Few-Shot Learning

Student: PhD(c) Gabriel Pirlogeanu

Coordinator: Assoc. Prof. Ana-Antonia Neacsu

Overview

- Introduction
- Meta Learning
- Few-Shot
- Few-Shot Algorithms
- Few-Shot Applications
- Conclusion

Classical Machine Learning

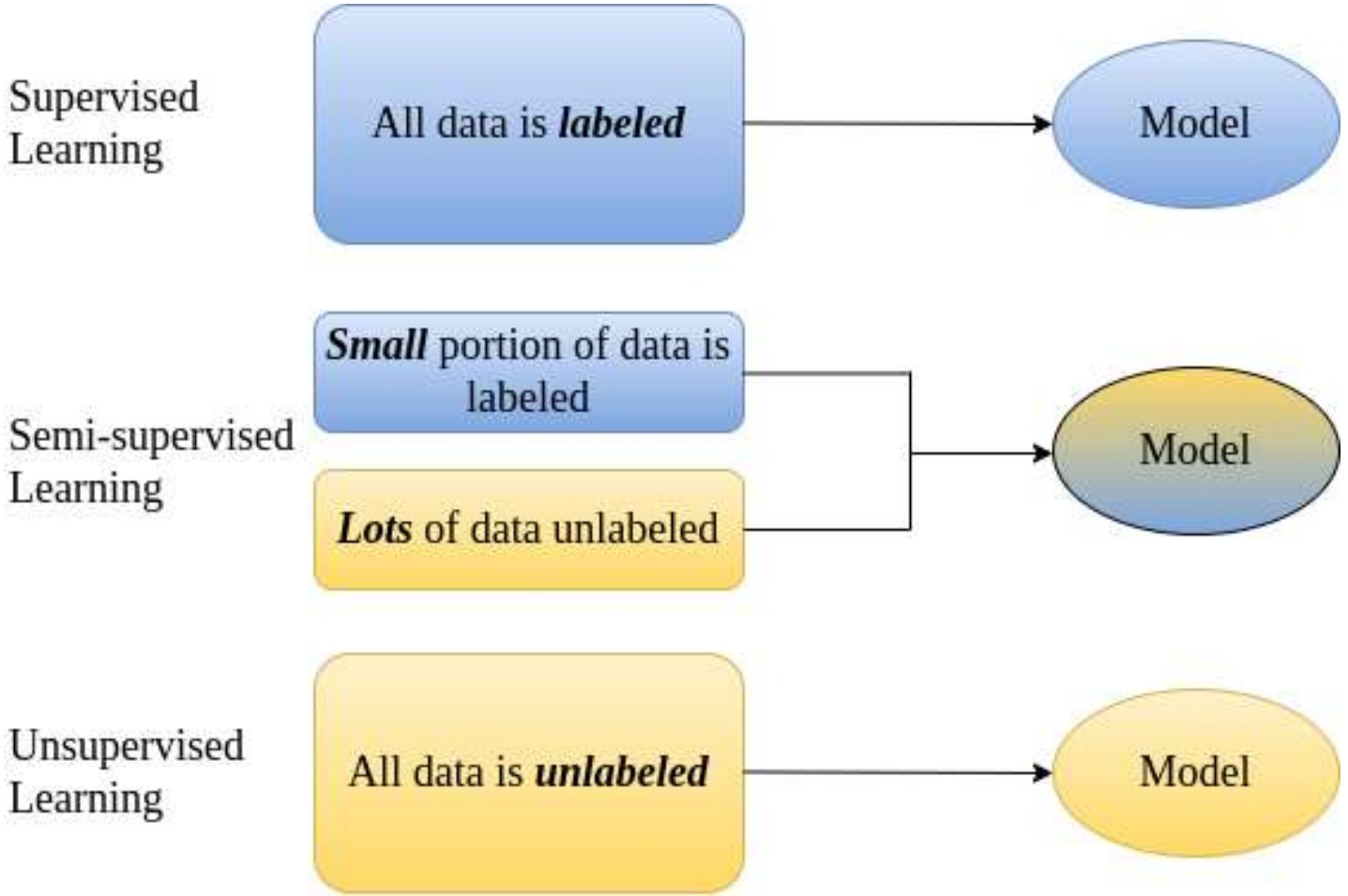


Image from: https://xpertnest.com/?x=276817816

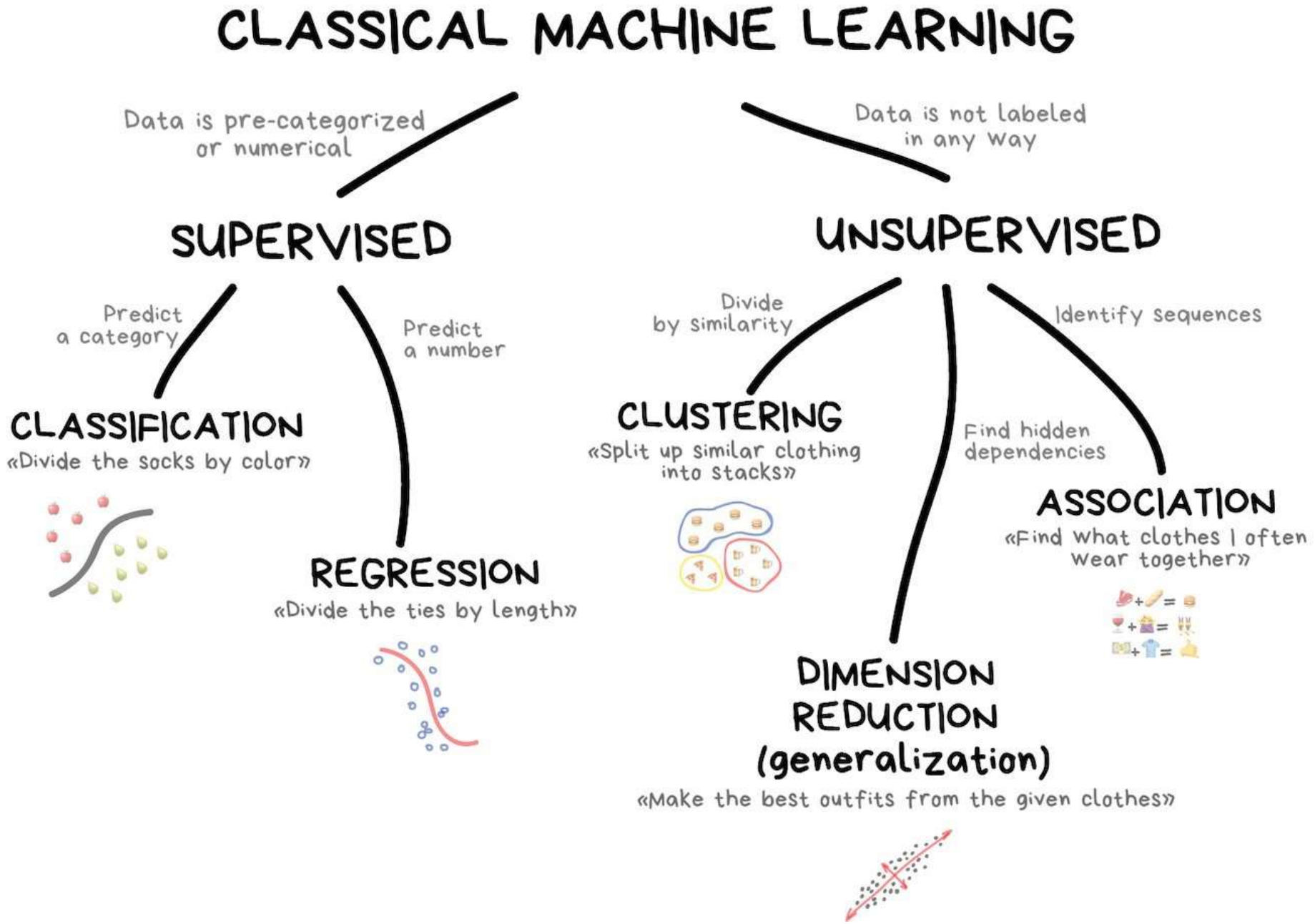
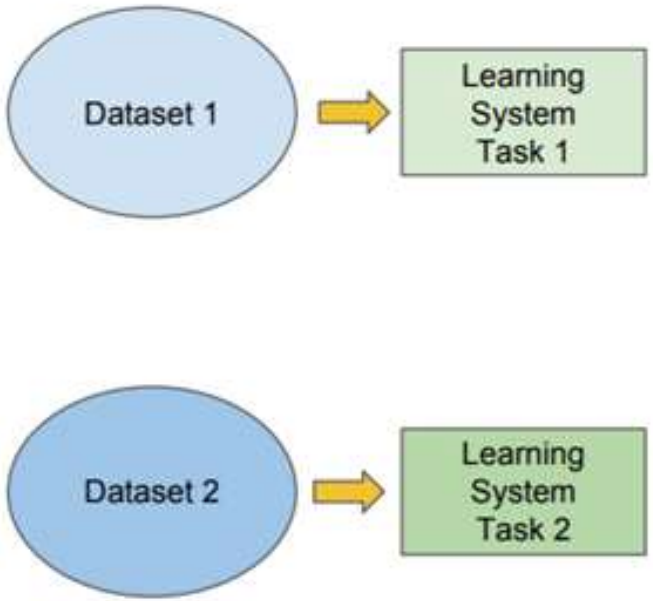


Image from: https://astrowizici.st/teaching/phs5000/10/

Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data

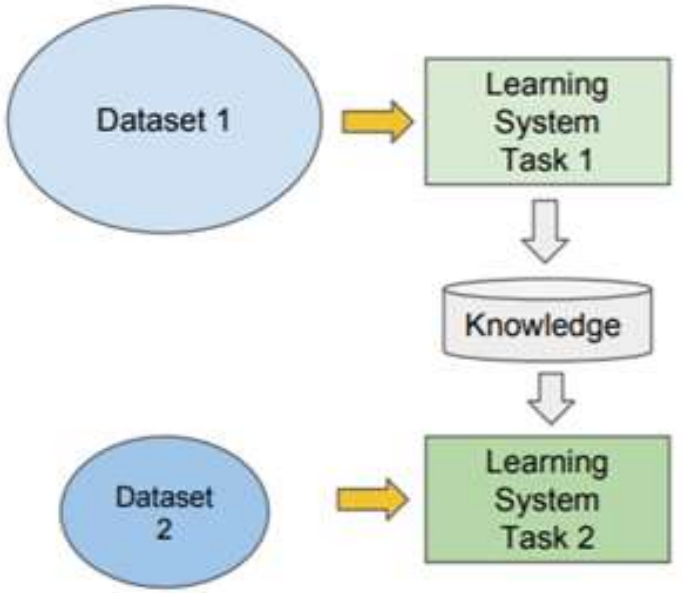


Image from: https://www.researchgate.net/figure/Traditional-Learning-vs-Transfer-Learning-Source-4_fig2_366064278

