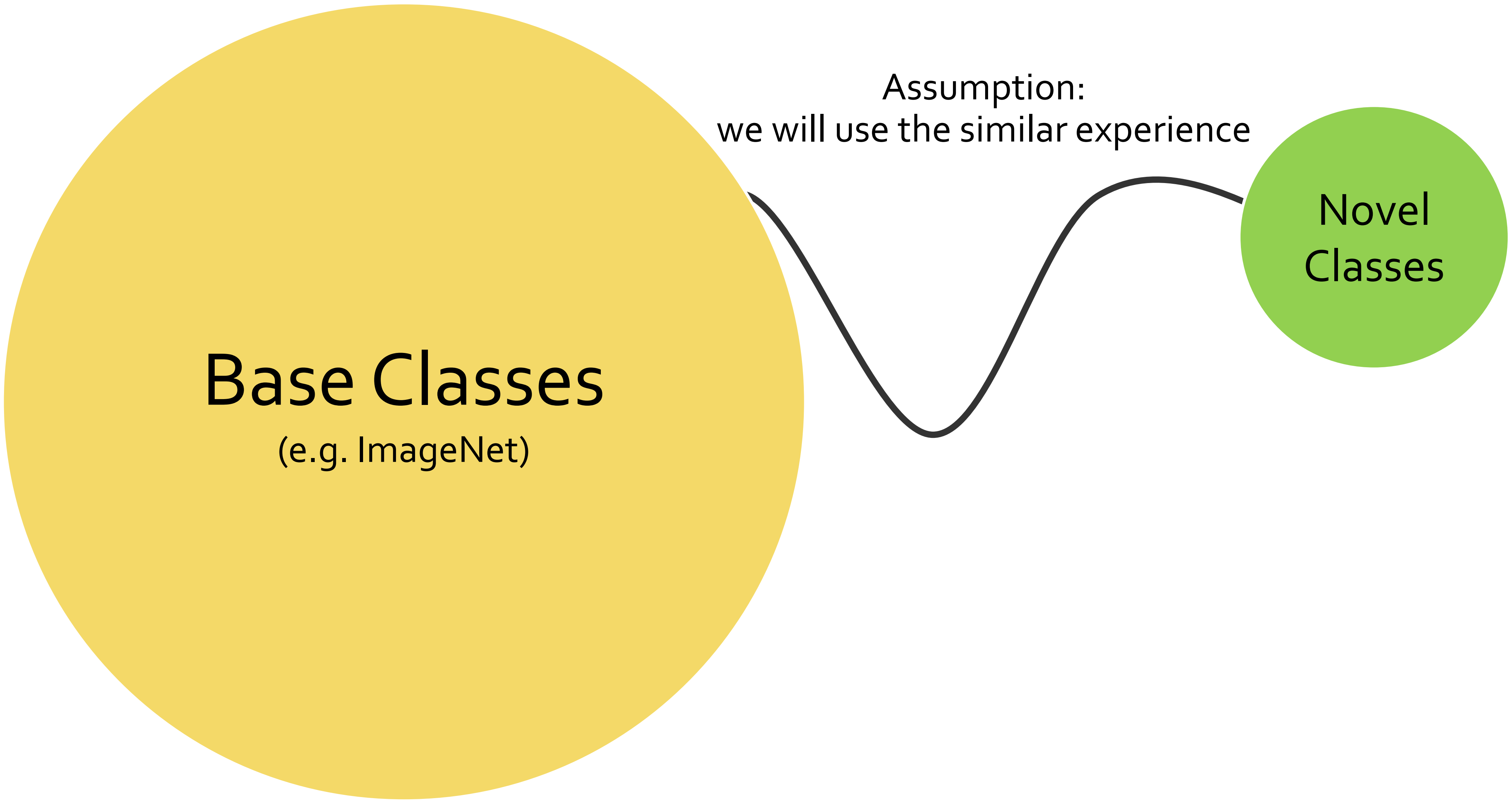
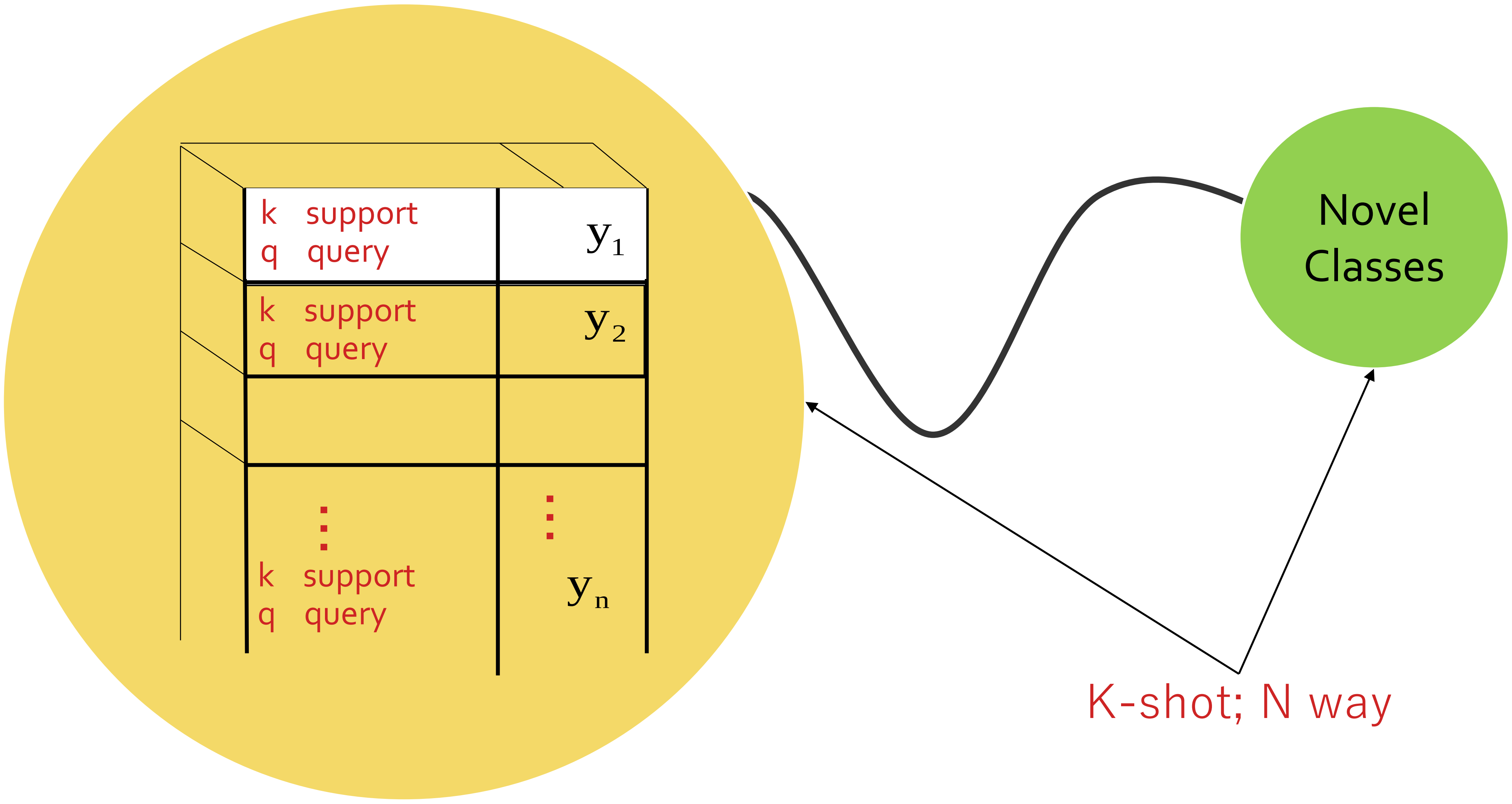
advances in few-shot learning

Arman Afrasiyabi
Université Laval

few-shot learning: is supervised transferring knowledge

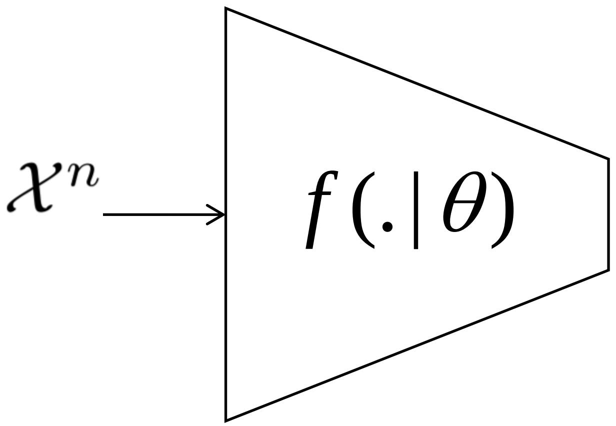


few-shot learning: is supervised transferring knowledge

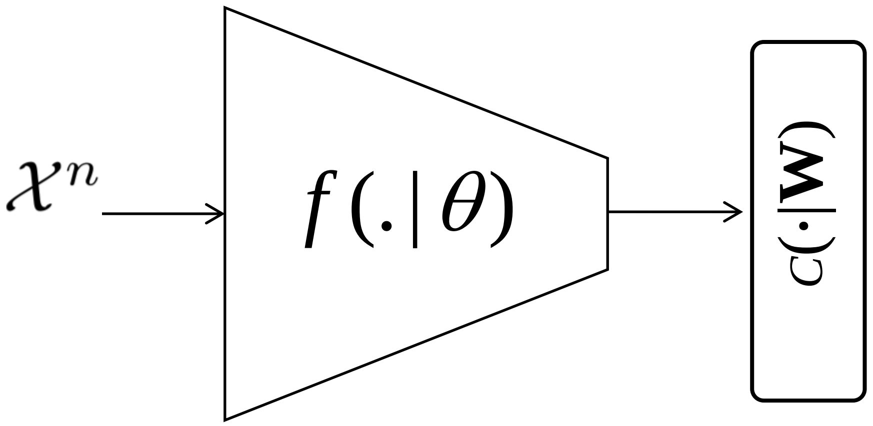


few-shot frameworks

meta learning



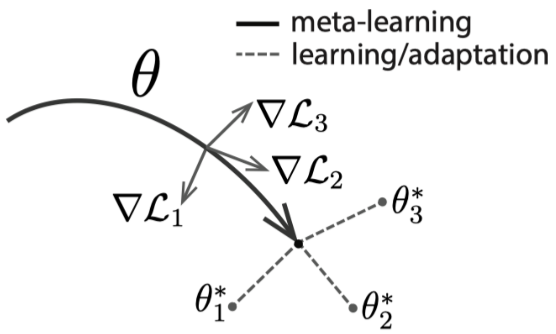
standard transfer learning



meta learning- initialization based

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

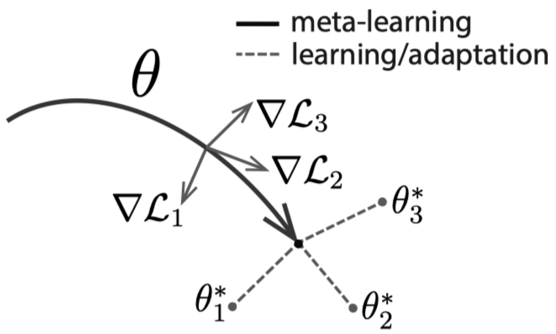


MAML
Finn et al., 2017

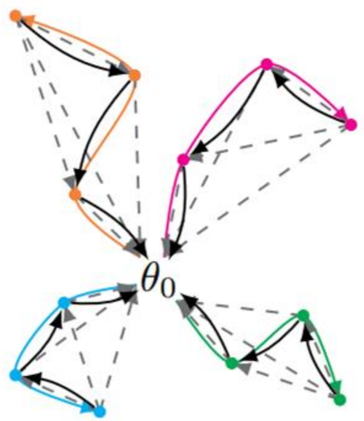
meta learning- initialization based

MAML problems:

- Vanishing/exploding meta-gradients
- Computationally costly
- Hard to make work for more than ~10 adaptation steps
- Compress all information into a single initial point



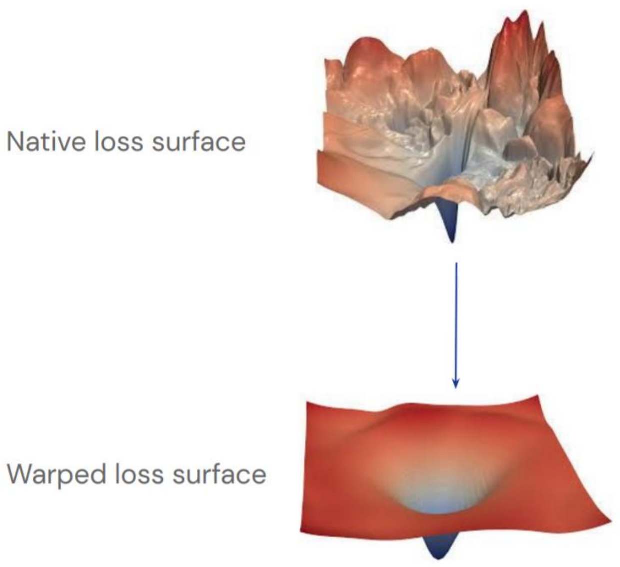
MAML
Finn et al., 2017

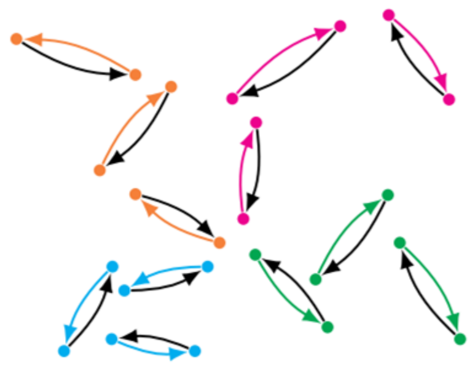


- colors denote tasks
- dashed line denote backpropagation
- solid line denote optimizer parameters gradient w.r.t. one of tasks

