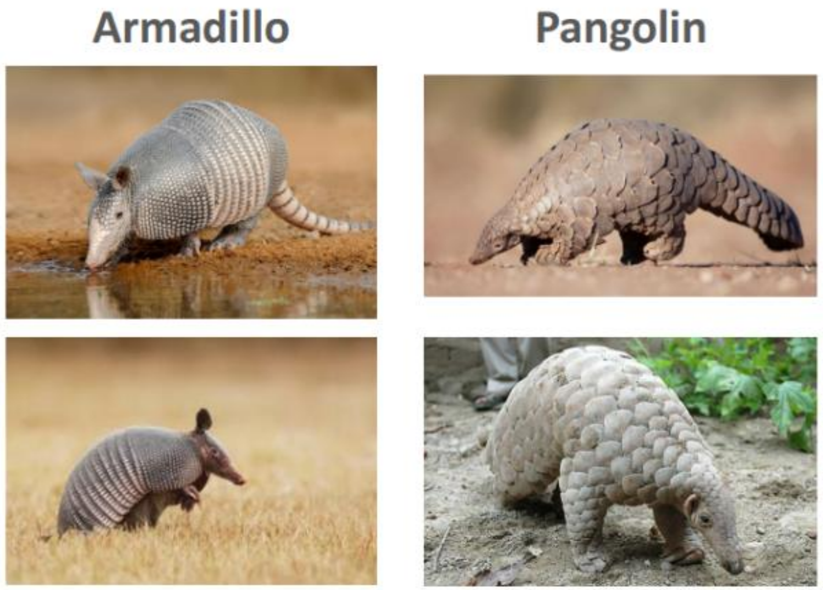
FeLMi : Few shot Learning with hard Mix-up

-Summary-

Introduction

- In supervised learning, we train model based on ‘training set’ and classify them into (already) given classes (from labels)



- But, what if we need to classify ‘unseen’ class (not in training set)?

Introduction

- In real worlds, we have (almost) unlimited classes to classify animals
  (cat, dog, dog with brown fur, tiger, Siberian tiger…)

  => It is impossible to classify them in supervised learning
  (due to data scarcity and unlimited model output size)

- One technique to detour this problem : 'Few shot learning' (type of meta-learning)
  - Goal : **check whether they are same class** rather what class they are in.

Bulldog

$\mathbf{x}_1$

Bulldog

$\mathbf{x}_2$

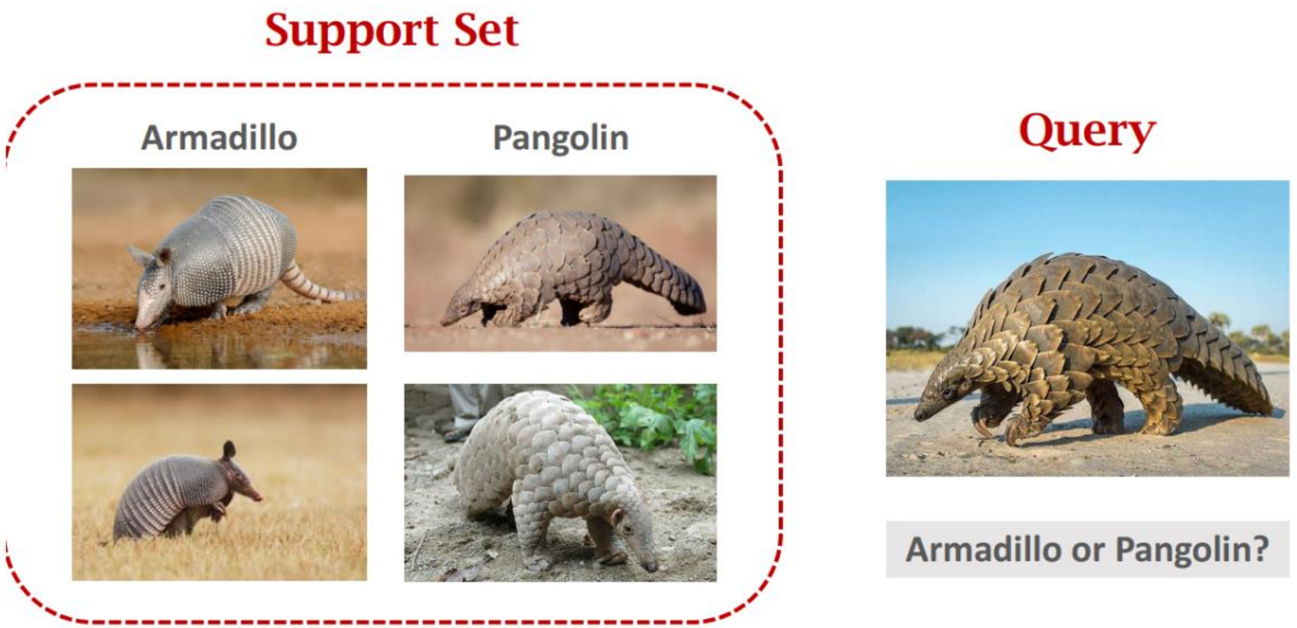
Fox

$\mathbf{x}_3$

$\mathrm{sim}(\mathbf{x}_1, \mathbf{x}_2) = 1, \quad \mathrm{sim}(\mathbf{x}_1, \mathbf{x}_3) = 0, \quad \text{and} \quad \mathrm{sim}(\mathbf{x}_2, \mathbf{x}_3) = 0$

Basic concept of Few shot learning (FSL)

