Imperial College London

AI in Medicine I

Advanced Machine Learning: Learning from sparse annotations

Julia Schnabel

I32 – Chair for Computational Imaging and AI in Medicine

Faculty of Informatics

Tutorium: Veronika Zimmer / **Ivan Ezhov** ivan.ezhov@tum.de

Note: Due to *Dies Academicus* on Thursday, there is no tutorial this week, but we provide one for self-study.

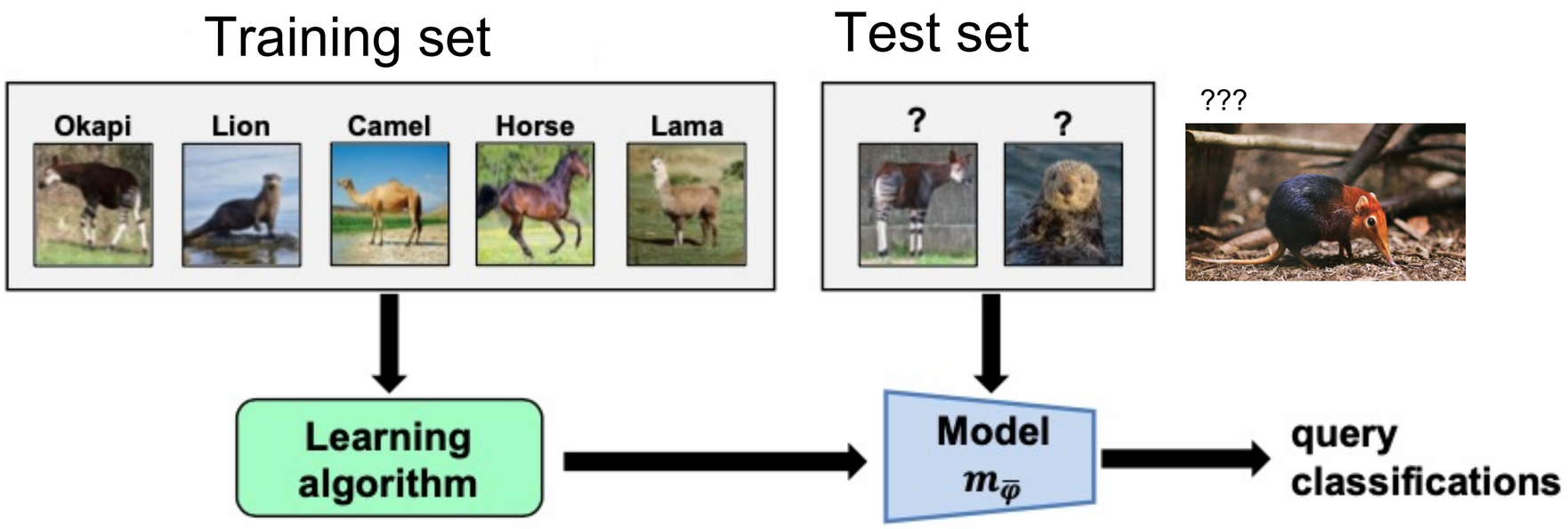
Introduction

- We have previously looked at the issue of **class imbalance**, the need for **data augmentation** and (briefly) **transfer learning**
  - We had **complete training data**, but perhaps **too little**, or **imbalanced**
  - Issues with **overfitting** or **bias**

- This week we will look at the specific issue of learning from sparse annotations. In particular, we will be looking at the following methods:
  - **Transfer learning** (revisited)
  - **Few-shot learning**
  - **Meta-learning**
  - **Incremental learning**
  - **Curriculum learning**

Recall classic set-up

- Train a classification network from scratch
- Overfit to training data -> poor accuracy on test data



Problem of sparse data

- Have you seen this animal before?

It’s an elephant shrew (or going by its Kenyian name, Boni Giant Sengi)

Would you now recognise it next time you happen to see one?

Probably yes – yet you have only seen a single sample!

Problem of sparse data

- It’s one of the rarest animals in the world (only ~13,000 still exist) – so you might come across just a few more samples in your life:



Humans can learn new concepts using only few training samples.

Can we train a machine learning algorithm to also learn from only few samples?

Problem of sparse data in medicine

- We can think of the same problem in the case of **rare medical diseases or conditions**
  - Less interest by drug companies, fewer clinical trials.
  - Also called **“orphan diseases”**

- **What is a rare disease**?
  - In the USA, a rare disease is a condition affecting **fewer than 200,000** (in 330M, 0.06%).
  - In the European Union, they are diseases diagnosed in **1 in 2,000 people** (0.05%)

- Example of perhaps the rarest disease in the world:
  - According to Journal of Molecular Medicine, **Ribose-5 phosphate isomerase (RPI) deficiency,** is the rarest disease in the world with MRI & DNA analysis providing ***only one case in history***

Problem of sparse data in medicine

- **Some rare diseases are less rare** than you might think:
  - **Multiple sclerosis**: prevalence of 90 in 100k *(just above threshold of rare disease)*
  - **Narcolepsy** (sleep disorder): 50 in 100k
  - **Primary biliary cholangitis** (auto-immune disease in the liver): 40 in 100k
  - **Fabry disease** (genetic disorder where fatty acids cannot be broken down): 30 in 100k
  - **Cystic fibrosis** (hereditary disorder): 25 in 100k

- **Some rare diseases are on the rise** due to our changing lifestyles or environmental factors:
  - **Diabetes Type II** in children due to childhood obesity: 18 in 100k (US and Germany)
  - **Breast implant-associated anaplastic large cell lymphoma**: 0.35-1 case per 1M

Problem of sparse data in medicine

- Generally there is simply a **lack of availability of large annotated medical (image) databases**, even for common diseases/conditions.

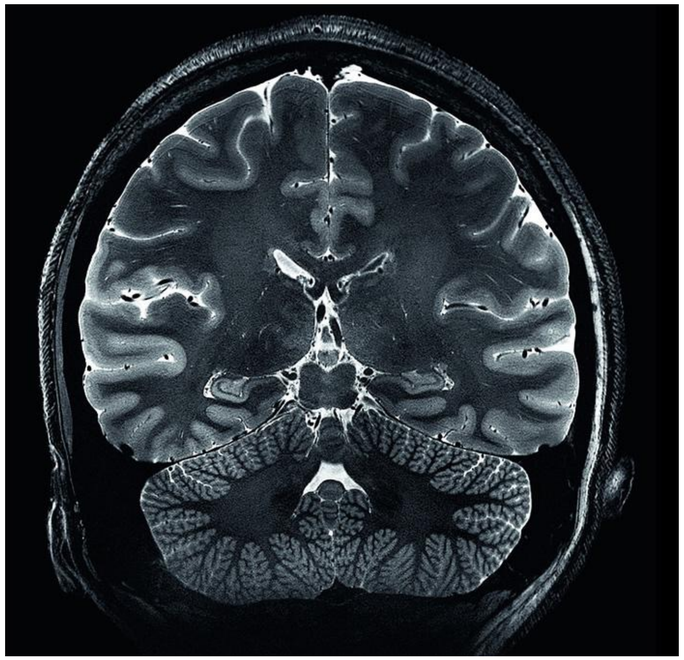
- This is due to **legal frameworks** and **logistics**:
  - Ethics, patient consent, data protection and privacy concerns
  - Digital infrastructure and data sharing

- But also due to the **lack of quantity or quality of annotations**:
  - Intense human labour needed to annotate and clean raw data
  - Time consuming and expensive
  - Error prone, intra-/inter-observer variability

Problem of sparse data in medical imaging

- Another problem is that **new medical imaging technology** is continuously developed and advancing, for which then only few instances exist in the beginning:

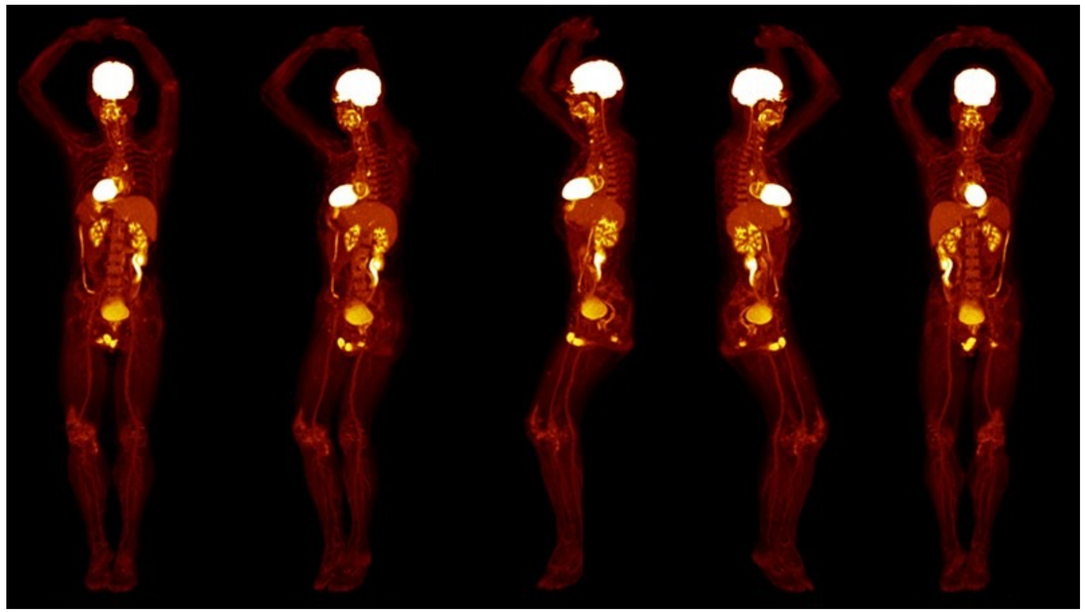
X-ray of Anna Röntgen's hand, Würzburg 1895



7T MR brain scan, Siemens Healthineers 2019 Deutscher Zukunftspreis

Problem of sparse data in medical imaging

- Also elsewhere in this world: total-body PET imaging



UC Davies & United Imaging, China

https://www.youtube.com/watch?v=YmC_-RAVCjg

Ramsey Badawi
23 subscribers

SUBSCRIBE

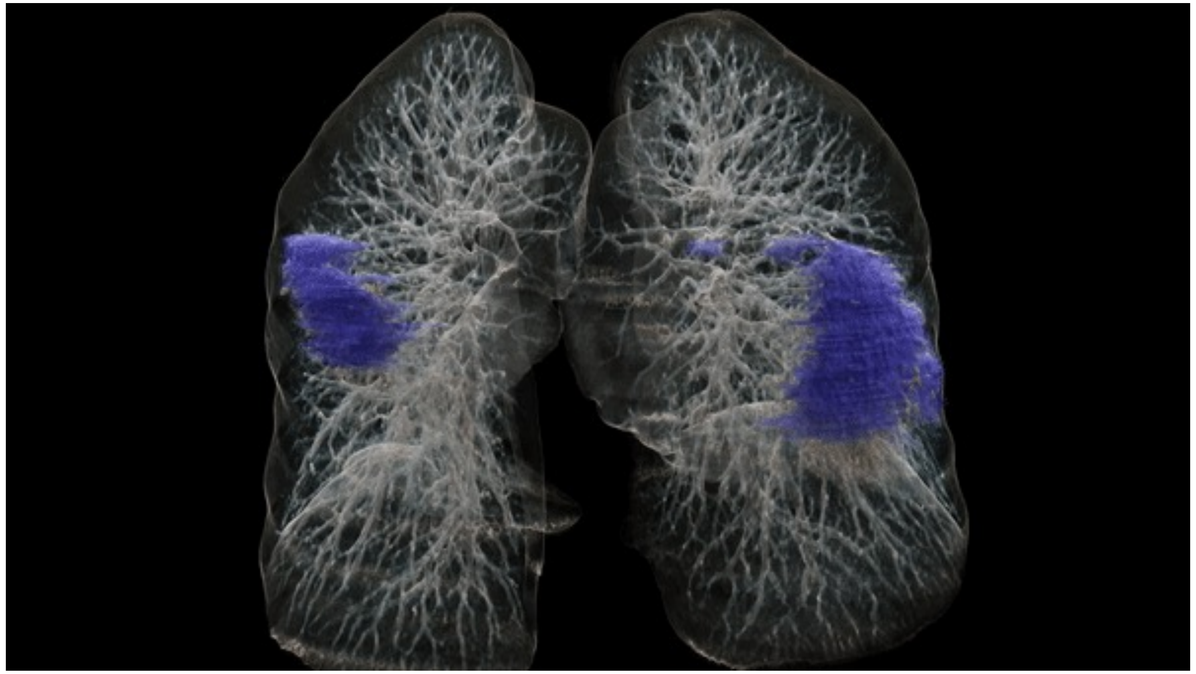
This is a slow-motion EXPLORER Total Body PET video of a PET radiotracer injected into the leg. The frame-rate is 10 frames per second. You can see the radiotracer go up the leg vein, enter the right ventricle, go to the lungs through the pulmonary artery, and then return to the left ventricle through the pulmonary veins. After that, you can see the left ventricle pump the blood into the aorta and the rest of the body. At around 30 seconds, you can see a zoom-in of the blood entering the heart and lungs. This is the first time individual heart-beats have been seen using PET.

This work was done by Dr. Xuezhu Zhang at UC Davis.

Nature **570**, 285-286 (2019)
doi: https://doi.org/10.1038/d41586-019-01833-z

Problem of changing data

- In a real world / clincial scenario, we need to **continuously adapt to new data**:



Example: COVID-19 Siemens Healthineers
https://www.siemens-healthineers.com/perspectives/mso-ai-prototyp-covid-19

Recall: Supervised learning

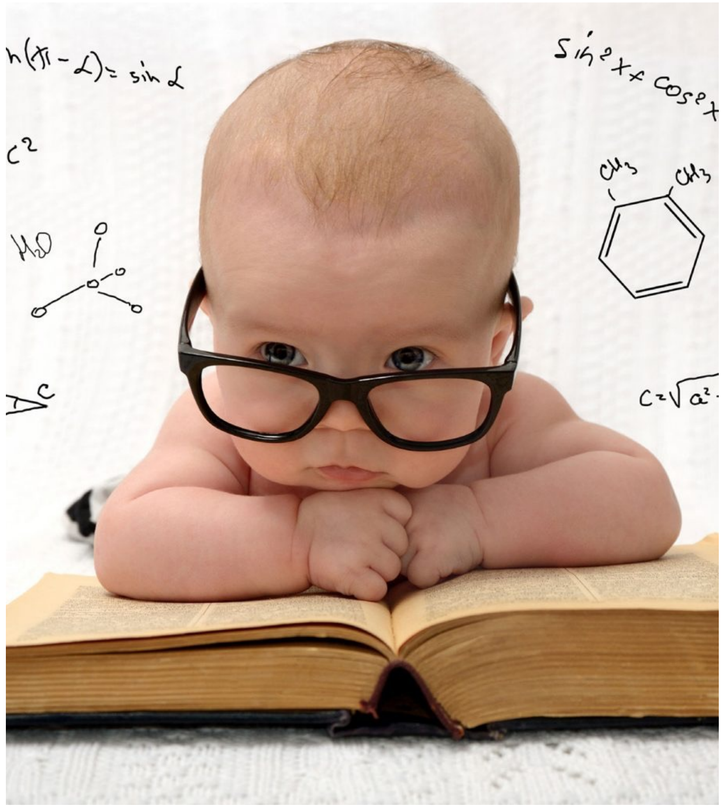
- Supervised learning methods tend to work well on fixed datasets
  - Training takes place with all data and concepts at once
- After training, it is hard to adapt to new data:
  - Need to retrain with all data (original and new), but if trained models are shared, there may not be access to original data anymore
- Alternatively, one can use transfer learning and fine-tune the network on new data or task
  - Risk of "**catastrophic forgetting**" of prior knowledge

Contrast this with Human learning

- (Most) humans can **learn with limited guidance and without constant supervision**

- We can **learn throughout our lives** while **retaining (most) knowledge** of our past experience

- *Can we **close this gap** between human learning and machine learning?*

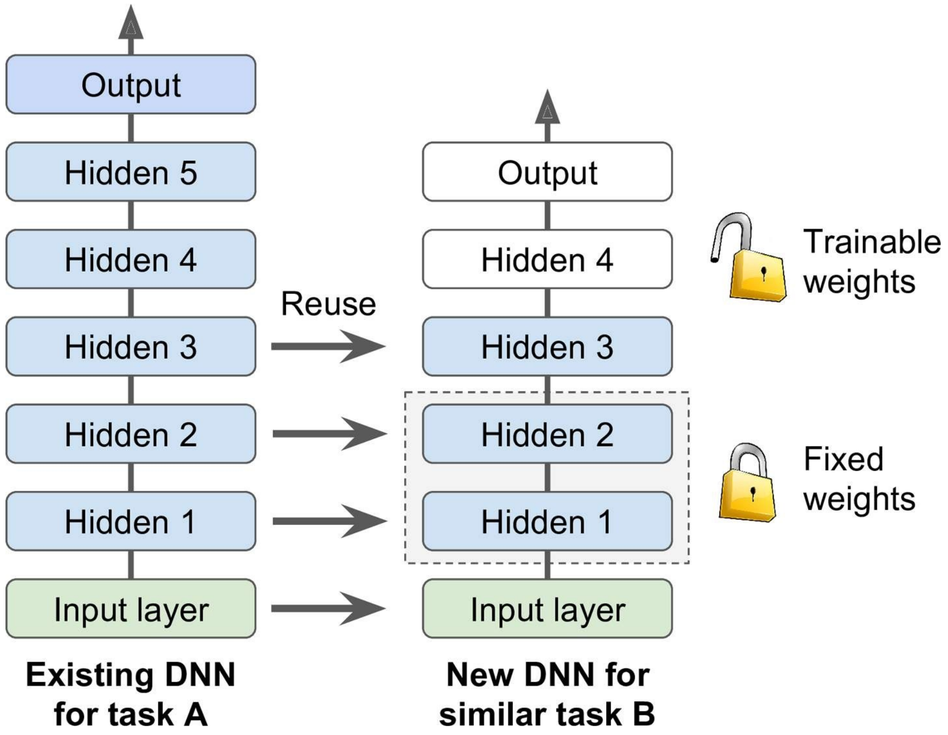


Overview

- **Transfer learning** (revisited)
- Few-shot learning
- Meta-learning
- Incremental learning
- Curriculum learning

Transfer learning (revisited)

- When moving from one (similar) imaging domain or clinical application to another, you may not wish to train your deep neural network architecture from scratch, but just recycle it:



