) = \begin{cases} e^{-\sigma_0^2(x_0-\mu_0)^2-\sigma_1^2(x_1-\mu_1)^2} & x_1 \in [\nu,\infty) \\ e^{-\sigma_0^2(x_0-\mu_0)^2-\sigma_1^2(\ln(\frac{x_1}{\nu})+\nu-\mu_1)^2} & x_1 \in (0,\nu) \\ 0 & \end{cases}$$

Remark. *Note that $s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}$ is continuous in x_1 as*

$$\lim_{x\to\nu} x = \lim_{x\to\nu} \ln\left(\frac{x}{\nu}\right)+\nu \quad \textit{and} \quad \lim_{x_1\to 0} s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}\big((x_0,x_1)\big) = \quad ,$$

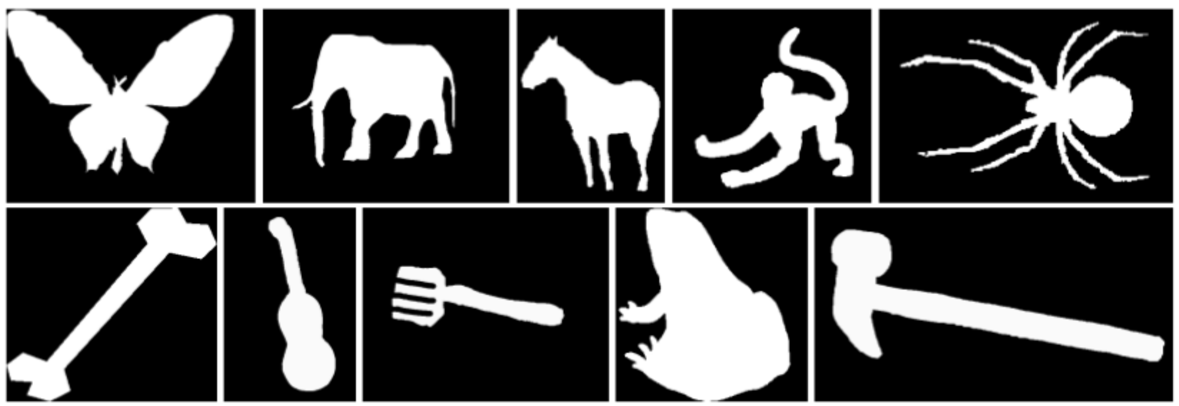
and $s_{\boldsymbol{\mu},\boldsymbol{\sigma},\nu}$ is differentiable on $\mathbb{R}\times\mathbb{R}^+$, since

$$1 = \lim_{x\to\nu^+} \frac{\partial x_1}{\partial x_1}(x) \quad \textit{and} \quad \lim_{x\to\nu^-} \frac{\partial\left(\ln\left(\frac{x_1}{\nu}\right)+\nu\right)}{\partial x_1}(x) = \lim_{x\to\nu^-} \frac{\nu}{x} = 1 \ .$$