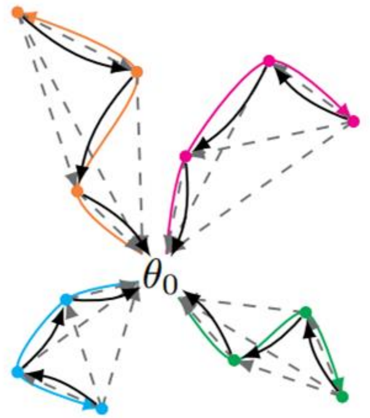
meta learning- initialization based

Warped Gradient Descent





WapedGrad
Flennerhag et al., ICLR 2020



MAML
Finn et al., 2017

- colors denote tasks
- dashed line denote backpropagation
- solid line denote optimizer parameters gradient w.r.t. one of tasks

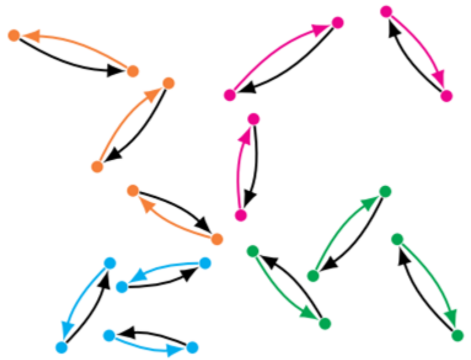
http://flennerhag.com/research/pres_warped_gradient_descent_neurips.pdf

meta learning- initialization based

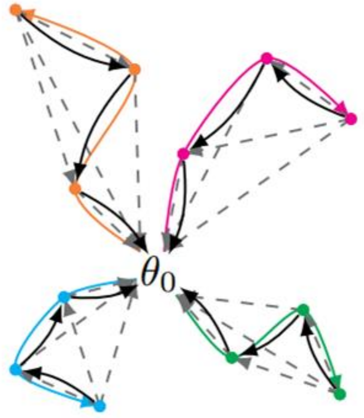
- WarpGrad learns to precondition gradients over a search space

- Stochastic gradient descent defines an empirical parameter distribution over this space

- Meta-learn warp-parameters to yield steepest directions of descent over search space



WapedGrad
Flennerhag et al., ICLR 2020



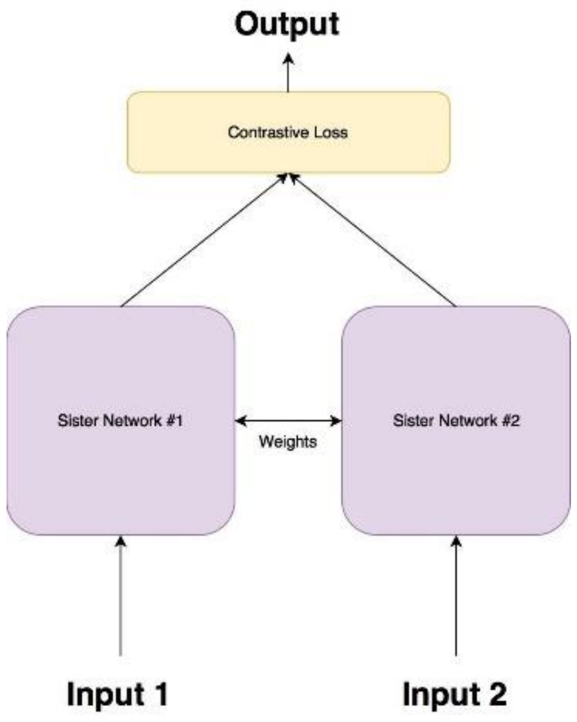
MAML
Finn et al., 2017

- colors denote tasks
- dashed line denote backpropagation
- solid line denote optimizer parameters gradient w.r.t. one of tasks

http://flennerhag.com/research/pres_warped_gradient_descent_neurips.pdf

meta learning- distance based

$$\delta(x^{(i)}, x^{(j)}) = \begin{cases} min \parallel \mathrm{f}\left(x^{(i)}\right) - \mathrm{f}\left(x^{(j)}\right) \parallel, i = j \\ max \parallel \mathrm{f}\left(x^{(i)}\right) - \mathrm{f}\left(x^{(j)}\right) \parallel, i \neq j \end{cases}$$



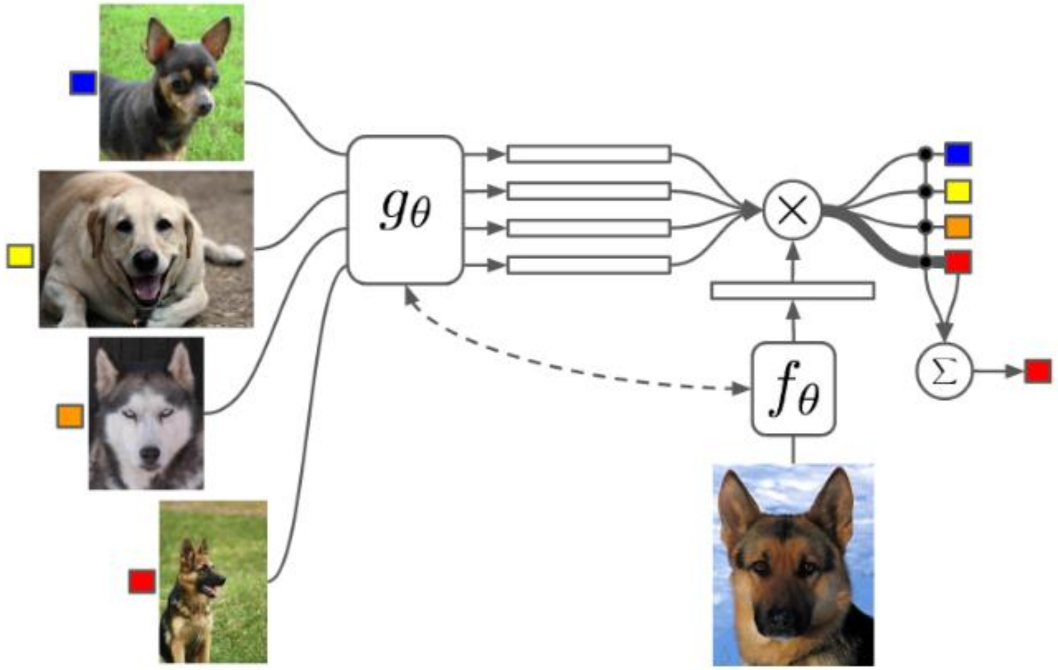
Siamese Net.
Koch et al. (2015)

meta learning- distance based

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

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

features derived from a CNN

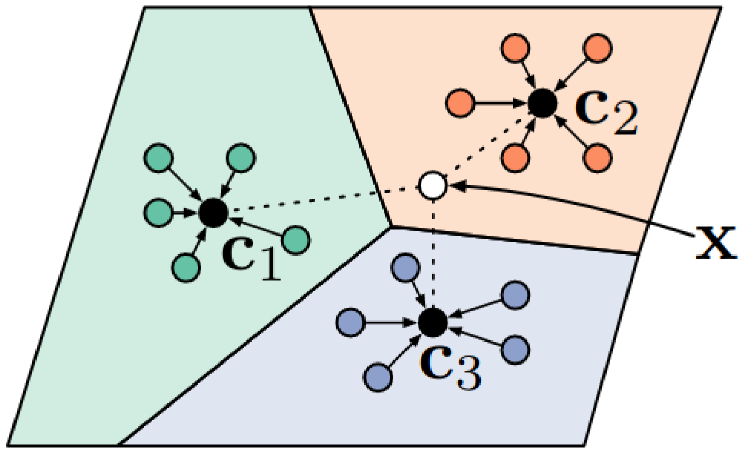


Matching Net.
Vinyals et al. (NeurIPS 2017)

meta learning- distance based

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

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

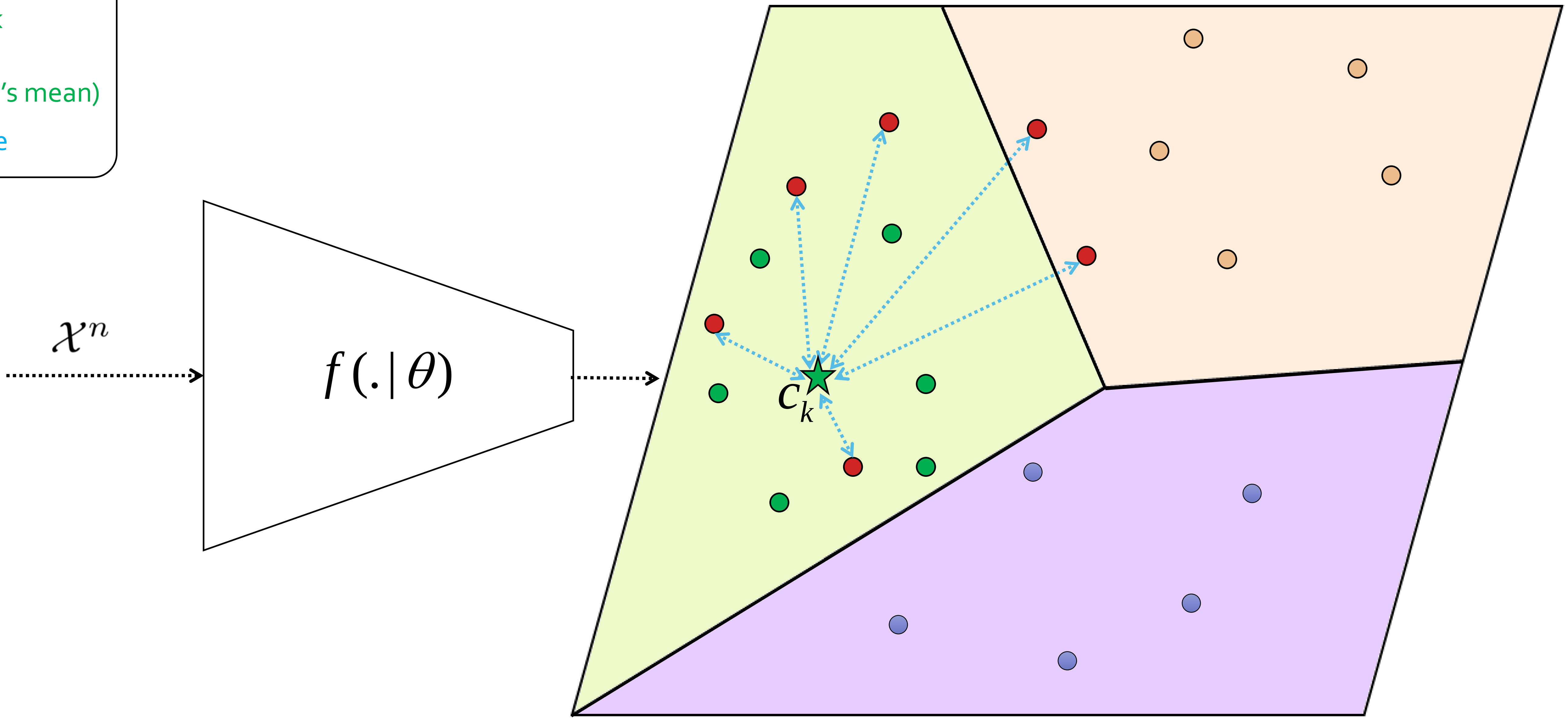


Proto. Net.
Snell et al. (NeurIPS 2017)

Prototypical Network [Snell et al. NIPS2017)]

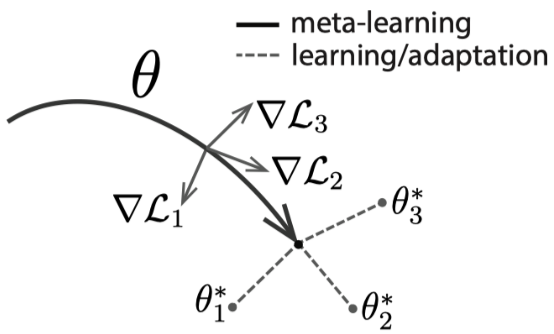
Distance based approach (e.g.)

