Lyon, Avril 17– 21

DLMI2023

CREATIS

UNIVERSITÉ DE LYON

ÉTS
Le génie pour l'industrie

Few-shot learning and uncertainty estimation

Jose Dolz

ÉTS, Montreal

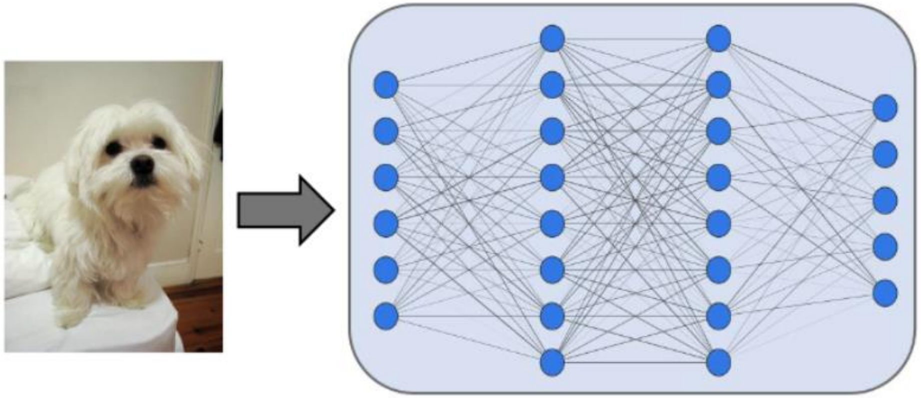
Model Adaptation

(few-shot learning)

Why?

Motivation

Standard training



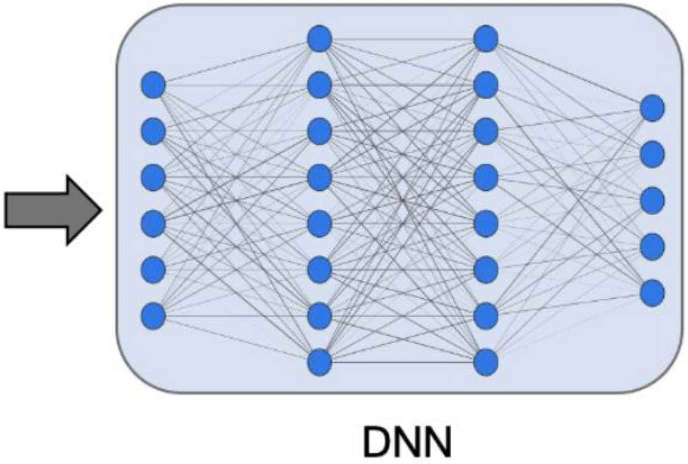
DNN

Accuracy (Dog): 99%

Why?

Motivation

Standard training



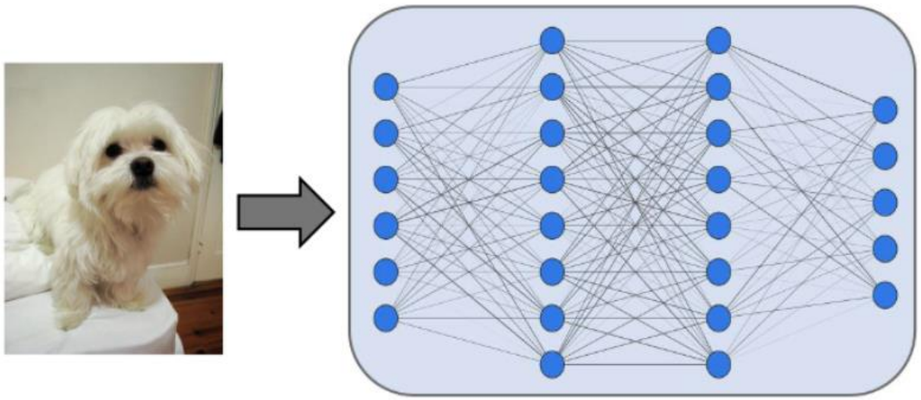
Accuracy (Dog): 99%

What if we add novel classes?

Why?

Motivation

Standard training

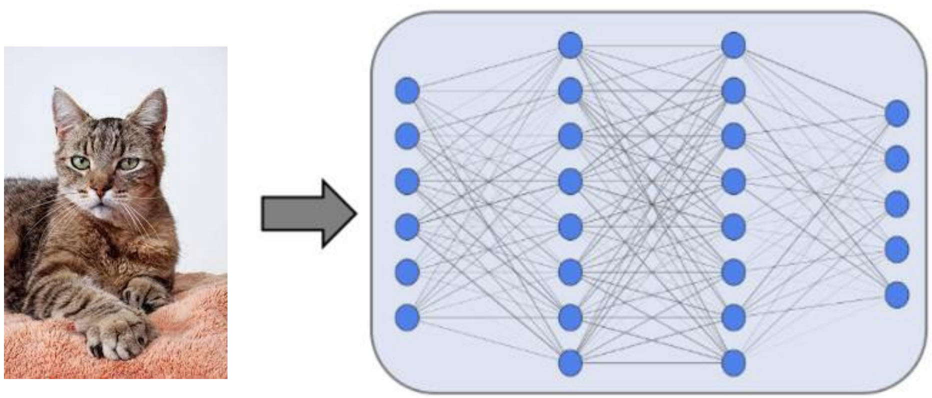


DNN

Accuracy (Dog): 99%

What if we add novel classes?

Fine-tunning



DNN

But, how many images are needed?

Why?

Motivation

Learning (in human beings)

One image
(learning)

N images
(generalization)

Why?

Motivation

Learning (in human beings)

image
rning)

-- **Humans recognize easily** with few examples

-- Modern ML generalize very poorly

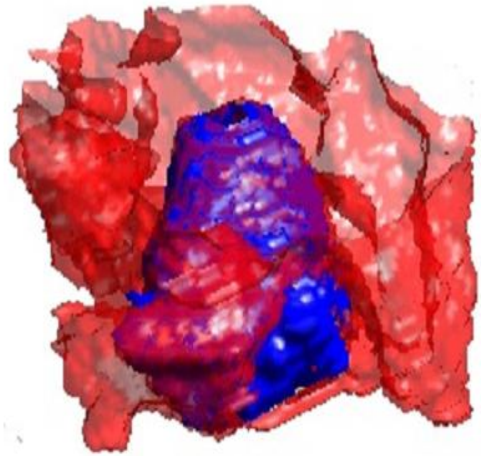
Why is this interesting?

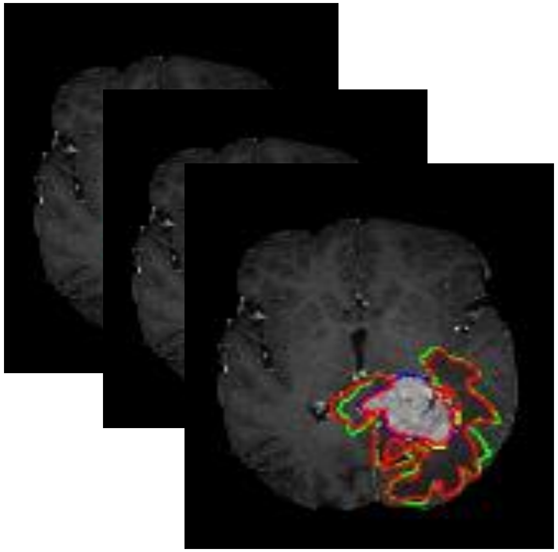
Building labels for dense predictions is even worse!!

Why is this interesting?

And it gets even more complex in some domains (e.g., medical imaging)

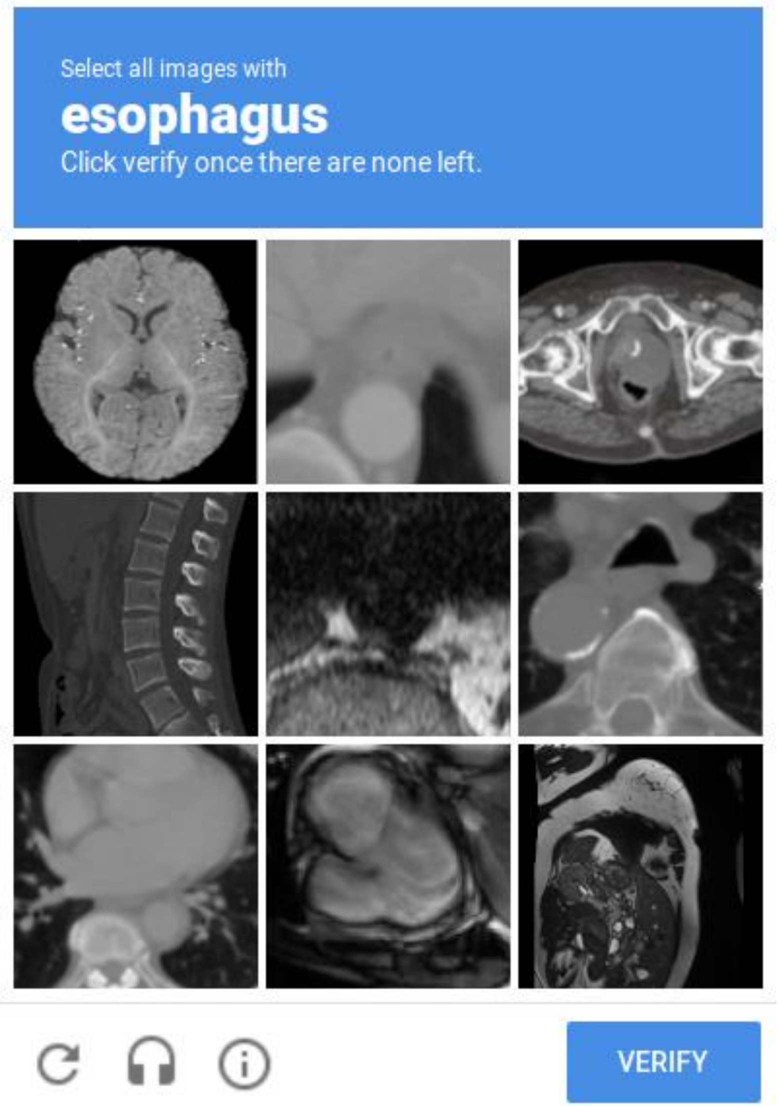
Dense 3D annotations: several hours
(of radiologist time)





Why is this interesting?

Labels not only expensive, but expert knowledge is required?

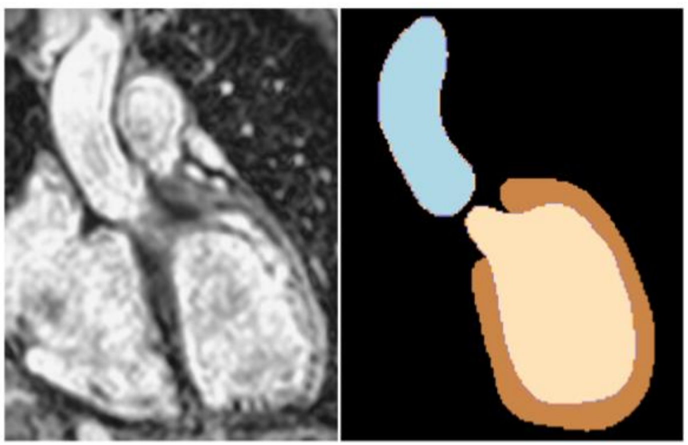


Not possible to do crowdsourcing

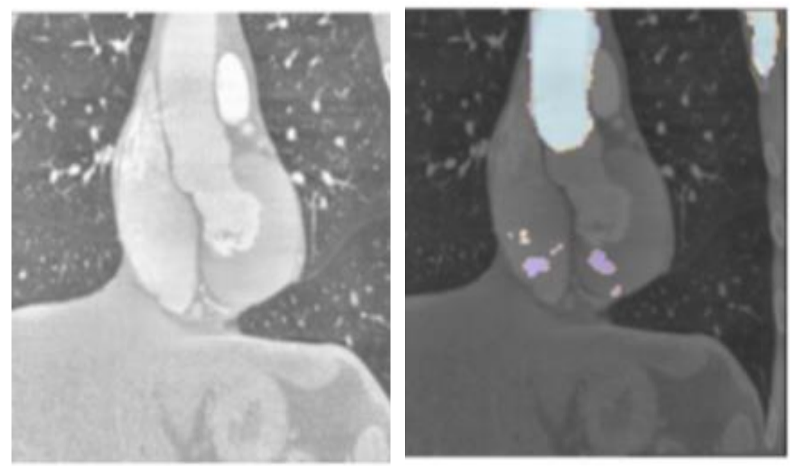
Why is this interesting?

Distributional shifts make things even worse

Source domain (MRI)



No adaptation
(bad generalization to the target)



Few-shot learning

Setting

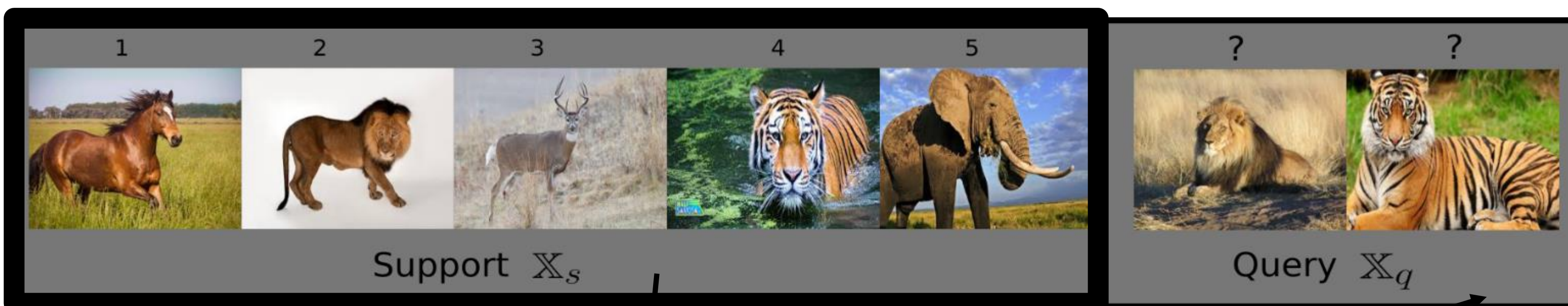
Training on **base** classes

Few-shot learning

Setting

Training on **base** classes

Few-shot tasks at
testing time



Learn from a few examples per **new** class

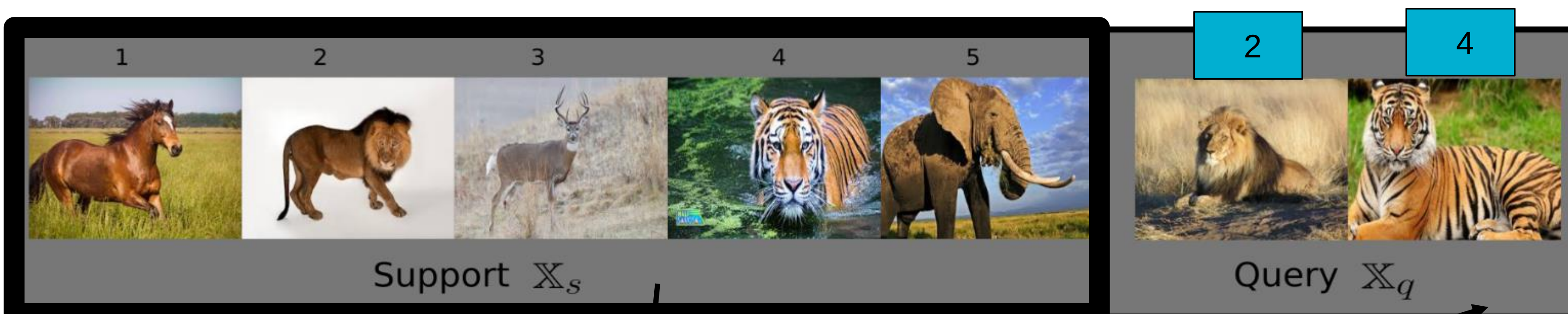
Classify
these

Few-shot learning

Setting

Training on **base** classes

Few-shot tasks at testing time



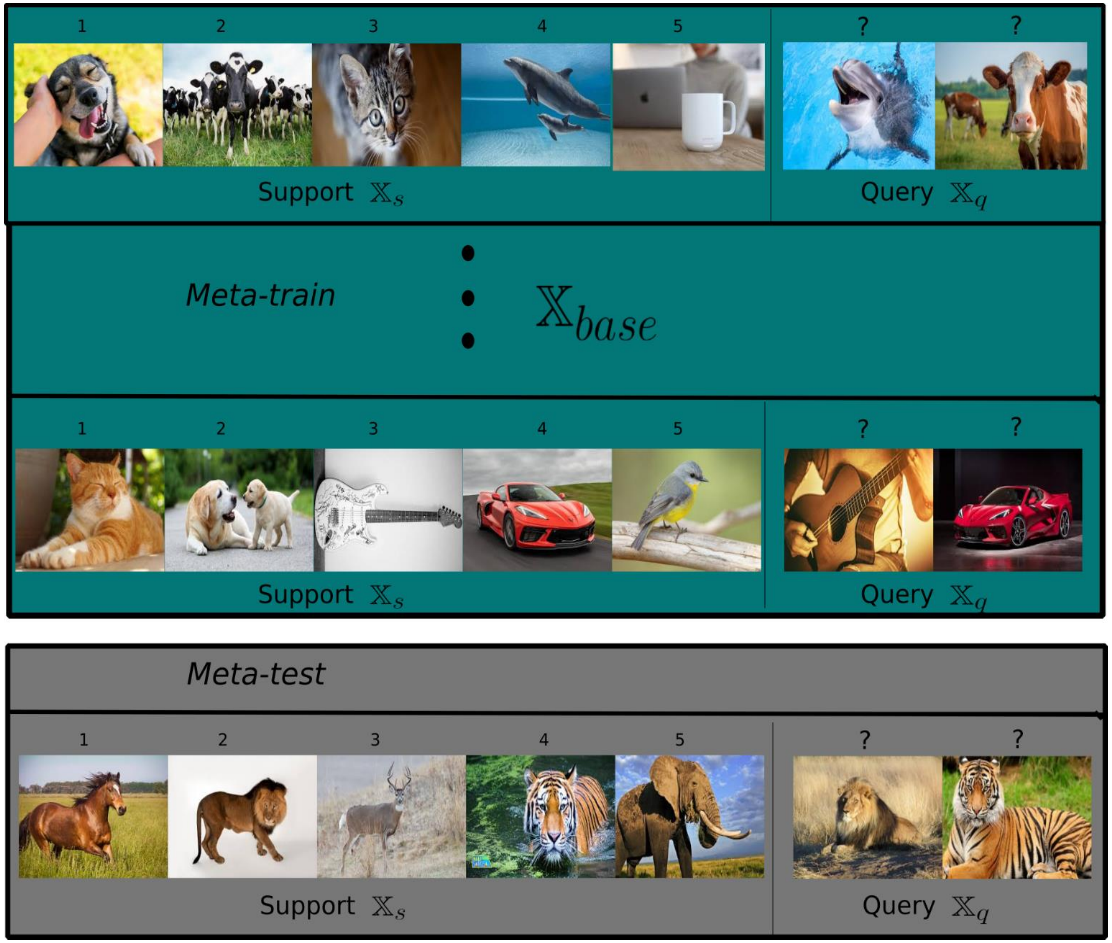
Learn from a few examples per **new** class

Classify these

Few-shot learning

Setting

Meta-Learning



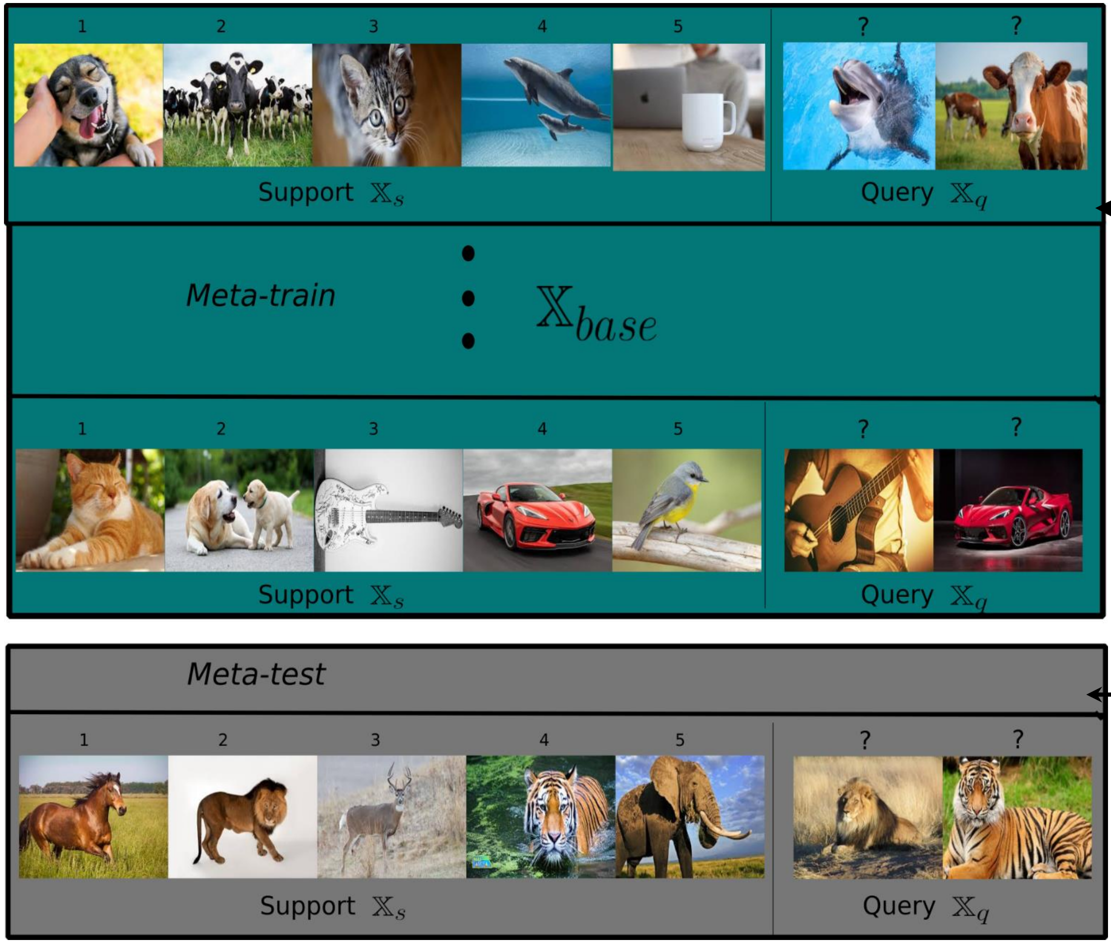
Base training with enough labeled data

(base classes ***different from the*** test classes)

Few-shot learning

Setting

Meta-Learning



Create artificial episodes for ***episodic training***
(Learn initial model)

Vinyal et al, (Neurips '16),
Snell et al, (Neurips '17),
Sung et al, (CVPR ' 18),
Finn et al, (ICML' 17),
Ravi et al, (ICLR' 17),
Lee et al, (CVPR' 19),
Hu et al, (ICLR '20),
Ye et al, (CVPR '20), . . .

Few-shot learning

Standard Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\}$$

$$\Rightarrow$$

$$\mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset$$
$$\mathcal{Y}_{Train} = \mathcal{Y}_{Test}$$

Few-shot learning

Standard Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\} \Rightarrow \begin{array}{l} \mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset \\ \mathcal{Y}_{Train} = \mathcal{Y}_{Test} \end{array}$$

Learning

$$\mathcal{D}_{Train} \quad f_{\theta*} \Rightarrow \mathcal{L}_{Train}$$

Few-shot learning

Standard Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\} \Rightarrow \begin{array}{l} \mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset \\ \mathcal{Y}_{Train} = \mathcal{Y}_{Test} \end{array}$$

Learning

$$\mathcal{D}_{Train} \quad f_{\theta*} \Rightarrow \mathcal{L}_{Train}$$

Inference

$$\mathcal{D}_{Test} \quad f_{\theta*} \Rightarrow$$ **Predictions**

Few-shot learning

Meta-Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\} \Rightarrow$$

$$\mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset$$

$$\mathcal{Y}_{Train} \cap \mathcal{Y}_{Test} = \emptyset$$

Few-shot learning

Meta-Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\} \Rightarrow$$

$$\mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset$$

$$\mathcal{Y}_{Train} \cap \mathcal{Y}_{Test} = \emptyset$$

Training

1. From $\mathcal{D}_{Train}$ sample $\mathcal{Y}_S \in \mathcal{Y}_{Train}$

$$\mathcal{Y}_{Train} = \{Horse, Bike, Dog, Cat, Lion\}$$

$$\mathcal{Y}_S = \{Dog, Cat\}$$

Few-shot learning

Meta-Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\} \Rightarrow \begin{array}{l} \mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset \\ \mathcal{Y}_{Train} \cap \mathcal{Y}_{Test} = \emptyset \end{array}$$

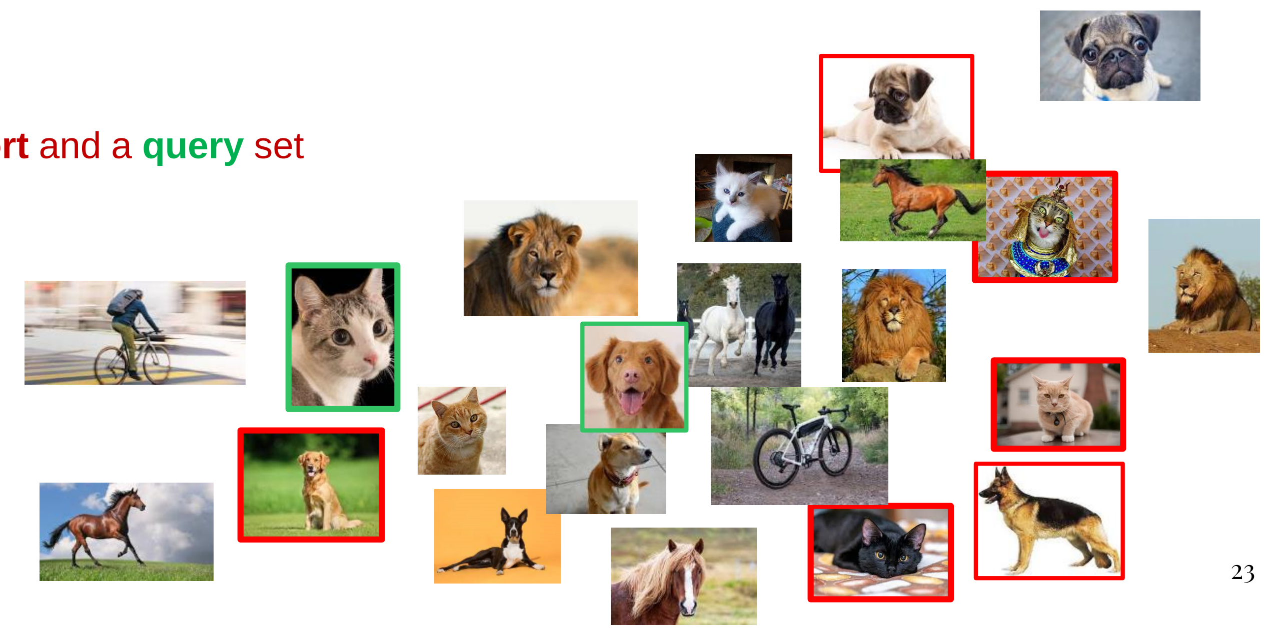
Training

1. From $\mathcal{D}_{Train}$ sample $\mathcal{Y}_S \in \mathcal{Y}_{Train}$

$$\mathcal{Y}_{Train} = \{Horse, Bike, Dog, Cat, Lion\}$$

$$\mathcal{Y}_S = \{Dog, Cat\}$$

2. Use $\mathcal{Y}_S$ to sample a **support** and a **query** set



Few-shot learning

Meta-Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\}$$

$$\mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset$$

$$\mathcal{Y}_{Train} \cap \mathcal{Y}_{Test} = \emptyset$$

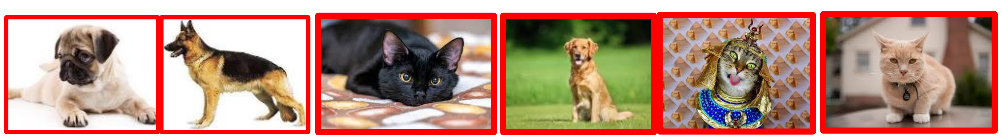
Training

1. From $\mathcal{D}_{Train}$ sample $\mathcal{Y}_S \in \mathcal{Y}_{Train}$

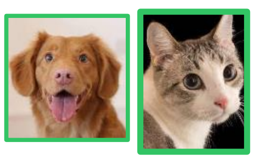
$$\mathcal{Y}_{Train} = \{Horse, Bike, Dog, Cat, Lion\}$$

$$\mathcal{Y}_S = \{Dog, Cat\}$$

2. Use $\mathcal{Y}_S$ to sample a **support** and a **query** set



Support set $\mathcal{S}$



Query set $\mathcal{Q}$

$$\boxed{\min \mathcal{L}(\hat{y}_{\mathcal{Q}})}$$

Few-shot learning

Meta-Learning

$$\mathcal{D} = \{\mathcal{D}_{Train}, \mathcal{D}_{Test}\}$$

$$\mathcal{D}_{Train} \cap \mathcal{D}_{Test} = \emptyset$$

$$\mathcal{Y}_{Train} \cap \mathcal{Y}_{Test} = \emptyset$$

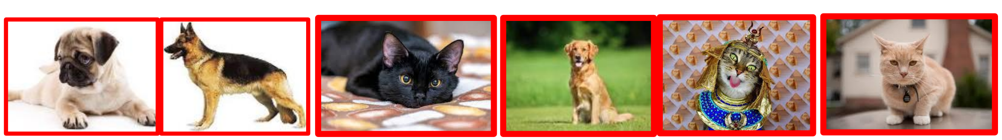
Training

1. From $\mathcal{D}_{Train}$ sample $\mathcal{Y}_S \in \mathcal{Y}_{Train}$

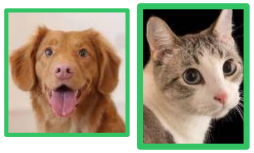
$$\mathcal{Y}_{Train} = \{Horse, Bike, Dog, Cat, Lion\}$$

$$\mathcal{Y}_S = \{Dog, Cat\}$$

2. Use $\mathcal{Y}_S$ to sample a **support** and a **query** set



Support set $\mathcal{S}$



Query set $\mathcal{Q}$

3. Repeat

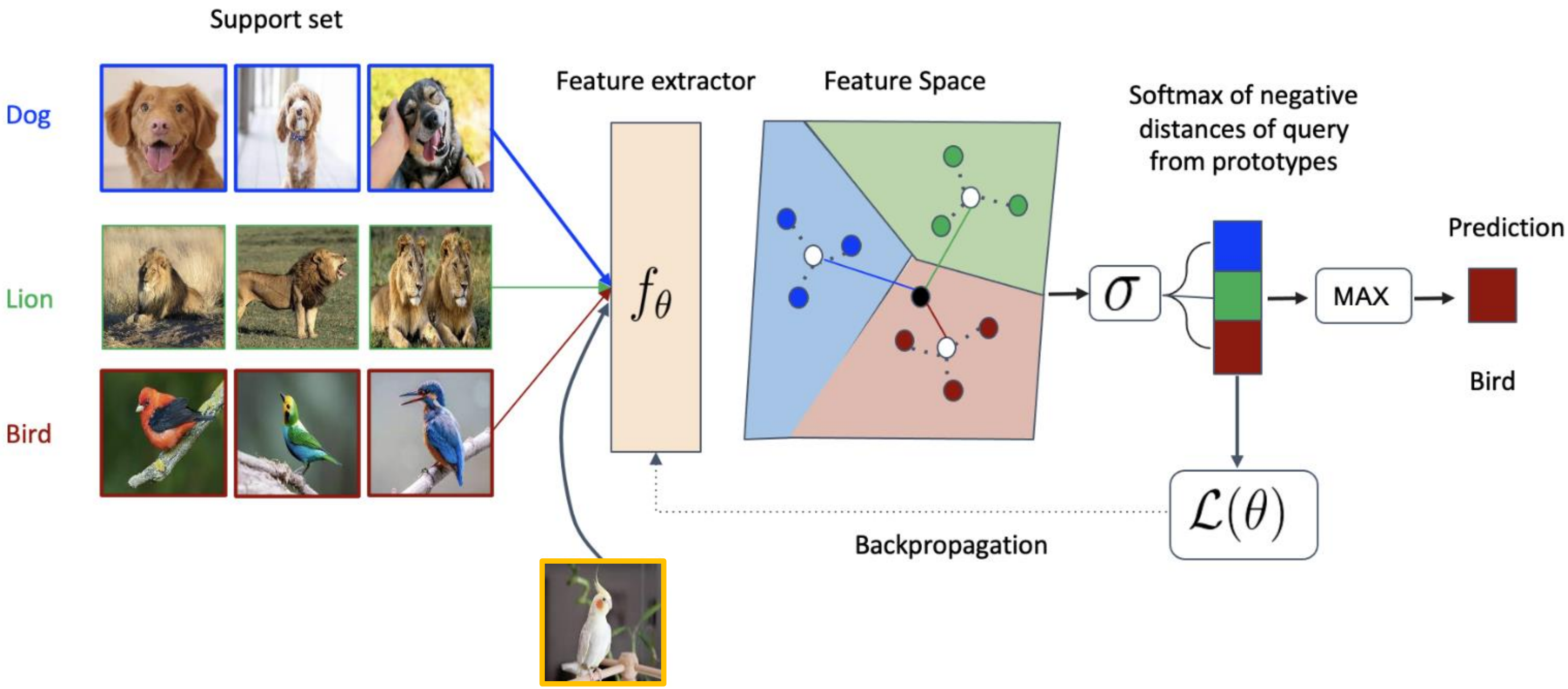
$$\boxed{\min \mathcal{L}(\hat{y}_{\mathcal{Q}})}$$

