Meta-Learning in Smart Voice Control Systems

PhD Oral Examination

Chen Yangbin

Department of Computer Science

Supervisor: **Prof. WANG Jianping**
External supervisor: **Prof. LI Qing**
Qualifying panel member: **Prof. WANG Cong**

Outline

- Background
- Meta-learning for text-independent speaker verification
- Meta-learning for user-defined spoken term classification
- Improved meta-learning with consistency regularization
- Conclusion and future work

Part I

Background

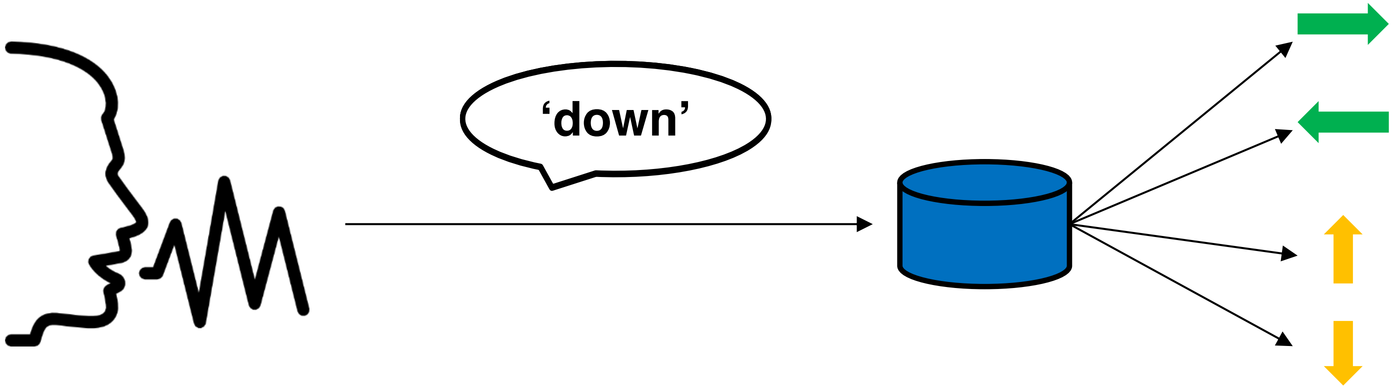
Smart voice control devices

Smart voice control systems

Two typical tasks in a voice control system

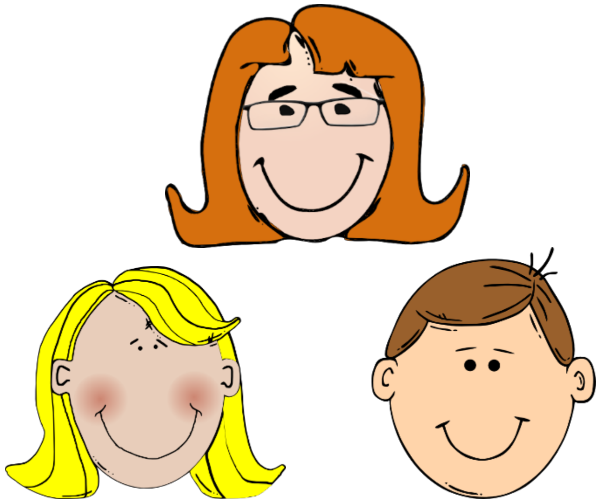
- Speaker Verification

- Spoken Term Classification (Command Recognition)

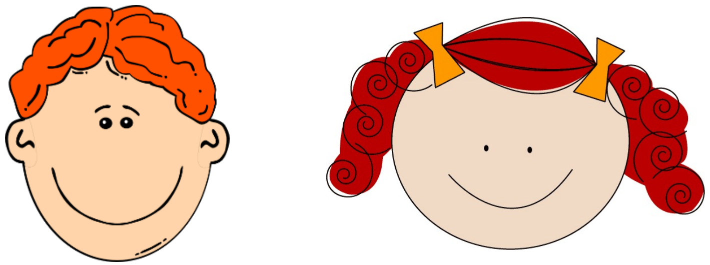


New challenges in a voice control system

- Speaker Verification



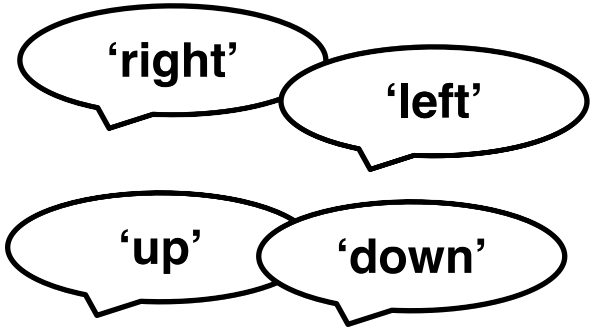
new speakers

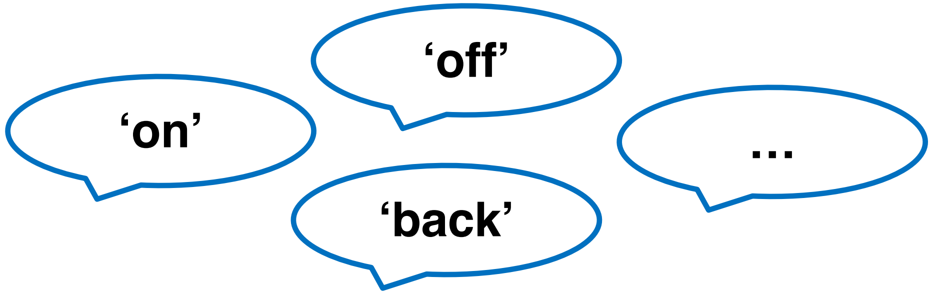


…

- Spoken Term Classification (Command Recognition)

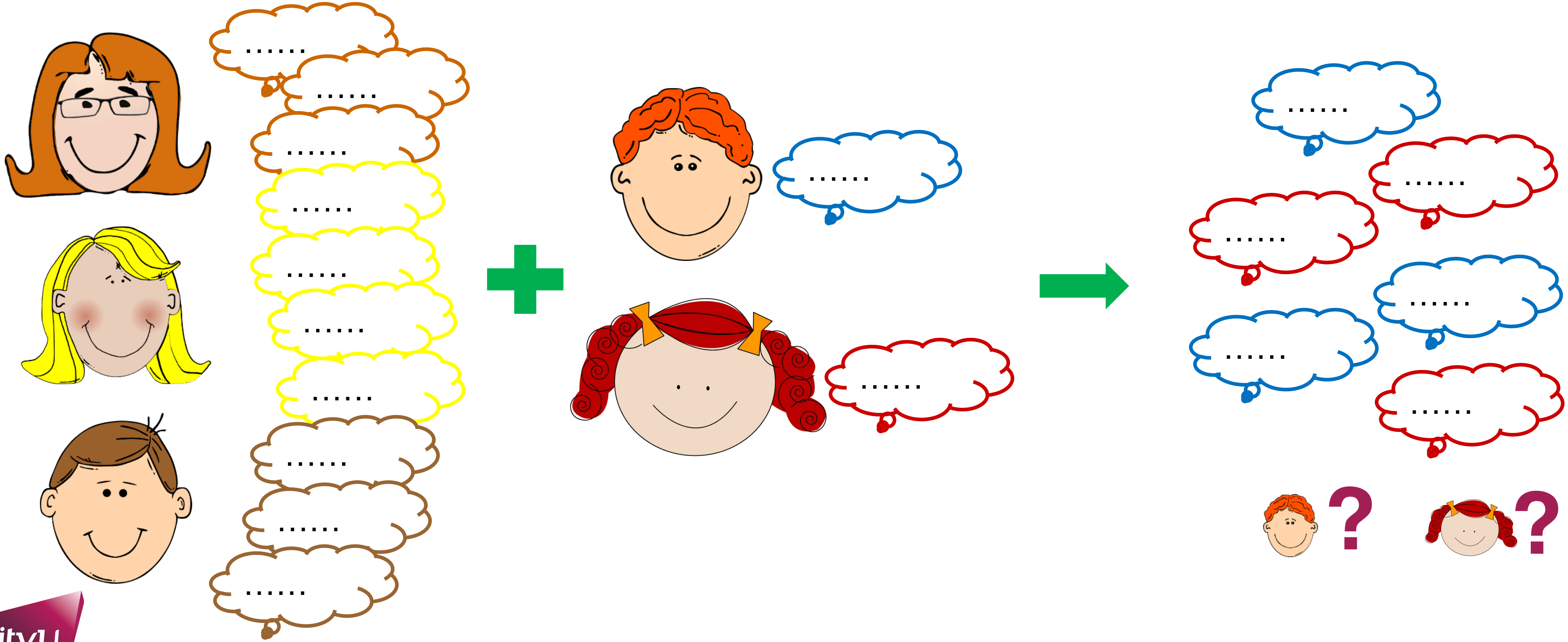
new commands





Problem definition

- Given a training set containing plenty of labelled data and a test set with novel classes containing very limited labelled data, how to learn to recognize the novel classes?