Transfer Learning

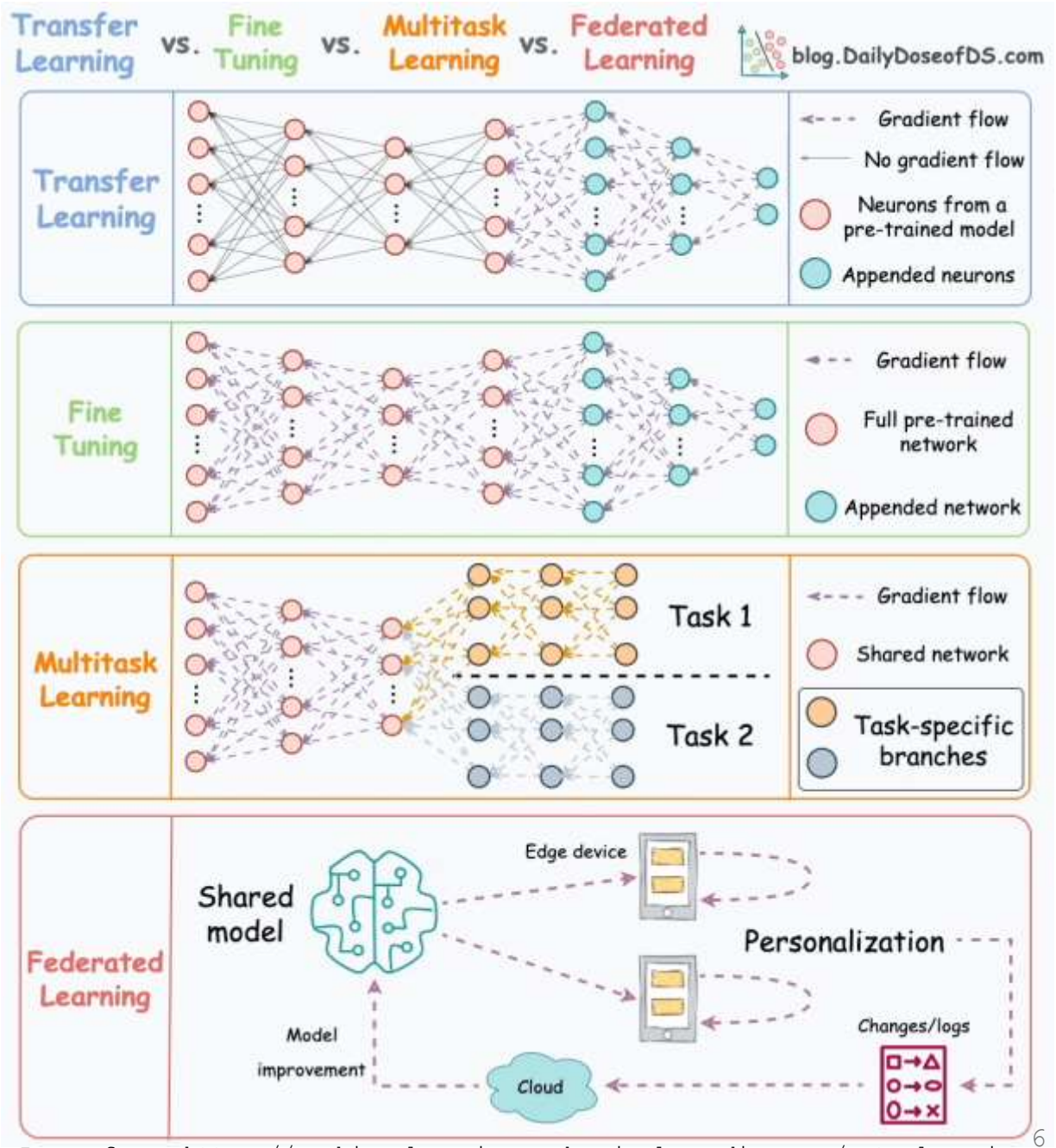


Image from: https://machine-learning-made-simple.medium.com/meta-learning-why-its-a-big-deal-it-s-future-for-foundation-models-and-how-to-improve-it-

Transfer Learning Challenges: Catastrophic Forgetting

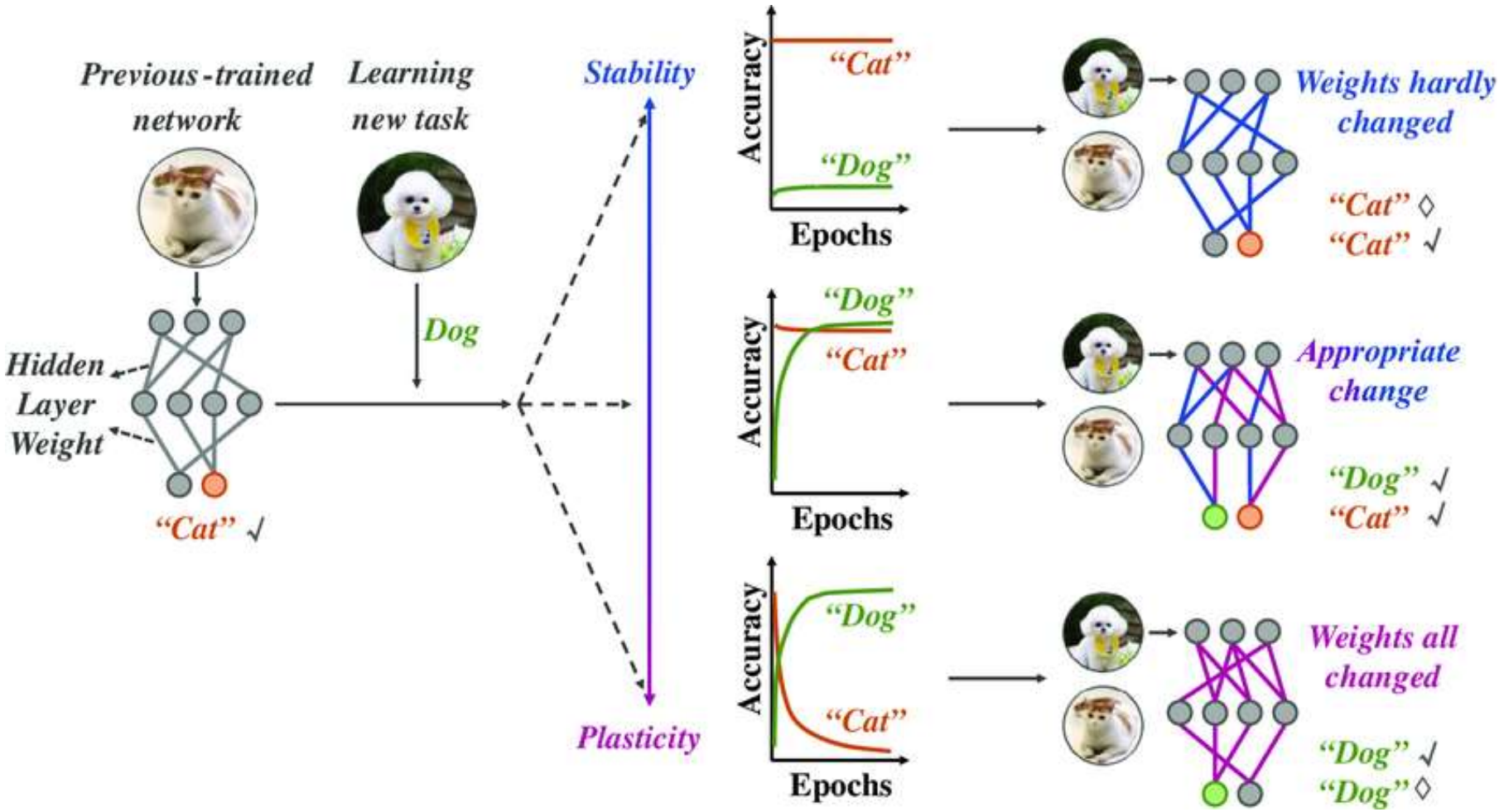
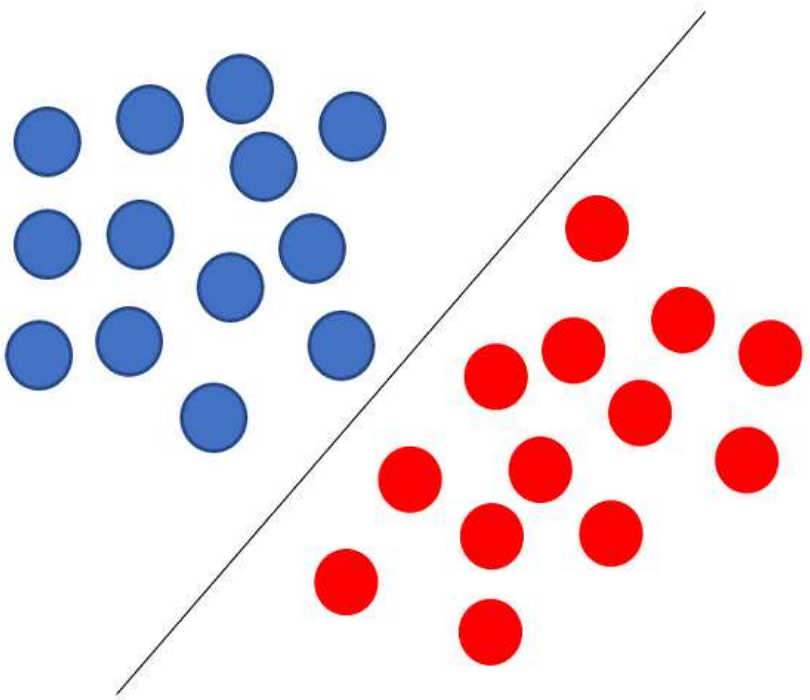
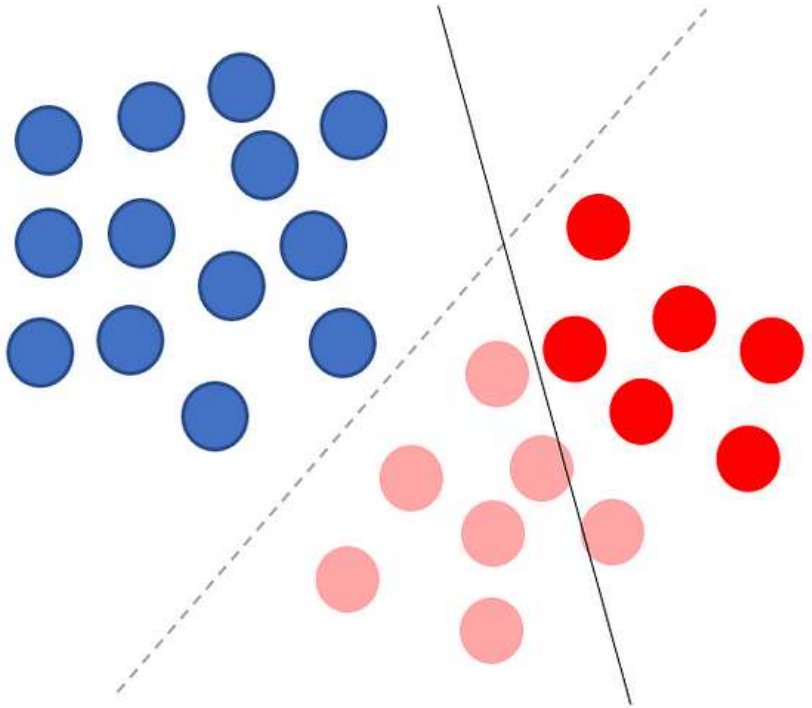


Image from: https://www.researchgate.net/figure/Schematics-of-the-catastrophic-forgetting-When-ANNs-perform-multitask-continual_fig1_360658692

Transfer Learning Challenges: Class Imbalance



Classifier with balanced class



Classifier with imbalanced class

Image from: https://medium.com/x8-the-ai-community/solving-class-imbalance-problem-in-cnn-9c7a5231c478

Putting things into perspective…

- What is this place?
- How many people are there?
- Is it day or night?
- Is it sunny or cloudy?
- Humans:
  - Can decompose
  - Accommodate to the task
  - Can generalize based on few examples (class imbalance)
  - We do not forget that easily (catastrophic forgetting)

To mimic human behaviour..