Few-shot learning

Classification

Prototypical Networks

[Snell et al., NeurIPS'17]

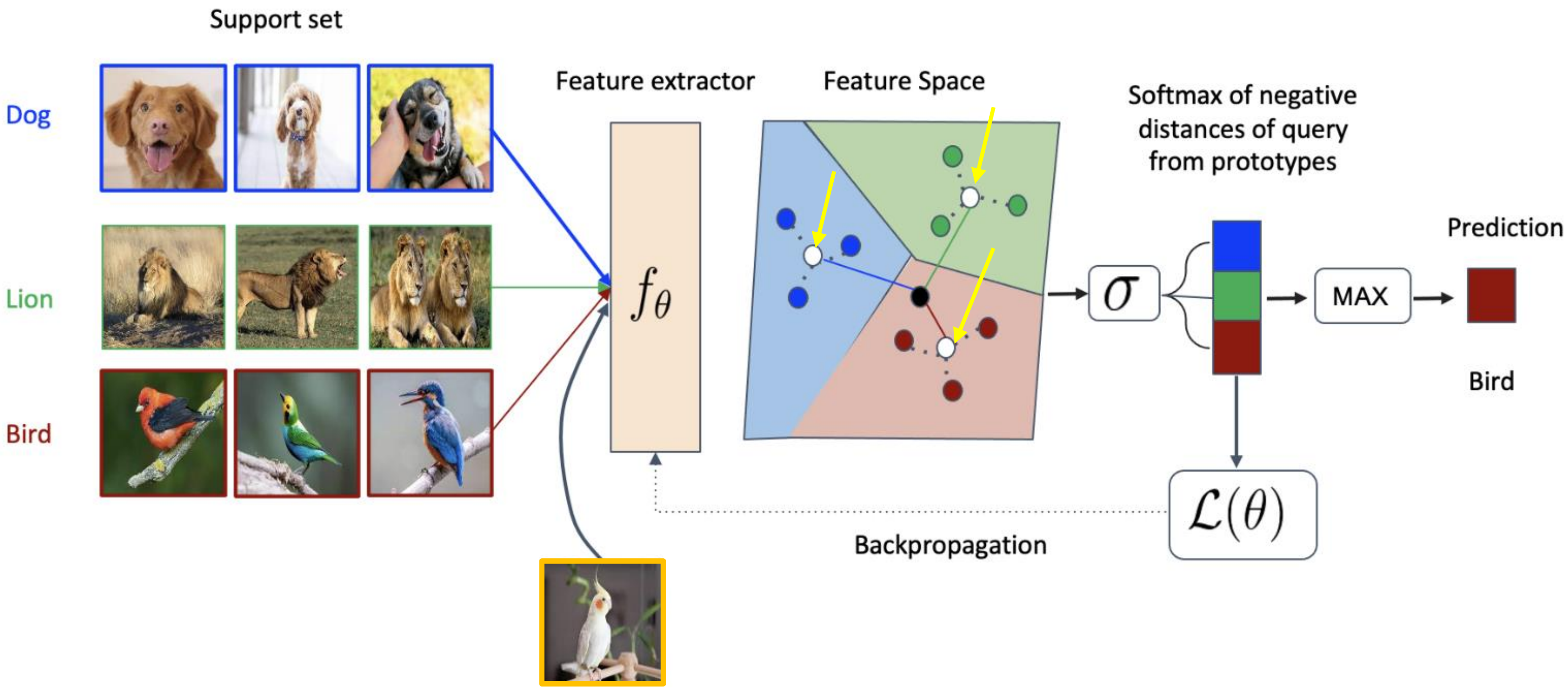


Few-shot learning

Classification

Prototypical Networks

[Snell et al., NeurIPS'17]



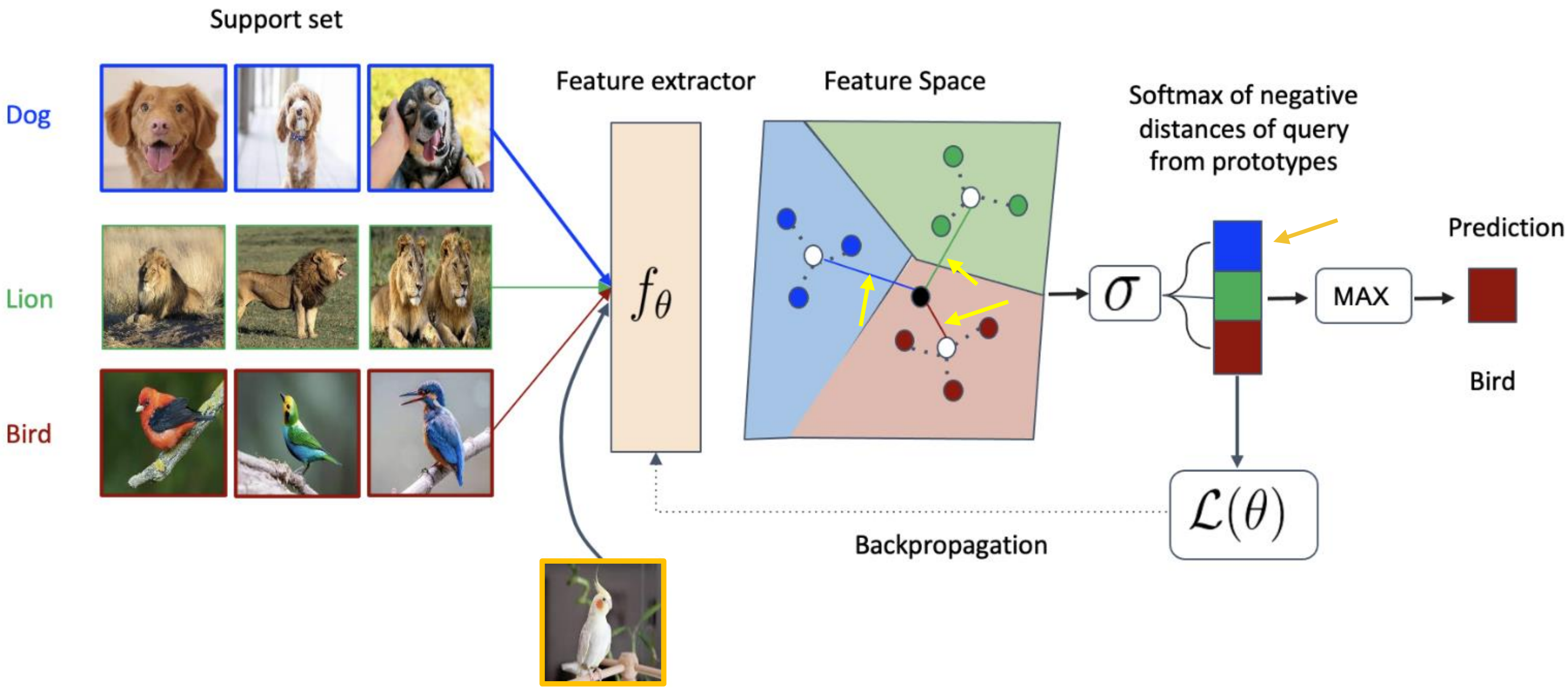
Compute **class prototypes** from support set: $\mu_c = \frac{1}{|\mathbb{X}_s^c|} \sum_{(x_k, y_k) \in \mathbb{X}_s^c} f_\theta(x_k)$

Few-shot learning

Classification

Prototypical Networks

[Snell et al., NeurIPS'17]



Output probability based on **similarity of query embedding to each class prototypes**:

$$P(y=c|\hat{x}) = softmax(-d(f_\theta(\hat{x}), \mu_c))$$

Few-shot learning

Classification

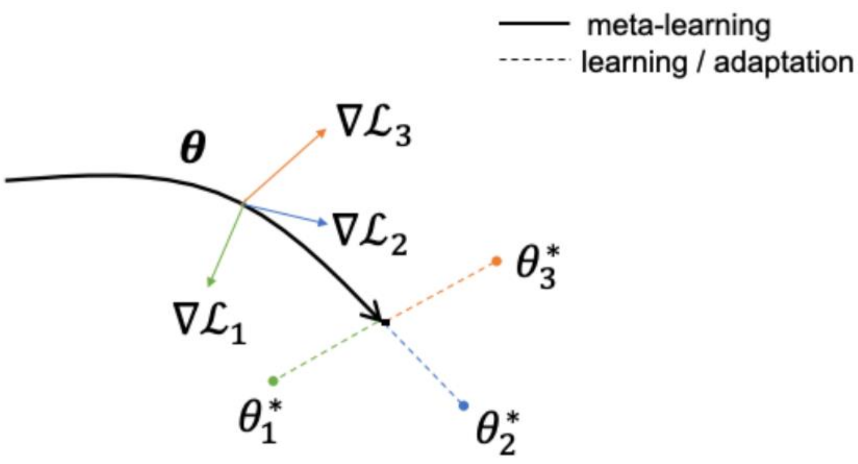
MAML

[Finn et al., ICML'17]

Goal:

Learn a **good initialization for a model**, such that :

it can be adapted to new few-shot tasks with **few gradient steps** to perform well with few training steps



Few-shot learning

Classification

MAML

[Finn et al., ICML'17]

Outer Loop

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

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

Inner Loop:

Few-shot learning

Classification

MAML

[Finn et al., ICML'17]

Outer Loop: Optimize for the performance of **all inner loop** models on all tasks.

Algorithm 1 Model-Agnostic Meta-Learning

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

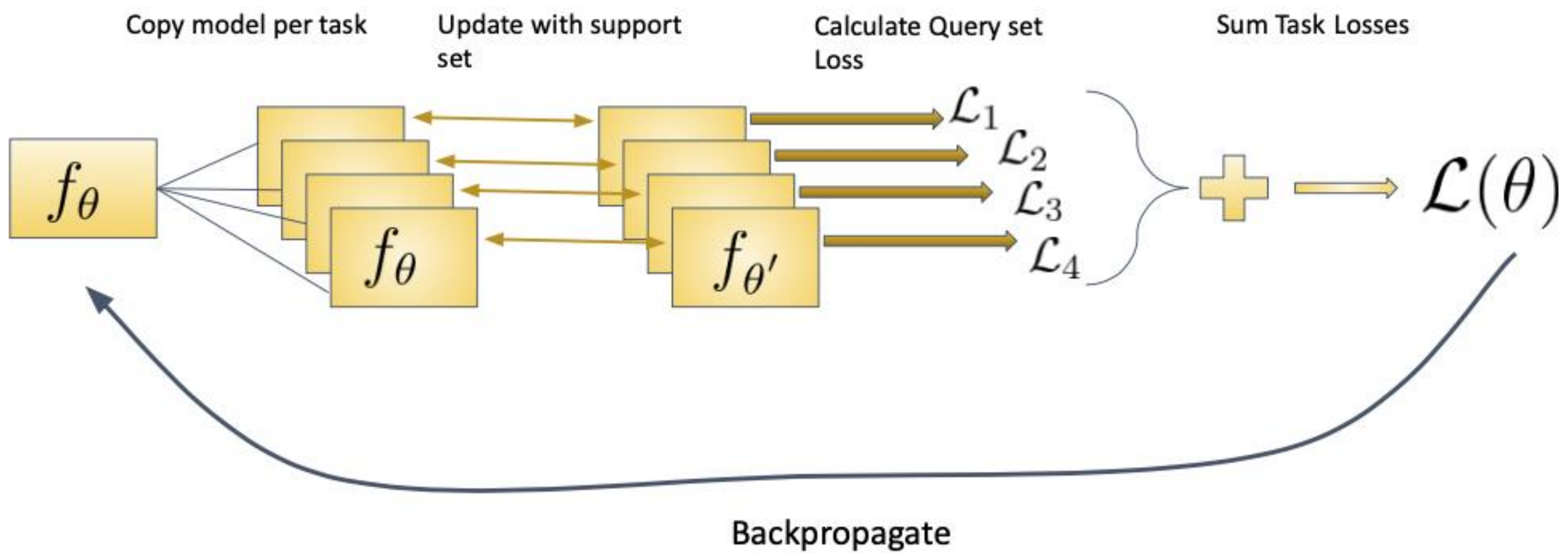
Inner Loop: Update for each task from an initialization

Few-shot learning

Classification

MAML

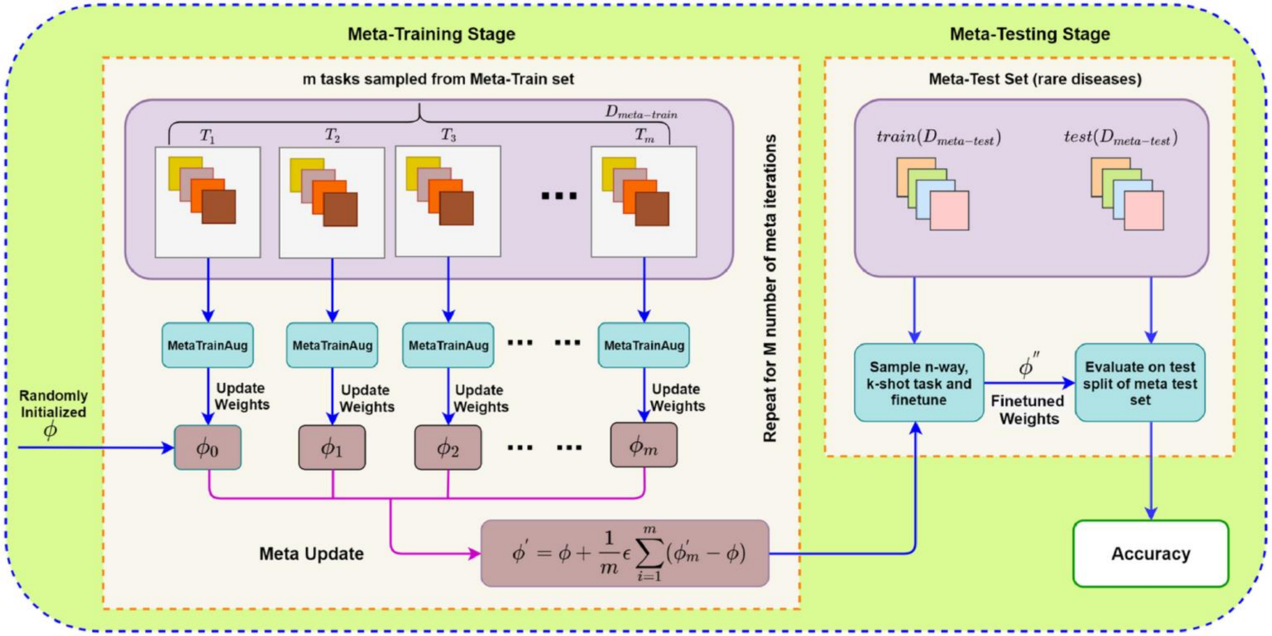
[Finn et al., ICML'17]

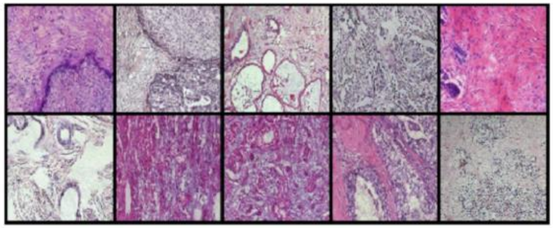


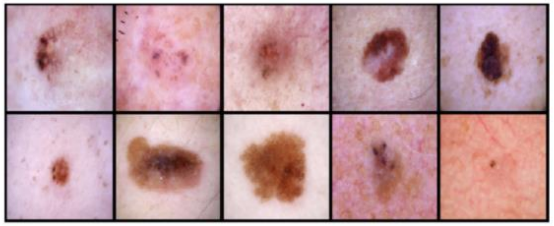
Few-shot learning

Classification

MAML-Based





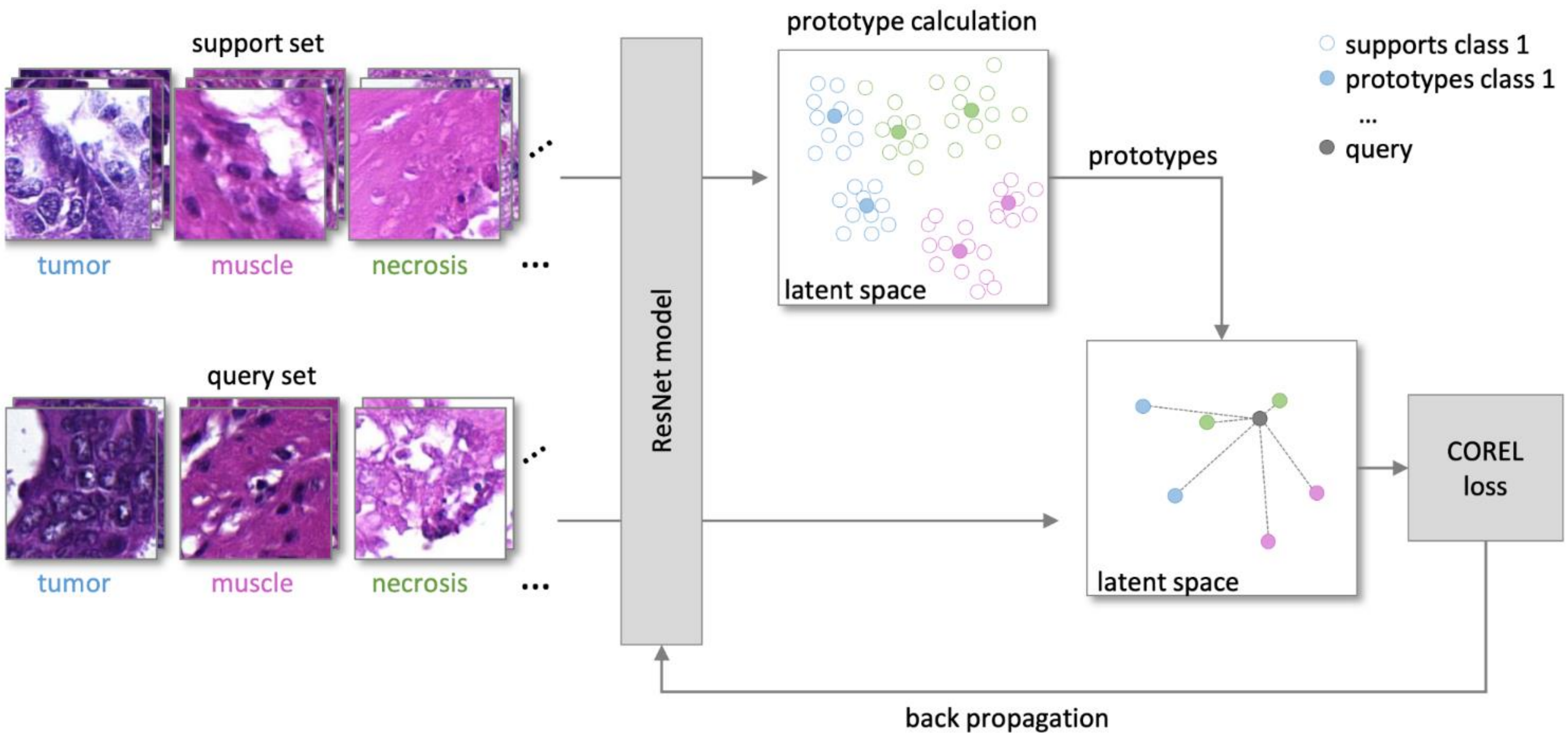


Singh et al. MetaMed: Few-shot medical image classification using gradient-based meta-learning. Pattern Recognition. 2021

Few-shot learning

Classification

Prototypes-Based



Deuschel et al. Multi-prototype few-shot learning in histopathology. ICCVw'21

Few-shot Segmentation

How does it work?

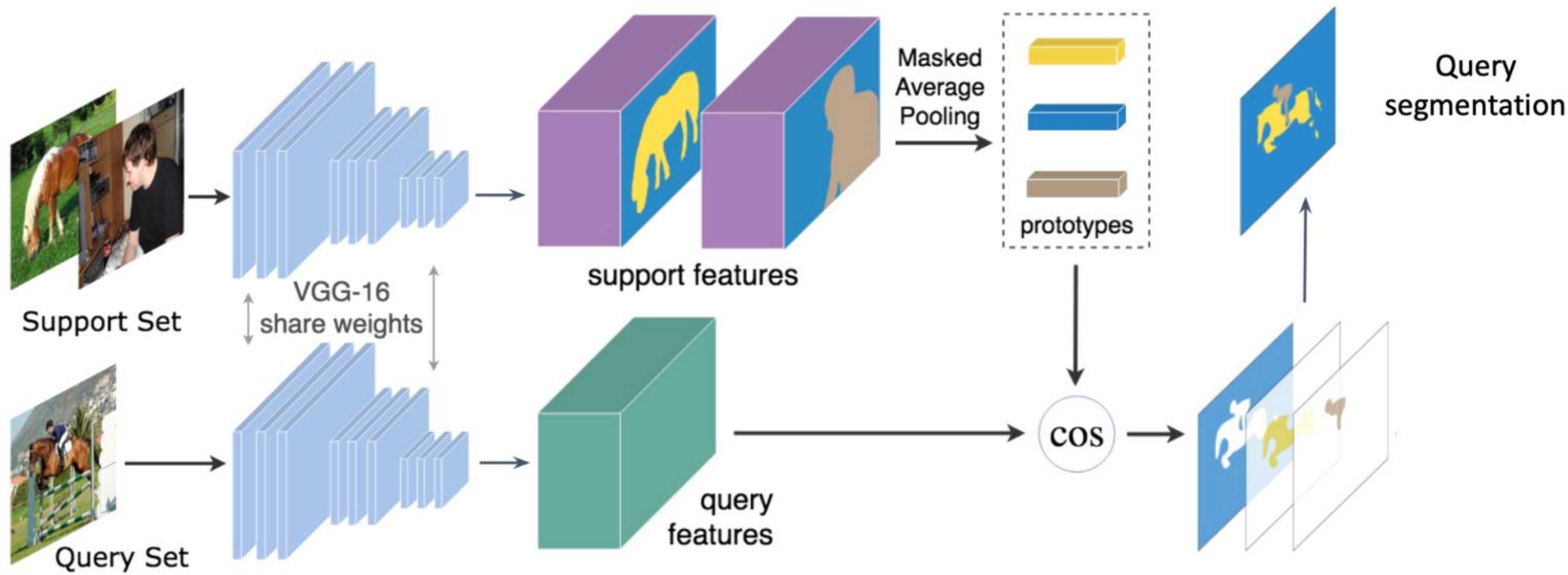
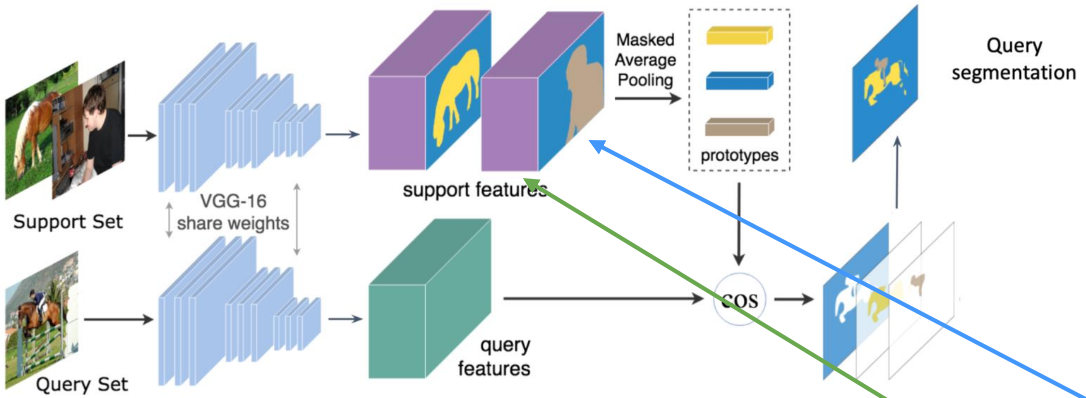


Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV’19

Few-shot Segmentation



How does it work?

Training

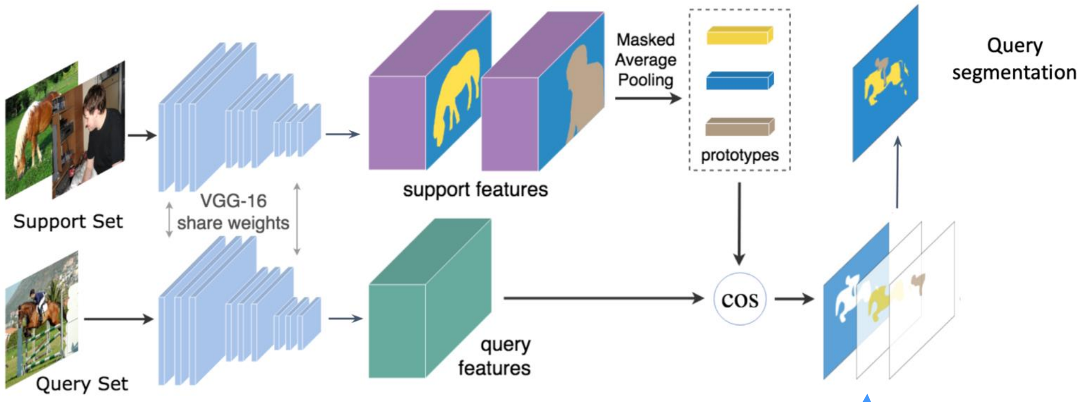
Prototype per class

$$p_c = \frac{1}{K}\sum_k \frac{\sum_{x,y} F_{c,k}^{(x,y)} \mathbb{1}[M_{c,k}^{(x,y)} = c]}{\sum_{x,y} \mathbb{1}[M_{c,k}^{(x,y)} = c]}$$

Over the k shots

Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV’19

Few-shot Segmentation



How does it work?

Training

Distance
(e.g., cosine)

Prototype per class

$$p_c = \frac{1}{K}\sum_k \frac{\sum_{x,y} F_{c,k}^{(x,y)} \mathbb{1}[M_{c,k}^{(x,y)} = c]}{\sum_{x,y} \mathbb{1}[M_{c,k}^{(x,y)} = c]}$$

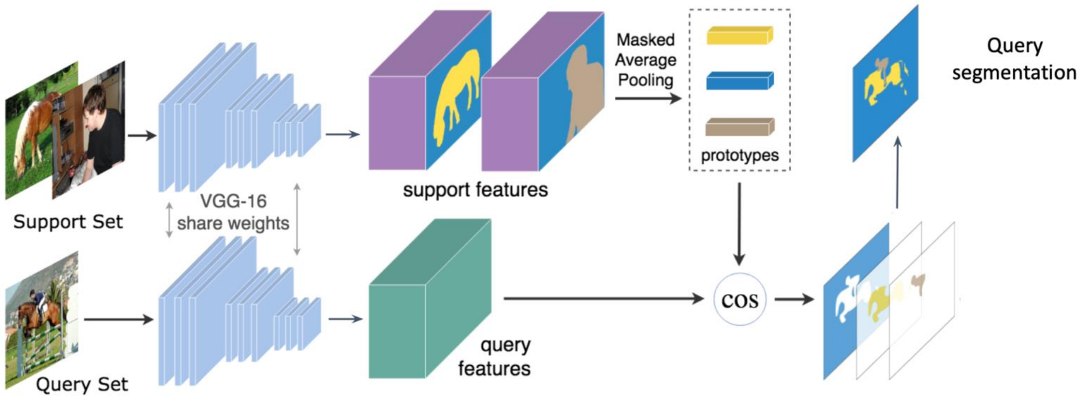
Softmax for class j

$$\tilde{M}_{q;j}^{(x,y)} = \frac{\exp(-\alpha d(F_q^{(x,y)}, p_j))}{\sum_{p_j \in \mathcal{P}} \exp(-\alpha d(F_q^{(x,y)}, p_j))}$$

Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV'19

Few-shot Segmentation

How does it work?



Training

Prototype per class

$$p_c = \frac{1}{K}\sum_k \frac{\sum_{x,y} F_{c,k}^{(x,y)} \mathbb{1}[M_{c,k}^{(x,y)} = c]}{\sum_{x,y} \mathbb{1}[M_{c,k}^{(x,y)} = c]}$$

Softmax for class j

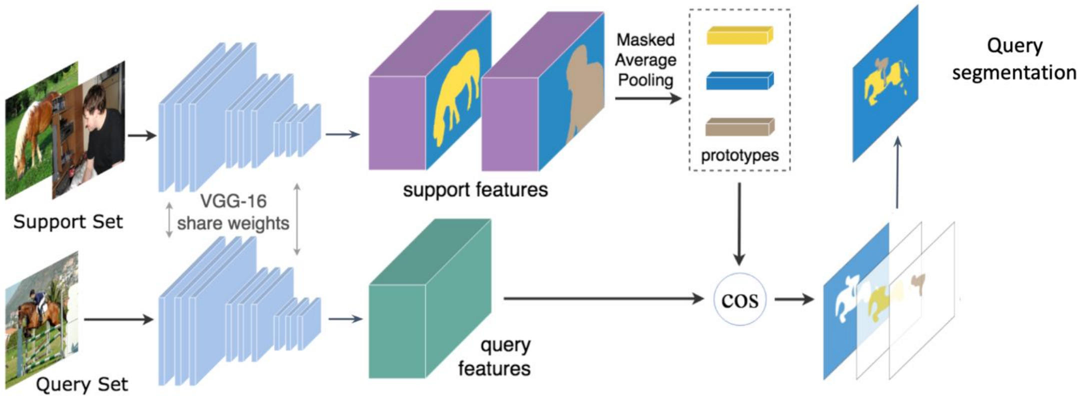
$$\tilde{M}_{q;j}^{(x,y)} = \frac{\exp(-\alpha d(F_q^{(x,y)}, p_j))}{\sum_{p_j \in \mathcal{P}} \exp(-\alpha d(F_q^{(x,y)}, p_j))}$$

Segmentation loss

$$\mathcal{L}_{\text{seg}} = -\frac{1}{N}\sum_{x,y}\sum_{p_j \in \mathcal{P}} \mathbb{1}[M_q^{(x,y)} = j] \log \tilde{M}_{q;j}^{(x,y)}$$

Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV'19

Few-shot Segmentation



How does it work?