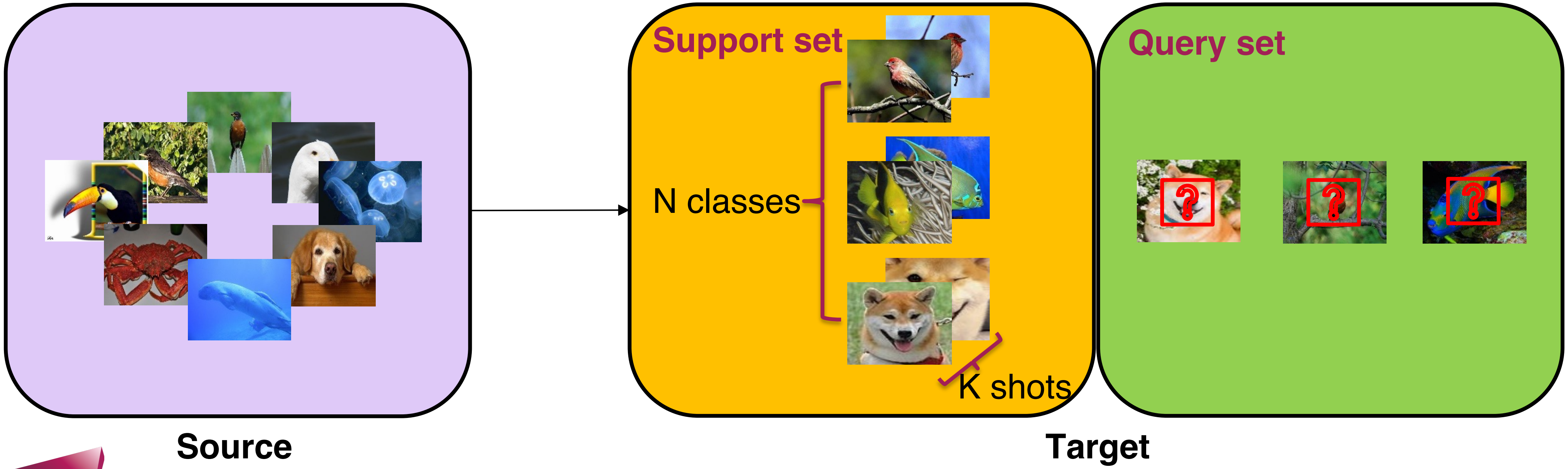


Few-shot learning

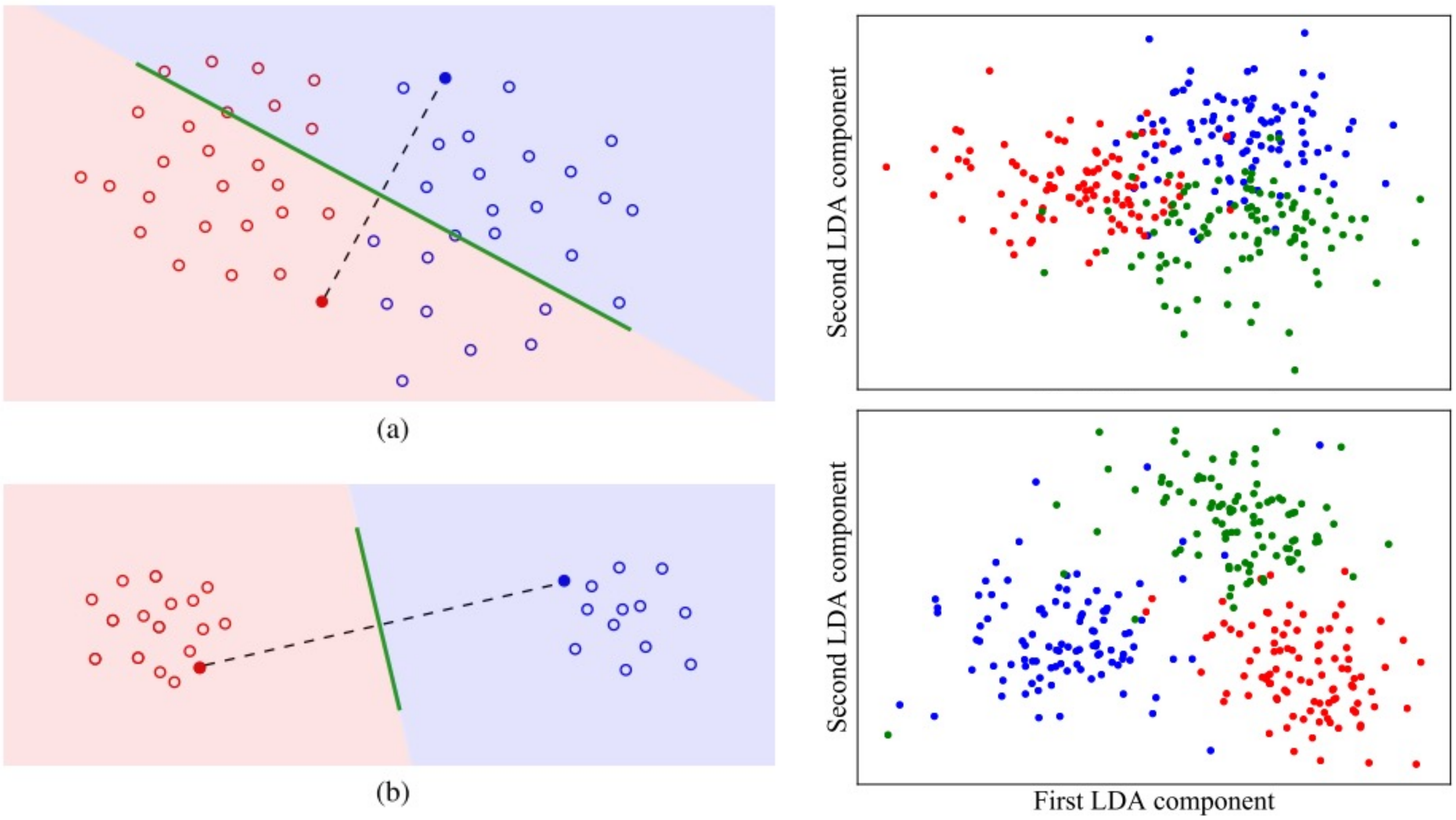
- **Few-Shot Learning (FSL) problem** is a machine learning problem that learns with limited labelled data of the target tasks by incorporating external source data, which has a different distribution from the target data.
- **Few-Shot Learning (FSL) tasks** are a set of tasks, such as few-shot classification, few-shot regression, and few-shot reinforcement learning.
- **Few-Shot Learning (FSL) methods** are a set of methods, which aim to solve the few-shot learning problem.

Few-shot classification

- **Few-Shot Classification** is a few-shot learning task, which is defined as **N-way, K-shot**, where
  - N is the number of classes in the target task
  - K is the number of labelled examples per class



Challenges and our solutions



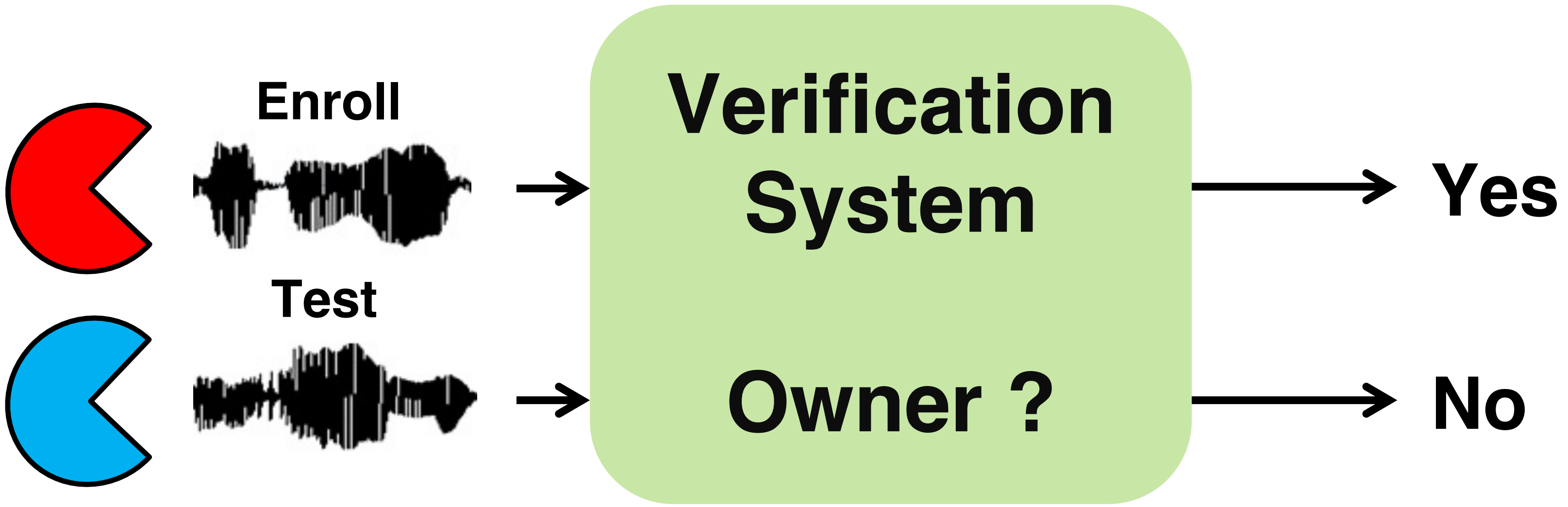
Goldblum, M., Reich, S., Fowl, L., Ni, R., Cherepanova, V., & Goldstein, T. Unraveling Meta-Learning: Understanding Feature Representations for Few-Shot Tasks. *Proceedings of International Conference on Machine Learning (ICML).* 2020.