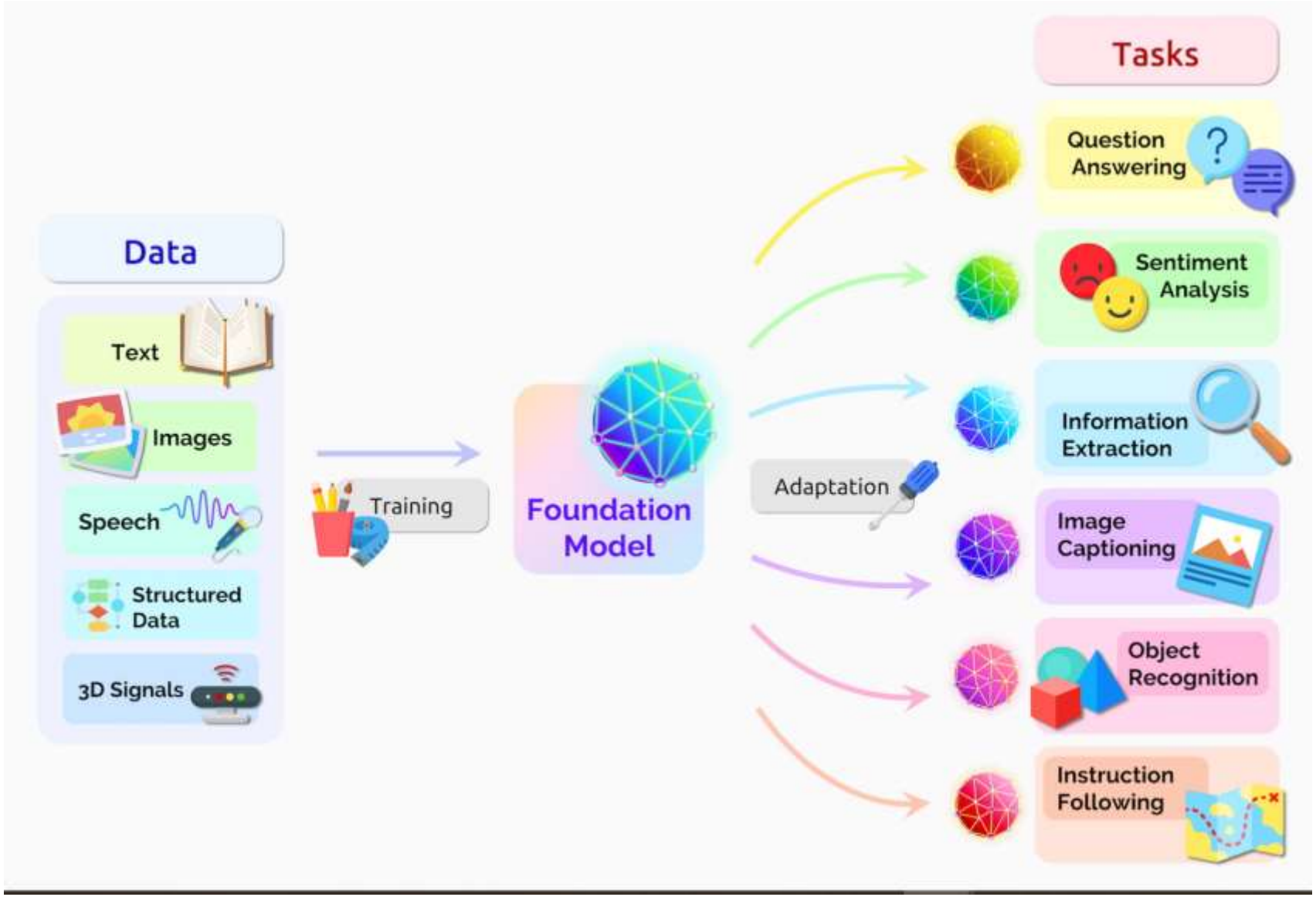


Image from: https://blogs.nvidia.com/blog/what-are-foundation-models/

Meta Learning: Learning to Learn

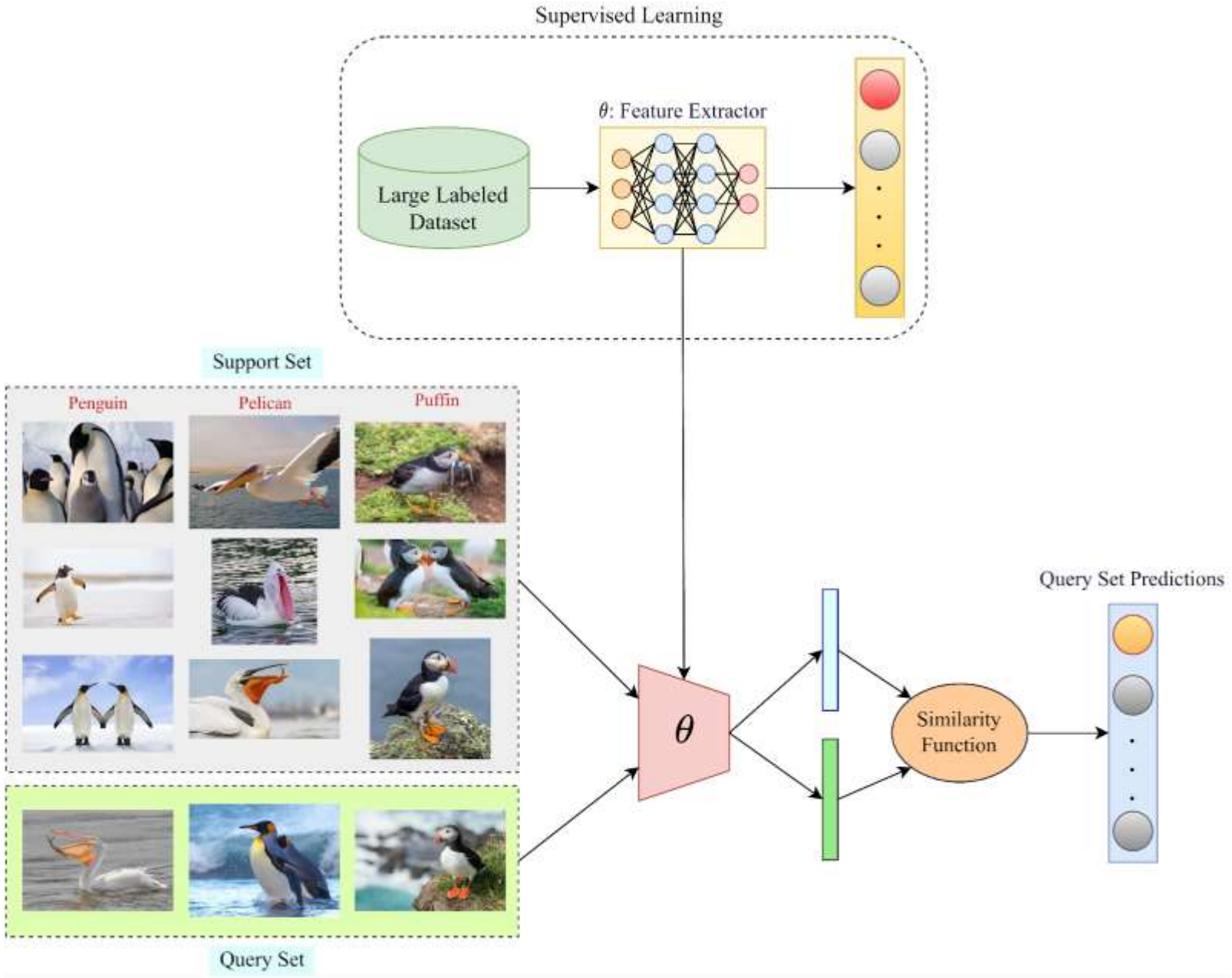
- Develop methods that are able to generalize across tasks, adapting to novel scenarios with little data
- A governing principle of Meta Learning is that the methods are **evaluated** on a previously **UNSEEN task.**
- 2 stages:
  - *Meta training phase*: base model is supplied with a wide array of tasks, learning patterns and how to generalize
  - *Meta testing phase*: the base model's performance is measure on a task that it has **NOT PREVIOUSLY SEEN !**

- Tasks: **Few-Shot,** Domain Adaptation,

To not be confused with Online/Continual Learning!

- While they are similar, they have nuanced differences.
- Main goal of Continual Learning is to **MITIGATE CATASTROPHIC FORGETTING** while transfer learning. (e.g.: use of Adapters)
- The methods are evaluated on tasks/classes that were **SEEN AT TRAINING**.

Classical Few-Shot Scenario



https://www.digitalocean.com/community/tutorials/few-shot-learning

Few-Shot Learning Mathematical Foundation

- $f(\theta)$ – represents the pretrained feature extractor
- D_{base} – **meta training** set with $\{(x_n, y_n)\}_{n=1}^{|Dbase|}$
- $(\overline{x_n})_1 \leq n \leq |Dbase|$ - raw embeddings extracted using feature extractor
- $(\overline{y_n})_1 \leq n \leq |Dbase|$ - corresponding labels with Y_{base} set of classes withing base set
- D_{test} – **meta testing set** with an entirely different set of Y_{test} classes than the meta training set
- $\boldsymbol{y_{base} \cap y_{test} = \emptyset}$

Few-Shot Learning Mathematical Foundation

- Furthermore, we define a **N-way K-shot** task using the following concepts too:
  - We split the test database in the **Query** (inference samples) **set Q** and the **Support set S** (labelled samples)
  - **N-way:** number of different classes in the Support set
  - **K-shot:** number of samples that are RANDOMLY SELECTED inside each class in the Support set
  - $|S| = N \cdot K$

Evaluation of Few-Shot

- Usually, at least 10000 tasks are generated.
- In classical settings, the Query distribution is uniform and contains the same classes as the Support set.
- Common values for K shots are 1 to 5.
- Common values for N ways is 5 to 20.
- The query usually has 15 samples per class.