- other classes support set
- support set class k
- query set class k
- proto (support set's mean)
- Euclidean distance



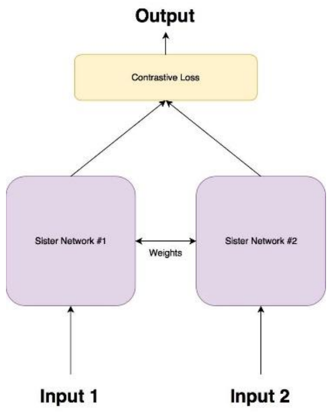
meta learning

1) initialization based

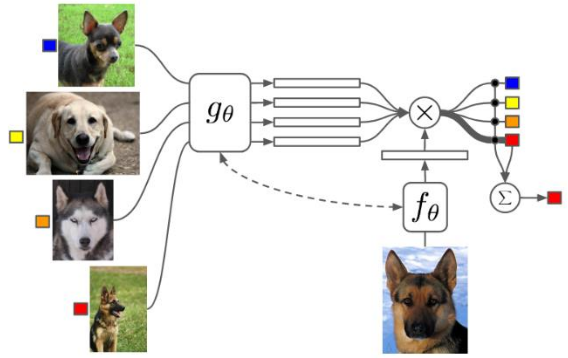


MAML
Finn et al., 2017

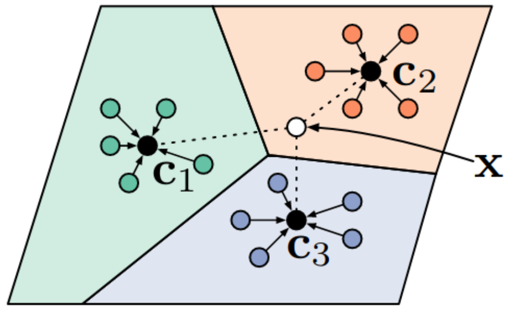
2) distance based



Siamese Net.
Koch et al. (2015)



Matching Net.
Vinyals et al. (2017)



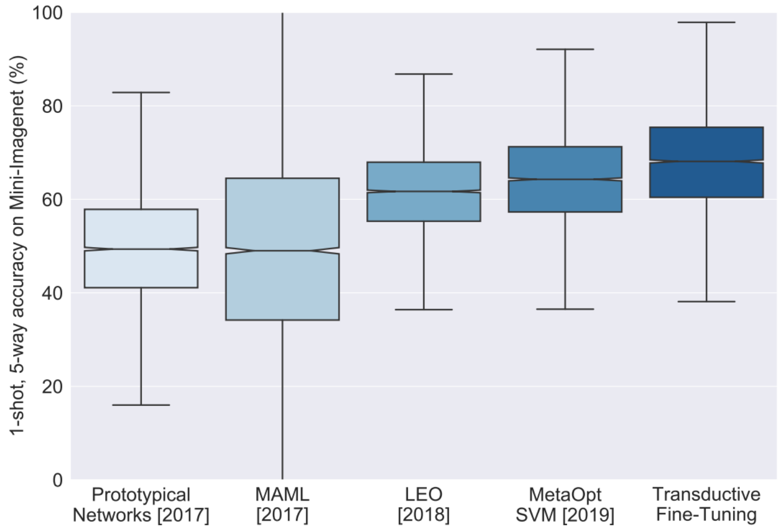
Proto. Net.
snell et al. (2017)

meta learning- transductive learning based

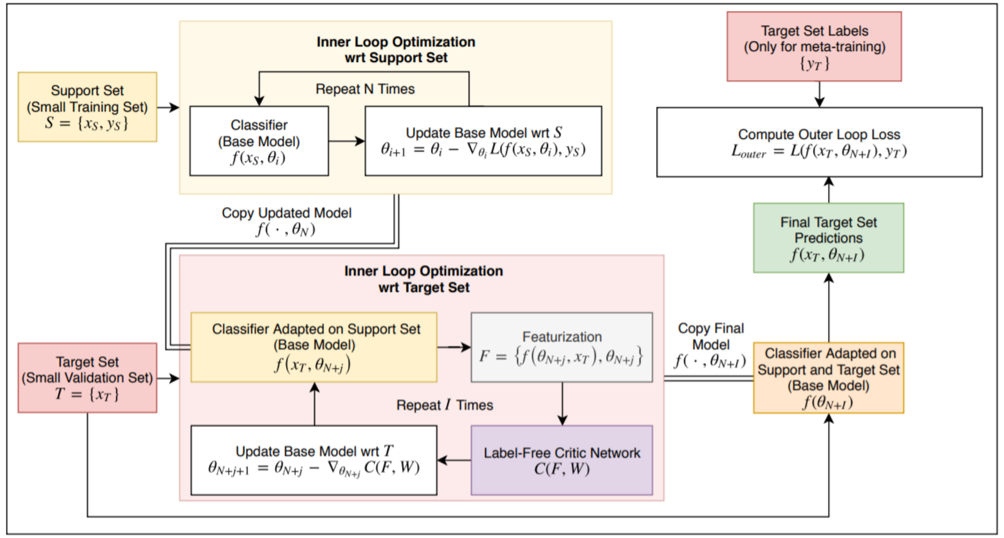
use information from the test example x to restrict the hypothesis space while searching for the classifier at test time (Joachims, 1999; Zhou et al., 2004; Vapnik, 2013)

$$\Theta^* = \arg\min_{\Theta} \frac{1}{N_s} \sum_{(x,y)\in\mathcal{D}_s} -\log p_\Theta\,(y \mid x) + \frac{1}{N_q} \sum_{(x,y)\in\mathcal{D}_q} \mathbb{H}(p_\Theta(\cdot \mid x))$$

low Shannon Entropy for regularization



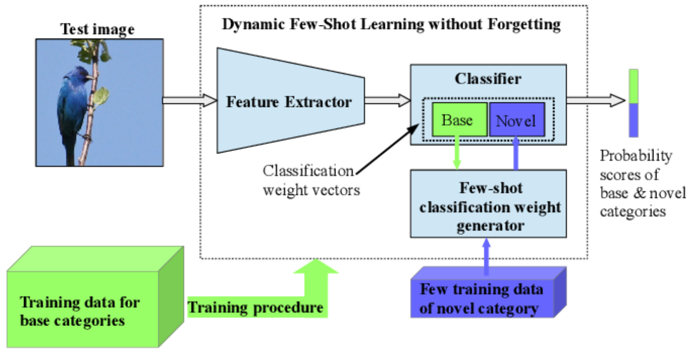
Transductive Fine-Tuning
Dhillon et al., ICLR 2020