Part II

Meta-Learning for Small Footprint Text-independent Speaker Verification

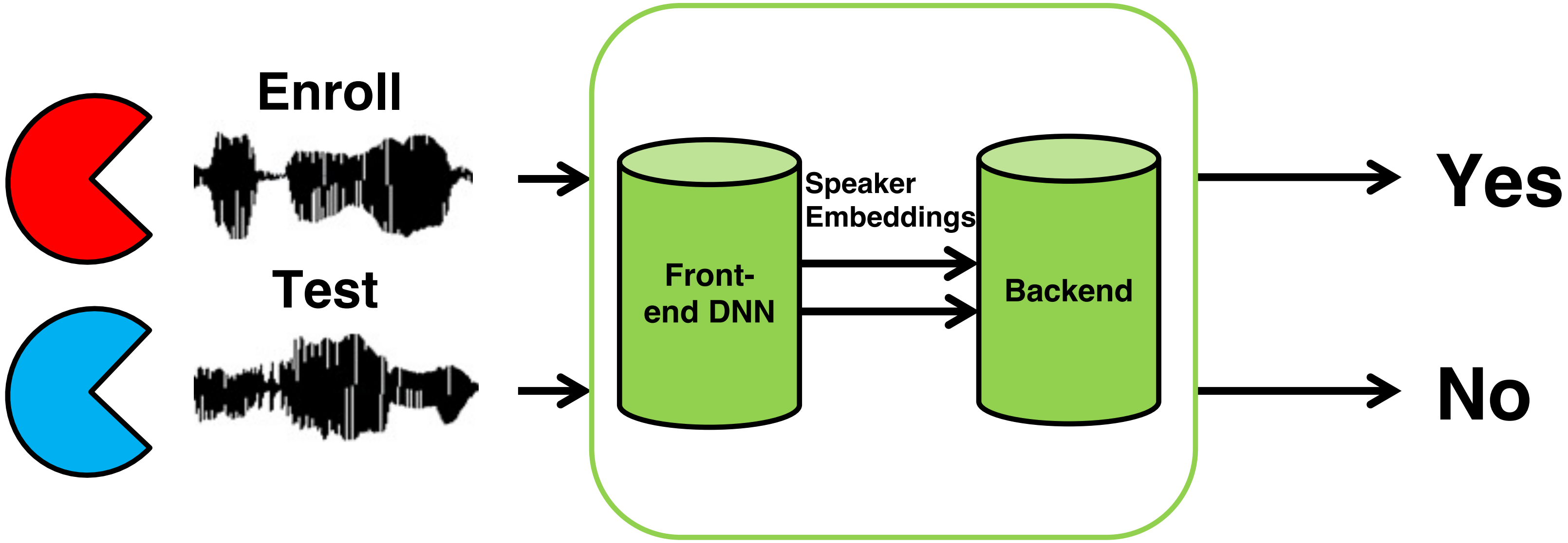
Text-independent speaker verification

- To verify if the test speaker and the enrolled speaker are the same one



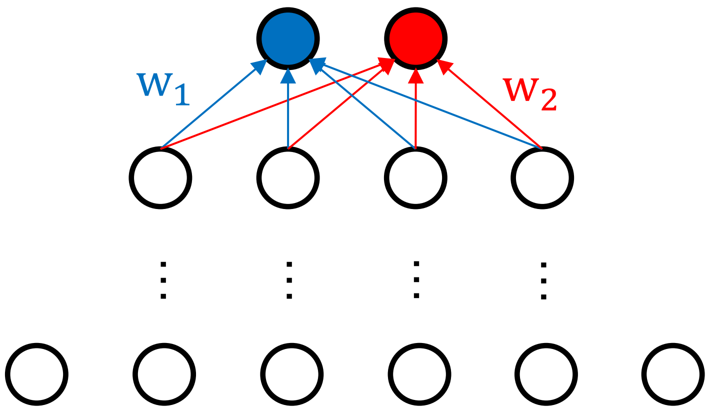
Speaker embedding approach

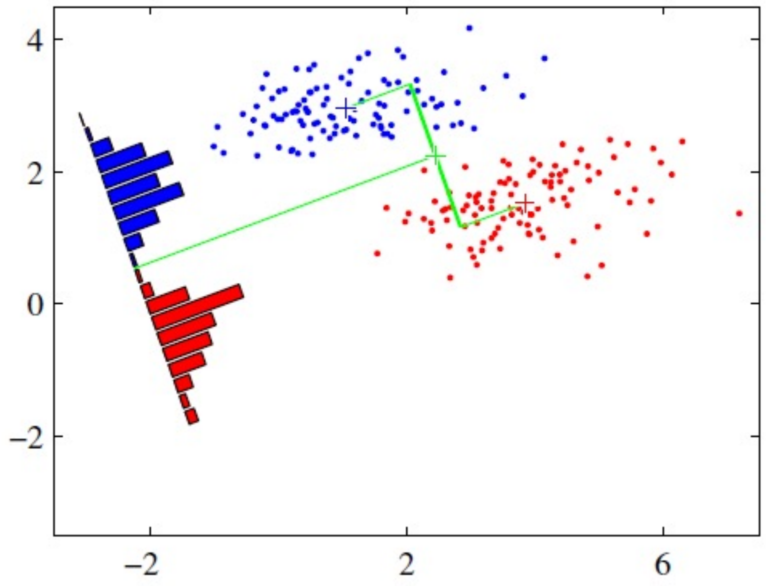
- Front-end DNN for speaker embedding extraction
- Backend for similarity measure



Motivation

- There is a **mismatch of the training objective** between the front-end DNN and the PLDA backend in the speaker embedding approaches.
- Prototypical Networks aim at learning a **non-linear mapping** from the input space to an embedding space with a **predefined distance metric**.
- It is worth to investigate the use of prototypical networks in a small footprint text-independent speaker verification task.





Source: "Pattern Recognition and Machine Learning" by Bishop.

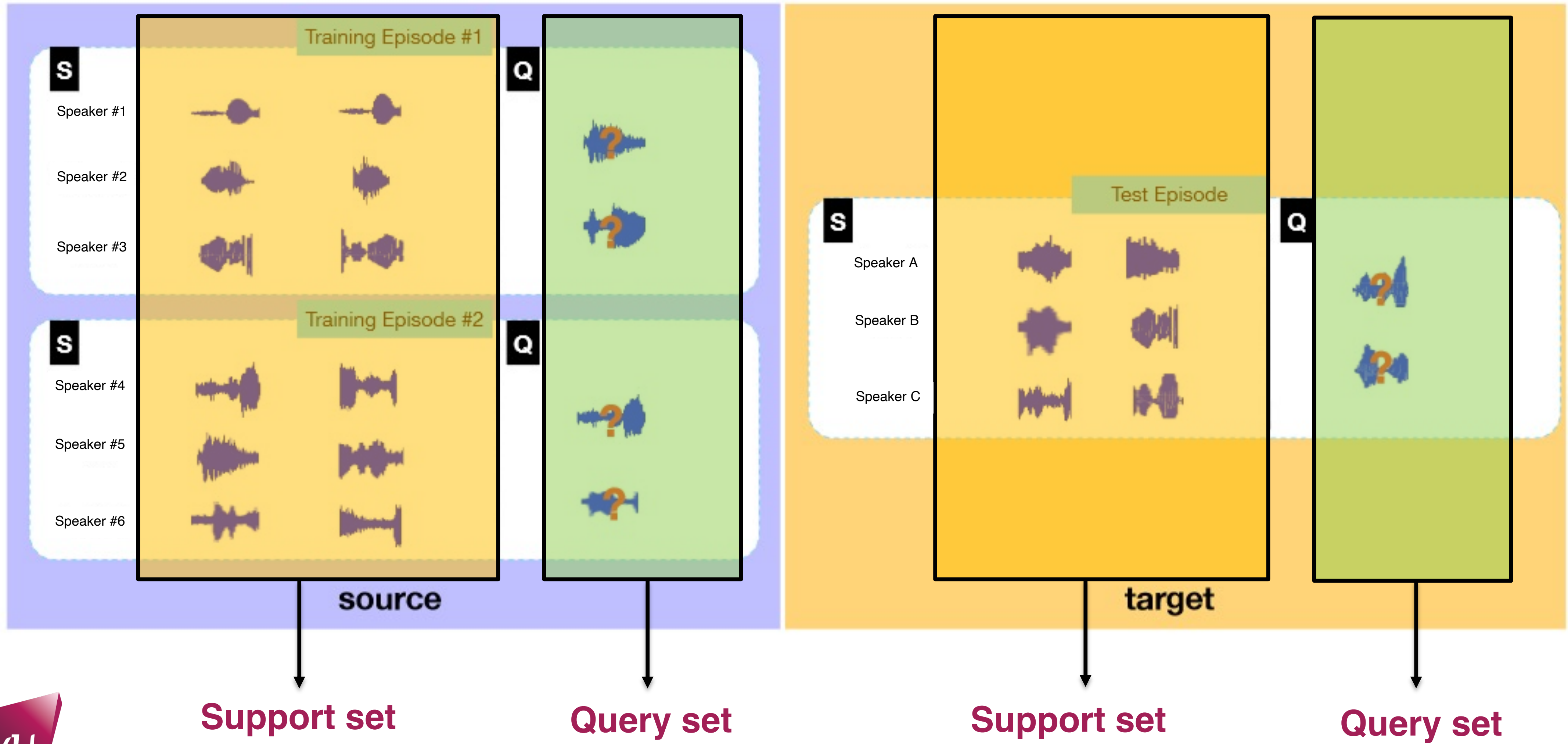
Prototypical Networks

- To train a model which can generalize to new classes not seen in the training set, given only a few examples per new class, needs to **learn a good representation**.
- It tends to minimize the within-class distance and maximize the between-class distance.
- The distance metric can be defined in a flexible way.

Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical networks for few-shot learning.” In: *Advances in neural information processing systems*. 2017. p. 4077-4087.

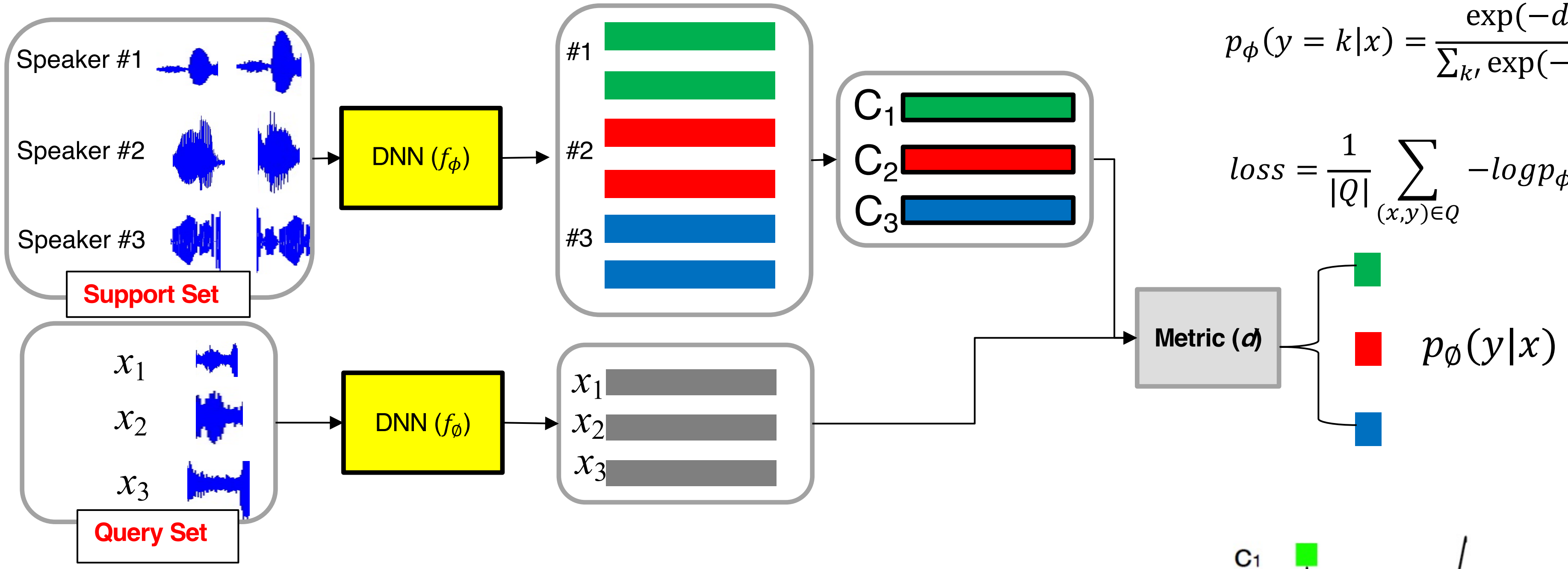
Episodic training in Prototypical Networks

- The model is trained on various meta-tasks and it treats an entire task as a training example.



