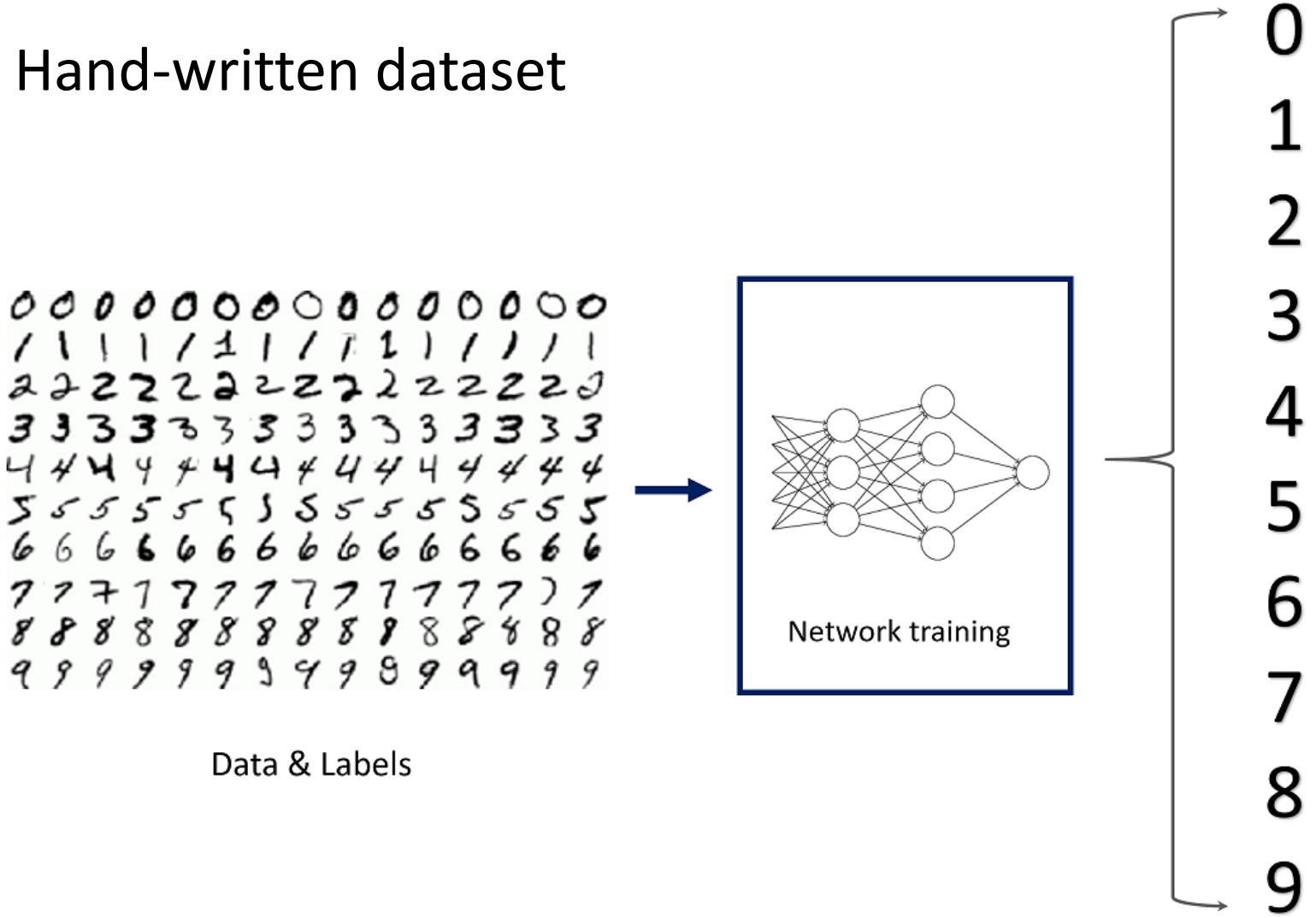
Prototypical network for Few-shot learning

Jake Snell, University of Toronto
Kevin Swersky, Twitter
Richard S. Zemel, University of Toronto

