Understanding Learned Representations in Deep Neural Networks without Supervision

Wonjoon Chang

Korea Advanced Institute of Science and Technology (KAIST), South Korea

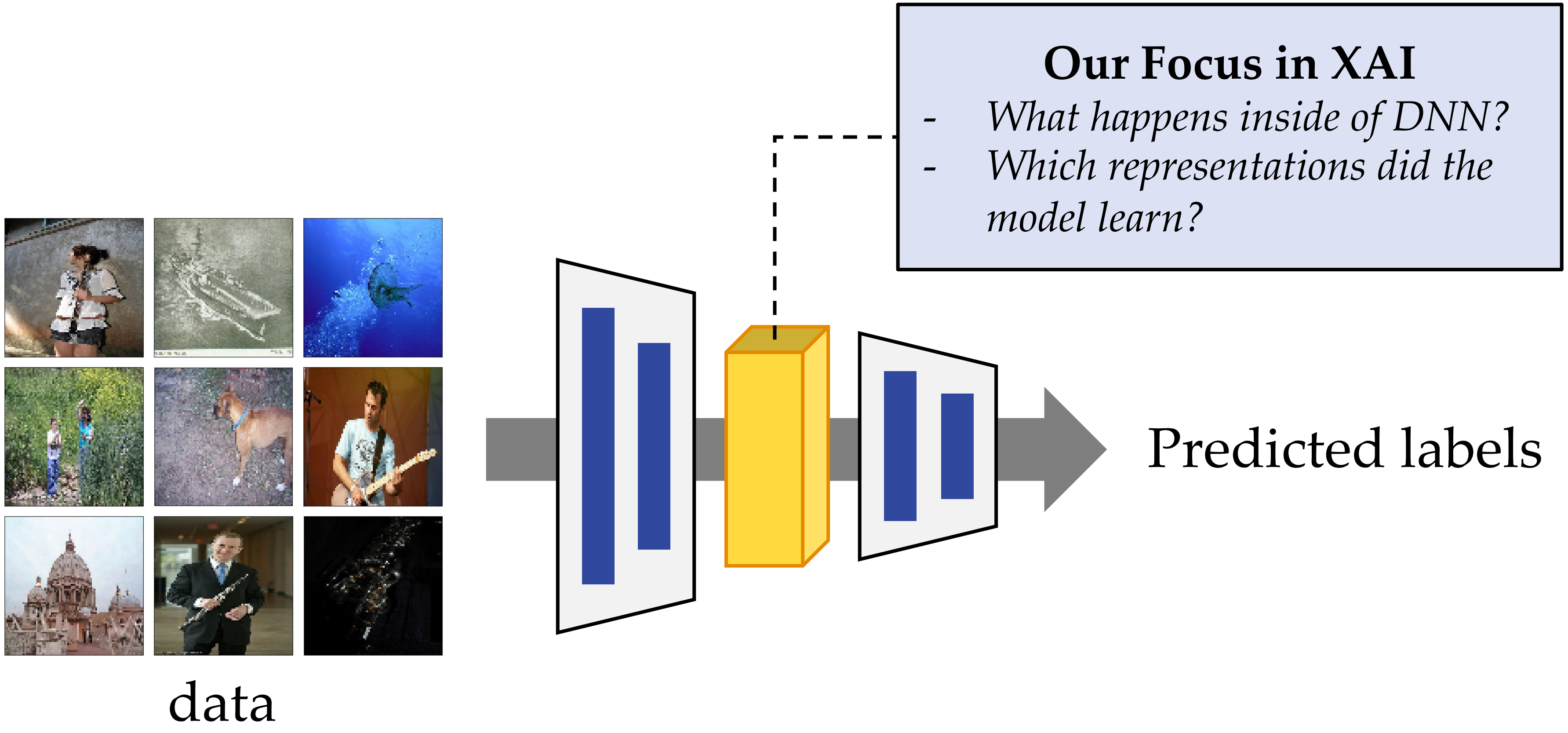
Statistical Artificial Intelligence Lab (SAILab)

one_jj@kaist.ac.kr

2024.03.21

Problem

- It is important to identify the ***internal representations*** that DNN implicitly learned for DNN interpretability.



Problem

- It is important to identify the ***internal representations* that DNN implicitly learned** for DNN interpretability.
- Understanding *"Coherent properties"* help us to explain and interpret the general behaviors of the model.

Subclass detection

misclassification

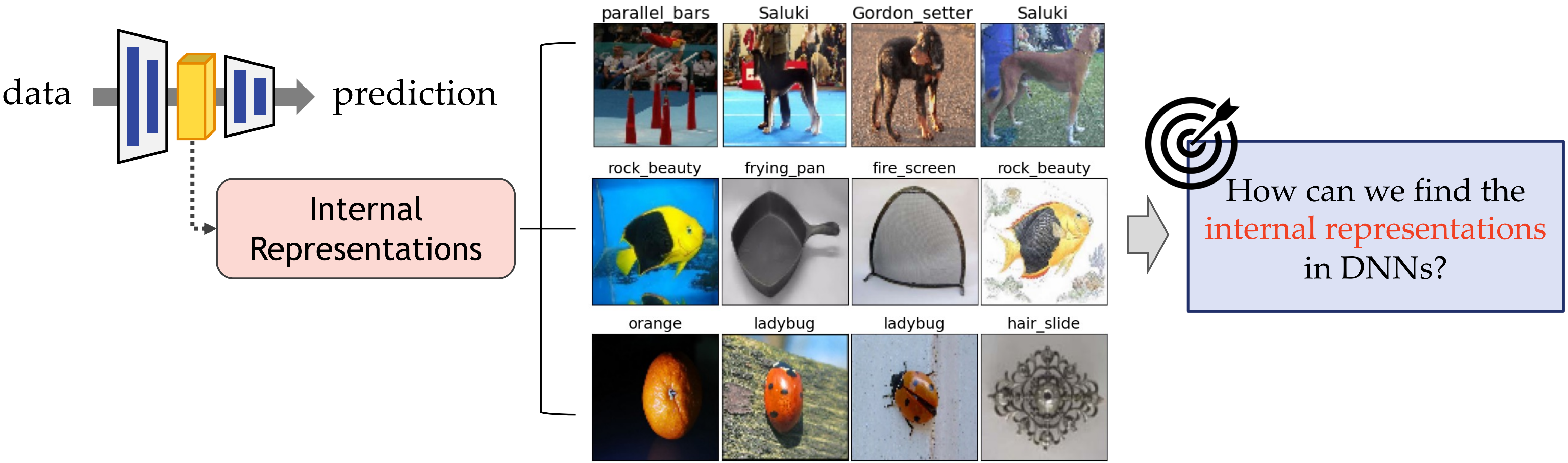
[Ground Truth]
French bulldog

[Prediction]
Saluki

Problem

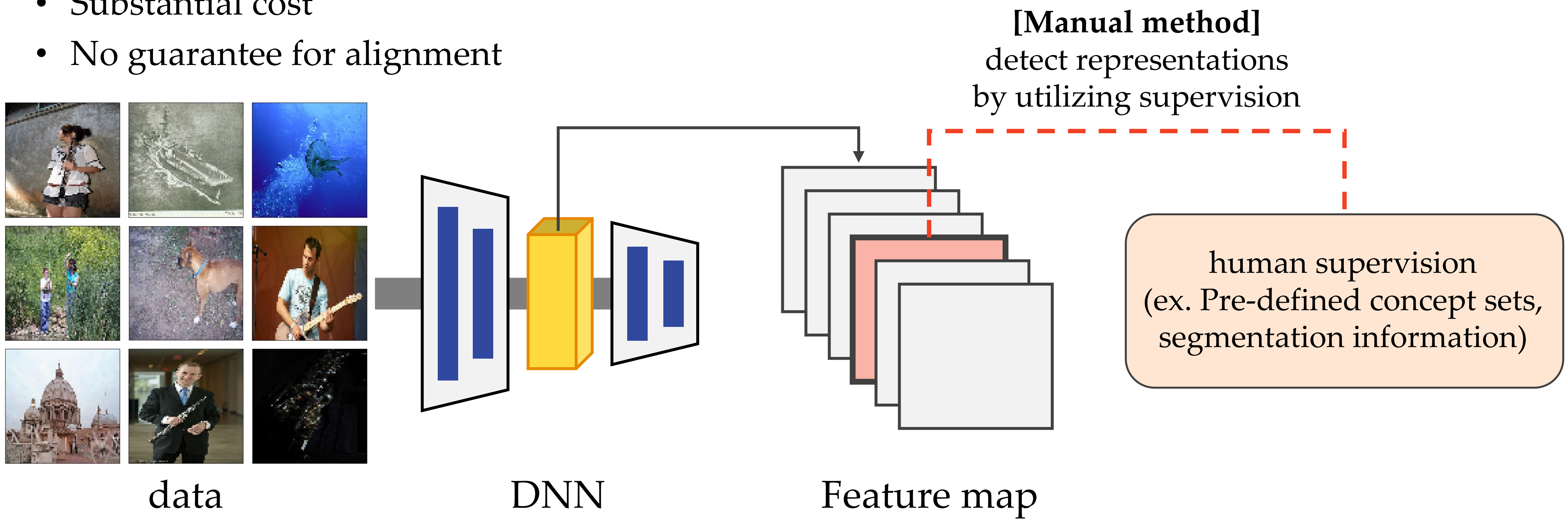
- It is important to identify the ***internal representations*** that DNN implicitly learned for DNN interpretability.
- **Internal representations**

  Implicitly learned concepts that multiple instances share in the internal feature space.