Prototypical Networks as SV frontend

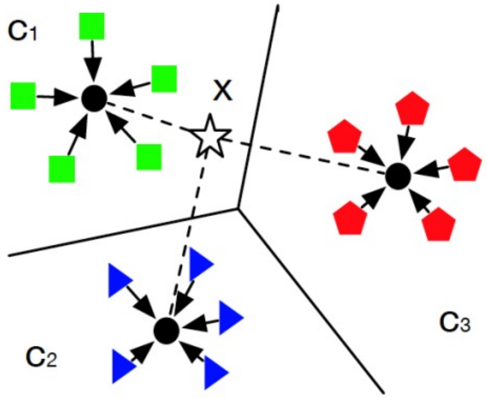
- Support sets are used for computing class centroids.



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

$$loss = \frac{1}{|Q|} \sum_{(x,y)\in Q} -log p_\phi(y|x)$$

Meta-Task #i



The learned DNN will be used as the frontend

Experimental setup

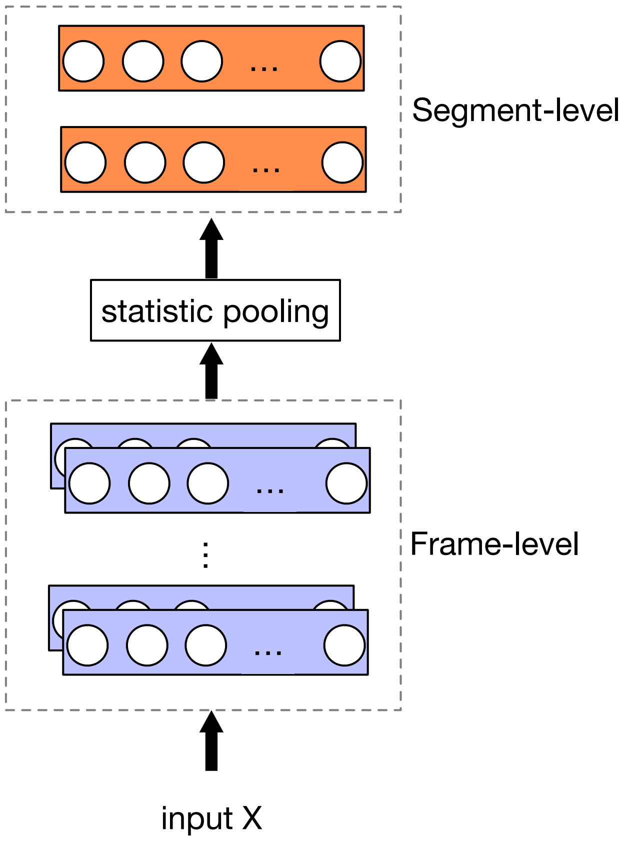
- Training data
  - SWBD dataset: 28k recordings from 2.6k speakers
  - SRE dataset: 35k recordings from 3.8k speakers

  - *4k_full*, *4k_2utt*, *2k_2utt* are sampled to compare the proposed method and the conventional one.
- Evaluation data
  - SRE10

  - Both the enrollment and test utterances are truncated to the first $T \in \{2,5,10,30\}$ seconds of speech, as determined by an energy-based VAD.

Model structure

- A similar model structure as the **X-vector** * approach
- Several layers removed to fulfill the **small footprint** requirement



*David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in IEEE International Conference on Acoustics, Speech and Signal Processing, 2018.

Practical implementation of Prototypical Networks

- Our work has a large number of speakers in each meta-task, which costs a high memory usage. To address this problem, we design an **expectation-maximization (EM) like algorithm** which saves the memory cost and does not affect the performance.
- In the E step, the embeddings of the support set are extracted, and the **class centroids** are estimated.
- In the M step, the embeddings of the query set are extracted, then the **distances and the losses** are estimated.

Baseline

- Conventional learning approach with different backend metrics

Backend Metric	2s-2s	5s-5s	10s-10s	30s-30s
Euclidean	45.85	46.07	45.85	46.48
Cosine	46.14	46.00	46.02	46.76
LDA+Euclidean	41.23	34.54	29.90	23.04
LDA+Cosine	36.66	28.77	21.94	15.32
LDA+PLDA	**34.51**	**26.26**	**18.39**	**12.27**

EER(%) of a conventional front-end with different backend metrics. The front-end models are trained with 2k_2utt training set.

Results

- Prototypical networks with different backend metrics

Front-end Metric	Backend Metric	2s-2s	5s-5s	10s-10s	30s-30s
Euclidean	Euclidean	40.94	34.50	30.06	26.01
Euclidean	LDA + Euclidean	43.66	38.57	33.19	27.29
Euclidean	LDA + PLDA	34.34	25.70	18.62	11.81
Cosine	Cosine	36.07	29.39	25.72	23.17
Cosine	LDA + Cosine	36.88	28.52	21.62	14.94
Cosine	LDA + PLDA	**33.42**	**24.59**	**17.37**	**10.97**

EER(%) of prototypical embeddings (10-shots) on SRE10. The front-end models are trained with 2k_2utt training set.

Results

- Comparing prototypical networks and baseline approach

Training set	System	2s-2s	5s-5s	10s-10s	30s-30s
2k_2utt	Baseline	34.51	26.26	18.39	12.27
	Cosine	**33.42**	**24.59**	**17.37**	**10.97**
4k_2utt	Baseline	33.47	24.98	17.44	11.61
	Cosine	**32.17**	**22.77**	**15.46**	**9.66**
4s_full	Baseline	**29.79**	21.48	13.96	**8.52**
	Cosine	30.14	**21.28**	**13.75**	8.55

EER(%) on SRE10 with various training sets.

Observations

- The prototypical networks are better than the conventional approach when the front-end is directly evaluated with Euclidean or Cosine distance.

- LDA brings negative impact when Euclidean distance is used while it does not bring negative impact to Cosine distance.

- When there are **limited amount of training data per speaker**, prototypical networks perform obviously better than the baseline approach. When the entire training set is used, the two approaches obtain similar performance.

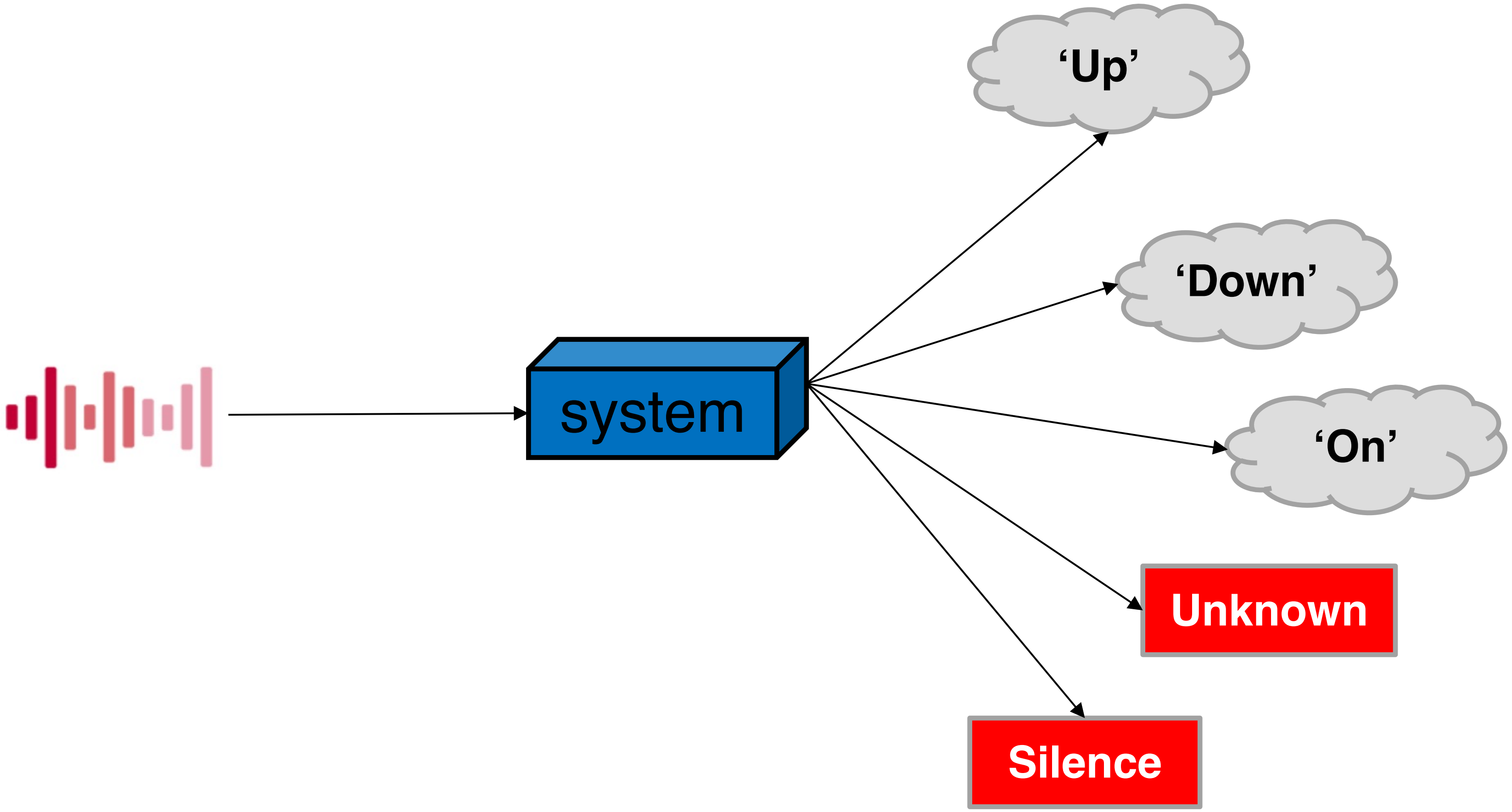
Ko, T., **Chen, Y.,** & Li, Q. Prototypical Networks for Small Footprint Text-independent Speaker Verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2020*, pp. 6804-6808.