Presenter: Viet Lai

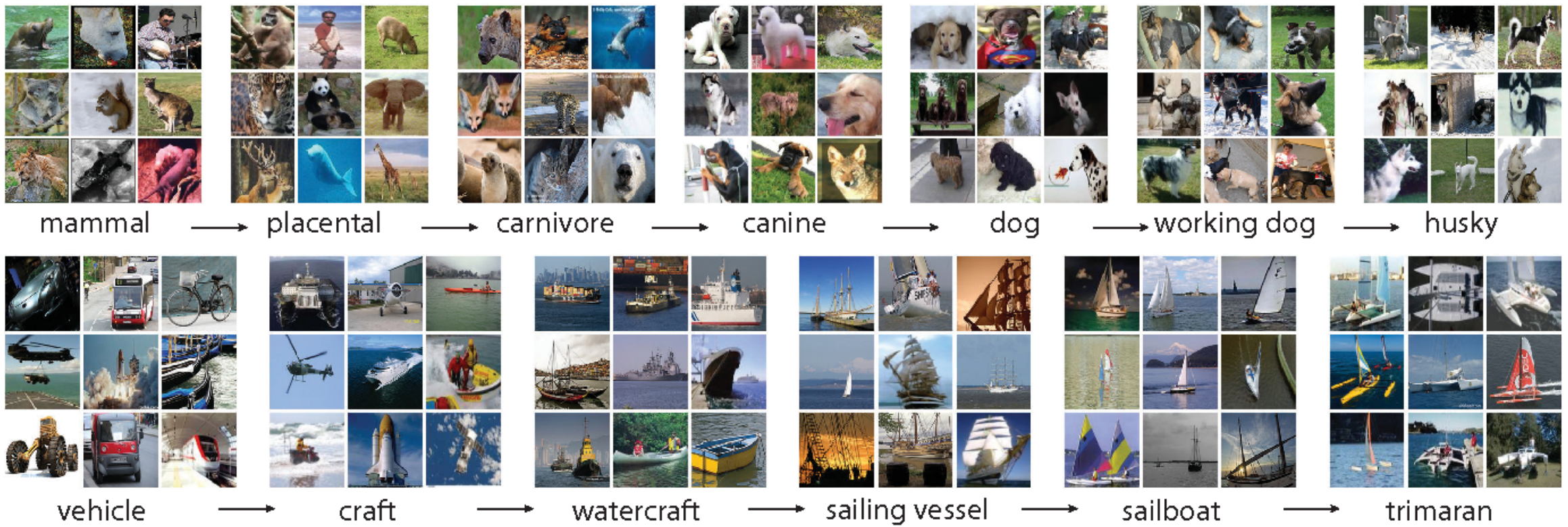
Supervised learning

- MNIST: Hand-written dataset

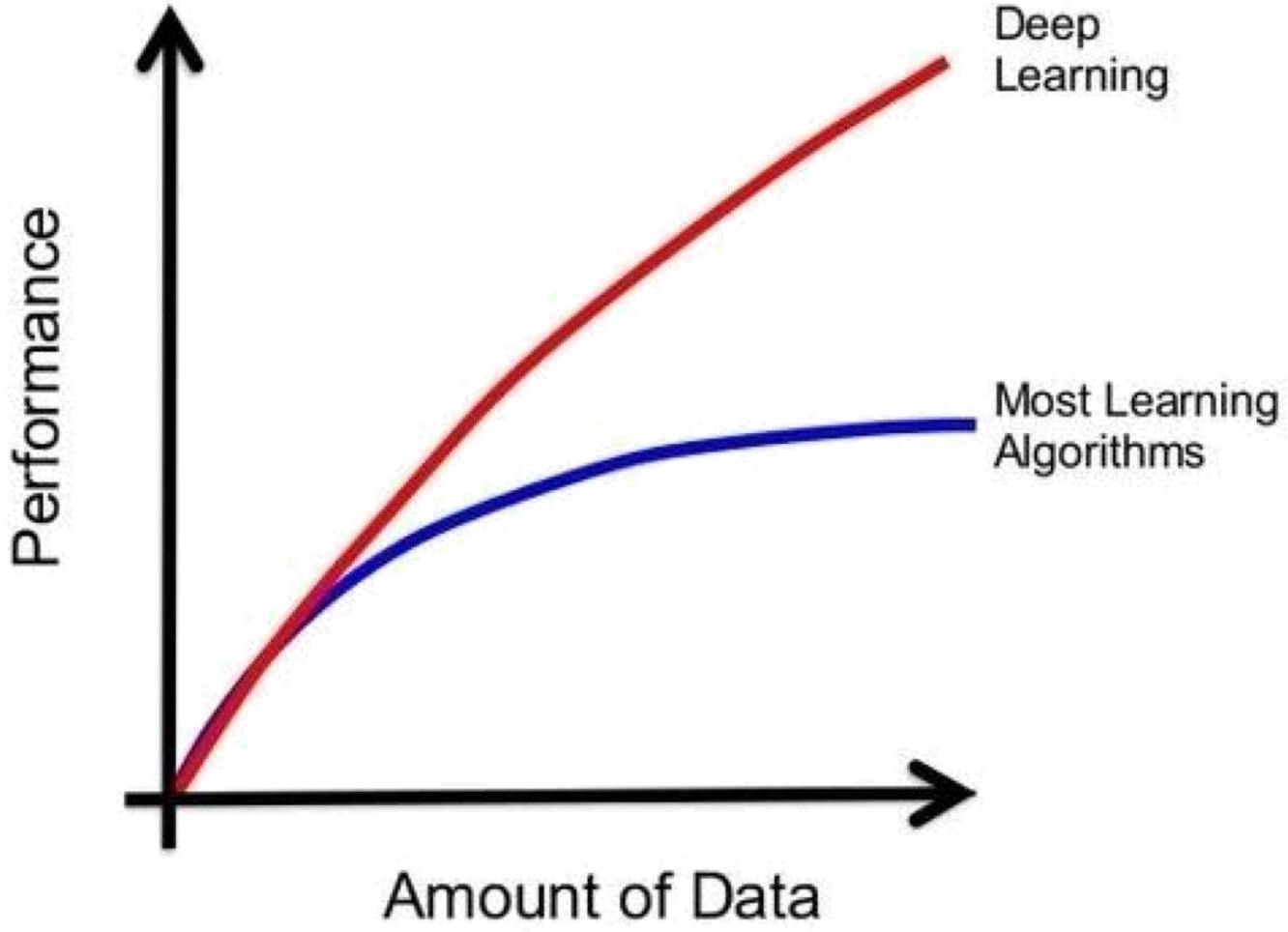


Supervised learning

- **ImageNet: Image database based on Wordnet**



Data is new kind of electricity



- MNIST:
  - 60,000 images
- ImageNet:
  - 14,197,122 image
- English-French