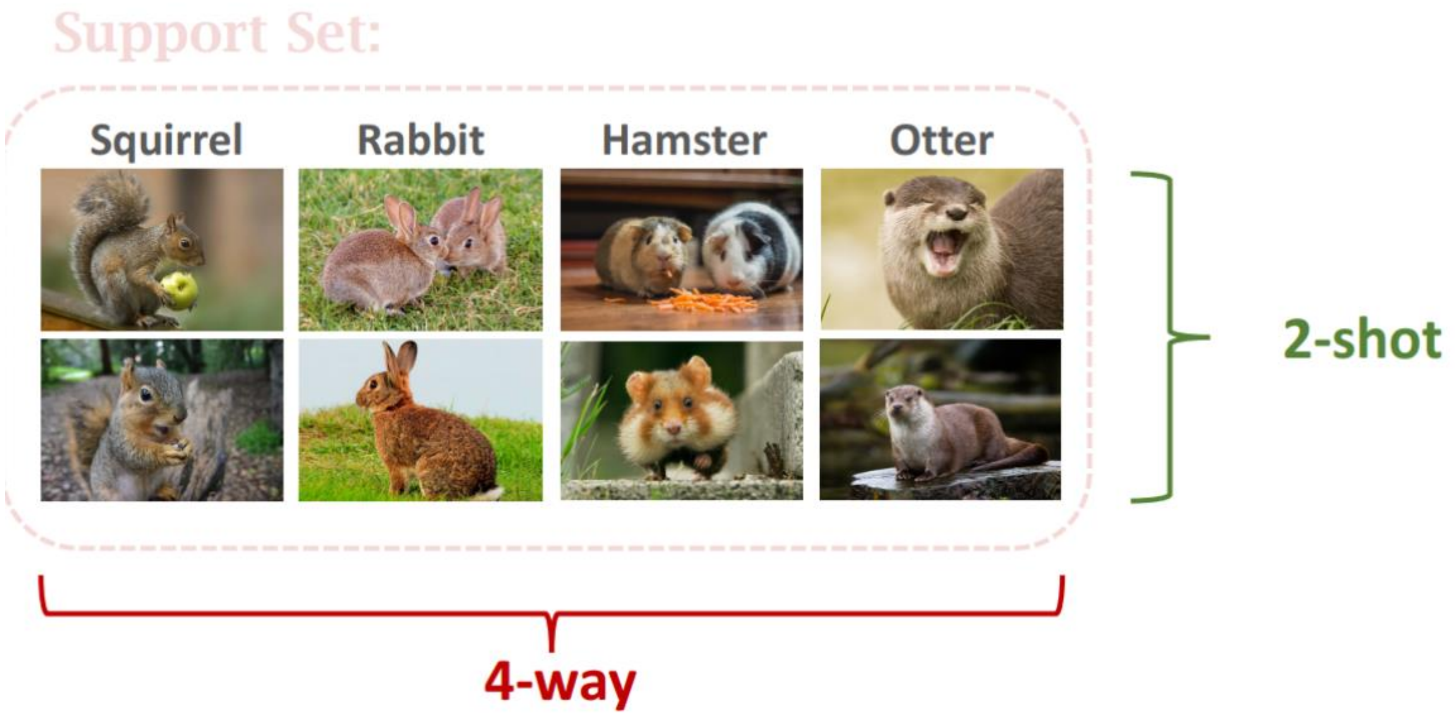
- In FSL, we have '**support set**' and **'query**' instead of 'training set' and 'test sample'



- Out precise goal : Given support set (typically 'few' data), select most similar class (among support set) of query sample.
  (i.e : we train model so that it can capture the similarity of samples)

Basic concept of Few shot learning (FSL)

- In general, the given support set is very small.
  (because, we only have few unseen dataset in real worlds.)
- Conventional format of supports set : K-way N-shot
  (K = # of class in support set, N = # of samples of each class in support set)



Few shot learning (Episodic training)

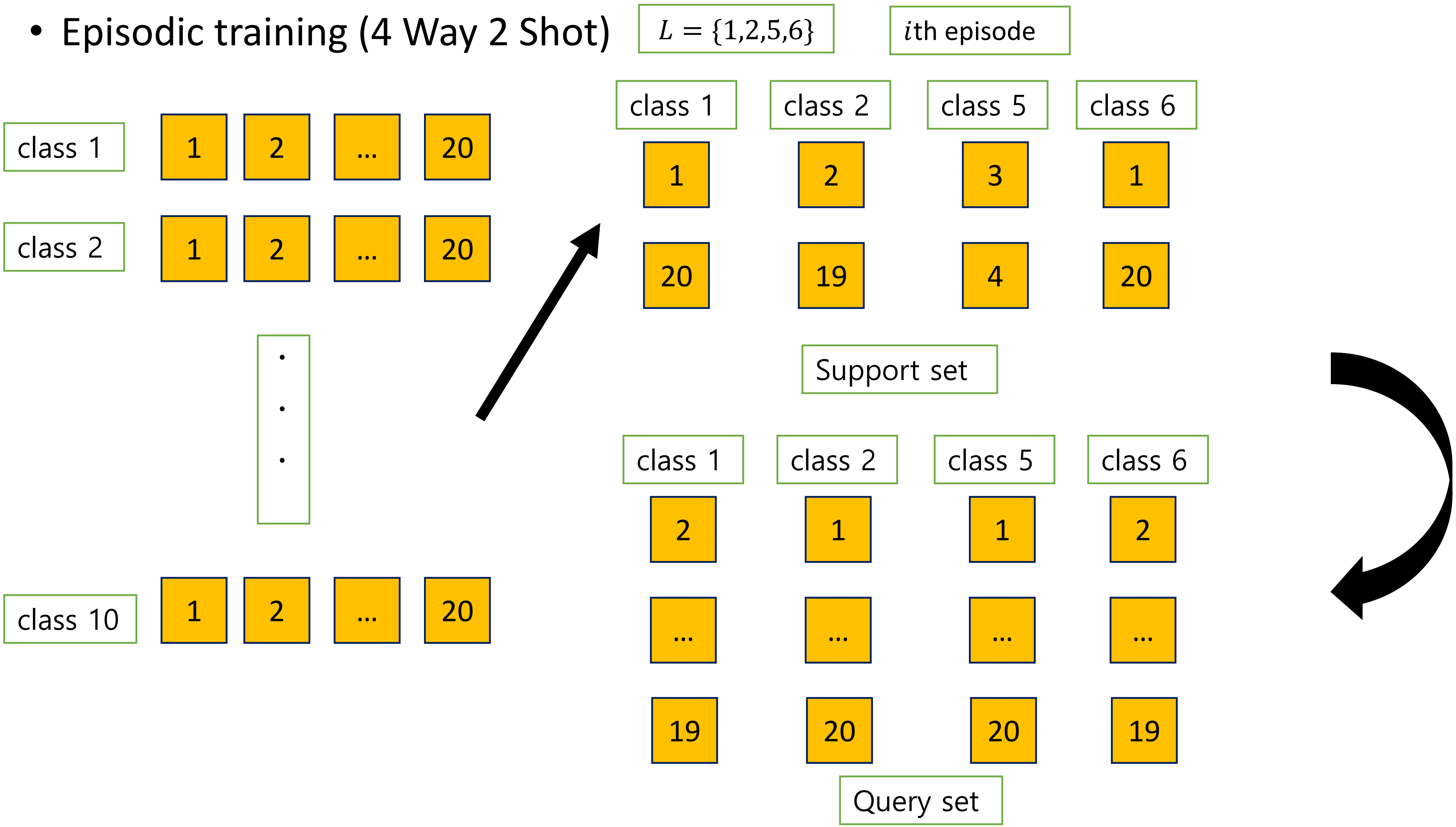
- Then, How to form support set or train model?
  - Current SOTA method for FSL : **episodic training / transfer learning**
- **Episodic training** [Vinalys., 2016]: $(K$-way N-shot, l episodes)
  1. Given training dataset, randomly pick K classes among training classes.
  2. Given selected K classes, pick N samples from training classes.
     (This forms support set of size NK for 1st episode)
  3. Given selected K classes, pick remaining samples from training classes
     (This forms query set for 1st episode)
  4. Train model using support set and compute criterion by query set.
  5. Repeat this l times to finish training.