Part III

Meta-Learning for Few-Shot Spoken Term Classification

Spoken term classification

- To recognize spoken terms in the voice signal



User-defined spoken term classification

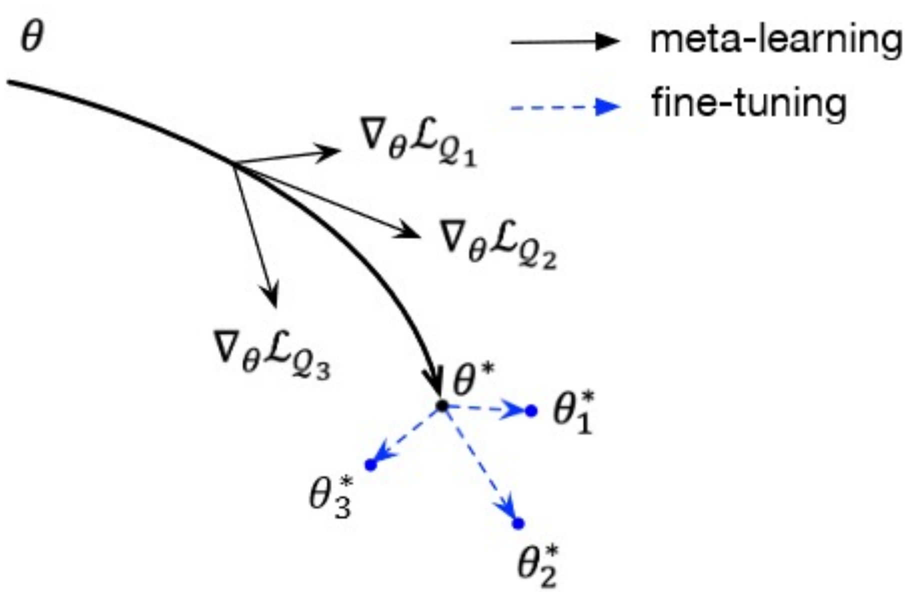
- Normally, the spoken term is predefined.
  - Given plenty of training data, conventional supervised learning could have solved the problem nicely.
- What about a user-defined scenario?
  - Users can define new spoken terms by providing a few audio examples.
- We formulate this problem as a few-shot learning problem, specifically, a few-shot classification task.

Motivation

- We try to build a **personalized command recognition system** for each set of user-defined commands.
- The system should be able to recognize **new commands** using **only a few examples**, while **external sources** can be incorporated during training.
- The characteristics of MAML match the requirements of building the system perfectly, and it is worth to investigate the feasibility of applying few-shot learning methods to speech tasks.

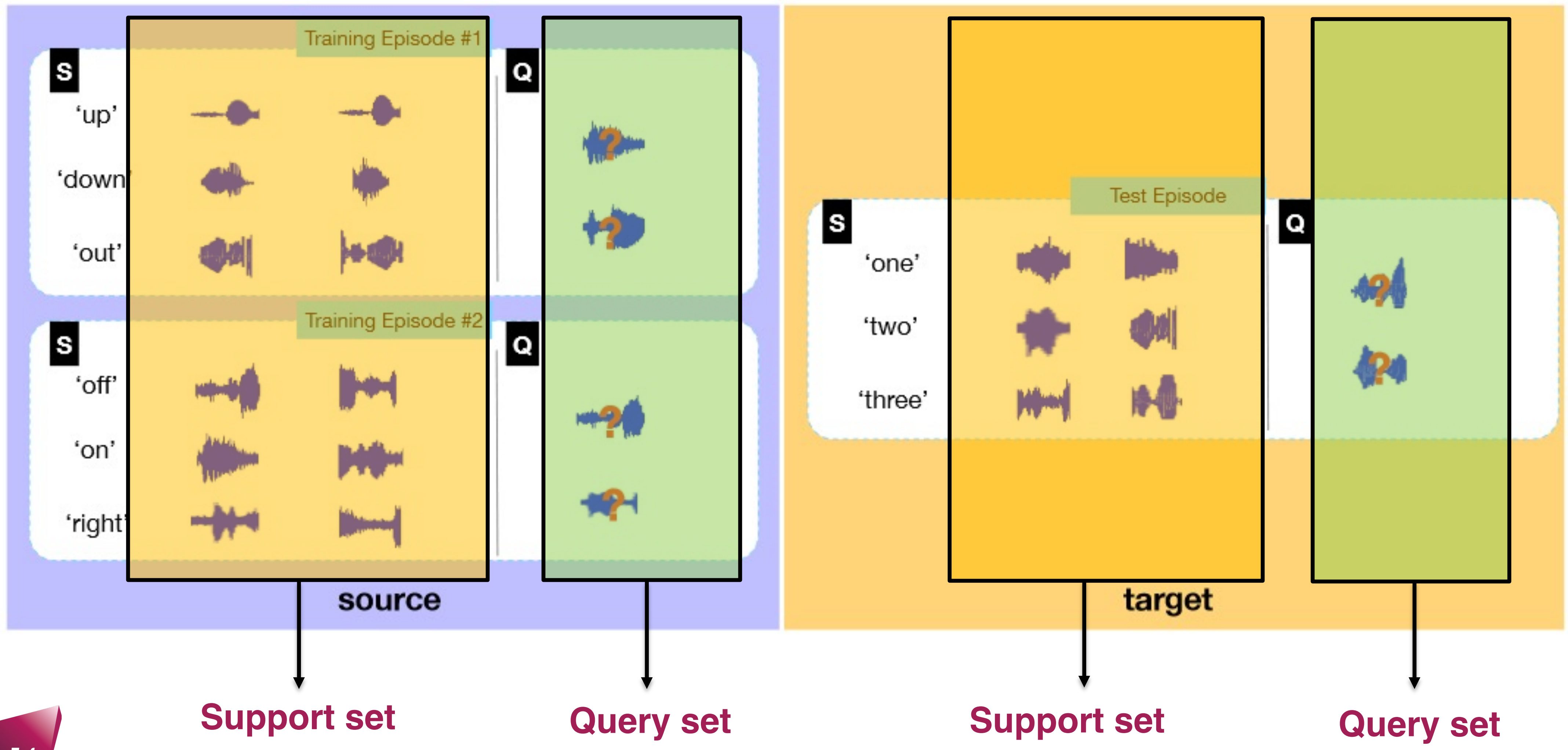
Model-Agnostic Meta-Learning (MAML)

- To train a model which can adapt to any new task using only a few labelled examples.
- The model is trained on various tasks (meta-tasks) and it treats the entire task as a training example.
- The model is forced to face different tasks so that it can get used to adapting to new tasks.



Chelsea Finn, Pieter Abbeel, Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,”in Proceedings of the 34th International Conference on Machine Learning (ICML). JMLR. 2017, pp. 1126–1135.

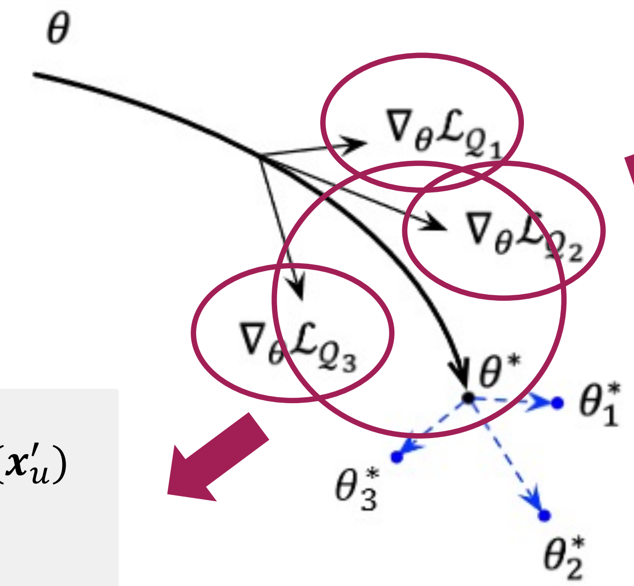
Episodic training in MAML



MAML – the meta-learning stage

$$\mathcal{L}_{S_i}(f_\theta) = -\sum_{(\boldsymbol{x}_j, \boldsymbol{y}_j)\in S_i} \boldsymbol{y}_j log f_\theta(\boldsymbol{x}_j)$$

$\theta_i' = \theta - \alpha\nabla_\theta\mathcal{L}_{S_i}(f_\theta)$ **inner loop**

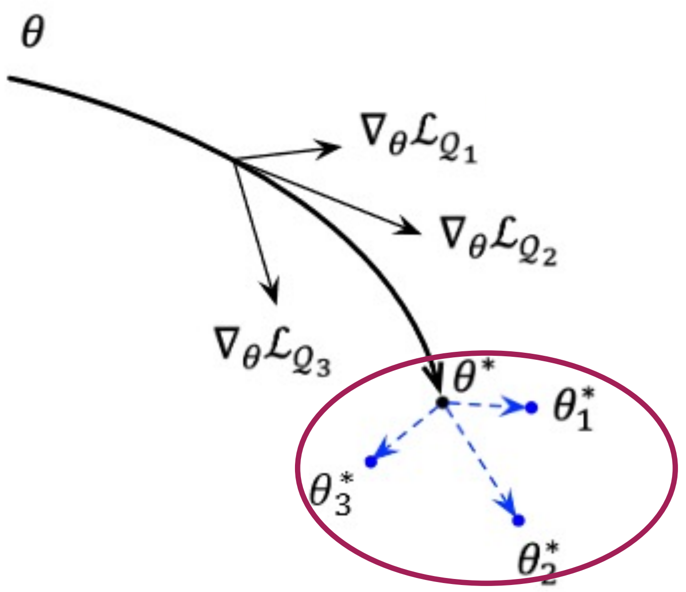


$$\mathcal{L}_{Q_i}\left(f_{\theta_i'}\right) = -\sum_{(\boldsymbol{x}_u', \boldsymbol{y}_u')\in Q_i} \boldsymbol{y}_u' log f_{\theta'}(\boldsymbol{x}_u')$$

$\theta^* \leftarrow \theta - \beta\nabla_\theta\sum_i \mathcal{L}_{Q_i}(f_{\theta_i'})$ **outer loop**

MAML – the fine-tuning stage

- Before evaluation, the model will be fine-tuned for a few iterations:



$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{S_i}(f_\theta)$$

Extend the few-shot classification problem

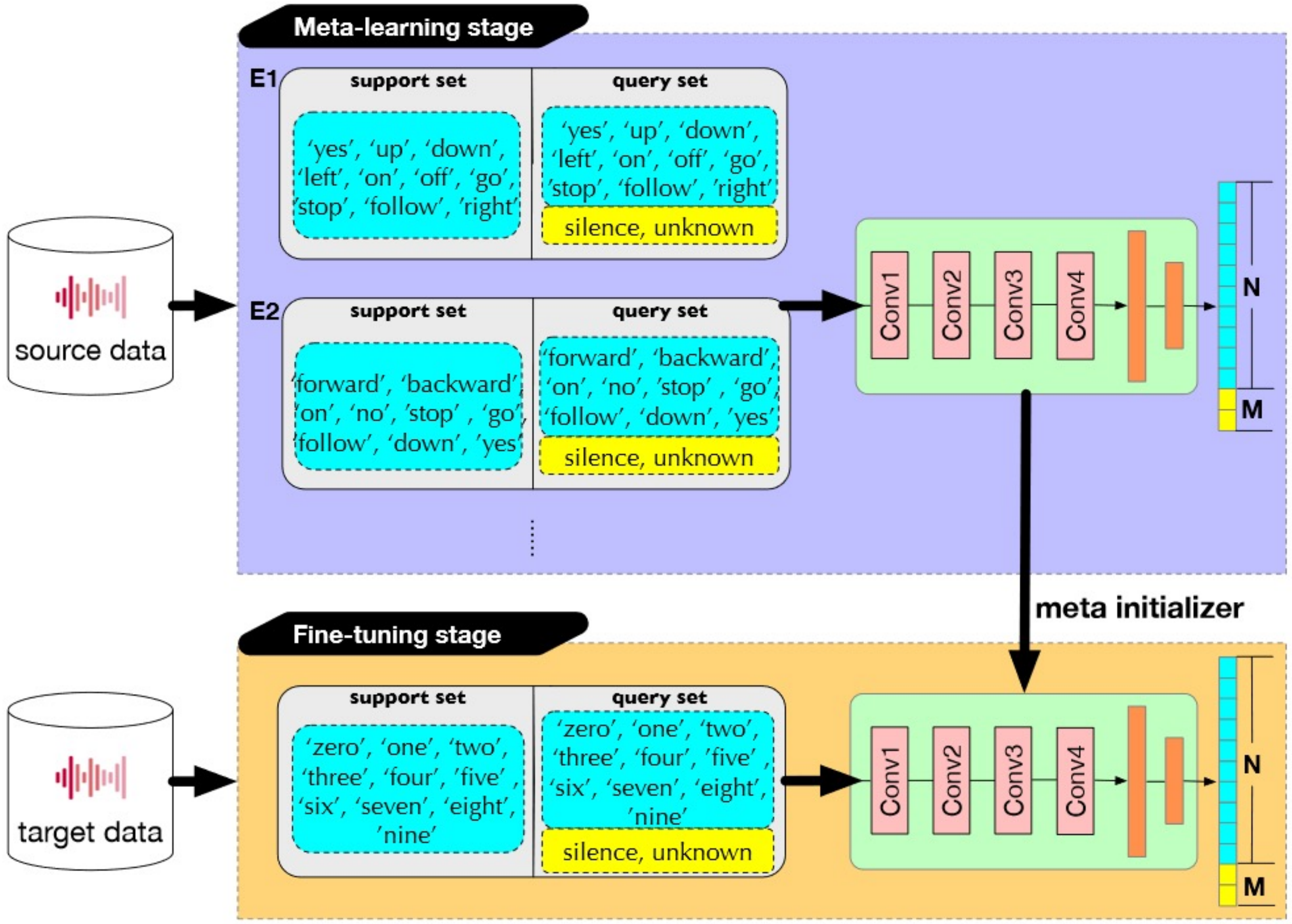
- In most few-shot studies, all the classes are assumed to be **new**.
- In real-life applications, some of the classes are **known**.
- We define an N+M-way, K-shot problem where
  - M is the number of **fixed** classes
  - N is the number of **new** classes in the target task
  - K is the number of examples of each **new** class

Our approach – extended MAML

- We **fix the output positions** of the fixed classes in the neural network classifier.
- The fixed classes occur in **every meta-task** in the meta-learning stage.
- The adaptation of fixed classes is not needed in the fine-tuning stage as they have already been learned in the meta-learning stage.

Few-shot spoken term classification

- 10+2-way, K-shot
- 10 keywords
- 2 fixed class: **silence** and **unknown**
- In the meta-learning stage, meta-tasks are randomly formed from a pool of keywords.



Framework of our extended-MAML approach for few-shot spoken term classification.

The algorithm

Algorithm 1 extended-MAML approach for few-shot spoken term classification

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\mathcal{X}$: training keywords set
Require: $\mathcal{S}_{il}$: silence class set, $\mathcal{U}_{nk}$: unknown class set
Require: $\mathcal{S}_i$: support set, $\mathcal{Q}_i$: query set
Require: α, β: learning rates

1: Randomly initialize base model parameters $\boldsymbol{\theta}$
2: **while** not done **do**
3:     Sample a batch of meta-tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4:     **for all** $\mathcal{T}_i$ **do**
5:         Sample a support set $\mathcal{S}_i$ from $\mathcal{X}$
6:         Compute the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$ using $\mathcal{S}_i$ and $\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$
7:         Update base model parameters with gradient descent: $\boldsymbol{\theta}'_i = \boldsymbol{\theta} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$ ▷ step 6 and step 7 can be repeated for several times
8:         Sample a query set $\mathcal{Q}_i$ from the union $\{\mathcal{X}, \mathcal{S}_{il}, \mathcal{U}_{nk}\}$ ▷ selected keywords from $\mathcal{X}$ in $\mathcal{Q}_i$ and $\mathcal{S}_i$ within $\mathcal{T}_i$ are the same
9:         Compute the loss $\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i})$ using $\mathcal{Q}_i$ and the updated model $f_{\boldsymbol{\theta}'}$
10:     **end for**
11:     Update parameters $\boldsymbol{\theta}$ using each $\mathcal{Q}_i$ and $\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'})$: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta\nabla_{\boldsymbol{\theta}}\sum_i \mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i})$
12: **end while**

Experimental setup

- Google Speech Commands dataset (v0.02)
- 105,829 1-second audio clips of 35 keywords
- We formulate two 10+2-way, K-shot tasks
  - ten keywords, silence, and unknown
  - **Digits classification**, which uses digits zero to nine as ten keywords
  - **Commands classification**, which contains ten keywords as: “yes”, “no”, “up”, “down”, “left”, “right”, “on”, “off”, “stop”, or “go”

Model setup

- 40-dimensional MFCCs
- CNN based model containing 4 convolutional blocks
- Each block comprises a 3 x 3 convolutions and 64 filters

Baselines

- Two baselines:
  - Conventional supervised learning approach
  - Original MAML (which treats the 10+2-way problem as a 12-way problem)

Results

- Few-shot digits classification

Methods	1-shot	5-shot	10-shot
Super. L	18.14 ± 0.44	24.83 ± 0.38	28.07 ± 0.34
MAML-ori	44.60 ± 0.98	60.88 ± 0.58	65.18 ± 0.62
MAML-ext	**47.42 ± 0.96**	**63.22 ± 0.71**	**69.48 ± 0.47**

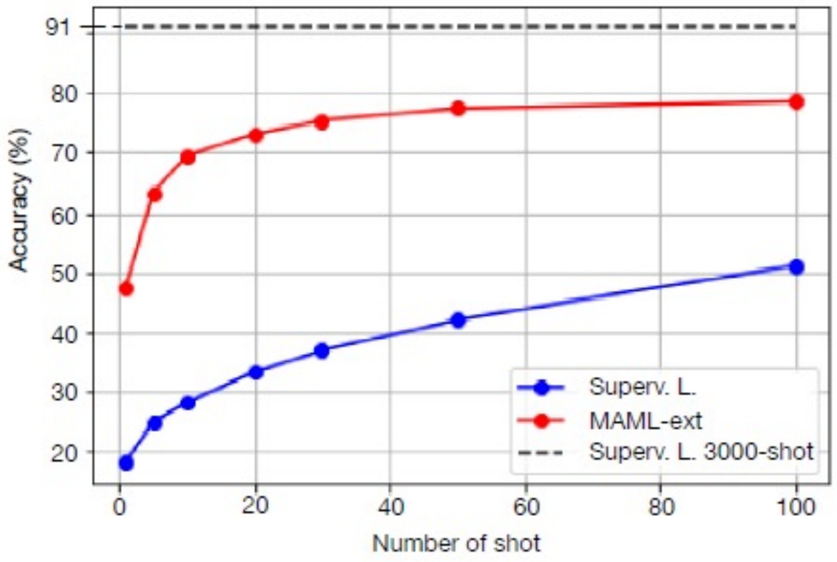
*Accuracy with 95% confidence intervals on **digits classification**.*

- Few-shot commands classification

Methods	1-shot	5-shot	10-shot
Super. L	17.03 ± 0.48	22.42 ± 0.33	25.60 ± 0.26
MAML-ori	33.35 ± 0.80	50.31 ± 0.50	57.34 ± 0.41
MAML-ext	**39.54 ± 0.62**	**52.20 ± 0.51**	**59.36 ± 0.39**

*Accuracy with 95% confidence intervals on **commands classification**.*

User-defined vs. predefined



*Accuracy with changing shots on **digits classification**.*

Observations

- The overall accuracy in digit classification is better than in command classification.
- MAML based approaches perform much better than conventional supervised learning in a few-shot situation.
- Our proposed approach outperforms the original MAML.
  - We attribute the improvement to the use of **prior information of the fixed classes**.
- There is a performance gap between **few-shot learning** and **many-shot learning**.

Chen, Y., Ko, T., Shang, L., Chen, X., Jiang, X.**,** & Li, Q. An Investigation of Few-Shot Learning in Spoken Term Classification. In *INTERSPEECH 2020*, pp. 2582-2586.

Part IV

Improved Meta-Learning with Interpolation-based Consistency Regularization

Motivation

- Applications in smart voice control systems prove that meta-learning is an effective solution to address the few-shot learning problem.
- There exist weaknesses in current meta-learning algorithms, especially in **forming generalizable decision boundaries** (i.e., meta-overfitting).
- We aim to propose a **regularization technique** to solve the **meta-overfitting** problem.