Challenges

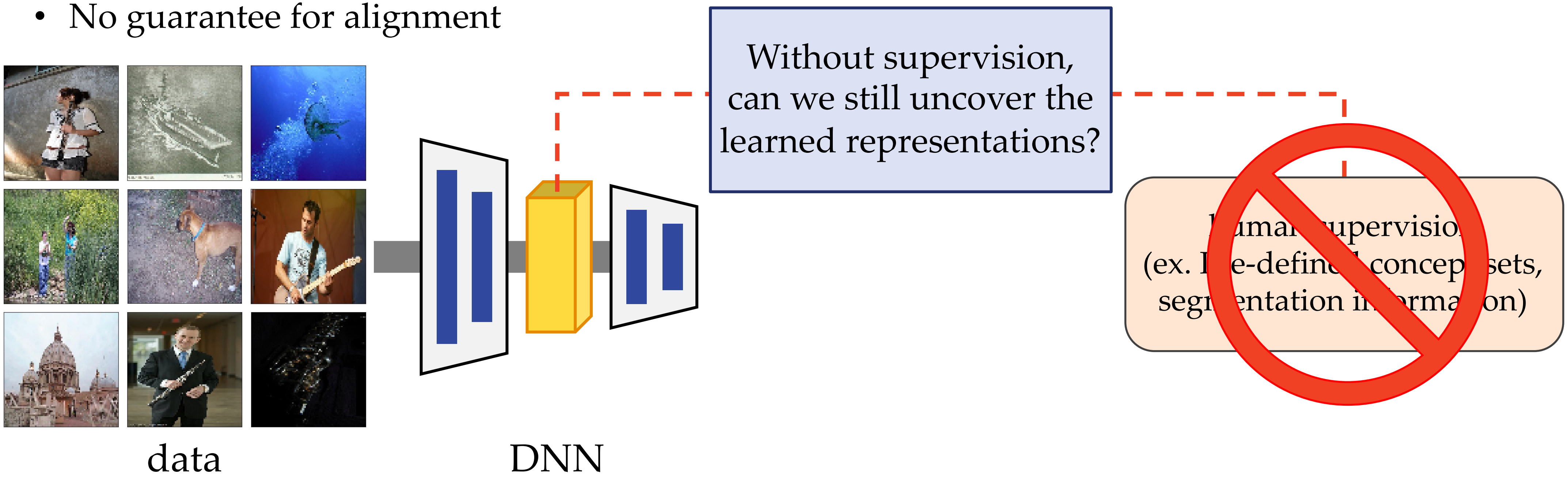
- How can we reveal learned representations in the intermediate feature space of DNN?
- Mostly, human supervision is necessary.
  - Substantial cost
  - No guarantee for alignment



- Network Dissection: Quantifying Interpretability of Deep Visual Representations, 2017
- Interpretability beyond feature attribution quantitative testing with concept activation vectors, 2018
- Best of both worlds: local and global explanations with human-understandable concepts, 2021

Challenges

- How can we reveal learned representations in the intermediate feature space of DNN?
- Mostly, human supervision is necessary.
  - Substantial cost
  - No guarantee for alignment

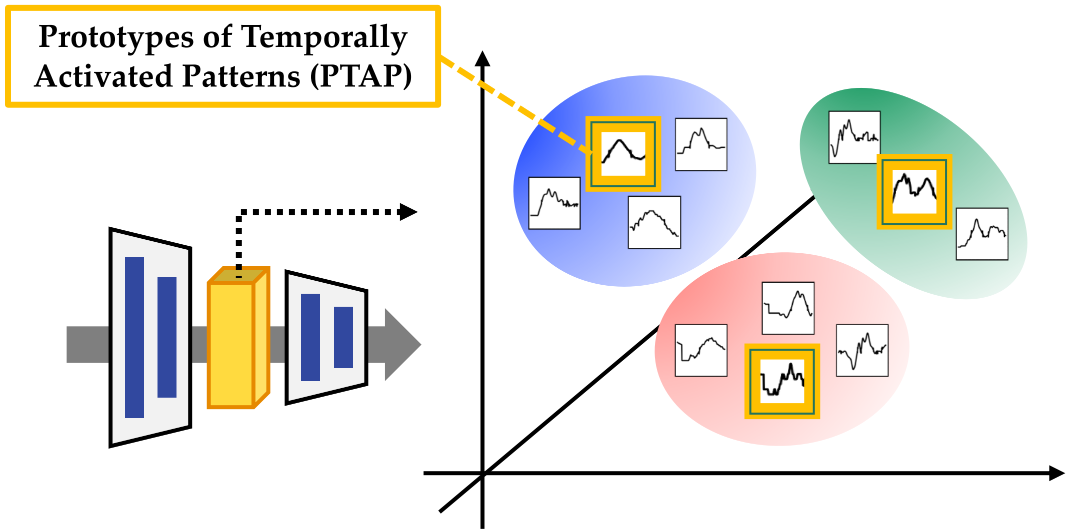


- Network Dissection: Quantifying Interpretability of Deep Visual Representations, 2017
- Interpretability beyond feature attribution quantitative testing with concept activation vectors, 2018
- Best of both worlds: local and global explanations with human-understandable concepts, 2021

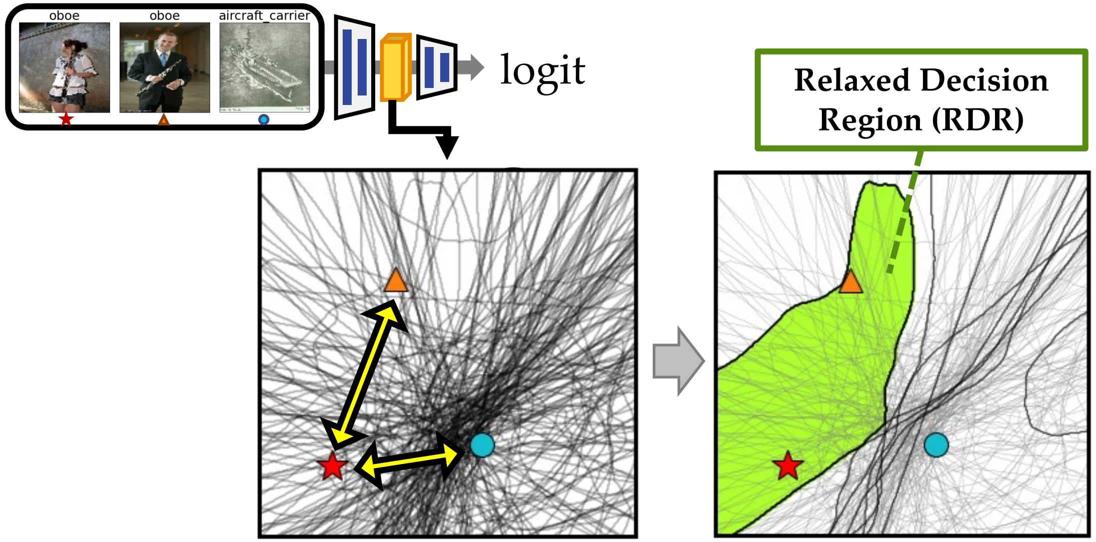
Overview

- How can we reveal **learned representations** in the intermediate feature space of DNN without human supervision?
- Our work

1. Interpreting Internal Activation Patterns in Deep Temporal Neural Networks by Finding **Prototypes** (KDD-21)



2. Understanding Distributed **Representations of Concepts** in Deep Neural Networks without Supervision (AAAI-24)



Interpreting Internal Activation Patterns in Deep Temporal Neural Networks by Finding Prototypes

Sohee Cho*, Wonjoon Chang*, Ginkyeng Lee, Jaesik Choi

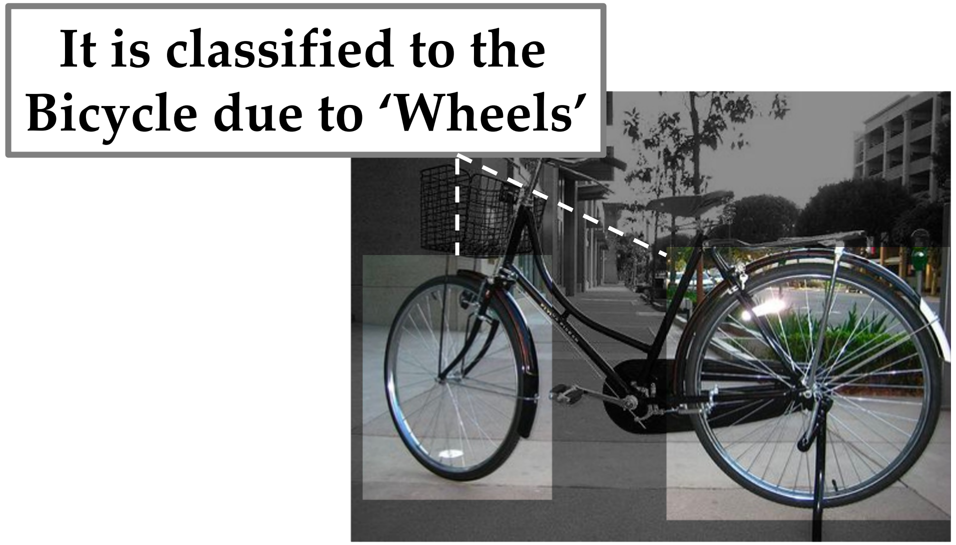
Korea Advanced Institute of Science and Technology (KAIST), South Korea

Statistical Artificial Intelligence Lab (SAILab)

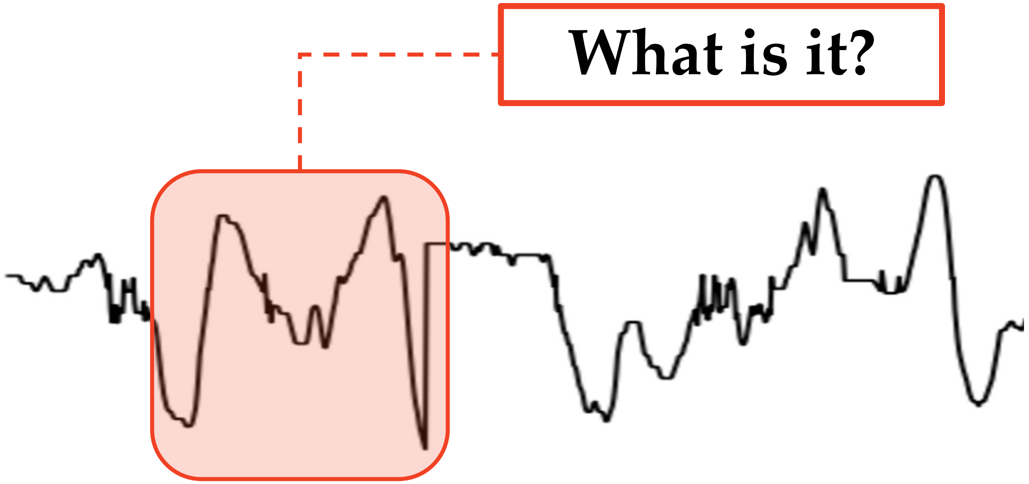
* Equal Contribution

Challenges

- How can we reveal implicit representations in the intermediate feature space of DNN? ➔ Human supervision (semantic labels) may be helpful.
- But, in time series data, there is usually no labels for representatives.