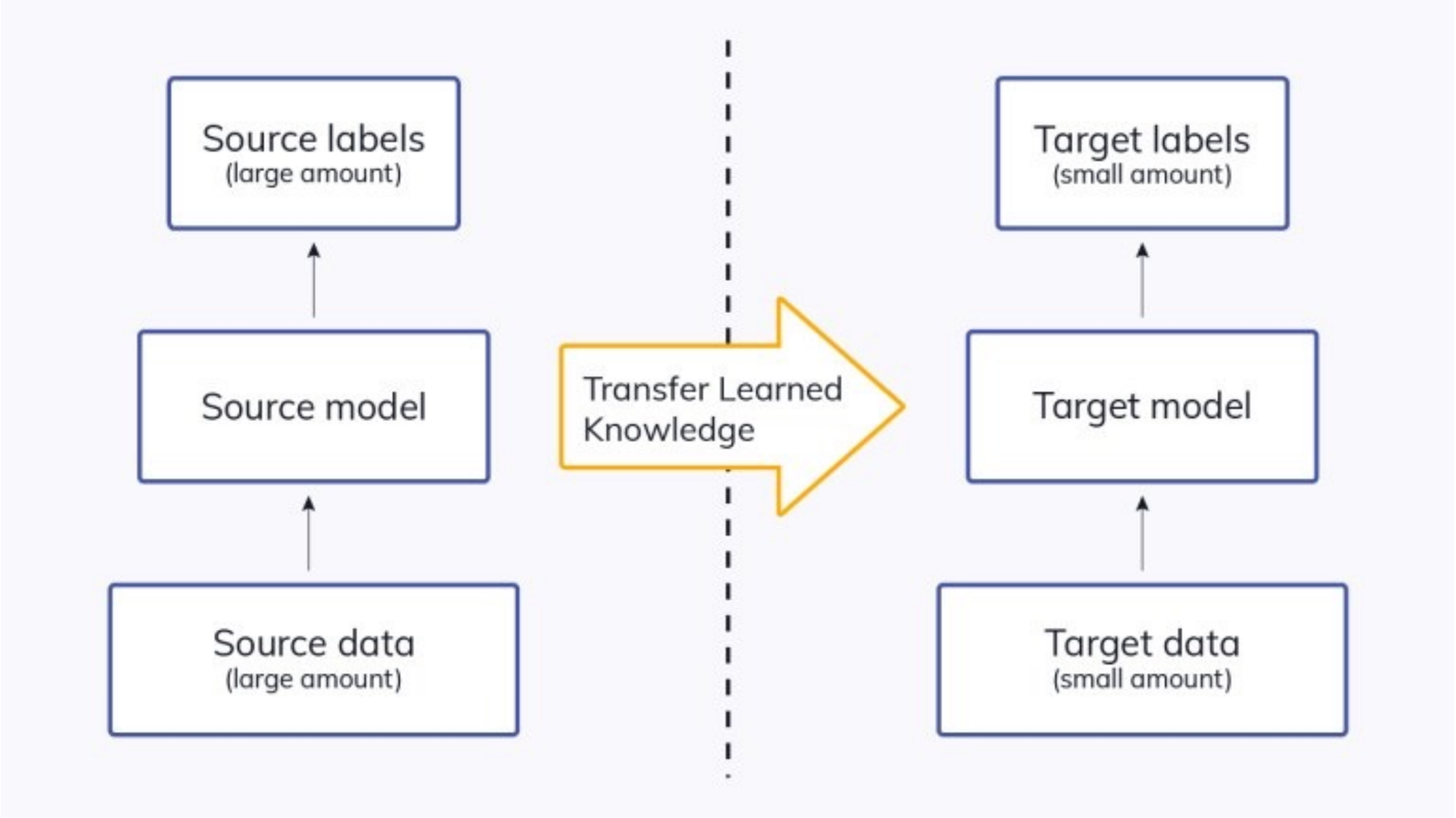
Src: Hands-on Machine Learning with Scikit-Learn & Tensorflow

You can re-use the pre-trained lower layers and add new layers for training on the new task

This will speed up training and will require fewer new training data.

Transfer learning (revisited)

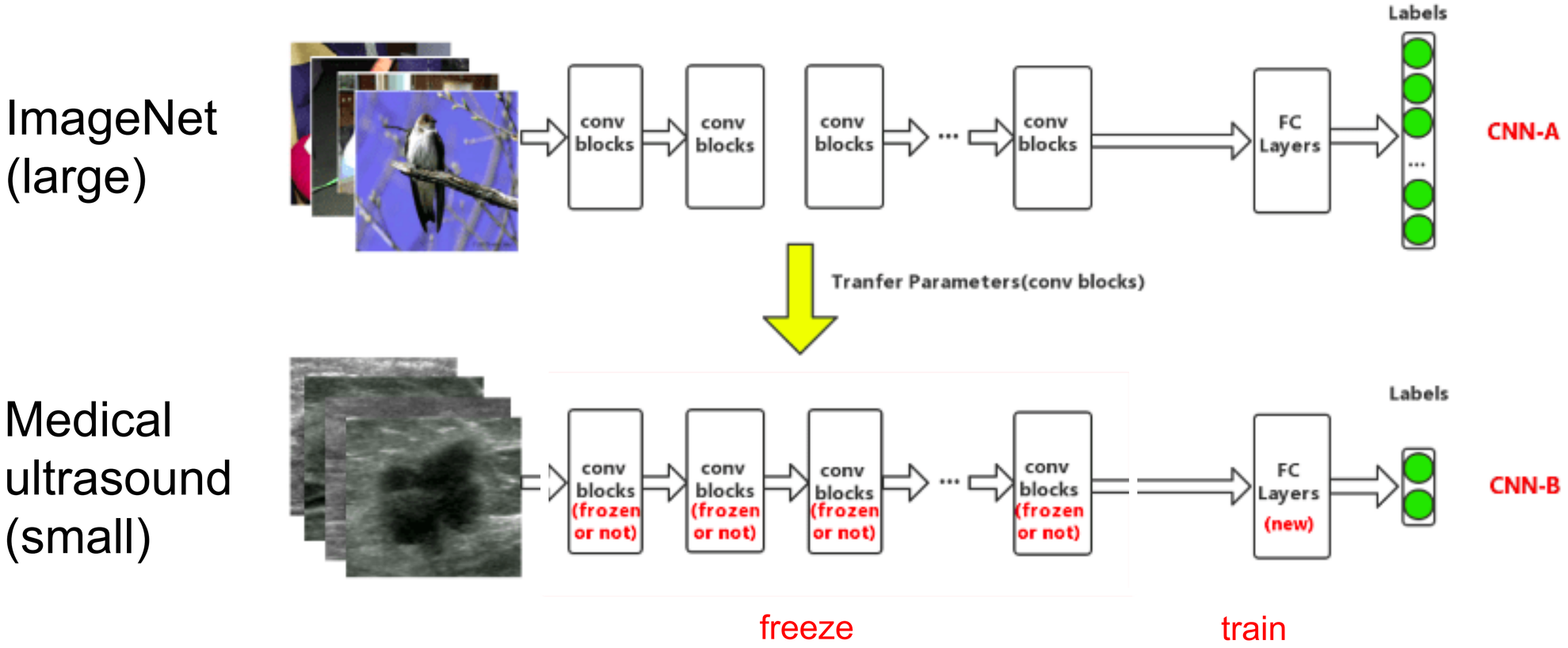
- Having a closer look with some definitions:



https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras

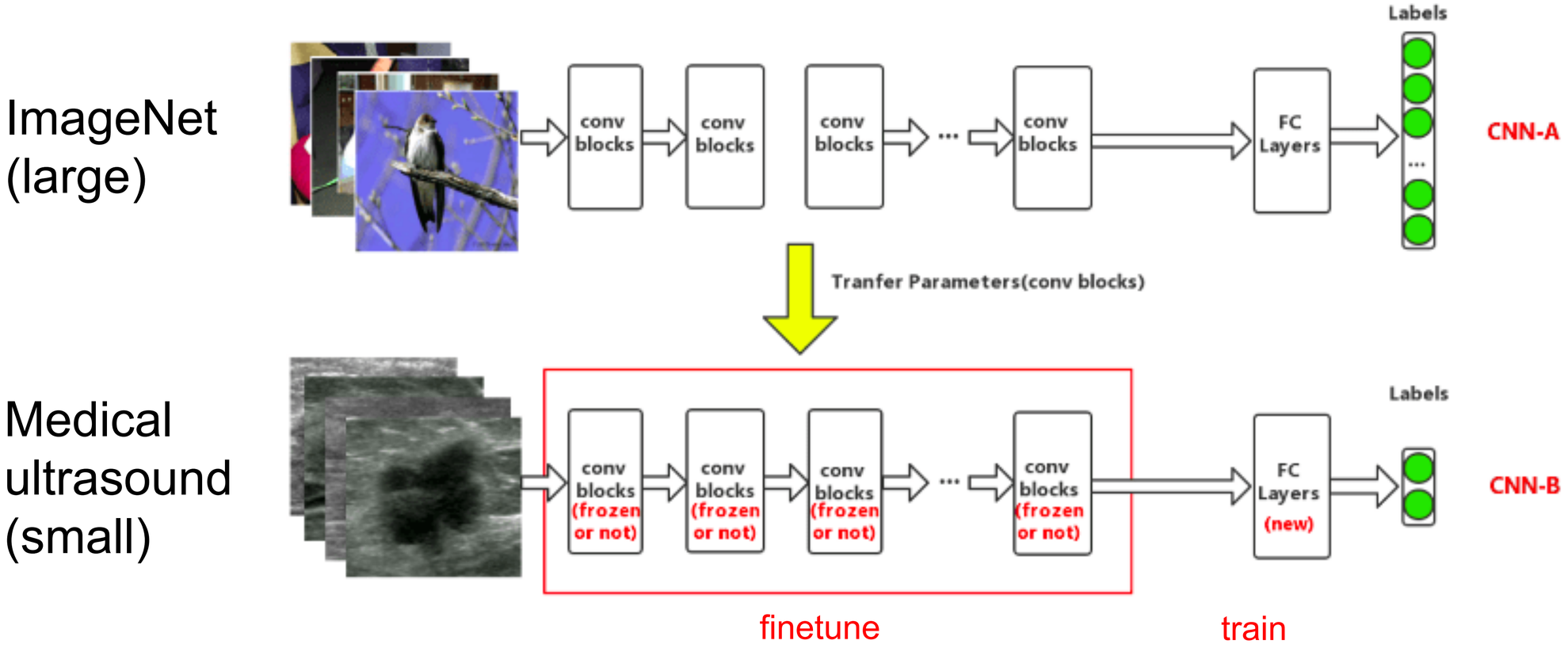
Transfer learning (revisited)

- Example: Train on ImageNet (13M images and >1000 classes), transfer to small medical database:



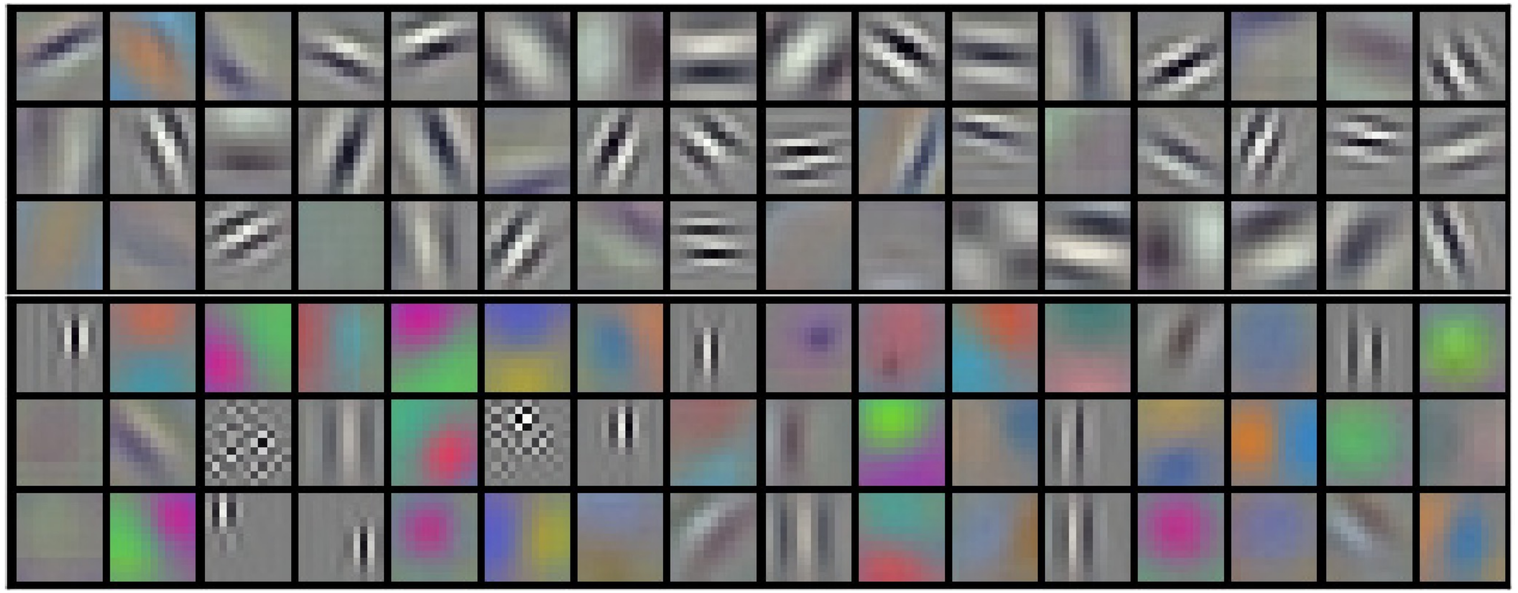
Transfer learning (revisited)

- Example: Train on ImageNet (13M images and >1000 classes), transfer to small medical database:



Transfer learning (revisited)

- What has the first network learnt?

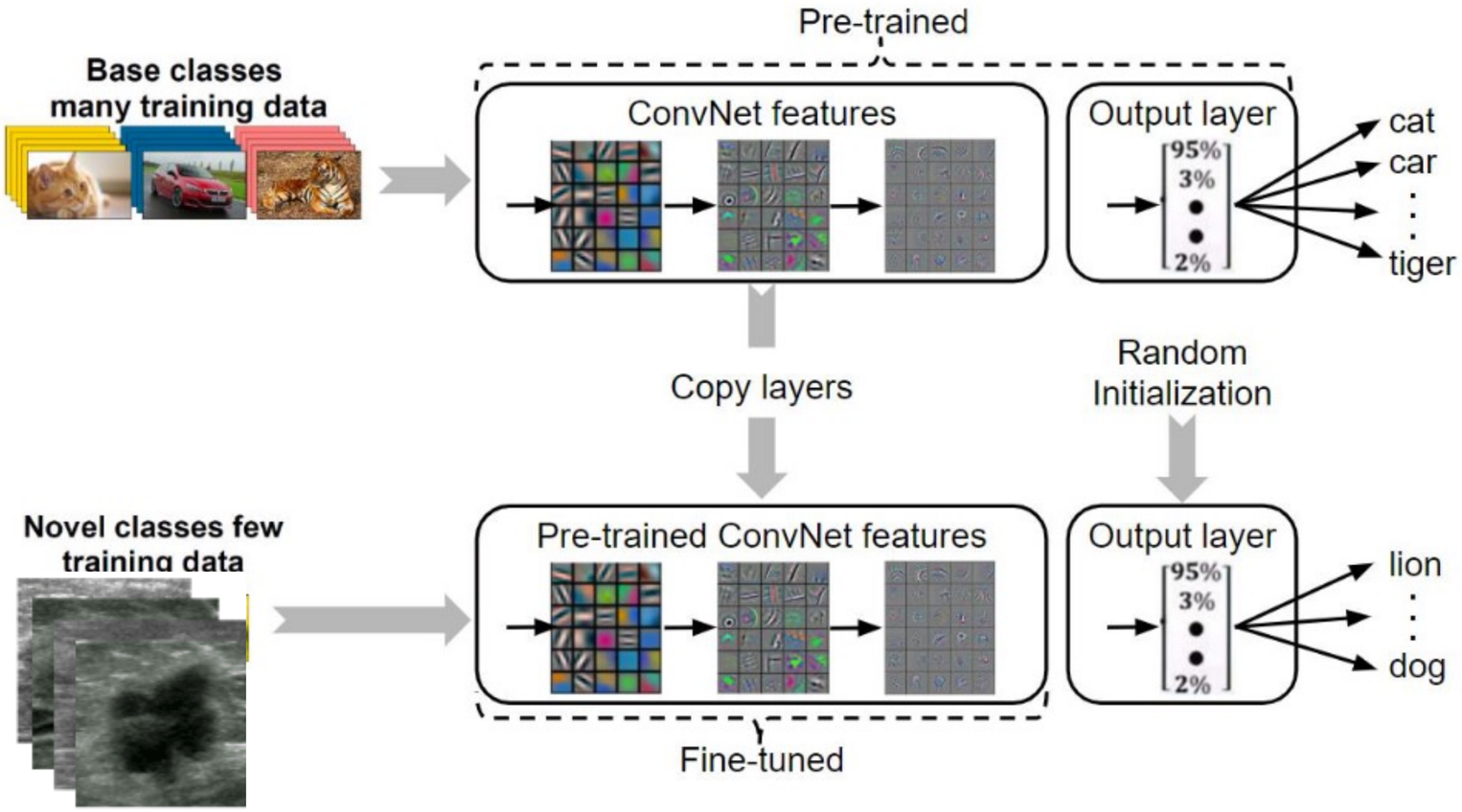


– Some hopefully helpful convolutional kernels than can serve as feature extractors (lines, blobs, texture, colour…)

Finetuning

ImageNet (large)

Medical ultrasound (small)



New terminology:

base classes == train classes
novel classes == test classes
} no overlap between them

Transfer learning vs finetuning

- **Training** just the new top layer(s) and **freezing** the old lower layers will help to preserve some of those basic feature extractors, learning new domain specific features just in the top layer(s)

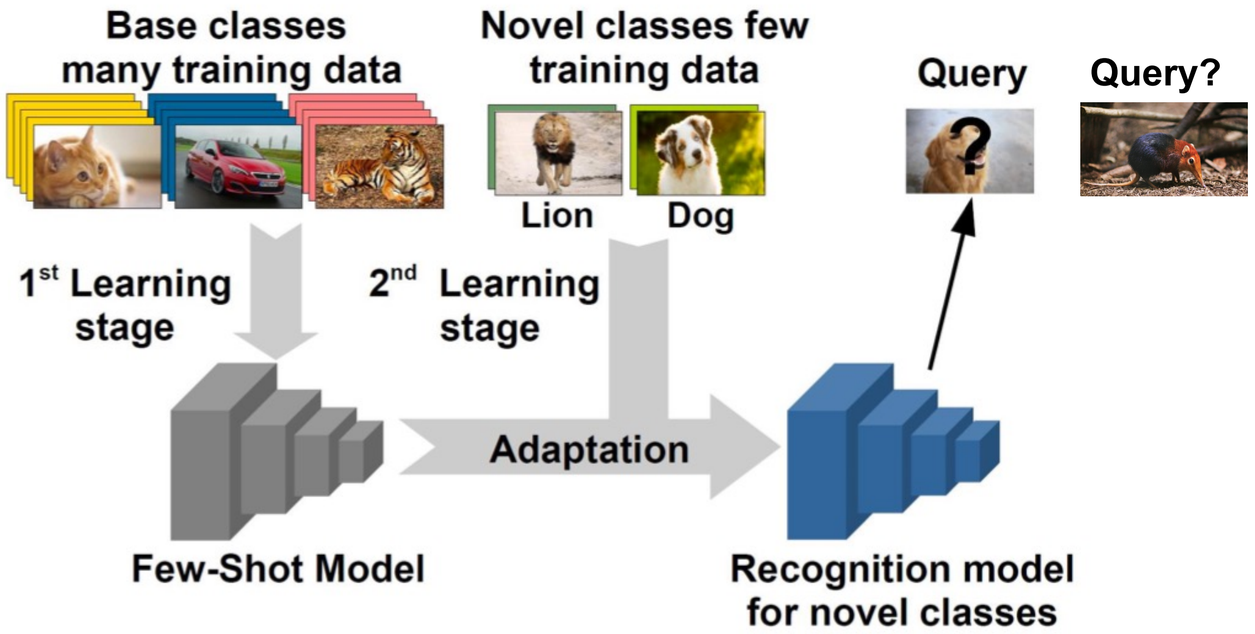
- **Finetuning** is an optional step in transfer learning: retraining the entire network (all layers**) carries the risk of:**
  - **Forgetting** previously learnt features
  - **Overfitting** to new smaller domain **in case of extremely limited data**
    - This is called: **few-shot**