Types of Few-Shot learning

- **Closed-set:**
  - At inference time, the Query set contains samples only from classes that are also present in support set.
- **Open-set:**
  - At inference time, classes outside the Support set can appear in the Query set. (hardest and most realistic scenario)

Types of Few-Shot learning

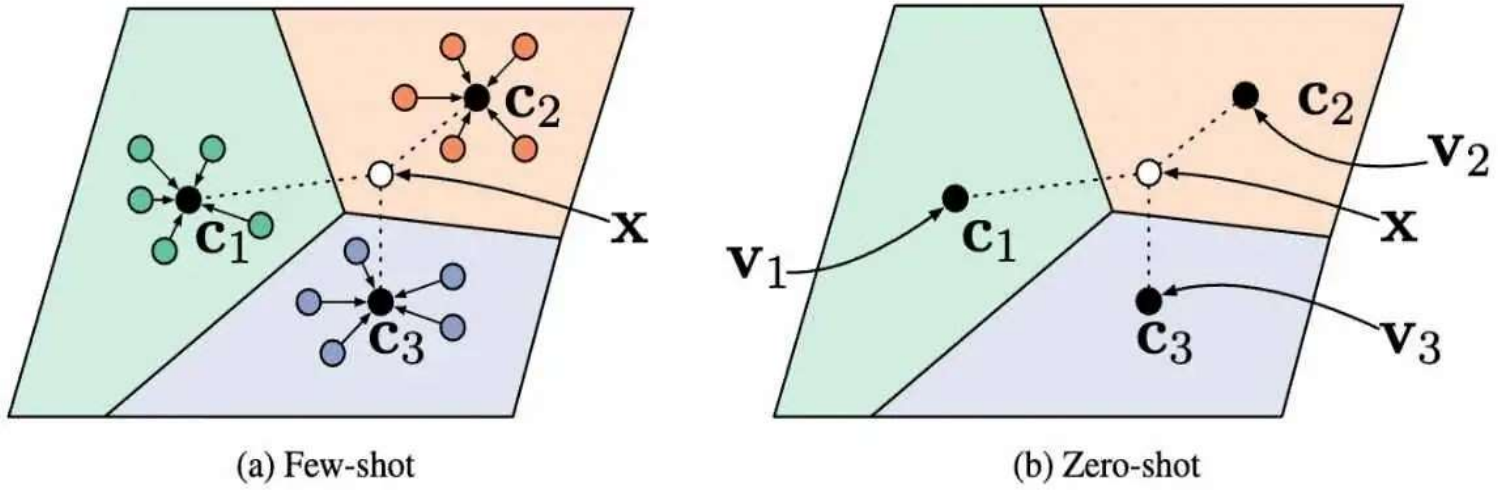
- **Inductive:**
  - Most machine learning methods are inductive
  - Make inference based on a single sample at a time
- **Transductive:**
  - Also utilizes the statistical distribution of all Query samples, improving the performance of the final decision
  - Achieves better results most of the time, compared to inductive methods

Different Few-Shot Methods

- Prototypical Networks
- Simpleshot
- PT-MAP
- LaplacianShot
- TIP-Adapter
- PADDLE

Prototypical Networks

- Learns a **metric space** to the **prototype** representations of each class.
- Imitates the meta testing setting at training => called **episodic training**
- More difficult to implement, as it's not the classical training setup



Class prototypes for Few-shot or Zero-shot [1]

Prototypical Networks

- The prototype for class k from the Support set:

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

- Given a distance metric *d*, prototypical networks produce a distribution over the
for a query point x based on the Softmax over distances to the prototypes in the em

$$p_{\boldsymbol{\phi}}(y = k \mid \mathbf{x}) = \frac{\exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_{k'}))} \quad (2)$$

- Learning happens by minimizing the negative log-probability J(θ) of the true class k v

$$J(\boldsymbol{\phi}) = -\log p_{\boldsymbol{\phi}}(y = k \mid \mathbf{x}) \quad (3)$$

SimpleShot

- Metric learning method [2], based on very simple nearest neighbor classification method using a pretrained model.

$$y(\hat{\boldsymbol{x}}) = \arg\min_{c \in \{1,\cdots,C\}} d(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_c).$$

- The authors also apply to operations that yield substantial improvements:

$$\bar{\boldsymbol{x}} = \frac{1}{|\mathcal{D}_{\text{base}}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{base}}} \boldsymbol{x}$$

$$\hat{\boldsymbol{x}} \leftarrow \frac{\hat{\boldsymbol{x}}}{\|\hat{\boldsymbol{x}}\|_2}.$$

- **Centering**, by substracting the mean feature vector on the meta training set and substracting it from the meta testing features
- **L2 normalization**, each feature vector is normalize a unit *l2* norm

- The breakthrough of this method was not it's mathematical complexity, but the fact that it showed that we such a simple method can surpass the more abstract Prototypical Networks.

Approach	Network	One shot	Five shots
ProtoNet [30]†	ResNet-18	54.16 ± 0.82	73.68 ± 0.65
SimpleShot (CL2N)	ResNet-18	**62.85 ± 0.20**	**80.02 ± 0.14**

PT-MAP

- A method that: 1) preprocesses feature vectors and 2) uses optimal-transport inspired algorithm (transductive setting). [3]
- No training needed! Applied at inference time directly.

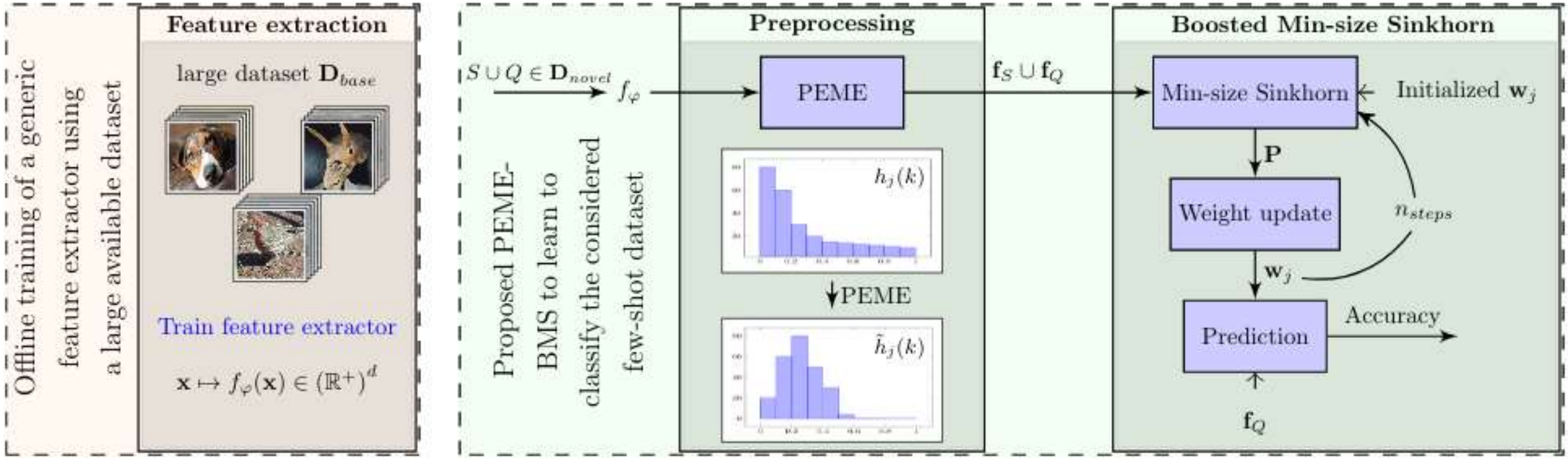


Image from: https://paperswithcode.com/paper/leveraging-the-feature-distribution-in

PT-MAP

1. The feature vectors on the meta testing set are extracted and passed through a **Power Transform** and obtain $\boldsymbol{f_S U f_Q}$.

2. We perform **Sinkhorn mapping** of class centroid c_j initialized on f_S to obtain class allocation matrix **M*** for f_Q that we update iteratively

3. After n steps, we obtain the accuracy on f_Q

$$f(\mathbf{v}) = \begin{cases} \frac{(\mathbf{v}+\epsilon)^\beta}{\|(\mathbf{v}+\epsilon)^\beta\|_2} & \text{if } \beta \neq 0 \\ \frac{\log(\mathbf{v}+\epsilon)}{\|\log(\mathbf{v}+\epsilon)\|_2} & \text{if } \beta = 0 \end{cases}, \quad \textbf{Power Transform}$$

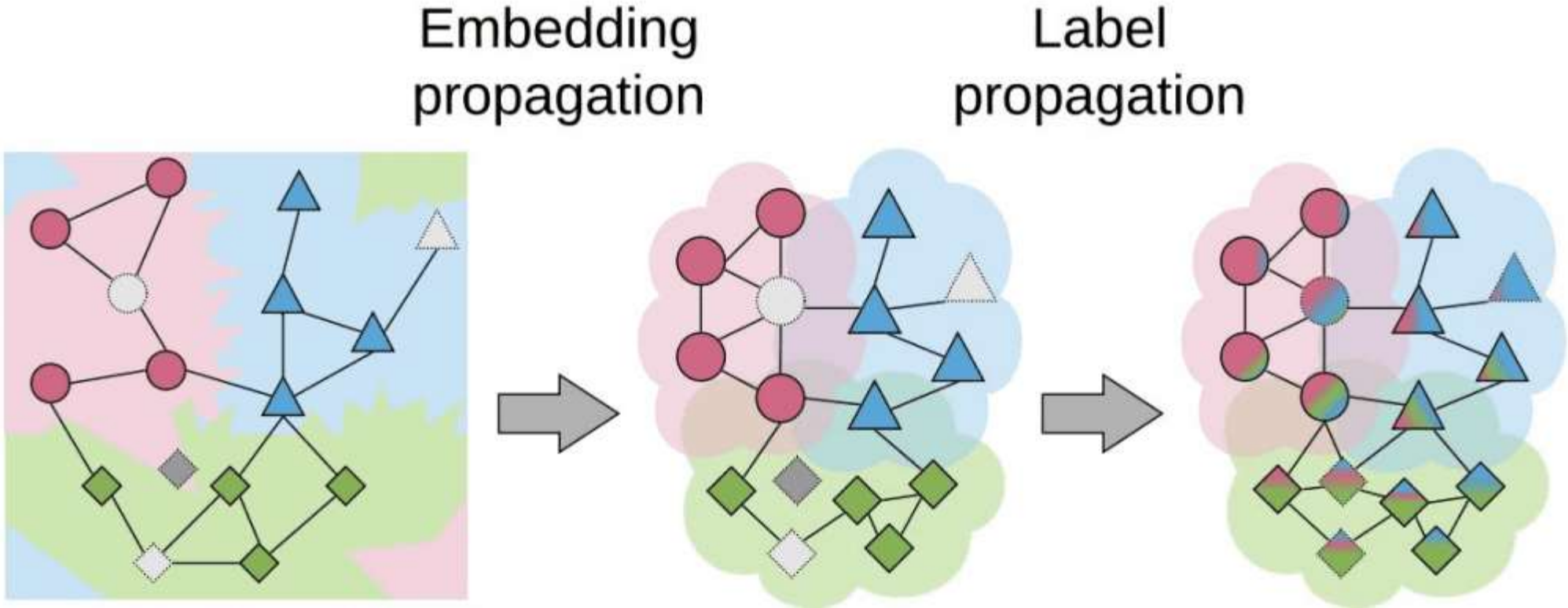
$$\begin{aligned} \mathbf{M}^* &= Sinkhorn(\mathbf{L}, \mathbf{p}, \mathbf{q}, \lambda) \\ &= \arg\min_{\mathbf{M}\in \mathrm{U}(\mathbf{p},\mathbf{q})} \sum_{ij} \mathbf{M}_{ij}\mathbf{L}_{ij} + \lambda H(\mathbf{M}), \end{aligned} \quad \textbf{Sinkhorn Mapping}$$

$$\boldsymbol{\mu}_j = g(\mathbf{M}^*, j) = \frac{\sum_{i=1}^{wq} \mathbf{M^*}_{ij}\mathbf{f}_i + \sum_{\mathbf{f}\in\mathbf{f}_S, \ell(\mathbf{f})=j} \mathbf{f}}{s + \sum_{i=1}^{wq} \mathbf{M^*}_{ij}}. \quad \text{and} \quad \mathbf{c}_j \leftarrow \mathbf{c}_j + \alpha(\boldsymbol{\mu}_j - \mathbf{c}_j). \quad \textbf{Iterative center estimation}$$