Self-Critique and Adapt
Antoniou et al., NeurIPS 2019

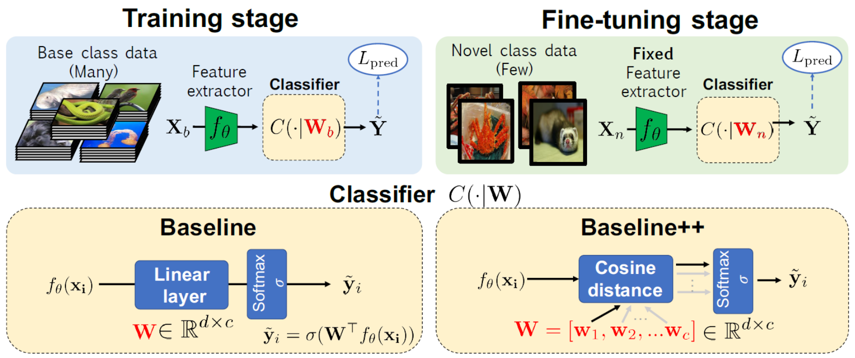
standard transfer learning

1) specific problem based



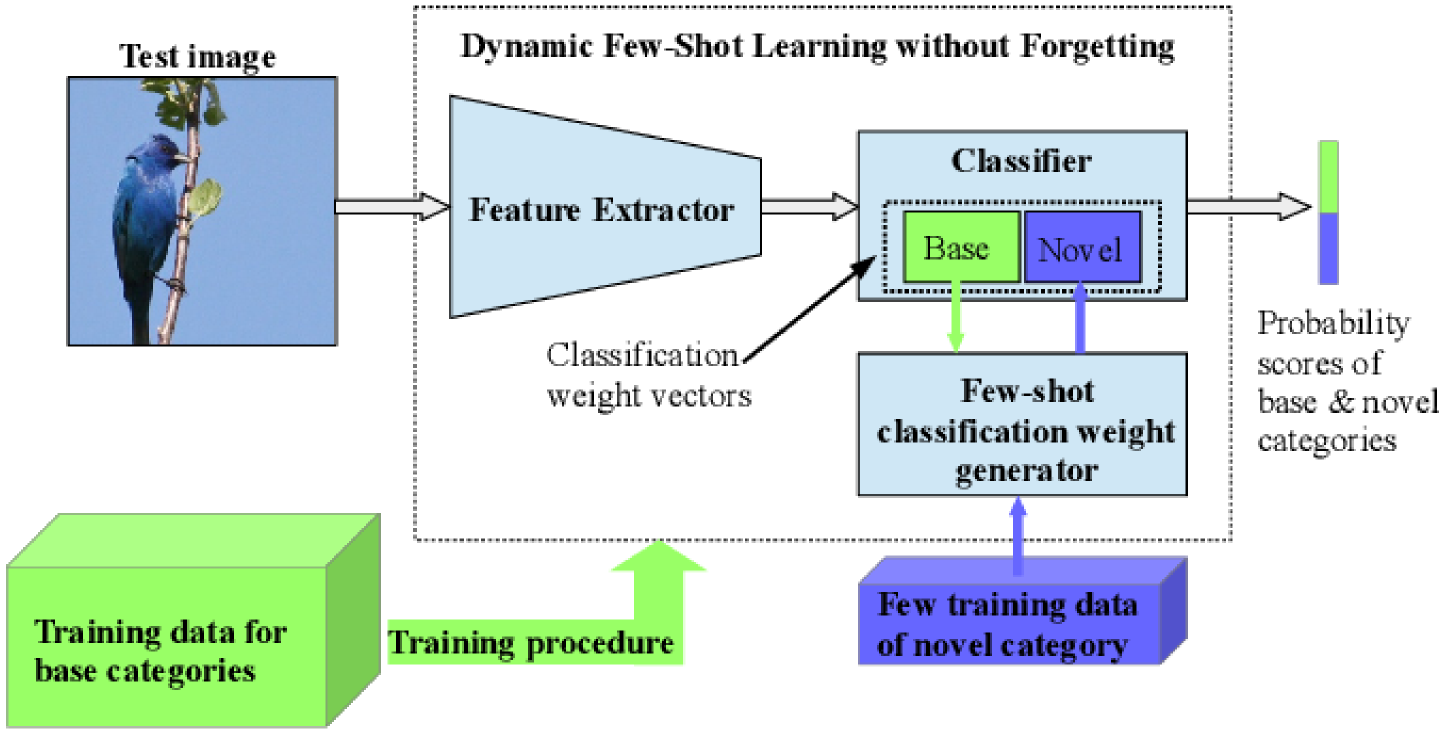
forgetting
Gidaris et al., 2018

2) "metric-learning" based



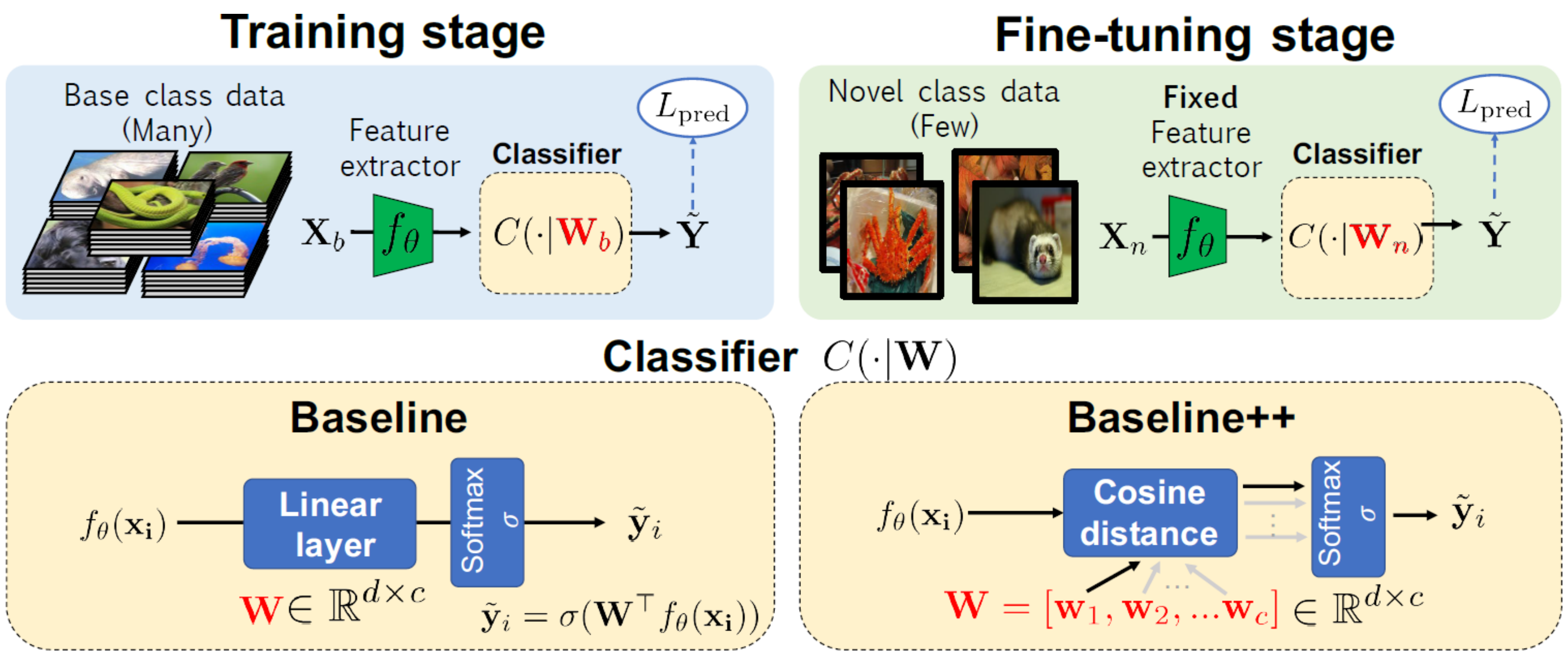
baseline++
Chen et al., 2019

standard transfer learning- dealing with forgetting



forgetting
Gidaris et al., CVPR2018

standard transfer learning –metric learning based

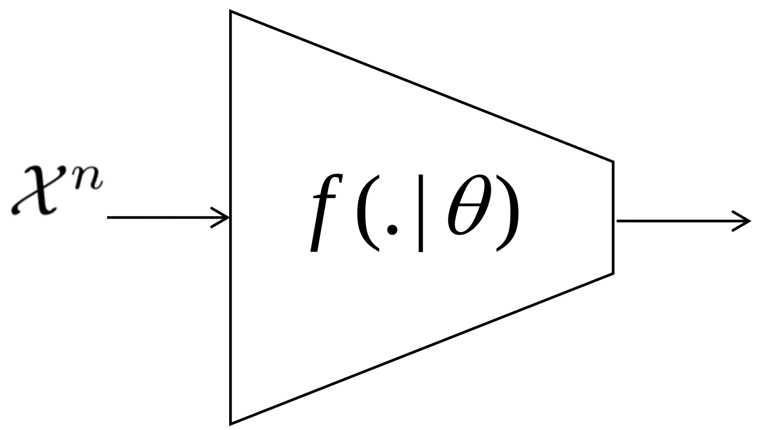


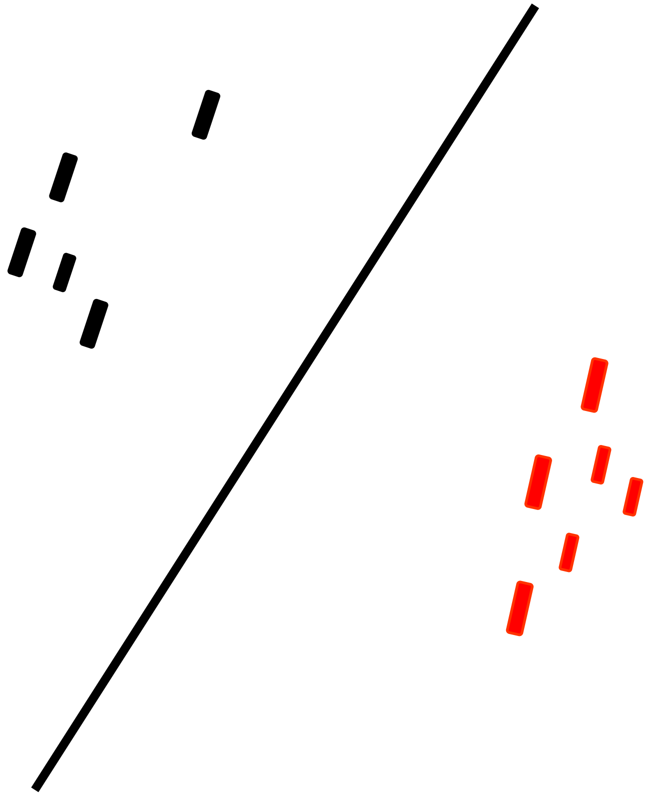
Baseline++
Chen et al., ICLR 2019

Our study for Few-shot Image Classification

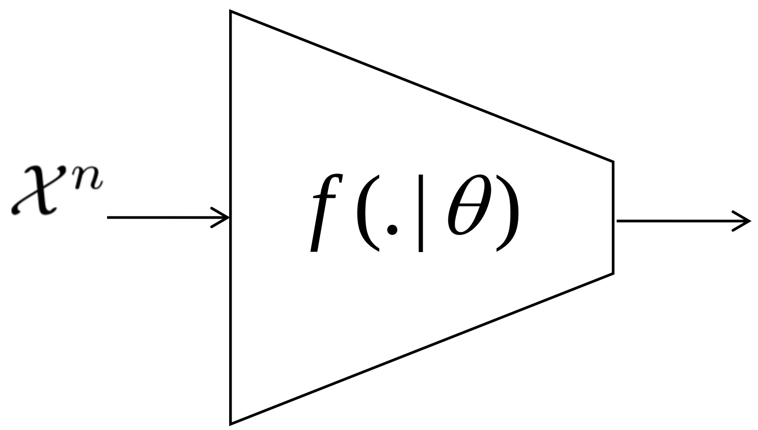
Arman Afrasiyabi, Jean-François Lalonde, Christian Gagné
Université Laval

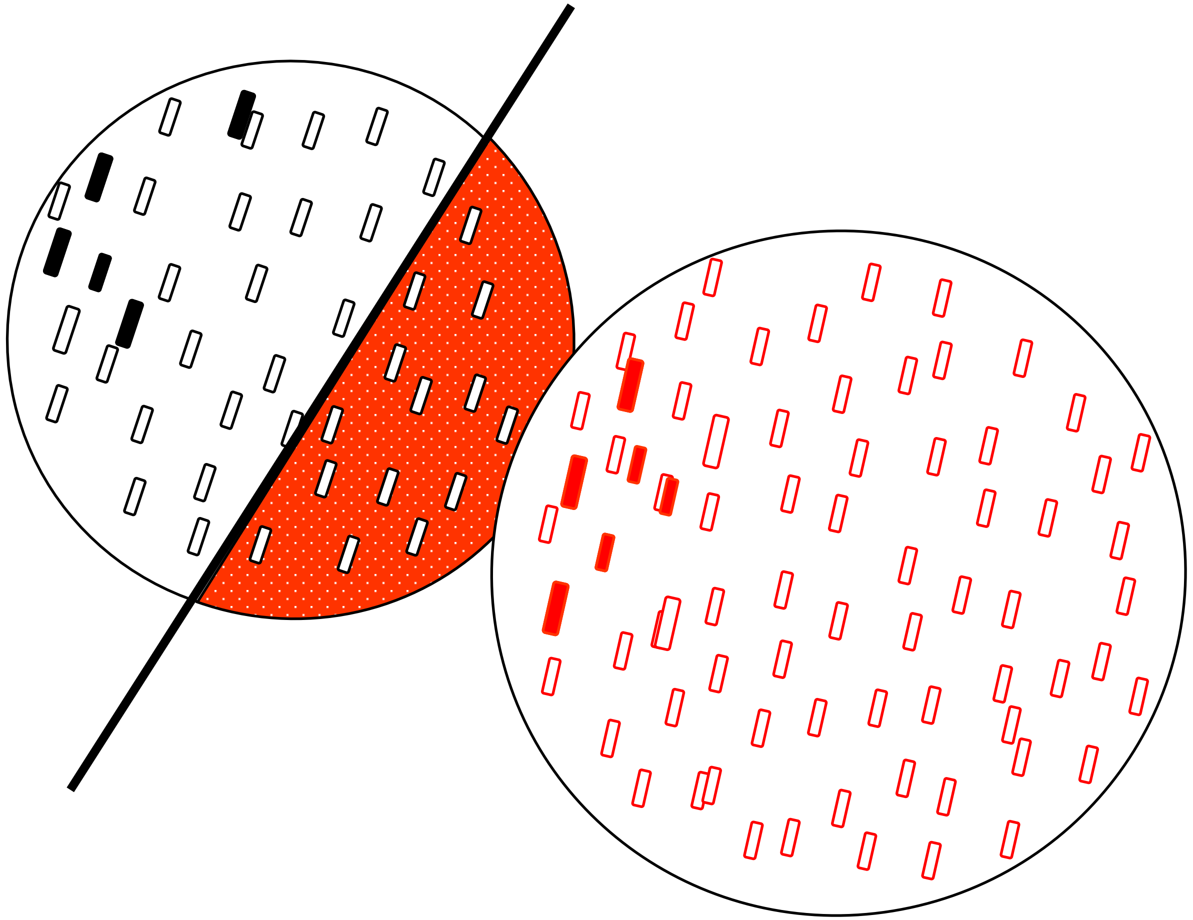
few-shot problem: sampling





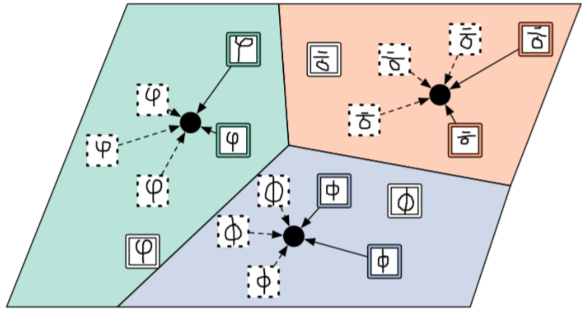
few-shot problem: sampling



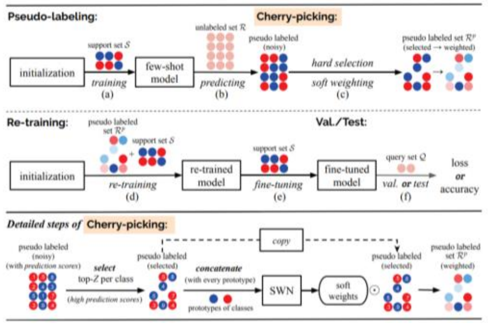


data augmentation based

1) semi-supervised learning

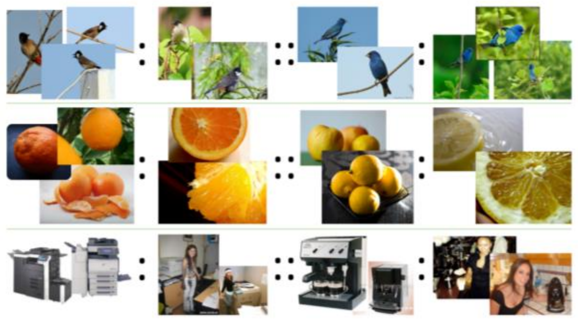


meta semi-supervised
Ren et al. 2018

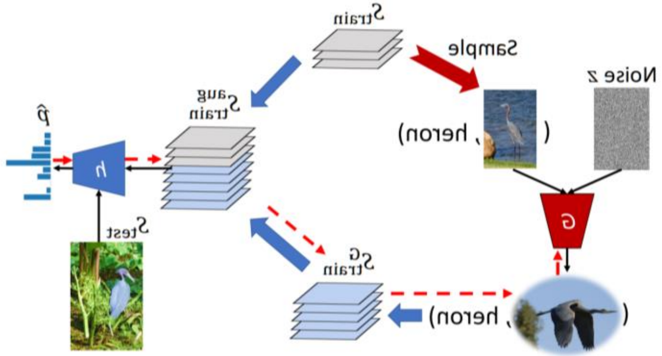


self-train semi-supervised
Li et al. 2019

2) generative or mapping models



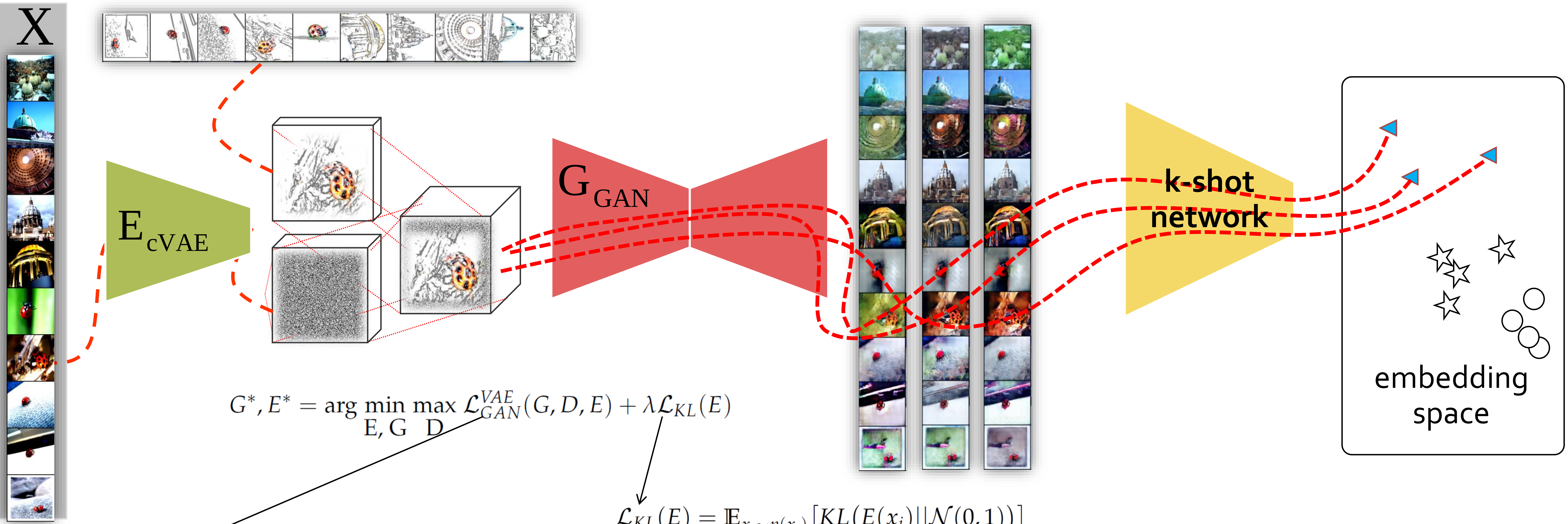
FH
Hariharan et al. 2017



Imaginary data
Wang et al. 2018

Extending k-shot to k-plus-shot Learning

Conditional Variational Autoencoder GAN



$$G^*, E^* = \arg\min_{E,G}\max_{D} \mathcal{L}^{VAE}_{GAN}(G,D,E) + \lambda\mathcal{L}_{KL}(E)$$