Overview

- Transfer learning (revisited)
- **Few-shot learning**
- Meta-learning
- Incremental learning
- Curriculum learning

Overcome scarcity with transfer learning

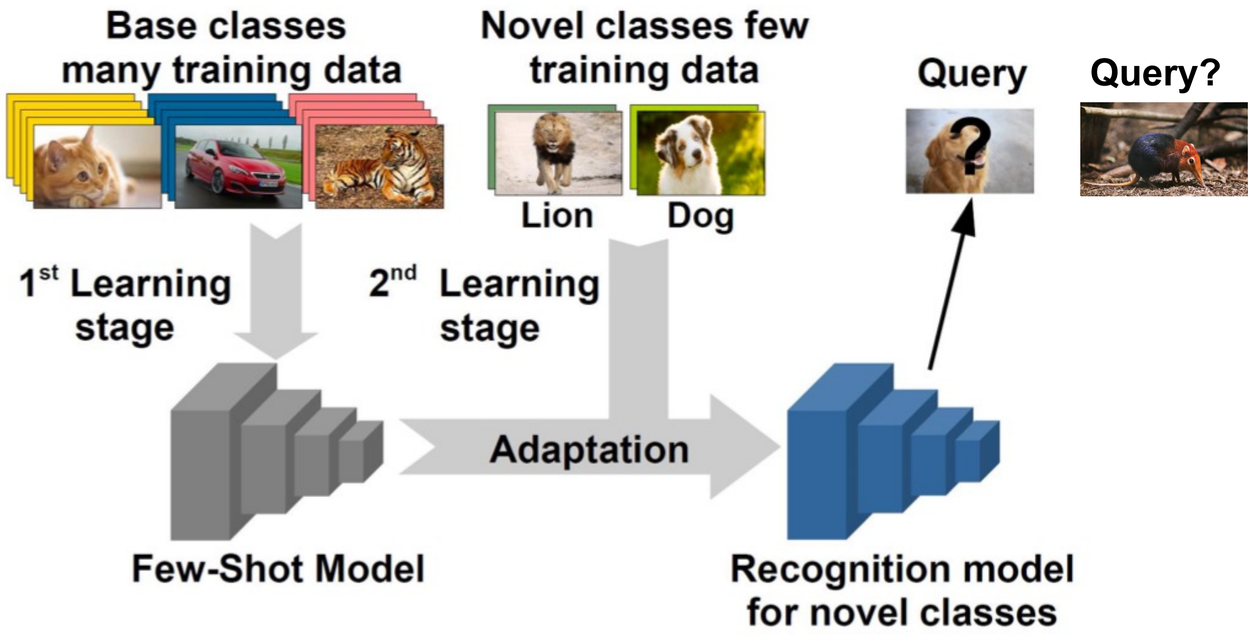
- Recipe followed by all **few-shot learning** techniques



1. **Acquire knowledge**: train on similar problems
2. **Transfer knowledge**: adapt to the problem of interest

Overcome scarcity with transfer learning

- Recipe followed by all **few-shot learning** techniques



1. **Acquire knowledge**: use many training data from some **base classes**
2. **Transfer knowledge**: adapt to **novel classes** with few training data

Few-shot learning

- Few-shot learning is the problem of making **predictions based on a limited number of samples**.

- This is different from **standard supervised learning**, where we let the model recognise the images in the training set and then generalise to the test set.

- Instead, the goal of few-shot learning is to “**learn to learn**”.

= Learning novel concepts from limited data

Overview

- Transfer learning (revisited)
- Few-shot learning
- **Meta-learning**
- Incremental learning
- Curriculum learning

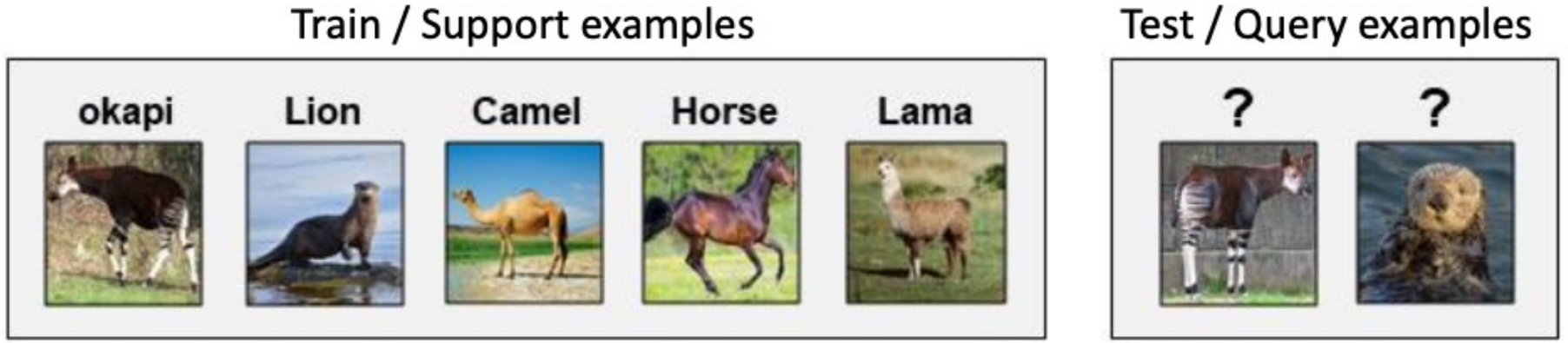
Meta-learning paradigm: *Learn to learn*

- Most (but not all) few-shot methods use **meta-learning**
  - *"Evolutionary principles in self-referential learning, or on learning how to learn",* Schmidhuber 1987
  - *"Meta-neural networks that learn by learning",* Naik et al. 1992
  - *"Lifelong learning algorithms",* Thrun 1998
  - *"Learning to learn by gradient descent by gradient descent",* Andrychowicz et al. 2016
  - ...

- **What is few-shot meta-learning**?

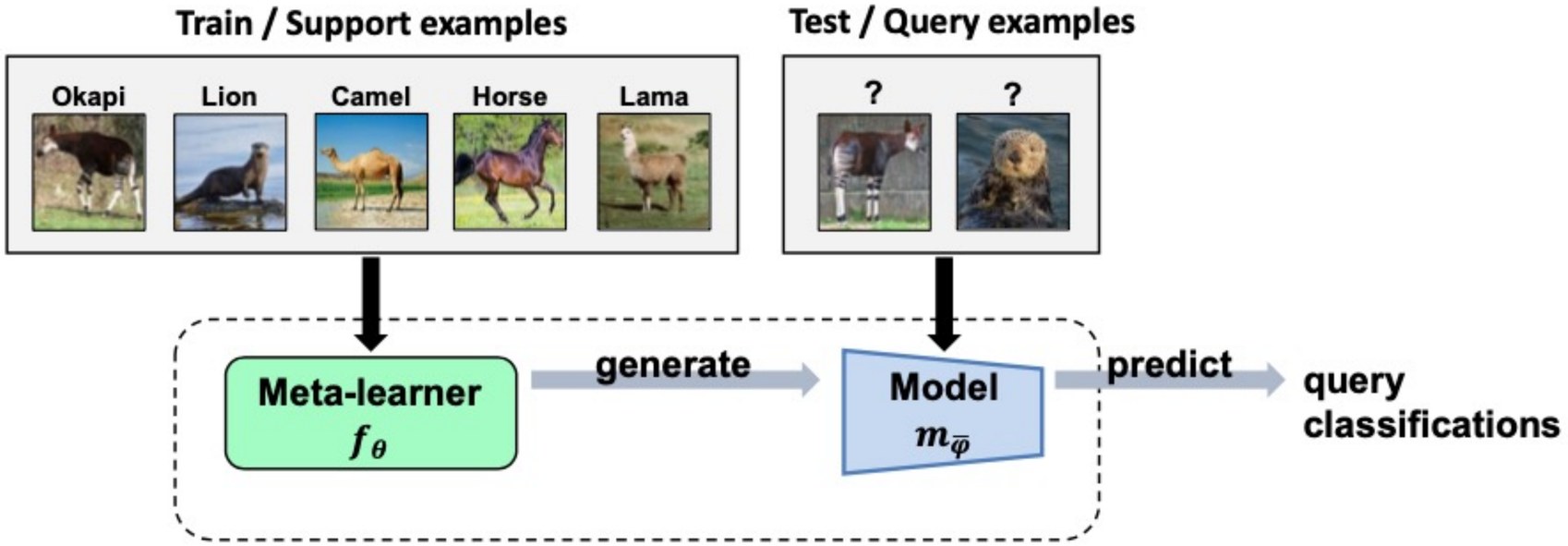
Formally: Learn N-way K-shot classification tasks

- **N =** number of classes
- **K =** training examples per class, **as small as 1 or 5!**



Example: 5-way 1-shot classification task

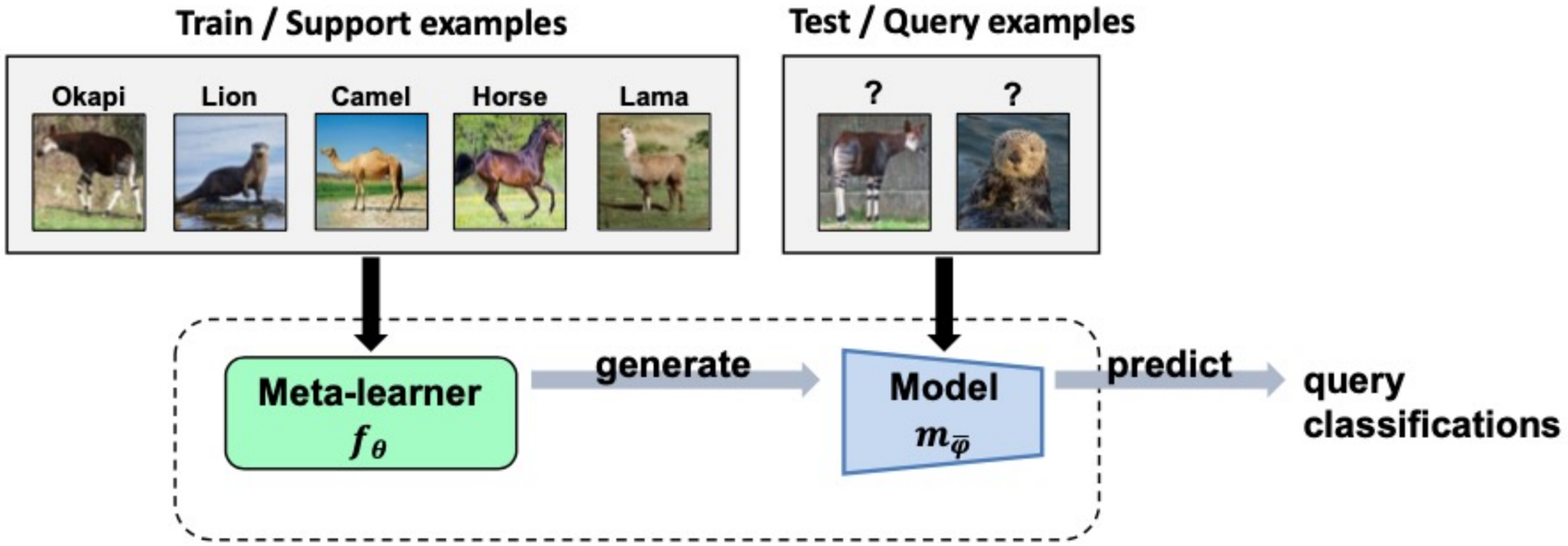
Few-shot classification with meta-learning



New terminology:

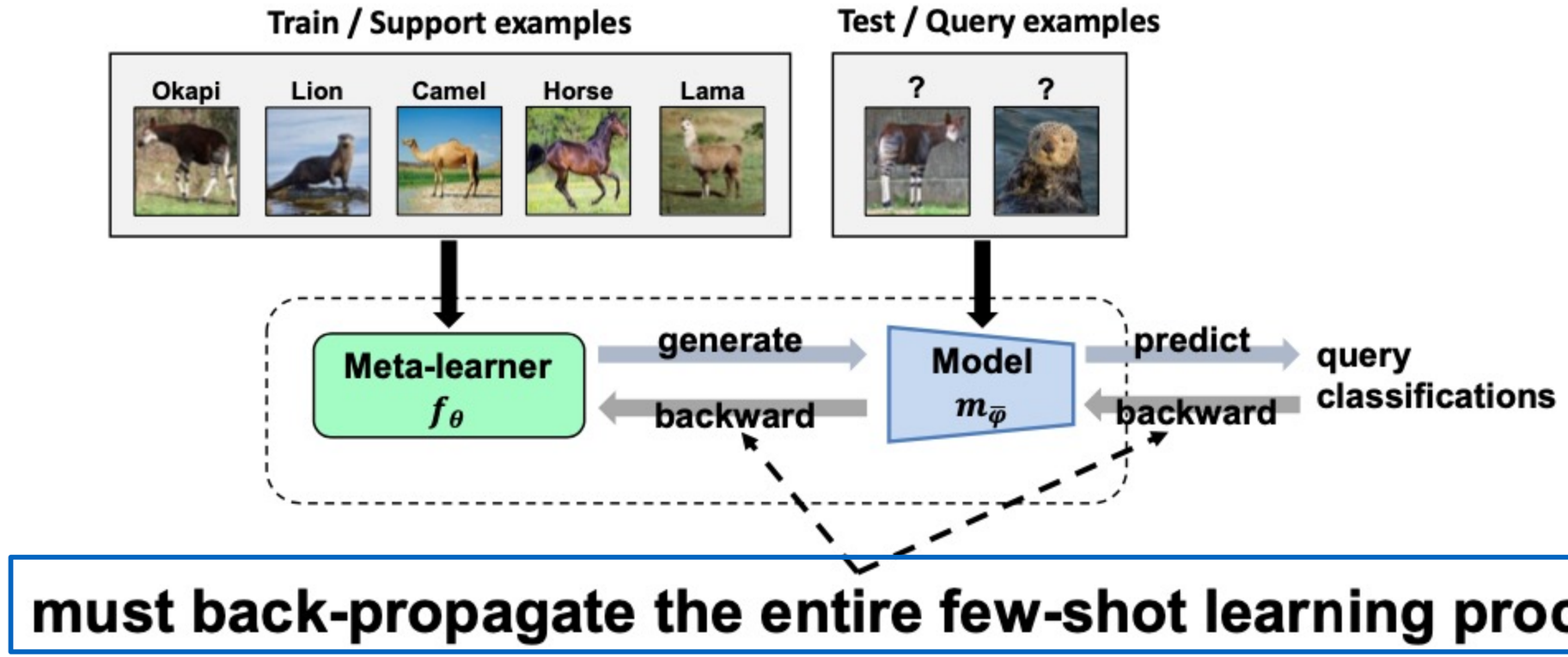
- **Input**: labelled **support data**, unlabelled **query data**
- **Intermediate output:** **model** for classifying the query images
- **Output:** predicted **query labels**

Few-shot classification with meta-learning



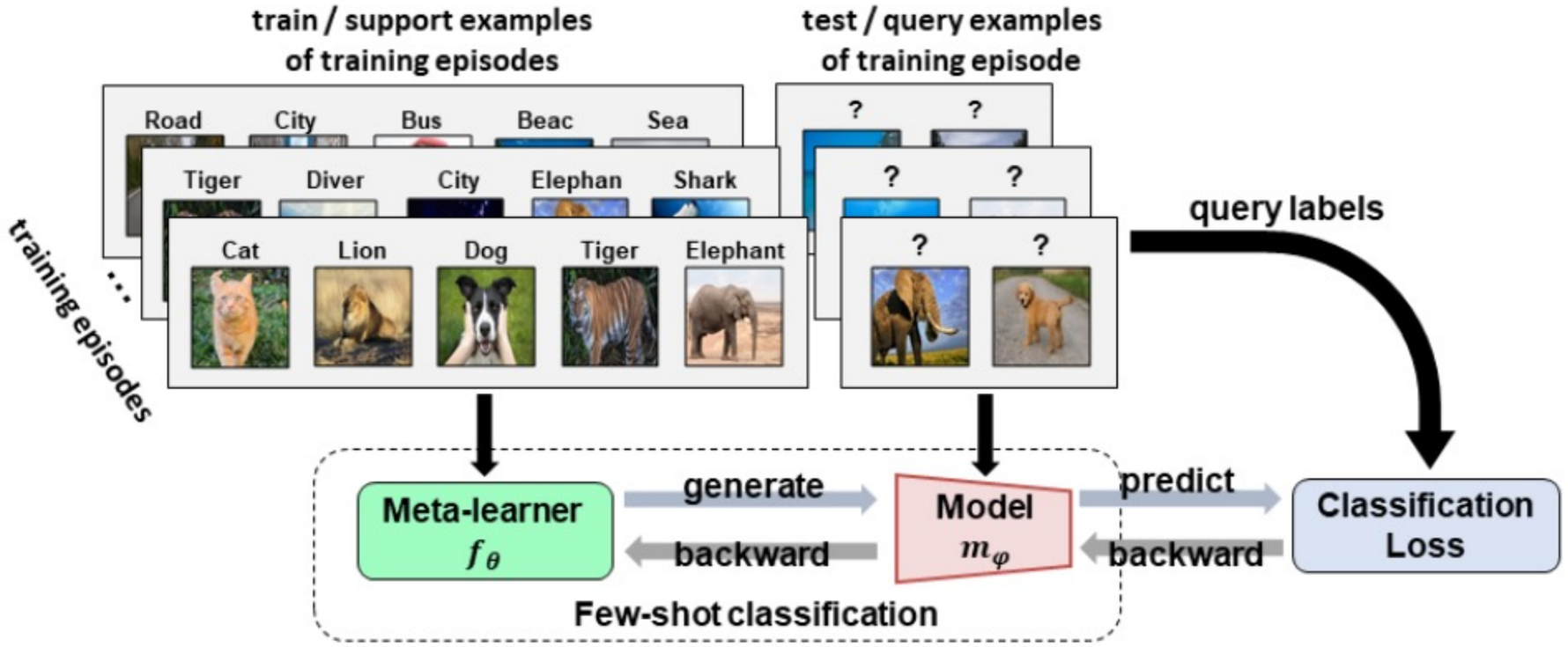
- **Train the learning algorithm** (instead of the classification model)
  - Implement it with a **meta-learner f_θ (somehow)**
  - Optimize f_θ on solving few-shot classification tasks **(learn-to-learn)**