  - 2,000,000 pairs of sentence

Case 1: Space vehicle launch

- Super expensive
  - $100M
- SpaceX’s budget
  - 3 launches before bankrupt

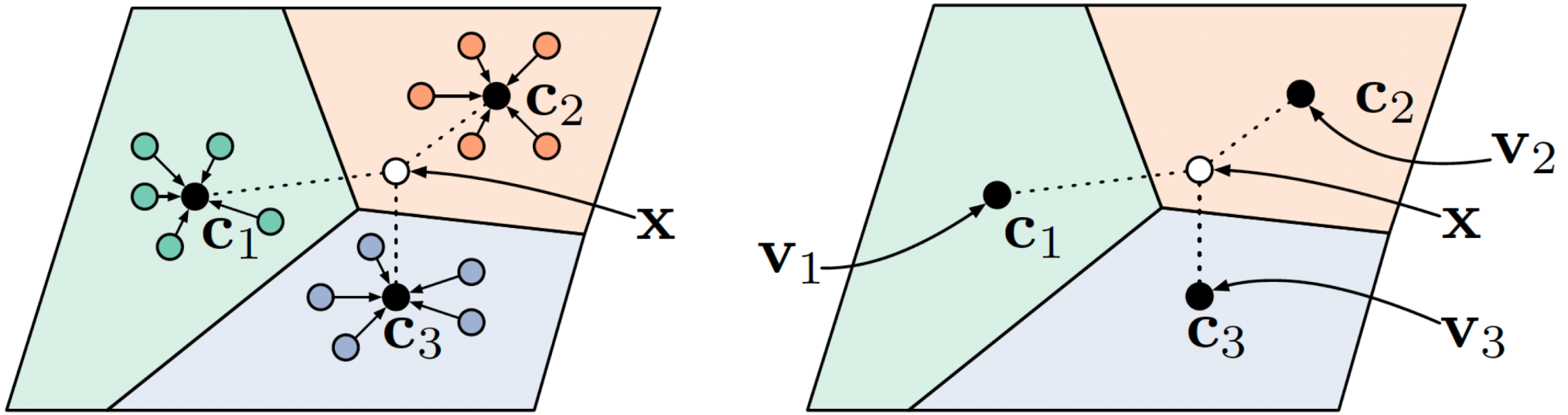
Case 2: Drug discovery

- Drug research is extremely expensive and long
- Using existing drug for new disease?
- How to confirm based on few cases?