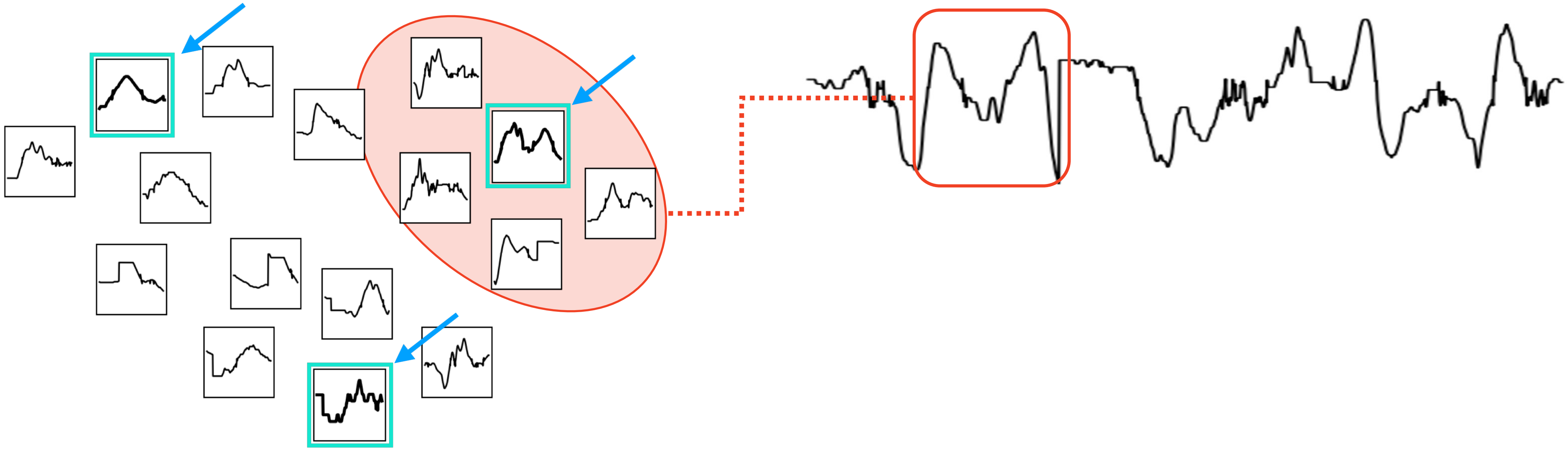
Image



Time series

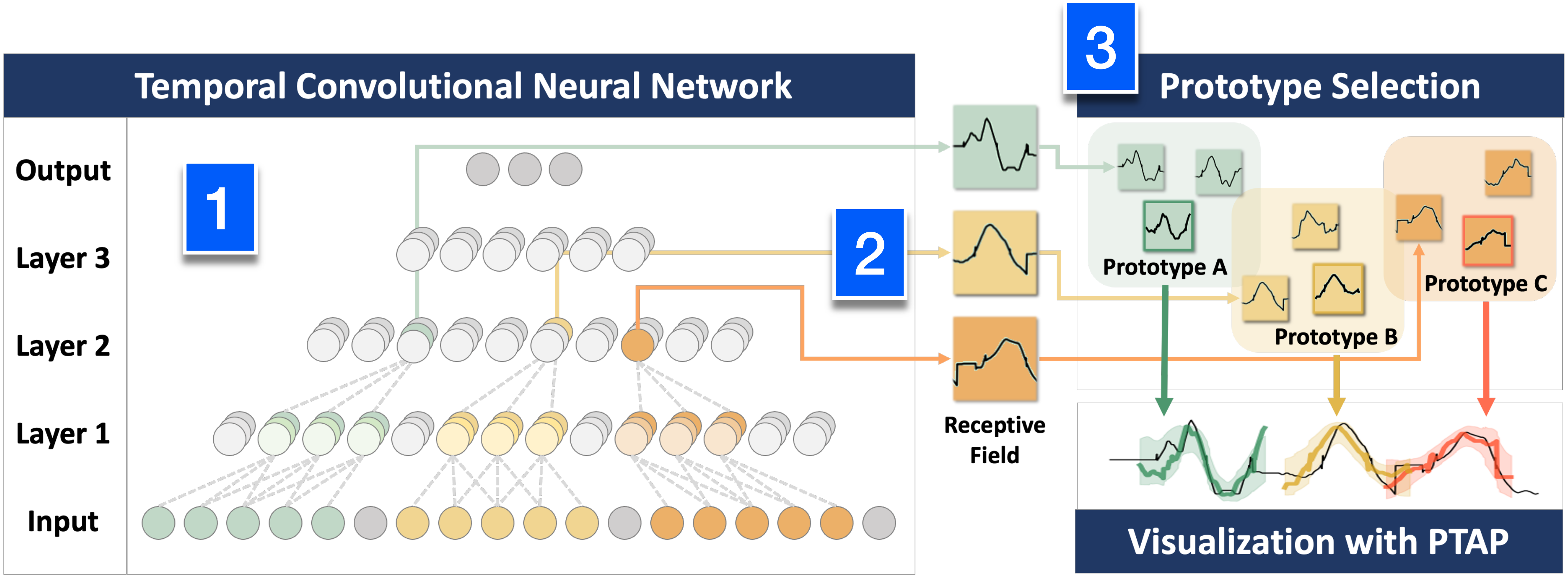
Challenges

- In time series data, there is usually no labels for representative patterns.
- Representative examples (Prototypes) help to understand captured patterns in data and summarize the distribution of patterns.
- How can we find appropriate representative examples in time series?



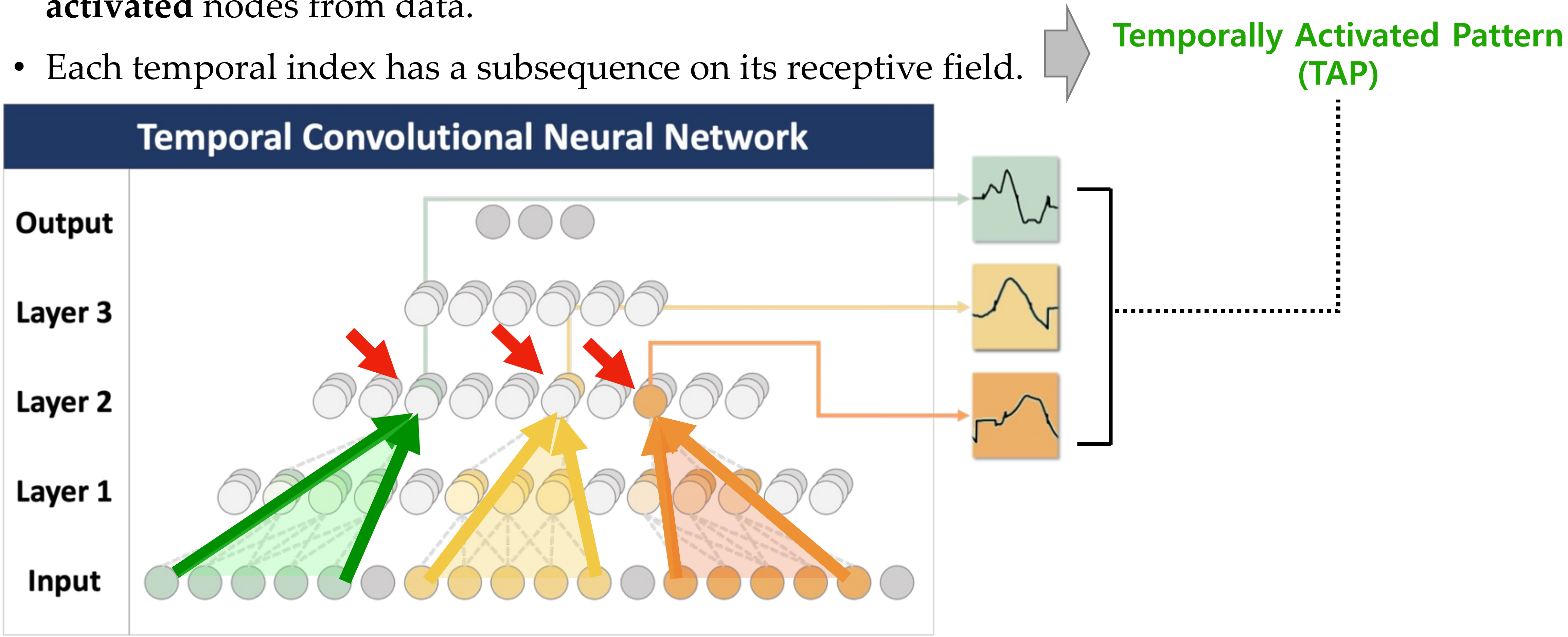
Overview – Prototypes of Temporally Activated Patterns

- Find appropriate representative temporal patterns by selection prototypes from highly activated subsequences.



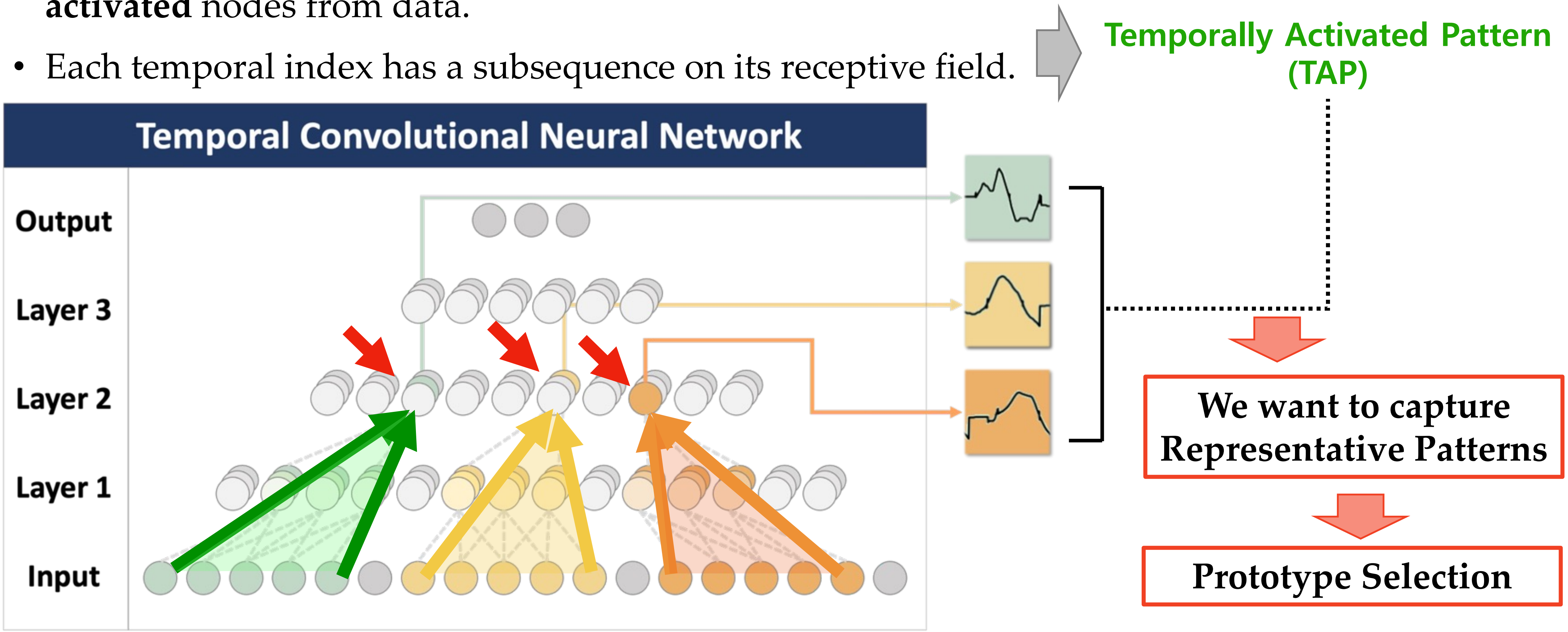
Subsequence Extraction

- Given a trained CNN for time series classification, find temporal indices that have **highly activated** nodes from data.
- Each temporal index has a subsequence on its receptive field.



Subsequence Extraction

- Given a trained CNN for time series classification, find temporal indices that have **highly activated** nodes from data.
- Each temporal index has a subsequence on its receptive field.