(L = label sets, S = support set, Q = query set)

One possible criterion? : $\theta = argmax_{\theta} \mathbb{E}_{L\sim T}\left[\mathbb{E}_{S\sim L, Q\sim L}\left[\sum_{(x,y)\in Q} \log P_{\theta}(y|x,S)\right]\right]$

Few shot learning (Episodic training)

- Episodic training (4 Way 2 Shot)



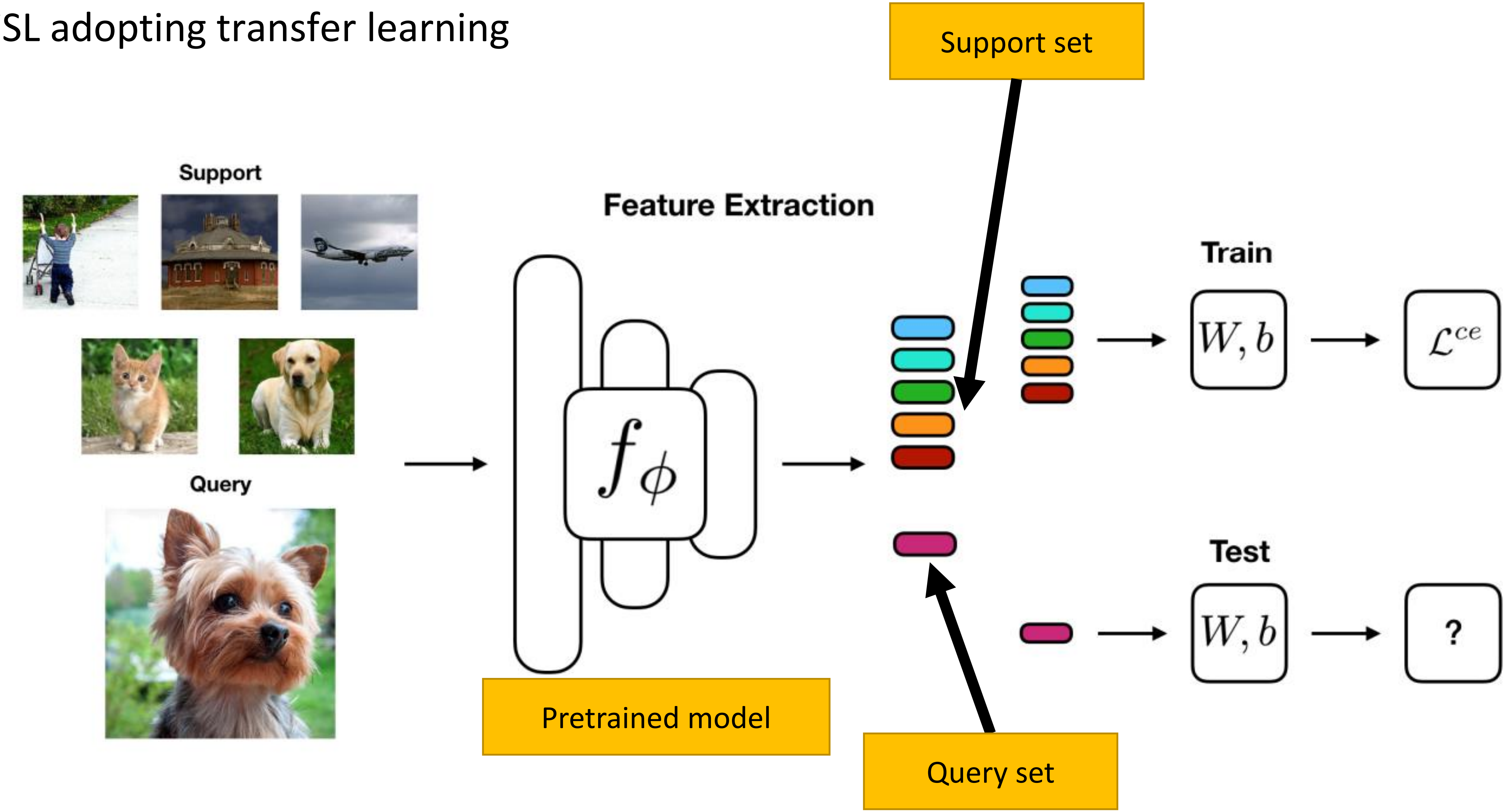
Few shot learning (transfer learning)

- One problem of episodic training :
  - **Time consuming** (for large episode, typically 1 million episode) and suffers from inductively generated **bias from previous episode** (unavoidable property of episodic learning)

- Another **SOTA method for FSL (adopting transfer learning)** [Tian et al., 2020]
  1. Pretrain model (ex : ResNet-18) based on huge training dataset (ex : ImageNet) to learn a good representation
  2. Now, for each support set, attach simple linear classifier and learn corresponding support set. (Transfer learning => Note : fix the pretrained model)
  3. Use this model to predict query set.