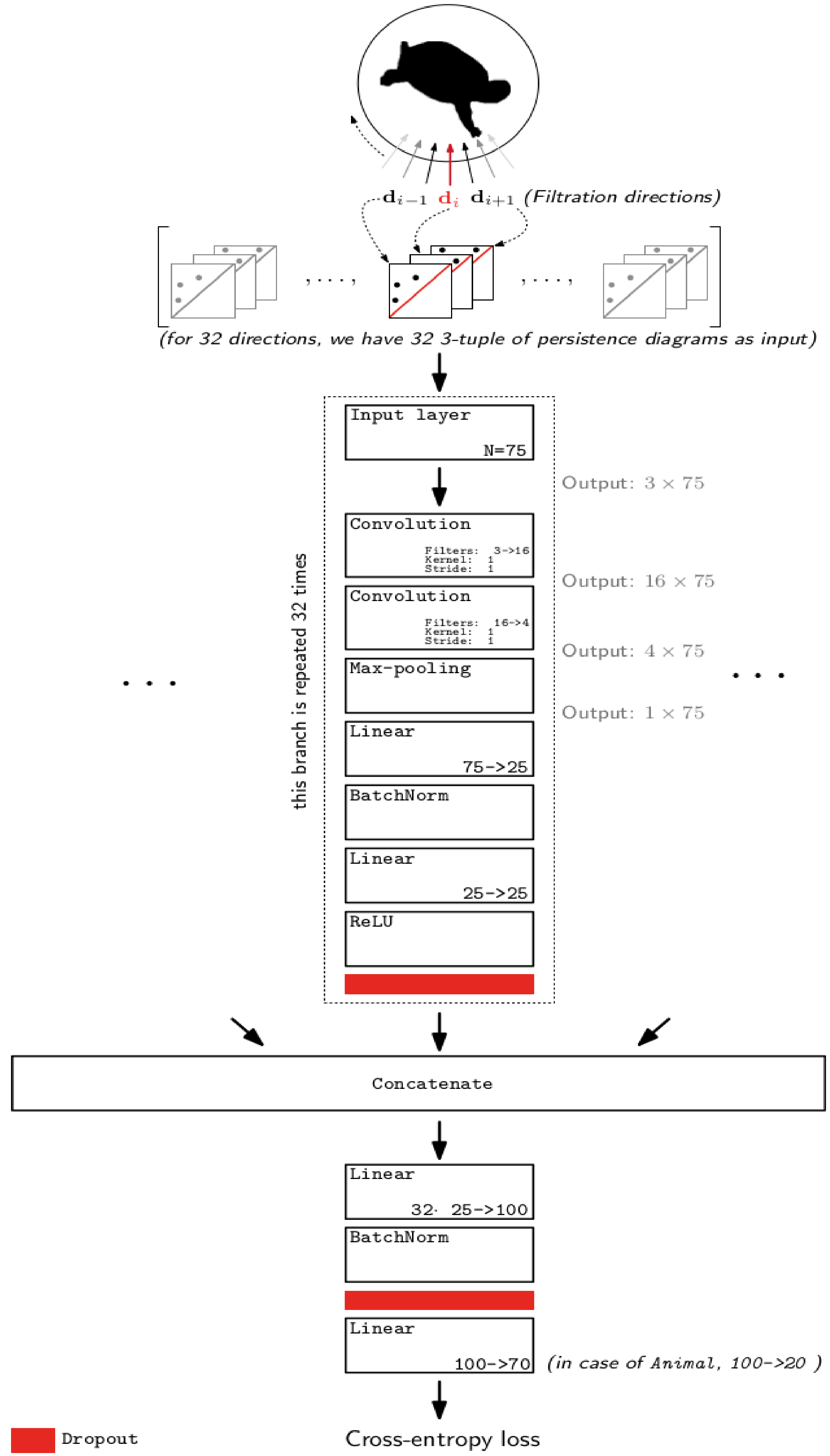
C. Hofer et al., **Deep Learning with Topological Signatures**, *NIPS*, 2017