Prototype Selection

- How can we choose good examples (prototypes) to represent temporal patterns from Temporally Activated Patterns (TAPs)?
- Efficient greedy algorithm to select prototypes from high dimensional data

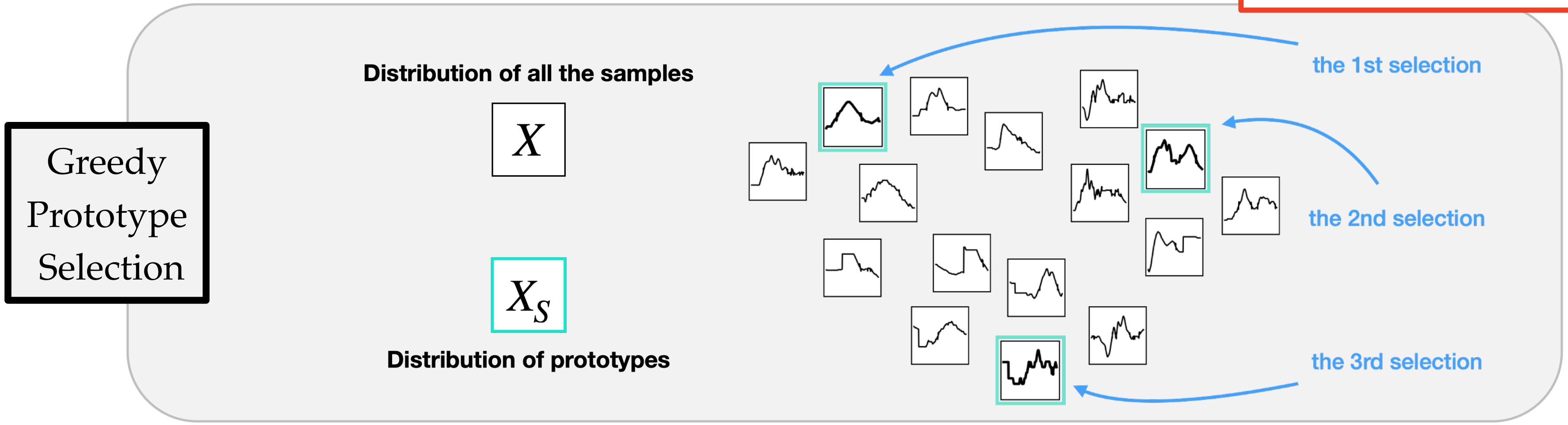
$J_b(S)$: objective

k : kernel function

maximize $$J_b(S) = \frac{2}{n|S|} \sum_{i \in [n], j \in S} k(x_i, y_j) - \frac{1}{|S|^2} \sum_{i,j \in S} k(x_i, x_j)$$

n : the number of the total samples

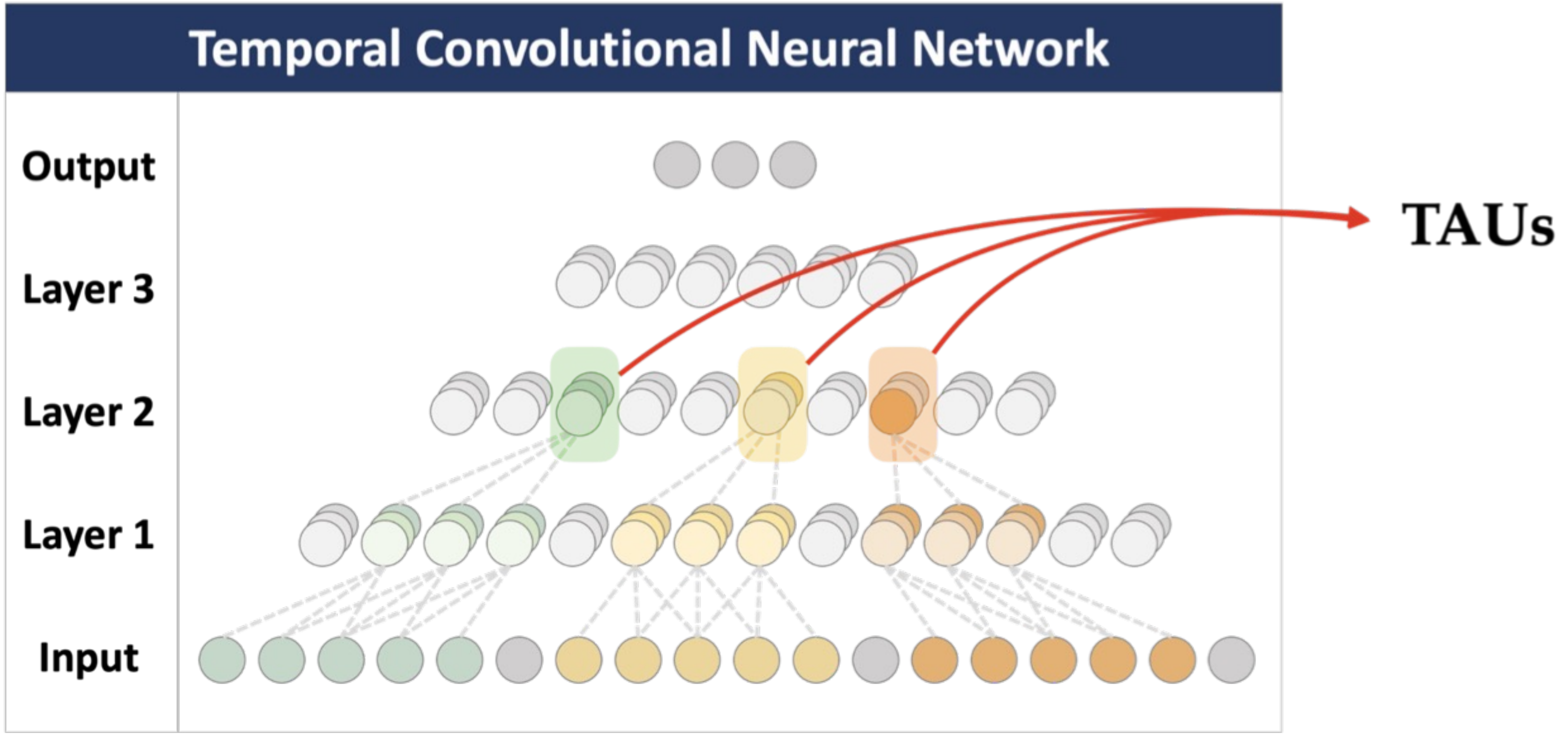
Kernel evaluates the similarity



- Examples are not enough, learn to criticize! Criticism for Interpretability, 2016

Feature-based Similarity

- Can we utilize the feature vectors in the internal nodes during prototype selection?
- Temporally Activated Unit (TAU) : the feature vector at the specific temporal point

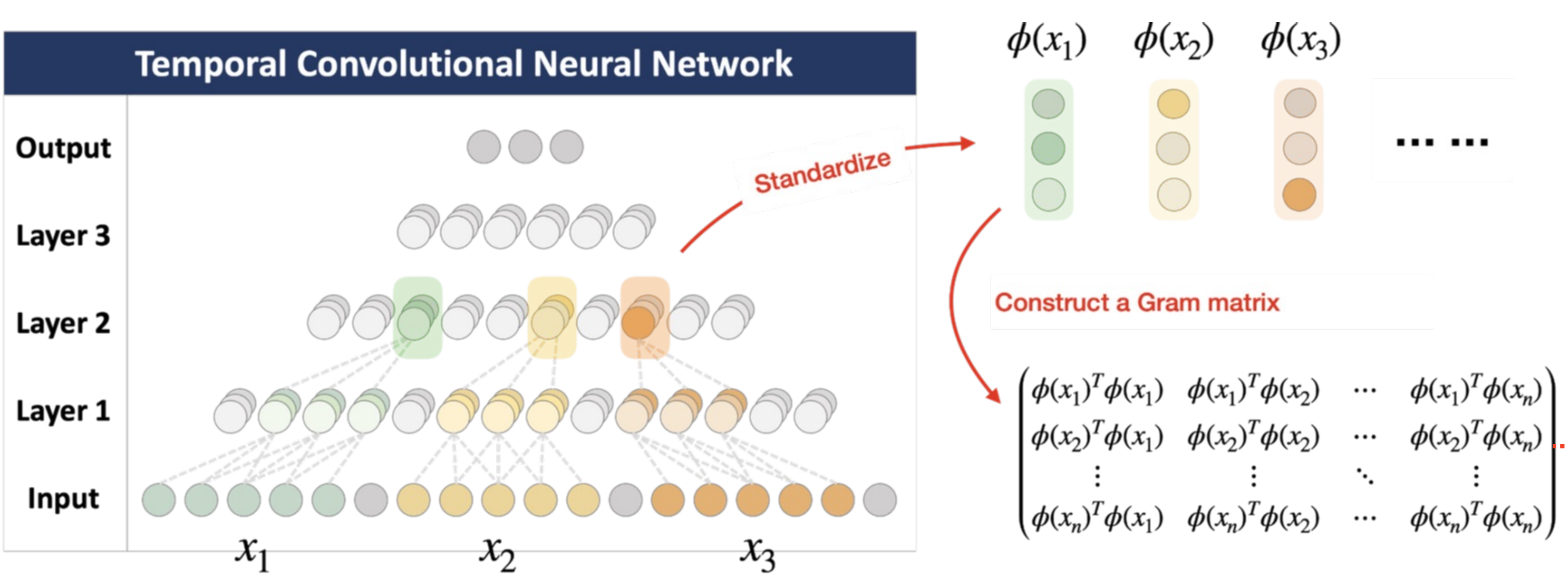


Gram Kernel Matrix

- We propose to use the **Gram kernel matrix** using Temporally Activated Units.

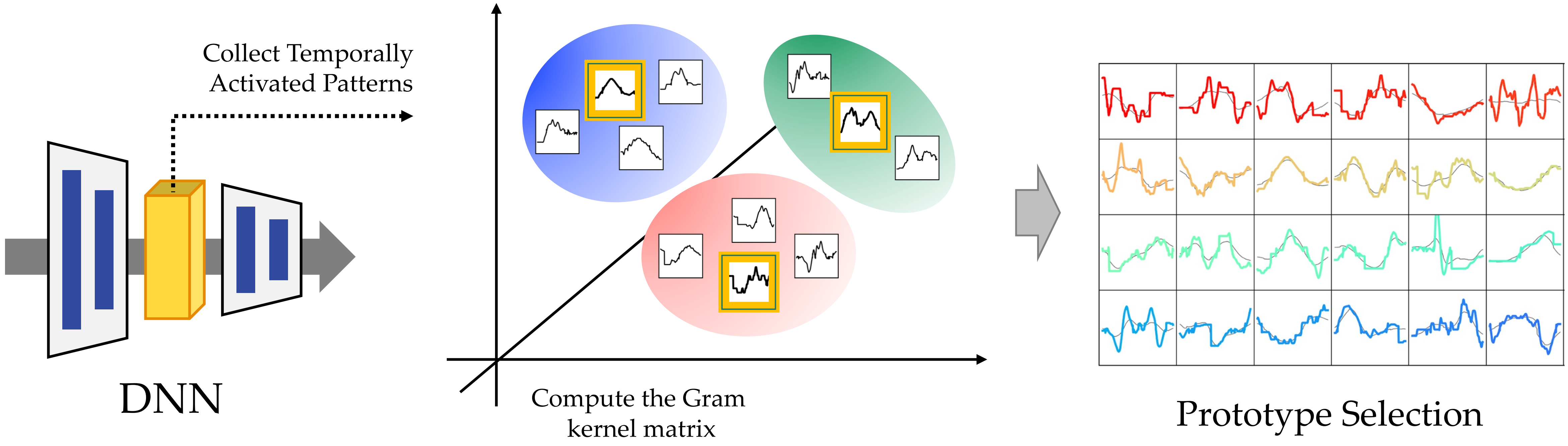
$$\frac{2}{n|S|}\sum_{i\in[n],j\in S} k(x_i, x_j) - \frac{1}{|S|^2}\sum_{i,j\in S} k(x_i, x_j)$$