Few shot learning (transfer learning)

- FSL adopting transfer learning



FeLMi (FSL with hard mix-up)

- Intuitively, the model performance on FSL increases as the # of shots(N) increase.
  - In practical, increasing shots in support set requires high cost => **Use mix-up to augment the # of shots!** (fundamental idea of FeLMi)

- Overall steps for FeLMI : Learning representations using training dataset.
  1. Pseudo-labeling of training dataset using classifier trained on support set. (pseudo-labels will be used for knowledge distillation on step 5)
  2. Entropy based pseudo-label filtering of training dataset
  3. Mix-up sample generation
  4. Hard mix-up sample generation
  5. Fine-tune on entire dataset

FeLMi (FSL with hard mix-up)

- Intuitively, the model performance on FSL increases as the # of shots(N) increase.
  - In practical, increasing shots in support set requires high cost => **Use mix-up to augment the # of shots!** (fundamental idea of FeLMi)

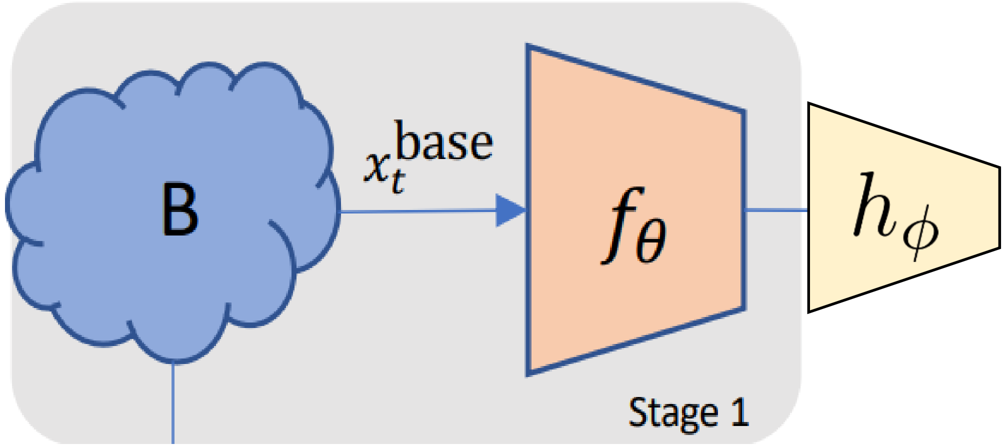
- Problem setting of FeLMi : (N-way K-shot)
  - Training dataset (base dataset) : $\mathcal{D}^{base} = \left\{x_t^{base}, y_t^{base}\right\}_{t=1}^{N^{base}}$
  - Novel dataset (place where support sets are sampled) : $\mathcal{D}^{novel} = \left\{x_t^{novel}, y_t^{novel}\right\}_{t=1}^{N^{novel}}$
  - Assume $C^{base} \cap C^{novel} = \phi$ and $\left|C^{base}\right| \geq |C^{novel}|$,where C is class set
  - The training and testing is performed in episodes on novel class samples
    : FSL learner trained on $\mathcal{D}_i^{support}$ for N novel classes containing K samples
    and evaluated on query set $\mathcal{D}_i^{query}$ on the same classes of $\mathcal{D}_i^{support}$ (size NK)

FeLMi (FSL with hard mix-up)

1. Learning representations using training dataset.

: Learn representation from training dataset $\mathcal{D}_{base}$ using model $h_\phi(f_\theta)$

(f_θ : representation learning model, h_ϕ : final classification layer)



$$(\theta^{base}, \phi^{base}) = argmin_{\theta,\phi} \mathbb{E}_{\{x,y\}\in\mathcal{D}^{base}}\left[L_{CE}\left(h_\phi\left(f_\theta(x)\right), y\right)\right]$$

Notation : B : training dataset ($\mathcal{D}_{base}$)

FeLMi (FSL with hard mix-up)

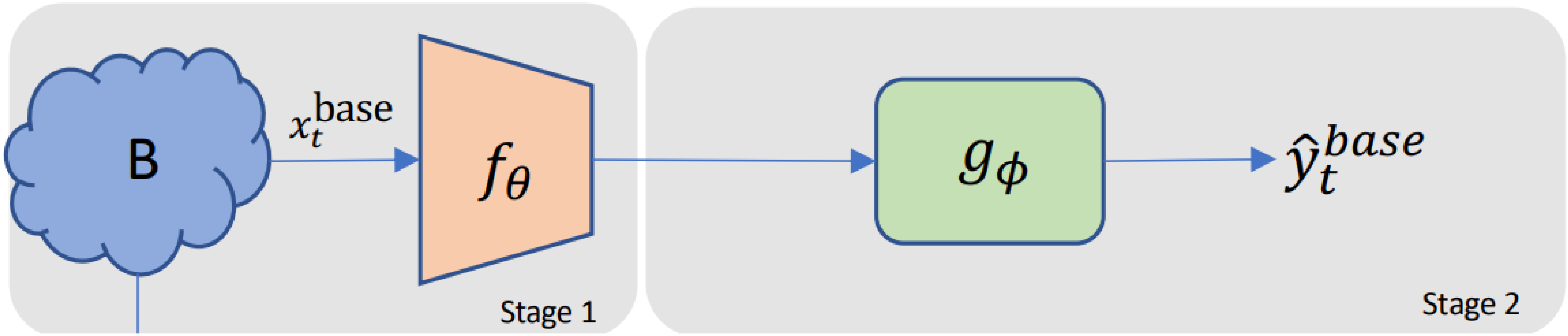
2. Pseudo-labeling of training dataset using classifier trained on support set.