Few-shot learning

- Supervised learning with a small amount of examples
- N-way K-shot
  - N classes
  - K samples for each class
  - K is very small
- Advantages
  - Reduce cost of annotation
  - Work in rare cases

Prototypical network



(a) Few-shot

(b) Zero-shot

Prototypical network

- Compute prototypes

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$$

- Compute the distribution among classes

$$p_{\boldsymbol{\phi}}(y = k \,|\, \mathbf{x}) = \frac{\exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_{k'}))}$$

- Compute loss

$$J(\boldsymbol{\phi}) = -\log p_{\boldsymbol{\phi}}(y = k \,|\, \mathbf{x})$$

As a linear model

- In case of Euclidean distance

$$-\|f_{\phi}(\mathbf{x}) - \mathbf{c}_k\|^2 = -f_{\phi}(\mathbf{x})^{\top} f_{\phi}(\mathbf{x}) + 2\mathbf{c}_k^{\top} f_{\phi}(\mathbf{x}) - \mathbf{c}_k^{\top} \mathbf{c}_k$$

$$2\mathbf{c}_k^{\top} f_{\phi}(\mathbf{x}) - \mathbf{c}_k^{\top} \mathbf{c}_k = \mathbf{w}_k^{\top} f_{\phi}(\mathbf{x}) + b_k, \text{ where } \mathbf{w}_k = 2\mathbf{c}_k \text{ and } b_k = -\mathbf{c}_k^{\top} \mathbf{c}_k$$

Algorithm

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.

Output: The loss J for a randomly generated training episode.

$V \leftarrow \textsc{RandomSample}(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode

for k in $\{1, \ldots, N_C\}$ **do**

$S_k \leftarrow \textsc{RandomSample}(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples

$Q_k \leftarrow \textsc{RandomSample}(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples

$$\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$$ ▷ Compute prototype from support examples