Kernel evaluates the similarity



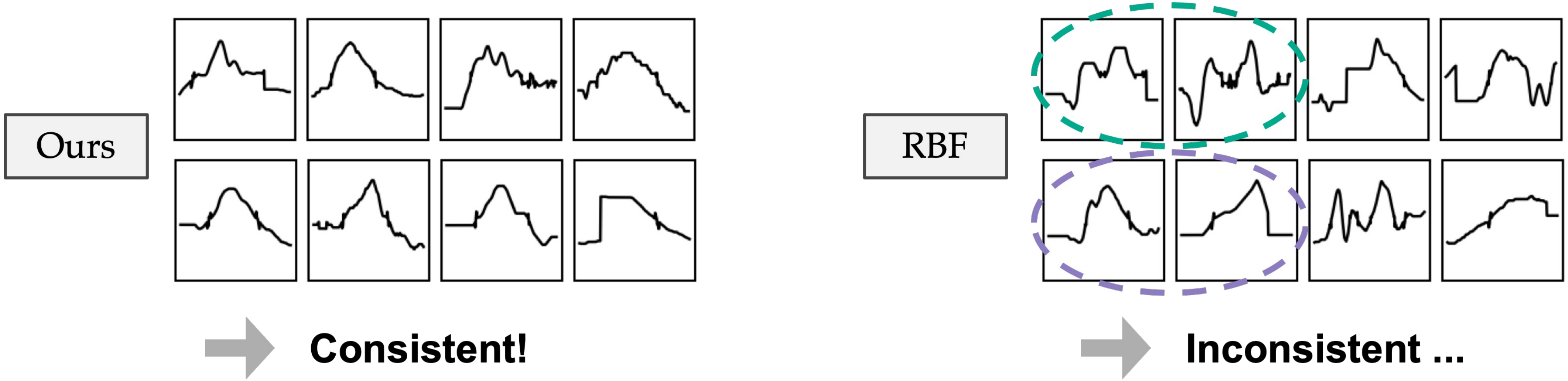
Prototype Selection in the feature space

- We propose to select prototypes with feature activations from the internal nodes of the neural network.
- We use the **Gram kernel matrix** constructed by feature vectors to use the greedy selection algorithm ➔ **Prototypes of Temporally Activated Patterns (PTAP)**



Effectiveness of PTAP

- The **Gram kernel matrix** is useful to capture learned temporal patterns.
- What is a good prototype for temporal data?
    1. Each prototype group must have a coherent pattern.

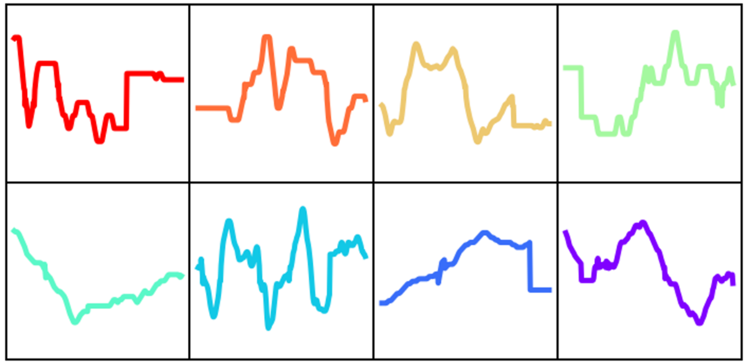


* Radial basis function (RBF) kernel: $\exp(-2\gamma\|x_i - x_j\|_2^2)$ with large γ

Effectiveness of PTAP

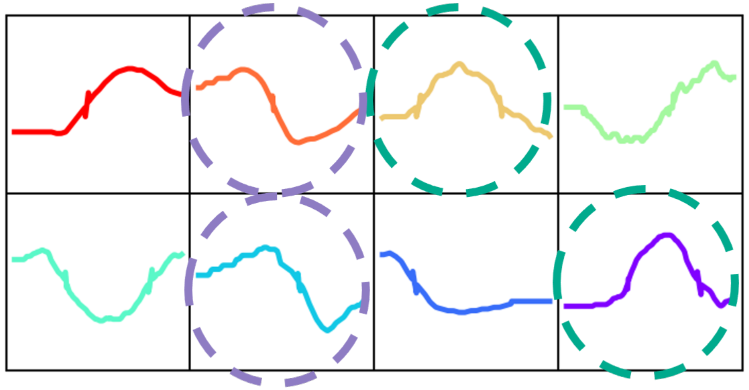
- The **Gram kernel matrix** is useful to capture learned temporal patterns.
- What is a good prototype for temporal data?
    1. Each prototype group has a coherent pattern.
    2. Prototypes have different shapes from each other.

Ours



Various shapes!

RBF

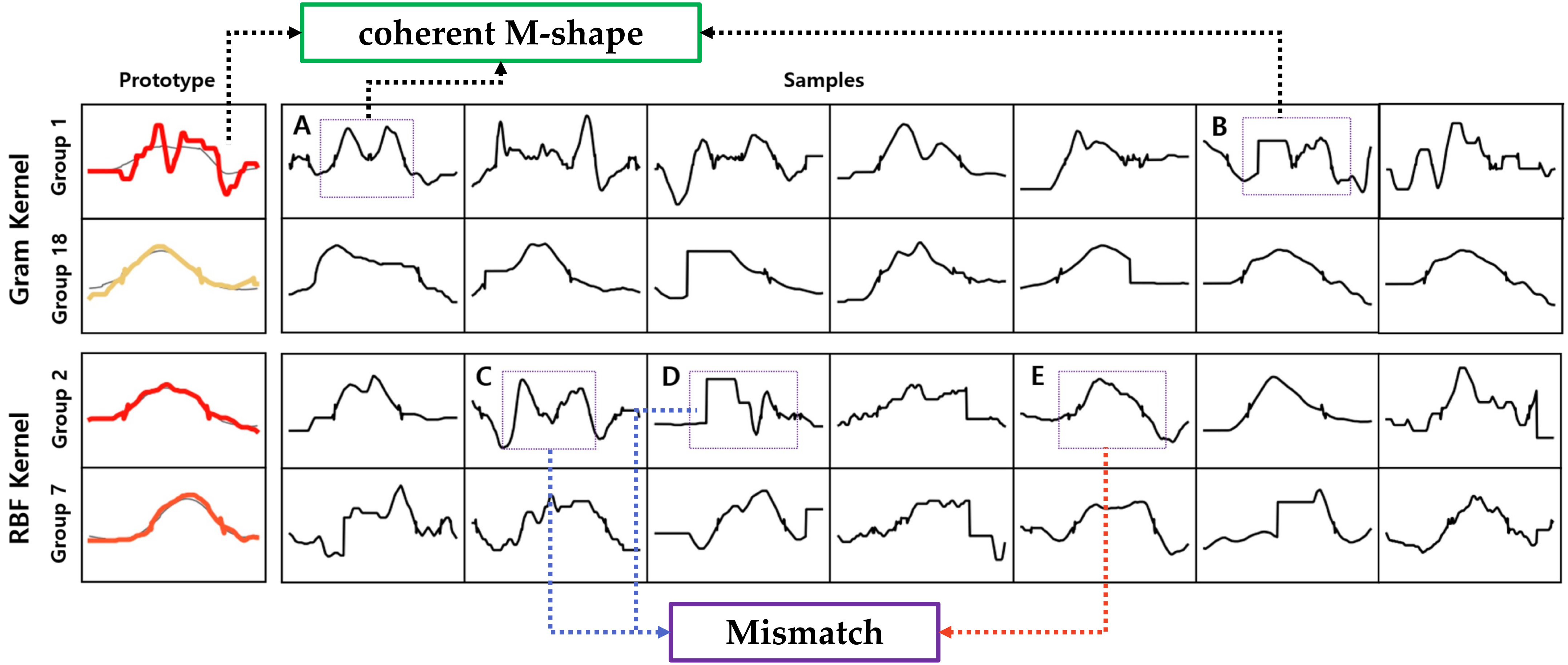


Shifted & Smoothed ...

* Radial basis function (RBF) kernel: $\exp(-2\gamma\|x_i - x_j\|_2^2)$ with large γ

Effectiveness of PTAP

- The results of prototype selection with the Gram kernel matrix



Understanding Distributed Representations of Concepts in Deep Neural Networks without Supervision

Wonjoon Chang*, Dahee Kwon*, Jaesik Choi

Korea Advanced Institute of Science and Technology (KAIST), South Korea

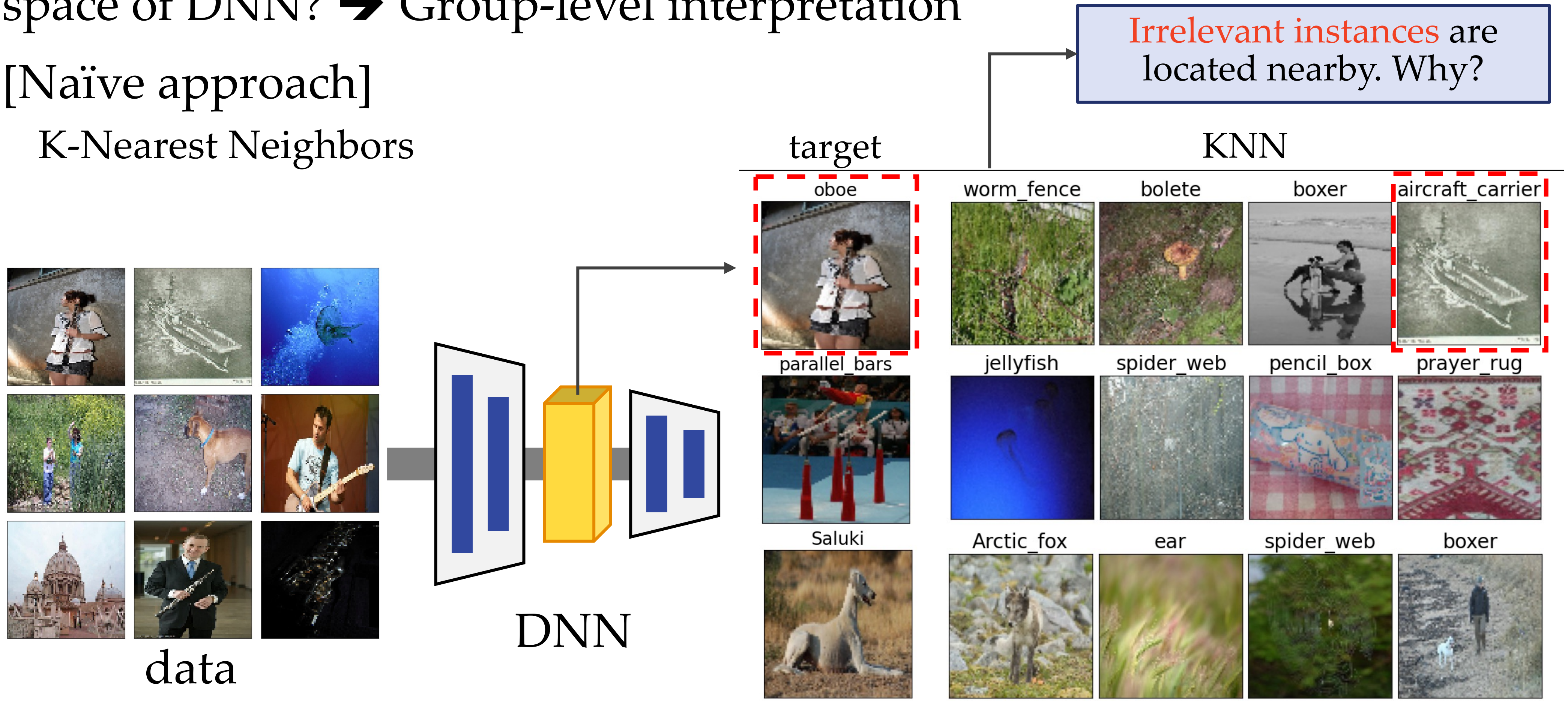
Statistical Artificial Intelligence Lab (SAILab)

