



Boulder

# Advanced Topics; Few-shot Learning



[YouTube Playlist](#)

**Maziar Raissi**

**Assistant Professor**

Department of Applied Mathematics

University of Colorado Boulder

[maziar.raissi@colorado.edu](mailto:maziar.raissi@colorado.edu)

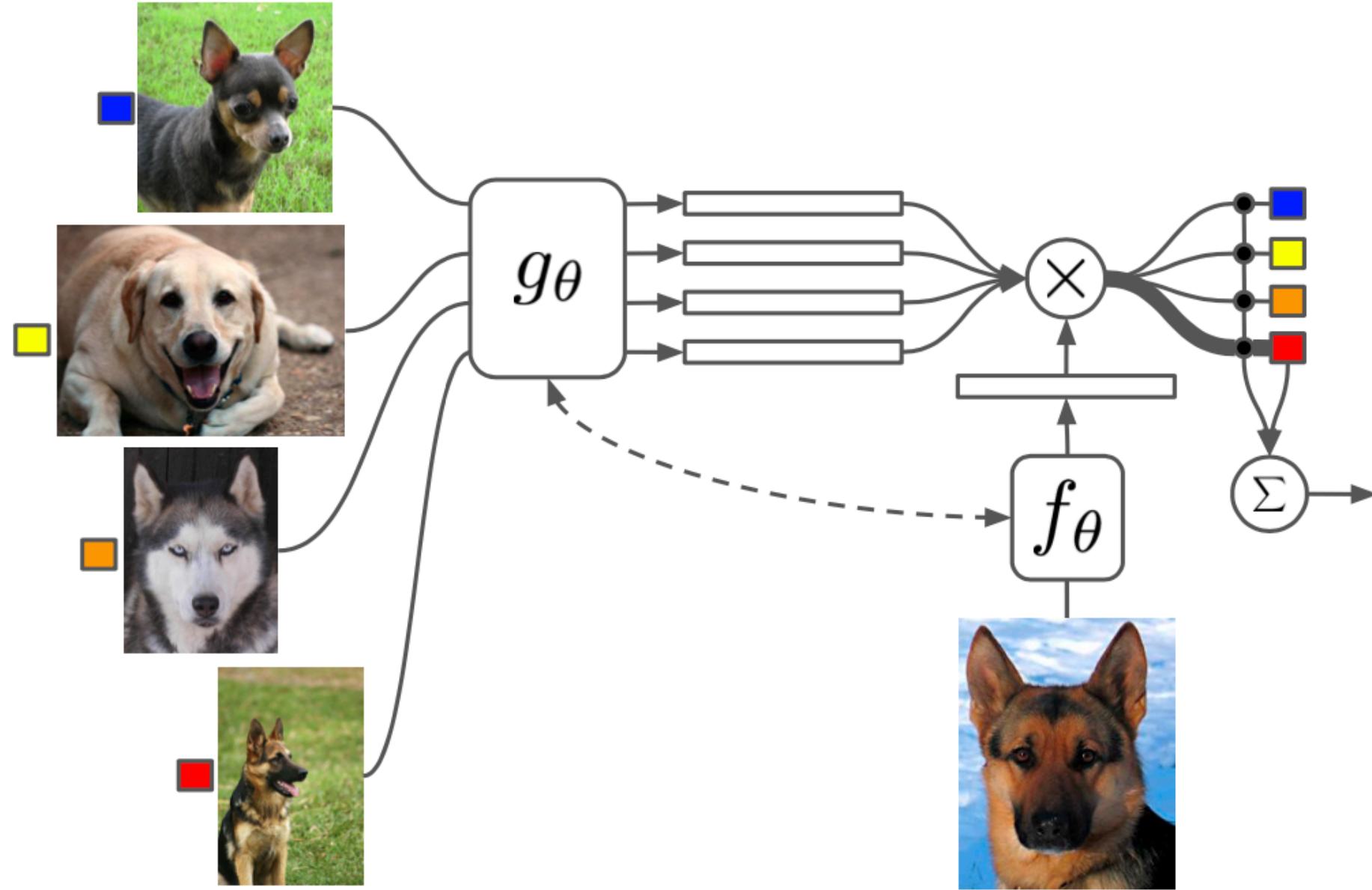


Boulder



[YouTube Video](#)