2-1 : Learn linear classifier ϕ_i using the support set of each i th episode (= $\mathcal{D}_i^{support}$) :

$$\phi_i = argmin_{\phi} \mathbb{E}_{\{x,y\} \in \mathcal{D}_i^{support}} \left[L_{CE} \left(g_{\phi} \left(f_{\theta^{base}}(x) \right), y \right) \right]$$

2-2 : Generated pseudo-labels on entire training dataset:

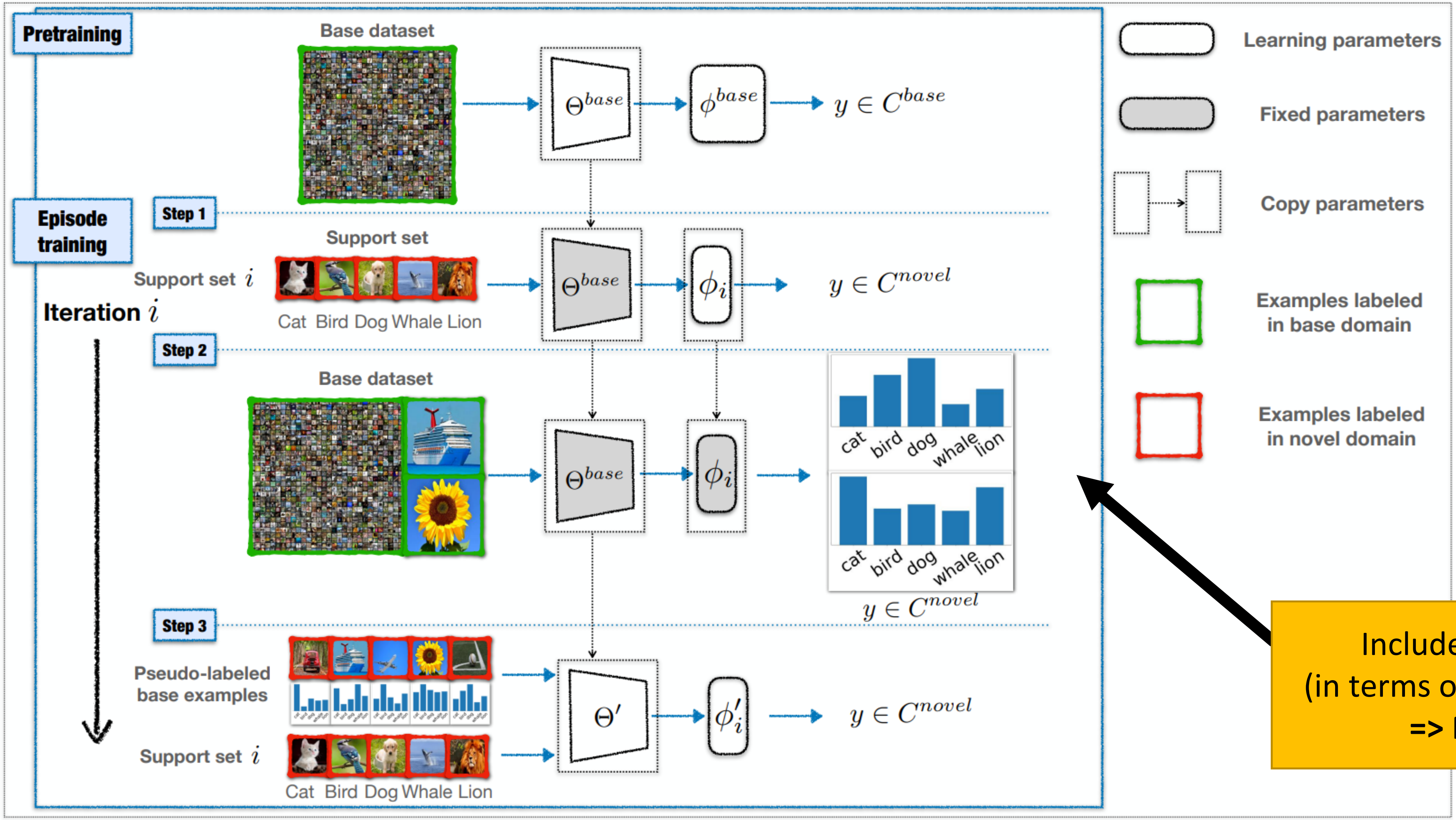
$$\hat{y}_t^{base} = g_{\phi_i} \left(f_{\theta^{base}}(x_t) \right) \quad for\ t = 1 \ldots N^{base}$$



FeLMi (FSL with hard mix-up)

Label Hallucination for Few-Shot classification [Jian et al., 2022]

2. Pseudo-labeling of training dataset using classifier trained on support set.

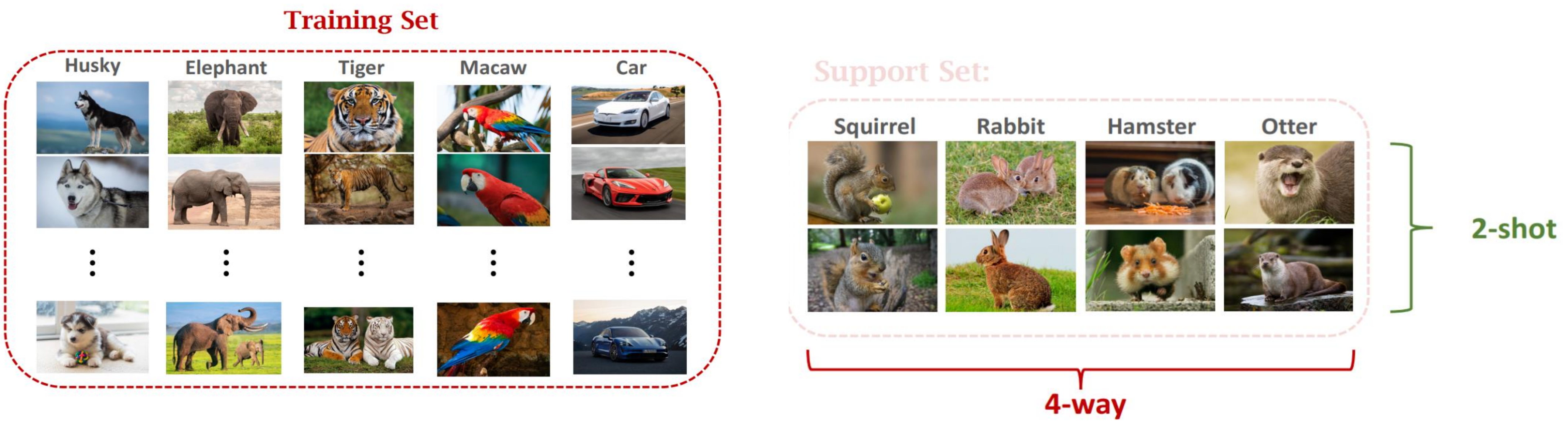


Include tons of useless data
(in terms of training on support set)
=> Perform filtering!