* Equal Contribution

Challenges

- How can we reveal implicit representations in the intermediate feature space of DNN? ➔ Group-level interpretation
- [Naïve approach]

  K-Nearest Neighbors



Challenges

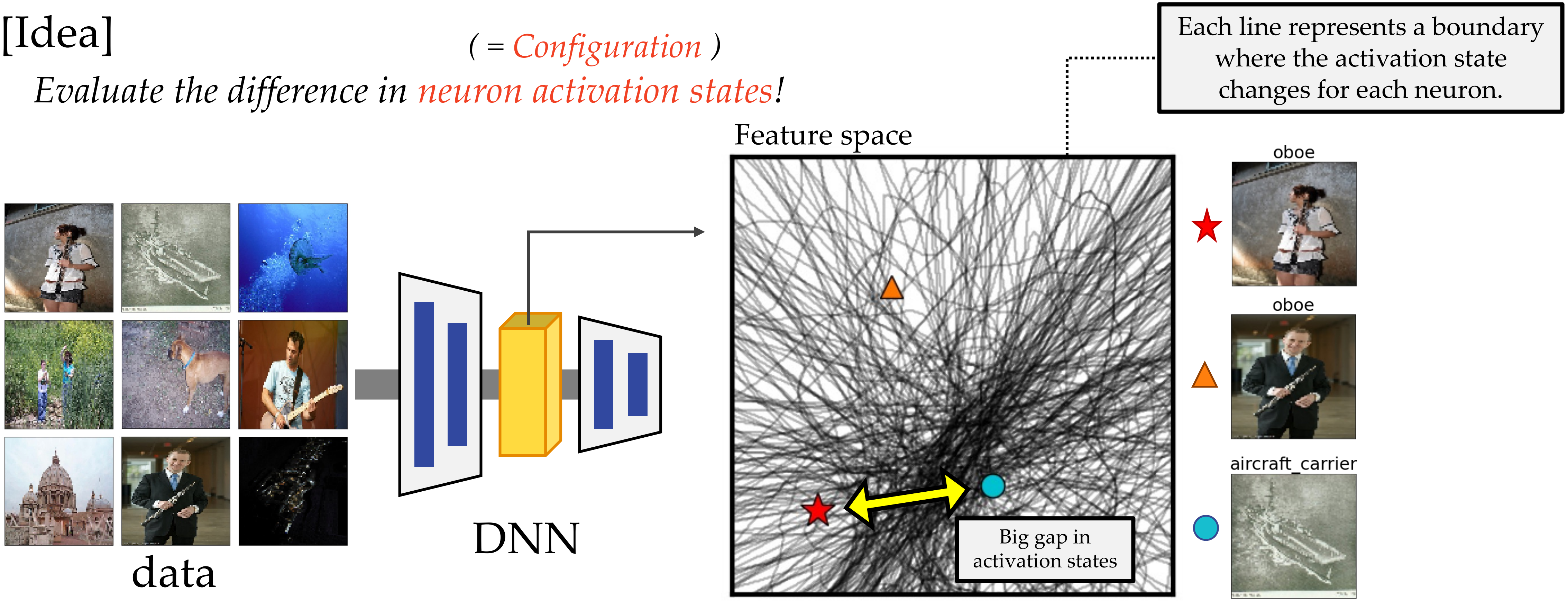
- [Challenge – *complex internal space*]

  DNN utilizes different information from data according to the local region of the internal space.

- [Idea]

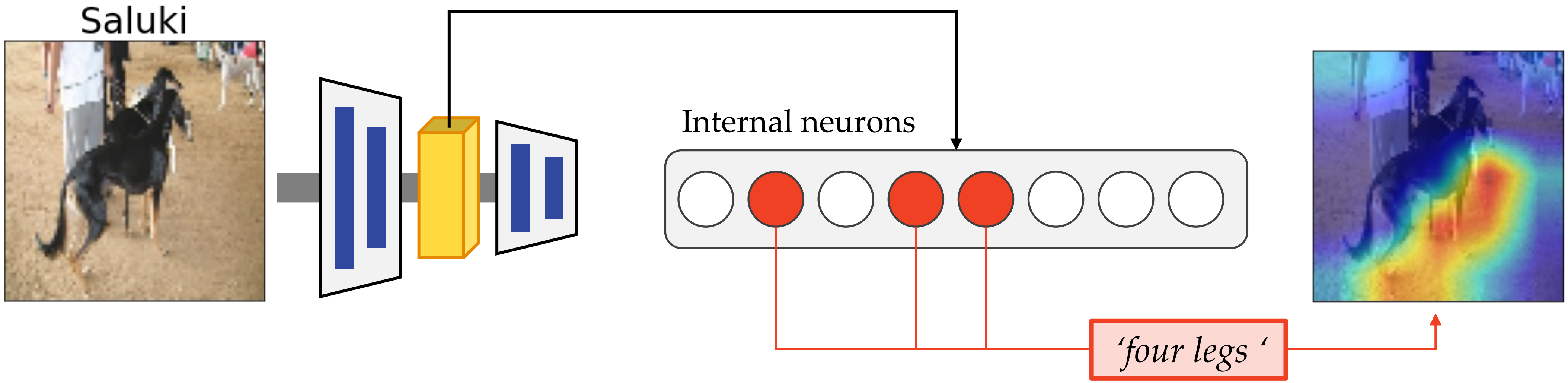
  (= Configuration)

  Evaluate the difference in neuron activation states!



Distributed Representations

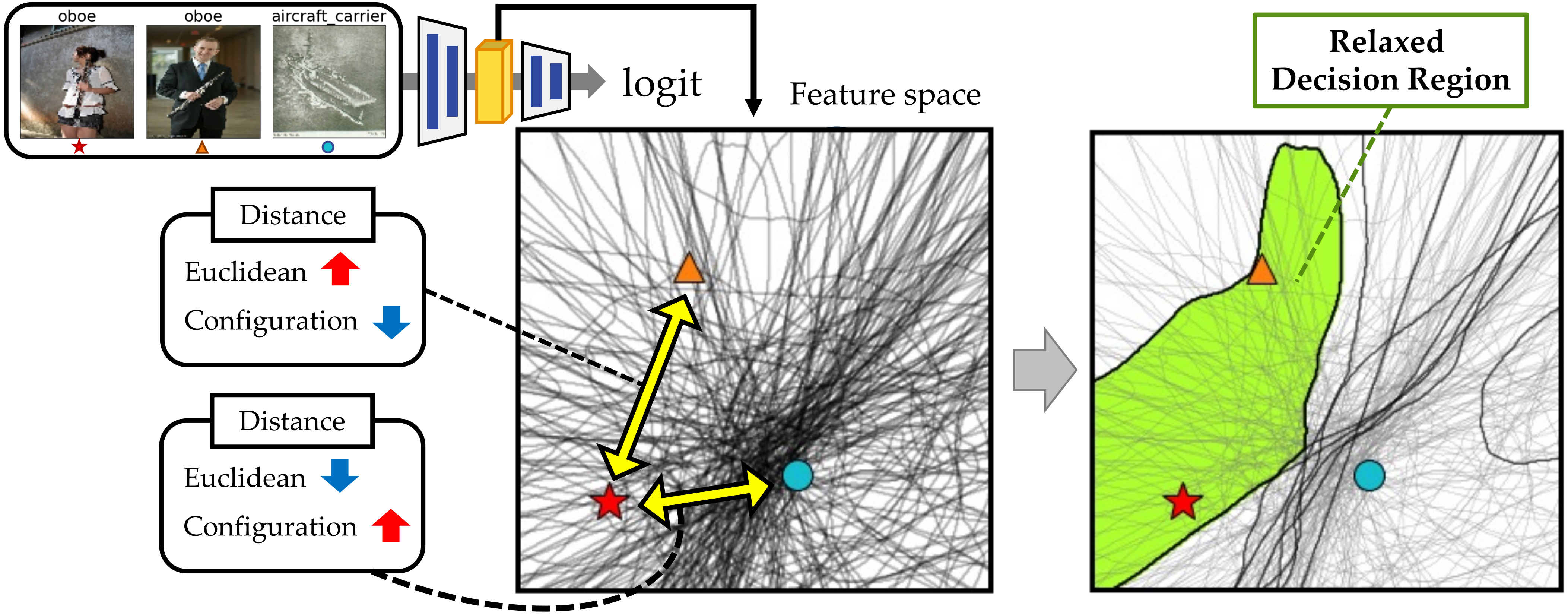
- *Distributed Representations*
  each concept that the model learned is represented by multiple internal neurons.
- Neuron activation states may be highly related to the concepts in DNNs.



- Learning distributed representations of concepts, 1986
- Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks, 2018

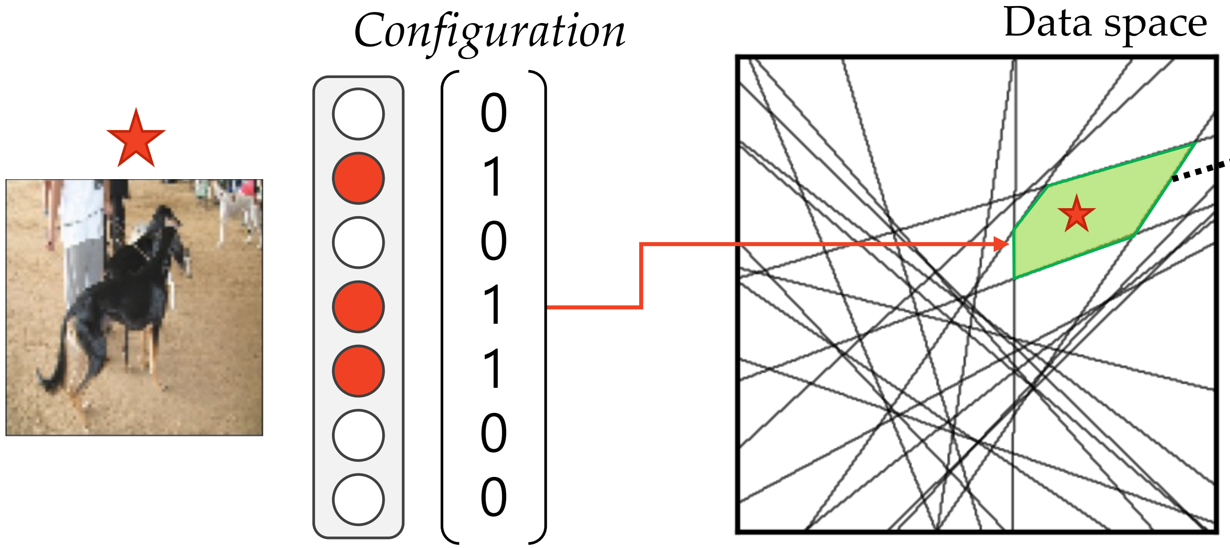
Overview – Relaxed Decision Region

- Find a principal configuration where a target and relevant samples share learned representations by using configuration information.