Inference

Softmax for class j

$$\tilde{M}_{q;j}^{(x,y)} = \frac{\exp(-\alpha d(F_q^{(x,y)}, p_j))}{\sum_{p_j \in \mathcal{P}} \exp(-\alpha d(F_q^{(x,y)}, p_j))}$$

Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV'19

Few-shot Segmentation

Improving prototypes

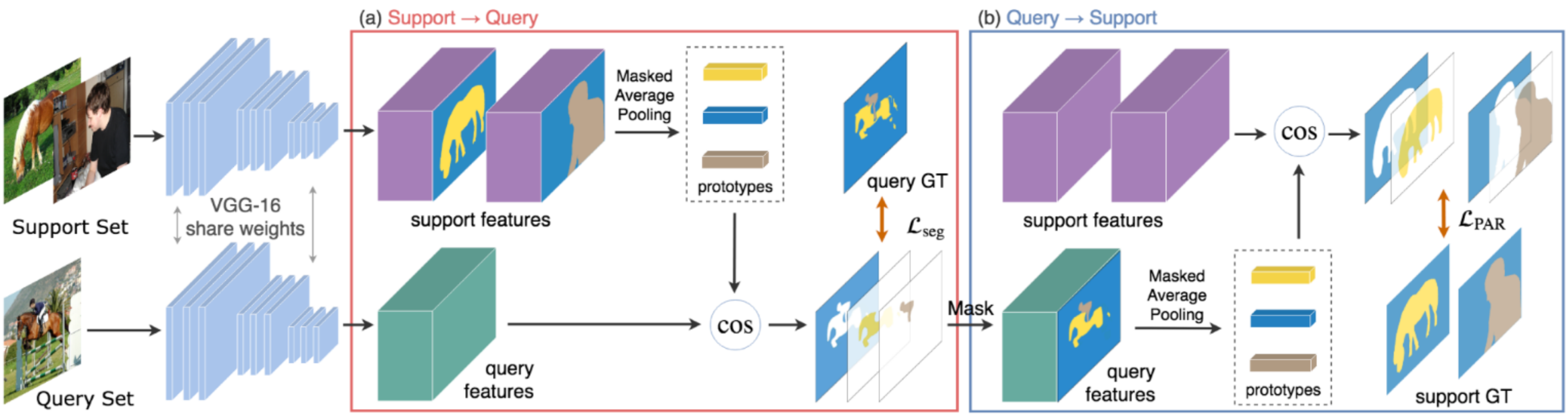
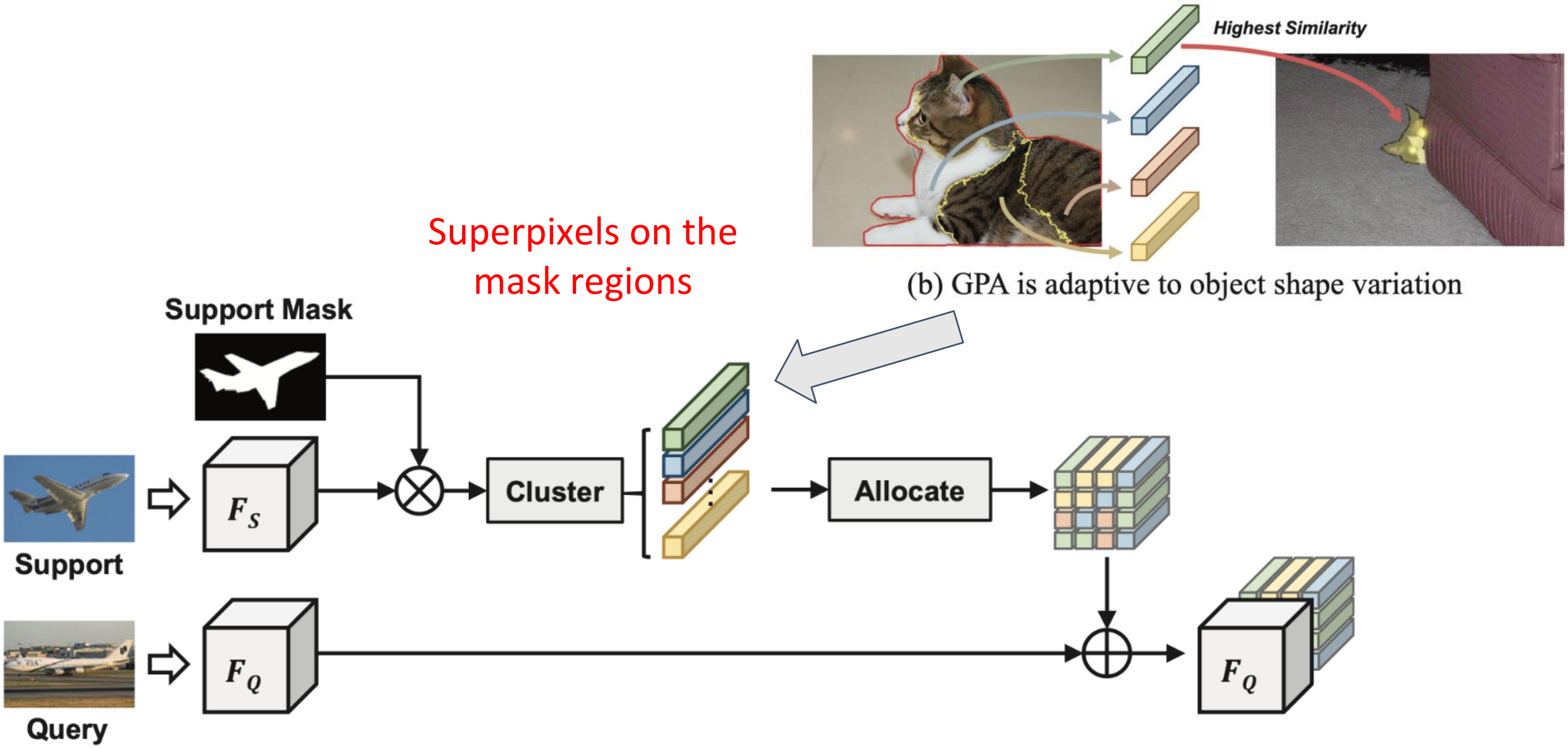


Image from Wang et al. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. ICCV'19

Few-shot Segmentation

Improving prototypes

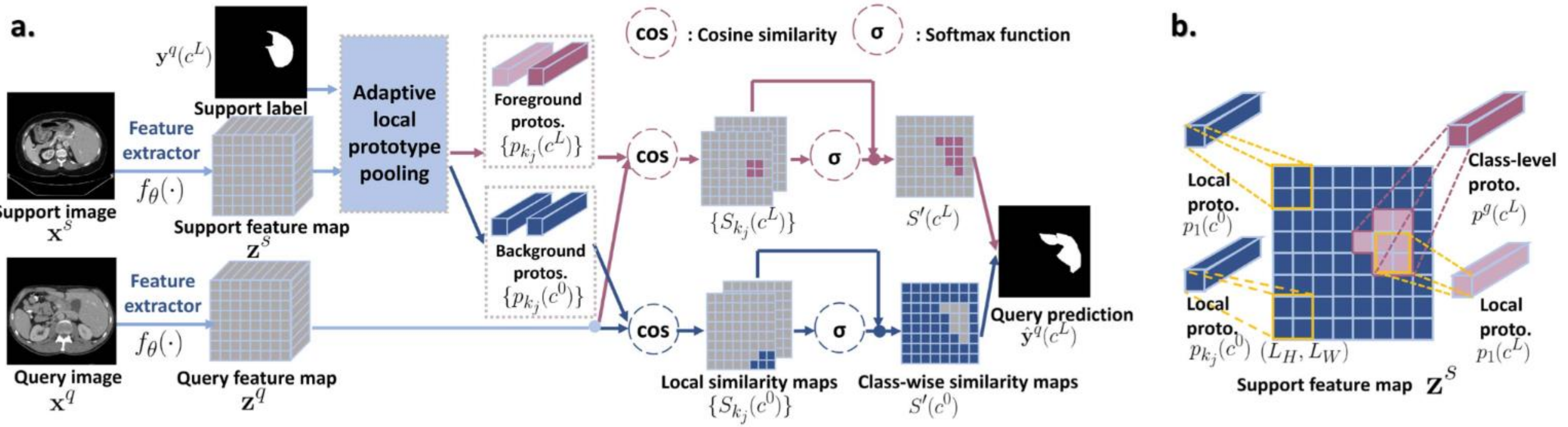
Superpixels on the mask regions

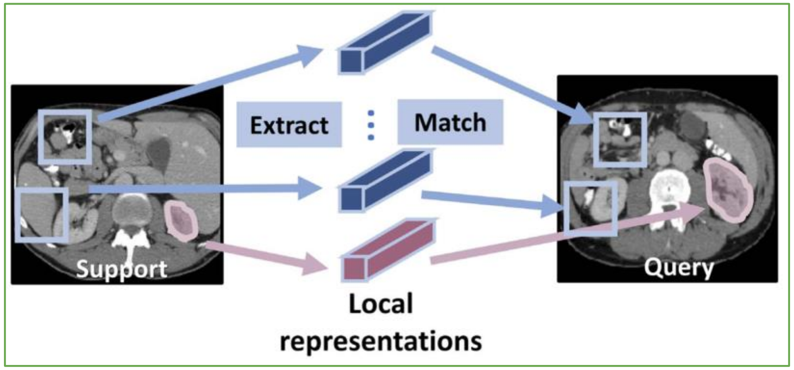


Li et al. Adaptive Prototype Learning and Allocation for Few-Shot Segmentation. CVPR'21

Few-shot Segmentation

Improving prototypes

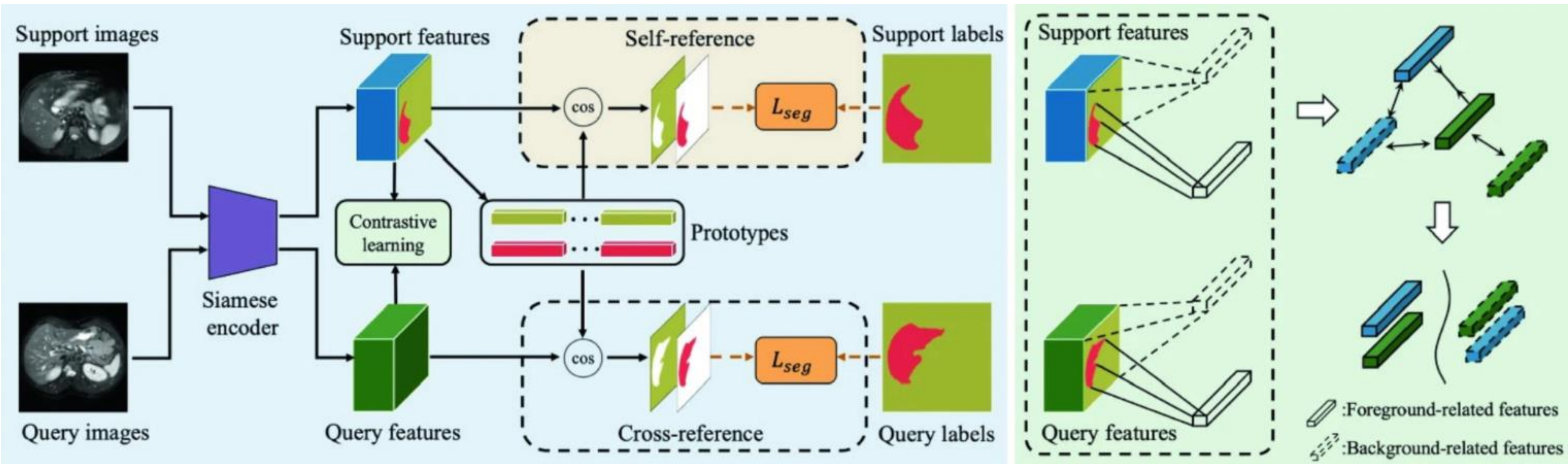




Ouyang et al. Self-supervised learning for few-shot medical image segmentation. IEEE TMI'22

Few-shot Segmentation

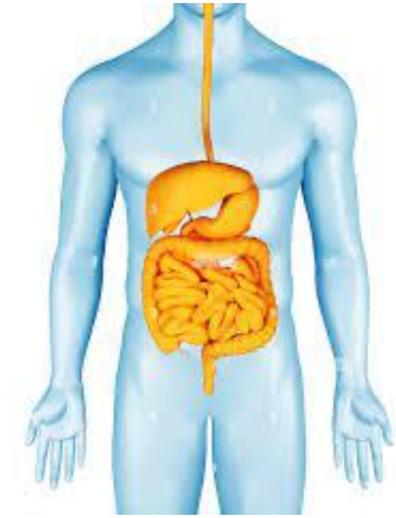
Improving prototypes



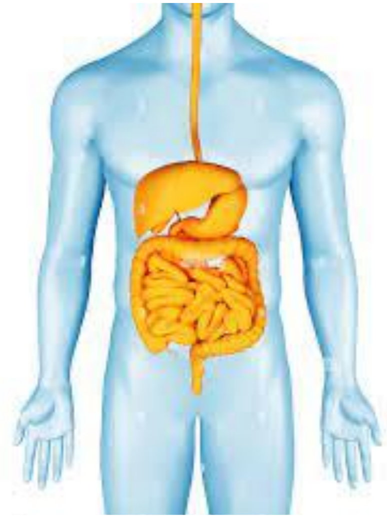
Wang et al. Few-shot Medical Image Segmentation Regularized with Self-reference and Contrastive Learning. MICCAI'22

Few-shot Segmentation

Limitations



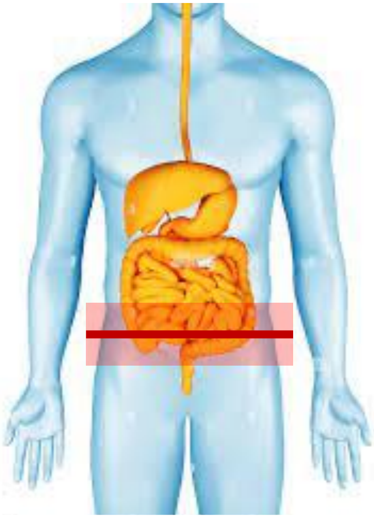
Support



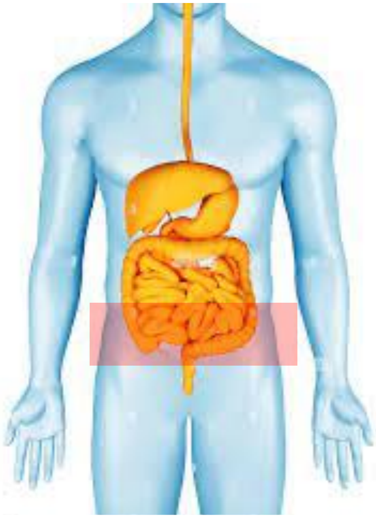
Query

Few-shot Segmentation

Limitations



Support

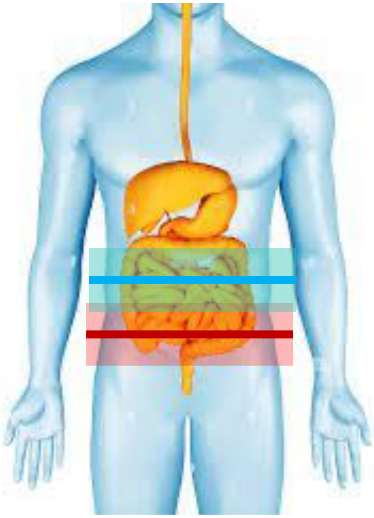


Query

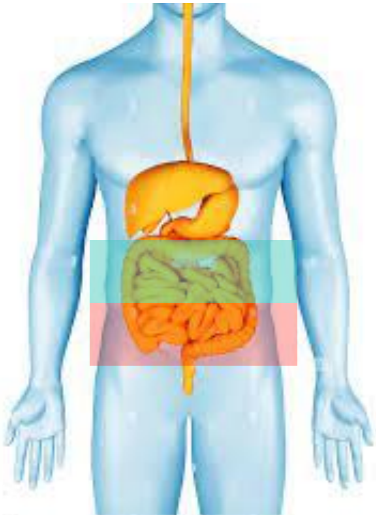
Chunk trick

Few-shot Segmentation

Limitations



Support

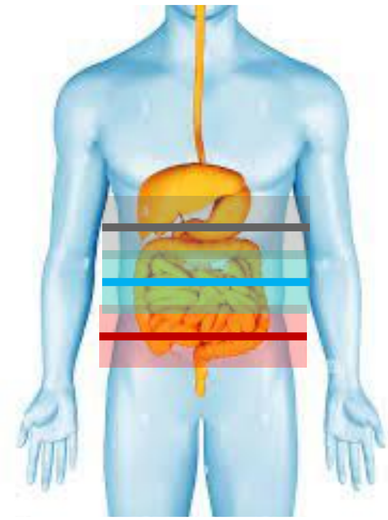


Query

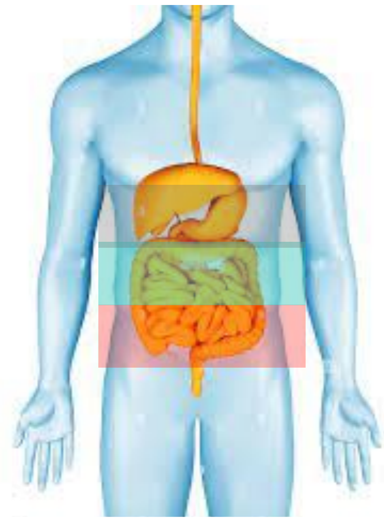
Chunk trick

Few-shot Segmentation

Limitations



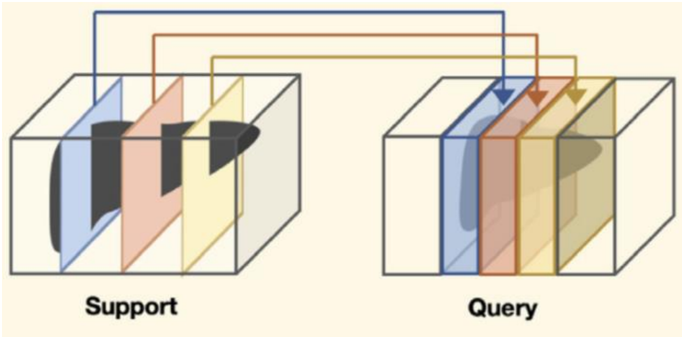
Support



Query

Chunk trick

The volume is pre-split into n chunks (n going from 3 to 12)



Few-shot Segmentation

Limitations (both classification and segmentation)

Most meta-learning to learn

Vinyal et al, (Neurips '16),
Snell et al, (Neurips '17),
Sung et al, (CVPR ' 18),
Finn et al, (ICML' 17),
Ravi et al, (ICLR' 17),
Lee et al, (CVPR' 19),
Hu et al, (ICLR '20),
Ye et al, (CVPR '20),
Ouyang et al, (TMI'22),...

- ❌ Convoluted meta-learning approaches
- ❌ Models trained under the learning-to-learn paradigm cannot be re-used with different models
- ❌ They are tailored to the same-task paradigm (e.g., models trained under 1-shot do not perform well when 5-shots are available)
- ❌ Significant performance degradation under domain shift

Few-shot learning

A few steps backwards

Inductive fine-tuning baseline

[Chen et al., ICLR’19]; [Tian et al., ECCV’20]; [Veilleux et al., NeurIPS’21];
[Dhillon et al., ICLR’20]; [Ziko et al., ICML’20]; [Boudiaf et al., NeurIPS’20];
[Boudiaf et al., CVPR’21]; [Hajimiri et al., CVPR’23]

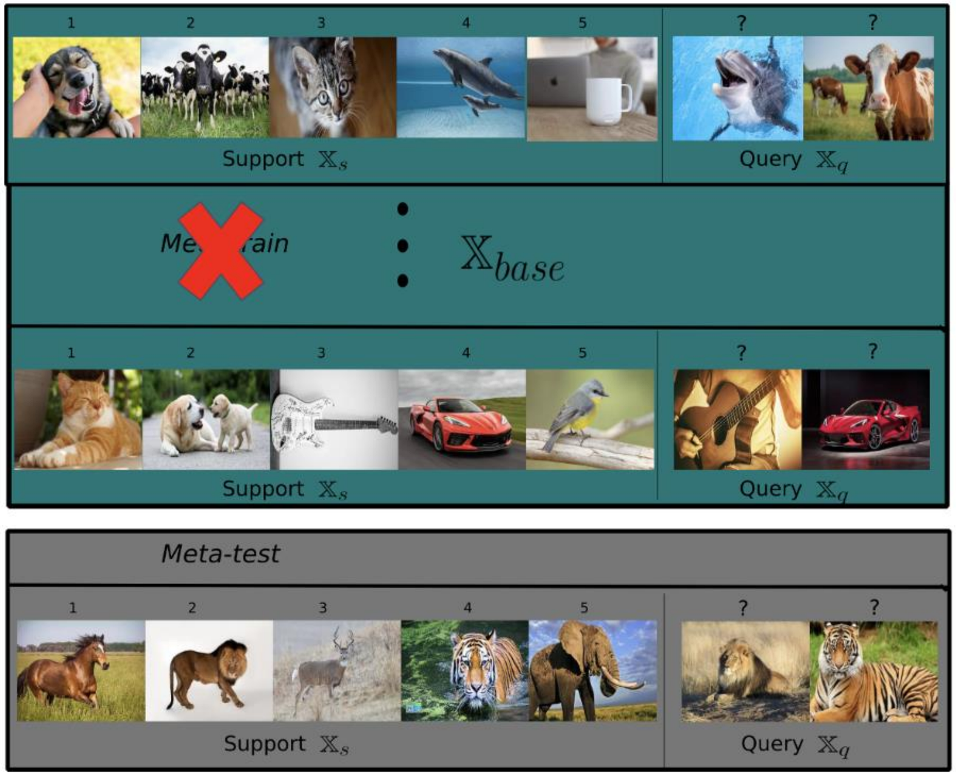
Entropy regularization

Manifold regularization

Mutual-information regularization

Few-shot learning

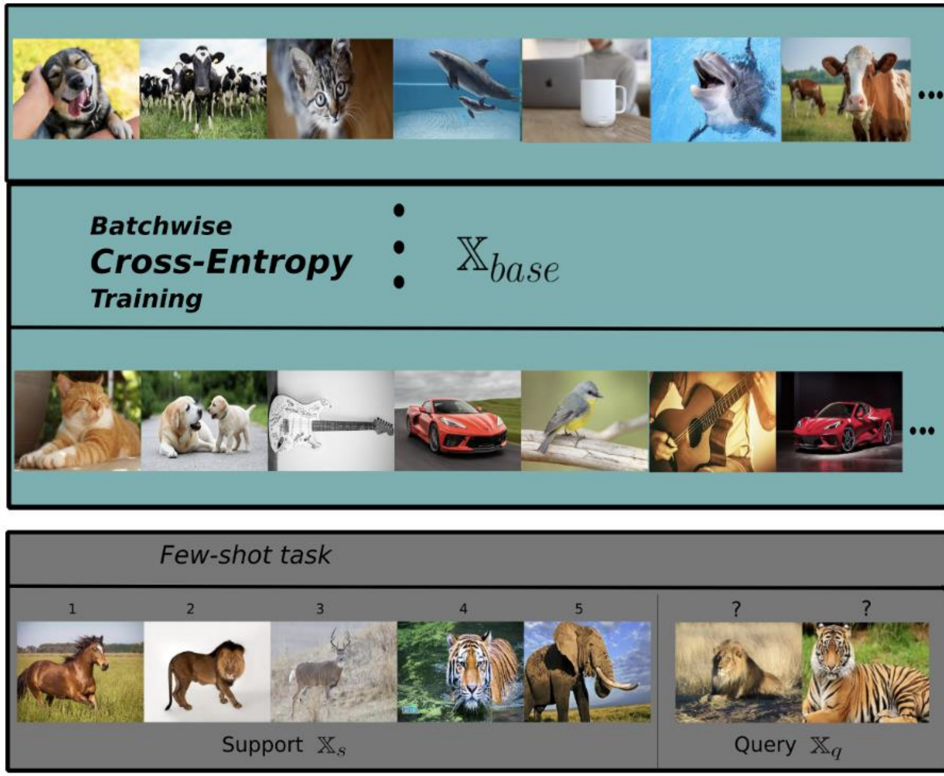
A few steps backwards



No need to ***meta-train***

Few-shot learning

A few steps backwards



Conventional training

Nearest prototype
[SimpleShot, Wang et al., Arxiv'19]

Cross-Entropy fine-tuning
[Closer look at FSC, Chen et al., ICLR'19]

46

Few-shot learning

Surprising results

Same domain

!!!!!

Method	Transd.	Backbone	*mini*-ImageNet 1-shot	*mini*-ImageNet 5-shot	*tiered*-ImageNet 1-shot	*tiered*-ImageNet 5-shot	CUB 1-shot	CUB 5-shot
MAML [9]		ResNet-18	49.6	65.7	-	-	68.4	83.5
RelatNet [40]		ResNet-18	52.5	69.8	-	-	68.6	84.0
MatchNet [45]		ResNet-18	52.9	68.9	-	-	73.5	84.5
ProtoNet [38]		ResNet-18	54.2	73.4	-	-	73.0	86.6
MTL [39]	✗	ResNet-12	61.2	75.5	-	-	-	-
vFSL [50]		ResNet-12	61.2	77.7	-	-	-	-
Neg-cosine [26]		ResNet-18	62.3	80.9	-	-	72.7	89.4
MetaOpt [22]		ResNet-12	62.6	78.6	66.0	81.6	-	-
SimpleShot [46]		ResNet-18	62.9	80.0	68.9	84.6	68.9	84.0
Distill [41]		ResNet-12	64.8	82.1	71.5	86.0	-	-
RelatNet + T [14]		ResNet-12	52.4	65.4	-	-	-	-
ProtoNet + T [14]		ResNet-12	55.2	71.1	-	-	-	-
MatchNet+T [14]		ResNet-12	56.3	69.8	-	-	-	-
TPN [28]		ResNet-12	59.5	75.7	-	-	-	-
TEAM [34]	✓	ResNet-18	60.1	75.9	-	-	-	-
Ent-min [7]		ResNet-12	62.4	74.5	68.4	83.4	-	-
CAN+T [14]		ResNet-12	67.2	80.6	73.2	84.9	-	-
LaplacianShot [51]		ResNet-18	72.1	82.3	79.0	86.4	81.0	88.7
TIM-ADM		ResNet-18	73.6	**85.0**	**80.0**	**88.5**	81.9	90.7
TIM-GD		ResNet-18	**73.9**	**85.0**	79.9	**88.5**	**82.2**	**90.8**

!!!!!

Few-shot learning

Surprising results

Domain shift

Methods	Backbone	*mini*-ImageNet → CUB 5-shot
MatchNet [45]	ResNet-18	53.1
MAML [9]	ResNet-18	51.3
ProtoNet [38]	ResNet-18	62.0
RelatNet [40]	ResNet-18	57.7
SimpleShot [46]	ResNet-18	64.0
GNN [42]	ResNet-10	66.9
Neg-Cosine [26]	ResNet-18	67.0
Baseline [5]	ResNet-18	65.6
LaplacianShot [51]	ResNet-18	66.3
TIM-ADM	ResNet-18	70.3
TIM-GD	ResNet-18	**71.0**

Few-shot learning

Example of a non meta-learning approach

Segmentation task

* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task

* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

$$\min \quad -\frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} l(\mathbf{y}^p, \mathbf{s}_\theta^p) - \lambda_{\mathcal{H}}\frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j \log(\mathbf{s}_\theta^j) + \lambda_{KL}(\hat{\mathbf{s}}_\theta^{\mathcal{Q}} \log(\frac{\hat{\mathbf{s}}_\theta^{\mathcal{Q}}}{\tau}))$$

Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task

* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

$$\min \quad -\frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} l(\mathbf{y}^p, \mathbf{s}_\theta^p) - \lambda_{\mathcal{H}}\frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j \log(\mathbf{s}_\theta^j) + \lambda_{KL}(\hat{\mathbf{s}}_\theta^{\mathcal{Q}} \log(\frac{\hat{\mathbf{s}}_\theta^{\mathcal{Q}}}{\tau}))$$

CE on supervised images (i.e., support)

Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task

* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

$$\min \quad -\frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} l(\mathbf{y}^p, \mathbf{s}_\theta^p) - \lambda_{\mathcal{H}}\frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j \log(\mathbf{s}_\theta^j) + \lambda_{KL}(\hat{\mathbf{s}}_\theta^{\mathcal{Q}} \log(\frac{\hat{\mathbf{s}}_\theta^{\mathcal{Q}}}{\tau}))$$

CE on supervised images (i.e., support)

Entropy on unsupervised images (i.e., queries)

Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task

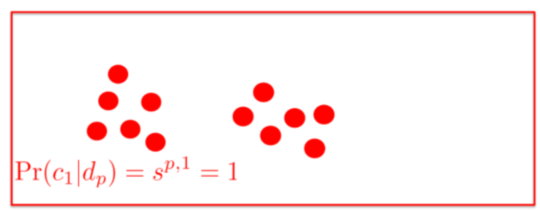
* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

$$\min \; -\frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} l(\mathbf{y}^p, \mathbf{s}_\theta^p) - \lambda_{\mathcal{H}}\frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j \log(\mathbf{s}_\theta^j) + \lambda_{KL}(\hat{\mathbf{s}}_\theta^{\mathcal{Q}} \log(\frac{\hat{\mathbf{s}}_\theta^{\mathcal{Q}}}{\tau}))$$

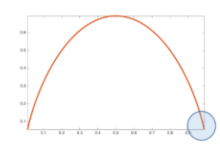
CE on supervised images (i.e., support)

Entropy on unsupervised images (i.e., queries)

Problem if only the entropy is minimized!



This bad solution also has a minimum entropy!!!



Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task

* The initial model is trained over the base classes following standard segmentation training (i.e., CE)

$$\min \quad -\frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} l(\mathbf{y}^p, \mathbf{s}_\theta^p) - \lambda_{\mathcal{H}}\frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j \log(\mathbf{s}_\theta^j) + \lambda_{KL}(\hat{\mathbf{s}}_\theta^{\mathcal{Q}} \log(\frac{\hat{\mathbf{s}}_\theta^{\mathcal{Q}}}{\tau}))$$

CE on supervised images (i.e., support)

Entropy on unsupervised images (i.e., queries)

KL (to impose target proportions)

$$\hat{\mathbf{s}}_\theta^{\mathcal{Q}} = \frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \mathbf{s}_\theta^j$$

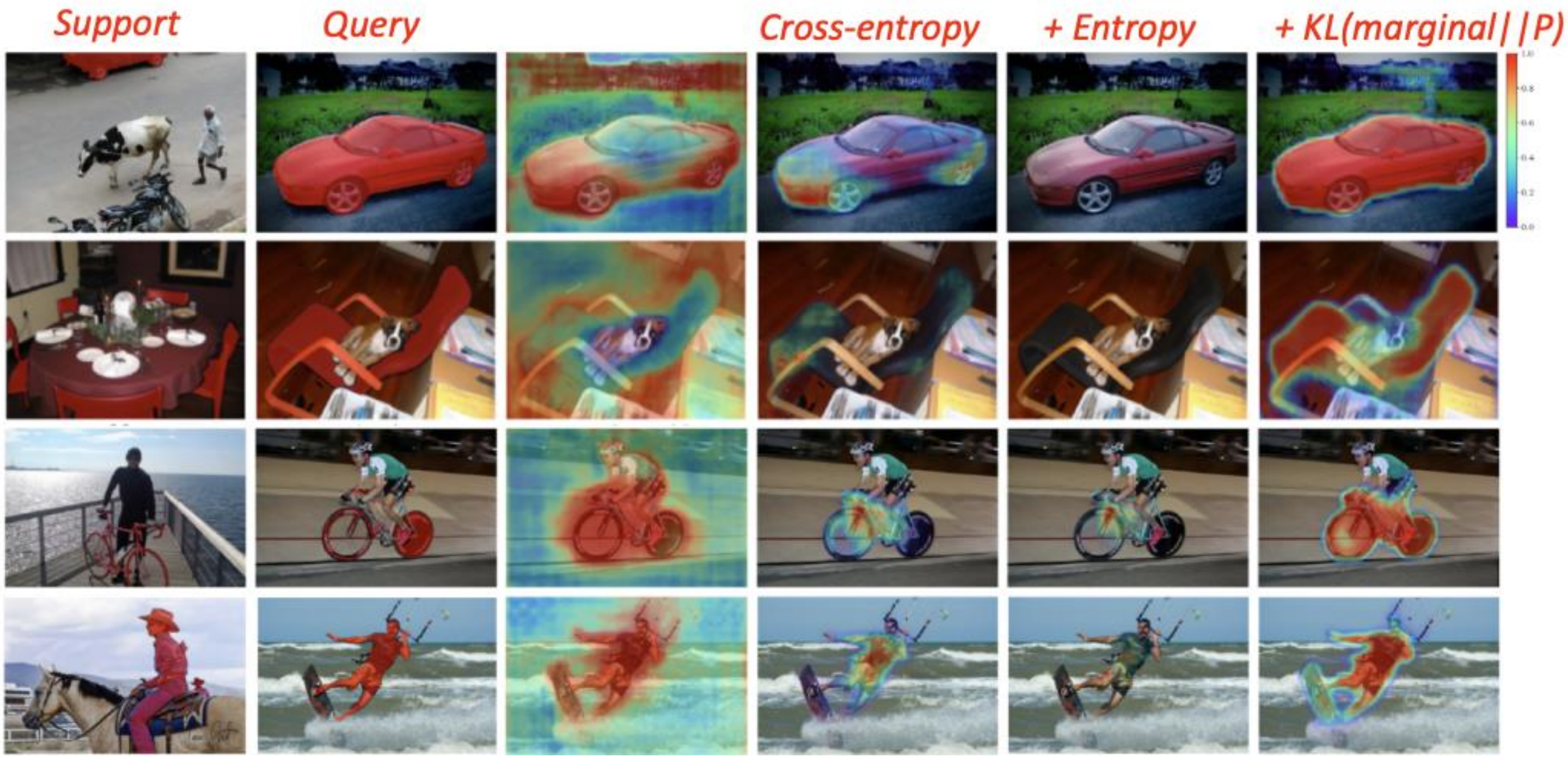
A priori proportion $\rightarrow$ $\tau \in [0, 1]$

Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

Few-shot learning

Example of a non meta-learning approach

Segmentation task



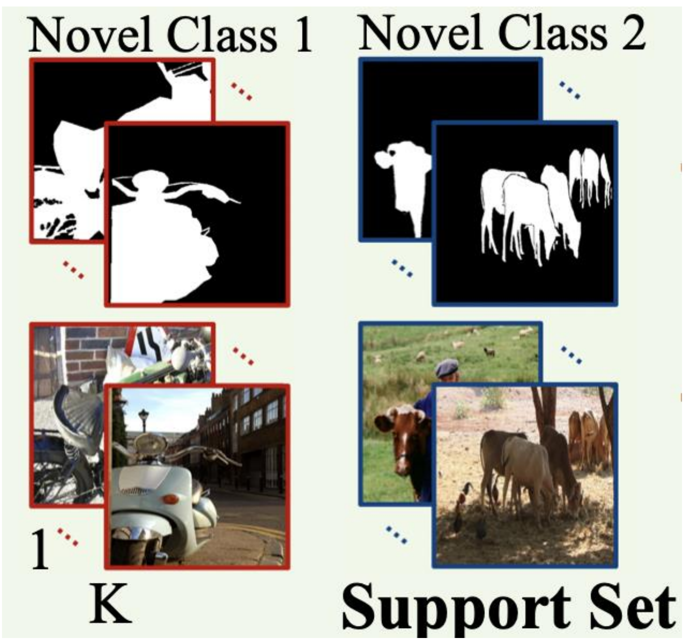
Boudiaf et al., Few-shot segmentation without meta-learning: A good transductive inference is all you need? CVPR 2021

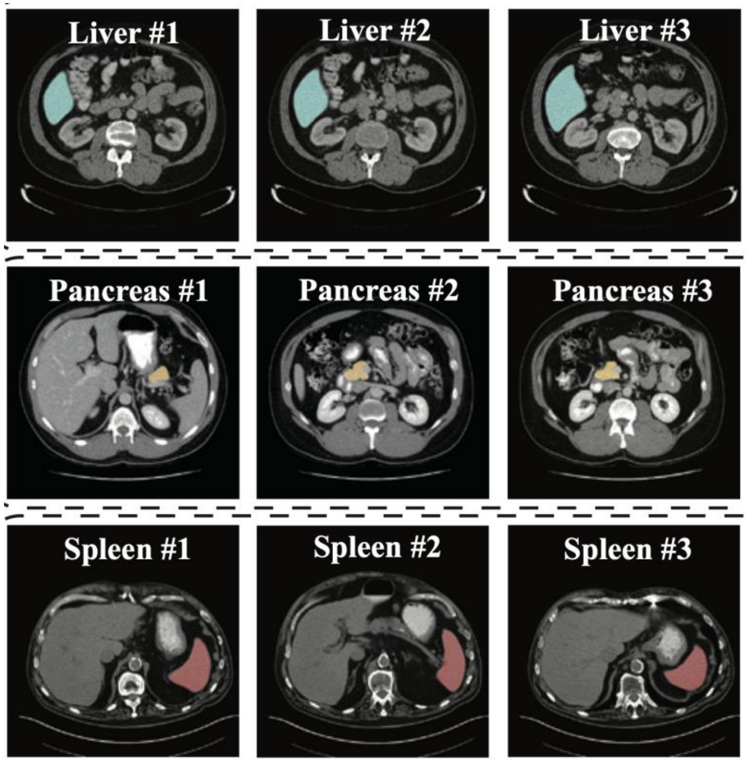
Few-shot learning

Limitations of few-shot segmentation

Class-ambiguity (in the context of generalization)

Standard FSS training assumes what is not the target class is background.



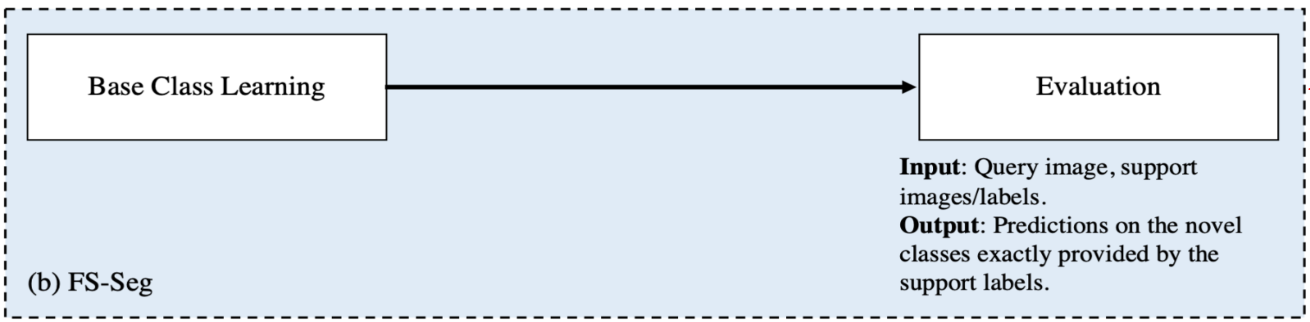


Few-shot learning

Limitations of few-shot segmentation

They cannot keep the performance on base classes (generalized segmentation)

Standard FSS setting:



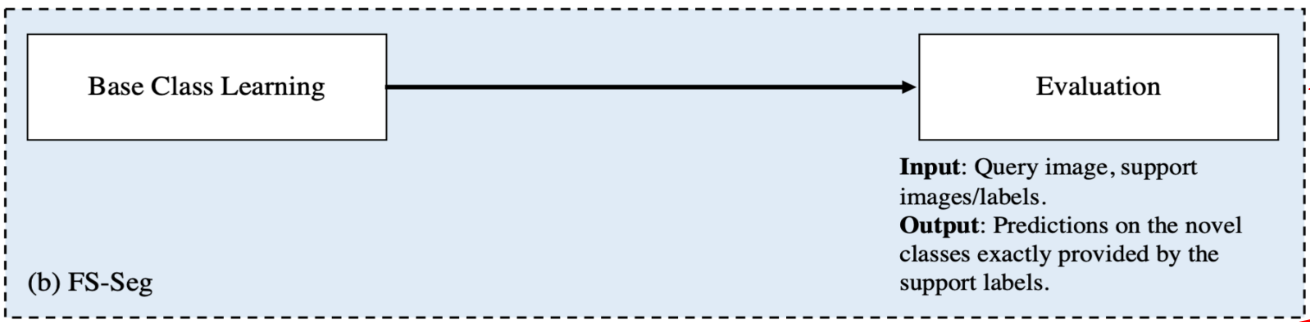
Novel categories

Few-shot learning

Limitations of few-shot segmentation

They cannot keep the performance on base classes (generalized segmentation)

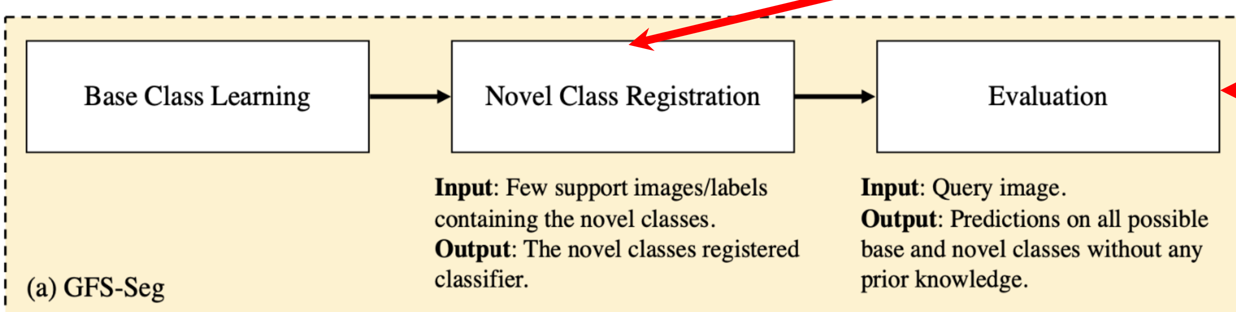
Standard FSS setting:



Novel categories

Support supervision (novel categories)

Generalized FSS setting:



Novel + base categories

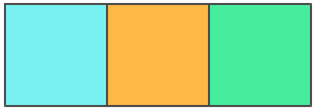
Efficient adaptation

Foundation models

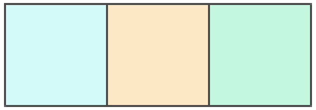
Preliminaries
(CLIP)

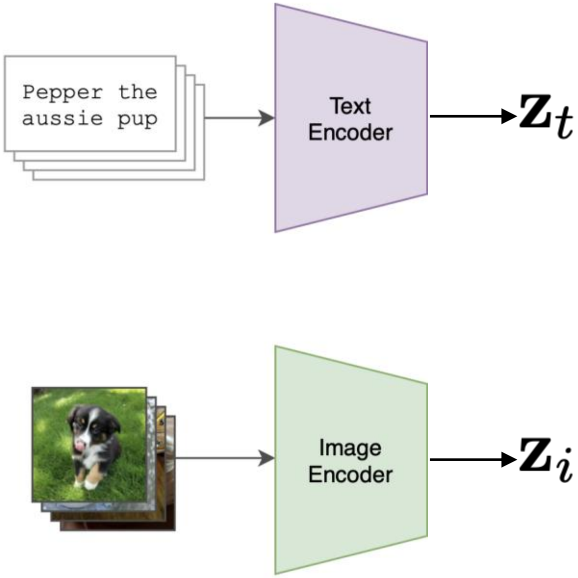
For a given batch

3 images



Corresponding texts





Efficient adaptation

Foundation models

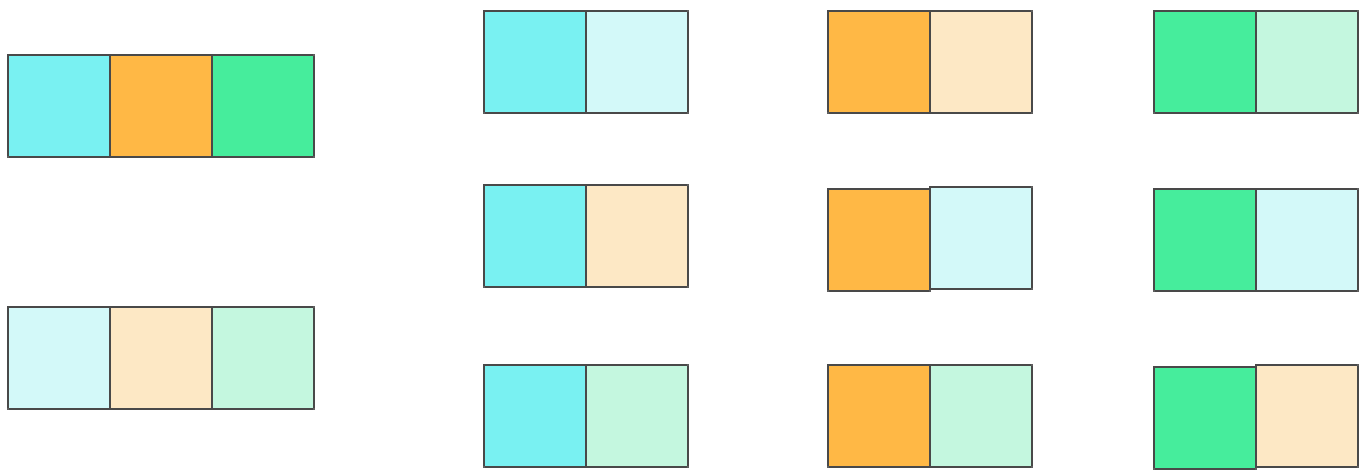
Preliminaries
(CLIP)

We generate image-text paris with all the images-texts

For a given batch

3 images

Corresponding texts



Efficient adaptation

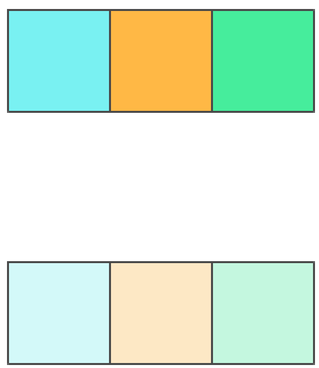
Foundation models

Preliminaries
(CLIP)

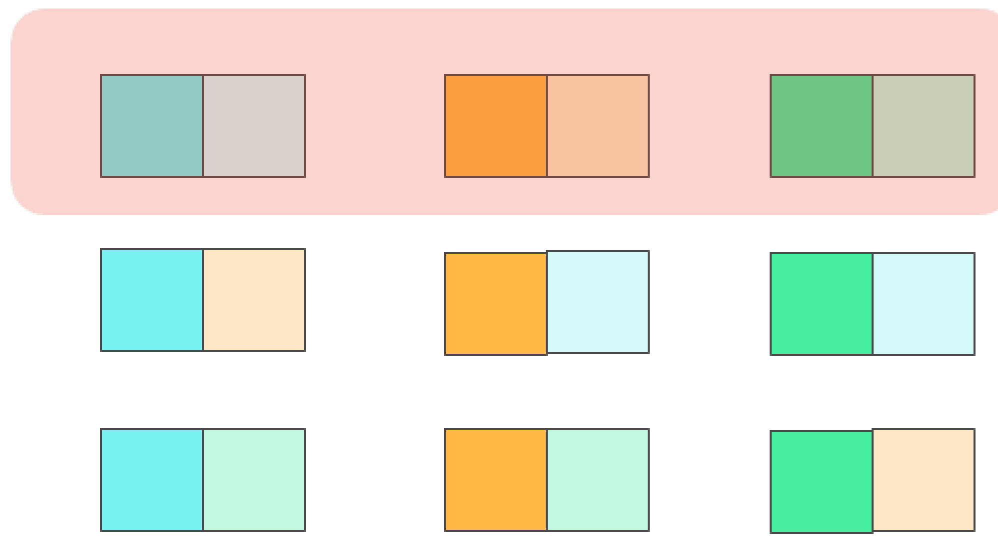
For a given batch

3 images

Corresponding texts



We generate image-text paris with all the images-texts



Maximize the cosine distance $(\mathbf{z}_i, \mathbf{t}_i)$

Efficient adaptation

Foundation models

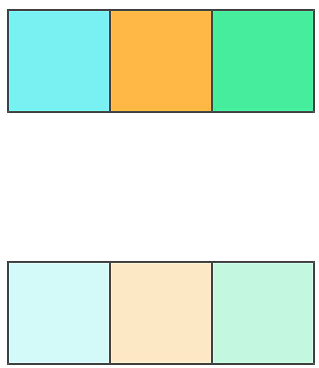
Preliminaries
(CLIP)

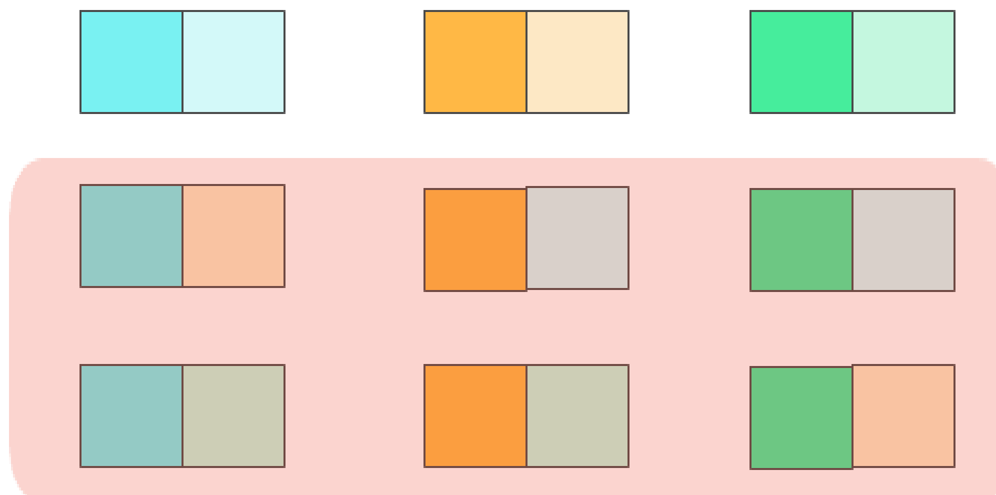
We generate image-text paris with all the images-texts

For a given batch

3 images

Corresponding texts





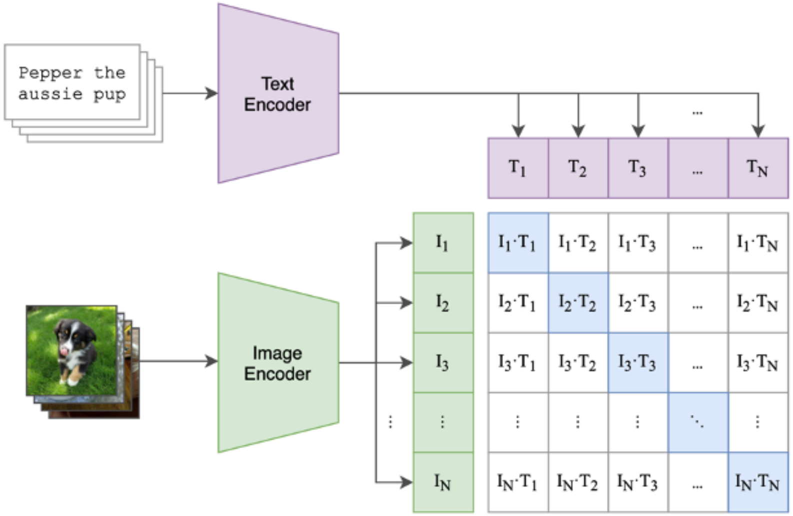
Minimize the cosine distance $(\mathbf{z}_i, \mathbf{t}_i)$

Efficient adaptation

Foundation models

Preliminaries
(CLIP)

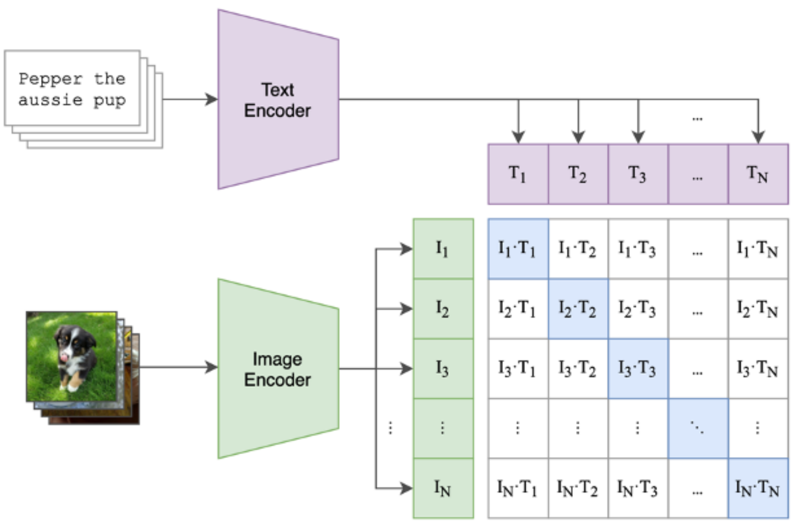
We generate image-text paris with all the images-texts



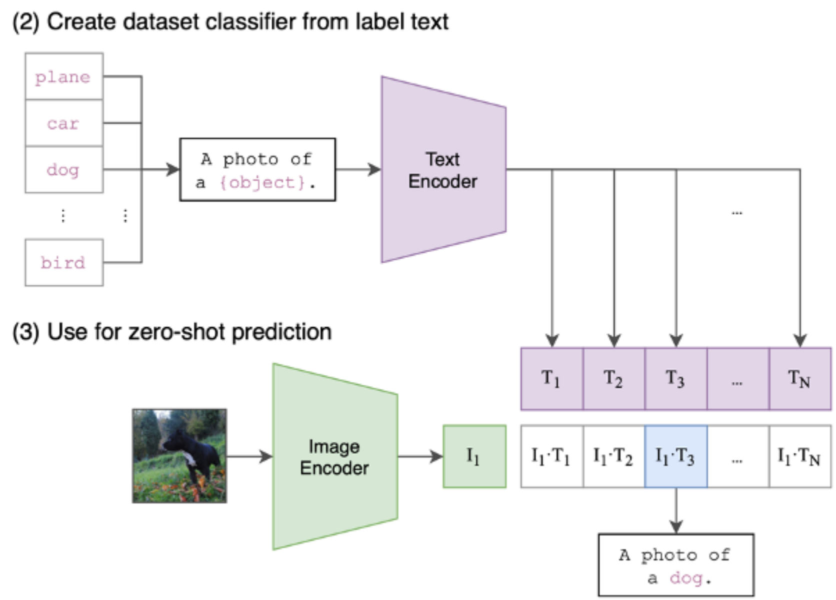
Efficient adaptation

Foundation models

Preliminaries
(CLIP)



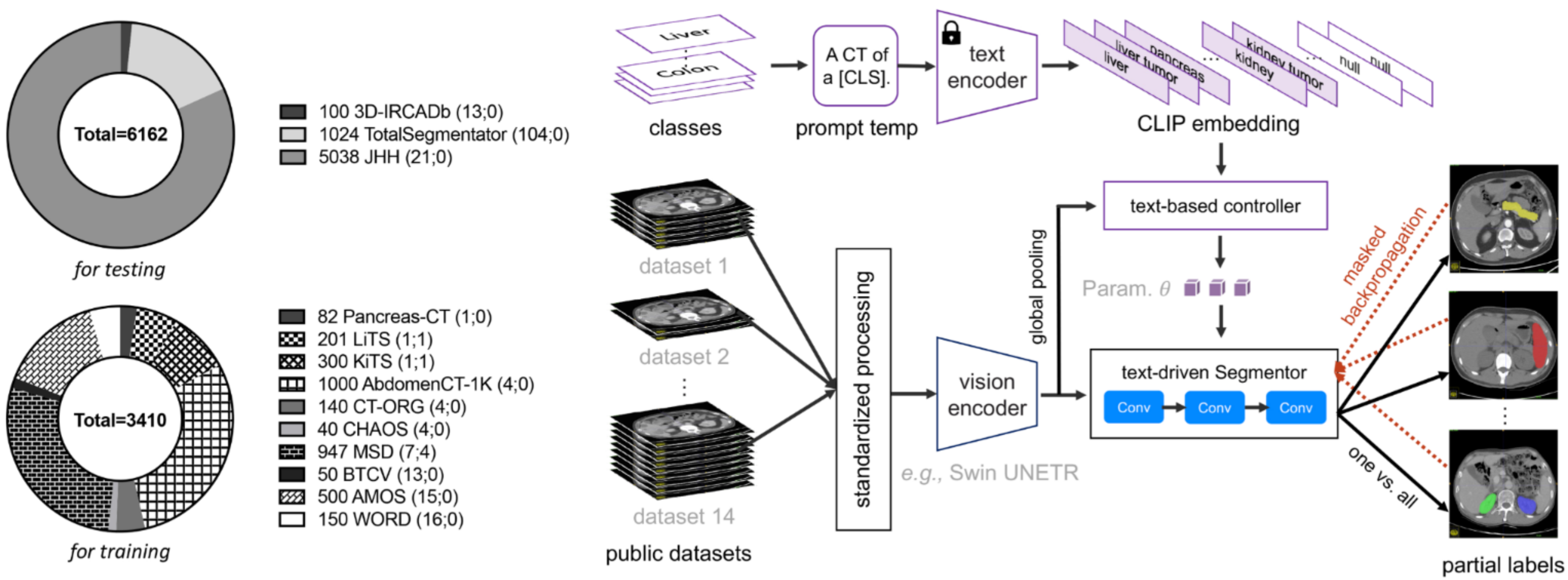
Inference (novel classes)



Efficient adaptation

Foundation models

Models that are trained on a large set of labeled datasets

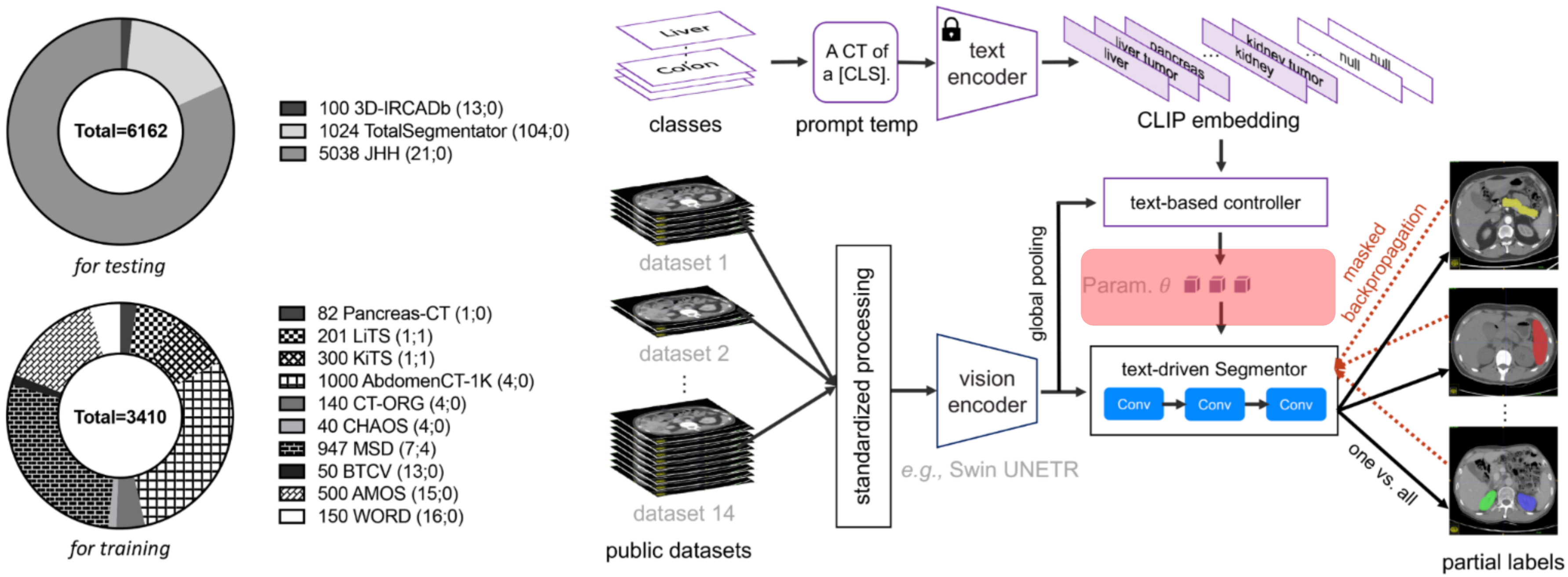


Liu et al., CLIP-Driven Universal Model for Organ Segmentation and Tumor Detection, Arxiv’23

Efficient adaptation

Foundation models

Models that are trained on a large set of labeled datasets



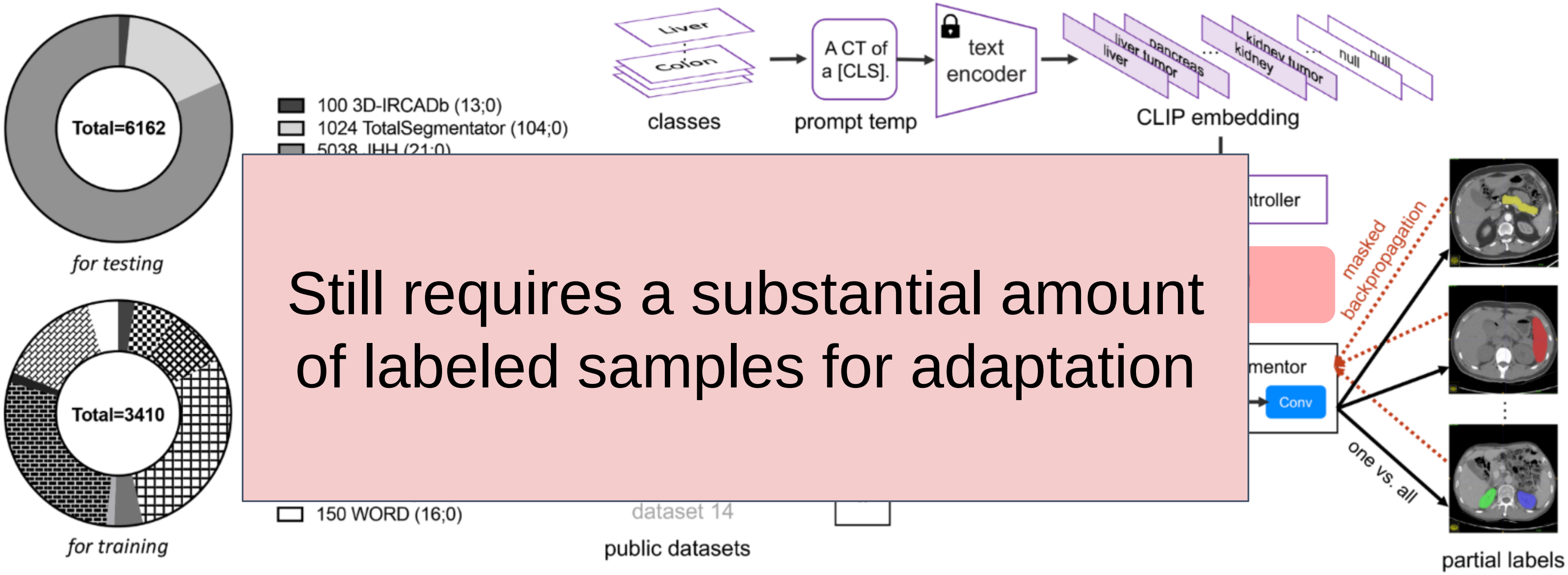
Only that is adapted

Liu et al., CLIP-Driven Universal Model for Organ Segmentation and Tumor Detection, Arxiv'23

Efficient adaptation

Foundation models

Models that are trained on a large set of labeled datasets



Still requires a substantial amount of labeled samples for adaptation

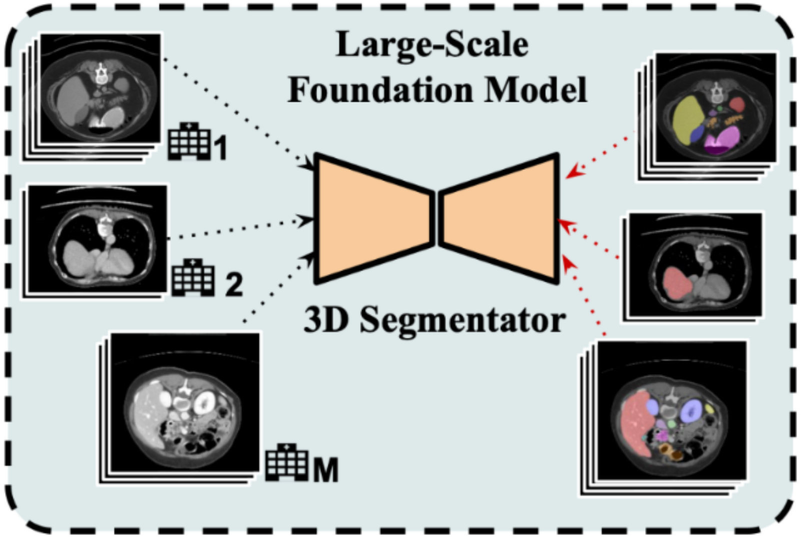
Only that is adapted

Liu et al., CLIP-Driven Universal Model for Organ Segmentation and Tumor Detection, Arxiv'23

Efficient adaptation

Foundation models

Trained on a large set of labeled datasets

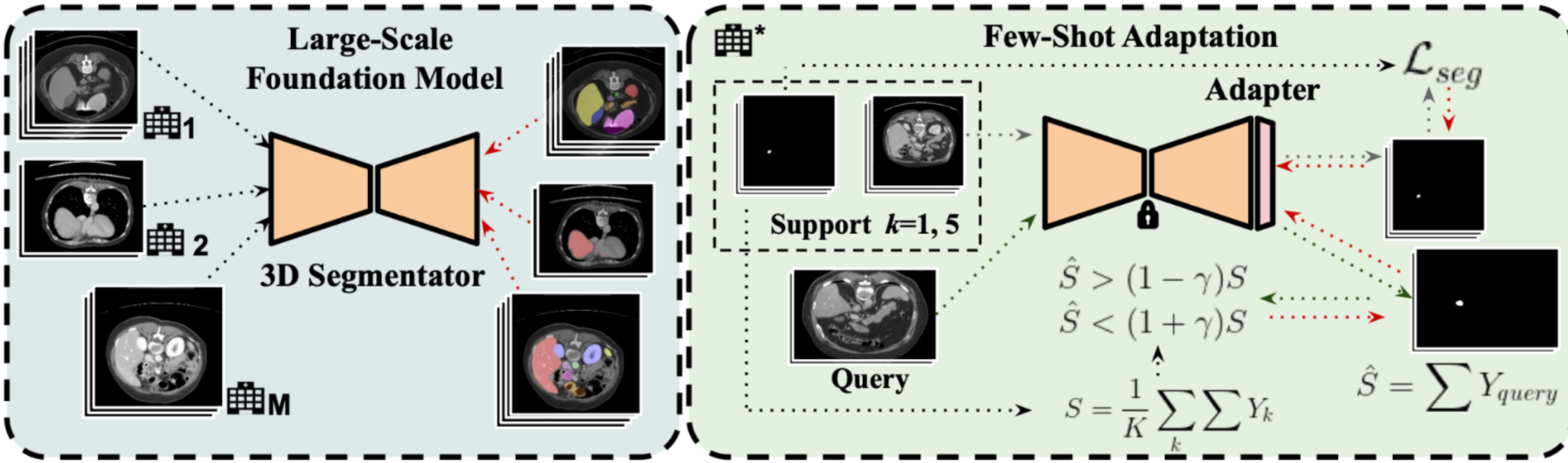


Silva-Rodriguez et al., Transductive few-shot adapters for medical image segmentation, Arxiv'23

Efficient adaptation

Foundation models

Adaptation is done only using k shots



Silva-Rodriguez et al., Transductive few-shot adapters for medical image segmentation, Arxiv'23

Take home message

- Few shot learning can alleviate the problem of scarce labeled data.