FeLMi (FSL with hard mix-up)

3. Entropy based pseudo-label filtering of training dataset

- Not all classes in base dataset will provide useful information for given support set (or under certain episode)
- For example :



In this case, pseudo-labeling for tiger and husky (which has similar representation) would be helpful **=> How to find helpful samples in training set?**

FeLMi (FSL with hard mix-up)

3. Entropy based pseudo-label filtering of training dataset

- Suggested method : filtering based on entropy
  - High entropy => confusing samples
  - Low entropy => somewhat useful samples

- Recall : there is correlation : C-score ⇔ negative Entropy (proxy) ⇔ data valuation score
  Hence, we are choosing easy samples.

- Set (empirically) entropy threshold τ, and filter $\mathcal{D}^{base}$ to obtain D^{base_filt} for each episode

$$\mathcal{D}^{base\_filt} = \{\hat{y}_t{}^{base} | H\left(\hat{y}_t{}^{base}\right) \leq \tau \quad where\ t = 1, \ldots, N^{base}\}$$

FeLMi (FSL with hard mix-up)

4. Mix-up sample generation

- We now have pseudo-labeled (filtered) base samples and labeled support set samples.

- To augment samples related with support set, we perform two types of mix-up:
  - Novel-Novel mix-up : manifold mix-up of novel samples
  - Base-Novel mix-up : Hard mix-up between base and novel samples based on entropy

FeLMi (FSL with hard mix-up)

4. Mix-up sample generation (Novel – Novel mix up)

4-1. Select $\{(x^{novel}, y^{novel}), (\bar{x}^{novel}, \bar{y}^{novel})\} \in \mathcal{D}^{support}$ (ignore subscript i) and perform manifold mix-up right after feature extraction layer :

$$x_{mix}^{N-N} = \lambda_n . f_{\theta^{\mathtt{base}}}(x^{\mathtt{novel}}) + (1 - \lambda_n) f_{\theta^{\mathtt{base}}}(\bar{x}^{\mathtt{novel}})$$
$$y_{mix}^{N-N} = \lambda_n . y^{\mathtt{novel}} + (1 - \lambda_n) \bar{y}^{\mathtt{novel}}$$

where $\lambda_n \sim beta(\alpha, \alpha)$

4-2. Form pool of novel-novel mixup samples of size l:

$$P_{N,N} = \left\{ \left(x_{mix}^{N-N}, y_{mix}^{N-N} \right)_i \right\}_{i=1}^{l}$$

FeLMi (FSL with hard mix-up)

4. Mix-up sample generation (Base-Novel mixup)

To perform base-novel mixup, we employ following policies :

1) No mix-up between two base examples (obviously useless for support set training) Instead, perform mix-up between base – novel samples.

2) Only mix with base samples that are close to novel samples (otherwise, generated mix-up samples would tend to be noisy or extremely hard)

3) Choose small mix-up parameter λ so that generated mix-up samples remain proximal to the distribution of novel samples.

FeLMi (FSL with hard mix-up)

4. Mix-up sample generation (Base-Novel mixup)

4-3. From $\mathcal{D}^{base\_filt}$, select k-lowest entropy base samples $(x_{sel}^{base}, \hat{y}_{sel}^{base})$:

$$\{(x_{sel}^{base}, \hat{y}_{sel}^{base})\} = \{(x_i, y_i) \mid i \in \text{bottom\_k}(H(\hat{y}))\}$$

(Choosing base examples that are close novel examples)

4-4. Perform mix-up with novel samples $(x^{novel}, y^{novel}) \in \mathcal{D}^{support}$ with selected base samples :

$$x_{mix}^{B-N} = \lambda_b . f_{\theta^{\texttt{base}}}(x_{sel}^{\texttt{base}}) + (1 - \lambda_b) f_{\theta^{\texttt{base}}}(x^{\texttt{novel}})$$
$$y_{mix}^{B-N} = \lambda_b . \hat{y}_{sel}^{\texttt{base}} + (1 - \lambda_b) y^{\texttt{novel}}$$

where $\boldsymbol{\lambda_b \sim uniform(0, 0.2)}$, and set the pool of base-novel mix-up samples of size l

$$P_{B,N} = \left\{ (x_{mix}^{B-N}, y_{mix}^{B-N})_i \right\}_{i=1}^{l}$$

FeLMi (FSL with hard mix-up)

5. Hard mix-up sample generation :

- Set $P_{mix} = P_{B,N} \cup P_{N,N}$ and choose hardest N samples based on a uncertainty measure

- Here, we set **'difference in top-2 probabilities'** as a measure for uncertainty (or margin) (Commonly adopted measure in active learning)

- Now, choose hardest k mix-up samples in P_{mix} and form P_{hard_mix}:

$$\mathcal{P}_{hard\_mix} = \texttt{bottom\_k}\{\texttt{margin}(g_{\phi_i}(f_{\theta^{\text{base}}}(x)) \mid (x, y) \in P_{mix}\}$$

(Recall : smaller margin => closer to decision boundary => harder examples)

FeLMi (FSL with hard mix-up)

To preserve model from deviate drastically from pre-trained model (Knowledge distillation)

6. Finetune on the entire dataset :

- Finally, the model is fine-tuned on a combined loss computed using $\mathcal{D}^{base\_filt}$, $\mathcal{D}^{novel}$, P_{hard_mix} :

$$\mathcal{L} = \mathbb{E}_{\{x,\hat{y}\} \in \mathcal{D}^{\texttt{base\_filt}}} L_{KD}(g_\phi(f_\theta(x)), \hat{y}) + \beta \mathbb{E}_{\{x,y\} \in \mathcal{D}^{\texttt{novel}}} L_{CE}(g_\phi(f_\theta(x)), y) + \gamma \mathbb{E}_{\{x,y\} \in \mathcal{P}_{\texttt{hard\_mix}}} L_{CE}(g_\phi(f_\theta(x)), y)$$