Image from: https://paperswithcode.com/paper/leveraging-the-feature-distribution-in

Laplacian Shot

- Transductive Few-Shot method.
- Transductive inference does not re-train the base model, and can be viewed as a graph clustering of the Query set, subject to supervision constraints from the Support set.
- Can use a network pretraining in a classical machine learning setup



Laplacian Shot

- Iterative transductive algorithm that introduces a Laplacian term.[4]
- Does not retrain the base model
- Applied at inference, achieving substantial improvements.

Algorithm 1 Proposed Algorithm for LaplacianShot

Input: $\mathbb{X}_s$, $\mathbb{X}_q$, λ, f_θ
Output: $Labels \in \{1, .., C\}^N$ for $\mathbb{X}_q$
Get prototypes $\mathbf{m}_c$.
Compute $\mathbf{a}_q$ using (8a) $\forall \mathbf{x}_q \in \mathbb{X}_q$.
Initialize $i = 1$.
Initialize $\mathbf{y}_q^i = \frac{\exp(-\mathbf{a}_q)}{\mathbf{1}^t \exp(-\mathbf{a}_q)}$.
repeat
  Compute $\mathbf{y}_q^{i+1}$ using (12)
  $\mathbf{y}_q^i \leftarrow \mathbf{y}_q^{i+1}$.
  $\mathbf{Y} = [\mathbf{y}_q^i]$; $\forall q$.
  $i = i + 1$.
until $\mathcal{B}_i(\mathbf{Y})$ in (7) does not change
$l_q = \arg\max_c \mathbf{y}_q$; $\forall \mathbf{y}_q \in \mathbf{Y}$.
$Labels = \{l_q\}_{q=1}^N$

TIP-Adapter: Few-Shot using CLIP

- CLIP is a vision foundational model.[7]
- CLIP has incredible zero-shot knowledge capabilities
- Leverages the generability of CLIP, as well as the efficiency of the Adapter paradigm. [8]

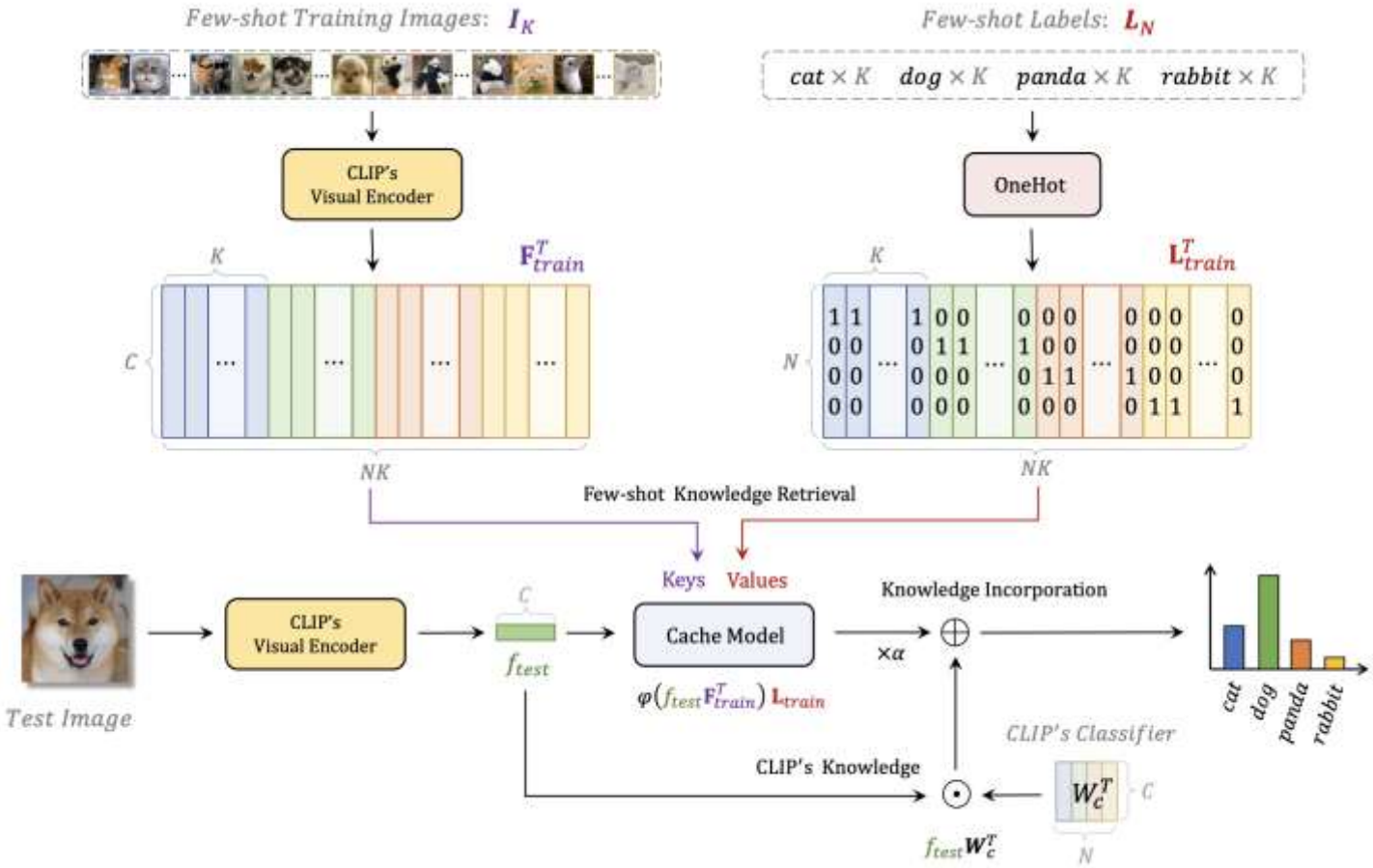


Image from: https://paperswithcode.com/paper/tip-adapter-training-free-adaption-of-clip

PADDLE: Few-Shot on many classes

- PADDLE (**P**rim**A**l Dual **M**inimum **D**escription **Le**ngth)[9] is a Few-Shot method designed for a more realistic scenario where the support set's number of ways is unconstrained.

- In classical few-shot scenarios, classification is performed 5-way or 10-way, which is not the most realistic or practical. As the number of ways increase, the performances drastically decrease.

$$\hat{u}_k = \frac{1}{|\mathbb{Q}|} \sum_{n=1}^{|\mathbb{Q}|} u_{n,k} \quad \forall k \in \{1, \dots, K\},$$

Class proportions for the Query samples

$$\underset{\boldsymbol{U},\boldsymbol{W}}{\text{minimize}} \quad \underbrace{\frac{1}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} u_{n,k} \|\boldsymbol{w}_k - \boldsymbol{z}_n\|^2}_{\text{data-fitting accuracy}} \underbrace{- \lambda \sum_{k=1}^{K} \hat{u}_k \ln(\hat{u}_k)}_{\text{partition complexity}},$$

$$\text{s.t} \quad \boldsymbol{u}_n \in \Delta_K \quad \forall n \in \{1, \dots, |\mathbb{Q}|\},$$
$$\boldsymbol{u}_n = \boldsymbol{y}_n \quad \forall n \in \{|\mathbb{Q}| + 1, \dots, N\}.$$

Objective function

Applications of Few-Shot Learning

- There are many fields in which Few-Shot learning can be applied, across many different modalities.

- As the Machine Learning environment becomes more and more dynamic, we will see a continuous search for methods that leverage foundational models and that can adapt to new tasks easily.

Applications: Large Language Models

- A good example of the few-shot paradigm is the ability of Large Language Models, to leverage their adaptability to new tasks by providing them examples.

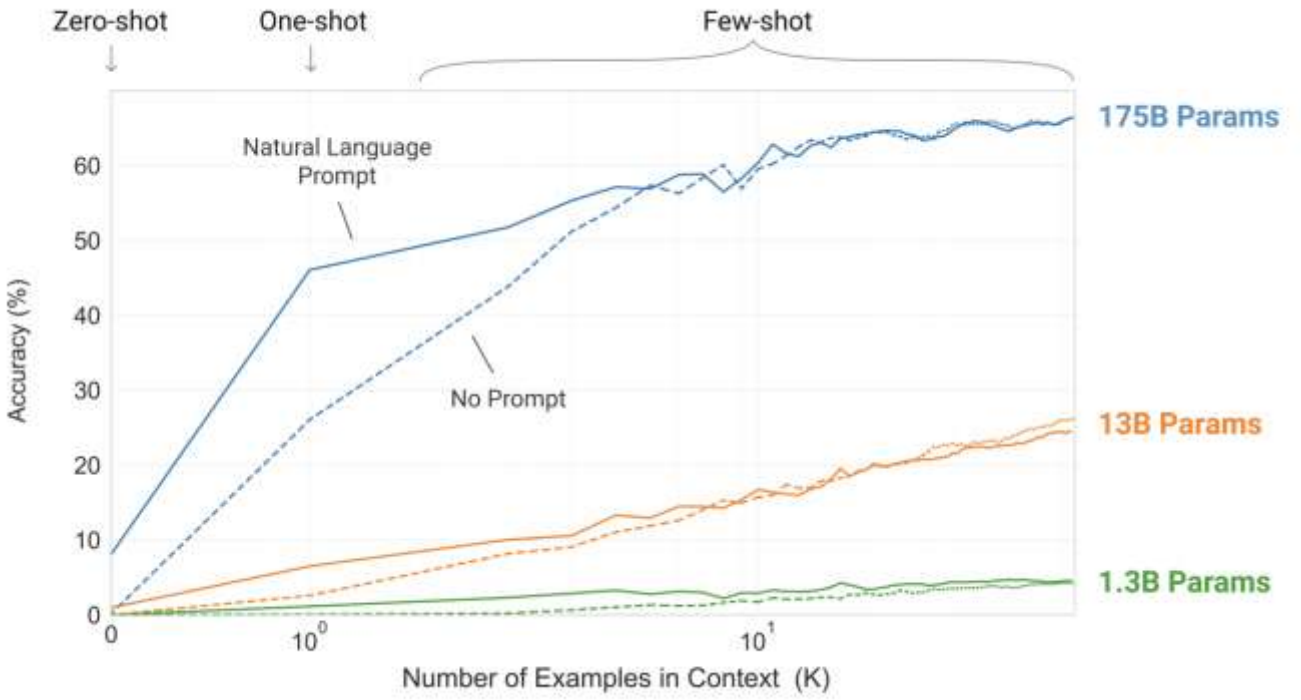


Image from: https://paperswithcode.com/paper/language-models-are-few-shot-learners

Applications: Classification, Object Detection, Segmentation

- It is widely applied across multiple fields in computer vision, ranging from classification to segmentation.