Few-shot classification with meta-learning



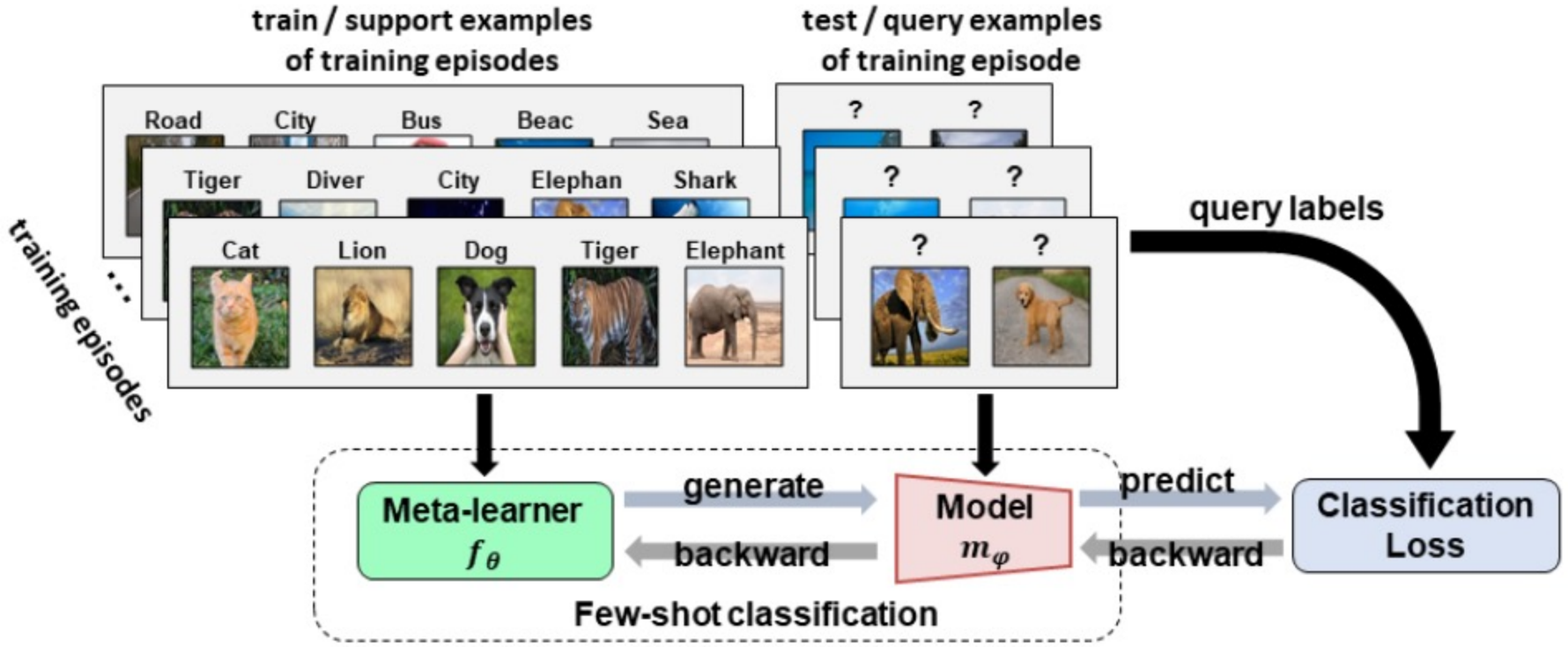
- **Train the learning algorithm** (instead of the classification model)
  - Implement it with a **meta-learner f_θ (somehow)**
  - Optimize f_θ on solving few-shot classification tasks **(learn-to-learn)**

Meta-learning: training time (1st learning stage)



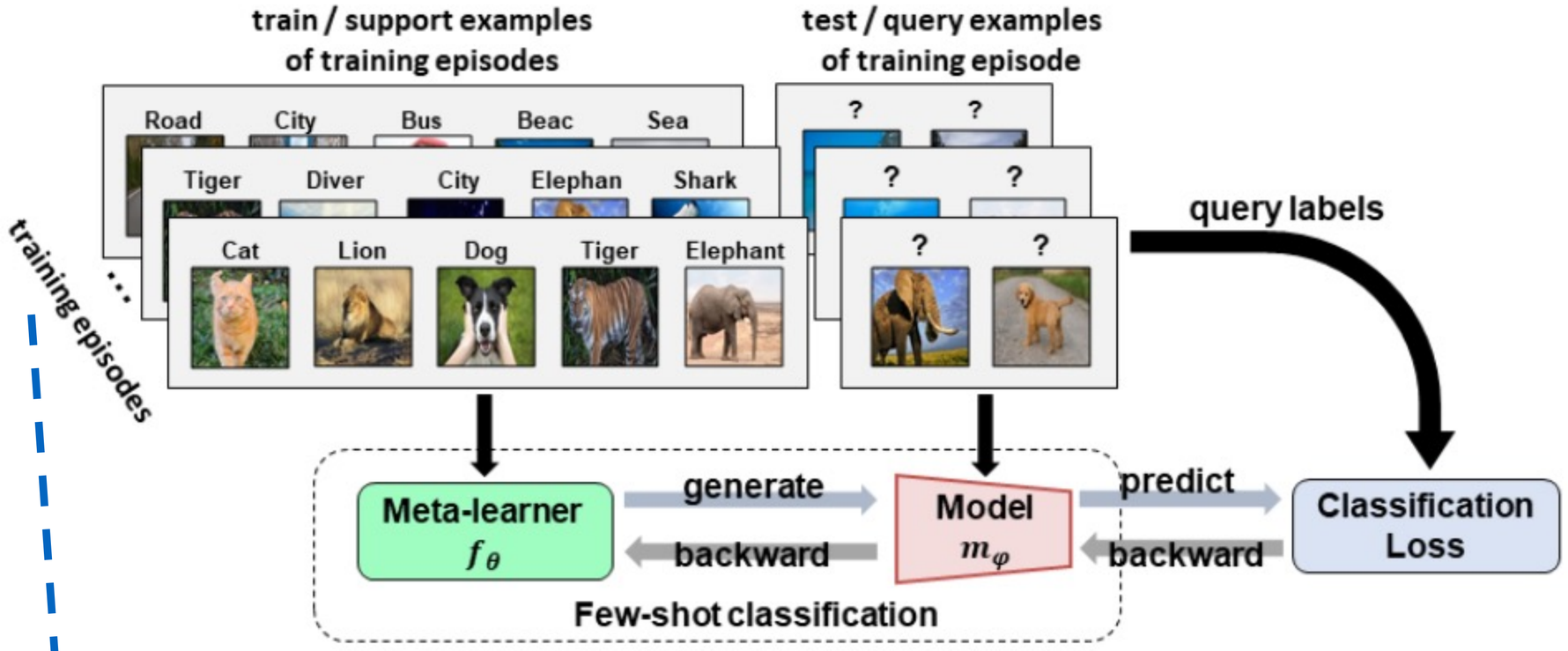
- **How to train a meta-learner**?
  - Train it on the same conditions it will be used in 2nd learning stage (meta-test)

Meta-learning: training time (1st learning stage)



- **How to train a meta-learner?**
  - Train **meta-learner** f_θ on solving a distribution of few-shot tasks (aka **episodes**)
  - Construct such **training episodes** using the base class data
    - by sampling **N** classes x (**K** support examples + **M** query examples)

Meta-learning: training time (1st learning stage)

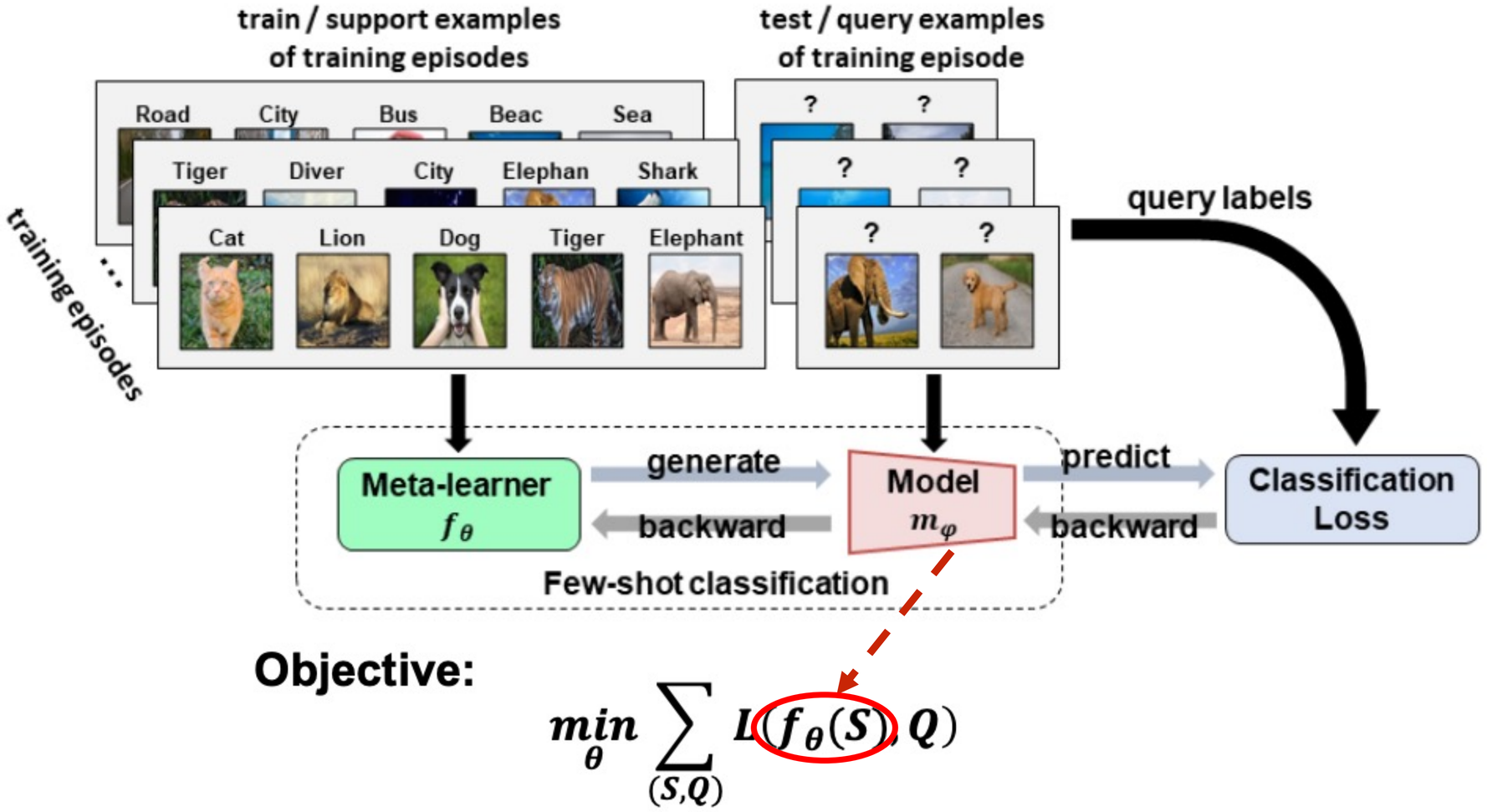


Objective:

$$\min_{\theta} \sum_{(S,Q)} L(f_\theta(S), Q)$$

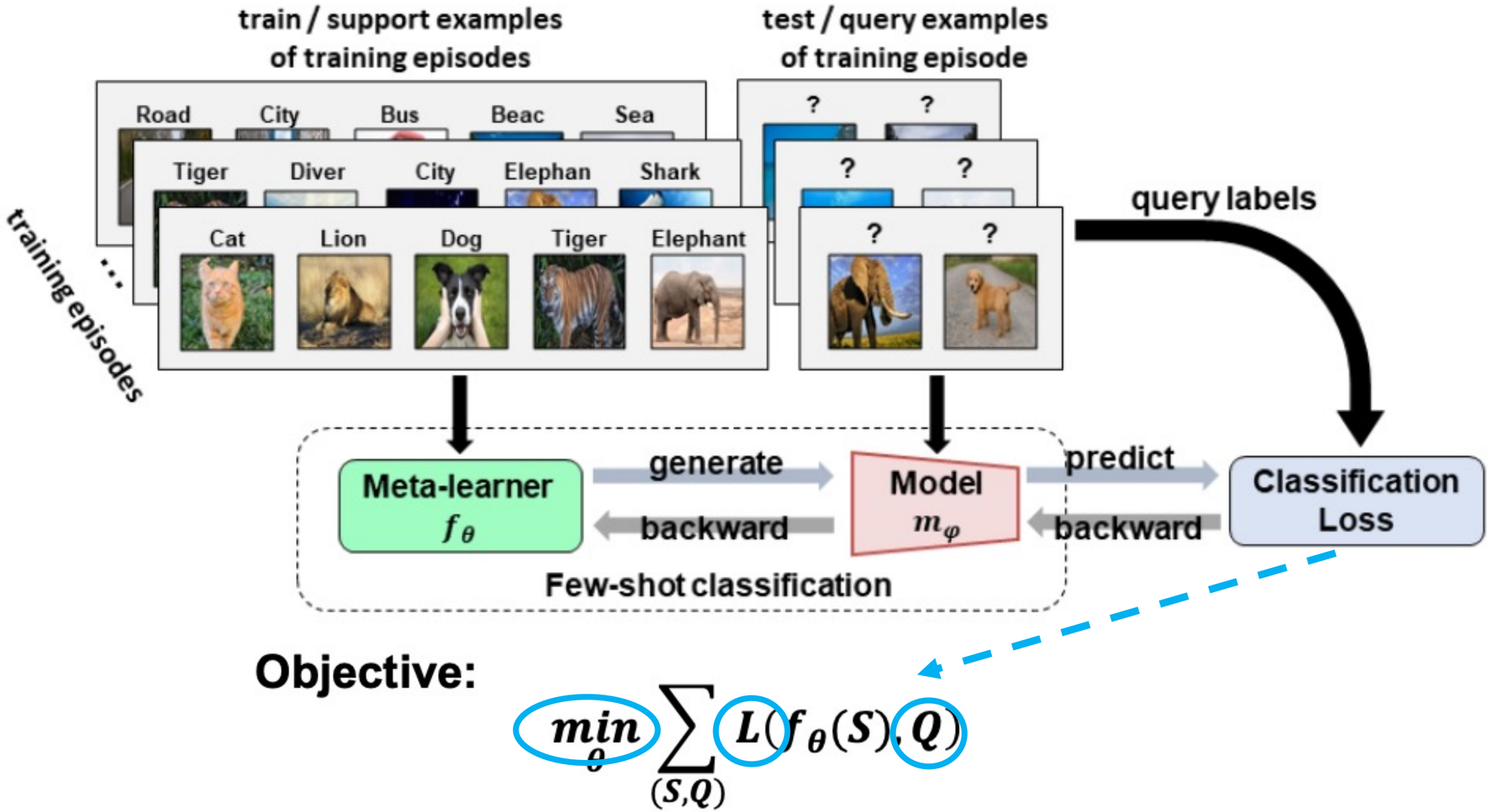
Episode *(S,Q)*: support set $S = \{x_k^S, y_k^S\}_{k=1}^{N*K}$ and query set $Q = \{x_m^Q, y_m^Q\}_{m=1}^{N*M}$

Meta-learning: training time (1st learning stage)



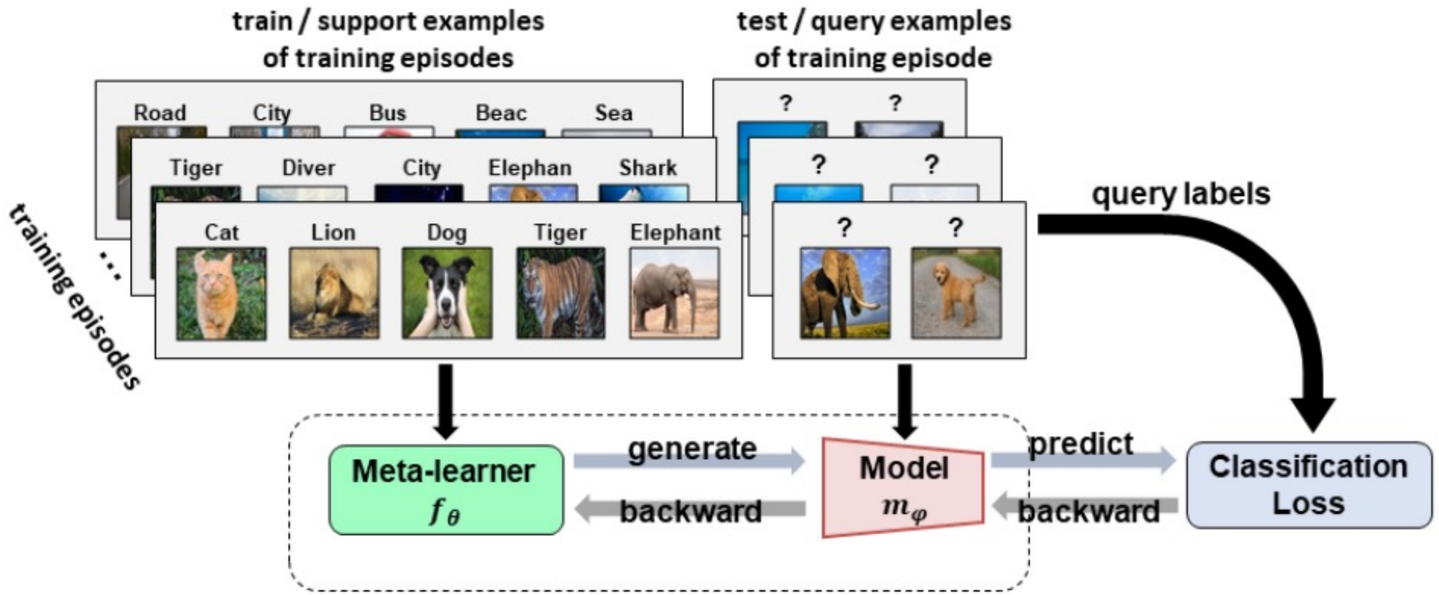
Inner part: generate using the support set *S* the classification model $m_\varphi = f_\theta(S)$

Meta-learning: training time (1st learning stage)



Outer part: optimize θ w.r.t. the queries classification loss $L(f_\theta(S), Q) = L(m_\varphi, Q)$

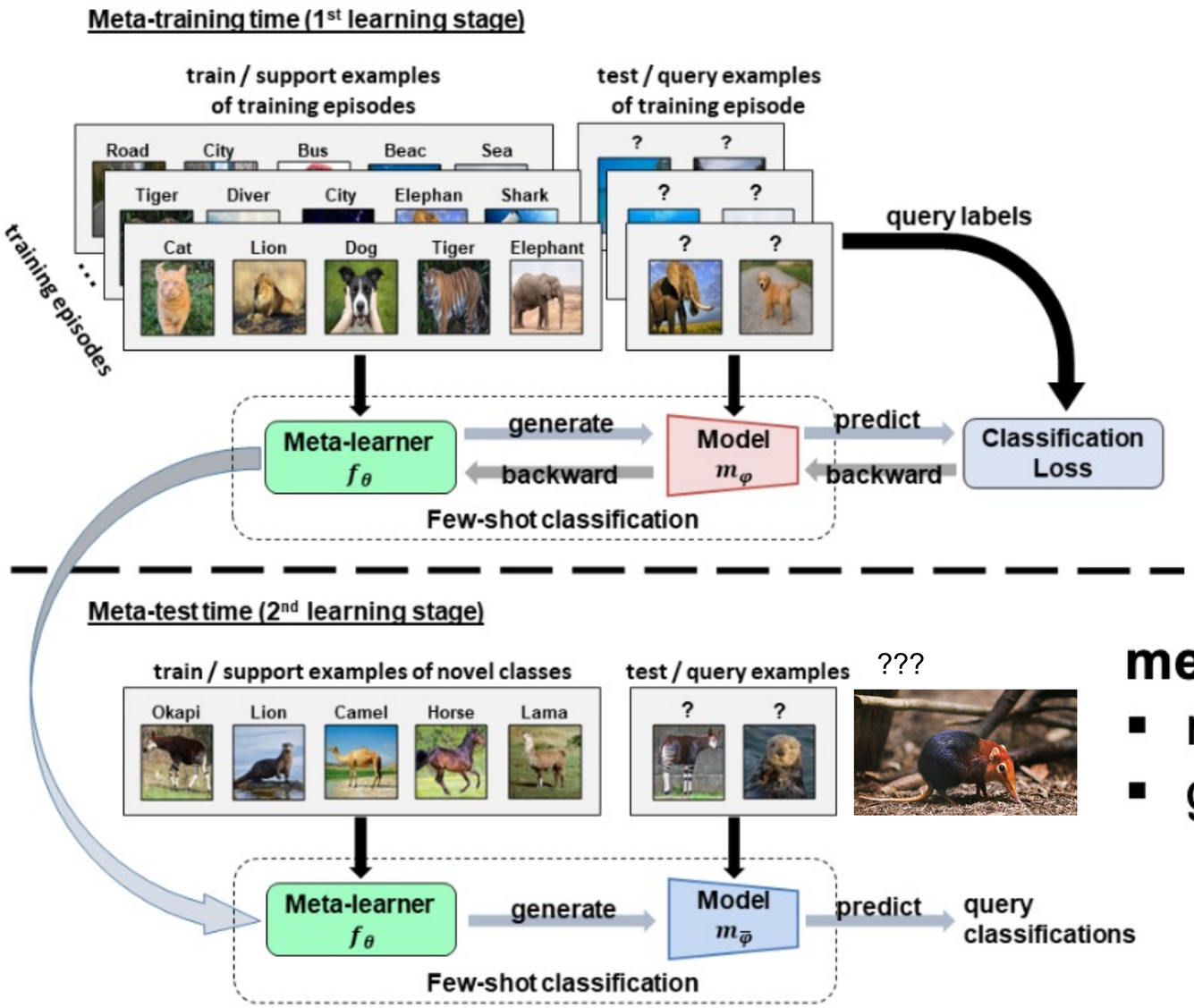
Meta-learning: training time (1st learning stage)