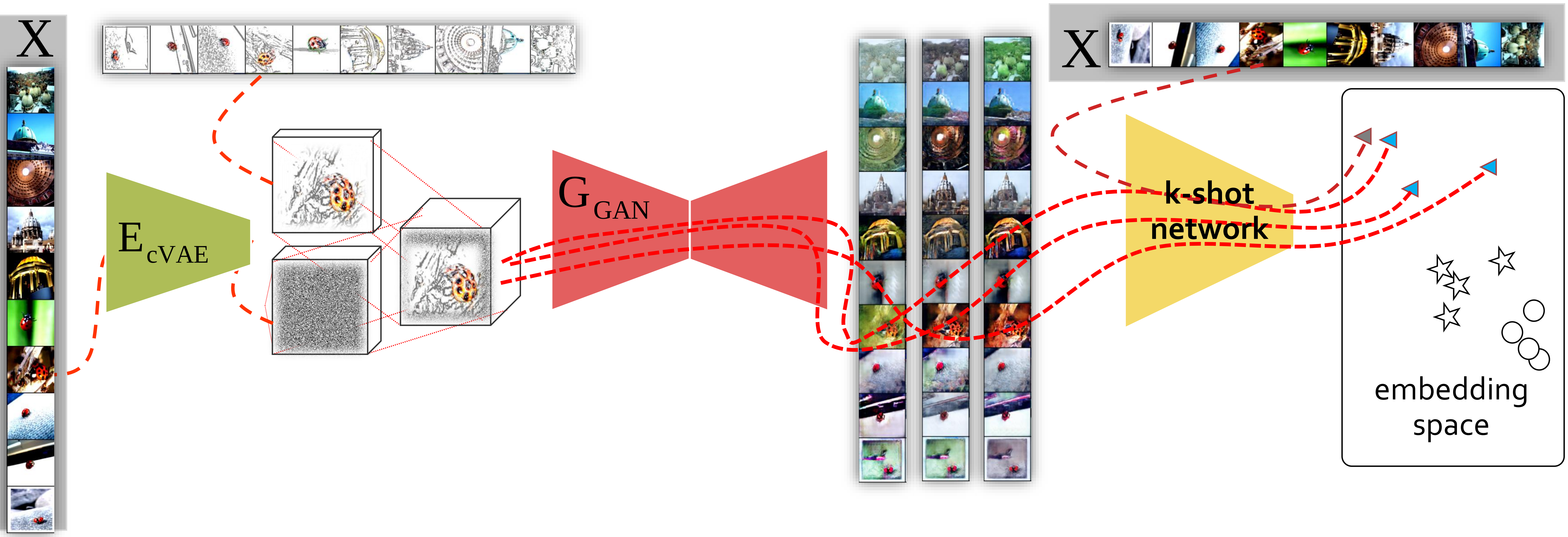
$$\mathcal{L}_{KL}(E) = \mathbb{E}_{x_i\sim p(x_i)}[KL(E(x_i)||\mathcal{N}(0,1))]$$

$$\mathcal{L}^{VAE}_{GAN} = \mathbb{E}_{x_i,x_i^c\sim p(x_i,x_i^c)}[log(D(x_i,x_i^c))] + \mathbb{E}_{x_i,x_i^c\sim p(x_i,x_i^c),z\sim E(x_i)}[log(1-D(x_i^c,G(x_i^c,z)))]$$

Consecutives Version

Extending k-shot to k-plus-shot Learning

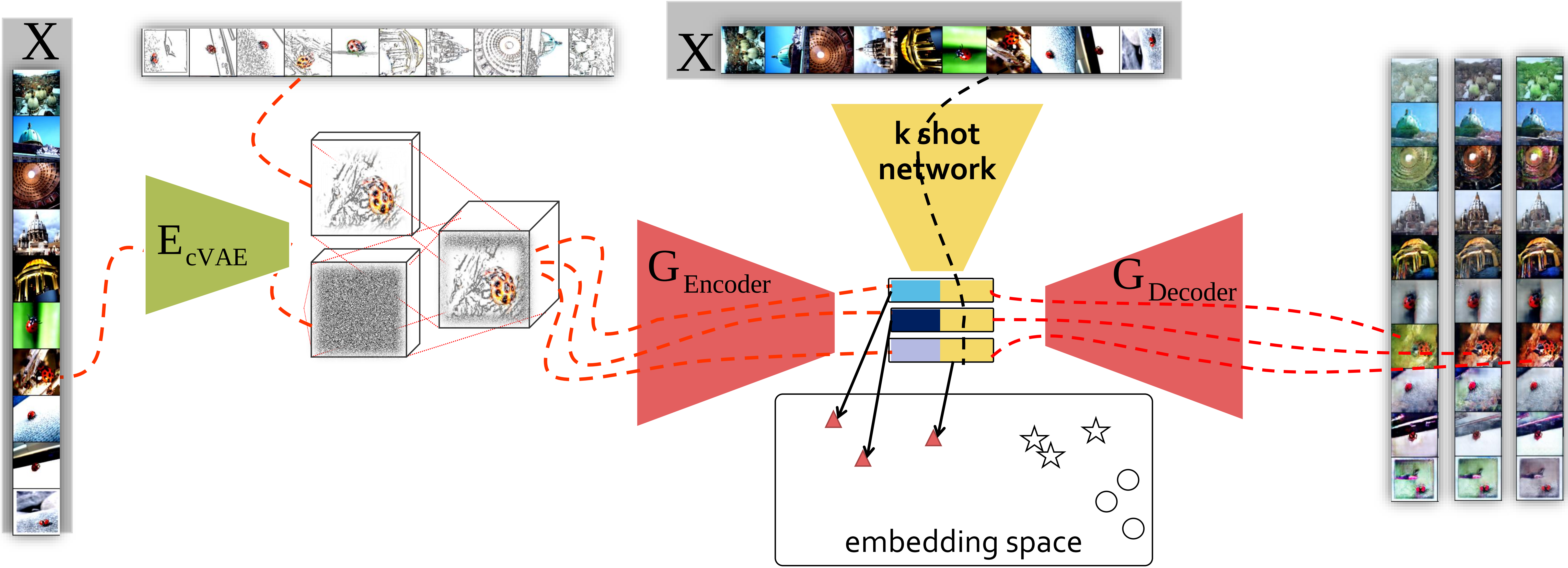
Conditional Variational Autoencoder GAN



Consecutives Version

Extending k-shot to k-plus-shot Learning

Conditional Variational Autoencoder GAN



Lateral Version

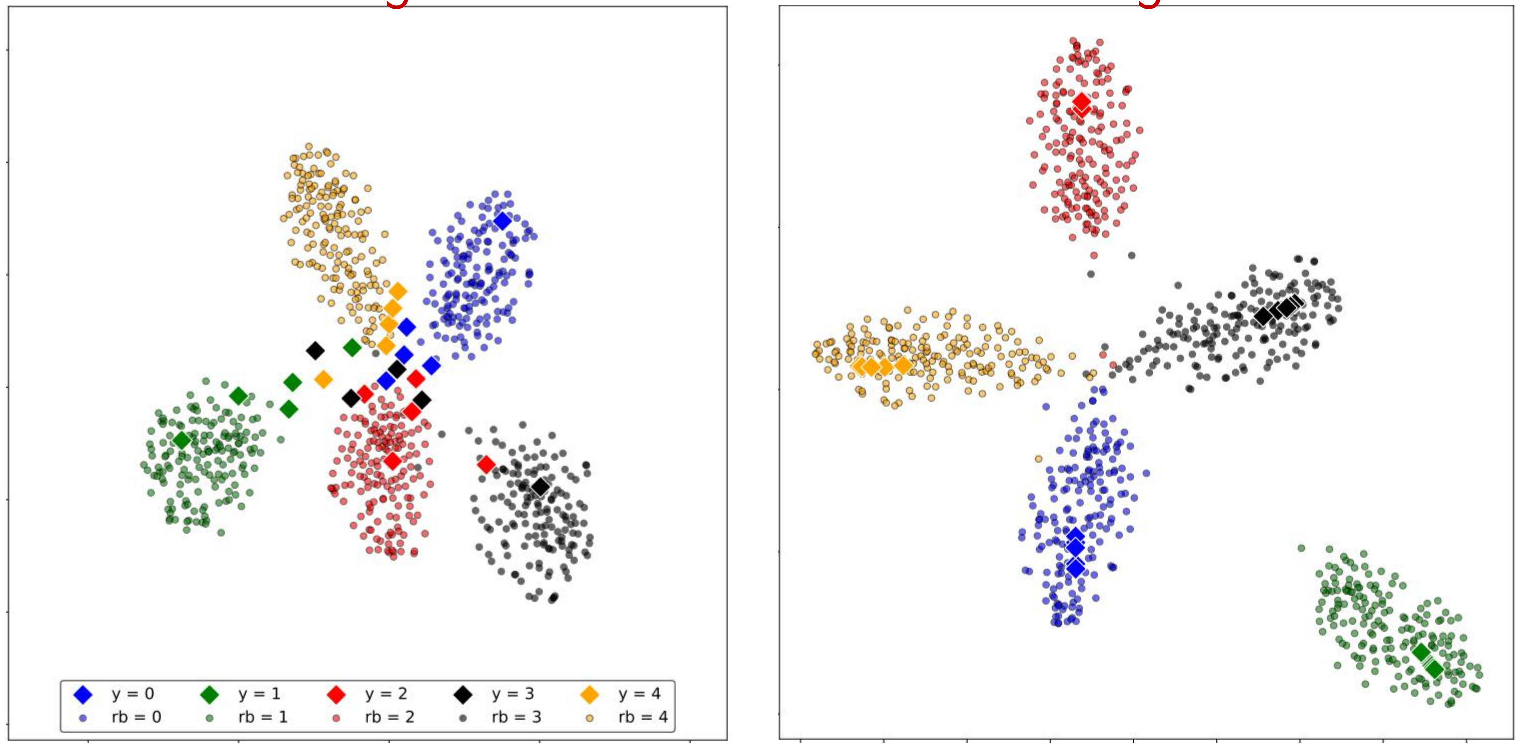
metric-learning and simplicity are necessary!