Specific Layer

2D Image Classification

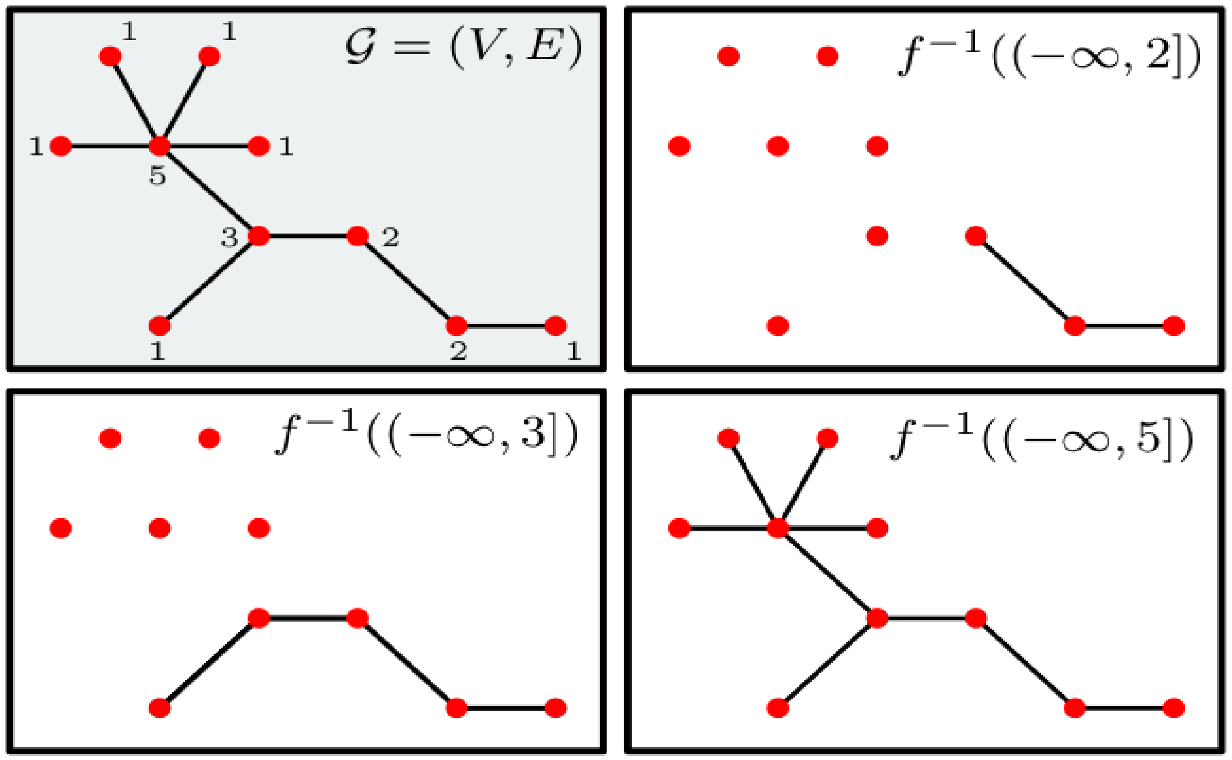


	MPEG-7	Animal
‡Skeleton paths	86.7	67.9
‡Class segment sets	90.9	69.7
†ICS	96.6	78.4
†BCF	97.2	83.4
Ours	91.8	69.5

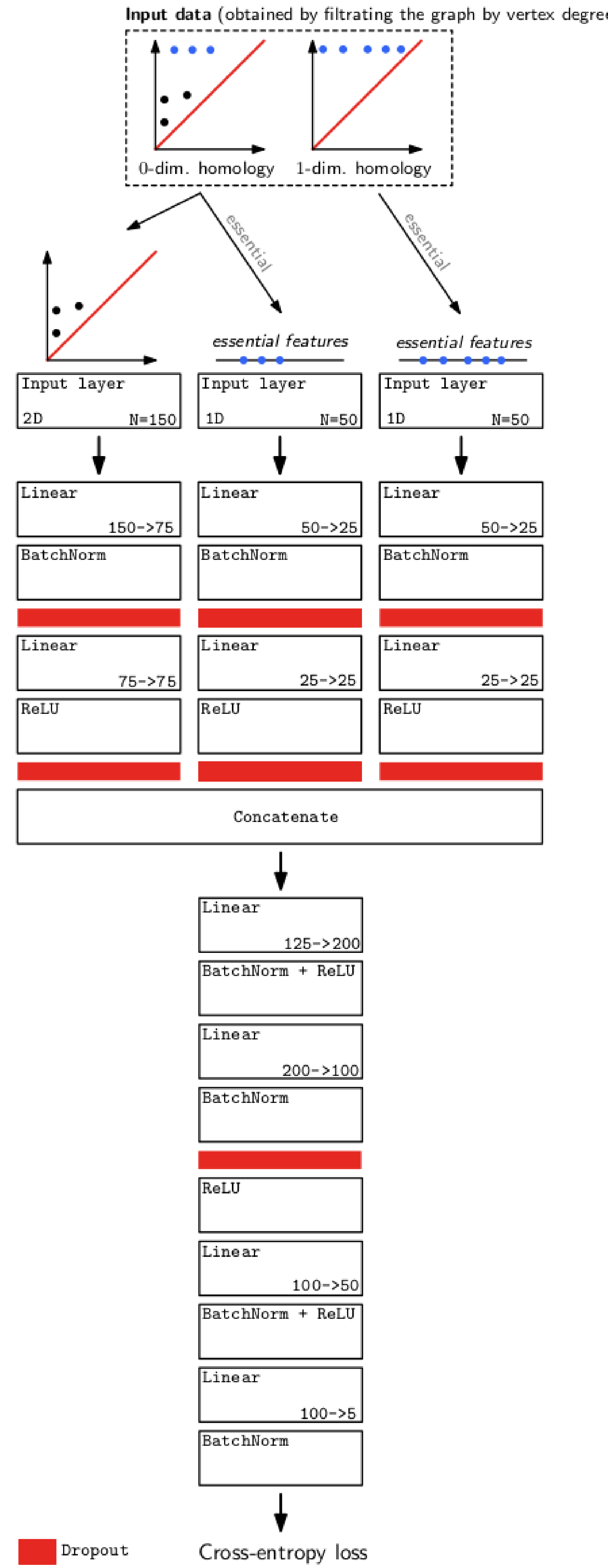


Specific Layer

Social Network Classification



	reddit-5k	reddit-12k
GK [29]	41.0	31.8
DGK [29]	41.3	32.2
PSCN [22]	49.1	41.3
RF [4]	50.9	42.7
Ours (w/o essential)	49.1	38.5
Ours (w/ essential)	**54.5**	**46.1**