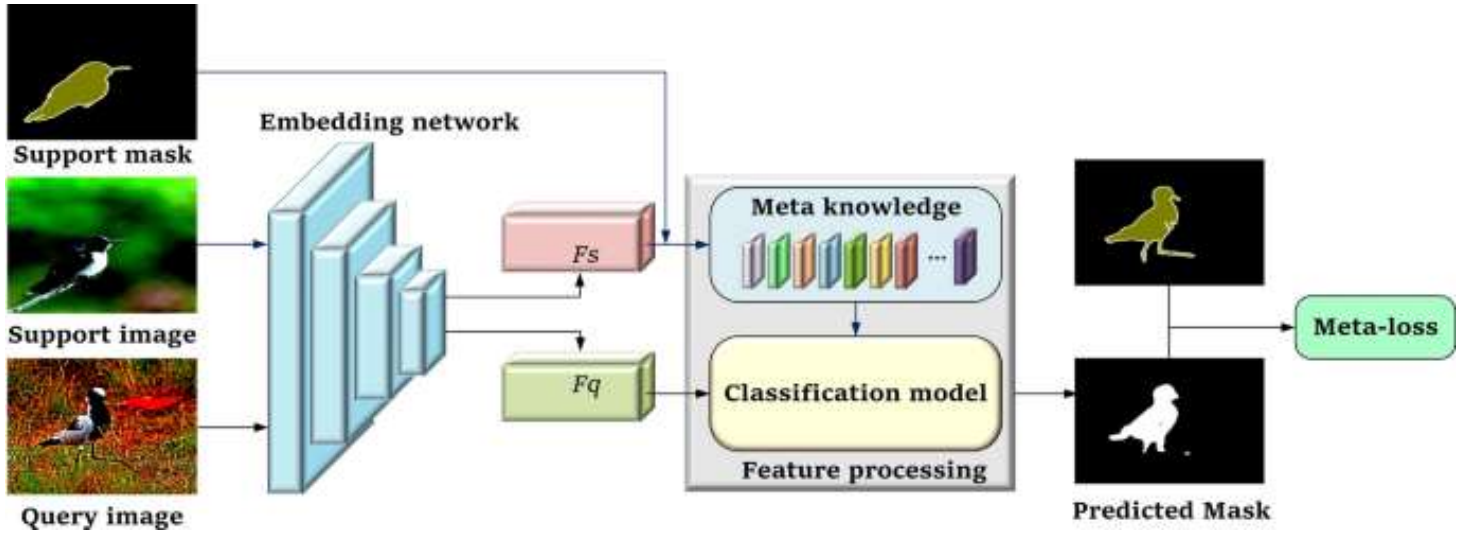


Image from: https://link.springer.com/article/10.1007/s00521-023-08758-9

Applications: Medical Imaging

- Transductive Few-Shot algorithms have been applied for tumour classification in Whole-Slide images with success [10]. It is a good solution for class imbalances that can be found in this type of

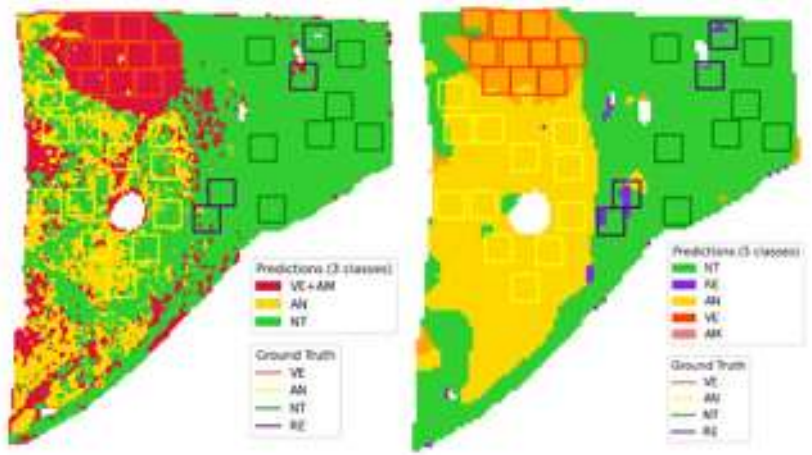


Fig. 2. (left) The 3-class fully supervised model predictions. (right) The few-shot 5-class model predictions.

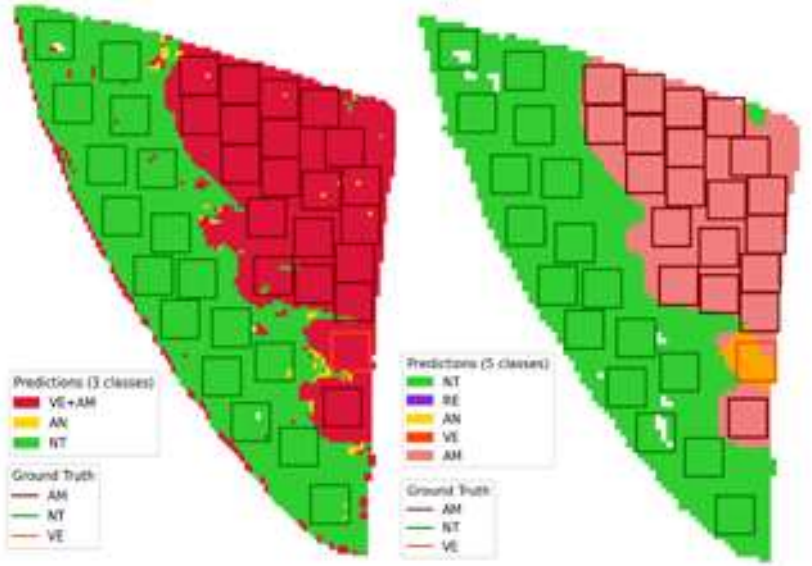


Fig. 3. (left) The 3-class fully supervised model predictions. (right) The few-shot 5-class model predictions.

Image from: "A transductive few-shot learning approach for classification of digital histopathological slider from liver cancer"[10]

Unseen Speaker Identification

- Support Set = Enrollment Set
- Query Set = Test Set
- **Single speaker scenario:**
  - Fraud detection, Forensics, Customer Identification
- **Multi-speaker scenario:**
  - Multi-Speaker Tracking
  - Very common in Audio-Video Tracking

Unseen Speaker Identification: State-of-the-Art

- Open-Set
- Few-Shot
- Real-World Applications

- Limited number of speaker (4-8)
- Usually focuses on the FAR/FRR

Unseen Speaker Identification: State-of-the-Art

- Open-Set Home-Hold applications
  - Limited number of speaker (4-8)
  - Usually focuses on the FAR/FRR

- Open-Set Speaker Identification
  - Focusing on larger watchlists
  - MCE2018 challenge

Unseen Speaker Identification: State-of-the-Art

- Open-Set Household Applications
  - Limited number of speaker (4-8)
  - Usually focuses on the FAR/FRR

- Open-Set Speaker Identification
  - Focusing on larger watchlists
  - MCE2018 challenge

- Closet-Set Speaker Identification
  - Normal Few-Shot Setup
  - 5-way Classification

Unseen Single Speaker Identification: Motivation

- Applications in Forensics, Criminal Investigations, Fraud Prevention or Speaker Retrieval usually contain a **single** speaker's utterances (phone channels, utterances aggregated automatically, etc.)

- Practical scenarios contain more than 5-way classification

- Results in the MCE2018 Challenge showed that Closed-Set identification gave a 40% Top-1 Error increase to the winning team.

Unseen Single Speaker Identification: Motivation

- Applications in Forensics, Criminal Investigations, Fraud Prevention or Speaker Retrieval usually contain a **single** speaker's utterances (phone channels, utterances aggregated automatically, etc.)

- Practical scenarios contain more than 5-way classification. Most benchmarks evaluate on In-Domain scenarios with VoxCeleb1

- Results in the MCE2018 Challenge showed that Closed-Set identification gave a 40% Top-1 Error increase to the winning team.

GOAL:

- Use Transductive methods for Unseen Single Speaker Identification with unconstrained number of speakers in the Support Set

Methods: **F**ew **S**hot for **A** s**i**ngle Class (FSA1C)

- We propose a new method for the case where we have a **single class in the query set** and an **unconstrained number of classes in the support set**
- K classes in Support Set; Keff classes in Query set (Keff << K); Keff = 1.

$$(\forall k \in \{1, \ldots, K\}) \quad u_{n,k} = \begin{cases} 1 & \text{if } n \text{ is in class } k \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

Methods: **F**ew **S**hot for **A** s**i**ngle Class (FSA1C)

- We propose a new method for the case where we have a **single class in the query set** and an **unconstrained number of classes in the support set**
- K classes in Support Set; Keff classes in Query set (Keff << K); Keff = 1.

$$(\forall k \in \{1, \ldots, K\}) \quad u_{n,k} = \begin{cases} 1 & \text{if } n \text{ is in class } k \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

$$\mathbf{x}_n \sim \mathcal{N}(\mathbf{w}_k, \sigma^2 \boldsymbol{I}_d), \quad \text{if } u_{n,k} = 1, \qquad (2)$$

Methods: **F**ew **S**hot for **A** s**i**ngle **C**lass (FSAiC)

- We propose a new method for the case where we have a **single class in the query set** and an **unconstrained number of classes in the support set**
- K classes in Support Set; Keff classes in Query set (Keff << K); Keff = 1.

$$(\forall k \in \{1, \ldots, K\}) \quad u_{n,k} = \begin{cases} 1 & \text{if } n \text{ is in class } k \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

$$\mathbf{x}_n \sim \mathcal{N}(\mathbf{w}_k, \sigma^2 \boldsymbol{I}_d), \quad \text{if } u_{n,k} = 1, \qquad (2)$$

$$\prod_{n \in \mathbb{S}} \Big(\sum_{k=1}^{K} u_{n,k}\, g(\mathbf{x}_n - \mathbf{w}_k \mid \sigma) \Big) \prod_{n \in \mathbb{Q}} g(\mathbf{x}_n - \mathbf{w}_q \mid \sigma),$$

Methods: **F**ew **S**hot for **A** s**i**ngle Class (FSAiC)