- Recent literature has taken a step-back (regarding the meta-learning or *learning to learn* paradigm)

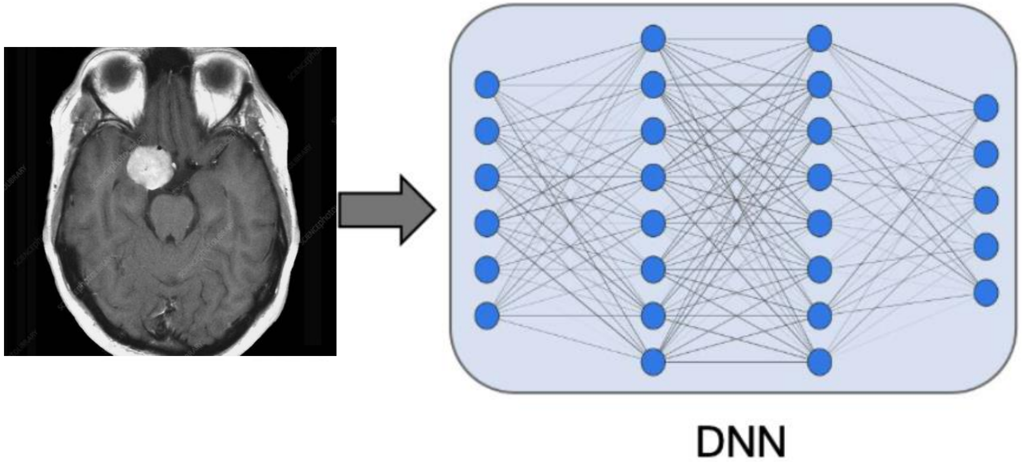
- If you have prior knowledge, use it.

- Foundation models with efficient adaptation could be a realistic alternative.

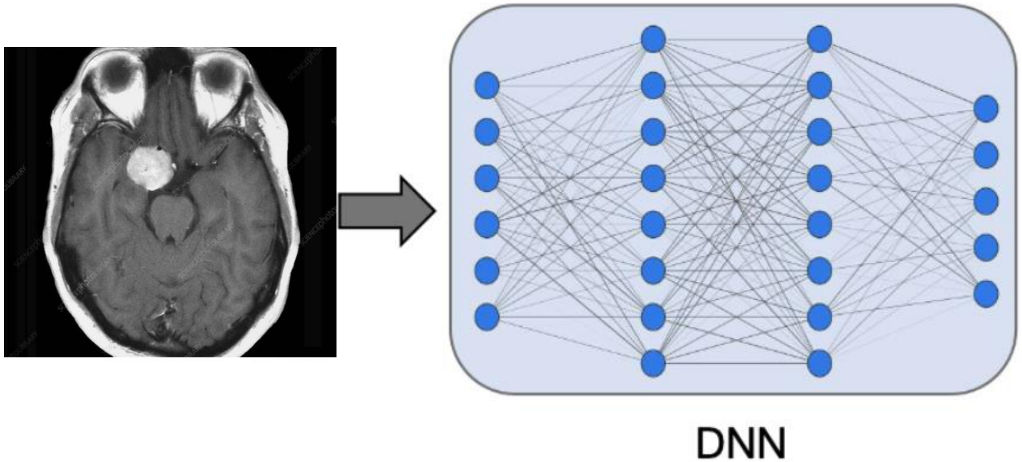
Calibration

Motivation

Which model would you choose?



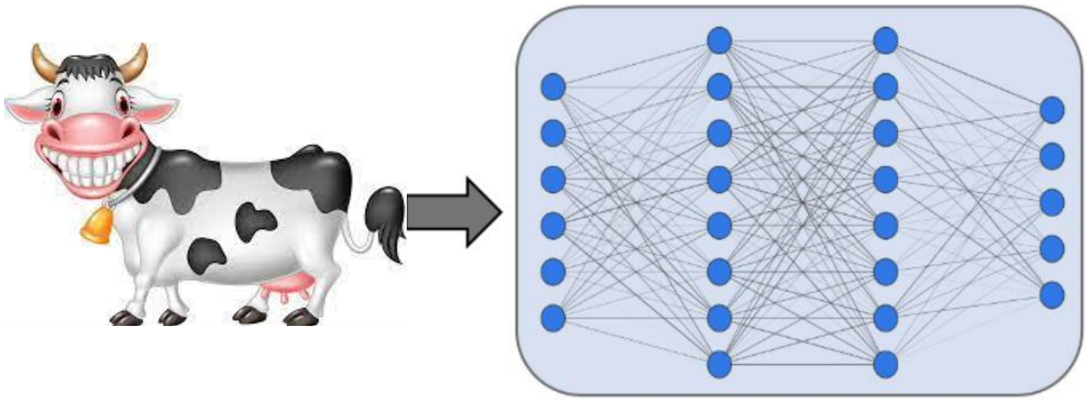
Accuracy: 99%



Accuracy: 92%

Motivation

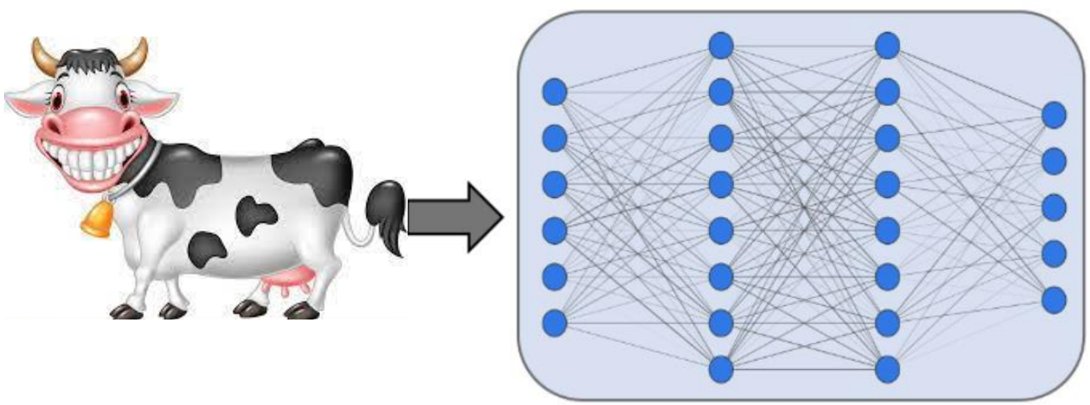
Which model would you choose?



DNN

Accuracy: 99%

Prediction:
'Tumour at 100%'



DNN

Accuracy: 92%

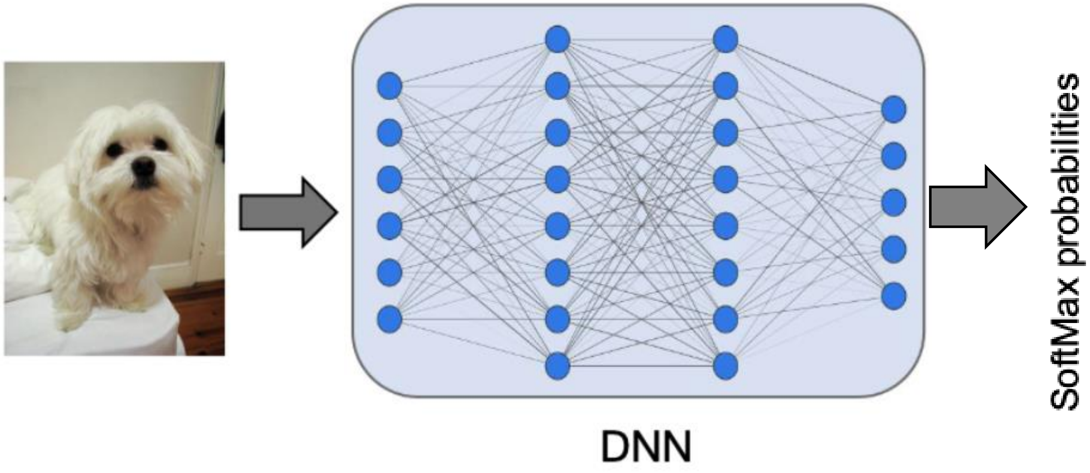
Prediction:
'I do not know'

Problem

Standard loss functions

Cross-entropy

The way we provide the labels (one-hot) encourages the network to have low-entropy predictions



$$\mathbf{s} = [s_0, s_1, ..., s_{N-1}]$$

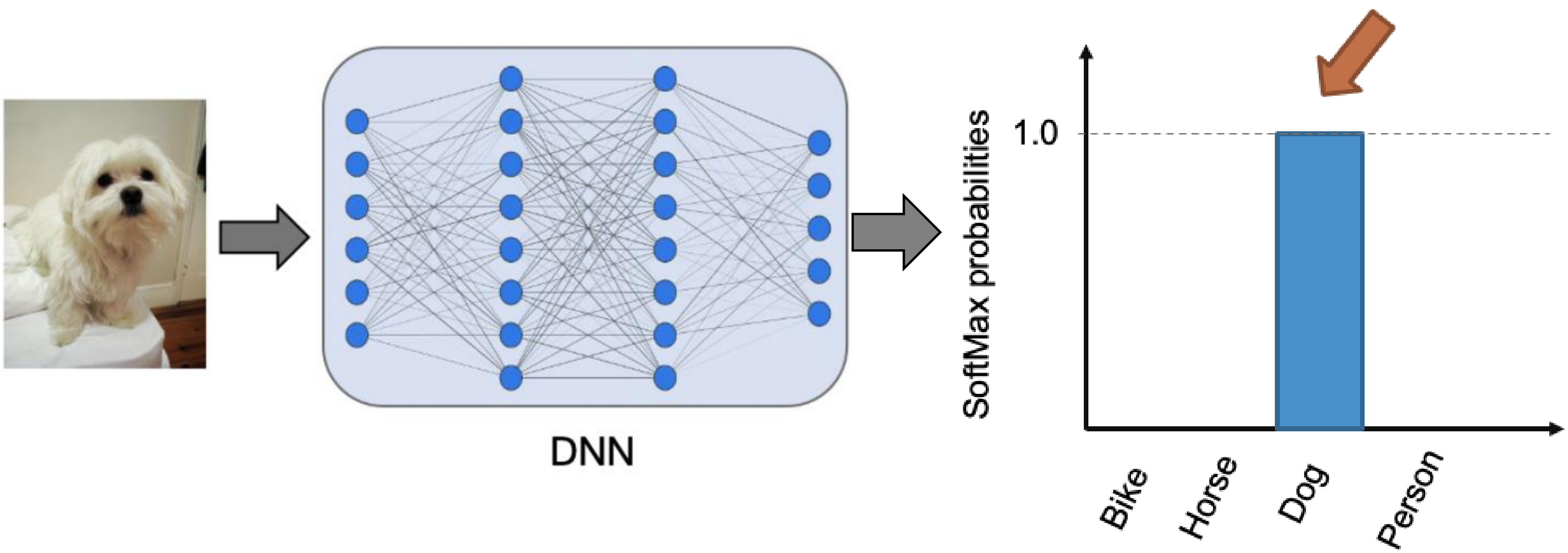
Problem

Standard loss functions

Cross-entropy

The way we provide the labels (one-hot) encourages the network to have low-entropy predictions

Target objective when training a neural network with CE



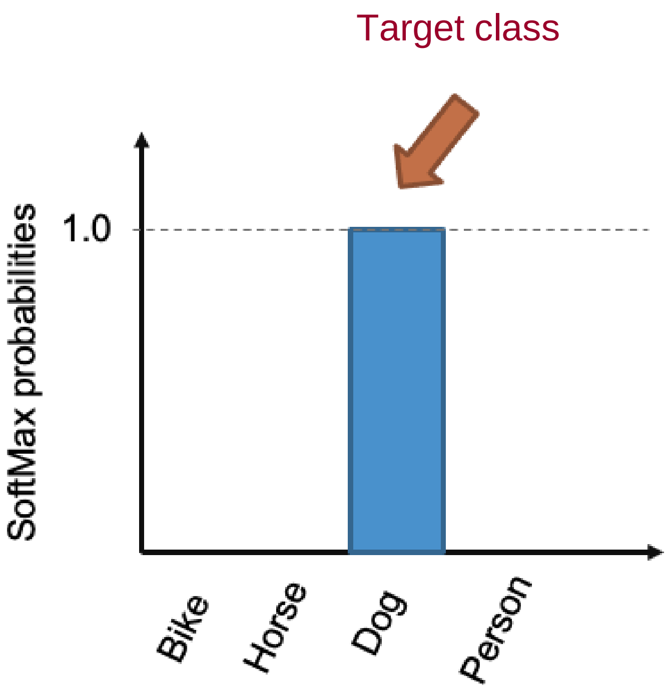
$$\mathbf{s} = [s_0, s_1, \ldots, s_{N-1}] \qquad \mathbf{y} = [y_0, y_1, \ldots, y_{N-1}]$$

Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.

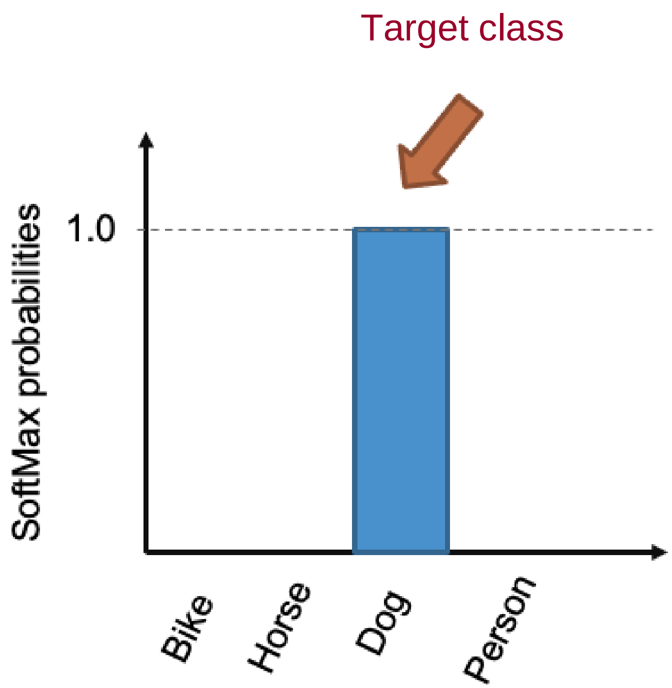


Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.



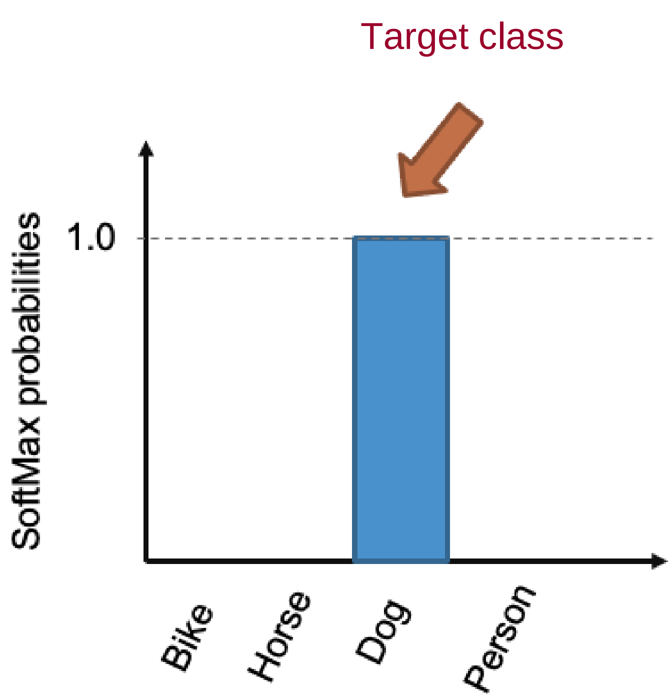
$$\mathcal{L}_{CE}(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k)$$

Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.



$$\mathcal{L}_{CE}(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k)$$

$$\mathcal{L}_{CE}(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i^0 \log \hat{y}_i^0 + y_i^1 \log \hat{y}_i^1 + y_i^2 \log \hat{y}_i^2 + y_i^3 \log \hat{y}_i^3)$$

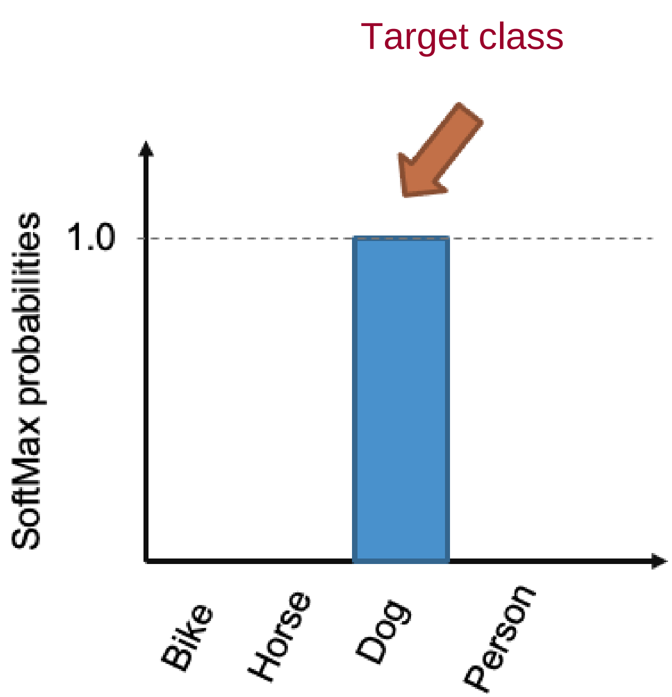
Assume
y=[0,1,0,0]

Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.



$$\mathcal{L}_{CE}(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k)$$

$$\mathcal{L}_{CE}(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i^0 \log \hat{y}_i^0 + y_i^1 \log \hat{y}_i^1 + y_i^2 \log \hat{y}_i^2 + y_i^3 \log \hat{y}_i^3)$$

Assume
y=[0,1,0,0]

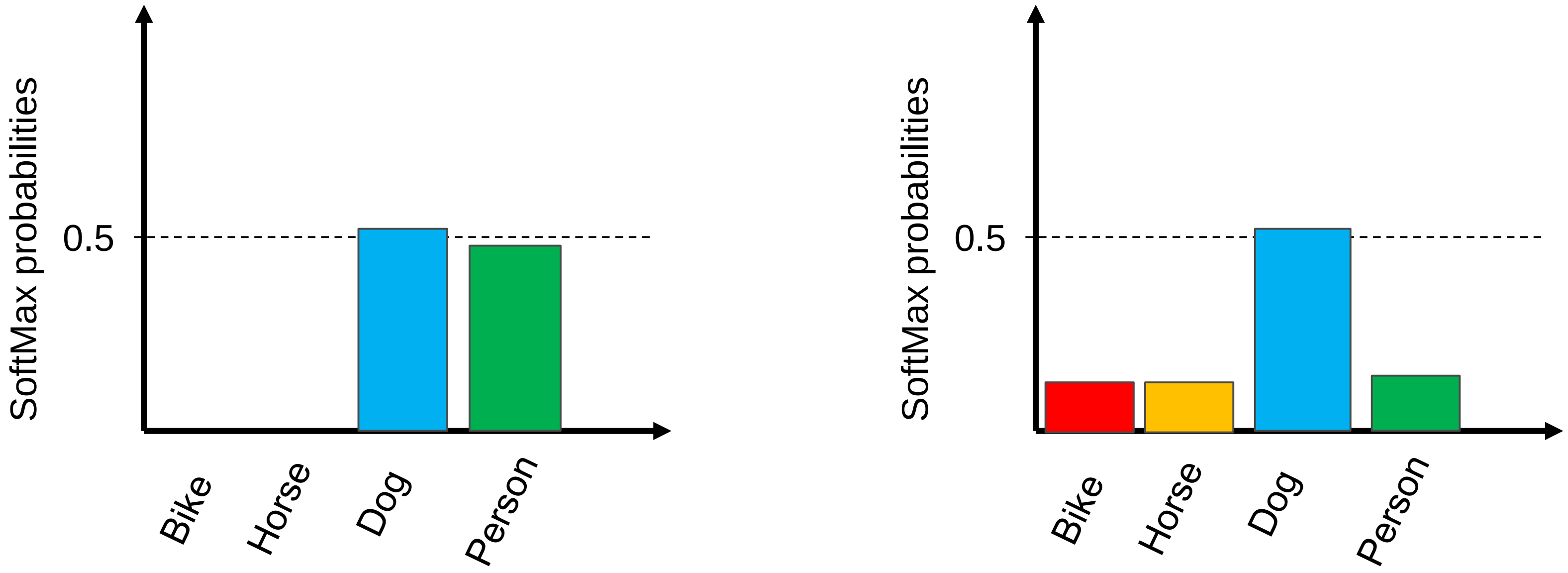
$$\mathcal{L}_{CE}(y,\hat{y}) = -(y_i^1 \log \hat{y}_i^1)$$

Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.



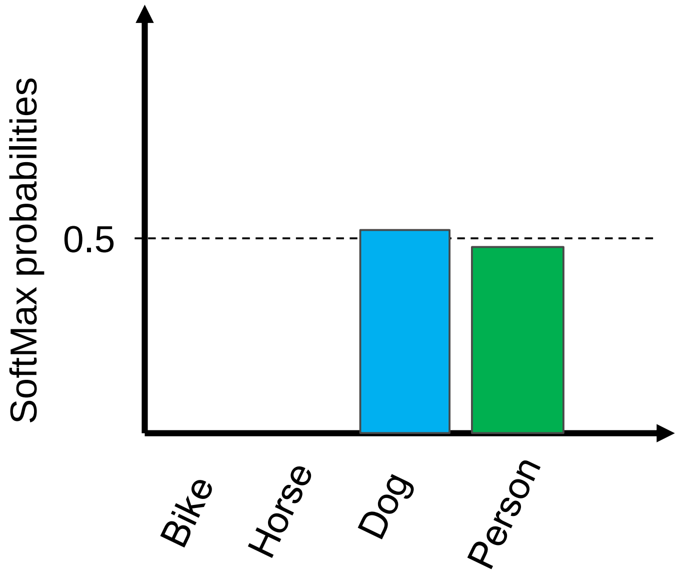
Which is the value of the CE in these two examples?

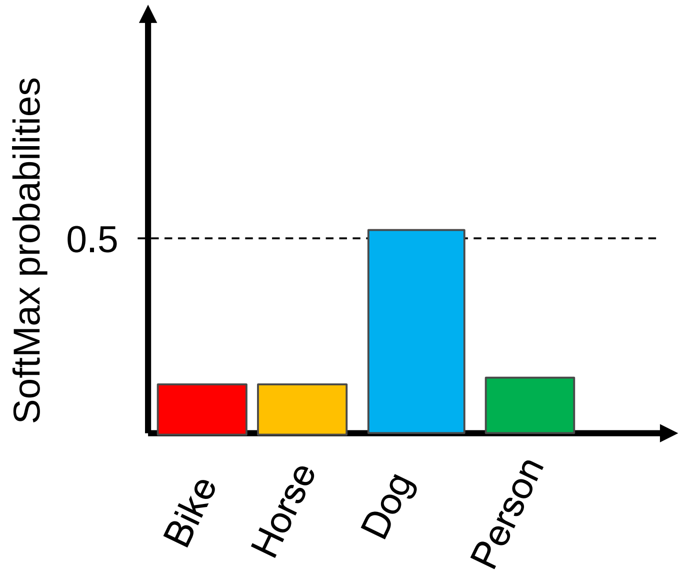
Problem

Standard loss functions

Cross-entropy

The supervision provided by cross-entropy is suboptimal for non-target classes in a multi-class scenario.





Exactly the same!!