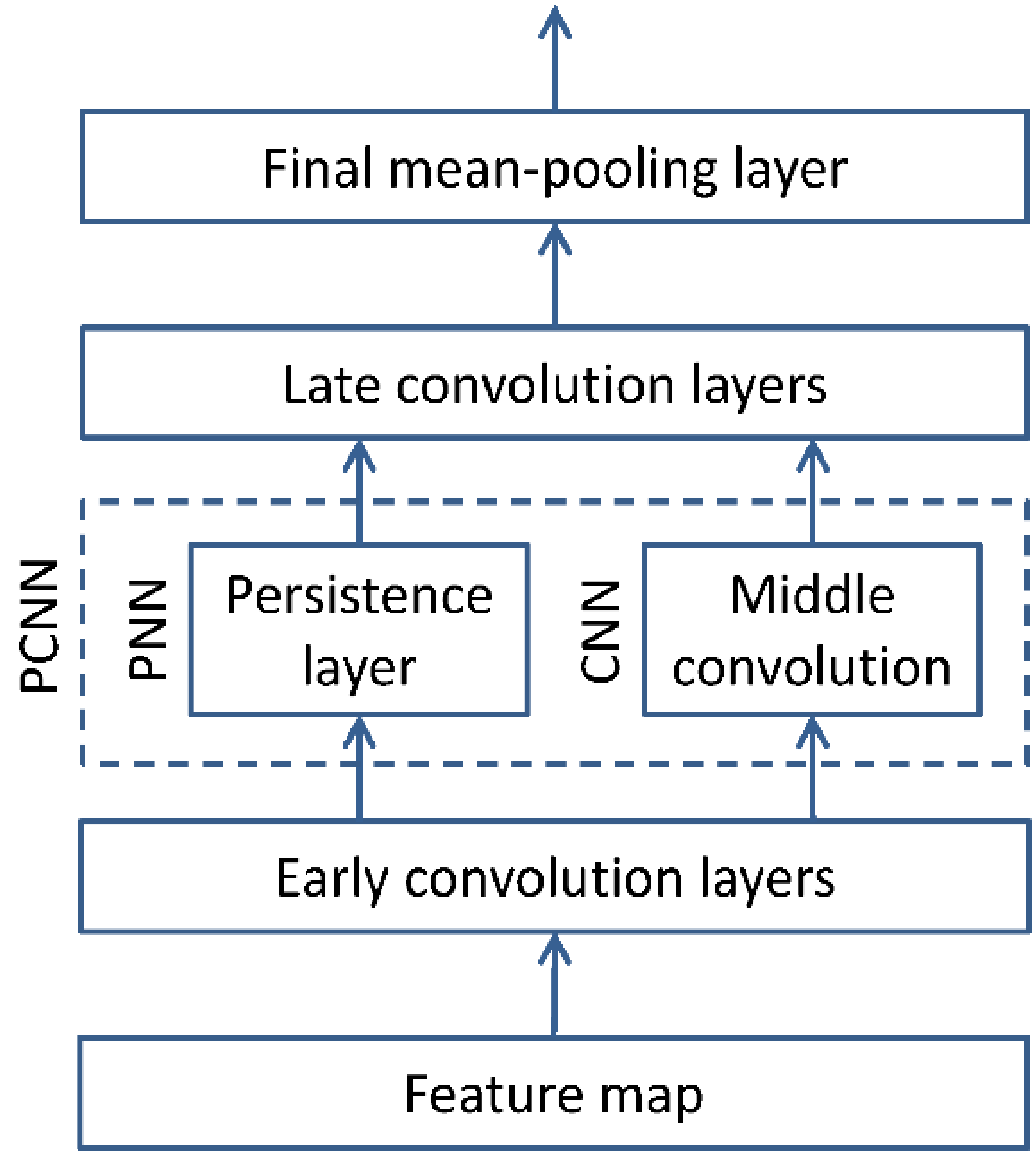


The mysterious Persistence Layer

Audio Signal Classification

Idea: Use filters of early convolution layers to compute Persistence Landscapes in so-called Persistence Layer

But how to backpropagate??



J.Y. Liu et al., **Applying Topological Persistence in Convolutional Neural Network for Music Audio Signals**, *arXiv*, 2016

The mysterious Persistence Layer

Audio Signal Classification

A. How Back-propagation Works through the Persistence Layer

Persistence landscapes are constructed from piece-wise linear functions $f_{(b_i,d_i)}$. The values of $f_{(b,d)}$ are composed of linear functions of b and d in a birth-death pairs (b,d), as shown in Equation ($\boxed{1}$). A persistence landscape is simply a re-ordering of the function values in its sampled matrix form. Note that the deaths and births are all local extrema. For an element in a persistence landscape matrix, the back-propagation is done through the elements which own the birth or death value.

J.Y. Liu et al., **Applying Topological Persistence in Convolutional Neural Network for Music Audio Signals**, *arXiv*, 2016

Thank you!!