- We propose a new method for the case where we have a **single class in the query set** and an **unconstrained number of classes in the support set**
- K classes in Support Set; Keff classes in Query set (Keff << K); Keff = 1;

$$\mathbf{w}_k^{\mathbb{S}} = \frac{\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n}{\|\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n\|} \tag{3}$$

$$\mathbf{w}_k^{\mathbb{S}\cup\mathbb{Q}} = \frac{\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n + \sum_{n\in\mathbb{Q}} \mathbf{x}_n}{\|\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n + \sum_{n\in\mathbb{Q}} \mathbf{x}_n\|}. \tag{4}$$

Methods: **F**ew **S**hot for **A** s**i**ngle Class (FSAiC)

- We propose a new method for the case where we have a **single class in the query set** and an **unconstrained number of classes in the support set**
- K classes in Support Set; Keff classes in Query set (Keff << K); Keff = 1;

$$\mathbf{w}_k^{\mathbb{S}} = \frac{\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n}{\|\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n\|} \quad (3)$$

$$\mathbf{w}_k^{\mathbb{S}\cup\mathbb{Q}} = \frac{\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n + \sum_{n\in\mathbb{Q}} \mathbf{x}_n}{\|\sum_{n\in\mathbb{S}} u_{n,k}\mathbf{x}_n + \sum_{n\in\mathbb{Q}} \mathbf{x}_n\|}. \quad (4)$$

$$C_q = \sum_{n\in\mathbb{S}} u_{n,q}(\|w_q^{\mathbb{S}\cup\mathbb{Q}} - \mathbf{x}_n\|^2 - \|w_q^{\mathbb{S}} - \mathbf{x}_n\|^2) + \sum_{n\in\mathbb{Q}} \|w_q^{\mathbb{S}\cup\mathbb{Q}} - \mathbf{x}_n\|^2, \quad (5)$$

Methods: Simpleshot (SS)

- Nearest neighbour rule (Euclidian distance)
- Inductive baseline (independent prediction for each sample)

$$(\forall n \in \mathbb{Q}) \quad q_n = \underset{k \in \{1,\ldots,K\}}{\arg\min} \|\mathbf{x}_n - \mathbf{w}_k^{\mathbb{S}}\|. \qquad (6)$$

Methods: Simpleshot Majority Vote (SMV)

- Majority vote over the independent query predictions from Simpleshot
- Results in a single prediction for all the samples in the Query Set

Methods: Simpleshot Centroid Distance (SSCD)

- Nearest neighbour rule (Euclidian distance)
- We compute the distance for the support and query centroids
- A single prediction for all samples in the Query Set

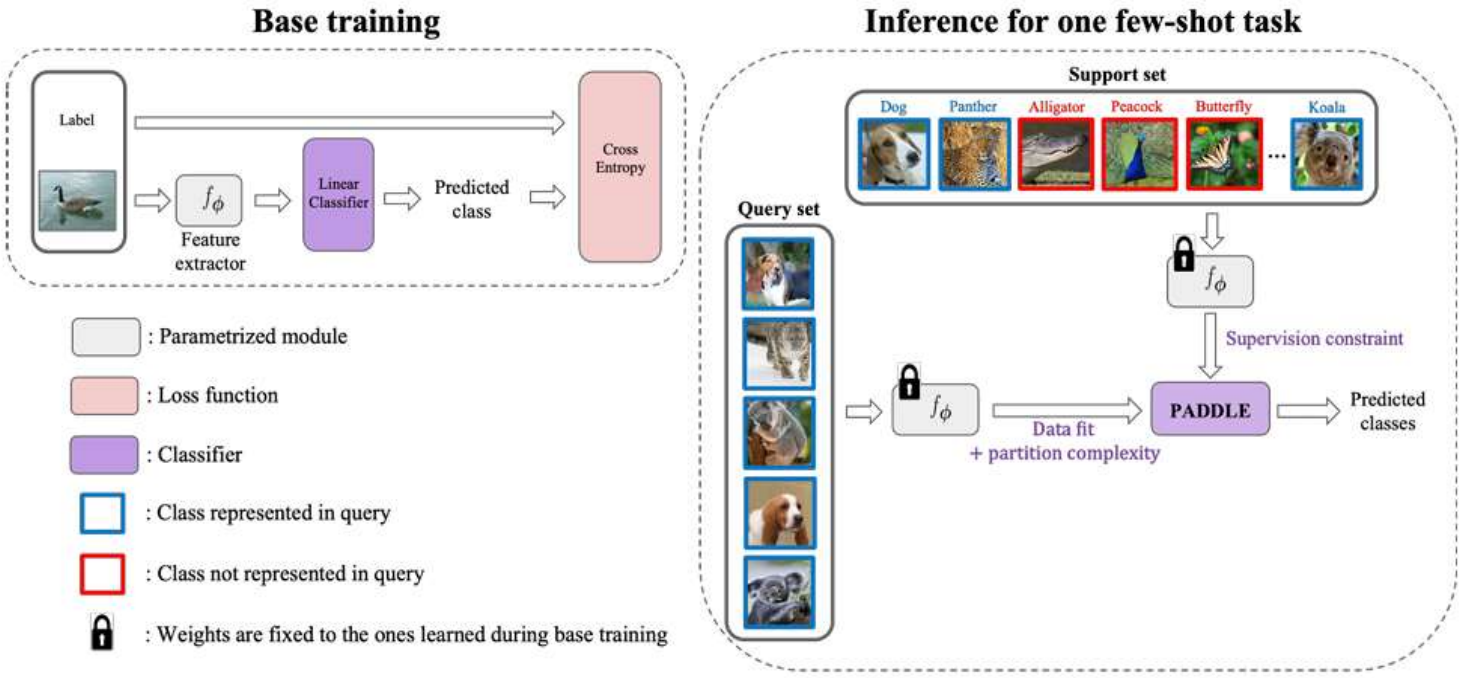
$$q = \underset{k \in \{1,\ldots,K\}}{\arg\min} \|\mathbf{w}^{\mathbb{Q}} - \mathbf{w}_k^{\mathbb{S}}\|, \tag{7}$$

where the normalized query set centroid is given by

$$\mathbf{w}^{\mathbb{Q}} = \frac{\sum_{n \in \mathbb{Q}} \mathbf{x}_n}{\|\sum_{n \in \mathbb{Q}} \mathbf{x}_n\|}. \tag{8}$$

Methods: Paddle with Majority Voting (Variation)

- Transductive Baseline
- Tailored for the case where K >> Keff
- We apply majority voting in order to obtain a single speaker based on the query predictions



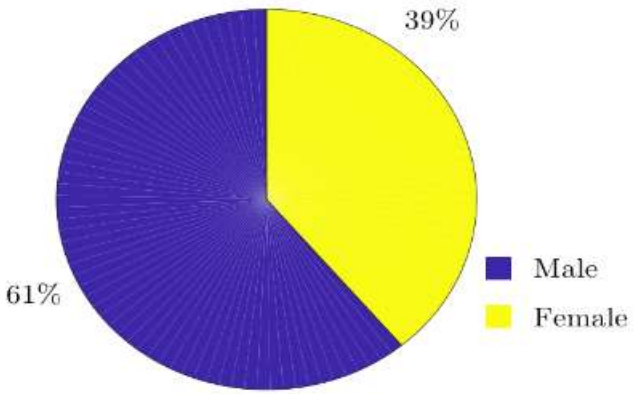
Paddle Few-Shot process. [1]

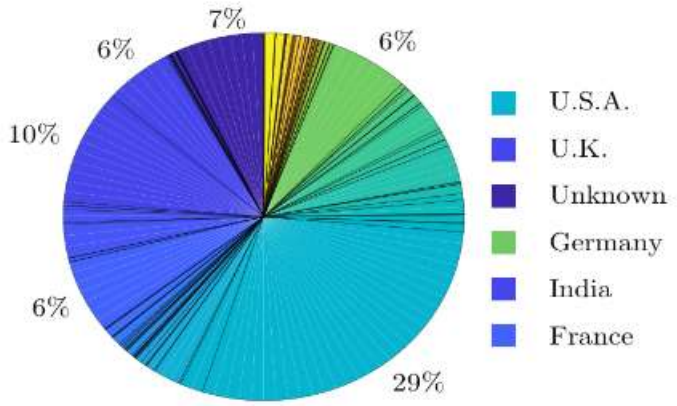
Datasets: VoxCeleb1 and VoxCeleb2

- Domain: Celebrities/Actors during interviews

- **VoxCeleb2:**
  - 6112 identities
  - 1 million utterances from 150k videos
  - Most benchmarks use it as a training set

- **VoxCeleb1:**
  - 1251 identities (1125 for test)
  - 150k utterances from 14k videos
  - Evaluation set



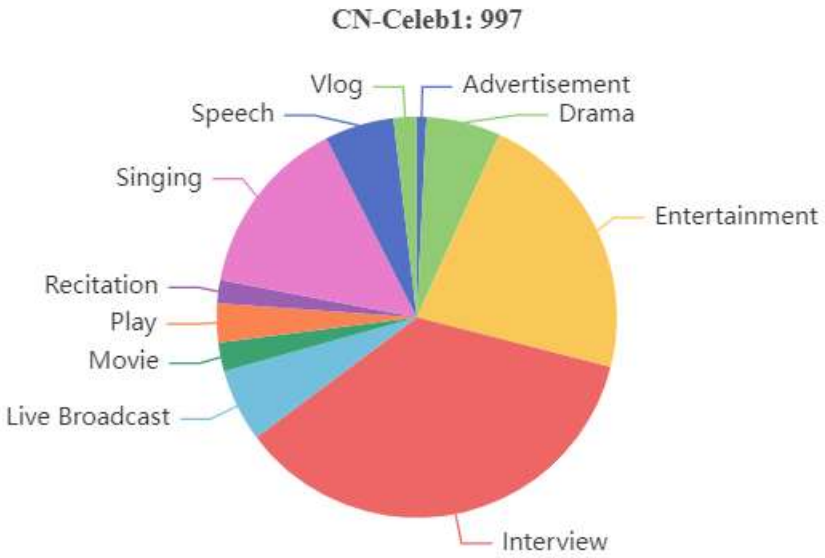


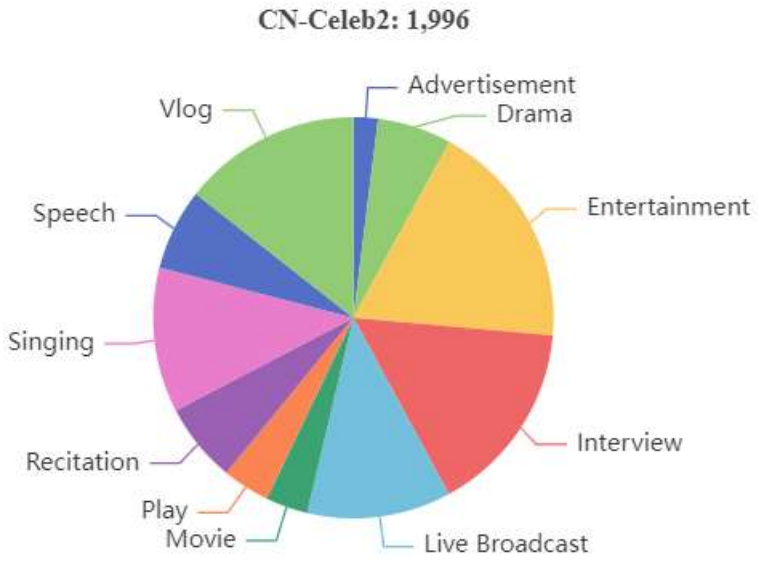
Datasets: CN-Celeb

- Domain: Chinese Celebrities, covering 11 domains

- **CN-Celeb2:**
  - 1996 identities
  - 520k recordings
  - Used for training

- **CN-Celeb1:**
  - 997 identities (856 for test)
  - 125k recordings
  - Used for evaluation