Configuration

- Why do we focus on neuron activation to capture representations of concepts?
- Configuration
  - a binary vector that represent activation states of neurons
  - *Configuration determines the mapping of DNN in the* ***local region*** *[3]*



Decision Region

There are too many regions inside DNN…
Can we relax regions to identify coherent representations?

- Exact and consistent interpretation for piecewise linear neural networks: A closed form solution, 2018

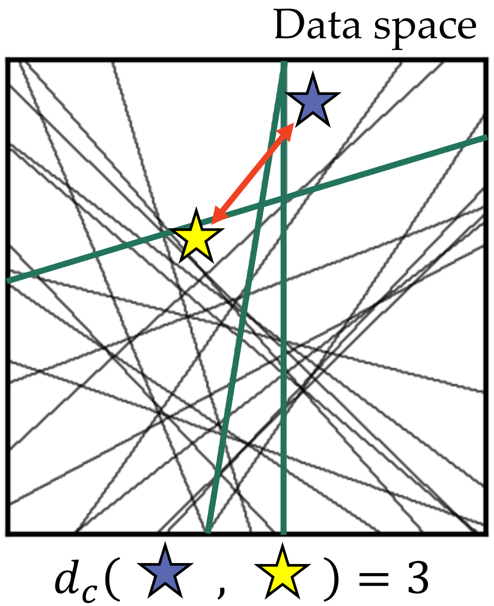
Configuration Distance

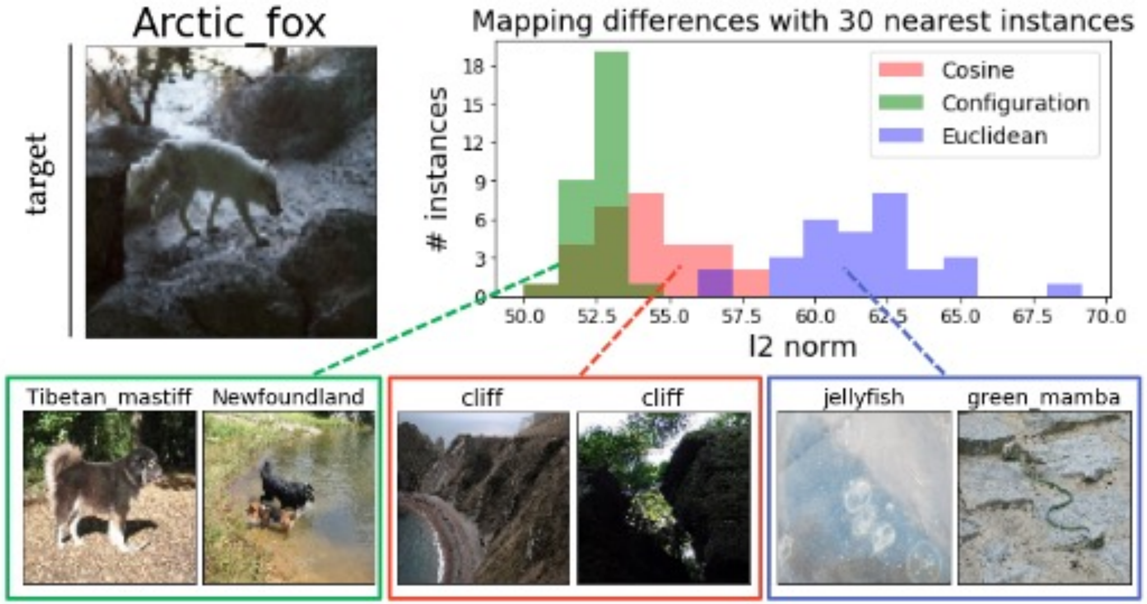
- Definition

  Given an instance $x, \tilde{x} \in \mathcal{X}$, the Configuration distance for a set of neurons N is defined as follows:

$$d_C(x, \tilde{x}) = d_H(c^N(x), c^N(\tilde{x}))$$

  where d_H denotes the Hamming distance.





Algorithm

- Select t principal neurons to construct an internal region that exhibits strong coherence with a target instance $\mathbf{x}$, while ensuring distinctiveness from irrelevant instances.

$$\min_{\substack{\mathbf{c}_\mathrm{p}\in\{0,1\}^t \\ N^*\subset N}} \mathbb{E}_{\mathbf{x}}[d_H(\mathbf{c}^{N^*}(\mathbf{x}), \mathbf{c}_\mathrm{p})] - \mathbb{E}_{\mathbf{y}}[d_H(\mathbf{c}^{N^*}(\mathbf{y}), \mathbf{c}_\mathrm{p})] \quad (5)$$

$$\text{s.t. } |N^*| = t$$

Exhibit strong **coherence** with the positive set,
while ensuring **distinctiveness** from the negative set.

Algorithm

- *positive set S*: automatically collect k-nearest neighbors ***based on*** $\boldsymbol{d_C}$

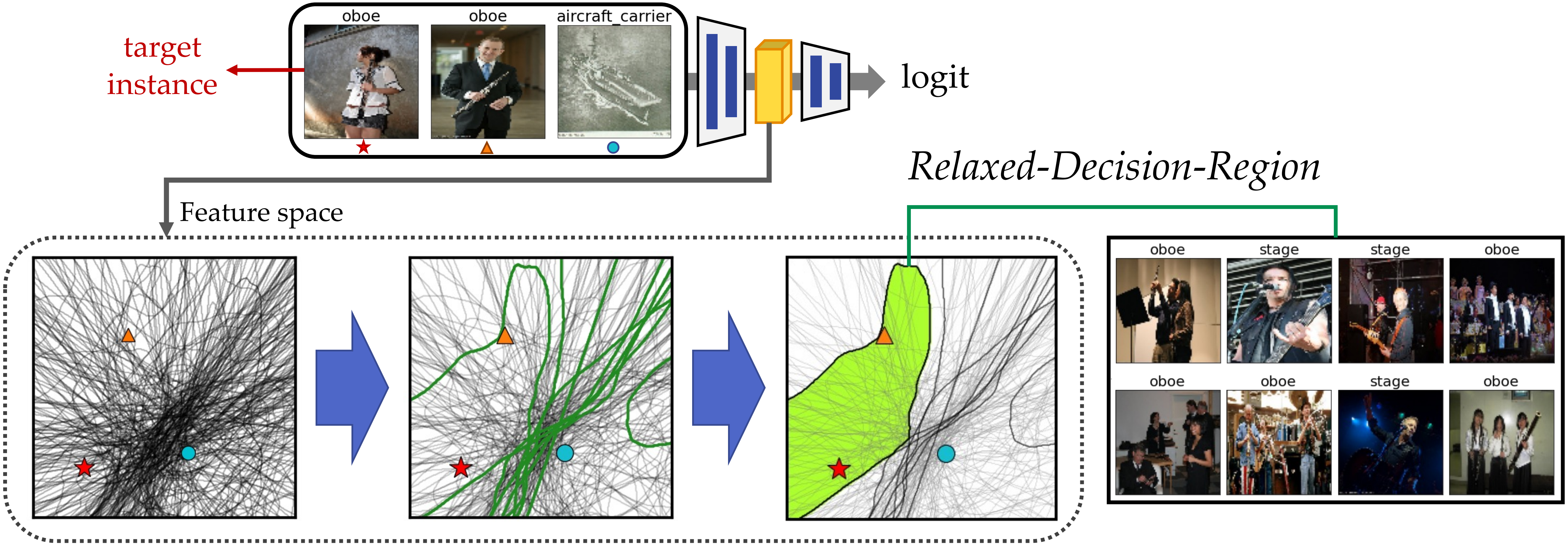
Configuration distance

k=8

- *negative set S_{neg}*: easily sample from remaining data points

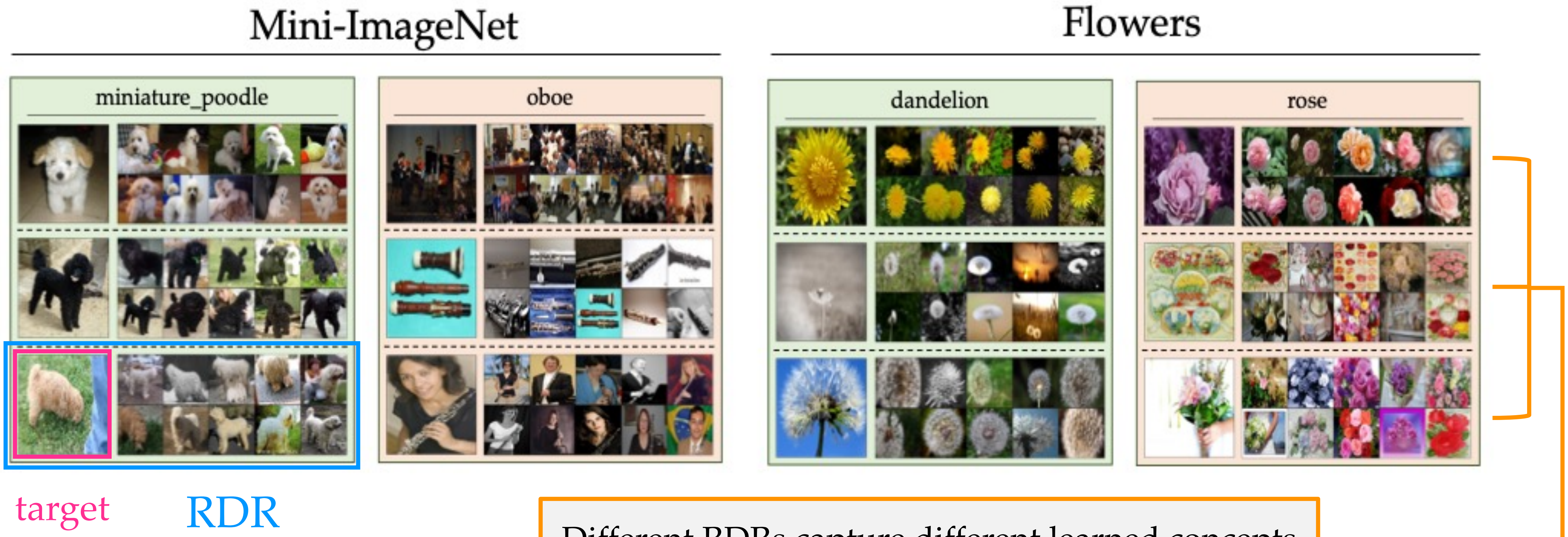
Algorithm

- In greedy method, we sequentially select configuration that minimize Equation (5).