The meta-overfitting problem

- Conventional meta-learning algorithms may face meta-overfitting problems, which form a decision boundary ***staying too close*** to the limited labelled examples in ***the few-shot tasks***.

expected risk: $R(h) = \int \ell(h(x), y)\, dp(x, y) = \mathbb{E}[\ell(h(x), y)]$

empirical risk: $R_I(h) = \frac{1}{I} \sum_{i=1}^{I} \ell(h(x_i), y_i)$

mixup – an interpolation-based regularization method

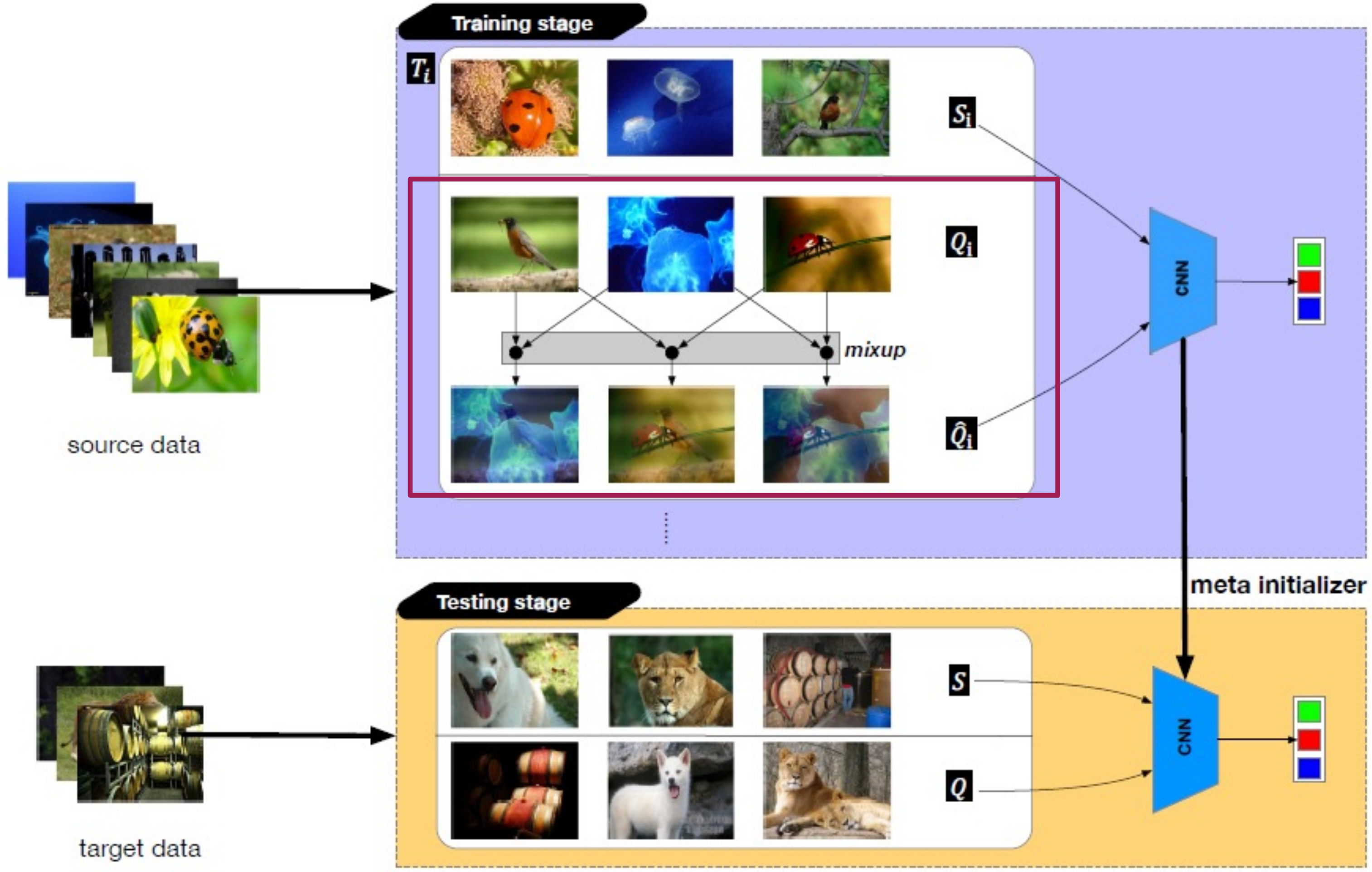
- Empirical Risk Minimization allows large neural networks to ***memorize*** (instead of ***generalize*** from) the training data [1].
- *mixup* encourages the model to behave linearly in-between training examples, which reduces the amount of undesirable oscillations when predicting outside the training examples.
- We have adopted *mixup* in **semi-supervised learning** [2] and **unsupervised domain adaptation** [3].

$$\hat{x}_z = \lambda x_m + (1 - \lambda) x_n$$

$$\hat{y}_z = \lambda y_m + (1 - \lambda) y_n$$

[1] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations (ICLR) 2018*.
[2] Ma, Y., Mao, X., **Chen, Y.,** & Li, Q. Mixing Up Real Samples and Adversarial Samples for Semi-Supervised Learning. International Joint Conference on Neural Networks (IJCNN), IEEE, 2020.
[3] Mao, X., Ma, Y., Yang, Z**., Chen, Y.,** & Li, Q. (2019). Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*.

MetaMix – our methodology



Algorithm 1 MetaMix with MAML

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\mathcal{S}_i$: support set; $\mathcal{Q}_i$: query set
Require: α, β : learning rate
Require: $\breve{\alpha}$: Beta distribution parameter
Require: $mix_\lambda(a,b) = \lambda a + (1-\lambda)b, \lambda \sim \mathrm{B}(\breve{\alpha}, \breve{\alpha})$

Randomly initialize model parameters θ
while not done **do**
  Sample a batch of episodes $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample a support set $\mathcal{S}_i = \{(x_j, y_j)\}_{j=1}^{J}$
    Evaluate $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$ using $\mathcal{S}_i$ and $\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$
    Sample a query set $\mathcal{Q}_i = \{(x_z, y_z)\}_{z=1}^{Z}$
    Randomly select pairs of examples $\{(x_m, y_m)\}_{m=1}^{Z}, \{(x_n, y_n)\}_{n=1}^{Z}$ from $\mathcal{Q}_i$
    $\hat{x}_z = mix_\lambda(x_m, x_n), \hat{y}_z = mix_\lambda(y_m, y_n)$
    Get new query set $\hat{\mathcal{Q}}_i = \{(\hat{x}_z, \hat{y}_z)\}_{z=1}^{Z}$
  end for
  Update $\theta \leftarrow \theta - \beta \cdot \nabla_\theta \sum_i \mathcal{L}_{\hat{\mathcal{Q}}_i}(f_{\boldsymbol{\theta}_i'})$
end while

MetaMix – our methodology

- We generate virtual examples only from the query set for two reasons:
  - The query set is responsible for optimizing the **meta-objective** across different training episodes, which is significant to the generalization of the learned initializer.
  - Virtual examples generated by interpolating examples from the query set are expected to better approximate the **real data distribution**.

Experimental setup

- Dataset
  - *mini*-ImageNet
    - 100 classes, 600 84 × 84 colored images per class, 64 training / 16 validation / 20 testing.
  - Caltech-UCSD Birds-200-2011 (CUB)
    - 200 classes, 11,788 84 × 84 colored images in total, 100 training / 50 validation / 50 testing.
  - Fewshot-CIFAR100 (FC100)
    - 100 classes, 600 32 × 32 colored images per class, 60 training / 20 validation / 20 testing.

CityU

Model setup

- Baselines
  - Prototypical Networks, Matching Network, Relation Network
  - MAML, First-Order MAML (FOMAML), Meta-SGD, Meta-Transfer Learning (MTL)
- Backbone model
  - Shallow CNN with 4 convolutional blocks (Conv([32, 3, 3])+ReLU+BN+MaxPooling([2, 2]))
  - ResNet-12 (in MTL)