Existing methods

Post-processing

Temperature Scaling
(Platts extension)

We change the distribution of the softmax vector

$$s_i = \frac{\exp(\hat{o}_i / T)}{\sum_{j=1}^{K} \exp(\hat{o}_j / T)}$$

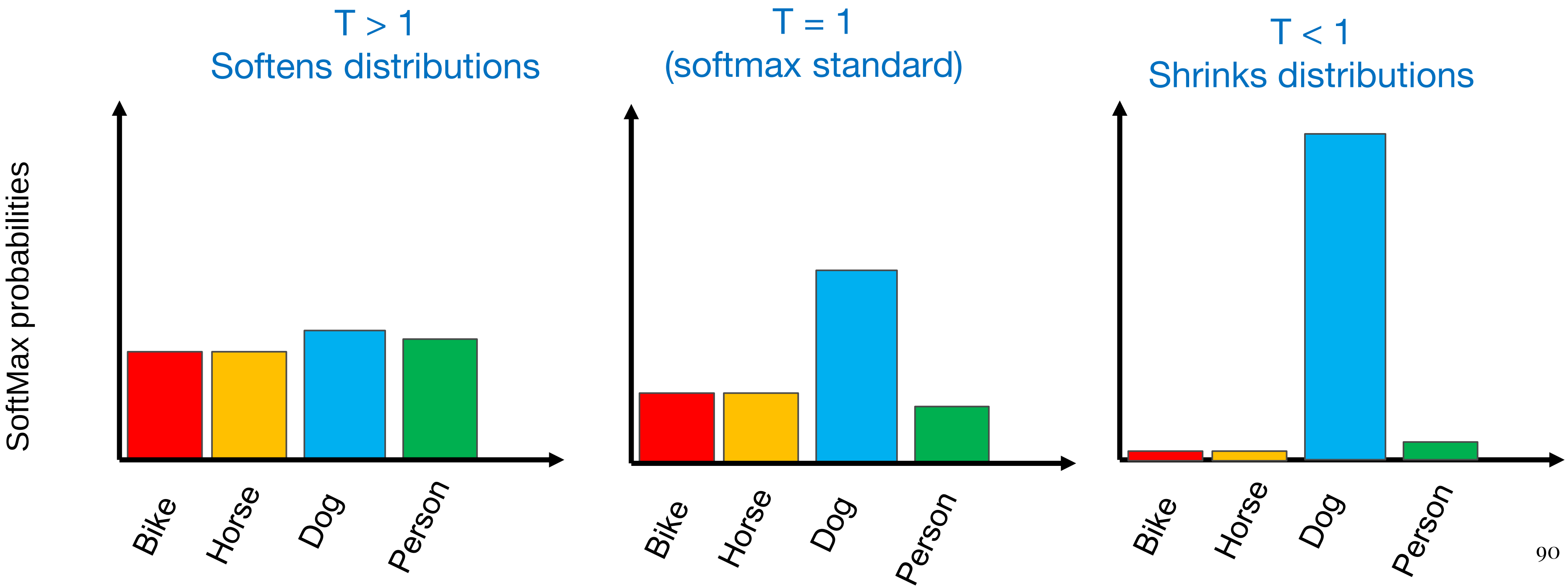
Existing methods

Post-processing

Temperature Scaling
(Platts extension)

We change the distribution of the softmax vector

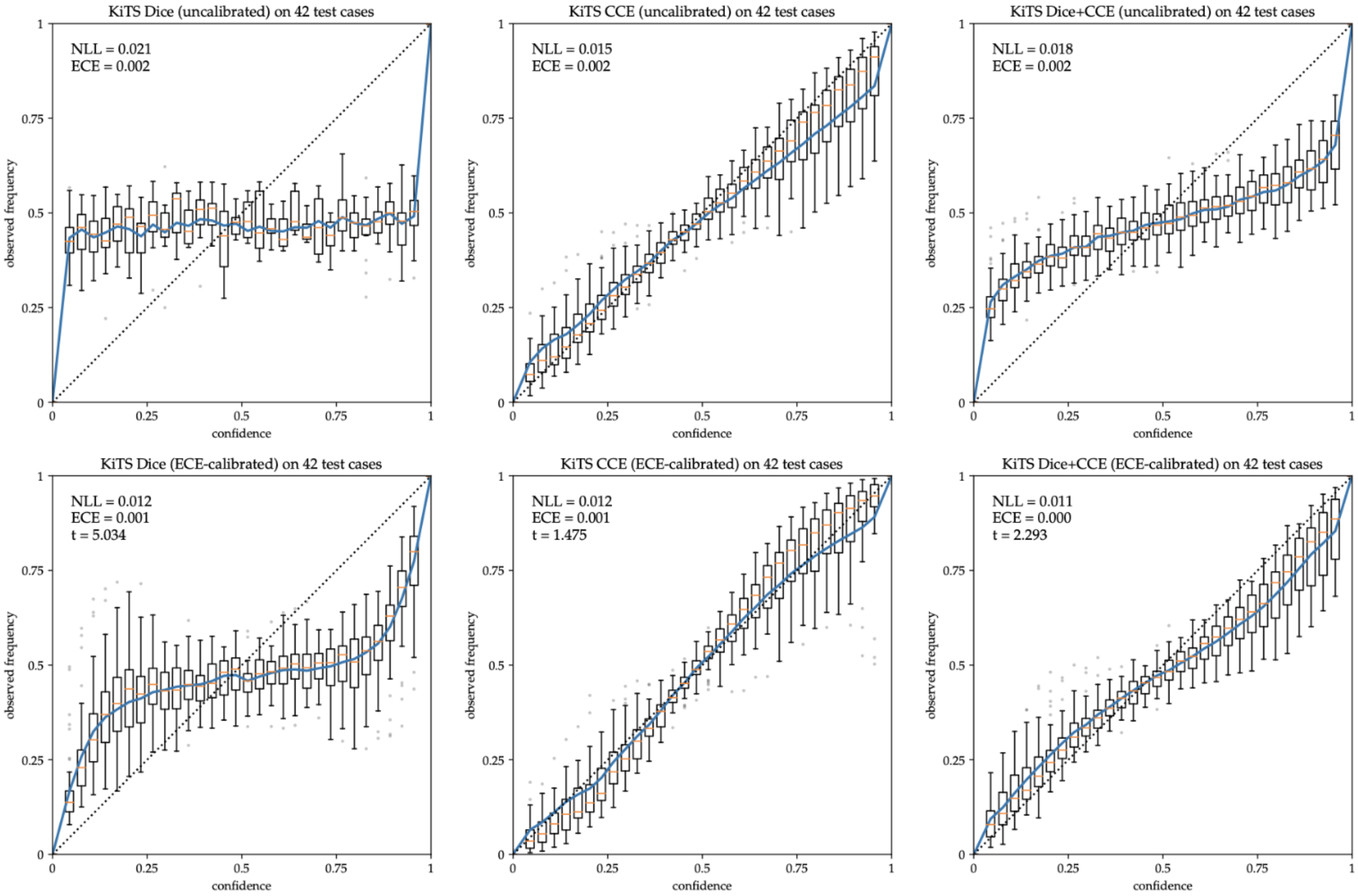
$$s_i = \frac{\exp(\hat{o}_i / T)}{\sum_{j=1}^{K} \exp(\hat{o}_j / T)}$$



Existing methods

Post-processing

Temperature Scaling
(Platts extension)



Uncalibrated

Calibrated

Kock et al. Confidence Histograms for Model Reliability Analysis and Temperature Calibration, MIDL'22

Existing methods

Post-processing

Temperature Scaling
(Platts extension)

We need an additional validation set to optimize T

Under distributional drift, Temperature scaling is suboptimal (Ovadia et al. NeurIPS’19)

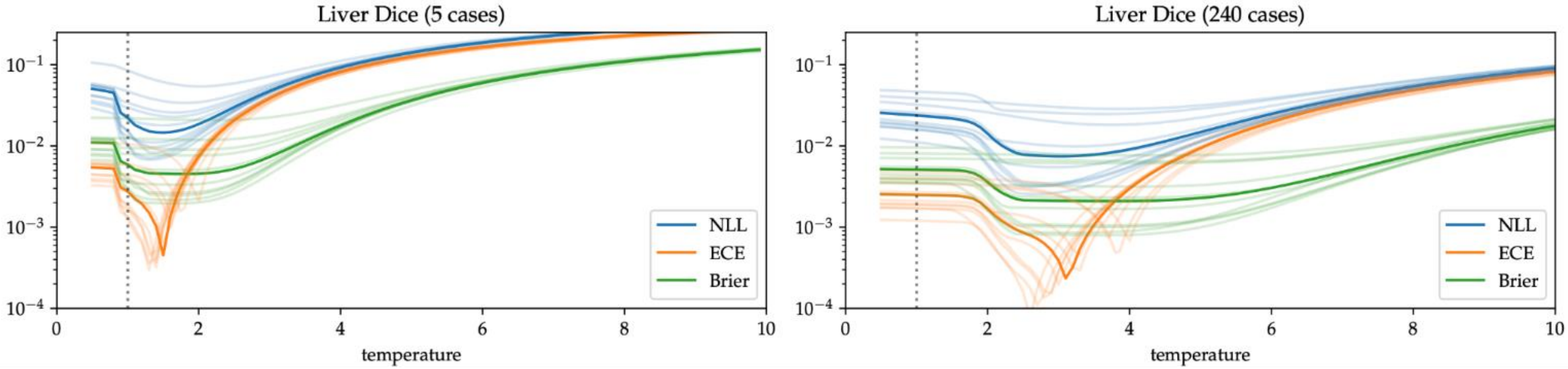
T value very sensitive to the dataset and network employed

Ovadia et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. NeurIPS’19

Existing methods

Post-processing

Temperature Scaling
(Platts extension)



Optimal T value also varies with the number of training samples!

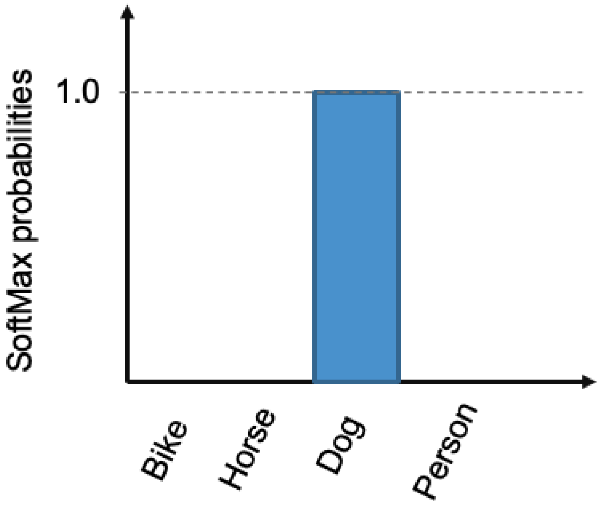
Kock et al. Confidence Histograms for Model Reliability Analysis and Temperature Calibration, MIDL’22

Existing methods

In the training (end-to-end)

Penalize high-entropies

High confident predictions (low-entropy) NOT DESIRED!



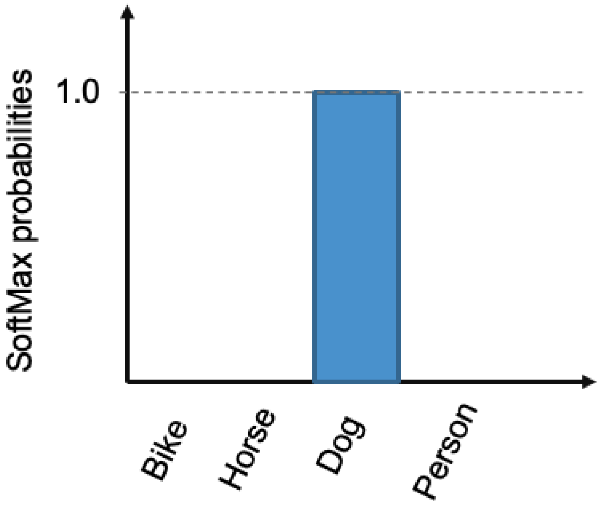
Pereyra et al., Regularizing Neural Networks by penalizing confident output distributions. ICLR 2017

Existing methods

In the training (end-to-end)

Penalize high-entropies

High confident predictions (low-entropy) NOT DESIRED!



For each example in the training set:

$$l(\mathbf{y}^i, \mathbf{s}_\theta^i) - \beta \mathcal{H}(\mathbf{s}_\theta^i)$$

Classification loss
(e.g., cross-entropy)

Entropy

Pereyra et al., Regularizing Neural Networks by penalizing confident output distributions. ICLR 2017

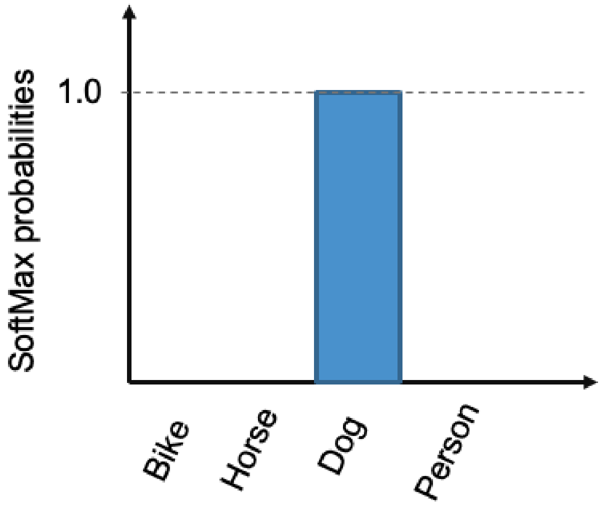
Existing methods

In the training (end-to-end)

Penalize high-entropies

High confident predictions (low-entropy) NOT DESIRED!

$$l(\mathbf{y}^i, \mathbf{s}_\theta^i) - \beta \mathcal{H}(\mathbf{s}_\theta^i)$$



Pereyra et al., Regularizing Neural Networks by penalizing confident output distributions. ICLR 2017

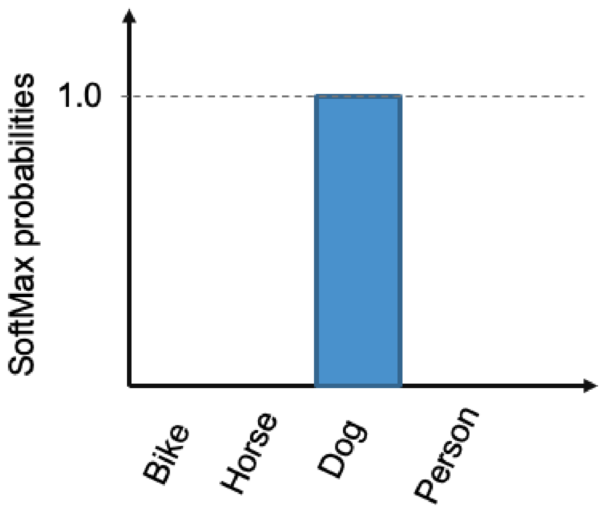
Existing methods

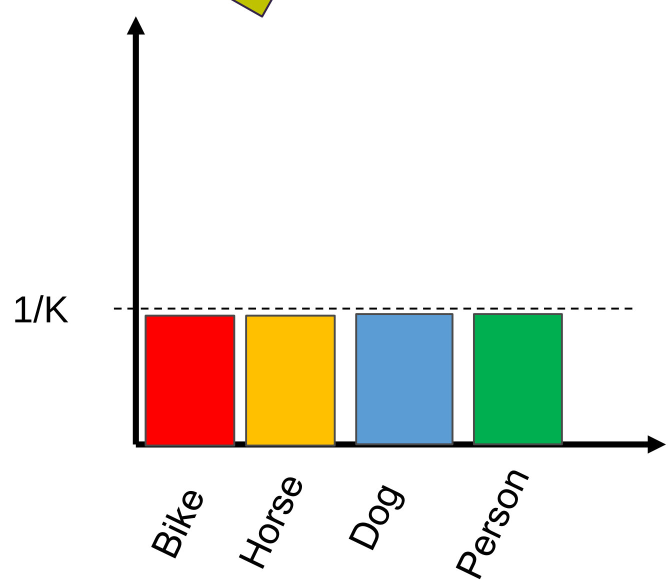
In the training (end-to-end)

Penalize high-entropies

High confident predictions (low-entropy) NOT DESIRED!

$$l(\mathbf{y}^i, \mathbf{s}_\theta^i) - \beta \mathcal{H}(\mathbf{s}_\theta^i)$$





Pereyra et al., Regularizing Neural Networks by penalizing confident output distributions. ICLR 2017

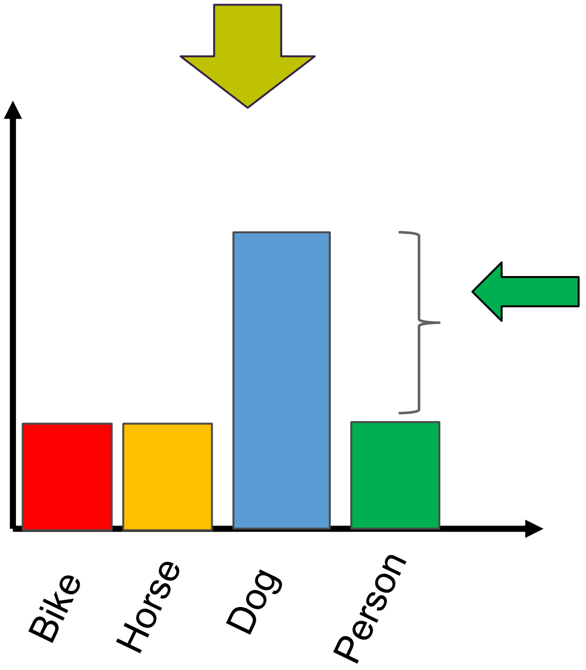
Existing methods

In the training (end-to-end)

Penalize high-entropies

High confident predictions (low-entropy) NOT DESIRED!

$$l(\mathbf{y}^i, \mathbf{s}_\theta^i) - \beta \mathcal{H}(\mathbf{s}_\theta^i)$$



The difference depends on β

Pereyra et al., Regularizing Neural Networks by penalizing confident output distributions. ICLR 2017

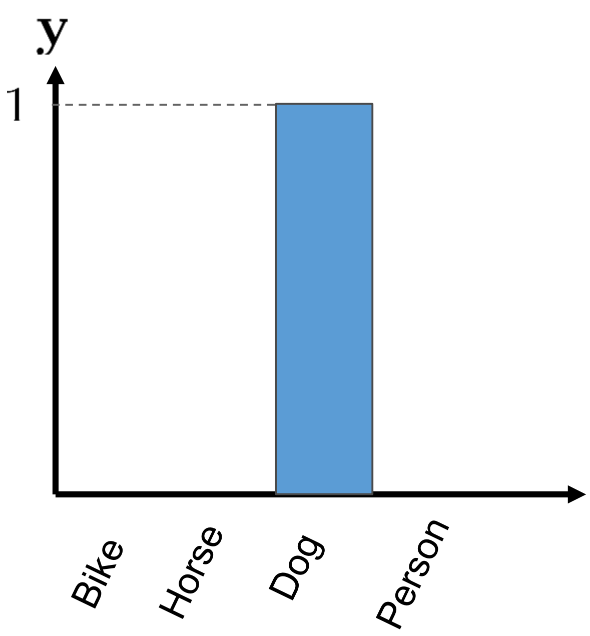
Existing methods

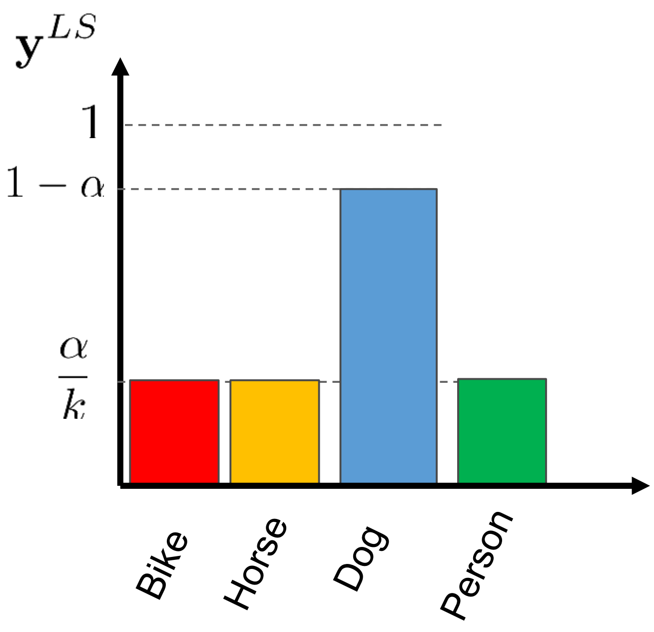
In the training (end-to-end)

Label Smoothing

$$y_k^{\mathrm{LS}} = y_k(1-\alpha) + \frac{\alpha}{K}$$

One-hot encoding



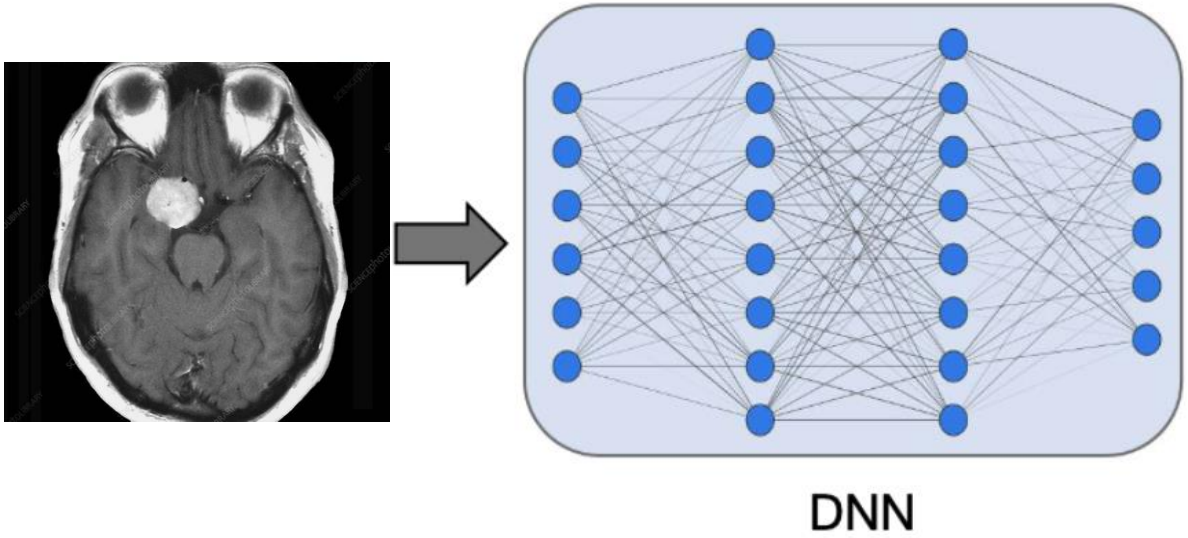


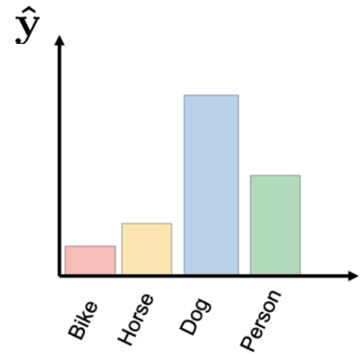
Existing methods

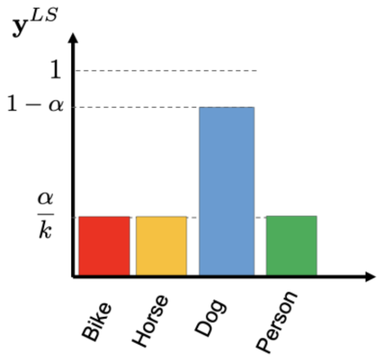
In the training (end-to-end)

Label Smoothing

$$y_k^{\mathrm{LS}} = y_k(1-\alpha) + \frac{\alpha}{K}$$







Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_{k}^{K} y_k^{LS} \log(s_k) = -\sum_{k}^{K}((1-\alpha)y_k + \frac{\alpha}{K}) \log(s_k)$$

Müller et al., When does label smoothing help. NeurIPS’19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_{k}^{K} y_k^{LS} \log(s_k) = -\sum_{k}^{K} \left((1-\alpha)y_k + \frac{\alpha}{K}\right) \log(s_k)$$

$$-\sum_{k}^{K} ((1-\alpha)y_k) \log(s_k) - \sum_{k}^{K} \frac{\alpha}{K} \log(s_k)$$

Müller et al., When does label smoothing help. NeurIPS'19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_{k}^{K} y_k^{LS} \log(s_k) = -\sum_{k}^{K} ((1-\alpha) y_k + \frac{\alpha}{K}) \log(s_k)$$

$$-\sum_{k}^{K} ((1-\alpha) y_k) \log(s_k) - \sum_{k}^{K} \frac{\alpha}{K} \log(s_k)$$

$$-\sum_{k}^{K} y_k \log(s_k) - \frac{\alpha}{(1-\alpha)} \sum_{k}^{K} \frac{1}{K} \log(s_k)$$

Müller et al., When does label smoothing help. NeurIPS'19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_{k}^{K} y_k^{LS} \log(s_k) = -\sum_{k}^{K} ((1-\alpha) y_k + \frac{\alpha}{K}) \log(s_k)$$

$$-\sum_{k}^{K} ((1-\alpha) y_k) \log(s_k) - \sum_{k}^{K} \frac{\alpha}{K} \log(s_k)$$

$$-\sum_{k}^{K} y_k \log(s_k) - \frac{\alpha}{(1-\alpha)} \sum_{k}^{K} \frac{1}{K} \log(s_k)$$

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) + \frac{\alpha}{1-\alpha} \mathcal{H}(\frac{1}{K}, \mathbf{s})$$

Cross-entropy between **y** and **s**

Müller et al., When does label smoothing help. NeurIPS'19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_k^K y_k^{LS} \log(s_k) = -\sum_k^K ((1-\alpha) y_k + \frac{\alpha}{K}) \log(s_k)$$

$$-\sum_k^K ((1-\alpha) y_k) \log(s_k) - \sum_k^K \frac{\alpha}{K} \log(s_k)$$

$$-\sum_k^K y_k \log(s_k) - \frac{\alpha}{(1-\alpha)} \sum_k^K \frac{1}{K} \log(s_k)$$

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) + \frac{\alpha}{1-\alpha} \mathcal{H}(\frac{1}{K}, \mathbf{s})$$

Cross-entropy between **1/K** and **s**

Müller et al., When does label smoothing help. NeurIPS’19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}', \mathbf{s}) = -\sum_{k}^{K} y_k^{LS} \log(s_k) = -\sum_{k}^{K}((1-\alpha)y_k + \frac{\alpha}{K}) \log(s_k)$$

$$-\sum_{k}^{K}((1-\alpha)y_k) \log(s_k) - \sum_{k}^{K} \frac{\alpha}{K} \log(s_k)$$

$$-\sum_{k}^{K} y_k \log(s_k) - \frac{\alpha}{(1-\alpha)} \sum_{k}^{K} \frac{1}{K} \log(s_k)$$

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) + \frac{\alpha}{1-\alpha} \mathcal{H}(\frac{1}{K}, \mathbf{s})$$

This measures the similarity of **s** wrt to the uniform distribution **1/K**

Cross-entropy between **1/K** and **s**

Müller et al., When does label smoothing help. NeurIPS'19

Existing methods

In the training (end-to-end)

Label Smoothing

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) + \frac{\alpha}{1-\alpha}\mathcal{H}(\frac{1}{K}, \mathbf{s})$$

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) - \frac{\alpha}{1-\alpha}\mathcal{H}(\mathbf{s})$$

We can replace the second term with an entropy maximization objective

Maximizes the entropy of **s**

Müller et al., When does label smoothing help. NeurIPS'19

Existing methods

In the training (end-to-end)

Focal Loss

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

Lin et al., Focal loss for dense object detection. ICCV'17

Existing methods

In the training (end-to-end)

Focal Loss

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

We introduce p_t

Lin et al., Focal loss for dense object detection. ICCV'17

Existing methods

In the training (end-to-end)

Focal Loss

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

We introduce p_t

We rewrite CE as

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_{\mathrm{t}}) = -\log(p_{\mathrm{t}})$$

Lin et al., Focal loss for dense object detection. ICCV’17

Existing methods

In the training (end-to-end)

Focal Loss

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

We introduce p_t

We rewrite CE as

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_{\mathrm{t}}) = -\log(p_{\mathrm{t}})$$

We add an additional term

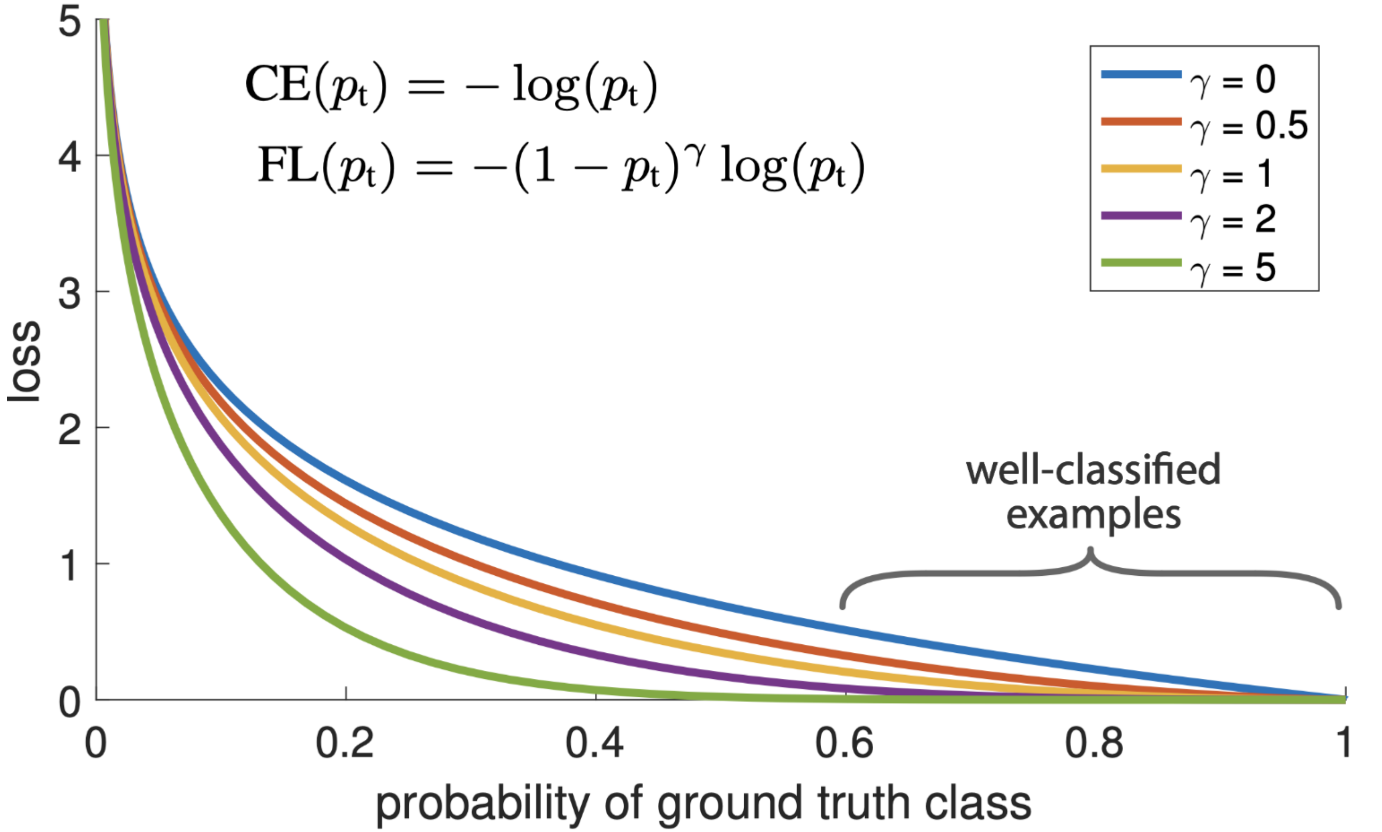
$$\mathrm{FL}(p_{\mathrm{t}}) = -(1-p_{\mathrm{t}})^{\gamma} \log(p_{\mathrm{t}})$$

Lin et al., Focal loss for dense object detection. ICCV'17

Existing methods

In the training (end-to-end)

Focal Loss



Lin et al., Focal loss for dense object detection. ICCV'17

Existing methods

In the training (end-to-end)

Focal Loss

$$\mathcal{L}_{FL} \geq \mathcal{D}_{KL}(\mathbf{y}||\mathbf{s}) - \gamma\mathcal{H}(\mathbf{s})$$

Mukhoti et al., Calibrating deep neural networks using focal loss. NeurIPS’20

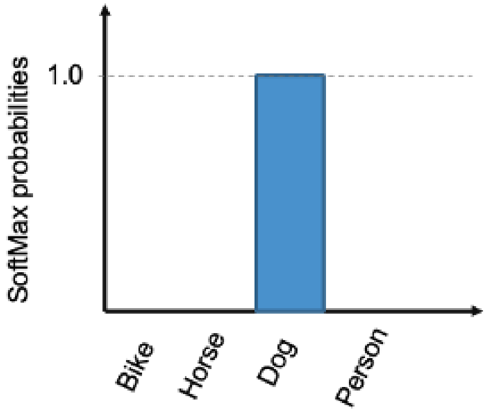
Existing methods

In the training (end-to-end)

Focal Loss

$$\mathcal{L}_{FL} \geq \mathcal{D}_{KL}(\mathbf{y}||\mathbf{s}) - \gamma\mathcal{H}(\mathbf{s})$$

Minimize the differences between network predictions and ground truth (like CE)



Mukhoti et al., Calibrating deep neural networks using focal loss. NeurIPS'20

Existing methods

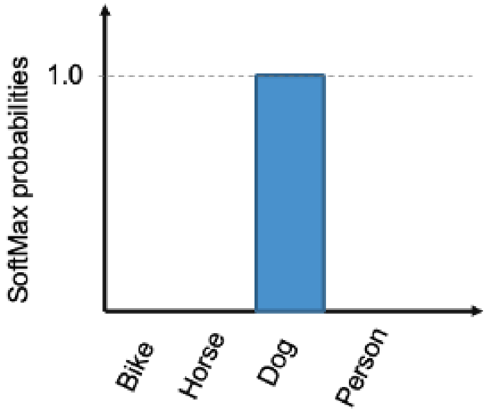
In the training (end-to-end)

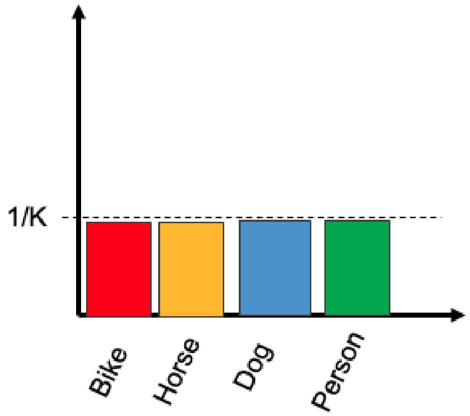
Focal Loss

$$\mathcal{L}_{FL} \geq \mathcal{D}_{KL}(\mathbf{y}||\mathbf{s}) - \gamma\mathcal{H}(\mathbf{s})$$

Minimize the differences between network predictions and ground truth (like CE)

Maximize the entropy (i.e., pushing uniform distributions)





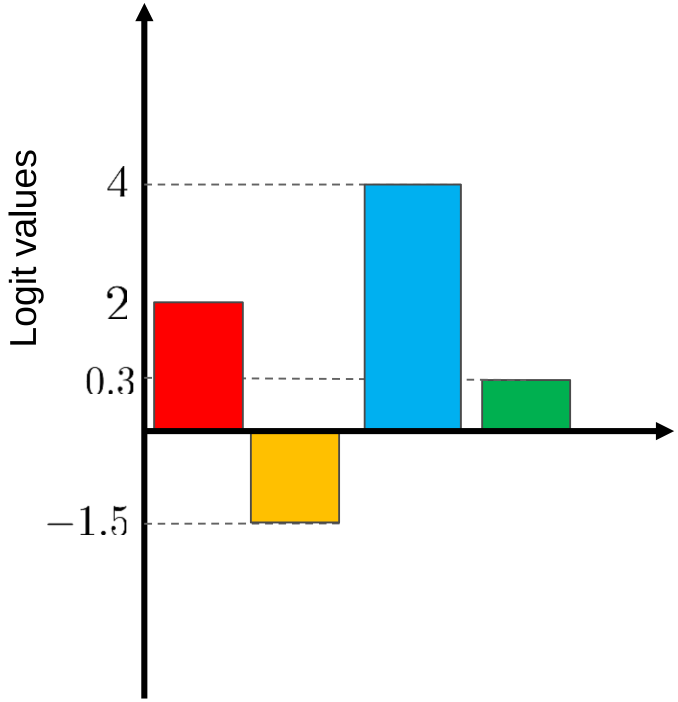
Mukhoti et al., Calibrating deep neural networks using focal loss. NeurIPS'20