Associative Alignment

alignment of novel categories to their related base categories

- prevent overfitting
- keeps the learning capacity

before alignment after alignment

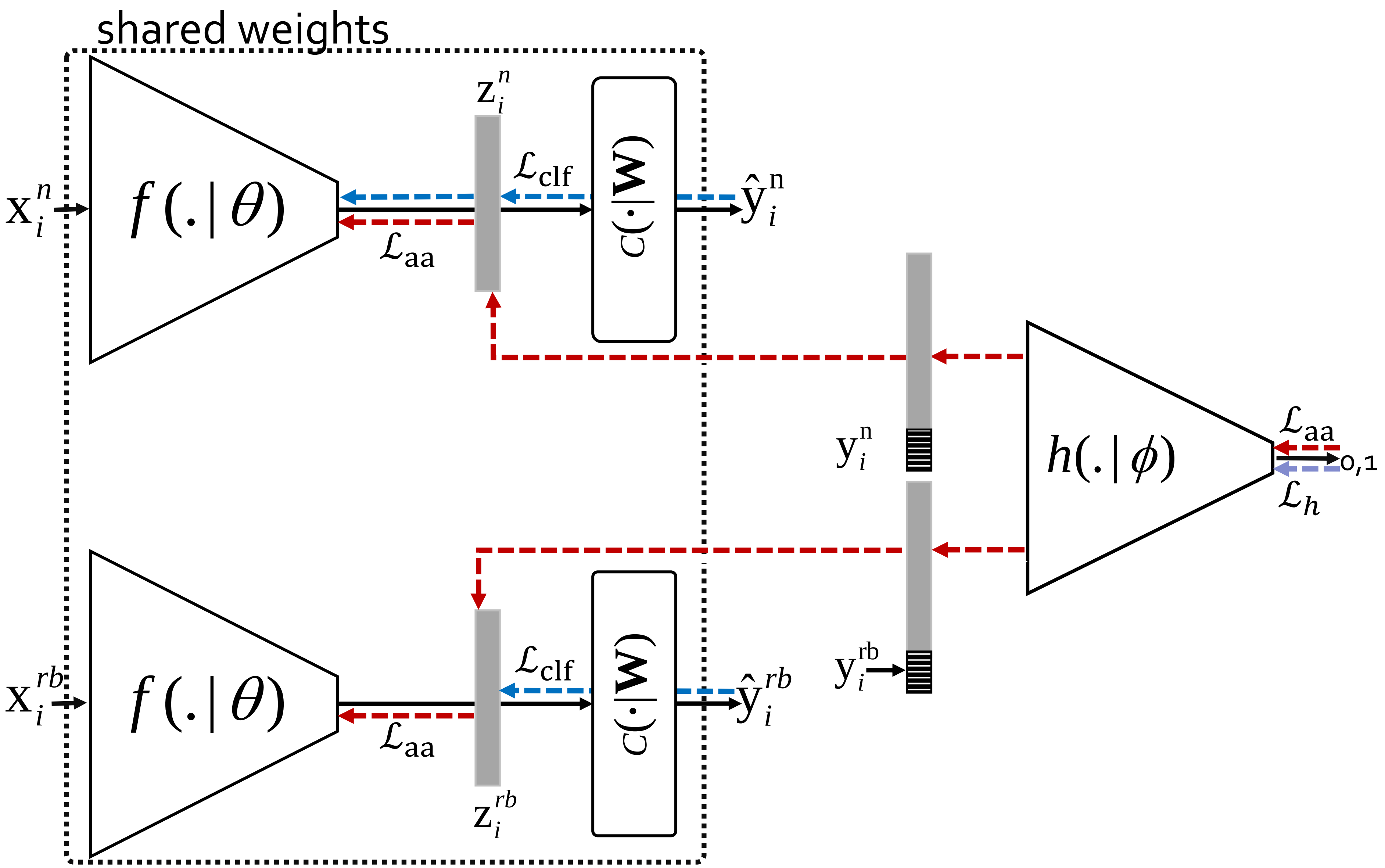


t-sne results: centroid alignment (B=1)

keeping the focus on novel categories:

- adversarial alignment: based on Wasserstein distance
- centroid alignment: based on Euclidian distance

adversarial alignment



Algorithm 1: Adversarial alignment algorithm.

Input: pre-trained model $f(\cdot|\theta)$, classifier $c(\cdot|\mathbf{W})$, novel class set $\mathcal{X}^n$, related base set $\mathcal{X}^{rb}$
Output: aligned network $c(f(\cdot|\theta)|\mathbf{W})$
while *not done* **do**
  $\widetilde{\mathcal{X}}^n \leftarrow$ sample a batch from $\mathcal{X}^n$
  $\widetilde{\mathcal{X}}^{rb} \leftarrow$ sample a batch from $\mathcal{X}^{rb}$

  for $i = 0, \ldots, n_{\text{critic}}$ **do**
    evaluate critic loss $\mathcal{L}_h(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$ with eq. 4
    update critic: $\phi \leftarrow \phi - \eta_h \nabla_\phi \mathcal{L}_h(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$
    $\phi \leftarrow \text{clip}(\phi, -0.01, 0.01)$
  end

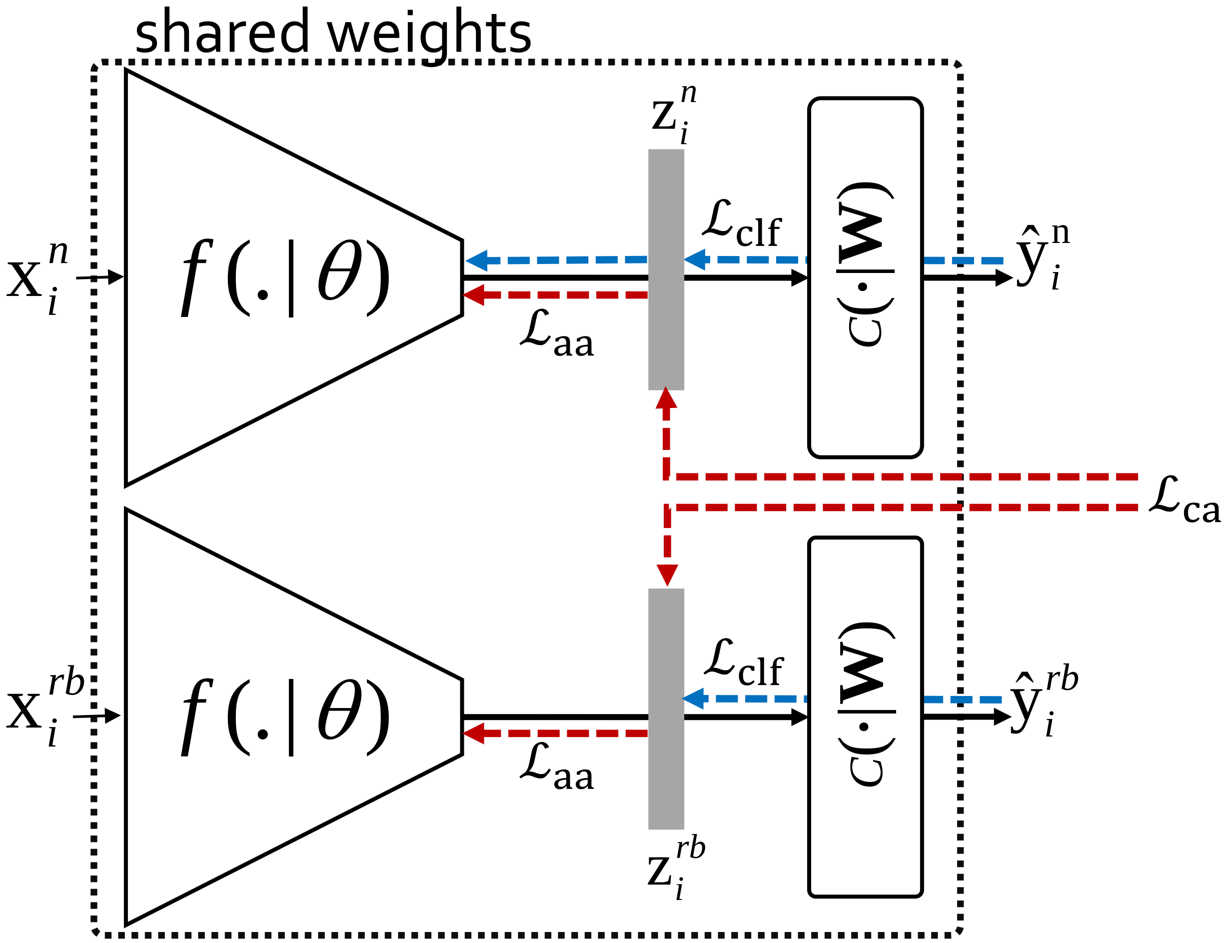
  evaluate alignment loss $\mathcal{L}_{aa}(\widetilde{\mathcal{X}}^n)$ with eq. 5
  $\theta \leftarrow \theta - \eta_{aa} \nabla_\theta \mathcal{L}_{aa}(\widetilde{\mathcal{X}}^n)$

  evaluate classification loss $\mathcal{L}_{clf}(\widetilde{\mathcal{X}}^{rb})$
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{clf} \nabla_{\mathbf{W}} \mathcal{L}_{clf}(\widetilde{\mathcal{X}}^{rb})$

  evaluate classification loss $\mathcal{L}_{clf}(\widetilde{\mathcal{X}}^n)$
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{clf} \nabla_{\mathbf{W}} \mathcal{L}_{clf}(\widetilde{\mathcal{X}}^n)$

  $\theta \leftarrow \theta - \eta_{clf} \nabla_\theta \mathcal{L}_{clf}(\widetilde{\mathcal{X}}^n)$
end

centroid alignment



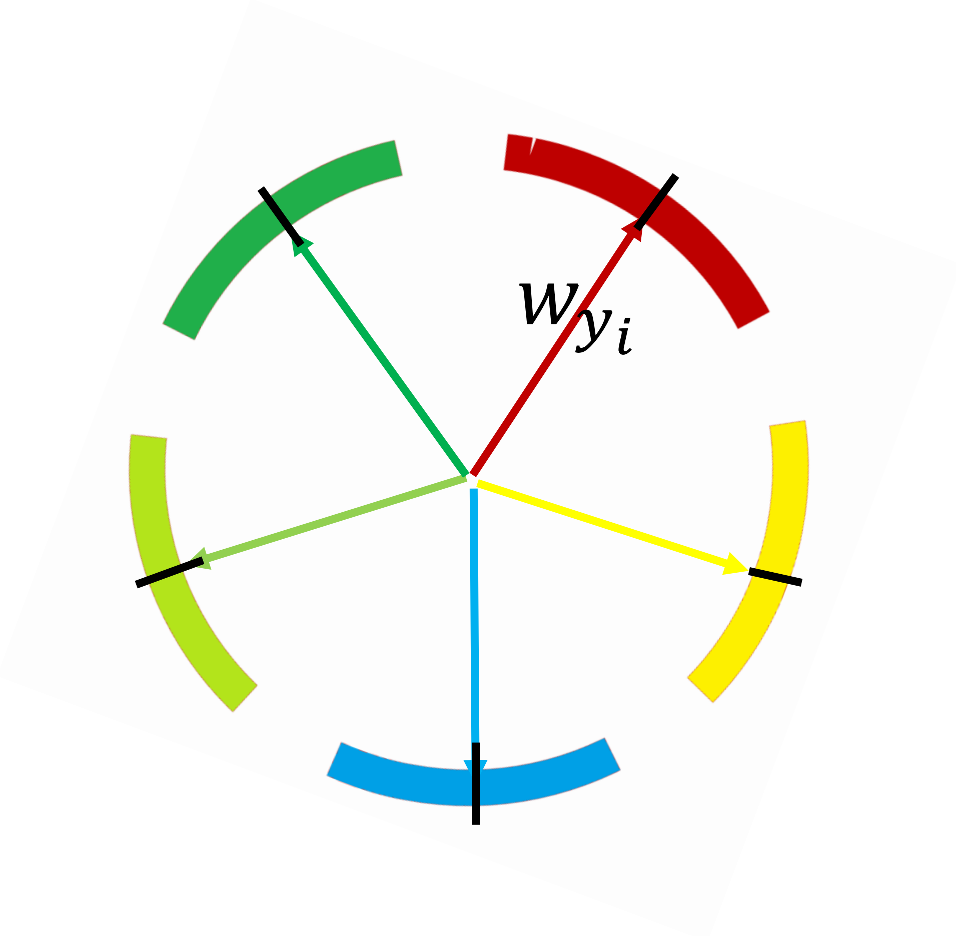
Algorithm 2: Centroid alignment algorithm.