# Matching Networks for One Shot Learning



$S = \{(x_i, y_i)\}_{i=1}^m \rightarrow$  a (small) support set of  $m$  examples

$$P(\hat{y}|\hat{x}, S) = \sum_{i=1}^m a(\hat{x}, x_i) y_i$$

$$a(\hat{x}, x_i) = \frac{\exp(c(f(\hat{x}), g(x_i)))}{\sum_{j=1}^m \exp(c(f(\hat{x}), g(x_j)))}$$

$a$  → attention mechanism

$c$  → cosine similarity distance

$f$  &  $g$  → neural networks

**Training**

$$\theta = \arg \max_{\theta} E_{L \sim T} \left[ E_{S \sim L, B \sim L} \left[ \sum_{(x,y) \in B} \log P_{\theta}(y|x, S) \right] \right]$$

$T \rightarrow$  task

$L \sim T \rightarrow$  pick  $N$  classes

$S \sim L \rightarrow$  provide the model with  $k$  examples per each class

$B \sim L \rightarrow$  provide the model with  $k$  examples per each class

$B \rightarrow$  Batch

$S \rightarrow$  Support Set

$N$ -way  $k$ -shot learning task

**Full Context Embeddings (FCE)**

$$f(\hat{x}, S) = \text{attLSTM}(f'(\hat{x}), g(S), K)$$

$f'$  → a neural network (e.g., VGG or Inception)

$K$  → number of processing steps

The state after  $k$  processing steps is as follows:

$$\hat{h}_k, c_k = \text{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1})$$

$$h_k = \hat{h}_k + f'(\hat{x})$$

$$r_{k-1} = \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i)) g(x_i)$$

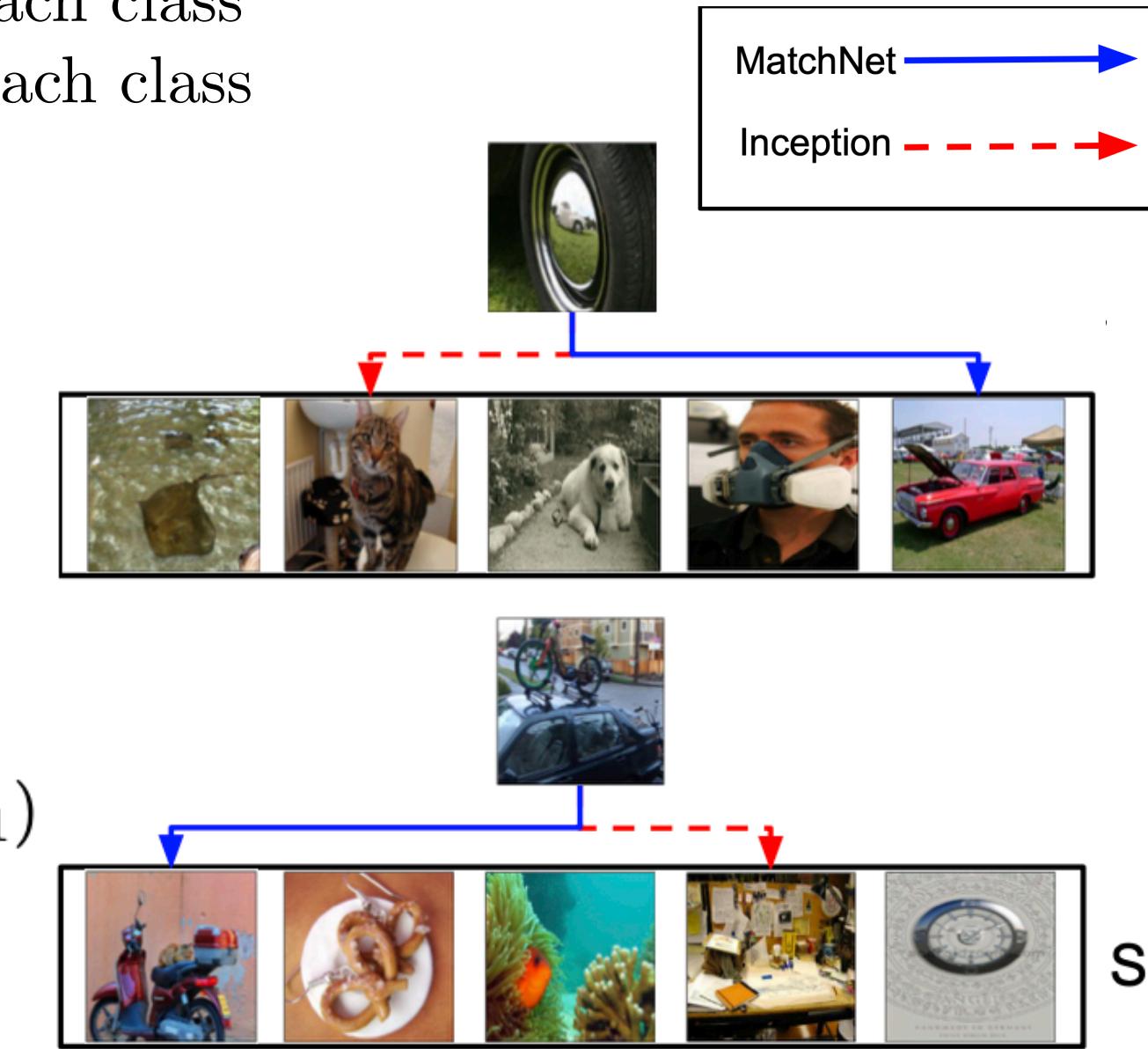
$$a(h_{k-1}, g(x_i)) = \text{softmax}(h_{k-1}^T g(x_i))$$