Results

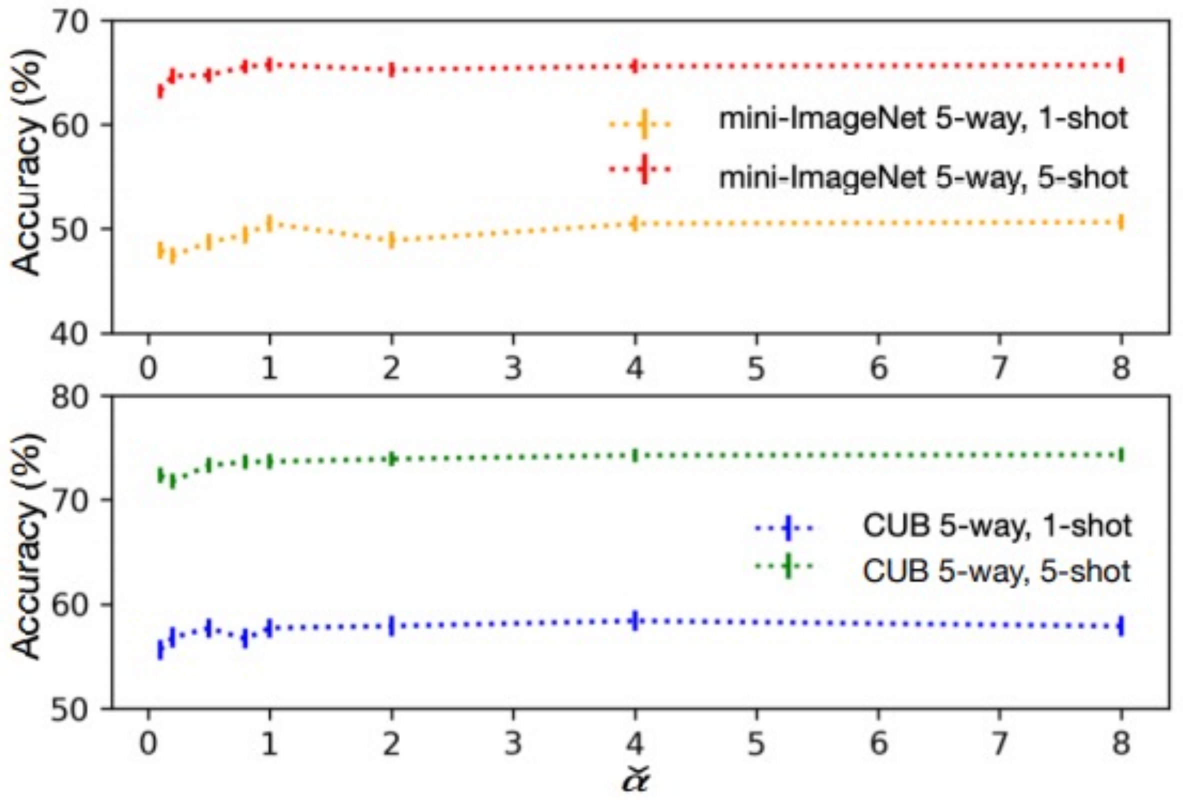
- Comparison with baselines

	mini-ImageNet		CUB		FC100	
Models	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Matching Network	50.47 ± 0.80	64.83 ± 0.67	57.70 ± 0.87	71.42 ± 0.71	36.97 ± 0.67	49.44 ± 0.71
Prototypical Network	49.33 ± 0.82	65.71 ± 0.67	51.34 ± 0.86	67.56 ± 0.76	36.83 ± 0.69	51.21 ± 0.74
Relation Network	50.48 ± 0.80	65.39 ± 0.72	59.47 ± 0.96	73.88 ± 0.74	36.40 ± 0.69	51.35 ± 0.69
MAML	48.18 ± 0.78	63.05 ± 0.71	54.32 ± 0.91	71.37 ± 0.76	35.96 ± 0.71	48.06 ± 0.73
MetaMix+MAML	**50.51 ± 0.86**	**65.73 ± 0.72**	**57.70 ± 0.92**	**73.66 ± 0.74**	**37.09 ± 0.74**	**49.31 ± 0.72**
FOMAML	45.22 ± 0.77	60.97 ± 0.70	53.12 ± 0.93	70.90 ± 0.75	34.97 ± 0.70	47.41 ± 0.73
MetaMix+FOMAML	**47.78 ± 0.77**	**63.55 ± 0.70**	**54.81 ± 0.97**	**72.90 ± 0.74**	**36.48 ± 0.67**	**49.48 ± 0.71**
MetaSGD	49.93 ± 1.73	64.01 ± 0.90	56.19 ± 0.92	69.14 ± 0.75	36.36 ± 0.66	49.96 ± 0.72
MetaMix+MetaSGD	**50.60 ± 1.80**	**64.47 ± 0.88**	**57.64 ± 0.88**	**70.50 ± 0.70**	**37.44 ± 0.71**	**51.41 ± 0.69**
MTL	61.37 ± 0.82	78.37 ± 0.60	71.90 ± 0.86	84.68 ± 0.53	42.17 ± 0.79	56.84 ± 0.75
MetaMix+MTL	**62.74 ± 0.82**	**79.11 ± 0.58**	**73.04 ± 0.86**	**86.10 ± 0.50**	**43.58 ± 0.73**	**58.27 ± 0.73**

*Accuracy with 95% confidence intervals of **5-way, K-shot (K=1, 5)** classification tasks on **mini-ImageNet, CUB,** and **FC100** datasets.*

Results

- Analysis of hyper-parameter in Beta distribution



Effect of Beta distribution. $\breve{\alpha}$ is set to 0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 4.0, 8.0.

Results

- Ablation study

	mini-ImageNet		CUB	
Set(s)	1-shot	5-shot	1-shot	5-shot
Q	**50.51 ± 0.86**	**65.73 ± 0.72**	**57.70 ± 0.92**	**73.66 ± 0.74**
S	47.87 ± 0.82	62.34 ± 0.65	54.39 ± 0.97	67.23 ± 0.74
Q+S	48.36 ± 0.81	64.06 ± 0.72	54.32 ± 0.93	70.30 ± 0.75
w/o mixup	48.18 ± 0.78	63.05 ± 0.71	54.32 ± 0.91	71.37 ± 0.76

An ablation study of doing mixup on different sets. Q denotes the query set and S denotes the support set.

Results

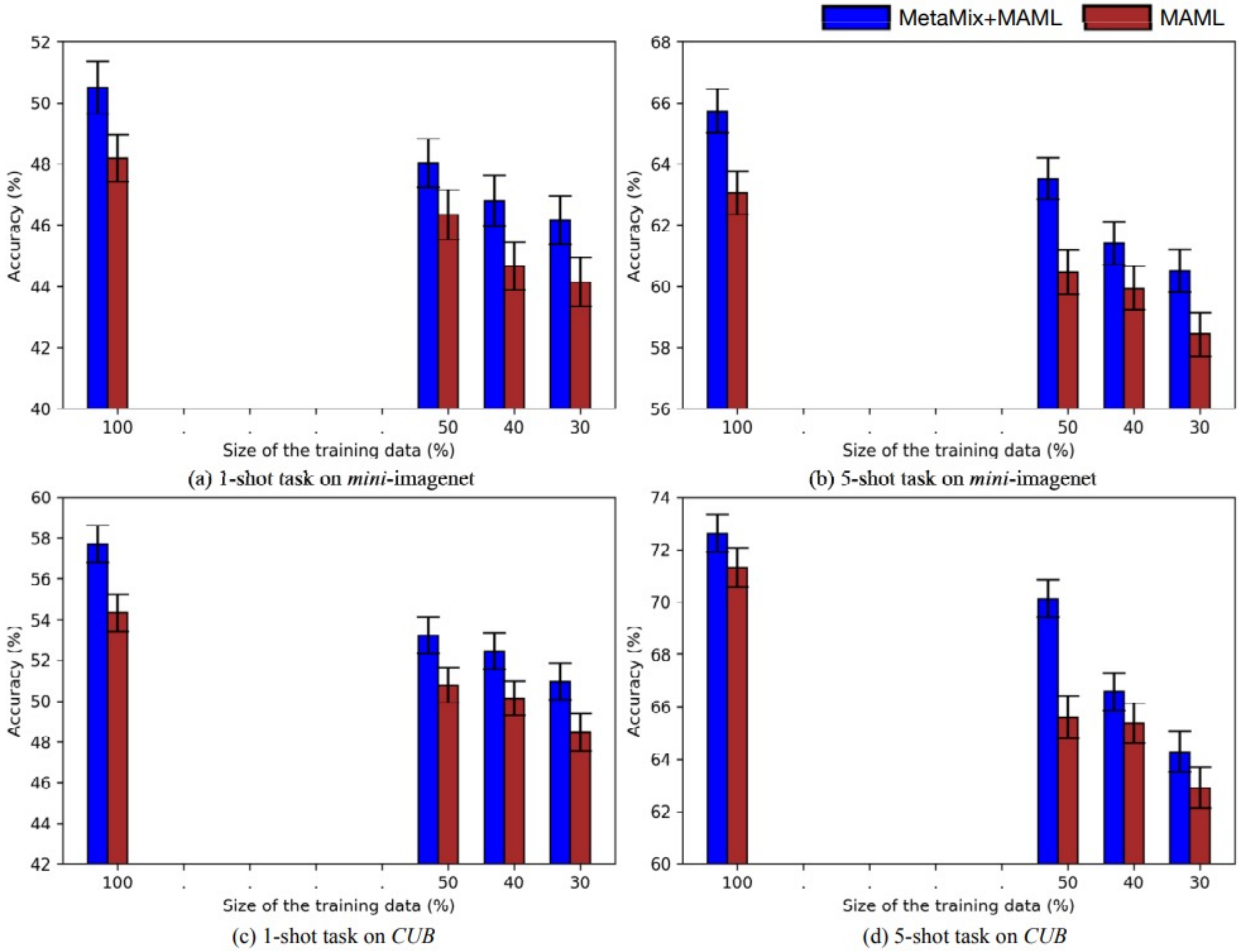
- Analysis of the effect of the size of training data

	mini-ImageNet		CUB		FC100	
Set(s)	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML(100%)	48.18 ± 0.78	63.05 ± 0.71	54.32 ± 0.91	71.37 ± 0.76	35.96 ± 0.71	48.06 ± 0.73
MetaMix+MAML(100%)	**50.51 ± 0.86**	**65.73 ± 0.72**	**57.70 ± 0.92**	**73.66 ± 0.74**	**37.09 ± 0.74**	**49.31 ± 0.72**
MAML(50%)	46.34 ± 0.82	60.47 ± 0.73	50.78 ± 0.86	65.60 ± 0.81	35.38 ± 0.71	47.93 ± 0.78
MetaMix+MAML(50%)	**48.04 ± 0.79**	**63.52 ± 0.67**	**53.22 ± 0.91**	**70.13 ± 0.70**	**36.35 ± 0.74**	**48.11 ± 0.69**

*A comparison between using 100% and 50% training data; accuracy with 95% confidence intervals of **5-way, K-shot (K=1, 5)** classification tasks on **mini-ImageNet**, **CUB**, and **FC100** datasets.*

Results

- Analysis of the effect of the size of training data



A comparison among using 100%, 50%, 40%, and 30% of the training data.

Observations

- MetaMix improves the performance of all MAML-based algorithms over three datasets; meanwhile, MetaMix with MTL achieves state-of-the-art performance.

- When $\breve{\alpha}$ is below 1.0, the accuracy is a little lower. When $\breve{\alpha}$ is 1.0 and above, the performance maintains a good level.

- Mixing examples from only the query set performs best, compared with mixing examples from only the support set and mixing examples from both the support set and the query set.

- MetaMix performs more robust with the reduction of the size of the training data.

Chen, Y., Ma, Y., Ko, T., Wang, J.**,** & Li, Q. Improved Meta-Learning with Interpolation-based Consistency Regularization. In *International Conference on Pattern Recognition (ICPR) 2020*.

Part V

Conclusion and Future Work

Conclusion

- We investigate the use of **Prototypical Networks** in a **small footprint text-independent speaker verification task**. It outperforms the conventional method, especially when there are a limited amount of training data per speaker.

- We extend the original **Model-Agnostic Meta-Learning(MAML)** algorithm to solve an **N+M-way, K-shot** problem and apply it to a **user-defined spoken term classification** task. It achieves better performance than the original MAML and conventional supervised method.

- We propose an improved meta-learning approach with the **interpolation-based consistency regularization** technique. It improves the performance of MAML-based algorithms and achieves state-of-the-art results when integrated with Meta-Transfer Learning. MetaMix is less sensitive to the reduction of the source training data, compared to MAML and its variants.

CityU

Future work – smart voice control systems

- Our proposed method for speaker verification still relies on the PLDA backend to achieve competitive results. We will look for other **learnable distance metrics** which can **facilitate PLDA's performance**.

- There is a performance gap between our user-defined system and a predefined one. In the future, we will try to **narrow the gap** by improving the algorithm and augmenting the data.

- We will find more tasks that **need a quick adaptation** in smart voice control systems and apply improved meta-learning algorithms to them.

Future work – meta-learning algorithms

- Quite a few works make a thorough analysis of meta-learning **theoretically**. In the future, we will do more study about **why and how** meta-learning can achieve better results than other few-shot learning methods.

- It is not analyzed about on which conditions meta-learning works. In the future, we will make **more comparisons on different conditions**, such as differences in the size of the source data, backbone models, and domains of the tasks.

- The N-way, K-shot is not a perfect setting, because both are changing in practical applications. In the future, we will redefine a few-shot classification task with **varying N and K**.

Thank you!

Email: robinchen2-c@my.cityu.edu.hk
Github: https://github.com/Codelegant92

CityU