Datasets: JukeBox-v1 Singing Dataset

- Domain: Singers, Singing Recordings

- JukeBox-v1 dataset:
  - 670 singers (505 for test)
  - 385 hours of recordings

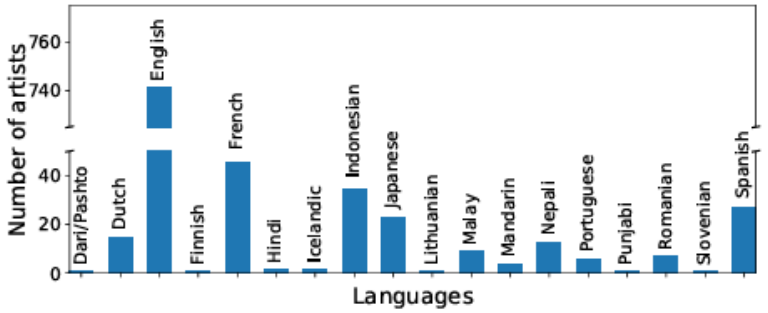


Figure 1: *Distribution of languages in the JukeBox dataset*

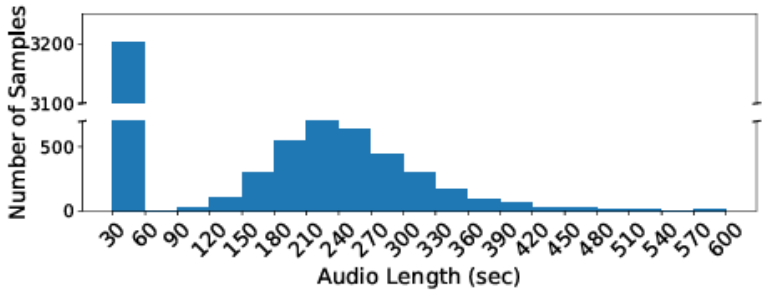
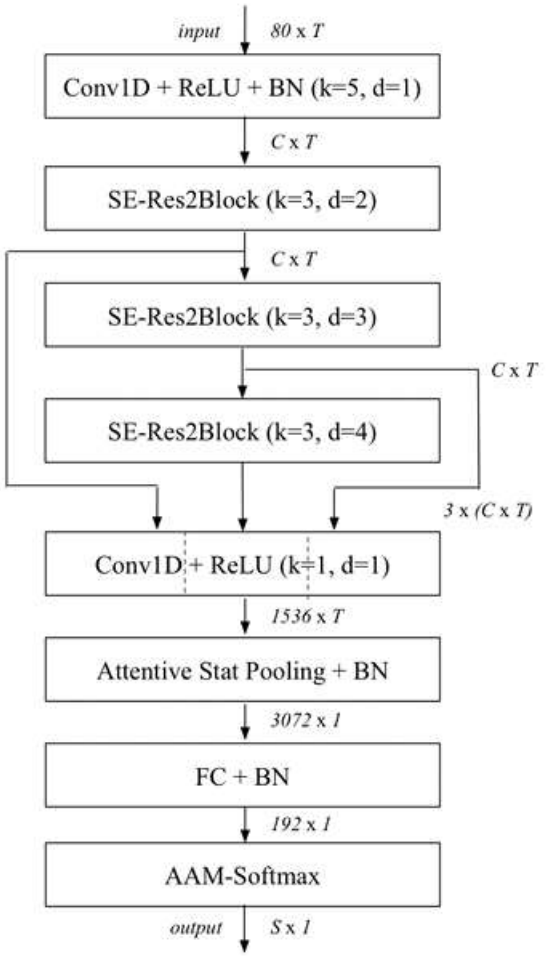


Figure 2: *Distribution of audio length in the JukeBox dataset*

Feature Extractors & Preprocessing

- **Architectures:**
  - Time-Delay Neural Networks: ECAPA-TDNN
- **Preprocessing:**
  - Audios in datasets are cropped to 3s
- **Feature Extraction:**
  - We apply L2 Normalization on extracted features
- **Training:**
  - 1 trained on VoxCeleb2 (5994 speakers, 2.27M recordings)
  - 1 trained on VoxCeleb2 + CNCeleb2 (7990 speaker, 3.35M recordings)



Ecapa-TDNN architecture

Experimental Setup

- We extract features with the trained frozen ECAPA-TDNN backbone
- 1 run 10k tasks, sampling a different Query/Support for each task
- We vary both the |Q| and |S| from 1 to 5.
- Between 505-ways and 1125-ways
- N_eff = 1 (number of classes in the Query Set)

Results: In-Domain

- 1125-ways classification; N_s = 5 shots; N_q = No. Query samples

RESULTS OF EVALUATED METHODS ON THE VOXCELEB1 DATASET.
BOLD = BEST RESULT; UNDERLINE = SECOND BEST RESULT.

N_Q	SS	SMV	SSCD	FSAiC	PADDLE
1	92.38	92.38	92.38	**92.48**	<u>92.42</u>
3	91.7	96.81	<u>99.15</u>	**99.18**	99.06
5	91.2	98.45	<u>99.55</u>	**99.56**	99.41

Results: Out-Of-Domain

- 856-ways for CN-Celeb1 and 505-ways for JukeBox-V1
- Ns = No. shots; Nq = no. Query samples;
- Model trained on English only

		CN-Celeb1					JukeBox-V1				
Train	N_S:N_Q	SS	SMV	SSCD	FSAiC	PADDLE	SS	SMV	SSCD	FSAiC	PADDLE
Vox2	1:1	**23.09**	**23.09**	**23.09**	**23.09**	**23.09**	**29.64**	**29.64**	**29.64**	**29.64**	**29.64**
	1:3	21.36	25.17	35.18	35.18	**35.85**	28.27	32.44	39.38	39.38	**42.24**
	1:5	20.58	31.03	39.75	39.75	**40.43**	27.49	37.50	42.02	42.02	**45.58**
	3:1	36.40	36.40	36.40	**36.44**	35.66	37.43	37.43	37.43	**38.34**	38.25
	3:3	34.84	40.93	53.76	**54.06**	51.10	36.01	41.49	51.58	**52.19**	50.87
	3:5	34.31	49.80	59.77	**60.62**	56.28	35.37	47.20	56.96	**57.54**	55.35
	5:1	42.17	42.17	42.17	**42.41**	41.92	41.10	41.10	41.10	**41.92**	41.78
	5:3	41.21	48.09	60.57	**61.10**	59.42	39.46	45.14	56.90	57.38	**57.49**
	5:5	40.50	57.41	67.12	**67.89**	65.19	38.85	51.33	62.65	**63.21**	62.22
AVG Vox2		32.72	39.34	46.42	**46.63**	45.44	34.85	40.36	46.41	46.85	**47.05**

Results: Out-Of-Domain

- 856-ways for CN-Celeb1 and 505-ways for JukeBox-V1
- Ns = No. shots; Nq = no. Query samples;
- Model trained on English + Chinese

		CN-Celeb1					JukeBox-V1				
Train	N_S:N_Q	SS	SMV	SSCD	FSAiC	PADDLE	SS	SMV	SSCD	FSAiC	PADDLE
Vox2+CN2	1:1	**34.35**	**34.35**	**34.35**	**34.35**	**34.35**	**34.46**	**34.46**	**34.46**	**34.46**	**34.46**
	1:3	31.24	36.64	47.92	47.92	**48.61**	32.30	36.69	46.77	46.77	**47.97**
	1:5	29.73	42.98	51.40	51.40	**52.26**	31.33	42.28	49.26	49.26	**50.57**
	3:1	49.19	49.19	49.19	**49.41**	48.63	46.08	46.08	46.08	**46.99**	46.89
	3:3	46.29	53.89	66.14	**66.33**	63.41	43.53	49.39	60.82	**61.22**	60.02
	3:5	44.72	61.69	71.56	**71.70**	67.31	42.59	56.05	65.80	**66.42**	63.62
	5:1	54.47	54.47	54.47	**54.89**	54.46	51.36	51.36	51.36	**51.87**	51.86
	5:3	51.68	59.45	71.82	**71.95**	69.52	47.86	54.18	65.97	**66.49**	65.76
	5:5	50.40	67.96	77.44	**77.65**	74.46	46.35	60.24	70.88	**71.42**	69.73
AVG Vox2+CN2		43.56	51.18	58.25	**58.40**	57.00	41.76	47.86	54.60	**54.96**	54.54

Conclusions

- Dramatic decrease in performance between In-Domain and Out-Of-Domain (99% -> 55%)
- Training on the Chinese dataset greatly improves for both the Chinese test set and the singing JukeBox dataset
- PADDLE greatly outperforms all the methods in the 1-shot scenario.
- SSCD and FSAiC improve over all other methods in the Multi-Shot scenario. However, FSAiC is slightly more robust to out-of-domain data.
- Speed: SSCD < FSAiC < Paddle (6ms < 34ms < 70ms) (averaged over 10k tasks)

Conclusions

- Transformative shift from the classical methods

- One of the aims of the Machine Learning community is to enable models to generalize across many different tasks, with as few data needed as possible. In some domains, data is simply not available: medical, finance or niche domains.

- Resource efficient and faster to implement than classical training pipelines

Thank you!

References

- [1] Jake Snell, Kevin Swersky, Richard S. Zemel, **Prototypical Networks for Few-shot Learning**
- [2] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, Laurens van der Maaten, **SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning**
- [3] Yuqing Hu, Vincent Gripon, Stéphane Pateux, **Leveraging the Feature Distribution in Transfer-based Few-Shot Learning**
- [4] Imtiaz Masud Ziko, Jose Dolz, Eric Granger, Ismail Ben Ayed, **Laplacian Regularized Few-Shot Learning**
- [5] Renrui Zhang, Zhang Wei, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, Hongsheng Li, **Tip-Adapter: Training-free Adaption of CLIP for Few-shot Classification**
- **[6]** Renrui Zhang, Zhang Wei, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, Hongsheng Li**, Tip-Adapter: Training-free Adaption of CLIP for Few-shot Classification**
- **[7] Radford et al., Learning Transferable Visual Models From Natural Language Supervision**
- **[8] Zhang et al., Tip-Adapter: Training-free Adaption of CLIP for Few-shot Classification**
- **[9] Martin et al., Towards Practical Few-Shot Query Sets: Transductive Minimum Description Length Inference**
- **[10] Sadraoui et al., A transductive few-shot learning approach for classification of digital histopathological slides from liver cancer**