end for

$J \leftarrow 0$ ▷ Initialize loss

for k in $\{1, \ldots, N_C\}$ **do**

for $(\mathbf{x}, y)$ in Q_k **do**

$$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) \right]$$ ▷ Update loss

end for

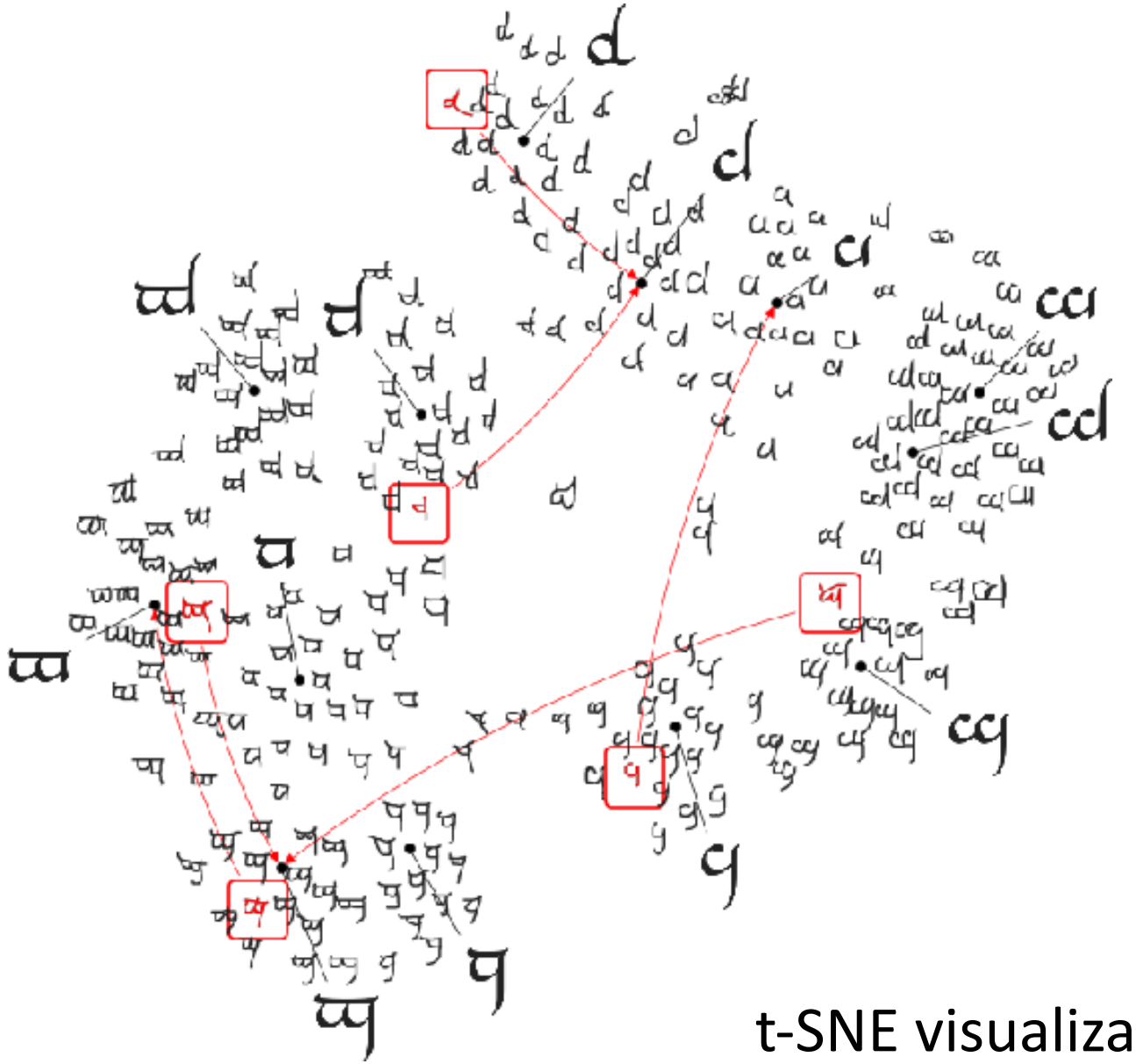
end for

Prototypical network vs Matching network

Prototypical network	Matching network
Euclidean distance	Cosine similarity
Mean of examples	Weighted sum of examples based on KNN
Simple	Complicated

Omniglot dataset

- 1623 written characters
- 50 alphabets
- 20 shots



t-SNE visualization

Prototypical network vs Matching network

Table 1: Few-shot classification accuracies on Omniglot.

Model	Dist.	Fine Tune	5-way Acc.		20-way Acc.	
			1-shot	5-shot	1-shot	5-shot
MATCHING NETWORKS [29]	Cosine	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETWORKS [29]	Cosine	Y	97.9%	98.7%	93.5%	98.7%
NEURAL STATISTICIAN [6]	-	N	98.1%	99.5%	93.2%	98.1%
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	**98.8%**	**99.7%**	**96.0%**	**98.9%**

miniImageNet dataset

- 60000 images
- 1000 classes
- 600 shots

Prototypical network vs Matching network

Table 2: Few-shot classification accuracies on *mini*ImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals. *Results reported by [22].

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot
BASELINE NEAREST NEIGHBORS*	Cosine	N	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
MATCHING NETWORKS [29]*	Cosine	N	$43.40 \pm 0.78\%$	$51.09 \pm 0.71\%$
MATCHING NETWORKS FCE [29]*	Cosine	N	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
META-LEARNER LSTM [22]*	-	N	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	$\mathbf{49.42 \pm 0.78\%}$	$\mathbf{68.20 \pm 0.66\%}$

Prototypical network vs Matching network

Table 3: Zero-shot classification accuracies on CUB-200.

Model	**Image Features**	**50-way Acc.** 0-shot
ALE [1]	Fisher	26.9%
SJE [2]	AlexNet	40.3%
SAMPLE CLUSTERING [17]	AlexNet	44.3%
SJE [2]	GoogLeNet	50.1%
DS-SJE [23]	GoogLeNet	50.4%
DA-SJE [23]	GoogLeNet	50.9%
PROTO. NETS (OURS)	GoogLeNet	**54.6%**

Question & Answer