$$\text{attLSTM}(f'(\hat{x}), g(S), K) = h_K$$

$$g(x_i, S) = \vec{h}_i + \vec{c}_i + g'(x_i)$$

$$\vec{h}_i, \vec{c}_i = \text{LSTM}(g'(x_i), \vec{h}_{i-1}, \vec{c}_{i-1})$$

$$\vec{h}_i, \vec{c}_i = \text{LSTM}(g'(x_i), \vec{h}_{i+1}, \vec{c}_{i+1})$$



Example of two 5-way problem instance on ImageNet.

Results on *miniImageNet*.

Model	Matching Fn	Fine Tune	5-way Acc 1-shot	5-way Acc 5-shot
PIXELS	Cosine	N	23.0%	26.6%
BASELINE CLASSIFIER	Cosine	N	36.6%	46.0%
BASELINE CLASSIFIER	Cosine	Y	36.2%	52.2%
BASELINE CLASSIFIER	Softmax	Y	38.4%	51.2%
MATCHING NETS (OURS)	Cosine	N	41.2%	56.2%
MATCHING NETS (OURS)	Cosine	Y	42.4%	58.0%
MATCHING NETS (OURS)	Cosine (FCE)	N	44.2%	57.0%
MATCHING NETS (OURS)	Cosine (FCE)	Y	<b>46.6%</b>	<b>60.0%</b>