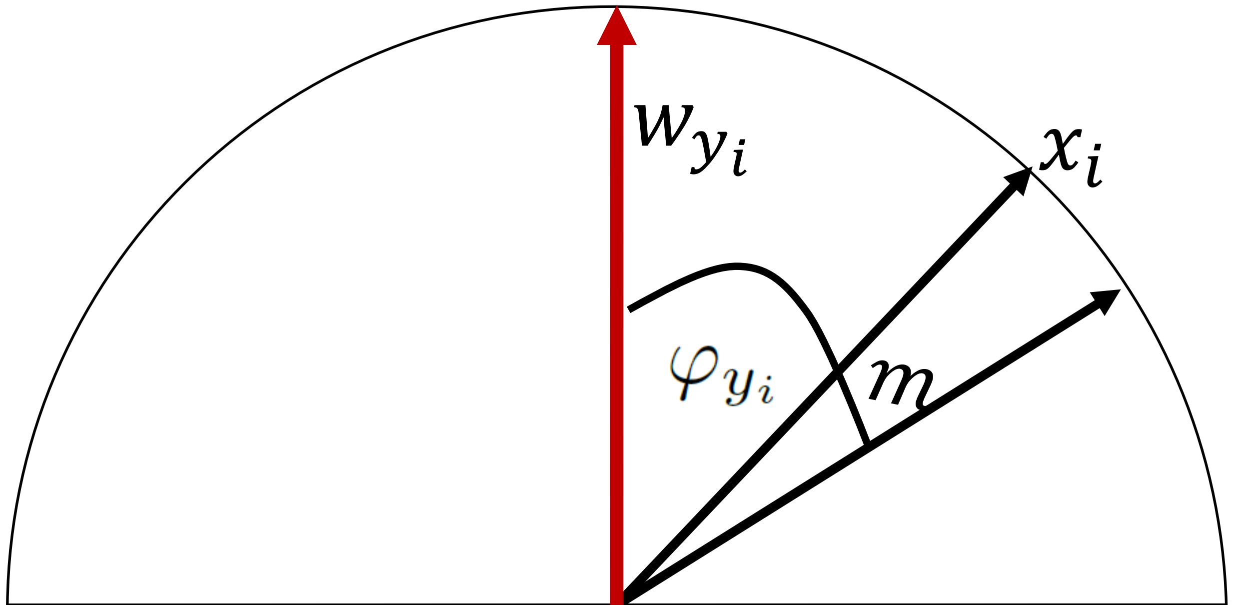
Input: pre-trained model $f(\cdot|\theta)$, classifier $c(\cdot|\mathbf{W})$, novel class set $\mathcal{X}^n$, related base set $\mathcal{X}^{rb}$
Output: aligned network $c(f(\cdot|\theta)|\mathbf{W})$
while *not done* **do**
- $\widetilde{\mathcal{X}}^n \leftarrow$ sample a batch from $\mathcal{X}^n$
- $\widetilde{\mathcal{X}}^{rb} \leftarrow$ sample a batch from $\mathcal{X}^{rb}$
- evaluate alignment loss $\mathcal{L}_{\text{ca}}(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$ with eq. 6
- $\theta \leftarrow \theta - \eta_{\text{ca}} \nabla_\theta \mathcal{L}_{\text{ca}}(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$
- evaluate classification loss $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$
- $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$
- evaluate classification loss $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
- $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
- $\theta \leftarrow \theta - \eta_{\text{clf}} \nabla_\theta \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$

end

strong baseline: spherical loss

$$\mathcal{L}_{\text{clf}} = \frac{-1}{N}\sum_{i=1}^{N}\log\frac{\exp(s\cos(\varphi_{y_i}+m))}{\exp(s\cos(\varphi_{y_i}+m))+\sum_{\forall j\neq y_i}\exp(s\cos\varphi_j)}$$



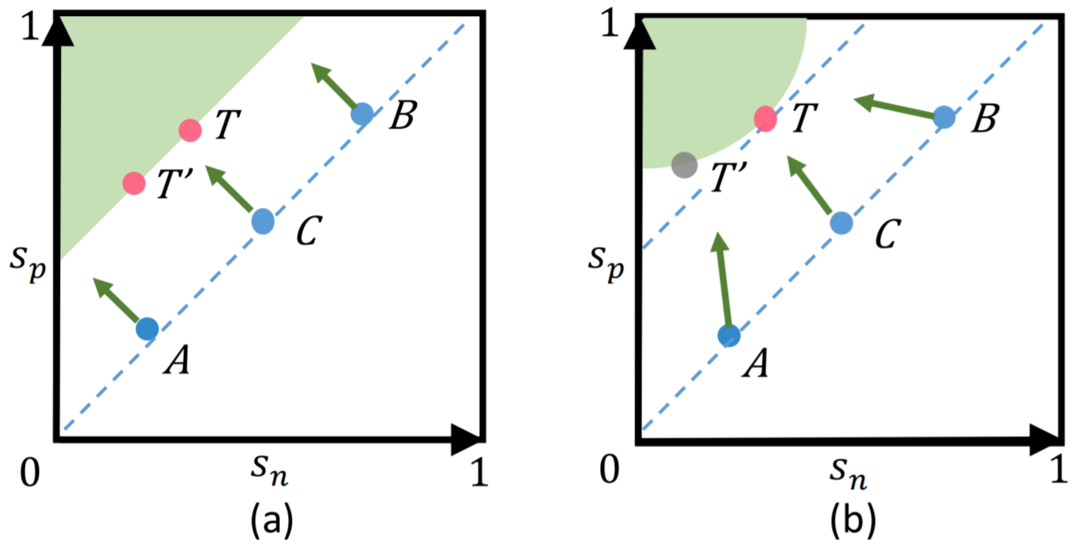


representation learning based

To minimize each s_n^j as well as to maximize s_p^i, ($\forall i \in \{1, 2, \cdots, K\}$, $\forall j \in \{1, 2, \cdots, L\}$), we propose a unified loss function by:

$$\begin{aligned}\mathcal{L}_{uni} &= \log\left[1 + \sum_{i=1}^{K}\sum_{j=1}^{L}\exp(\gamma(s_n^j - s_p^i + m))\right] \\ &= \log\left[1 + \sum_{j=1}^{L}\exp(\gamma(s_n^j + m))\sum_{i=1}^{K}\exp(\gamma(-s_p^i))\right],\end{aligned} \tag{1}$$

in which γ is a scale factor and m is a margin for better similarity separation.



Circle Loss
Sun et al., CVPR 2020

within class compactness
between class discrepancy

representation learning based

The Poincaré ball model $(\mathbb{D}^n, g^{\mathbb{D}})$ is defined by the manifold $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n \colon \|\mathbf{x}\| < 1\}$ endowed with the Riemannian metric $g^{\mathbb{D}}(\mathbf{x}) = \lambda_{\mathbf{x}}^2 g^E$, where $\lambda_{\mathbf{x}} = \frac{2}{1-\|\mathbf{x}\|^2}$ is the *conformal factor* and g^E is the Euclidean metric tensor $g^E = \mathbf{I}^n$. In this model the *geodesic distance* between two points is given by the following expression:

$$d_{\mathbb{D}}(\mathbf{x}, \mathbf{y}) = \operatorname{arccosh}\left(1 + 2\frac{\|\mathbf{x}-\mathbf{y}\|^2}{(1-\|\mathbf{x}\|^2)(1-\|\mathbf{y}\|^2)}\right). \quad (1)$$

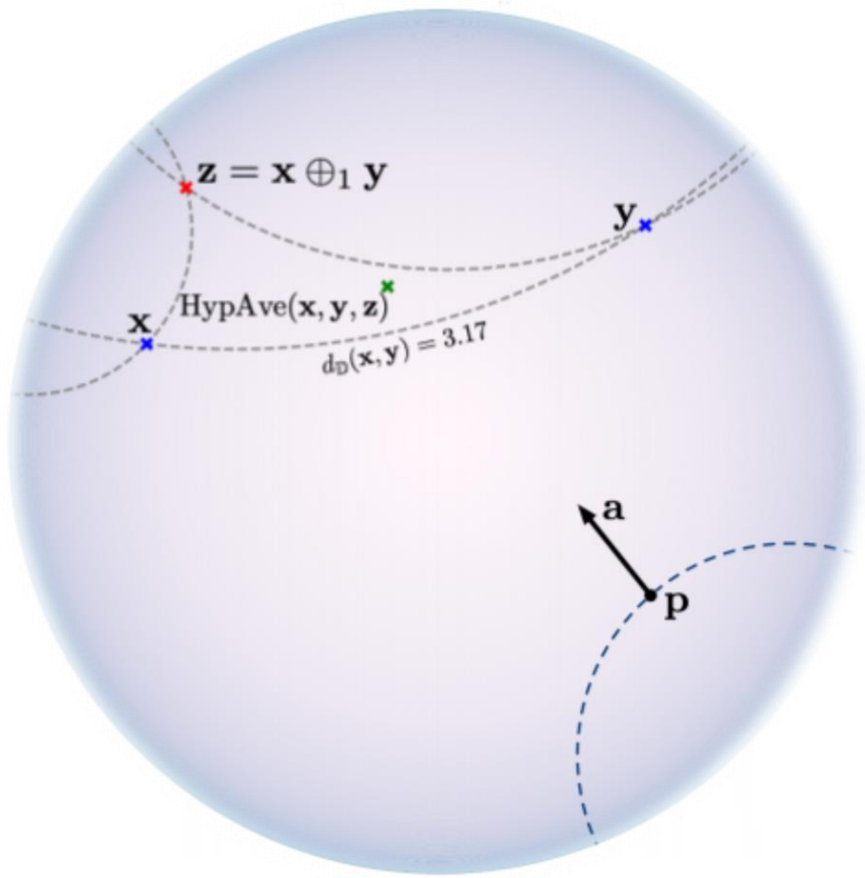


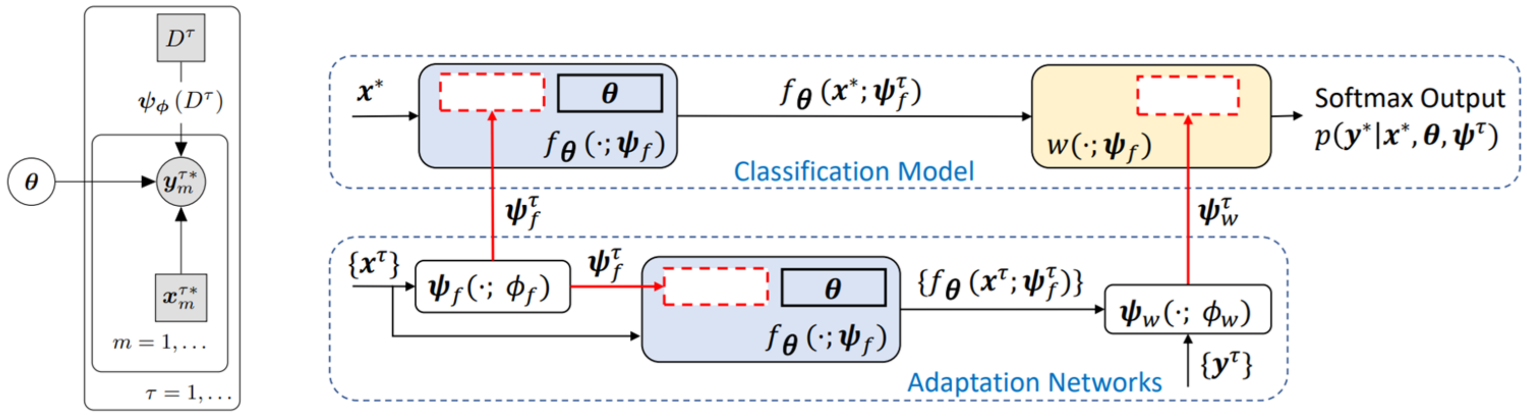
Figure 3: Visualization of the two–dimensional Poincaré ball. Point $\mathbf{z}$ represents the *Möbius sum* of points $\mathbf{x}$ and $\mathbf{y}$. HypAve stands for hyperbolic averaging. Gray lines represent *geodesics*, curves of shortest length connecting two points. In order to specify the *hyperbolic hyperplanes* (bottom), used for multiclass logistic regression, one has to provide an origin point $\mathbf{p}$ and a normal vector $\mathbf{a} \in T_{\mathbf{p}}\mathbb{D}^2 \setminus \{\mathbf{0}\}$.

Hyperbolic Image Embeddings
Khrulkov et al., CVPR 2020

graphical model based

Conditional Neural Adaptive Processes, NeurIPS 2019

avoids both over-fitting in low-shot regimes and under-fitting in high-shot regimes



(a) (b)

$$p\left(\boldsymbol{y}^* | \boldsymbol{x}^*, \boldsymbol{\theta}, D^\tau\right) = p\left(\boldsymbol{y}^* | \boldsymbol{x}^*, \boldsymbol{\theta}, \psi^\tau = \psi_\phi\left(D^\tau\right)\right)$$

thank you!

CVPR Review

Circle Loss Sun et al., CVPR 2020

To minimize each s_n^j as well as to maximize s_p^i, ($\forall i \in \{1, 2, \cdots, K\}, \forall j \in \{1, 2, \cdots, L\}$), we propose a unified loss function by:

$$\begin{aligned} \mathcal{L}_{uni} &= \log \left[1 + \sum_{i=1}^{K} \sum_{j=1}^{L} \exp(\gamma(s_n^j - s_p^i + m))\right] \\ &= \log \left[1 + \sum_{j=1}^{L} \exp(\gamma(s_n^j + m)) \sum_{i=1}^{K} \exp(\gamma(-s_p^i))\right], \end{aligned} \tag{1}$$

in which γ is a scale factor and m is a margin for better similarity separation.