Existing methods

In the training (end-to-end)

Margin-based LS

$$\mathbf{d}(\mathbf{l}) = (\max_j(l_j) - l_k)_{1 \leq k \leq K} \in \mathbb{R}^K$$



$$\mathbf{d}(\mathbf{l}) = [2, 5.5, 0, 3.7]$$

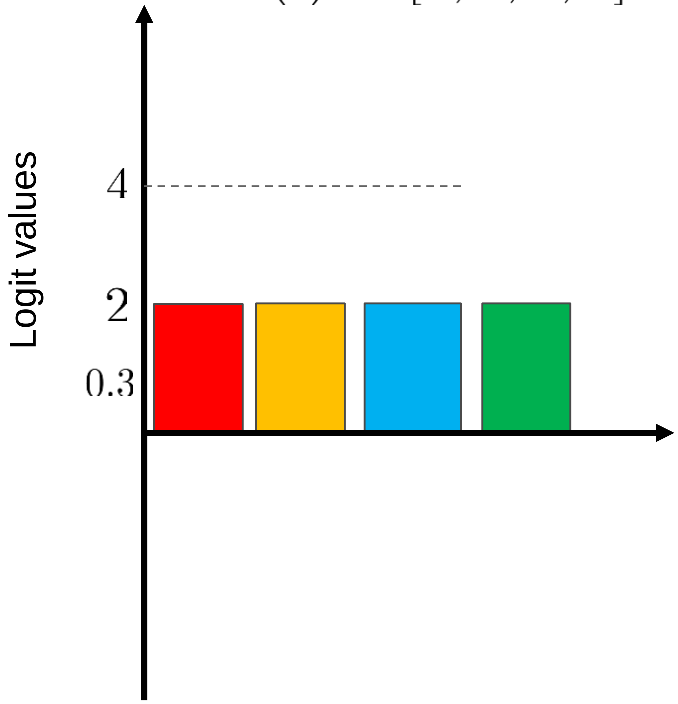
Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

Existing methods

In the training (end-to-end)

Maximizing the entropy can be seen as a constraint that forces the distance vector **d(l) to be zero.**

$$\mathbf{d}(\mathbf{l}) = [0, 0, 0, 0] = \mathbf{0}$$



Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) - \mathcal{H}(\mathbf{s})$$

Pereyra, ICLR'17
Müller, NeurIPS'19
Mukhoti, NeurIPS'20

$$\boxed{\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) = \mathbf{0}}$$

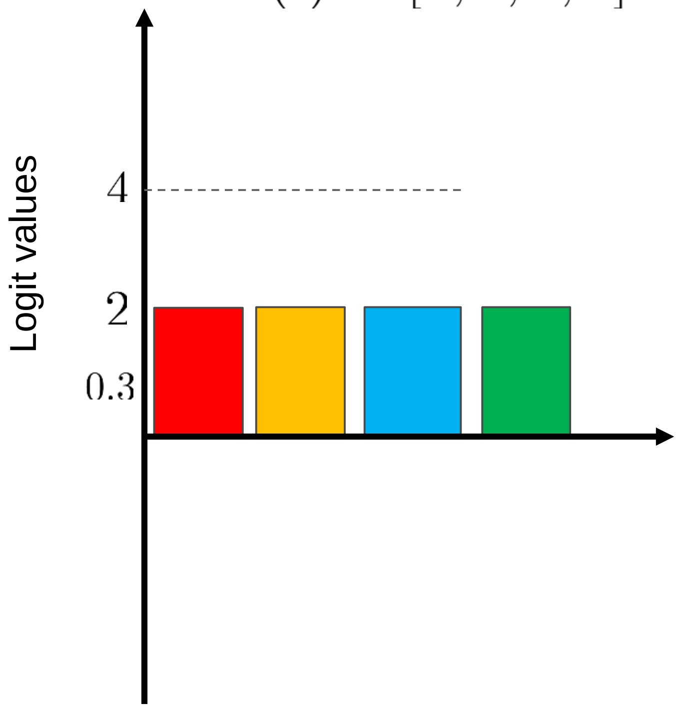
Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

Existing methods

In the training (end-to-end)

Maximizing the entropy can be seen as a constraint that forces the distance vector **d(l) to be zero.**

$$\mathbf{d}(\mathbf{l}) = [0, 0, 0, 0] = \mathbf{0}$$

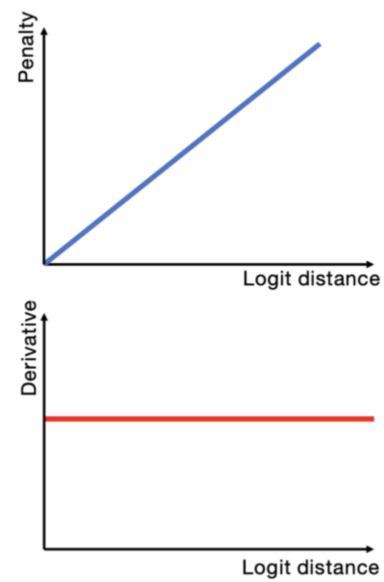


Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) - \mathcal{H}(\mathbf{s})$$

$$\boxed{\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) = \mathbf{0}}$$

Pereyra, ICLR'17
Müller, NeurIPS'19
Mukhoti, NeurIPS'20



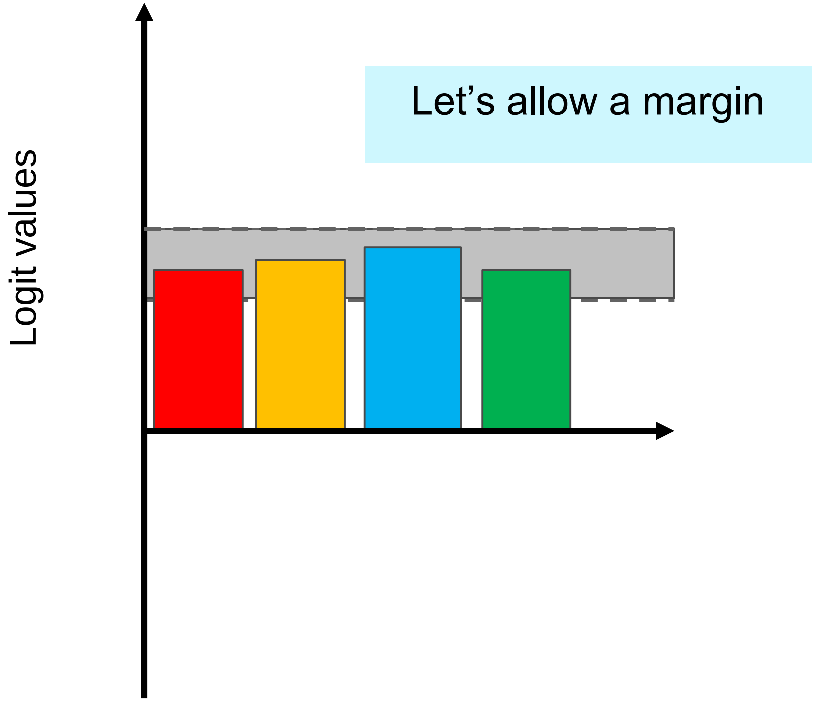
Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

Existing methods

In the training (end-to-end)

Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) \leq \mathbf{m}$$



Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

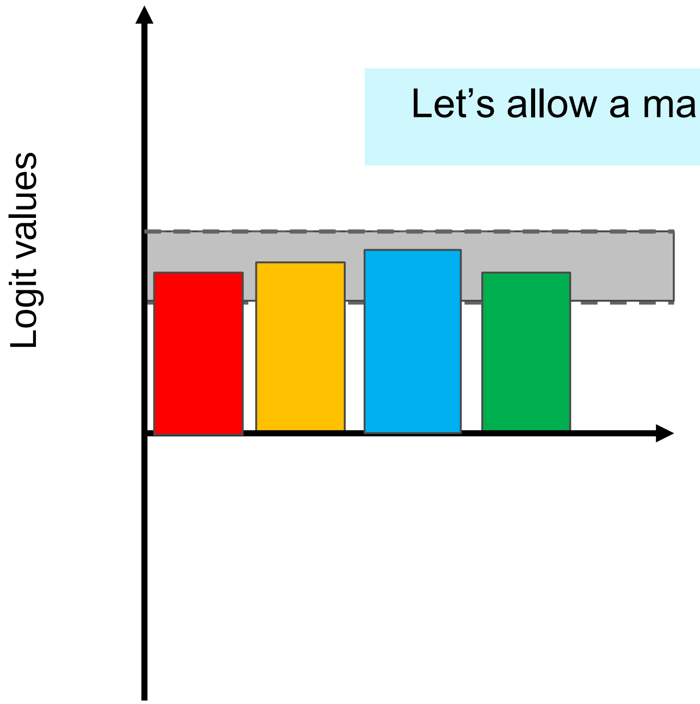
Existing methods

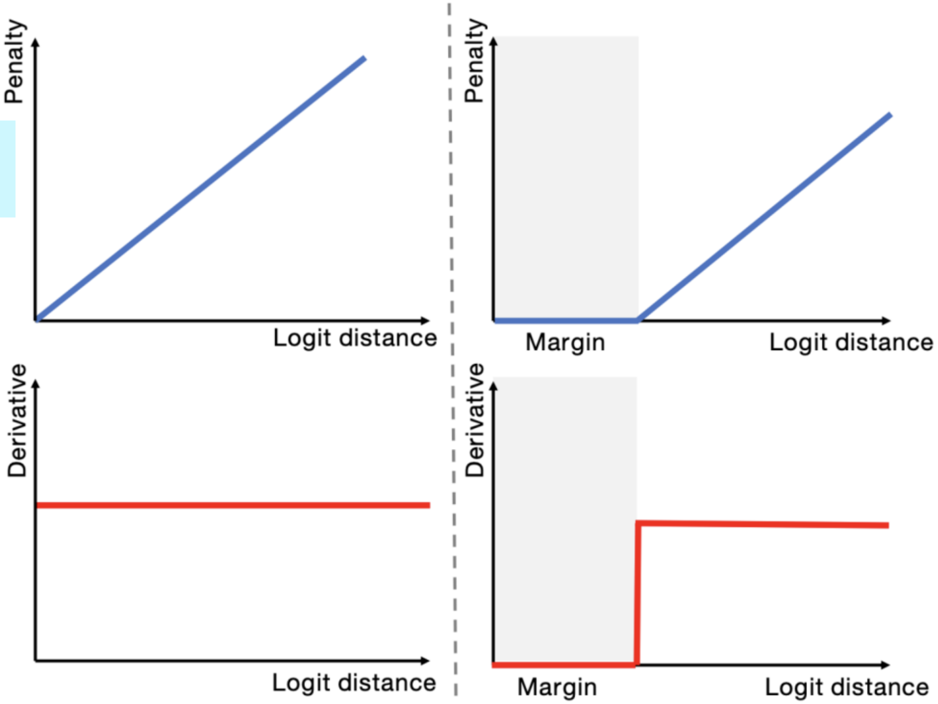
In the training (end-to-end)

Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) \leq \mathbf{m}$$

Let's allow a margin





Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

Existing methods

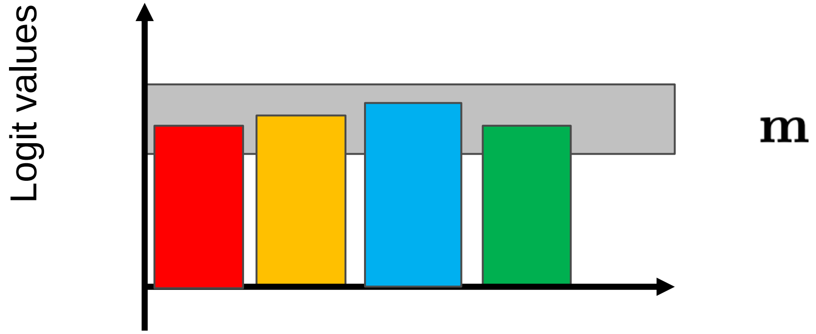
In the training (end-to-end)

Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) \leq \mathbf{m}$$

$$\min \quad \mathcal{L}_{\text{CE}} + \lambda \sum_k \max(0, \max_j(l_j) - l_k - m)$$

$$\max_j l_j - l_k \leq m$$



Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR’22

Existing methods

In the training (end-to-end)

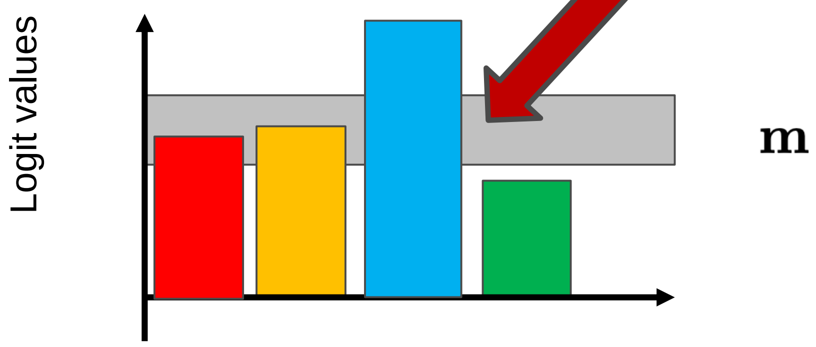
Margin-based LS

$$\mathcal{H}(\mathbf{y}, \mathbf{s}) \quad \text{s.t.} \quad \mathbf{d}(\mathbf{l}) \leq \mathbf{m}$$

This will violate the constraint, and there will be a penalty

$$\min \quad \mathcal{L}_{\text{CE}} + \lambda \sum_k \max(0, \max_j(l_j) - l_k - m)$$

$$\max_j l_j - l_k > m$$

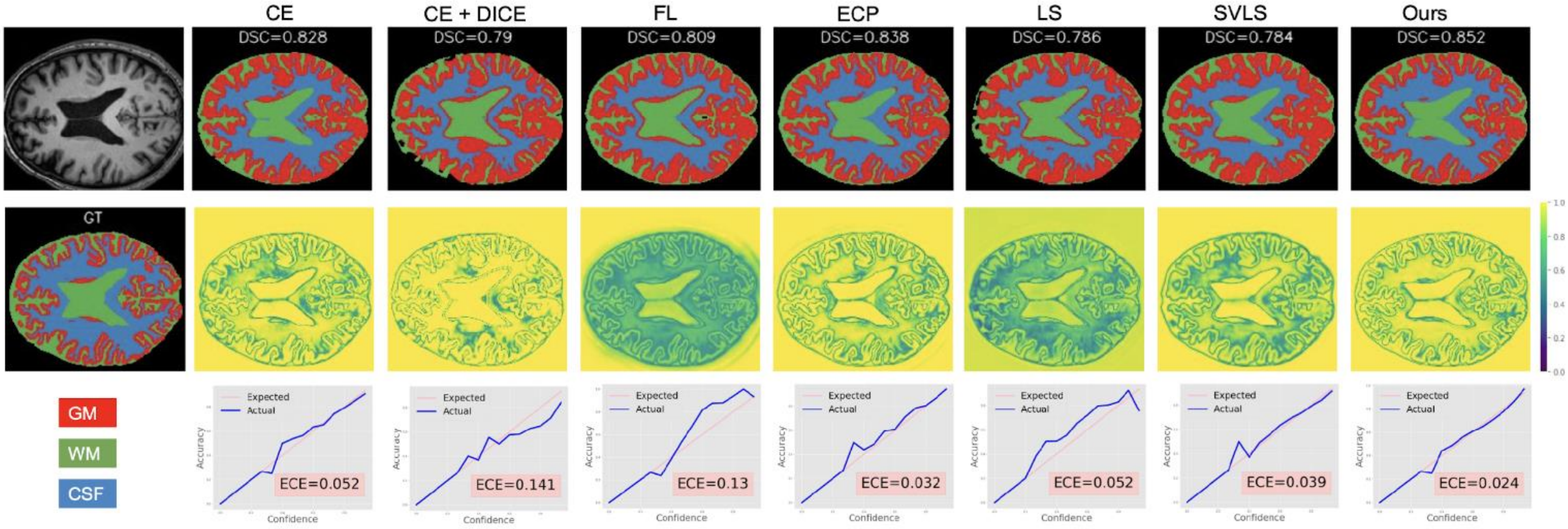


Liu et al., The devil is in the margin: Margin-based label smoothing for network calibration. CVPR'22

Existing methods

In the training (end-to-end)

Margin-based LS

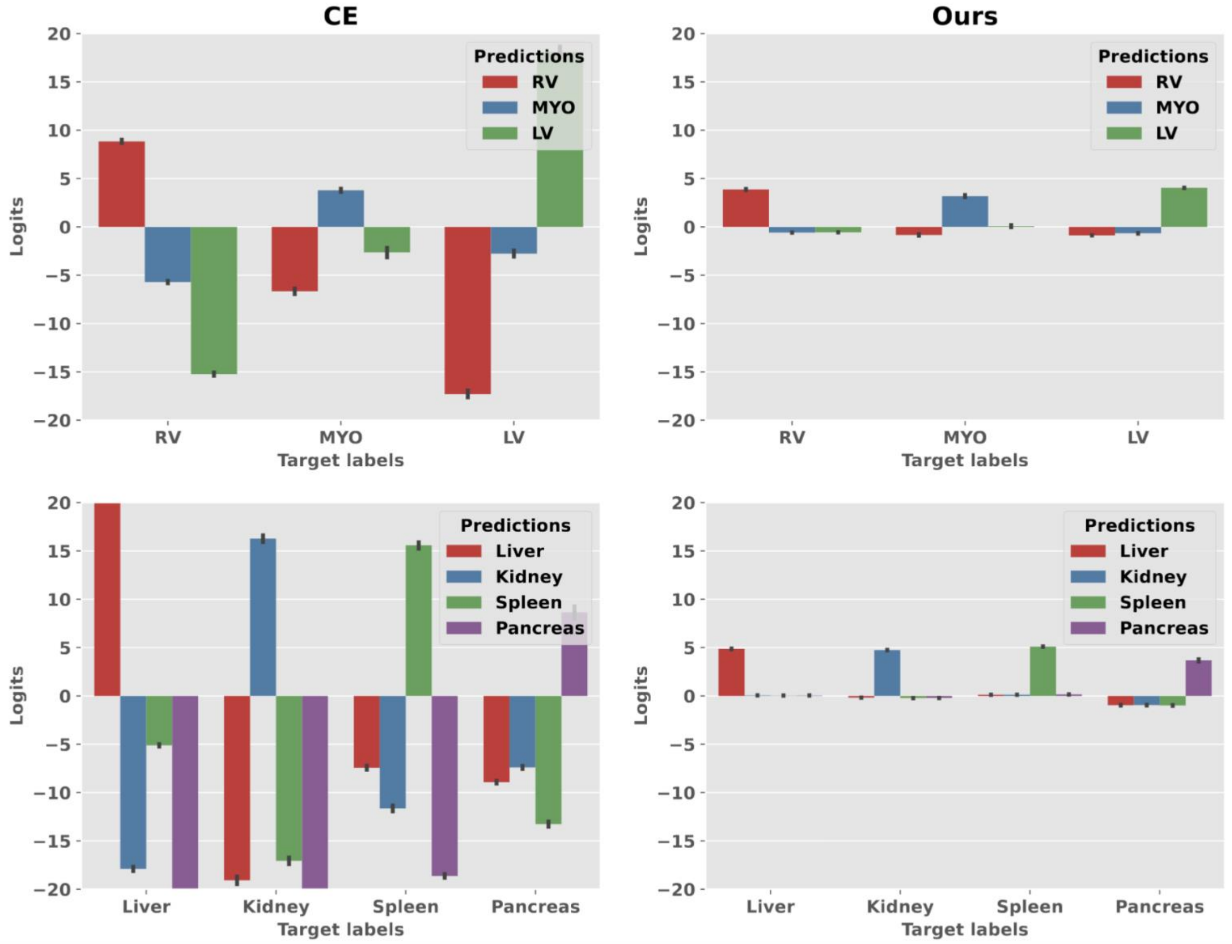


Murugesan et al., Calibrating Segmentation Networks with Margin-based Label Smoothing. MedIA'23

Existing methods

In the training (end-to-end)

Margin-based LS

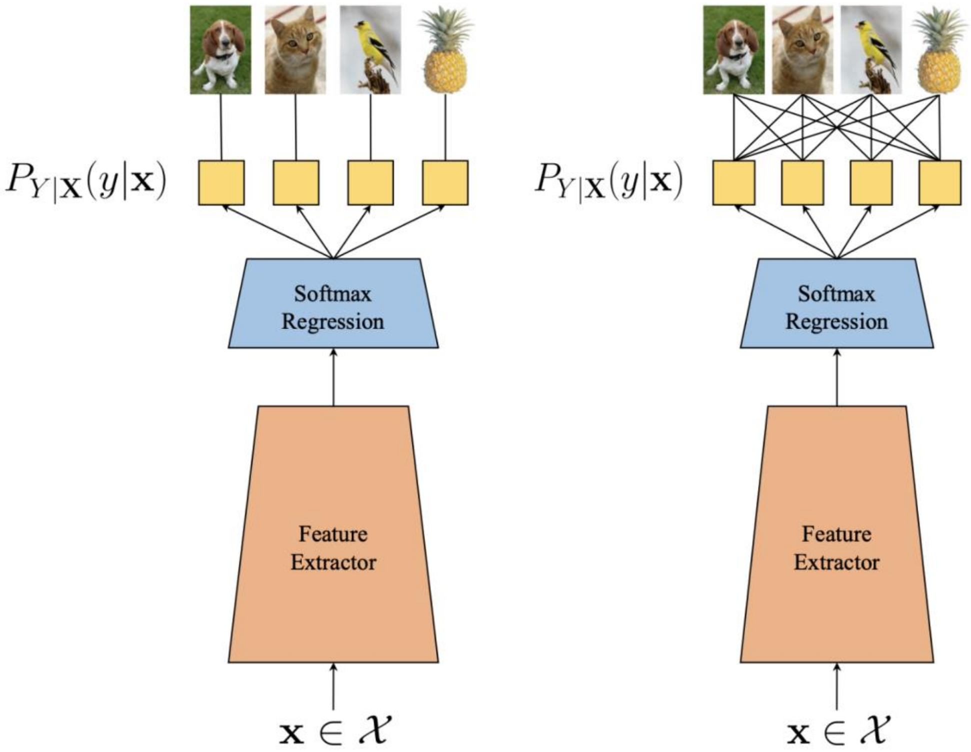


Murugesan et al., Calibrating Segmentation Networks with Margin-based Label Smoothing. MedIA’23

Existing methods

In the training (end-to-end)

Pairwise contraints

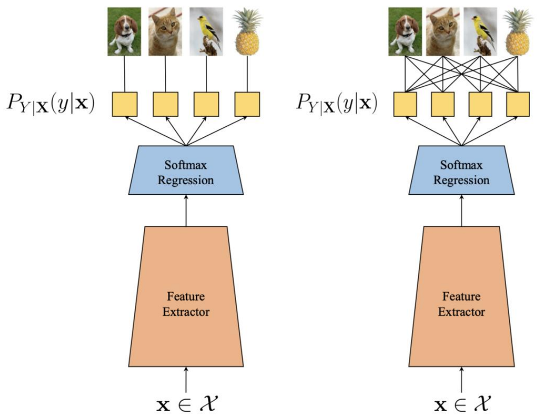


Cheng et al., Calibrating Deep Neural Networks by Pairwise Constraints. CVPR’22

Existing methods

In the training (end-to-end)

Pairwise contraints



Posterior probability

$$\beta_{ij} = \frac{\pi_i}{\pi_i + \pi_j}$$

For the pairs containing the target class:

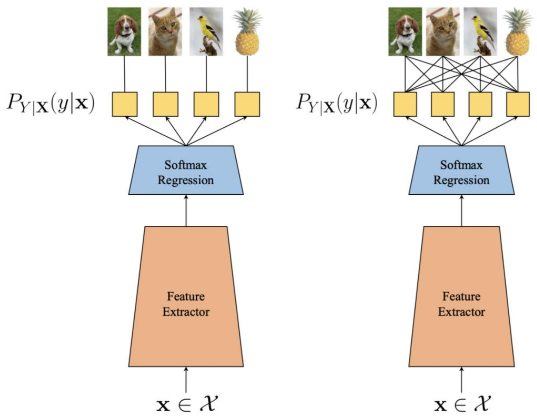
$$\mathcal{L}_{ij}^{1v1}(\mathbf{x}, y; \Theta) = -\mathbb{1}_{y=i} \log \beta_{ij}(\mathbf{x}) - \mathbb{1}_{y=j} \log \beta_{ji}(\mathbf{x}).$$

Cheng et al., Calibrating Deep Neural Networks by Pairwise Constraints. CVPR'22

Existing methods

In the training (end-to-end)

Pairwise contraints



Posterior probability

$$\beta_{ij} = \frac{\pi_i}{\pi_i + \pi_j}$$

For the pairs containing the target class:

$$\mathcal{L}_{ij}^{1v1}(\mathbf{x}, y; \Theta) = -\mathbb{1}_{y=i} \log \beta_{ij}(\mathbf{x}) - \mathbb{1}_{y=j} \log \beta_{ji}(\mathbf{x}).$$

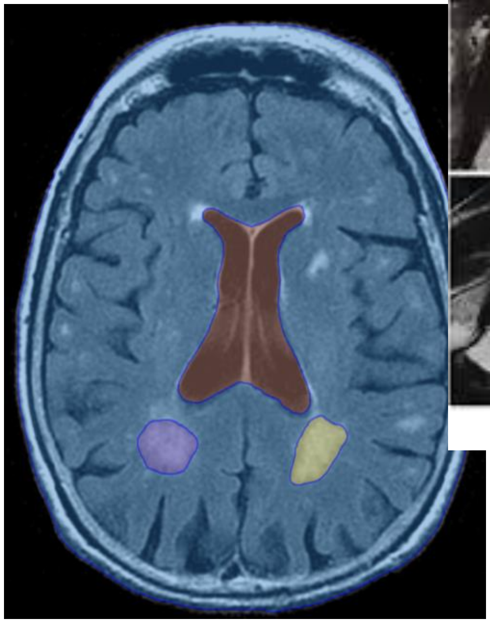
For the pairs that DO NOT contain the target class:

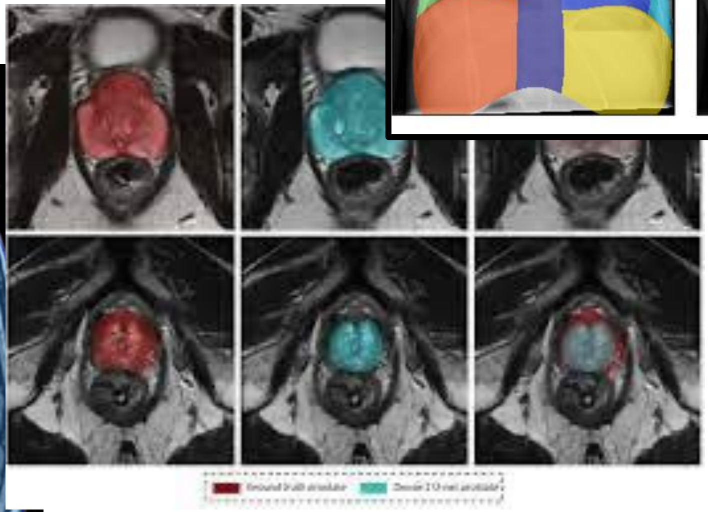
$$\beta_i^*(\mathbf{x}) = \beta_j^*(\mathbf{x}) = 1/2.$$

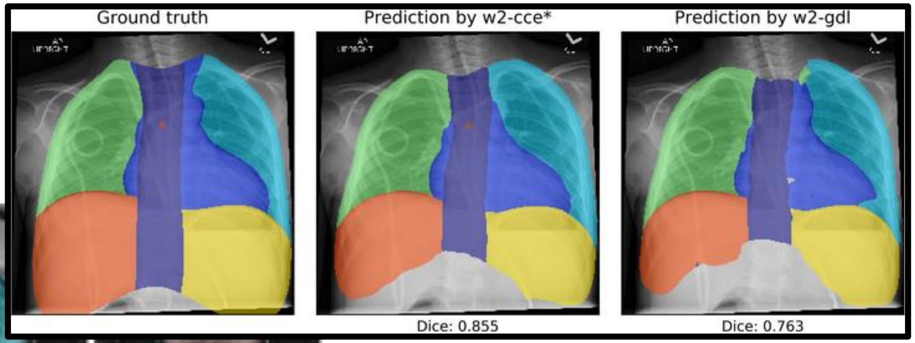
Is this familiar to you??