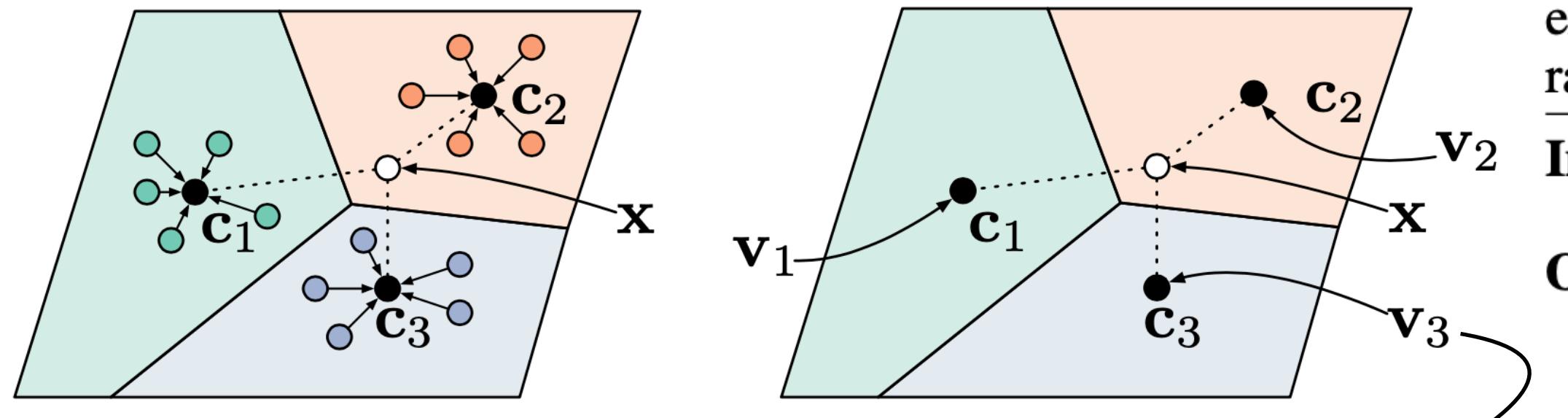
Boulder



[YouTube Video](#)

# Prototypical Networks for Few-shot Learning

Few-shot classification is a task in which a classifier must be adapted to accommodate new classes not seen in training, given only a few examples of each of these classes.



## Inference

$$S = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

↳ a small support set of  $N$  labeled examples

$x_i \in \mathbb{R}^D \rightarrow D$ -dimensional feature vector of an example

$y_i \in \{1, 2, \dots, K\} \rightarrow$  corresponding label

$S_k \rightarrow$  set of examples labeled with class  $k$

$c_k \rightarrow$  prototype ( $M$ -dimensional representation of each class)

$f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M \quad \phi \rightarrow$  learnable parameters

↳ embedding function

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i)$$

$$p_\phi(y = k | x) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{k'} \exp(-d(f_\phi(x), c_{k'}))}$$

$d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, +\infty)$   
↳ distance function

$$J(\phi) = -\log p_\phi(y = k | x) \rightarrow \text{Training}$$

**Algorithm 1** Training episode loss computation for Prototypical Networks.  $N$  is the number of examples in the training set,  $K$  is the number of classes in the training set,  $N_C \leq K$  is the number of classes per episode,  $N_S$  is the number of support examples per class,  $N_Q$  is the number of query examples per class.  $\text{RANDOMSAMPLE}(S, N)$  denotes a set of  $N$  elements chosen uniformly at random from set  $S$ , without replacement.

**Input:** Training set  $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , where each  $y_i \in \{1, \dots, K\}$ .  $\mathcal{D}_k$  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.

```

 $V \leftarrow \text{RANDOMSAMPLE}(\{1, \dots, K\}, N_C)$                                 ▷ Select class indices for episode
for  $k$  in  $\{1, \dots, N_C\}$  do                                                 
     $S_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k}, N_S)$                                 ▷ Select support examples
     $Q_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k} \setminus S_k, N_Q)$                                 ▷ Select query examples
     $\mathbf{c}_k \leftarrow \frac{1}{N_S} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$                                 ▷ Compute prototype from support examples
    end for
     $J \leftarrow 0$ 
    for  $k$  in  $\{1, \dots, N_C\}$  do
        for  $(\mathbf{x}, y)$  in  $Q_k$  do
             $J \leftarrow J + \frac{1}{N_C N_Q} \left[ d(f_\phi(\mathbf{x}), \mathbf{c}_k) + \log \sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_{k'})) \right]$           ▷ Update loss
        end for
    end for

```

The first term does not affect the softmax probabilities!

$$\begin{aligned} -\|f_\phi(\mathbf{x}) - \mathbf{c}_k\|^2 &= -f_\phi(\mathbf{x})^\top f_\phi(\mathbf{x}) + 2\mathbf{c}_k^\top f_\phi(\mathbf{x}) - \mathbf{c}_k^\top \mathbf{c}_k \\ 2\mathbf{c}_k^\top f_\phi(\mathbf{x}) - \mathbf{c}_k^\top \mathbf{c}_k &= \mathbf{w}_k^\top f_\phi(\mathbf{x}) + b_k \\ \mathbf{w}_k &= 2\mathbf{c}_k \text{ and } b_k = -\mathbf{c}_k^\top \mathbf{c}_k \end{aligned}$$