Experiments

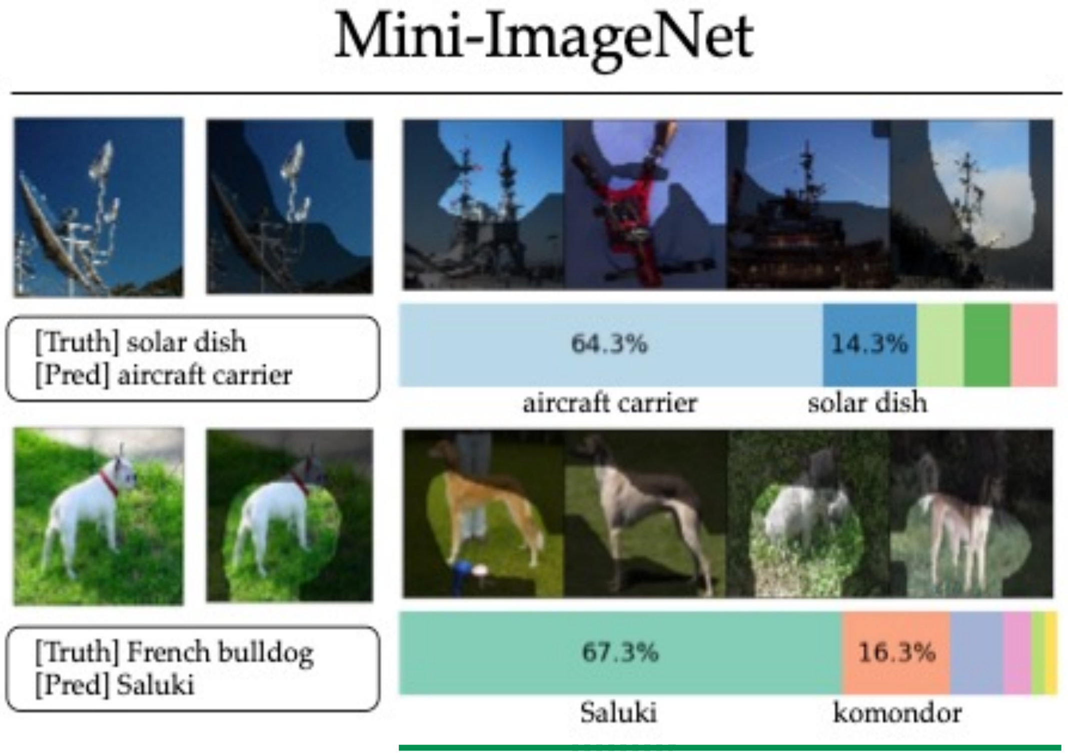
- Finding Unlabeled Subclasses

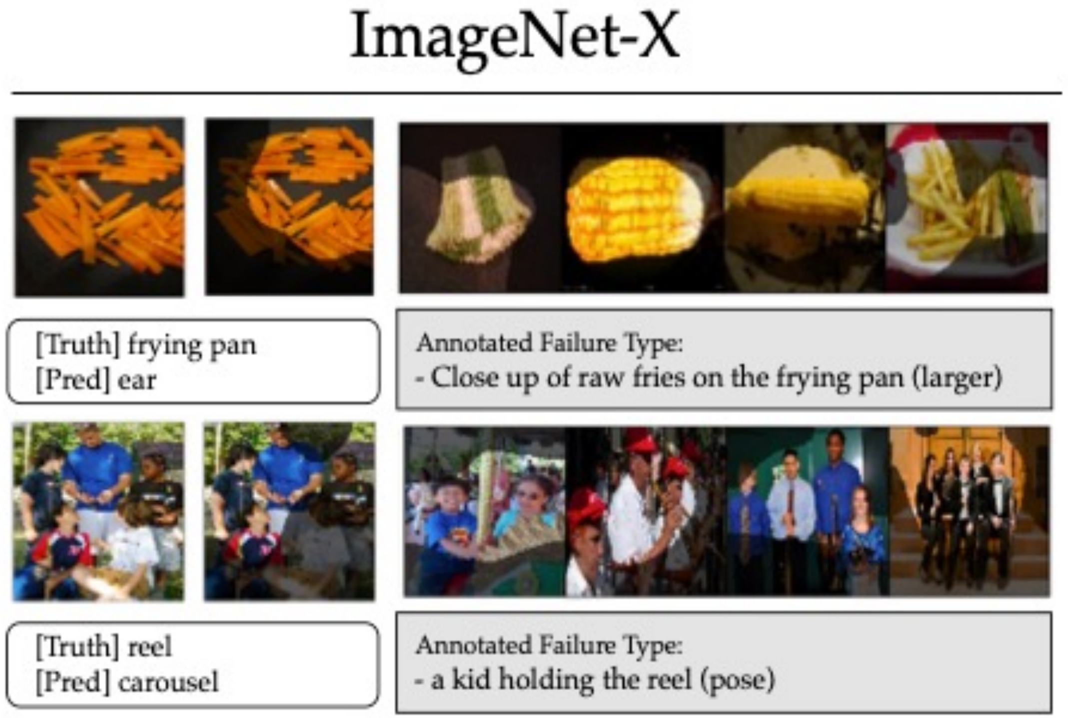


Different RDRs capture different learned concepts without prior knowledge of sublabel information.

Experiments

- Reasoning Misclassified Classes

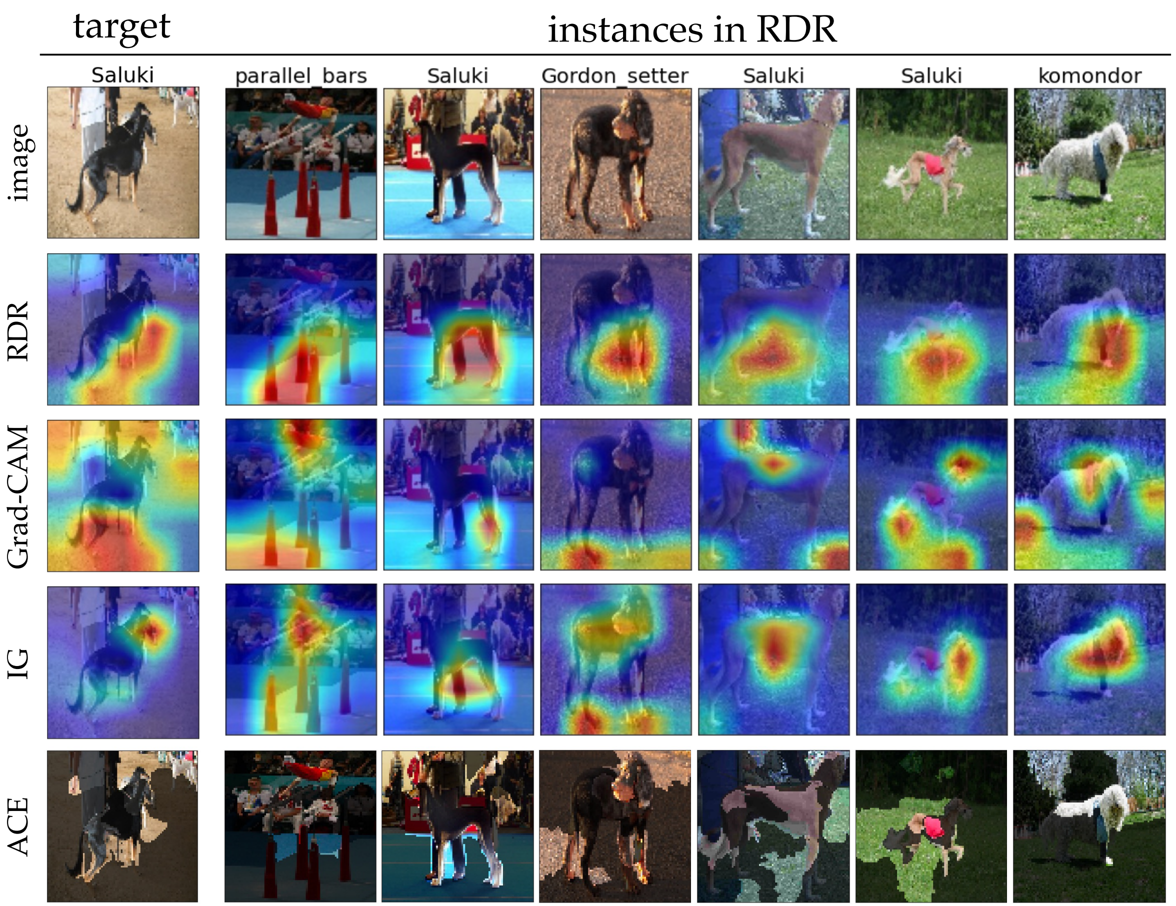




The target sample has similar properties with Saluki due to its long, thin legs

Experiments

- Comparison with other XAI methods



	Purity			Entropy		
	VGG	RSN	MBN2	VGG	RSN	MBN2
RDR	**0.351**	**0.408**	**0.346**	**1.527**	**1.372**	1.531
KNN_C	0.303	0.328	0.329	1.588	1.497	**1.498**
CAR_C	0.022	0.038	0.036	2.264	2.153	2.527
CAV_C	0.314	0.387	0.323	1.549	1.416	1.575
RI	0.045	0.056	0.056	2.161	1.971	2.369
RDR_{Euc}	0.241	0.252	0.303	1.76	1.779	1.76
KNN_{Euc}	0.183	0.166	0.275	1.835	1.862	1.791
CAR_{Euc}	0.039	0.037	0.037	2.272	2.17	2.476
CAV_{Euc}	0.207	0.240	0.283	1.811	1.787	1.745
RDR_{Cos}	0.309	0.307	0.346	1.613	1.7	1.628
KNN_{Cos}	0.250	0.232	0.283	1.672	1.771	1.635
CAR_{Cos}	0.042	0.027	0.036	2.251	2.14	2.576
CAV_{Cos}	0.261	0.283	0.274	1.596	1.734	1.651

$$\text{Purity} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}_{[y_t=\tilde{y}]}$$

$$\text{Entropy} = \sum_{y:\mathcal{P}_y\neq 0} -\mathcal{P}_y * \log \mathcal{P}_y$$

where $\mathcal{P}_y = \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}_{[y_t=y]}$ (empirical distribution).

Conclusion

[Goal]

- Identify **the internal representations that DNN implicitly learned** for DNN interpretability without supervision.

[Prototypes of Temporally Activated Patterns]

- We propose a new framework to interpret decision-making process of a temporal CNN classifier by **finding representative temporal patterns** detected by the networks.

[Relaxed Decision Region]

- Our **Relaxed Decision Region** framework detects a principal configuration where a target and relevant samples share learned representations by using configuration information.

Thank you!

Wonjoon Chang
SAILab, KAIST AI
one_jj@kaist.ac.kr
2024.03.21