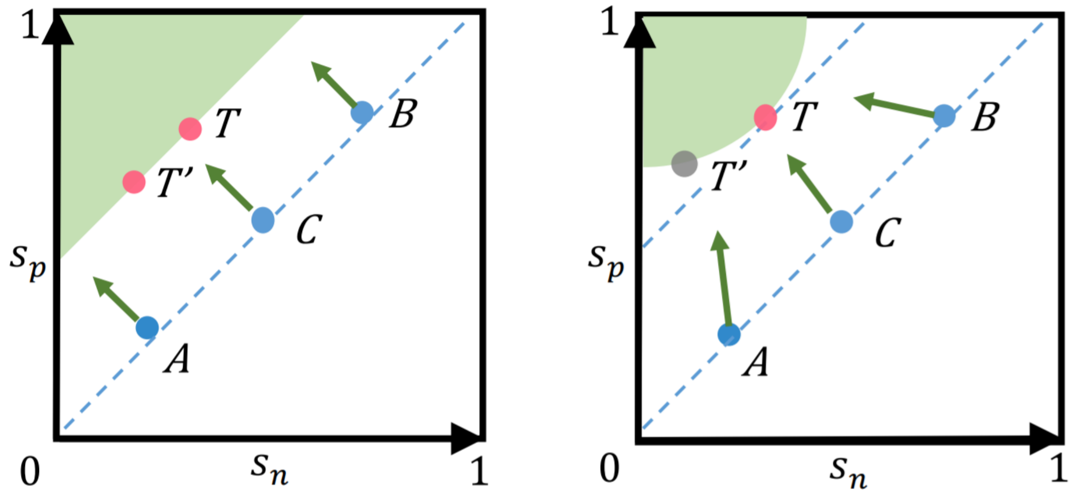
Based on the unified viewpoint, we simply make a further generalization by:

$(s_n - s_p)$ ⇒ $(\alpha_n s_n - \alpha_p s_p)$

- Inflexible optimization
- Ambiguous convergence

✓ More flexible optimization

✓ More definite convergence



(a) (b)

within class compactness s_p

between class discrepancy s_n

Hyperbolic Image Embeddings, Khrulkov et al., CVPR 2020

The Poincaré ball model $(\mathbb{D}^n, g^{\mathbb{D}})$ is defined by the manifold $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$ endowed with the Riemannian metric $g^{\mathbb{D}}(\mathbf{x}) = \lambda_{\mathbf{x}}^2 g^E$, where $\lambda_{\mathbf{x}} = \frac{2}{1-\|\mathbf{x}\|^2}$ is the *conformal factor* and g^E is the Euclidean metric tensor $g^E = \mathbf{I}^n$. In this model the *geodesic distance* between two points is given by the following expression:

$$d_{\mathbb{D}}(\mathbf{x}, \mathbf{y}) = \operatorname{arccosh}\left(1 + 2\frac{\|\mathbf{x}-\mathbf{y}\|^2}{(1-\|\mathbf{x}\|^2)(1-\|\mathbf{y}\|^2)}\right). \quad (1)$$

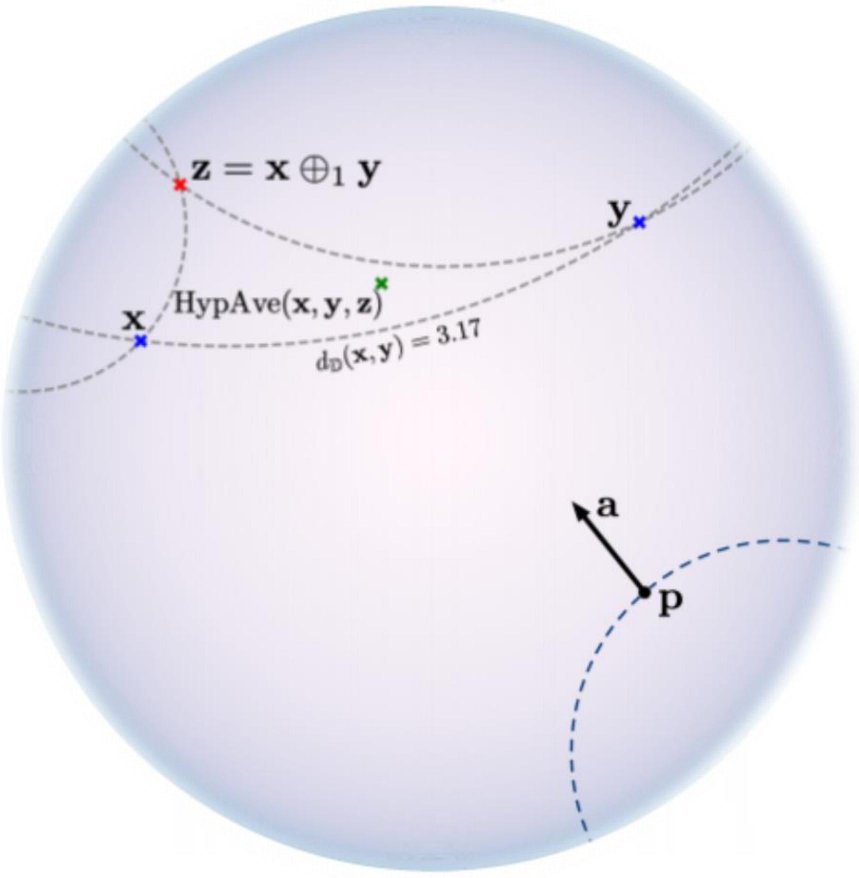


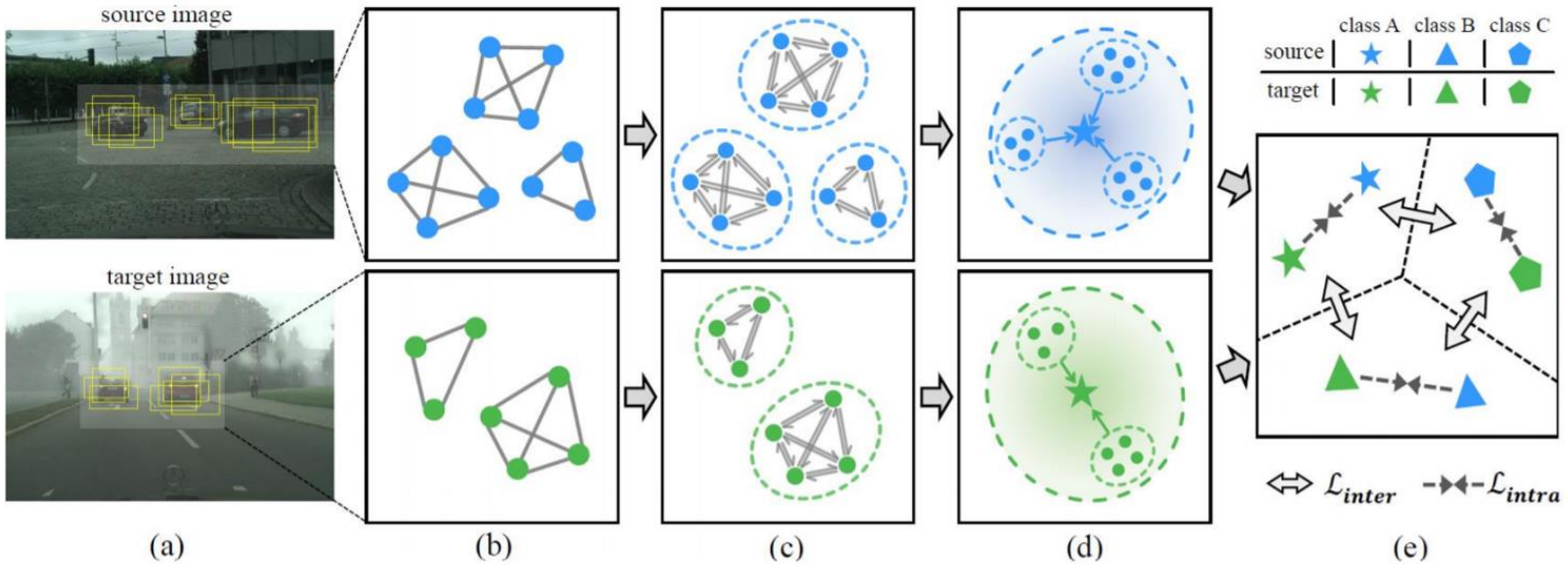
Figure 3: Visualization of the two–dimensional Poincaré ball. Point $\mathbf{z}$ represents the *Möbius sum* of points $\mathbf{x}$ and $\mathbf{y}$. HypAve stands for hyperbolic averaging. Gray lines represent *geodesics*, curves of shortest length connecting two points. In order to specify the *hyperbolic hyperplanes* (bottom), used for multiclass logistic regression, one has to provide an origin point $\mathbf{p}$ and a normal vector $\mathbf{a} \in T_{\mathbf{p}}\mathbb{D}^2 \setminus \{\mathbf{0}\}$.

Graph-Induced Prototype Alignment, Xu et al., CVPR 2020

Subject:
Domain Adaptation based object detection

Problem:
RPN results are not aligned with the objects

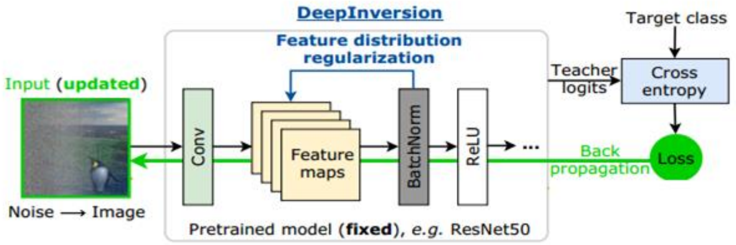
Solution:
prototypical alignment



(a) Generate region proposals.
(b) Construct relation graph.
(c) Obtain instance-level feature representations.
(d) Derive per-category prototypes.
(e) Category-level domain alignment.

Dreaming to Distill, Yin et al., CVPR 2020

$$\min_{\hat{x}} \mathcal{L}(\hat{x}, y) + \mathcal{R}_{\text{prior}}(\hat{x}) + \mathcal{R}_{\text{feature}}(\hat{x}) + \mathcal{R}_{\text{compete}}(\hat{x})$$



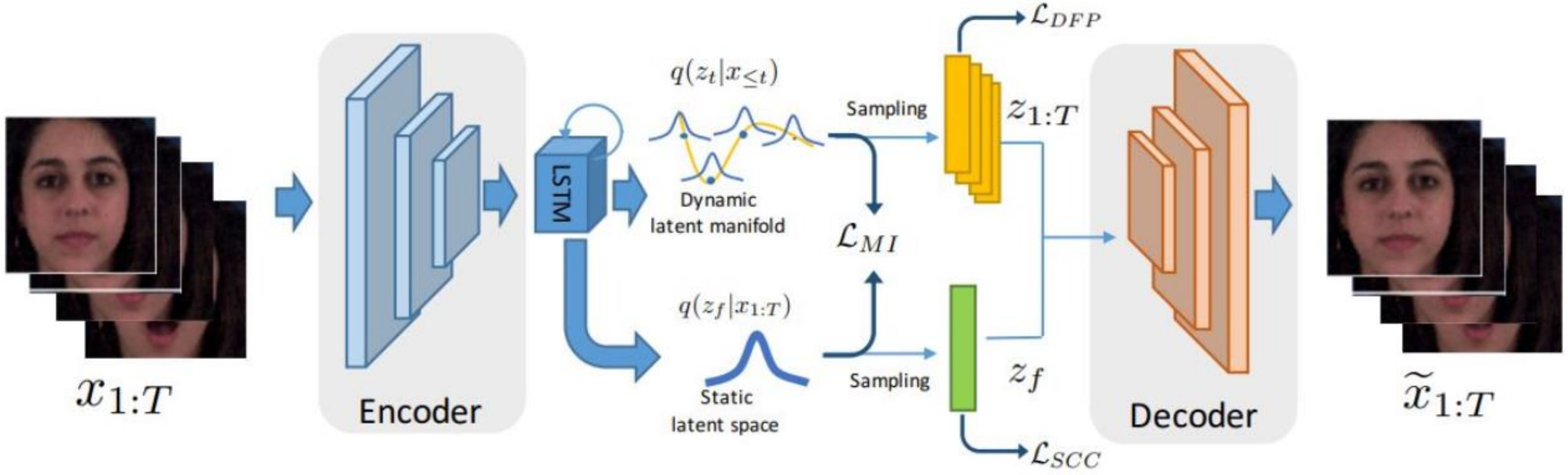
$\mathcal{R}_{\text{prior}}(\hat{x})$ improve image quality (but fail to generate the natural images)

$\mathcal{R}_{\text{feature}}(\hat{x})$ to enforce BN statistics of feature maps

$\mathcal{R}_{\text{compete}}(\hat{x})$ improve image diversity by encouraging teacher/student competition

Synthesized Images from ResNet-50

S3VAE, Zhu et al., CVPR 2020



- Encoder
- Decoder
- LSTM in the latent space

VAE Objectives: $$\mathcal{L}_{VAE} = \mathbb{E}_{q(\boldsymbol{z}_{1:T}, \boldsymbol{z}_f|\boldsymbol{x}_{1:T})}[-\sum_{t=1}^{T} \log p(\boldsymbol{x}_t|\boldsymbol{z}_f, \boldsymbol{z}_t)] + \mathrm{KL}(q(\boldsymbol{z}_f|\boldsymbol{x}_{1:T})||p(\boldsymbol{z}_f)) + \sum_{t=1}^{T} \mathrm{KL}(q(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t})||p(\boldsymbol{z}_t|\boldsymbol{z}_{<t}))$$

few-shot learning

is supervised transferring pre-knowledge