Meta-training routine:

1. Sample training episode (S, Q)
2. Generate classification model $m_\varphi = f_\theta(S)$
3. Predict classification scores $p_m = m_\varphi(x_m^Q)$ for each x_m^Q in Q
4. Optimize θ w.r.t. the queries classification loss $L(f_\theta(S), Q)$

Meta-learning: test time (2nd learning stage)



meta-learner at test time:

- remains fixed (typically)
- generates a model for novel classes

From **supervised learning** to **meta-learning**

- training -> meta-training
- test time -> meta-test time
- mini-batch of images -> mini-batch of few-shot episodes
- training data -> meta-training data
  = all possible training episodes
- test data -> meta-test data
  = test episodes

Few-shot learning vs Meta-learning

- **Few-shot learning:**
  - Any transfer learning method that targets on transferring well with limited data
  - E.g.: pre-train + fine-tuning, or using meta-learning

- **Meta-learning:**
  - Learn the learning algorithm itself
  - *"Learning to learn by gradient descent by gradient descent",* Andrychowicz et al. 2016
  - Ingredient of many few-shot algorithms
  - Also used in multi-task learning, reinforcement learning... (which we will deal with later!)

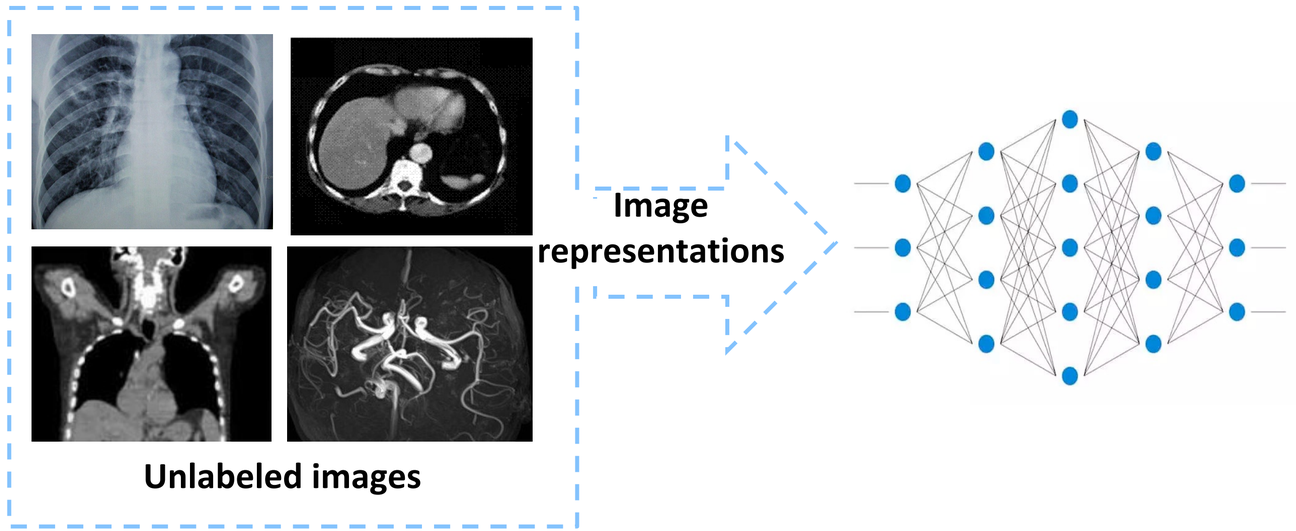
Example: SSL-ALPNet

Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

- **Self-supervised few-shot semantic segmentation** framework for medical imaging
- **Main idea**: Learn generalisable image representations directly from unlabeled images



Example: SSL-ALPNet

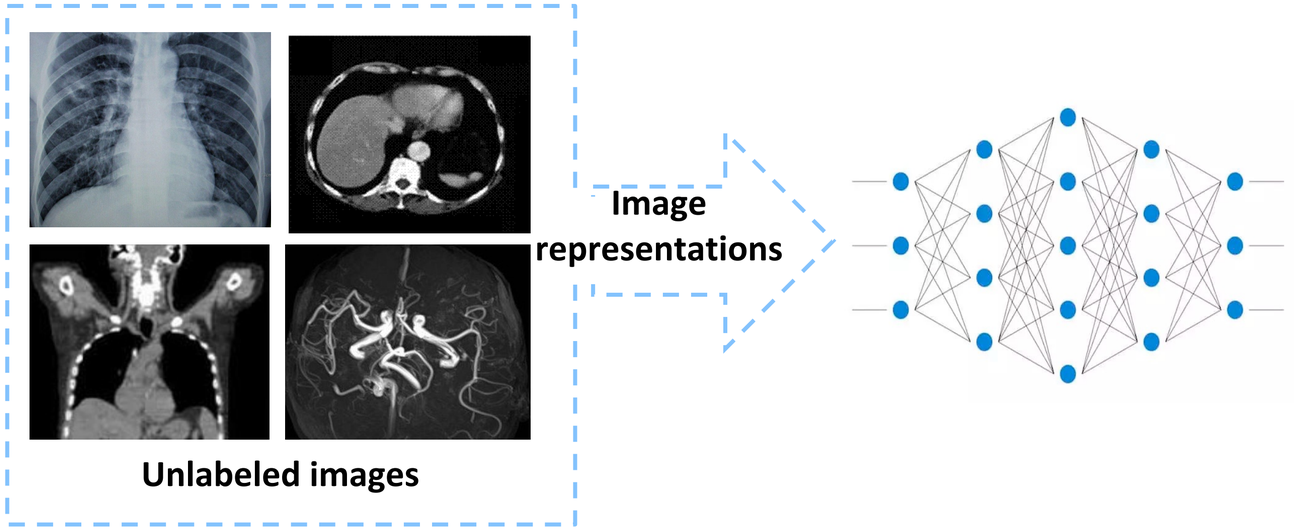
Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

Framework: SSL-ALPNet

- **SSL**: learning generalizable image representations by self-supervised learning (SSL)
- **ALPNet**: improving network representation capability with adaptive local prototype pooling (ALP)



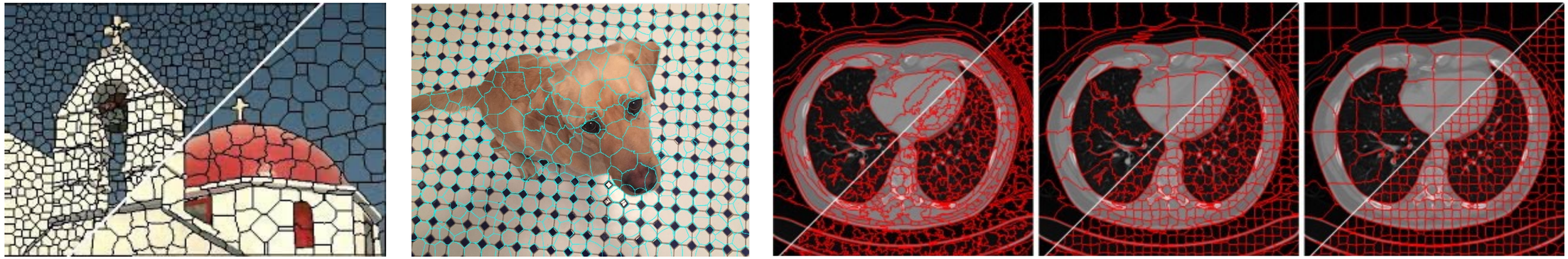
Example: SSL-ALPNet

Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

1. Superpixel*-based self-supervised learning for eliminating the need for manual annotations:



   – *Superpixels are an image clustering technique and thus an unsupervised method themselves, but here are enabling self-supervised learning (similar to pretext learning)

2. Adaptive local prototype pooling empowered prototypical network (ALPNet)
   – Improve segmentation accuracy by preserving local information in learnt representations

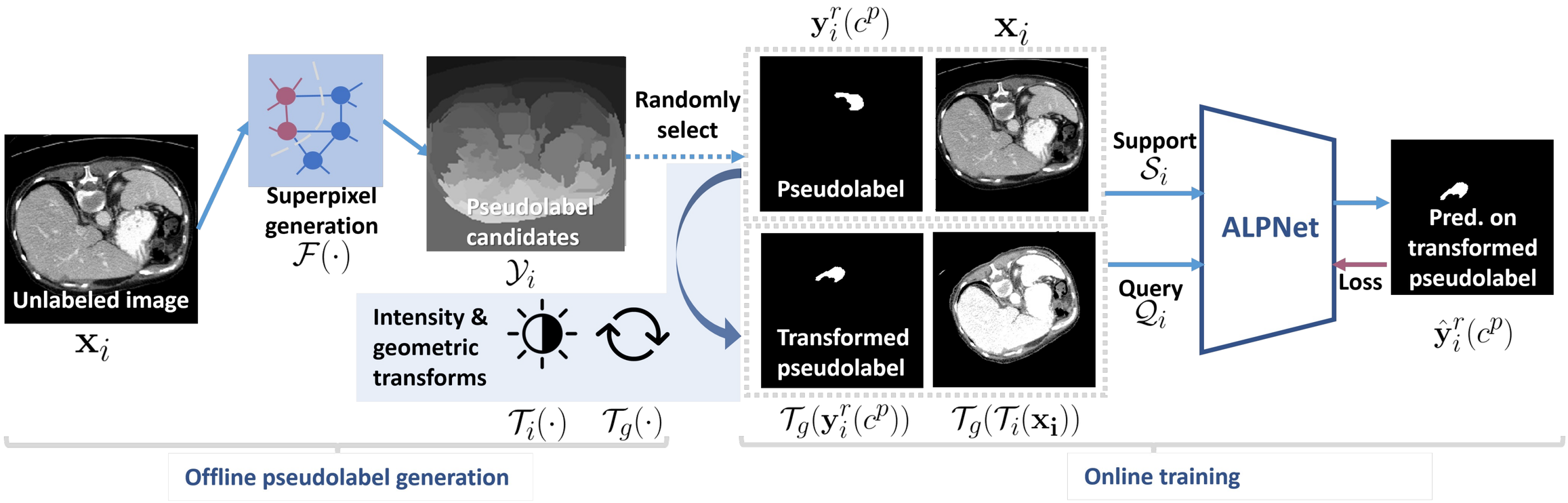
Example: SSL-ALPNet

Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

- Workflow:



Segmentation loss propagates back through the network

Example: SSL-ALPNet

Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

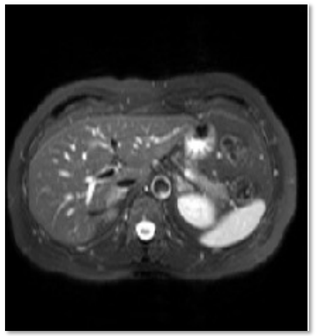
Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

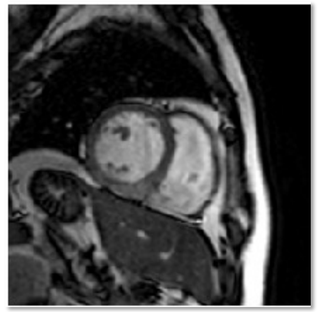
- Evaluated on three different combinations of ROI, labels and imaging modalities

ROI	Modality	Sequence	View	Labels	Clinical Data?
Abdominal	CT	Non-contrast	Axial	L-kidney R-kidney Spleen Liver	Y
Abdominal	MRI	T2-SPIR	Axial		N
Cardiac	MRI	bSSFP	Short-axis	L-ventricle R-ventricle Myocardium	Y

Abdominal CT



Abdominal MRI



Cardiac MRI

Example: SSL-ALPNet

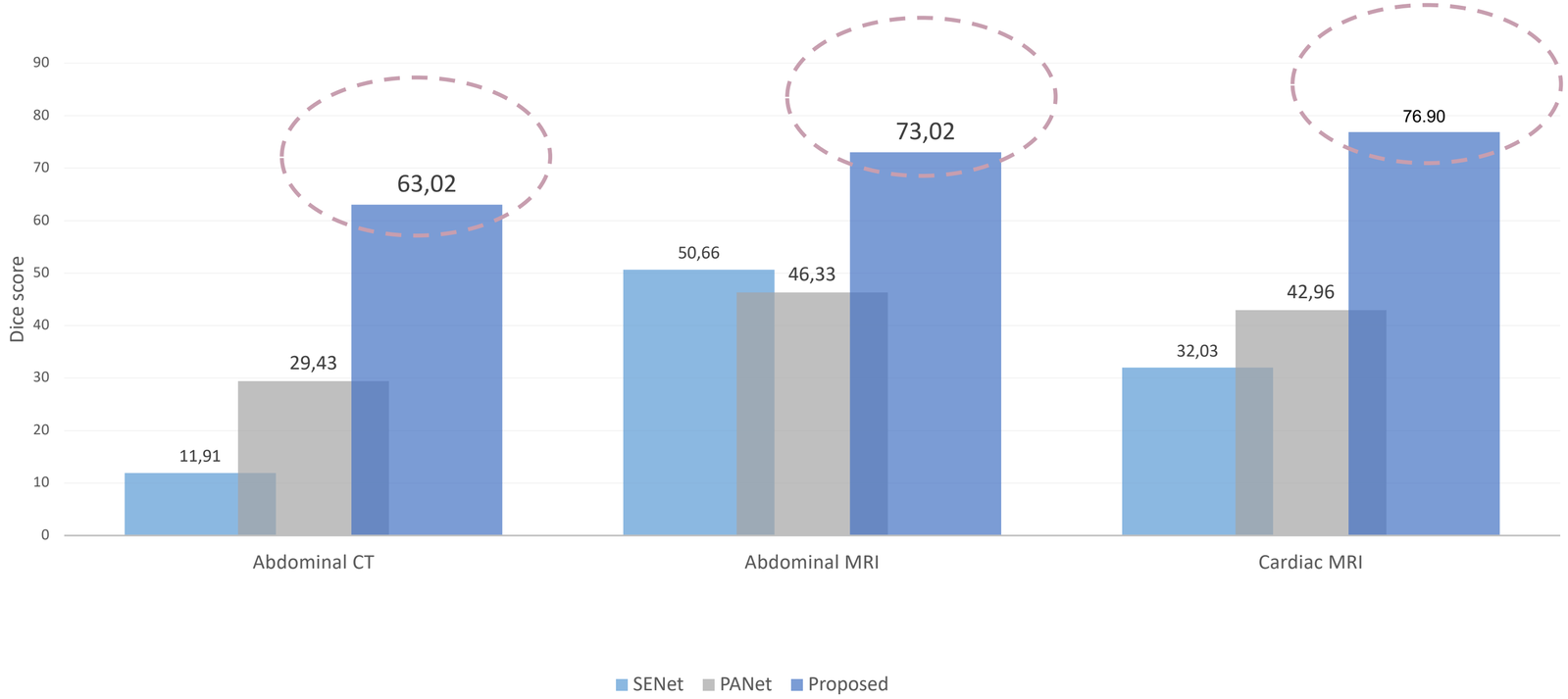
Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

- Compared with peer methods
  - PANet: Baseline prototypical network (w/o ALP Module)
  - SE-Net: FSS model specially designed for medical imaging

Require manual annotations during training



- The proposed method outperforms peer methods by large margins.
- Unlike peer methods, the proposed method does not require manual annotations during training.