▷ Initialize loss

▷ Update loss

Model	Dist.	Fine Tune	miniImageNet Few-shot Classification	
			1-shot	5-shot
BASELINE NEAREST NEIGHBORS*	Cosine	N	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
MATCHING NETWORKS [32]*	Cosine	N	$43.40 \pm 0.78\%$	$51.09 \pm 0.71\%$
MATCHING NETWORKS FCE [32]*	Cosine	N	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
META-LEARNER LSTM [24]*	-	N	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML [9]	-	N	$48.70 \pm 1.84\%$	$63.15 \pm 0.91\%$
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$

The miniImageNet dataset consists of 60,000 color images of size  $84 \times 84$  divided into 100 classes with 600 examples each. The splits use a set of 100 classes, divided into 64 training, 16 validation, and 20 test classes.



Boulder

# Learning to Compare: Relation Network for Few-Shot Learning



[YouTube Video](#)

## Few-Shot Classifier Learning

Newly emerging (eg. new consumer devices) or rare (eg. rare animals) categories where numerous annotated images may simply never exist.

- training set – support set – testing set
- same label set for support and testing sets
- disjoint label set for training from support and testing sets
- $C$ -way  $K$ -shot:  $K$  labeled examples per each of  $C$  unique classes

## Episode-based Training

per each episode, randomly sample  $C$  classes from the training set  
 $\{l_1, \dots, l_C\} \rightarrow$  randomly sampled labels

$$\mathcal{S}_i = \{(x_k, y_k) : y_k = l_i\}_{k=1}^K \rightarrow K \text{ examples per each class } i = 1, \dots, C$$

$$\mathcal{S} = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_C \rightarrow \text{sample set (mimics the support set)}$$

$$m := |\mathcal{S}| = KC$$

$$\mathcal{Q} = \mathcal{Q}_1 \cup \dots \cup \mathcal{Q}_C \rightarrow \text{query set (mimics the test set)}$$

$$\mathcal{Q}_c \rightarrow \text{a fraction of the remainder of training samples having class } l_c$$

## K-Shot Learning

$$r_{ij} = g_\phi(\sum_{k \in \mathcal{S}_i} f_\varphi(x_k), f_\varphi(x_j)), i = 1, \dots, C, x_j \in \mathcal{Q}$$

$$\{r_{ij} \in (0, 1), i = 1, \dots, C\} \rightarrow C \text{ relation scores for query } x_j \in \mathcal{Q}$$