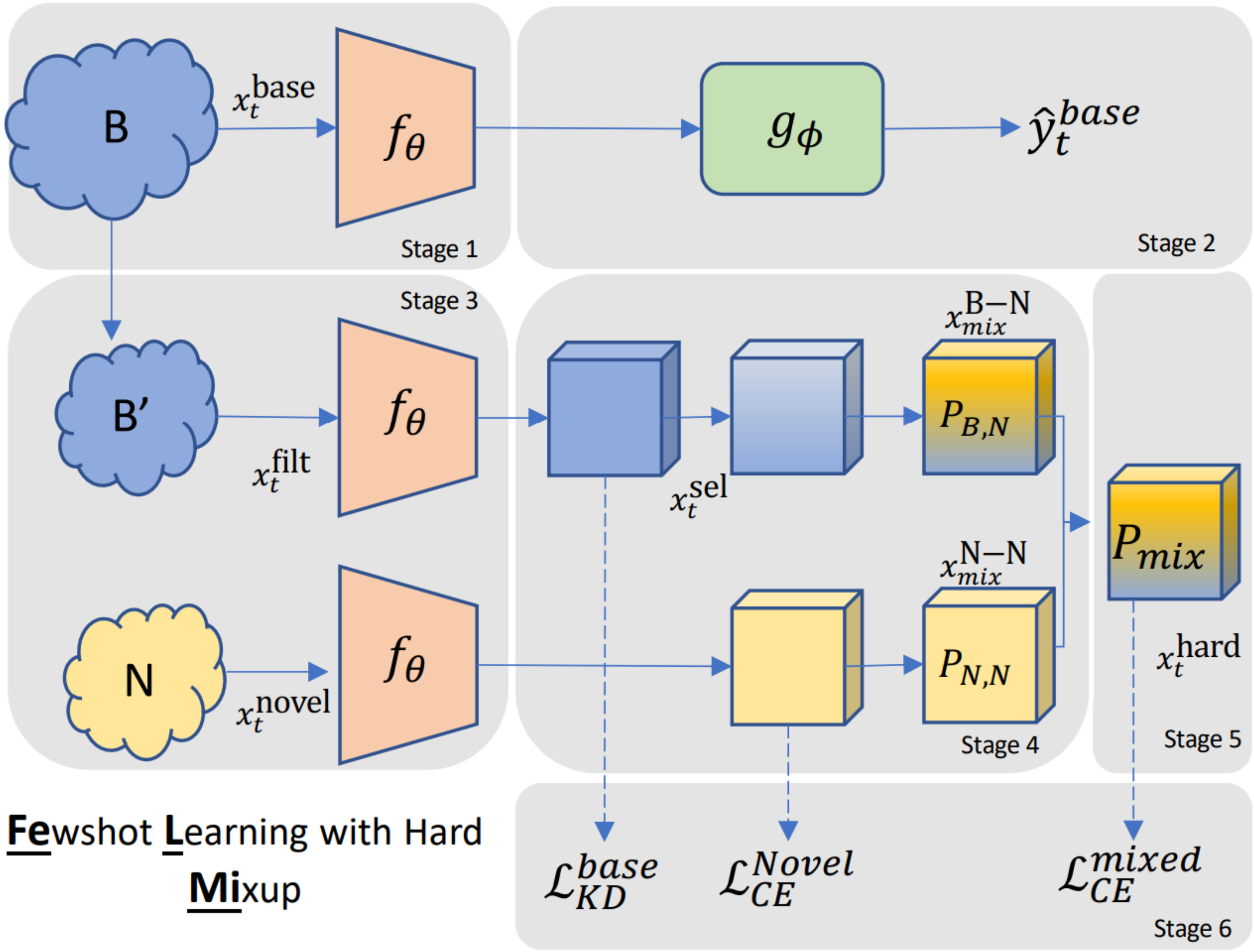
To finetune on novel dataset

To learn mix-up samples (data augmentation on support set)

Note : $\boldsymbol{\beta}, \boldsymbol{\gamma}$ is scaling parameter and $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are trained for each episode $\boldsymbol{i}$

FeLMi (FSL with hard mix-up)

Overview of FeLMi :



Fewshot **L**earning with Hard **Mi**xup

FeLMi (FSL with hard mix-up) : Experiment results

- Experiment result on CIFAR-FS (re-partitioned CIFAR-100 for FSL)

Model	Backbone	CIFAR-FS 5-way	
		1-shot	5-shot
ProtoNet [28] (NIPS'17)	ResNet-12	72.2 ± 0.7	83.5 ± 0.5
MetaOptNet [13] (CVPR'19)	ResNet-12	72.6 ± 0.7	84.3 ± 0.5
Shot-Free [19] (ICCV'19)	ResNet-12	$69.2 \pm$ n/a	$84.7 \pm$ n/a
DSN-MR [27] (CVPR'20)	ResNet-12	75.6 ± 0.9	86.2 ± 0.6
RFS-simple [31] (ECCV'20)	ResNet-12	71.5 ± 0.8	86.0 ± 0.5
RFS-distill [31] (ECCV'20)	ResNet-12	73.9 ± 0.8	86.9 ± 0.5
SKD-GEN1 [18] (Arxiv'20)	ResNet-12	76.6 ± 0.9	88.6 ± 0.5
IER-distill [22] (CVPR'21)	ResNet-12	77.6 ± 1.0	89.7 ± 0.6
PAL [15] (ICCV'21)	ResNet-12	77.1 ± 0.7	88.0 ± 0.5
Label-Halluc [11] (AAAI'22)	ResNet-12	$78.0 \pm 1.0^{\S}$	$89.37 \pm 0.6^{\S}$
FeLMi	ResNet-12	$\mathbf{78.22 \pm 0.7}$	$\mathbf{89.47 \pm 0.5}$