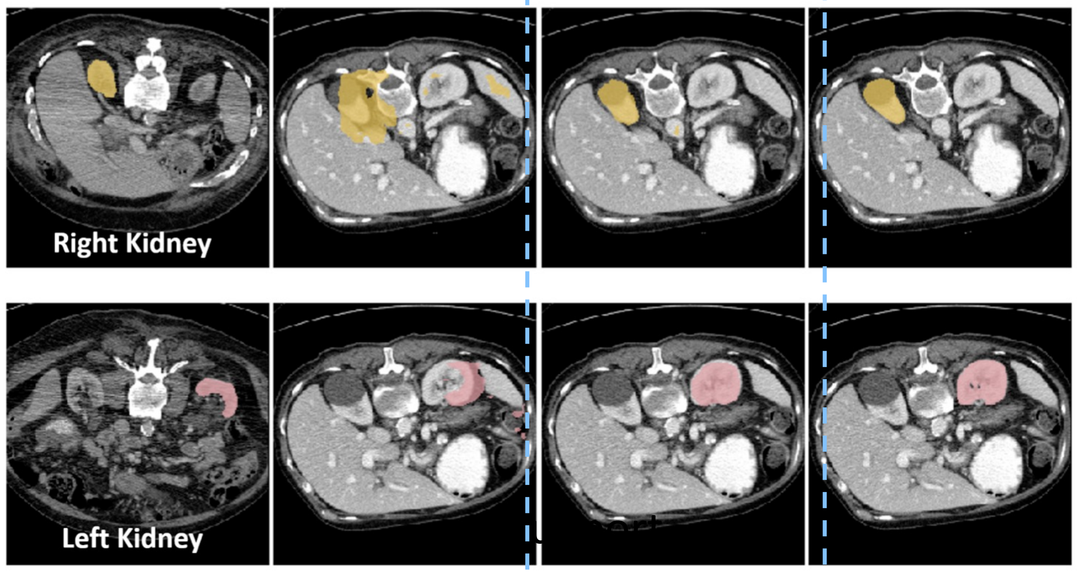
Wang et al., ICCV, 2019; Roy et al., MIA, 2020;

Self-supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation

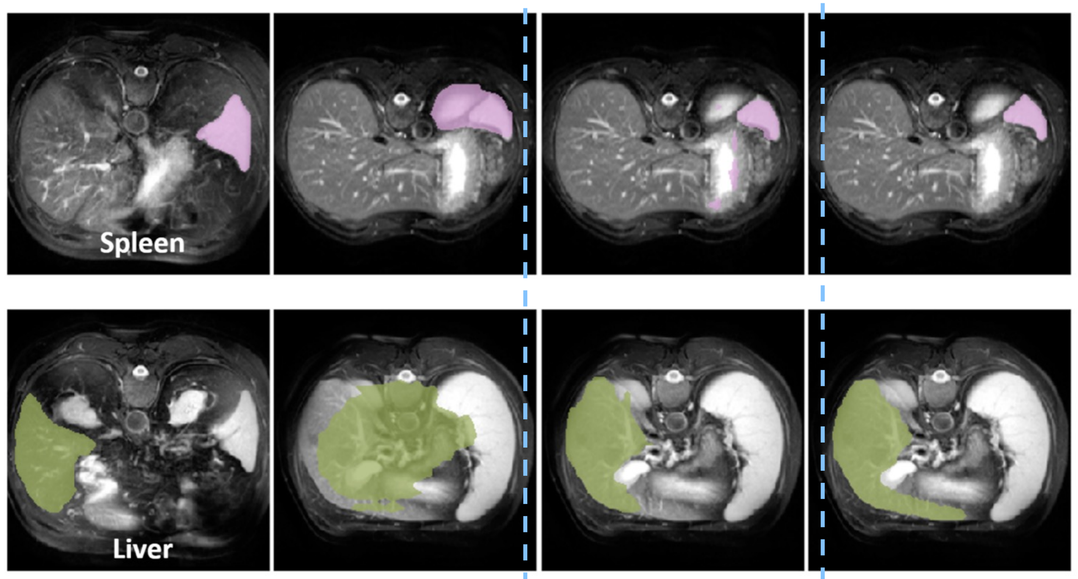
Cheng Ouyang[1](✉), Carlo Biffi[1*], Chen Chen[1*], Turkay Kart[1*], Huaqi Qiu[1], and Daniel Rueckert[1]

BioMedIA Group, Department of Computing, Imperial College London, UK
c.ouyang@imperial.ac.uk

- Abdominal CT

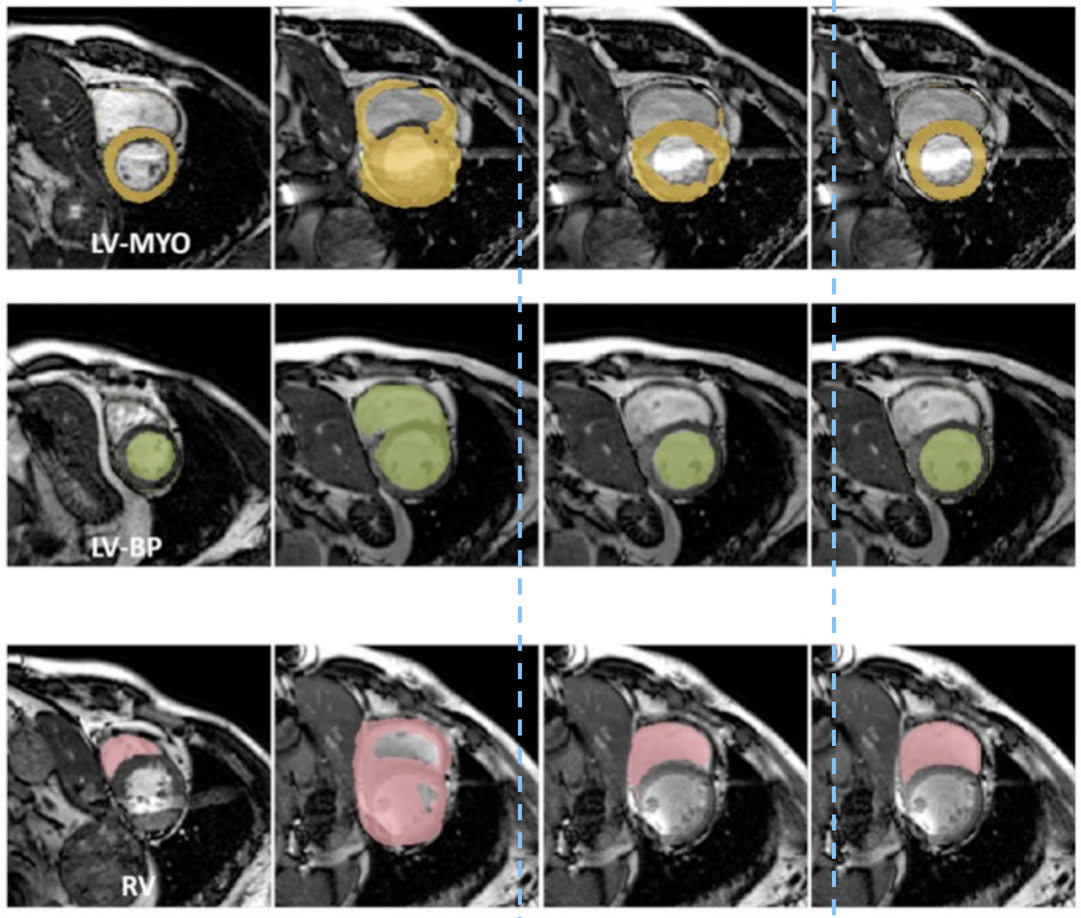


- Abdominal MRI



Support PANet Proposed Ground Truth

- Cardiac MRI



Support PANet Proposed Ground Truth

Overview

- Transfer learning (revisited)
- Few-shot learning
- Meta-learning
- **Incremental learning**
- Curriculum learning

Incremental learning

- **Incremental learning:** method of machine learning in which input data is continuously added to extend the existing model's knowledge i.e. to further train the model.
  - Dynamic technique of supervised learning plus unsupervised learning
  - Can be applied when training data becomes available gradually over time or its size is out of system memory limits *(scalability issues)*

- *The aim of incremental learning is for the learning model to adapt to new data without forgetting its existing knowledge.*

= Learning new concepts without forgetting previous concepts

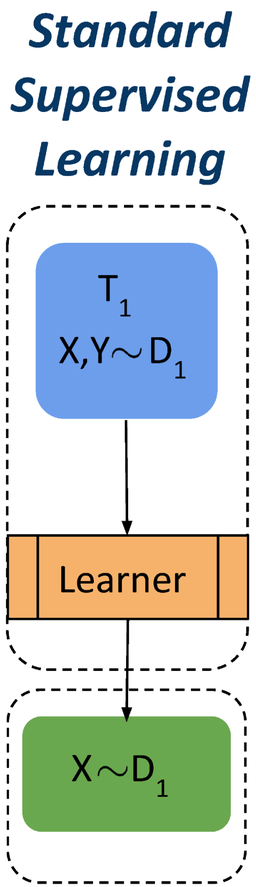
(one of the hallmarks of human intelligence)

Other names for incremental learning

- **Continual learning**
  - model that learns a large number of tasks sequentially without forgetting knowledge obtained from the preceding tasks, *where the data in the old tasks are not available any more during training new ones.*
- **Lifelong learning**
  - model that learns continuously, accumulates the knowledge learnt in the past, and uses/adapts it to help future learning and problem solving.
- **Never-ending learning**
  - Model that learns many types of knowledge, using previously learnt knowledge to improve subsequent learning, with sufficient self-reflection to avoid plateaus in performance as it learns.
- **Sequential learning**
  - Not to be confused with **sequence learning**, which is operating on sequential data (e.g. time series, speech data, …) – typically approached with recurrent networks like LSTM

Incremental learning

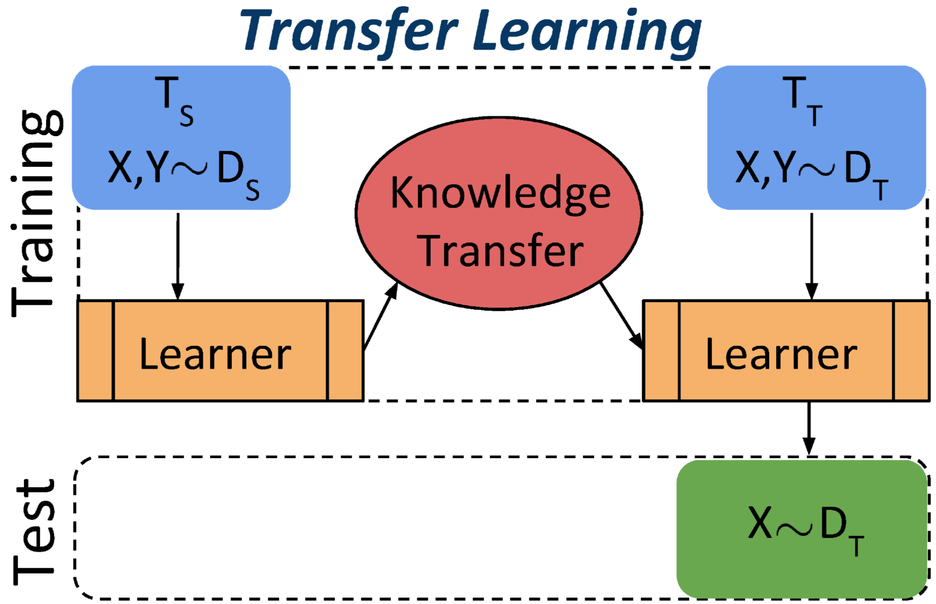
- **Standard supervised learning** assumes that training, validation and testing data are all sampled from the same distribution:



We can only predict the distribution we have trained on (D_1)

Incremental learning

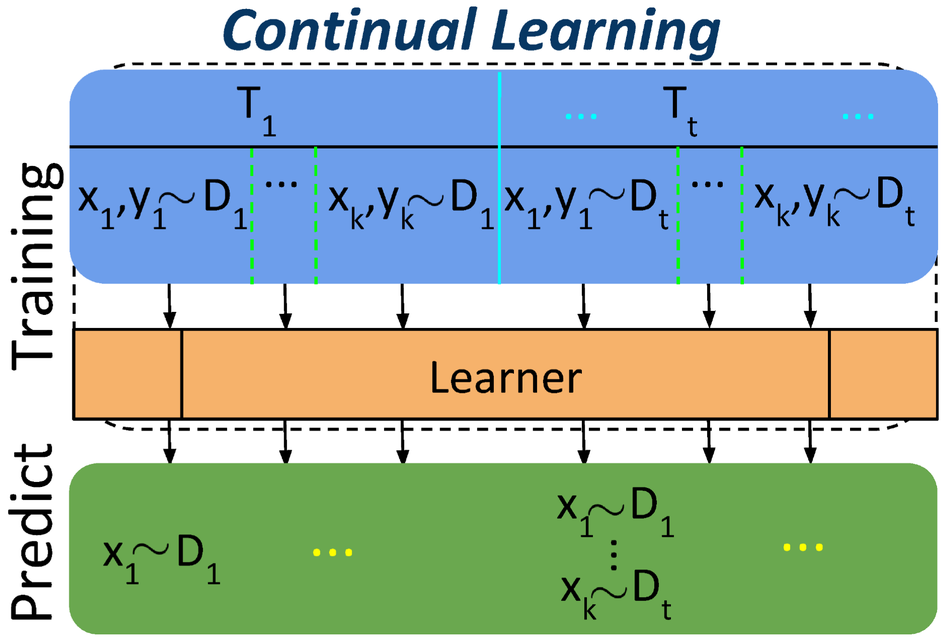
- **Transfer learning** facilitates some knowledge transfer from an original source distribution D_s to a new target distribution D_t:



We can only robustly predict the target distribution D_T, but in the process of retraining/finetuning may be less able to predict original source domain (D_S)

Incremental learning

- **Continual (=incremental) learning** adds further tasks or distributions, *without forgetting old ones*



We can predict, over time as new tasks or distributions are being added, any of the previously seen sample distributions.

Types of incremental learning

- **Task**-incremental learning:
  - e.g. going from image classification to detection or segmentation

- **Class**-incremental learning
  - E.g. going from class A to class B etc.
  - Or anrrowing down a class into subclasses (disease types)

- **Domain**-incremental learning
  - E.g. going from ImageNet to medical images
  - Or going from one imaging modality to another
  - Or just going from one scanner type to another

Approaches for incremental learning

- **Zero-shot learning**
  – No training step for unseen classes

- **Continuously update the training set**
  – Keep data and retrain

- **Use a fixed data representation**
  – Simplify the learning problem

Basic rules for incremental learning

1. Learn one task* after the other
2. Without storing data from previous tasks
3. Without memory footprint growing over time
4. Without forgetting old tasks

**task here also stands for class or domain*

Basic rules for incremental learning (relaxed)

1. Learn one task after the oher
2. Without storing (many) data from previous tasks
3. Withour memory footprint growing (significantly) over time
4. Without (completely) forgetting old tasks

**task here stands also for class or distribution*

Regularisation-based incremental learning

- When training a new task:
  - Add a regularisation term to the loss function.
  - E.g. to penalise “catastrophic forgetting”

- We distinguish between:
  - data-focused methods
  - model-focused or prior-focused methods

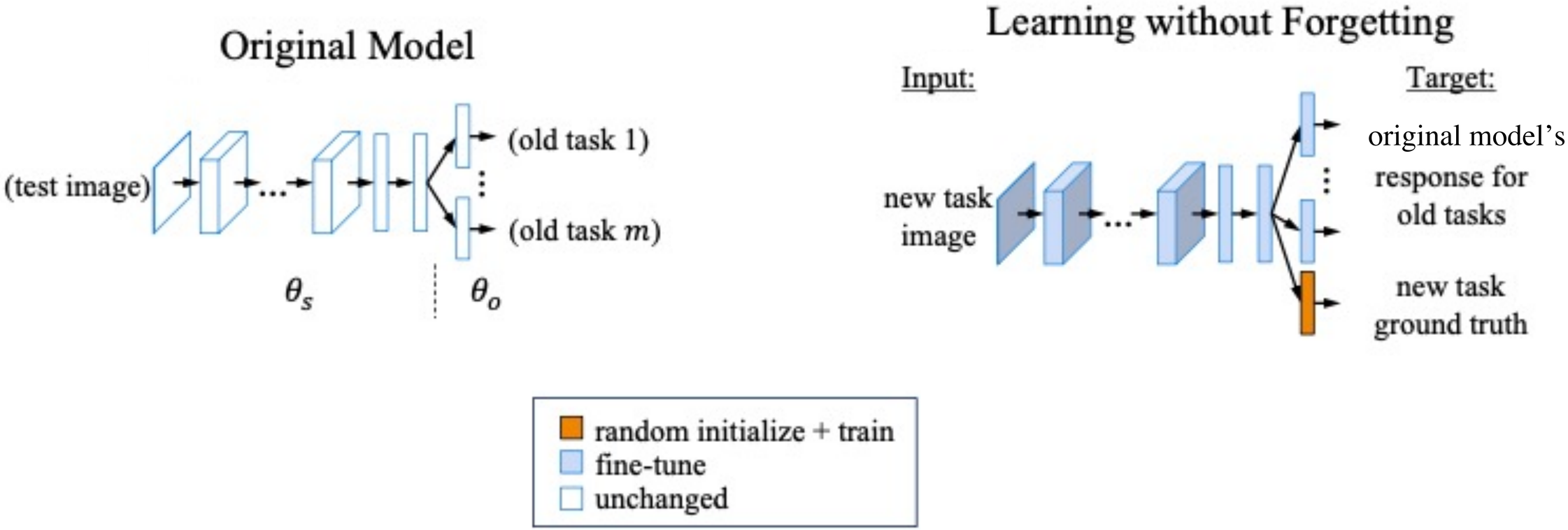
Learning without forgetting *(Li and Hoiem 2016)*

- Example method:
  - Add task-specific parameters θ_n for a new task
  - Learn parameters that work well on old ***and*** new tasks, using images and labels from ***only*** the new task (i.e., *without using data from existing tasks*).

- Advantage:
  - Simple method, good results
- Disadvantage:
  - Poor results for unrelated tasks

? Need to store old model

Learning without forgetting (Li and Hoiem 2016)



Learning without forgetting (Li and Hoiem 2016)

LEARNINGWITHOUTFORGETTING:

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

X_n, Y_n: training data and ground truth on the new task

Initialize:

$Y_o \leftarrow \text{CNN}(X_n, \theta_s, \theta_o)$ // *compute output of old tasks for new data*

$\theta_n \leftarrow \text{RANDINIT}(|\theta_n|)$ // *randomly initialize new parameters*

Train:

Define $\hat{Y}_o \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_o)$ // *old task output*

Define $\hat{Y}_n \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_n)$ // *new task output*

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n}{\text{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$

Learning without forgetting (Li and Hoiem 2016)

LEARNINGWITHOUTFORGETTING:

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

X_n, Y_n: training data and ground truth on the new task

Initialize:

$Y_o \leftarrow \text{CNN}(X_n, \theta_s, \theta_o)$ // *compute output of old tasks for new data*

$\theta_n \leftarrow \text{RANDINIT}(|\theta_n|)$ // *randomly initialize new parameters*

Train:

Define $\hat{Y}_o \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_o)$ // *old task output*

Define $\hat{Y}_n \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_n)$ // *new task output*

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n}{\text{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$

Loss balance weight

Responses of old model

Responses of new model

Model parameter regularisation

Data-focused vs Model-focused regularisation

- *Learning without Forgetting* in its original form had both data-focused and model-focused regularisation

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s,\hat{\theta}_o,\hat{\theta}_n}{\operatorname{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$

Weighted balance between old model and new model response

Old/new model parameter regularisation

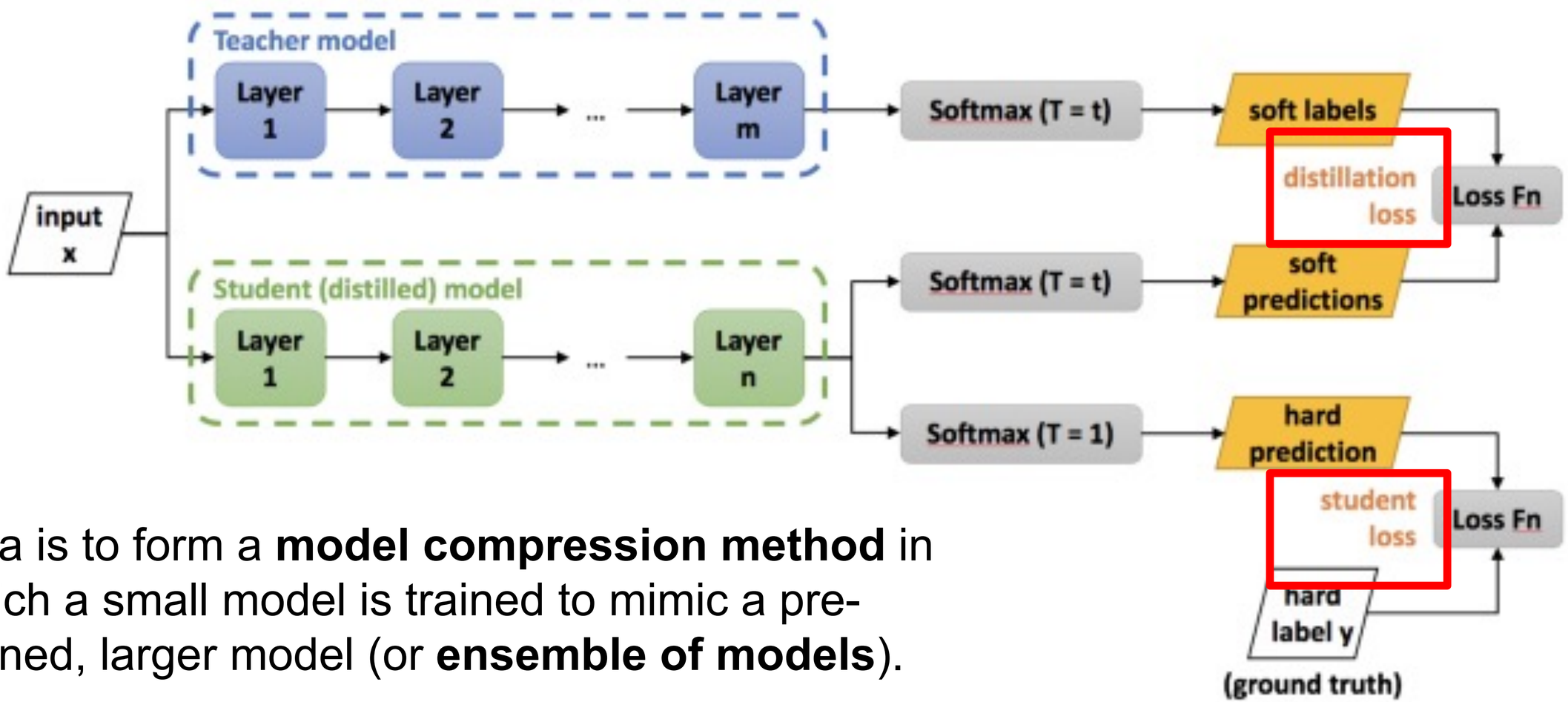
  - E.g. penalise changes to “important” model parameters, e.g.:

$$\mathcal{R}(\theta) = \alpha \sum_k \lambda_k (\theta_k^n - \theta_k^{n-1})^2$$

n: current task
n-1: previous task

Knowledge distillation

- **Knowledge distillation** is the process of transferring knowledge from a large model to a smaller one without loss of validity.