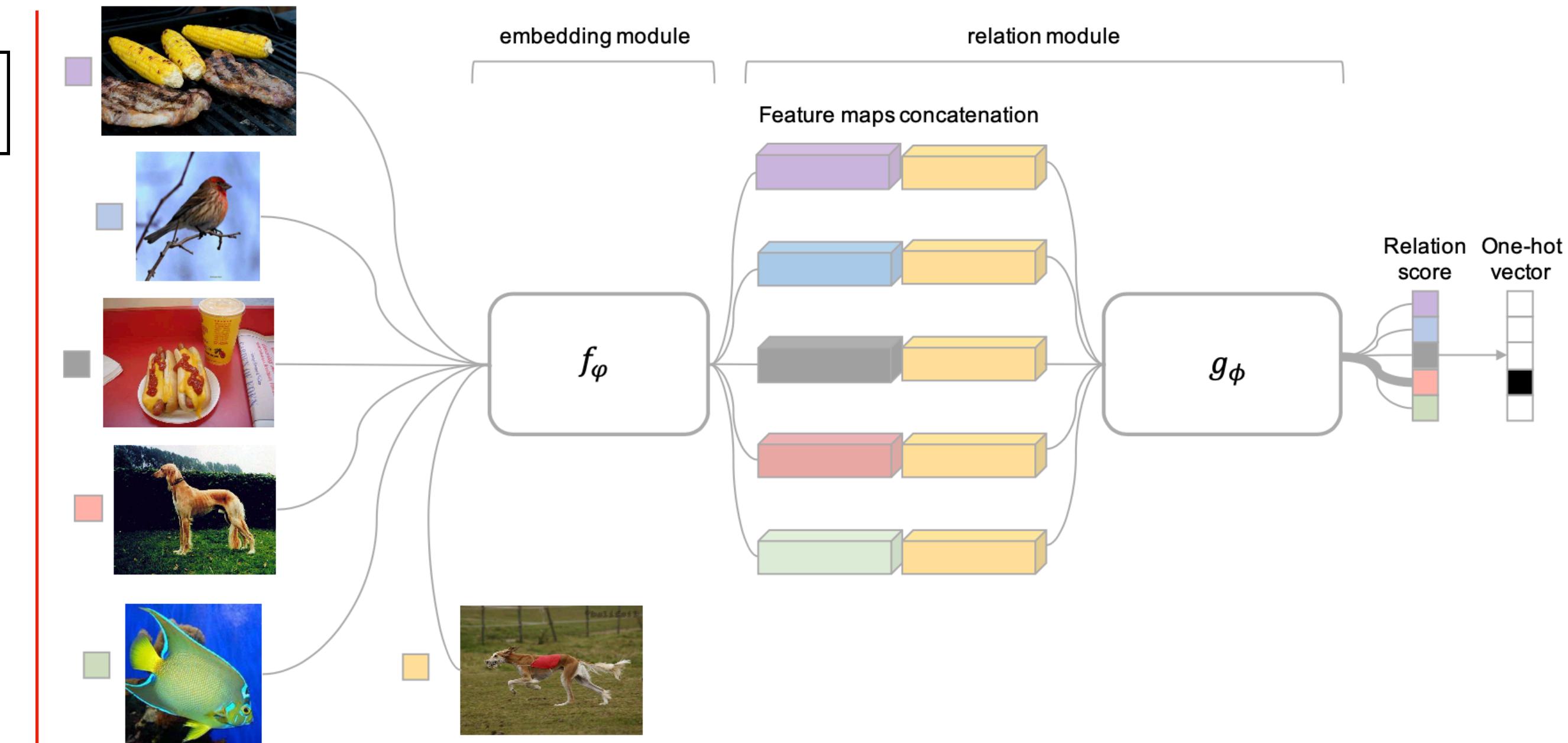
$f_\varphi \rightarrow$  embedding module

$g_\phi \rightarrow$  relation module

$$\arg \min_{\varphi, \phi} \sum_i \sum_j (r_{ij} - \mathbb{1}(y_j = l_i))^2$$

## Zero-Shot

$$r_{ij} = g_\phi(f_\varphi(v_i), f_\varphi(x_j)), v_i \rightarrow \text{semantic class embedding vector}$$



Model	Fine Tune	5-way Acc.		20-way Acc.	
		1-shot	5-shot	1-shot	5-shot
MANN [32]	N	82.8%	94.9%	-	-
CONVOLUTIONAL SIAMESE NETS [20]	N	96.7%	98.4%	88.0%	96.5%
CONVOLUTIONAL SIAMESE NETS [20]	Y	97.3%	98.4%	88.1%	97.0%
MATCHING NETS [39]	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETS [39]	Y	97.9%	98.7%	93.5%	98.7%
SIAMESE NETS WITH MEMORY [18]	N	98.4%	99.6%	95.0%	98.6%
NEURAL STATISTICIAN [8]	N	98.1%	99.5%	93.2%	98.1%
META NETS [27]	N	99.0%	-	97.0%	-
PROTOTYPICAL NETS [36]	N	98.8%	99.7%	96.0%	98.9%
MAML [10]	Y	98.7 ± 0.4%	99.9 ± 0.1%	95.8 ± 0.3%	98.9 ± 0.2%
<b>RELATION NET</b>	N	<b>99.6 ± 0.2%</b>	<b>99.8 ± 0.1%</b>	<b>97.6 ± 0.2%</b>	<b>99.1 ± 0.1%</b>

Omniglot few-shot classification.



Boulder



# Questions?

[YouTube Playlist](#)

---