Test accuracy adopting 95% confidence interval

FeLMi (FSL with hard mix-up) : Experiment results

- Experiment result on FC-100 (another re-partitioned CIFAR-100 for FSL)

Model	Backbone	FC-100 5-way 1-shot	FC-100 5-way 5-shot
ProtoNet [28] (NIPS'17)	ResNet-12	37.5 ± 0.6	52.5 ± 0.6
TADAM [17] (NIPS'18)	ResNet-12	40.1 ± 0.4	56.1 ± 0.4
MetaOptNet [13] (CVPR'19)	ResNet-12	41.1 ± 0.6	55.5 ± 0.6
MTL [29] (CVPR'19)	ResNet-12	45.1 ± 1.8	57.6 ± 0.9
DeepEMD [38] (CVPR'20)	ResNet-12	46.5 ± 0.8	63.2 ± 0.7
RFS-simple [31] (ECCV'20)	ResNet-12	42.6 ± 0.7	59.1 ± 0.6
RFS-distill [31] (ECCV'20)	ResNet-12	44.6 ± 0.7	60.9 ± 0.6
AssoAlign [1] (ECCV'20)	ResNet-18	45.8 ± 0.5	59.7 ± 0.6
SKD-GEN1 [18] (Arxiv'20)	ResNet-12	46.5 ± 0.8	64.2 ± 0.8
InfoPatch [10] (AAAI'21)	ResNet-12	43.8 ± 0.4	58.0 ± 0.4
IER-distill [22] (CVPR'21)	ResNet-12	48.1 ± 0.8	65.0 ± 0.7
PAL [15] (ICCV'21)	ResNet-12	47.2 ± 0.6	64.0 ± 0.6
Label-Halluc [11] (AAAI'22)	ResNet-12	$47.37 \pm 0.7^{\S}$	$67.92 \pm 0.7^{\S}$
FeLMi	ResNet-12	$\mathbf{49.02} \pm \mathbf{0.7}$	$\mathbf{68.68} \pm \mathbf{0.7}$

Test accuracy adopting 95% confidence interval

FeLMi (FSL with hard mix-up) : Experiment results

- Experiment result (Contribution of Mix-up and Hard selection + different mixup strategy):

Approach	Accuracy
IER [22]	65.00
+ pseudo-label [11]	67.92
+ entropy filtering	67.96
+ Mixup	68.49
+ hard selection	68.68

Contributions for each method for FSL

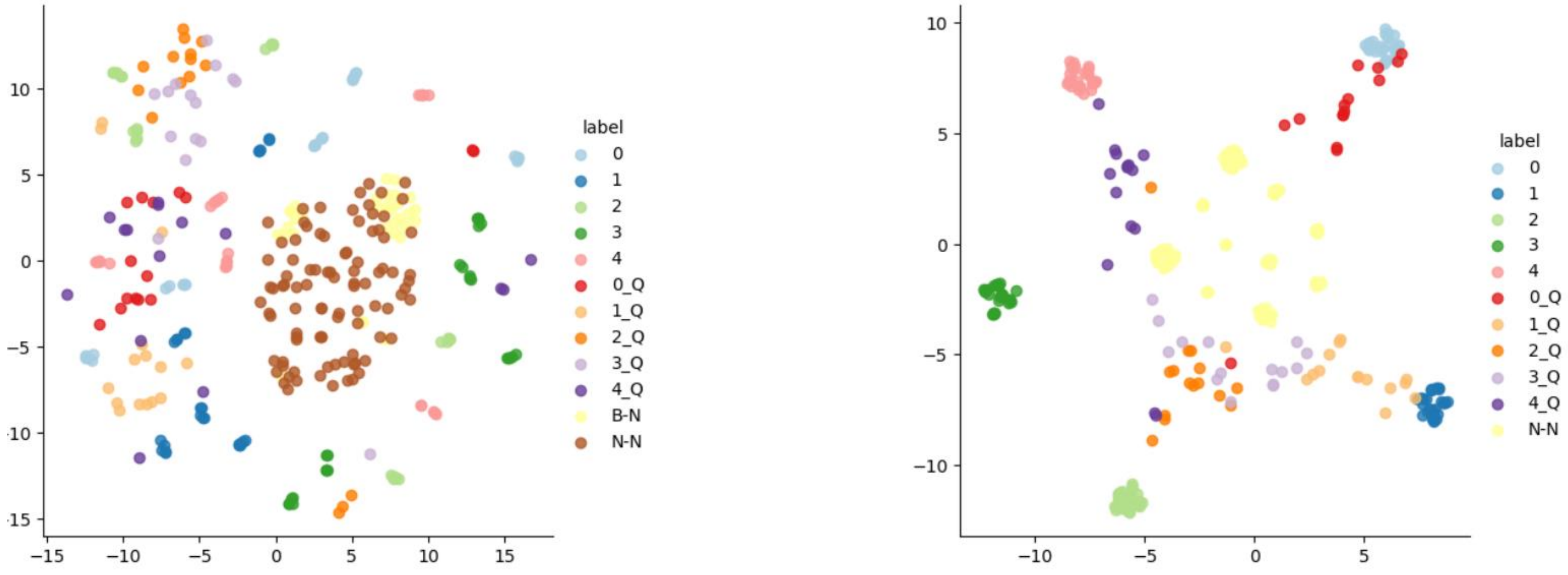
Mixup Approach	λ_b	λ_n	FC-100	CIFAR-FS
B-N + N-N	$\mathrm{U}(0, 0.2)$	$\mathrm{B}(1, 1)$	68.68	89.47
B-N + N-N	$\mathrm{B}(1, 1)$	$\mathrm{B}(1, 1)$	68.49	89.26
N-N	-	$\mathrm{B}(1, 1)$	68.57	89.4
None	-	-	67.92	89.37

Results using different mix-up strategy

- IER : method for learning good representation from training dataset (step 1)

FeLMi (FSL with hard mix-up) : Experiment results

- Experiment result (t-SNE after training of one episode)



(a) First training step (b) Last training step

t-SNE visualization of learned representation at the start of training and end for one random episode

- Mix-up samples offer a good training signal to learn better class boundaries