Idea is to form a **model compression method** in which a small model is trained to mimic a pre-trained, larger model (or **ensemble of models**).

Knowledge distillation

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

- Distillation loss:
  - Knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model.
  - I.e. the output of a softmax function on the teacher model's logits:

$$p_i = \frac{exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

where T is the temperature parameter.

i.e. when T=1 -> standard softmax function.

  - As T grows, the probability distribution generated by the softmax function becomes softer, providing more information as to which classes the teacher found more similar to the predicted class. (“**dark knowledge**”)
  - Use same T to compute the softmax on the student's logits (**distillation loss**)

Knowledge distillation

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†] Google Inc. Mountain View geoffhinton@google.com
Oriol Vinyals[†] Google Inc. Mountain View vinyals@google.com
Jeff Dean Google Inc. Mountain View jeff@google.com

- Student loss:
  - Calculate the "standard" loss between the student's predicted class probabilities and the ground-truth labels (also called "hard labels/targets").
  - When calculating the class probabilities for the student loss we use T=1.

- Overall loss function:

$$\mathcal{L}(x; W) = \alpha * \mathcal{H}(y, \sigma(z_s; T = 1)) + \beta * \mathcal{H}(\sigma(z_t; T = \tau), \sigma(z_s, T = \tau))$$

Student loss — Distillation loss

  - where x is the input, W are the student model parameters, y is the ground truth label
  - *H* is the cross-entropy loss function,
  - σ is the softmax function parameterized by the temperature T,
  - α and β are coefficients.
  - z_s and z_t are the logits of the student and teacher respectively.

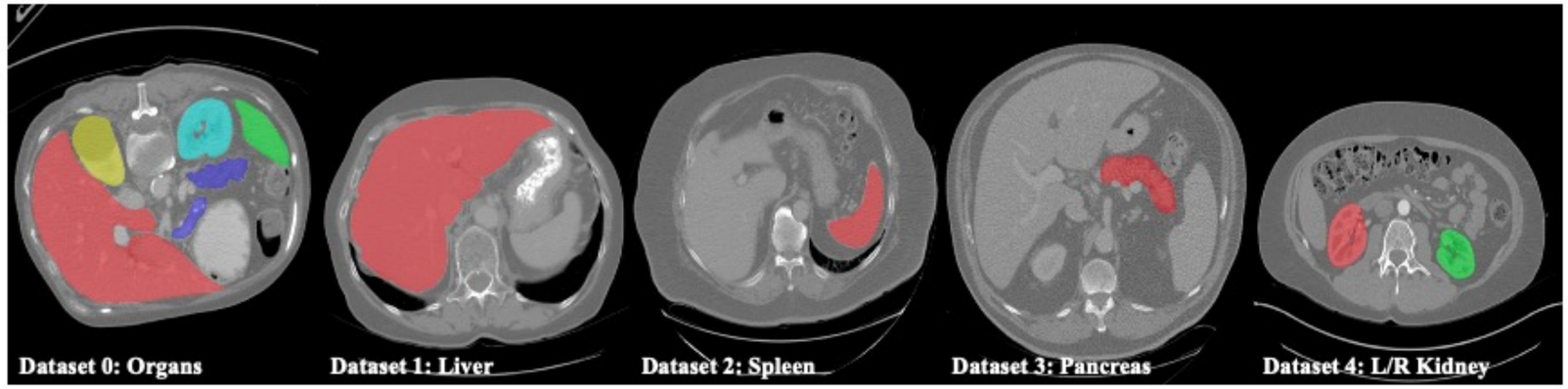
Example: Incremental learning for multi-label organ segmentation

Incremental Learning for Multi-organ Segmentation with Partially Labeled Datasets

Pengbo Liu[1], Li Xiao[1], and S. Kevin Zhou[1]

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{liupengbo2019,xiaoli}@ict.ac.cn
s.kevin.zhou@gmail.com

- Address problem of datasets for organ segmentation, which are partially annotated, and sequentially constructed :



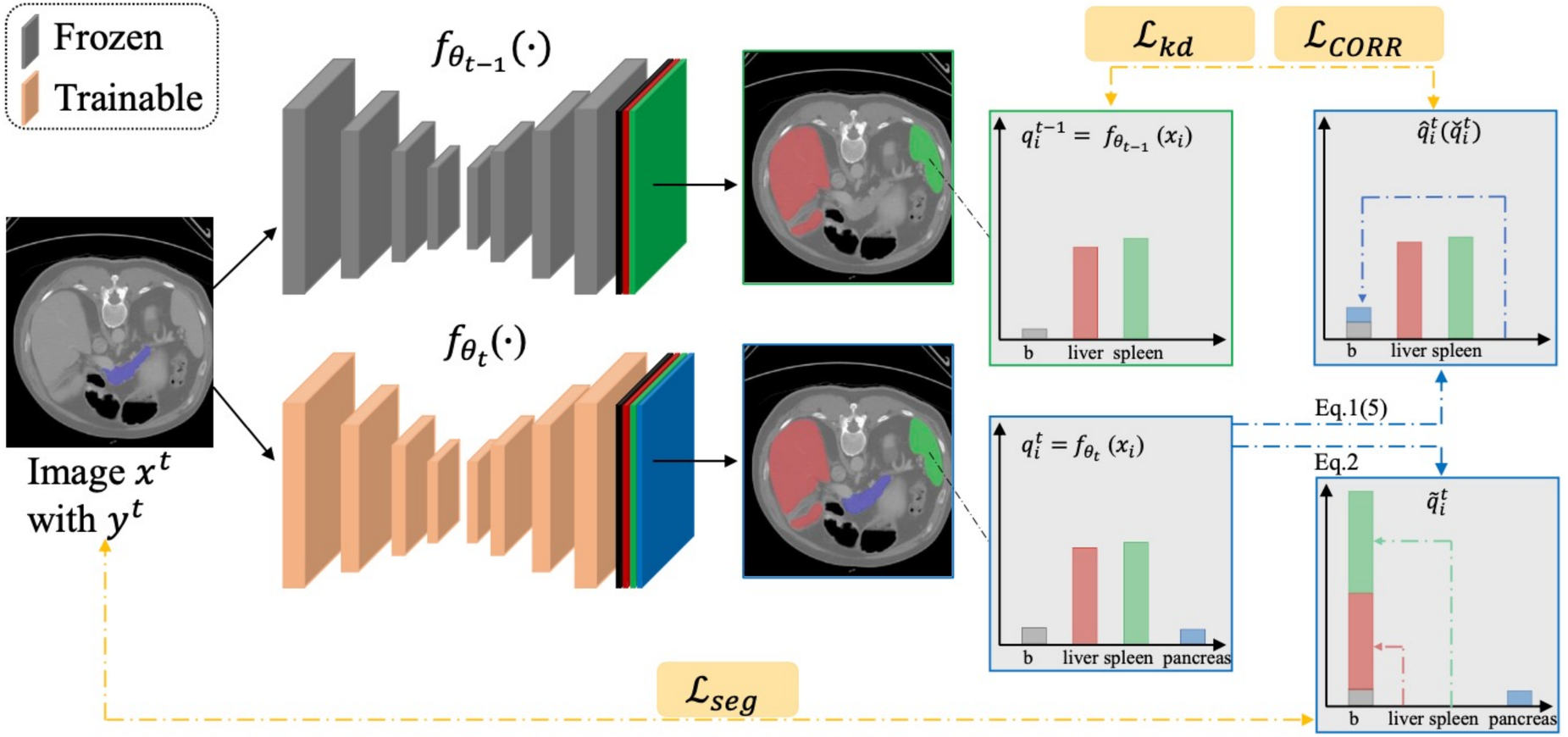
- In each in incremental learning stage:
  - lose access to the previous annotations, whose knowledge is assumingly captured by the current model
  - gain access to a new dataset with annotations of new organ categories, to update the organ segmentation model to include the new organs.

Example: Incremental learning for multi-label organ segmentation

Incremental Learning for Multi-organ Segmentation with Partially Labeled Datasets

Pengbo Liu[1], Li Xiao[1], and S. Kevin Zhou[1]

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{liupengbo2019,xiaoli}@ict.ac.cn
s.kevin.zhou@gmail.com



- Overview of t[th] stage of incremental learning for multi-organ segmentation

Example: Incremental learning for multi-label organ segmentation

Incremental Learning for Multi-organ Segmentation with Partially Labeled Datasets

Pengbo Liu[1], Li Xiao[1], and S. Kevin Zhou[1]

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{liupengbo2019,xiaoli}@ict.ac.cn
s.kevin.zhou@gmail.com

- Loss function is composed of:
  - **Segmentation loss** for learning new knowledge of new categories
  - **Knowledge distillation**, for preserving old knowledge distilled from the previous model

$$\mathcal{L} = \mathcal{L}_{seg}(\tilde{q}^t, y^t) + \mathcal{L}_{kd}(\hat{q}^t, \sigma(q^{t-1}))$$
$$= \mathcal{L}_{CE}(\tilde{q}^t, y^t) + \mathcal{L}_{Dice}(\tilde{q}^t, y^t) + \mathcal{L}_{kd}(\hat{q}^t, \sigma(q^{t-1}))$$

Where σ is the softmax operation.

  - For the segmentation loss, cross-entropy loss is the most commonly used, but can also or additionally use Dice loss

Example: Incremental learning for multi-label organ segmentation

Incremental Learning for Multi-organ Segmentation with Partially Labeled Datasets

Pengbo Liu[1], Li Xiao[1], and S. Kevin Zhou[1]

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{liupengbo2019,xiaoli}@ict.ac.cn
s.kevin.zhou@gmail.com

- Comparison to state-of-the-art trained either directly or incrementally:

Haussdorff distance

Methods\Organs	Liver∈F	Liver∈P_1	Spleen∈F	Spleen∈P_2	Pancreas∈F	Pancreas∈P_3	R Kidney∈F	R Kidney∈P_4	L Kidney∈F	L Kidney∈P_4	Mean
ϕ_{F+P_1} (Liver)	2.39 ± 0.66	10.81 ± 25.54	-	-	-	-	-	-	-	-	-
ϕ_{F+P_2} (Spleen)	-	-	1.58 ± 0.41	24.74 ± 62.02	-	-	-	-	-	-	-
ϕ_{F+P_3} (Pancreas)	-	-	-	-	23.23 ± 33.60	6.45 ± 10.02	-	-	-	-	-
ϕ_{F+P_4} (R/L Kidney)	-	-	-	-	-	-	26.49 ± 54.34	15.15 ± 43.06	30.13 ± 61.08	6.67 ± 16.59	14.76
ϕ_F (Five organs)	1.58 ± 0.41	12.12 ± 17.81	1.00 ± 0.00	1.35 ± 0.41	5.39 ± 3.82	9.41 ± 8.98	1.36 ± 0.40	6.19 ± 4.25	2.27 ± 1.87	11.67 ± 16.45	5.23
FT	nan	nan	nan	nan	nan	nan	4.85 ± 2.33	8.16 ± 31.29	3.97 ± 1.66	**3.01±.6.71**	-
LwF [11]	2.33 ± 0.48	11.19 ± 24.66	46.11 ± 96.71	30.31 ± 76.26	.4.89 ± 3.04	9.33 ± 13.54	16.03 ± 23.95	35.90 ± 57.56	25.68 ± 22.29	49.63 ± 54.46	23.14
ILT [16]	2.36 ± 0.53	11.13 ± 25.34	66.61 ± 102.64	30.59 ± 76.31	16.02 ± 19.58	10.37 ± 15.08	4.63 ± 2.21	29.34 ± 56.31	4.31 ± 1.21	21.80 ± 36.70	19.72
MiB [2]	2.56 ± 0.76	11.52 ± 25.03	**1.48±0.37**	**29.04±72.38**	**3.59±1.35**	6.76 ± 9.71	4.87 ± 2.47	**.8.09±30.75**	3.63 ± 1.35	10.29 ± 28.95	8.19
MiBOrgan(MiB+CORR)	**2.19±0.72**	**11.06±24.24**	1.96 ± 0.99	30.11 ± 75.21	3.97 ± 2.14	**6.13±6.04**	**4.44±2.24**	8.58 ± 33.12	**3.04±0.91**	5.21 ± 12.16	**7.45**
MargExc MIA [20]	2.84 ± 1.53	4.04 ± 2.64	17.58 ± 7.27	1.00 ± 0.09	3.24 ± 0.69	3.96 ± 3.27	1.43 ± 0.14	1.28 ± 0.07	3.13 ± 0.58	1.68 ± 0.68	4.02

Dice overlap

Methods\Organs	Liver∈F	Liver∈P_1	Spleen∈F	Spleen∈P_2	Pancreas∈F	Pancreas∈P_3	R Kidney∈F	R Kidney∈P_4	L Kidney∈F	L Kidney∈P_4	Mean
ϕ_{F+P_1} (Liver)	.958 ± .017	.964 ± .030	-	-	-	-	-	-	-	-	-
ϕ_{F+P_2} (Spleen)	-	-	.951 ± .010	.955 ± .028	-	-	-	-	-	-	-
ϕ_{F+P_3} (Pancreas)	-	-	-	-	.809 ± .053	.850 ± .071	-	-	-	-	-
ϕ_{F+P_4} (R/L Kidney)	-	-	-	-	-	-	.917 ± .038	.970 ± .027	.913 ± .031	.963 ± .042	.925
ϕ_F (Five organs)	.967 ± .010	.948 ± .027	.969 ± .007	.955 ± .005	.786 ± .091	.704 ± .149	.949 ± .016	.884 ± .093	.926 ± .057	.825 ± .172	.891
FT	.000 ± .000	.000 ± .000	.000 ± .000	.000 ± .000	.000 ± .000	.000 ± .000	.917 ± .016	**.978±.011**	.919 ± .015	**.973±.021**	.379
LwF [11]	.959 ± .017	.961 ± .032	.940 ± .021	.956 ± .024	.804 ± .044	.807 ± .102	.912 ± .016	.944 ± .043	.879 ± .044	.900 ± .104	.906
ILT [16]	.958 ± .017	.962 ± .029	.937 ± .020	.949 ± .035	.795 ± .046	.807 ± .096	.913 ± .022	.955 ± .039	.912 ± .017	.919 ± .103	.911
MiB [2]	**.961±.017**	.959 ± .037	**.953±.015**	**.953±.033**	**.817±.048**	**.819±.111**	**.918±.018**	.972 ± .035	.920 ± .016	.952 ± .073	**.922**
MiBOrgan(MiB+CORR)	**.961±.017**	**.960±.034**	.950 ± .016	.950 ± .035	.809 ± .049	.814 ± .111	.917 ± .018	.971 ± .028	**.921±.019**	.953 ± .077	.921
MargExc MIA [20]	.969 ± .012	.957 ± .009	.924 ± .009	.970 ± .008	.836 ± .006	.808 ± .041	.946 ± .012	.952 ± .013	.978 ± .013	.972 ± .004	.931

Overview

- Transfer learning (revisited)
- Few-shot learning
- Meta-learning
- Incremental learning
- **Curriculum learning**

Curriculum learning

- **Curriculum learning** is a concept of learning in which you first start out with only easy examples of a task and then gradually increase the task difficulty.
  – E.g. in classification task, some of the classes seem to be harder to learn than others.
- In contrast to standard **stochastic gradient descent** (SGD), where mini-batches are randomly sampled from training distribution:

Curriculum learning: scoring and pacing function

- **Scoring function**: used to rank the samples in the training set according to their difficulty

- **Pacing function**: used to determine how the samples are incremented during the training

Curriculum learning: scoring function

Scoring functions -> difficulty ranking

1. **Transfer learning**: use scores from another pre-trained network

2. **Bootstrapping**: use scores from pre-trained network of same architecture

Curriculum learning: pacing function

Pacing functions -> increment of harder samples

- **Step** -> learning iterations where pacing function “*g*” is not changed
- **Step length** -> * number of each iterations in step
- **Increase** -> e.g. exponential factor increase in data size used in sampling mini-batches
- **Starting percent** -> % of data used in initial step

Curriculum learning: pacing function

Pacing functions -> increment of harder samples

- **exponential pacing**:
  - **Fixed**: present a small % of data and exponentially increase amount every fixed it number
  - **Varied:** allow variance of iterations from step to step

- **single-step pacing**:
  - samples are initially sampled from easy data and then from the whole data set as usual

- **baby-step pacing**:
  - keep previously introduced training samples in pool of training examples rather than replacing them with new ones.

Curriculum learning algorithm

Algorithm 1 Curriculum learning method

Input: *pacing function* g_ϑ, *scoring function* f, data $\mathbb{X}$.