Cheng et al., Calibrating Deep Neural Networks by Pairwise Constraints. CVPR’22

Designed for the segmentation task



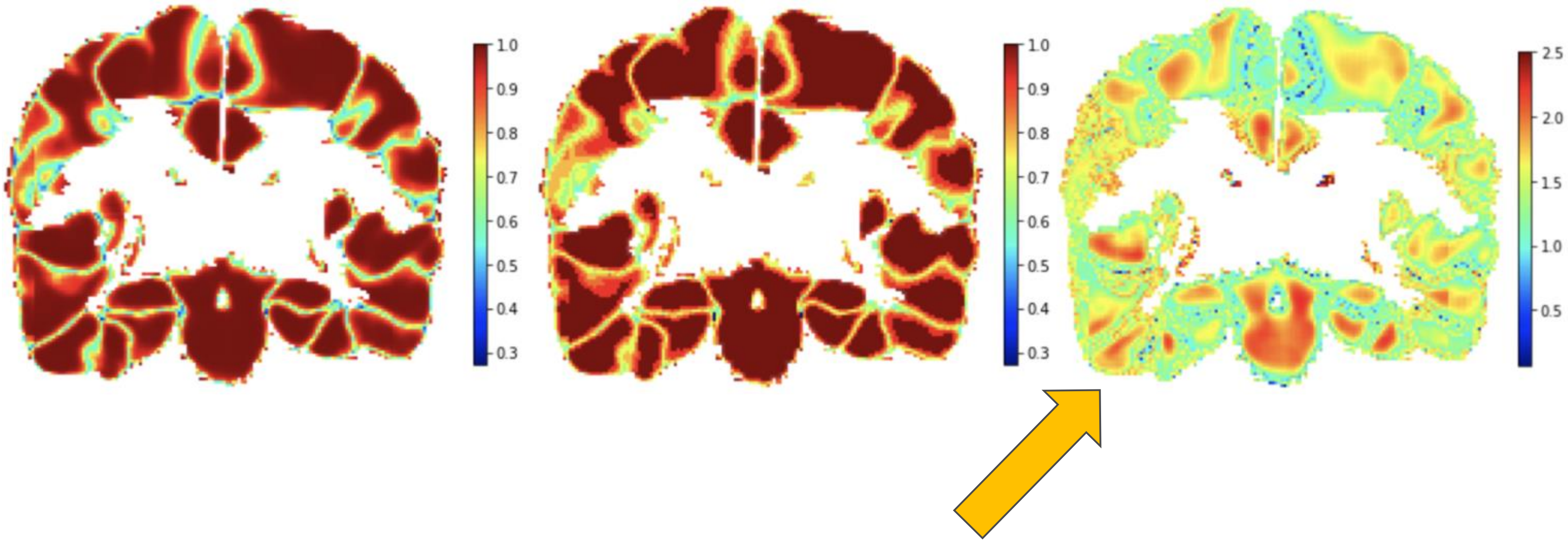




Existing methods

Post-processing

Local TS



T value (*Temperature Scaling*) is not the same for all the pixels

Ding et al., Local Temperature Scaling for Probability Calibration. ICCV'21

Existing methods

Post-processing

Local TS

$$T^* = \arg\min_T \left(-\sum_{i=1}^{n} \sum_{x \in \Omega} \log \left(\sigma_{SM} \left(\mathbf{z}_i(x) / T \right)^{(S_i(x))} \right) \right)$$

$$s.t. \quad T > 0,$$

In standard *Temperature Scaling*, T value is the same

Ding et al., Local Temperature Scaling for Probability Calibration. ICCV’21

Existing methods

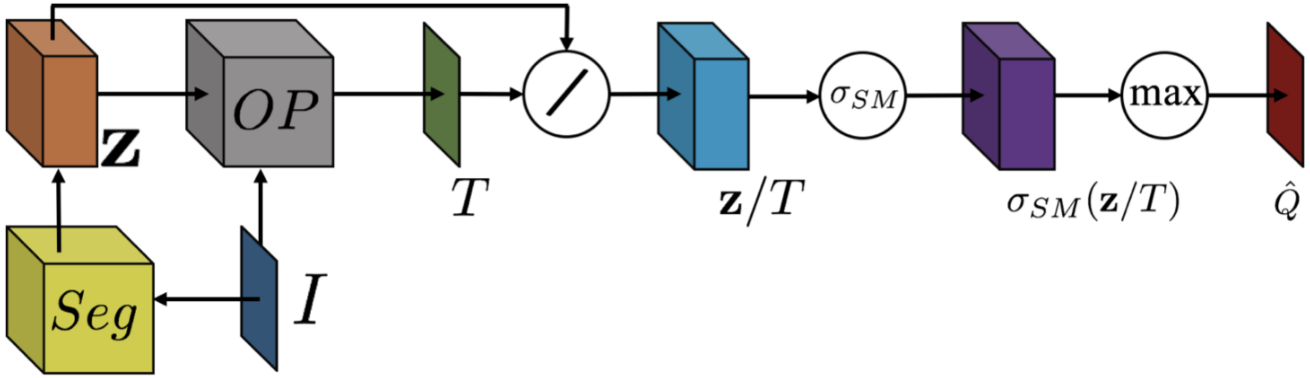
Post-processing

Local TS

$$T^* = \arg\min_T \left(- \sum_{i=1}^{n} \sum_{x \in \Omega} \log \left(\sigma_{SM} \left(\mathbf{z}_i(x)/T \right)^{(S_i(x))} \right) \right)$$

$$s.t. \quad T > 0,$$

In standard *Temperature Scaling*, T value is the same



Ding et al., Local Temperature Scaling for Probability Calibration. ICCV'21

Existing methods

In the training (end-to-end)

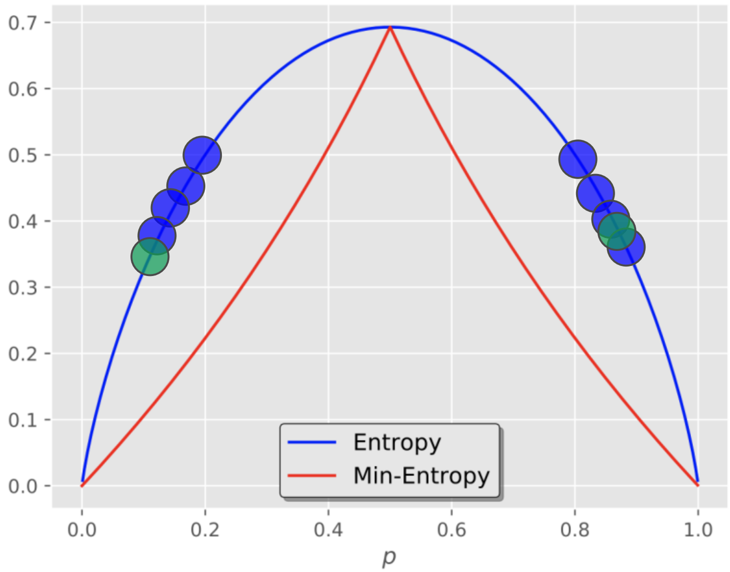
MEEP

Key idea

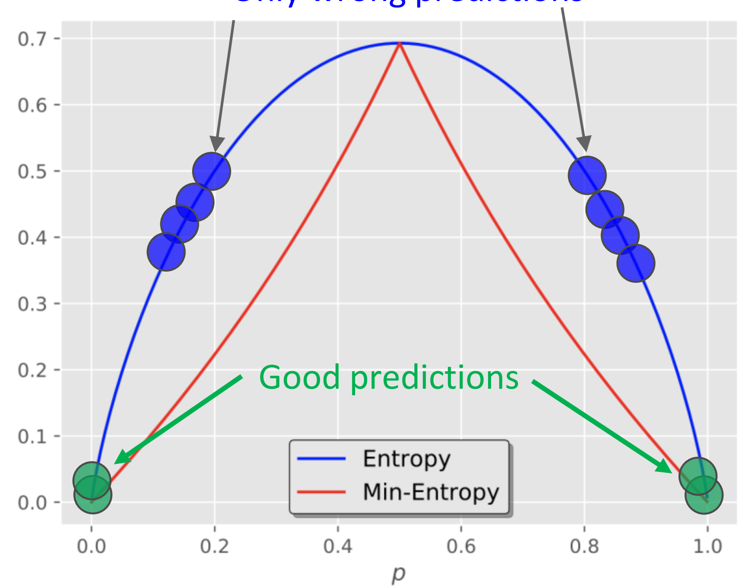
Instead of maximizing the entropy over all the samples (i.e., pixels), do it **only for those pixels that produce erroneous predictions**

Extension of Pereyra et al., ICLR’17 but in a smarter manner

Pereyra et *al.*
ALL Samples



MEEP
Only wrong predictions



Larrazabal et al., Maximum Entropy on Erroneous Predictions (MEEP): Improving model calibration for medical image segmentation, Arxiv’21

Existing methods

In the training (end-to-end)

MEEP

Formal definition

Instead of maximizing the entropy over all the samples (i.e., pixels), do it **only for those pixels that produce erroneous predictions**

Global objective

$$\mathcal{L} = \mathcal{L}_{Seg}(\mathbf{y}, \mathbf{s}) - \lambda \mathcal{L}_{me}(\mathbf{s}_w)$$

Standard segmentation loss over ALL the pixels

Regularization term over wrong predictions

Larrazabal et al., Maximum Entropy on Erroneous Predictions (MEEP): Improving model calibration for medical image segmentation, Arxiv'21

Existing methods

In the training (end-to-end)

MEEP

Formal definition

Instead of maximizing the entropy over all the samples (i.e., pixels), do it **only for those pixels that produce erroneous predictions**

Global objective

$$\mathcal{L} = \mathcal{L}_{Seg}(\mathbf{y}, \mathbf{s}) - \lambda \mathcal{L}_{me}(\mathbf{s}_w)$$

$$\mathcal{H}(\mathbf{s}_w) = -\frac{1}{|\mathbf{s}_w|} \sum_{k,i \in \mathbf{s}_w} s_{i,k} \log s_{i,k}$$

Larrazabal et al., Maximum Entropy on Erroneous Predictions (MEEP): Improving model calibration for medical image segmentation, Arxiv'21

Existing methods

In the training (end-to-end)

MEEP

Formal definition

Instead of maximizing the entropy over all the samples (i.e., pixels), do it **only for those pixels that produce erroneous predictions**

Global objective

$$\mathcal{L} = \mathcal{L}_{Seg}(\mathbf{y}, \mathbf{s}) - \lambda \mathcal{L}_{me}(\mathbf{s}_w)$$

$$\mathcal{H}(\mathbf{s}_w) = -\frac{1}{|\mathbf{s}_w|} \sum_{k,i \in \mathbf{s}_w} s_{i,k} \log s_{i,k}$$

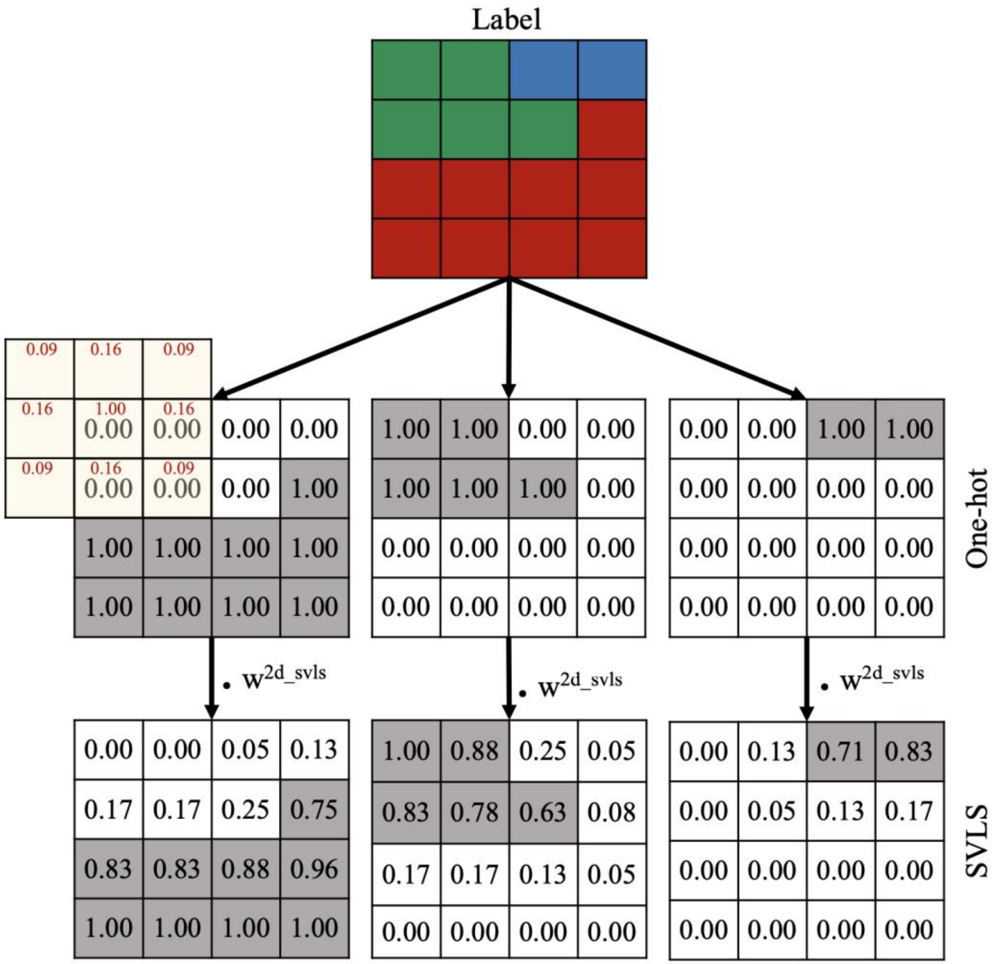
$$\mathcal{D}_{KL}(\mathbf{u} || \mathbf{s}_w) \stackrel{\text{K}}{=} \mathcal{H}(\mathbf{u}, \mathbf{s}_w)$$

Larrazabal et al., Maximum Entropy on Erroneous Predictions (MEEP): Improving model calibration for medical image segmentation, Arxiv'21

Existing methods

In the training (end-to-end)

Spatially Varying Label Smoothing (SVLS)

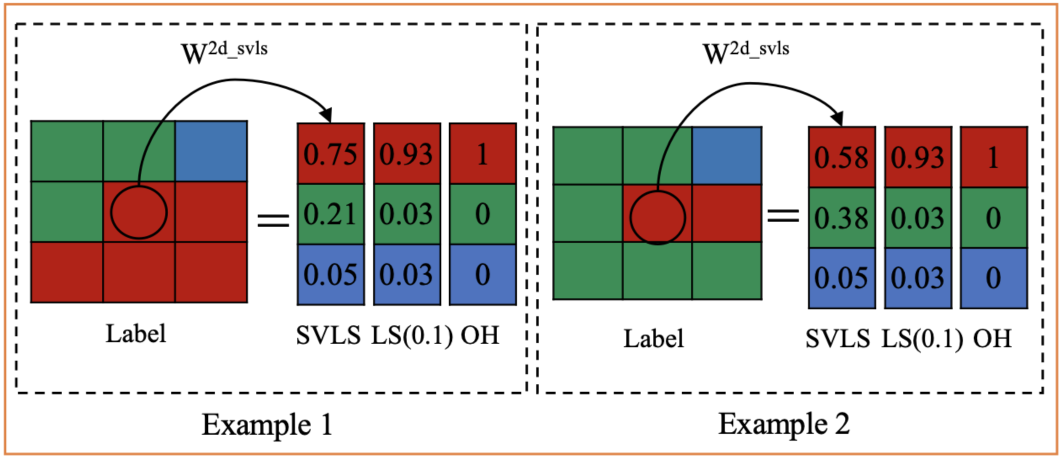


Islam and Glocker, Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations. IPMI’21

Existing methods

In the training (end-to-end)

Spatially Varying Label Smoothing (SVLS)

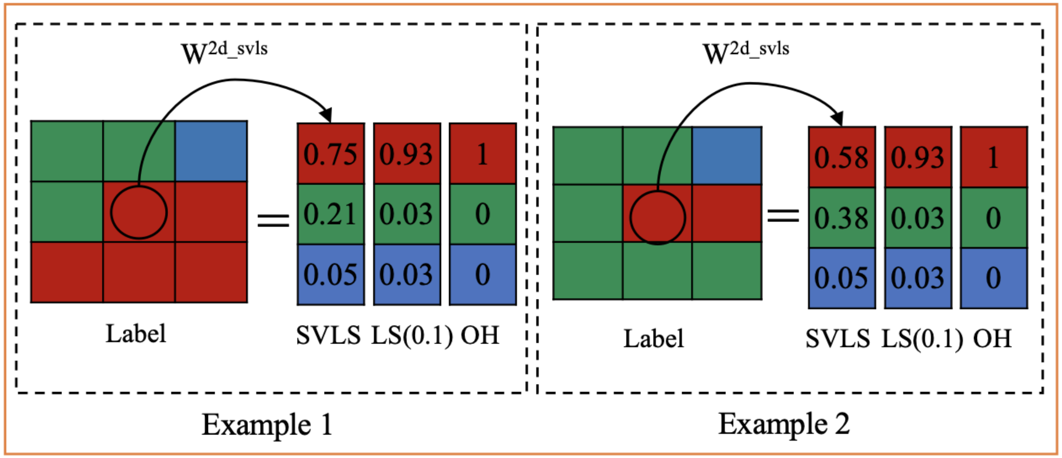


Islam and Glocker, Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations. IPMI'21

Existing methods

In the training (end-to-end)

Spatially Varying Label Smoothing (SVLS)



Smoothed SVLS labels

$$\tilde{y}_p^k = \frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i.$$

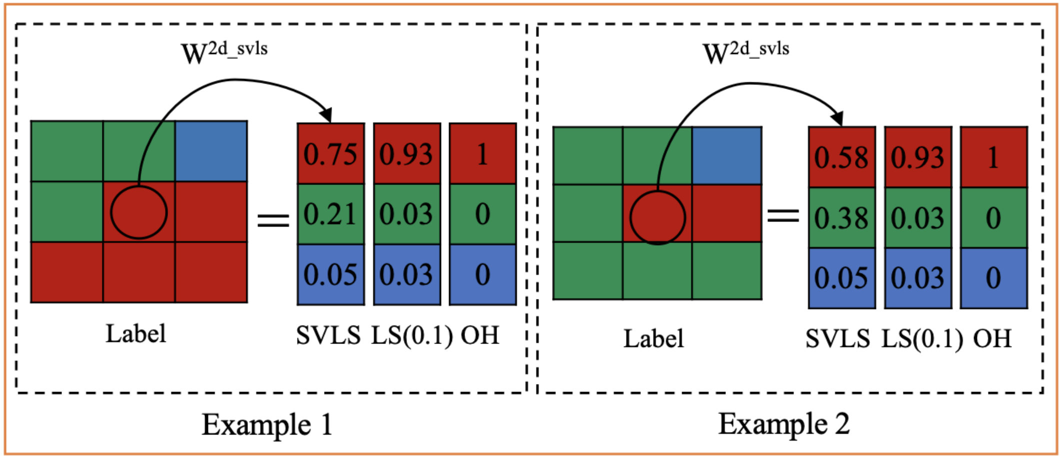
Kernel to smooth labels

Islam and Glocker, Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations. IPMI'21

Existing methods

In the training (end-to-end)

Spatially Varying Label Smoothing (SVLS)



Smoothed SVLS labels

$$\tilde{y}_p^k = \frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i.$$

One hot encoded is smoothed by the class distribution in the patch

Integrated in the standard CE

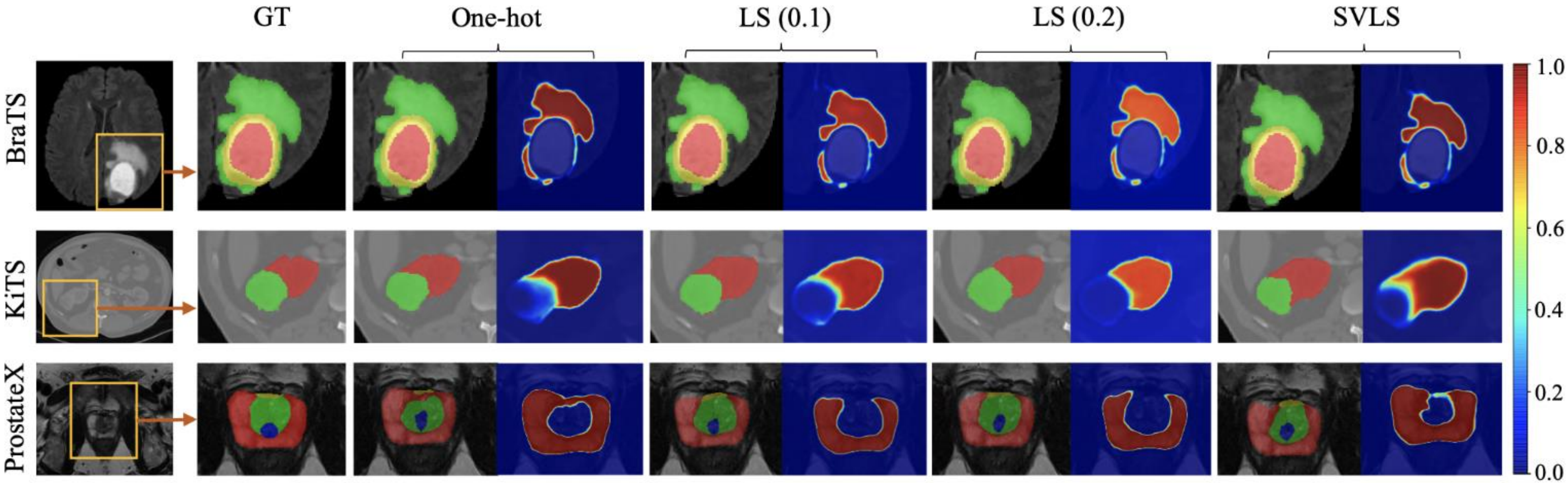
$$\mathcal{L} = -\sum_k \tilde{y}_p^k \log s_p^k,$$

Islam and Glocker, Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations. IPMI'21

Existing methods

In the training (end-to-end)

Spatially Varying Label Smoothing (SVLS)



Islam and Glocker, Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations. IPMI'21

Existing methods

In the training (end-to-end)

Limitations of SVLS

SVLS in the standard CE

$$\mathcal{L} = -\sum_k \left(\frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i\right) \log s_p^k,$$

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Limitations of SVLS

SVLS in the standard CE

$$\mathcal{L} = -\sum_k \left(\frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i\right) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|} \sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|} \sum_k \left(\sum_{\substack{i=1 \\ i \neq p}}^{d} y_i^k w_i\right) \log s_p^k,$$

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

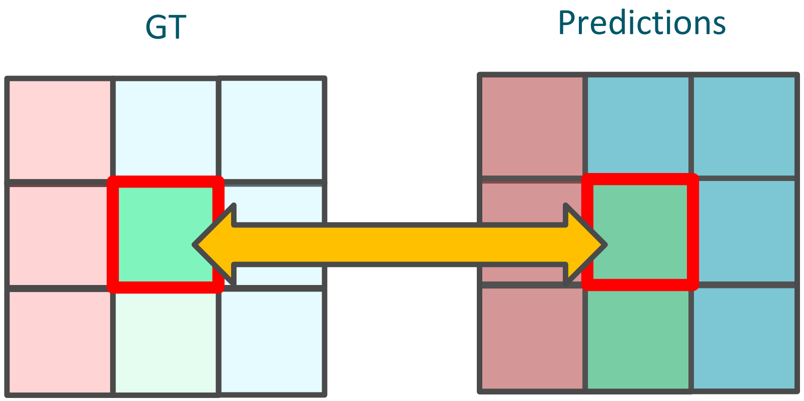
Limitations of SVLS

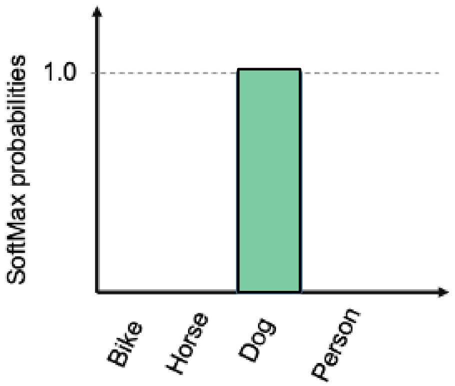
SVLS in the standard CE

$$\mathcal{L} = -\sum_k \Big(\frac{1}{|\sum_i^d w_i|}\sum_{i=1}^{d} y_i^k w_i\Big) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|}\sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|}\sum_k \Big(\sum_{\substack{i=1\\ i\neq p}}^{d} y_i^k w_i\Big) \log s_p^k,$$

Standard CE (wrt the GT)





Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

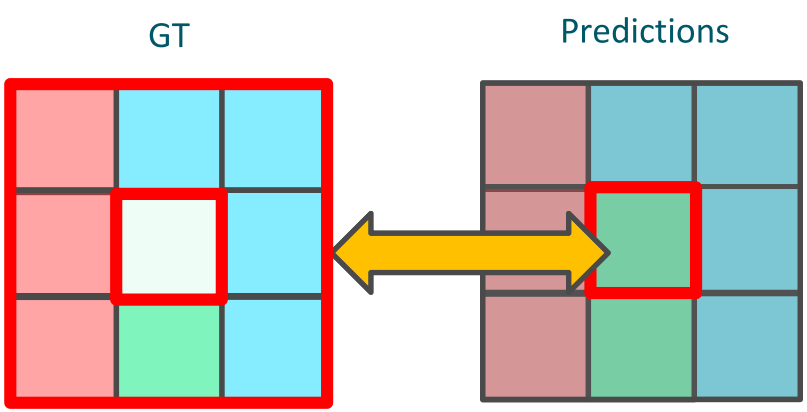
Limitations of SVLS

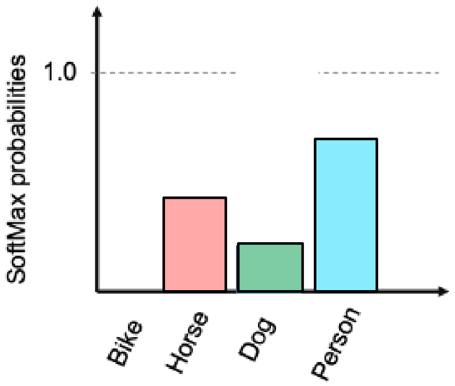
SVLS in the standard CE

$$\mathcal{L} = -\sum_k \left(\frac{1}{|\sum_i^d w_i|}\sum_{i=1}^{d} y_i^k w_i\right) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|}\sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|}\sum_k \left(\sum_{\substack{i=1 \\ i \neq p}}^{d} y_i^k w_i\right) \log s_p^k,$$

Standard CE
(wrt the class-distribution value)





Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Limitations of SVLS

SVLS in the standard CE

$$\mathcal{L} = -\sum_k \Big(\frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i \Big) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|} \sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|} \sum_k \Big(\sum_{\substack{i=1 \\ i \neq p}}^{d} y_i^k w_i \Big) \log s_p^k,$$

$$\mathcal{L} = \underbrace{-\sum_k y_p^k \log s_p^k}_{CE} \underbrace{-\sum_k \tau_k \log s_p^k}_{\text{Constraint on } \boldsymbol{\tau}} .$$

Distribution of each class within the patch

$$\boxed{\tau_k = \sum_{\substack{i=1 \\ i \neq p}}^{d} y_i^k w_i}$$

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Limitations of SVLS

SVLS in the standard CE

$$\mathcal{L} = -\sum_k \Big(\frac{1}{|\sum_i^d w_i|}\sum_{i=1}^{d} y_i^k w_i\Big) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|}\sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|}\sum_k \Big(\sum_{\substack{i=1\\ i\neq p}}^{d} y_i^k w_i\Big) \log s_p^k,$$

Two main problems

$$\mathcal{L} = \underbrace{-\sum_k y_p^k \log s_p^k}_{CE} \underbrace{- \sum_k \tau_k \log s_p^k}_{\text{Constraint on } \boldsymbol{\tau}} .$$

1 – No mechanism to control the importance of the constraint

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Limitations of SVLS

SVLS in the standard CE

$$\mathcal{L} = -\sum_k \Big(\frac{1}{|\sum_i^d w_i|} \sum_{i=1}^{d} y_i^k w_i\Big) \log s_p^k,$$

$$\mathcal{L} = -\frac{1}{|\sum_i^d w_i|} \sum_k y_p^k \log s_p^k - \frac{1}{|\sum_i^d w_i|} \sum_k \Big(\sum_{\substack{i=1 \\ i \neq p}}^{d} y_i^k w_i\Big) \log s_p^k,$$

$$\mathcal{L} = \underbrace{-\sum_k y_p^k \log s_p^k}_{CE} \underbrace{- \sum_k \tau_k \log s_p^k}_{\text{Constraint on } \boldsymbol{\tau}} .$$

Two main problems

1 – No mechanism to control the importance of the constraint

2 – The a priori distribution cannot be computed easily

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Proposed solution

Our solution

$$\min \quad \mathcal{L}_{CE} \quad \text{s.t.} \quad \boldsymbol{\tau} = \mathbf{l},$$

A priori distribution

Logit distribution

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Proposed solution

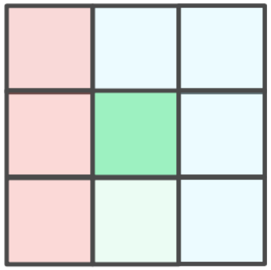
Our solution

$$\min \quad \mathcal{L}_{CE} \quad \text{s.t.} \quad \boldsymbol{\tau} = \mathbf{l},$$

A priori distribution

Logit distribution

$$\min \quad \mathcal{L}_{CE} + \lambda \sum_k \max(0, |\tau_k - l_k|),$$



Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Existing methods

In the training (end-to-end)

Proposed solution

	FLARE				ACDC				BraTS			
	DSC	HD	ECE	CECE	DSC	HD	ECE	CECE	DSC	HD	ECE	CECE
CE+DSC ($\lambda = 1$)	0.846	5.54	0.058	0.034	**0.828**	3.14	0.137	0.084	0.777	**6.96**	0.178	0.122
FL [10] ($\gamma = 3$)	0.834	6.65	0.053	0.059	0.620	7.30	0.153	0.179	**0.848**	9.00	**0.097**	0.119
ECP [21] ($\lambda = 0.1$)	**0.860**	**5.30**	**0.037**	**0.027**	0.782	4.44	0.130	0.094	0.808	8.71	0.138	0.099
LS [22] ($\alpha = 0.1$)	**0.860**	5.33	0.055	0.049	0.809	3.30	**0.083**	0.093	0.820	7.78	**0.112**	0.108
SVLS [7] ($\sigma = 2$)	0.857	5.72	0.039	0.036	0.824	**2.81**	0.091	0.083	0.801	8.44	0.146	0.111
MbLS [11] (m=5)	0.836	5.75	0.046	0.041	0.827	2.99	0.103	**0.081**	0.838	7.94	0.127	**0.095**
Ours ($\lambda = 0.1$)	**0.868**	**4.88**	**0.033**	**0.031**	**0.854**	**2.55**	**0.048**	**0.061**	**0.850**	**5.78**	**0.112**	**0.097**

Empirical validation of several choices

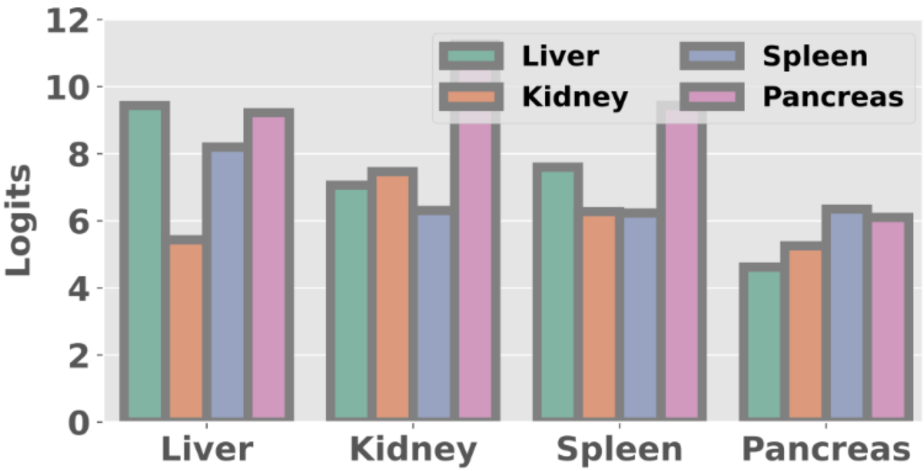
	FLARE				ACDC				BraTS			
	DSC	HD	ECE	CECE	DSC	HD	ECE	CECE	DSC	HD	ECE	CECE
Constraint on **s**	0.862	5.14	0.043	0.030	0.840	2.66	0.068	0.071	0.802	8.28	0.145	0.104
L2-penalty	0.851	5.48	0.065	0.054	0.871	1.78	0.059	0.080	0.851	7.90	0.078	0.091
Patch size: 5×5	0.875	5.96	0.032	0.031	0.813	3.50	0.078	0.077	0.735	7.45	0.119	0.092

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

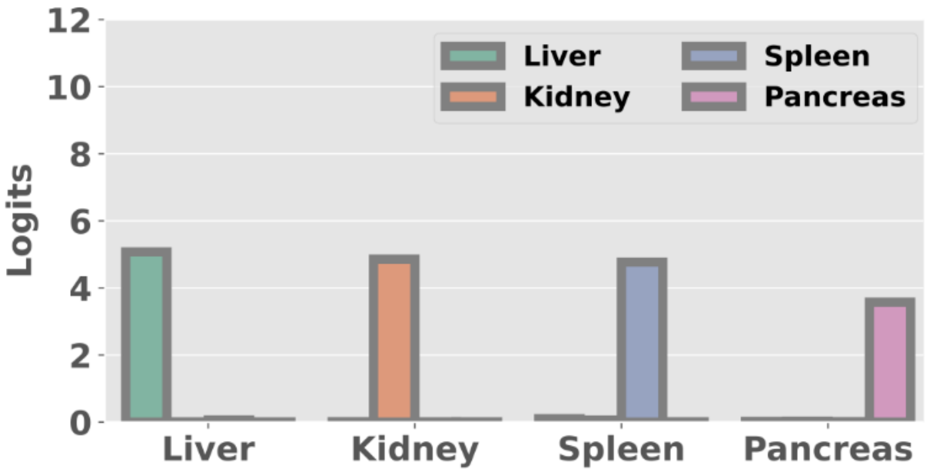
Existing methods

In the training (end-to-end)

Proposed solution



SVLS



Proposed

Murugesan et al. Trust your neighbours: Penalty-based constraints for model calibration. Arxiv'23

Take home message

- Precise uncertainty estimates are very important in a broad span of problems.

- Integrating a mechanism to control overconfidence predictions during training seems to be an efficient alternative.

- Strategies specifically designed for segmentation tasks are required.

- Despite the importance of the topic, calibrating segmentation networks, and particularly in the medical domain is underexplored.