Output: sequence of mini-batches $\left[\mathbb{B}'_1, ..., \mathbb{B}'_M\right]$.

sort $\mathbb{X}$ according to f, in ascending order

$result \leftarrow []$

for all $i = 1, ..., M$ **do**

  $size \leftarrow g_\vartheta(i)$

  $\mathbb{X}'_i \leftarrow \mathbb{X}[1, ..., size]$

  uniformly sample $\mathbb{B}'_i$ from $\mathbb{X}'$

  append $\mathbb{B}'_i$ to $result$

end for

return $result$

Curriculum and anti-curriculum learning

- **Curriculum learning**: ordered in ascending difficulty
- **Anti-curriculum learning**: ordered in descending difficulty
- **Random**: similar to SGD (if batches are very small)

Example:

Medical Image Analysis 55 (2019) 136–147

Contents lists available at ScienceDirect

Medical Image Analysis

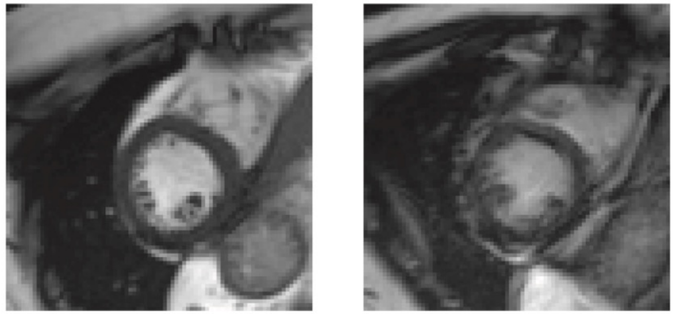
journal homepage: www.elsevier.com/locate/media

Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning

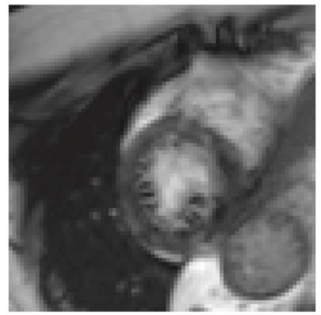
Ilkay Oksuz[a,*], Bram Ruijsink[a,b], Esther Puyol-Antón[a], James R. Clough[a], Gastao Cruz[a], Aurelien Bustin[a], Claudia Prieto[a], Rene Botnar[a], Daniel Rueckert[c], Julia A. Schnabel[a], Andrew P. King[a]

[a] School of Biomedical Engineering & Imaging Sciences, King's College, London, UK
[b] Guy's and St Thomas' Hospital NHS Foundation Trust, London, UK
[c] Biomedical Image Analysis Group, Imperial College, London, UK

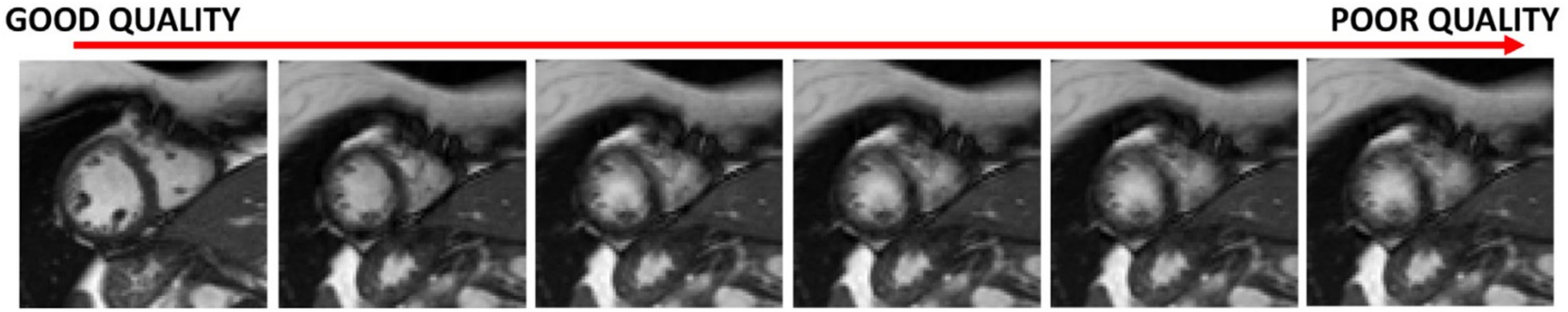
- Motion artefact detection in cardiac MRI
  - Training data: different quality images using synthetic degredation of k-space (raw MRI)
  - Curriculum learning to train from most easy to most difficult samples



(a) Good quality image (b) Motion artefact image



(c) Synthetic image



Example:

Medical Image Analysis 55 (2019) 136–147

Contents lists available at ScienceDirect

Medical Image Analysis

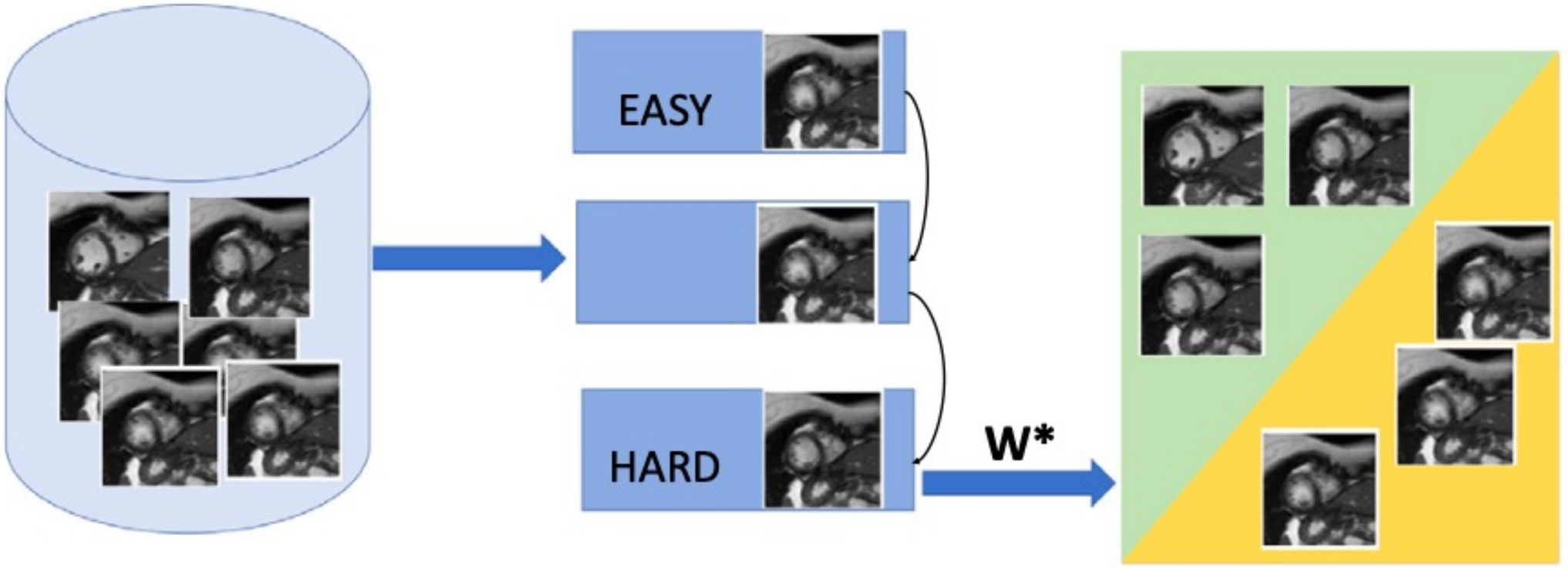
journal homepage: www.elsevier.com/locate/media

Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning

Ilkay Oksuz[a,*], Bram Ruijsink[a,b], Esther Puyol-Antón[a], James R. Clough[a], Gastao Cruz[a], Aurelien Bustin[a], Claudia Prieto[a], Rene Botnar[a], Daniel Rueckert[c], Julia A. Schnabel[a], Andrew P. King[a]

[a] School of Biomedical Engineering & Imaging Sciences, King's College, London, UK
[b] Guy's and St Thomas' Hospital NHS Foundation Trust, London, UK
[c] Biomedical Image Analysis Group, Imperial College, London, UK

- Curriculum learning:



Example:

Medical Image Analysis 55 (2019) 136–147

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning

Ilkay Oksuz[a,*], Bram Ruijsink[a,b], Esther Puyol-Antón[a], James R. Clough[a], Gastao Cruz[a], Aurelien Bustin[a], Claudia Prieto[a], Rene Botnar[a], Daniel Rueckert[c], Julia A. Schnabel[a], Andrew P. King[a]

[a] School of Biomedical Engineering & Imaging Sciences, King's College, London, UK
[b] Guy's and St Thomas' Hospital NHS Foundation Trust, London, UK

Algorithm 1 Proposed curriculum learning strategy for motion artefact detection.

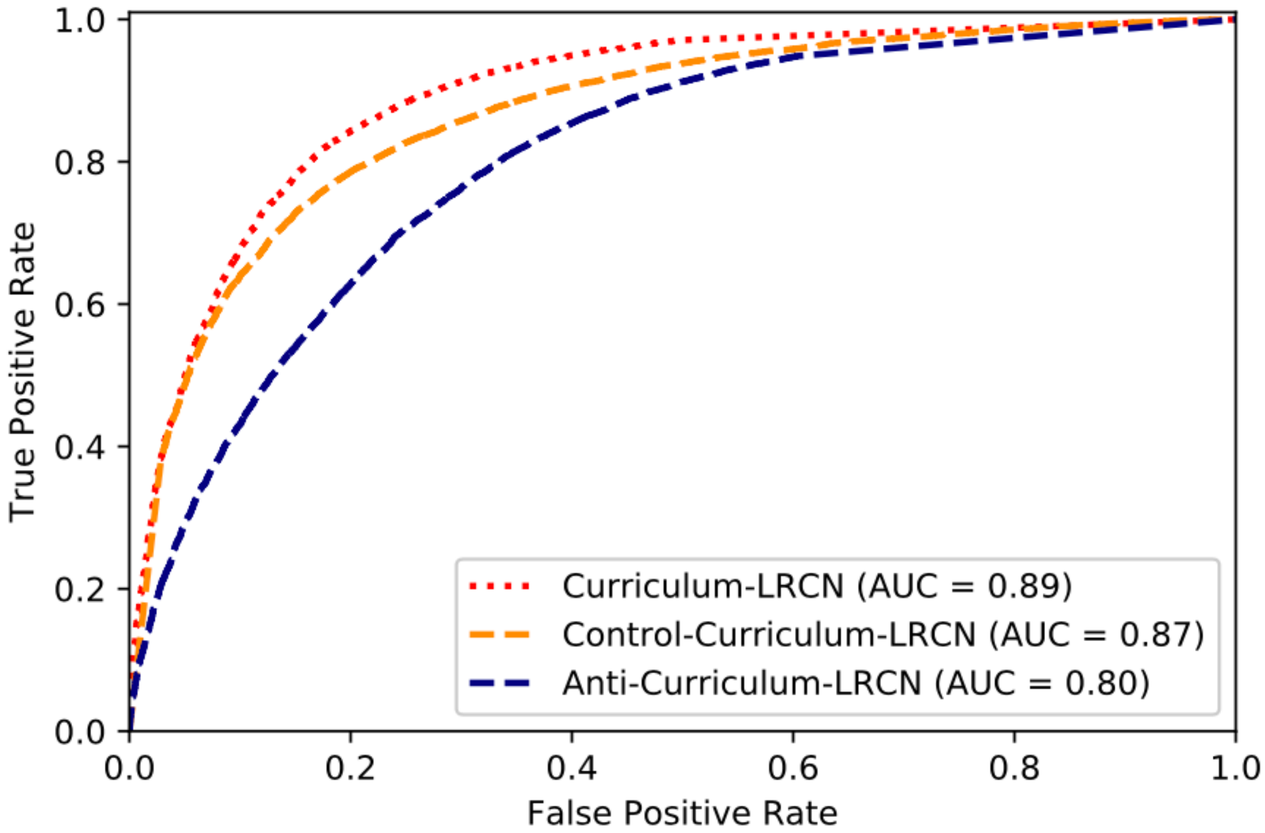
INPUT: Data set of synthetically generated image sequences $D = \{D^i\}_{i=1}^{b}$ ordered by a pre-defined curriculum

OUTPUT: Optimized model parameters W^*

1: D^{train} = Original Data set of Image Sequences
2: **for** i={1,...,b} **do**
3: $D^{\text{train}} = D^{\text{train}} \bigcup D^i$
4: **for** epoch={1,...,k} **do**
5: train (W, D^{train})
6: **end for**
7: select best W^*
8: **end for**

Here we use baby-step pacing

Example:



Medical Image Analysis 55 (2019) 136–147

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning

Ilkay Oksuz[a,*], Bram Ruijsink[a,b], Esther Puyol-Antón[a], James R. Clough[a], Gastao Cruz[a], Aurelien Bustin[a], Claudia Prieto[a], Rene Botnar[a], Daniel Rueckert[c], Julia A. Schnabel[a], Andrew P. King[a]

[a] School of Biomedical Engineering & Imaging Sciences, King's College, London, UK
[b] Guy's and St Thomas' Hospital NHS Foundation Trust, London, UK
[c] Biomedical Image Analysis Group, Imperial College, London, UK

- ROC analysis shows improvement over anti-curriculum and control-curriculum approaches

Example:

Medical Image Analysis 55 (2019) 136–147

Contents lists available at ScienceDirect

Medical Image Analysis

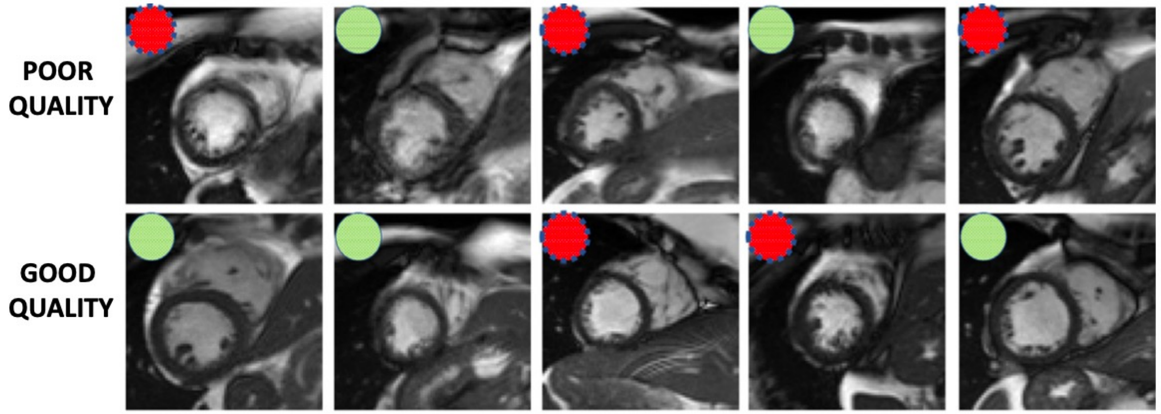
journal homepage: www.elsevier.com/locate/media

Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning

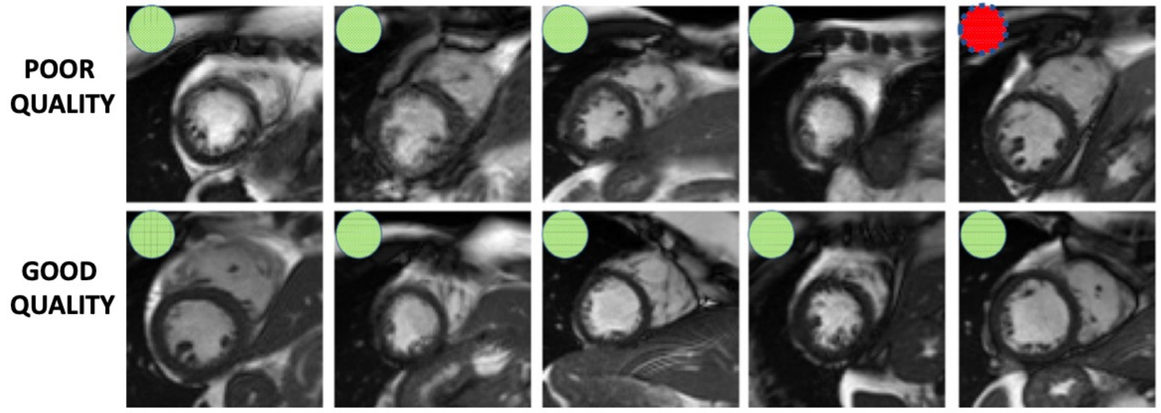
Ilkay Oksuz[a,*], Bram Ruijsink[a,b], Esther Puyol-Antón[a], James R. Clough[a], Gastao Cruz[a], Aurelien Bustin[a], Claudia Prieto[a], Rene Botnar[a], Daniel Rueckert[c], Julia A. Schnabel[a], Andrew P. King[a]

[a] School of Biomedical Engineering & Imaging Sciences, King's College, London, UK
[b] Guy's and St Thomas' Hospital NHS Foundation Trust, London, UK
[c] Biomedical Image Analysis Group, Imperial College, London, UK

- Visual classification results for difficult cases:



(a) LRCN-Control Curriculum



(b) LRCN Curriculum

Curriculum with random difficulty (ie SGD)

Curriculum with ascending difficulty

Final note on curriculum learning

- Uses full data set, but presents it to network as sample batches in ascending order of difficulty
  - Some similarity to **incremental learning,** especially when using **baby-step pacing**
  - Often also referred to as "**teacher-student**" model
    - but different from knowledge distillation

- Disadvantage:
  - Could introduce class imbalance bias

Summary

- In medical (imaging) applications, we suffer from the lack of large, high-quality annotated databases, and need to learn from sparse annotations. To adress this, we can use:
- **Transfer learning:**
  – transfer of knowledge from larger annotated database to new problem, but *carries risk of forgetting and overfitting*
- **Few-shot learning**
  – uses support set to learn how to make *prediction based on a limited number of samples*
- **Meta-learning**
  – *learns a learning strategy* to adjust well to a new few-shot learning task
- **Incremental learning**
  – learns new tasks or distributions without forgetting old ones
- **Curriculum learning**
  – is training on *samples of increasing difficulty*

Evolution of machine learning

- **Classic ML:**
  - One dataset, one task, one heavy training
- **Few-shot ML:**
  - Heavy offline training, then easy learning on similar tasks
- **Developing ML:**
  - Continuous, life-long learning on various tasks

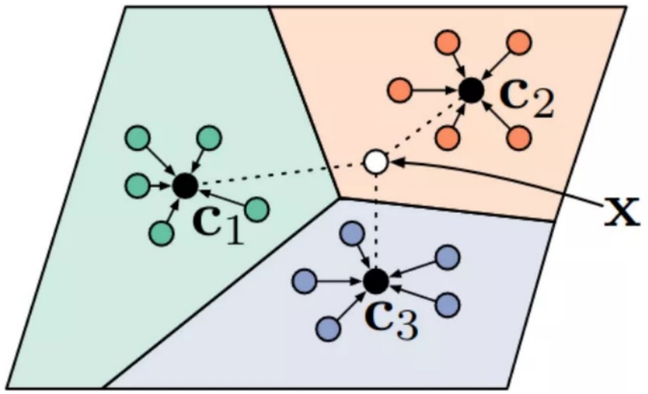
Some literature

- CVPR 2020 Tutorial: Towards Annotation-Efficient Learning https://annotation-efficient-learning.github.io/
- Few shot learning - State of the Art http://www.research.ibm.com/haifa/dept/imt/ist_dm.shtml
- Self-Supervision with Superpixels: Training Few-shot Medical Image Segmentation without Annotation https://arxiv.org/abs/2007.09886
- A continual learning survey: Defying forgetting in classification tasks. https://arxiv.org/abs/1909.08383
- Learning without forgetting. https://arxiv.org/abs/1606.09282
- Distilling the Knowledge in a Neural Network. https://arxiv.org/abs/1503.02531
- Incremental Learning for Multi-organ Segmentation with Partially Labeled Datasets https://arxiv.org/abs/2103.04526
- Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning https://doi.org/10.1016/j.media.2019.04.009

Self-Study Tutorial: Meta Learning – Metric Learning

Prototype Networks

- Few-shot learning
- Very similar to clustering
- Find centroid objects for each class
- Predictions are made by measuring the inverse distance to each centroid



Tutorial for classification using MedMNIST

- MedMNIST: https://medmnist.com
- Tutorial: https://colab.research.google.com/drive/1muaDJuR2f8c9sPHINsxmeNnR__hz_r9i?usp=sharing
- Author: Ivan Ezhov ivan.ezhov@tum.de

Snell, et al. "Prototypical networks for few-shot learning." NeurIPS 2017.

(At least for now!)