Ph.D. Thesis Defense

Learning with Limited Labels

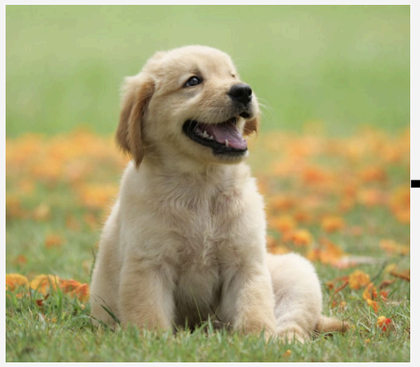
Yassine Ouali
Advisors: Céline Hudelot & Myriam Tami

Université Paris-Saclay, CentraleSupélec, MICS

Ismail Ben Ayed: Rapporteur & Examinateur
Matthieu Cord: Rapporteur & Examinateur
Vincent Lepetit: Examinateur
Stéphane Herbin: Examinateur
Clément Rambour: Examinateur

Area of interest: Machine Learning

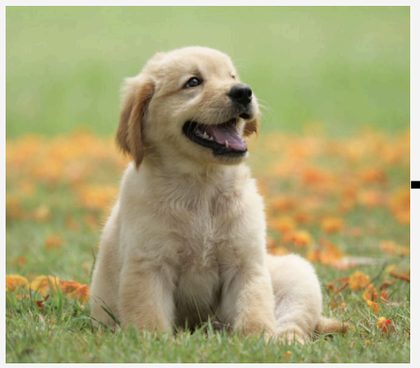
The need for ML/DL based methods



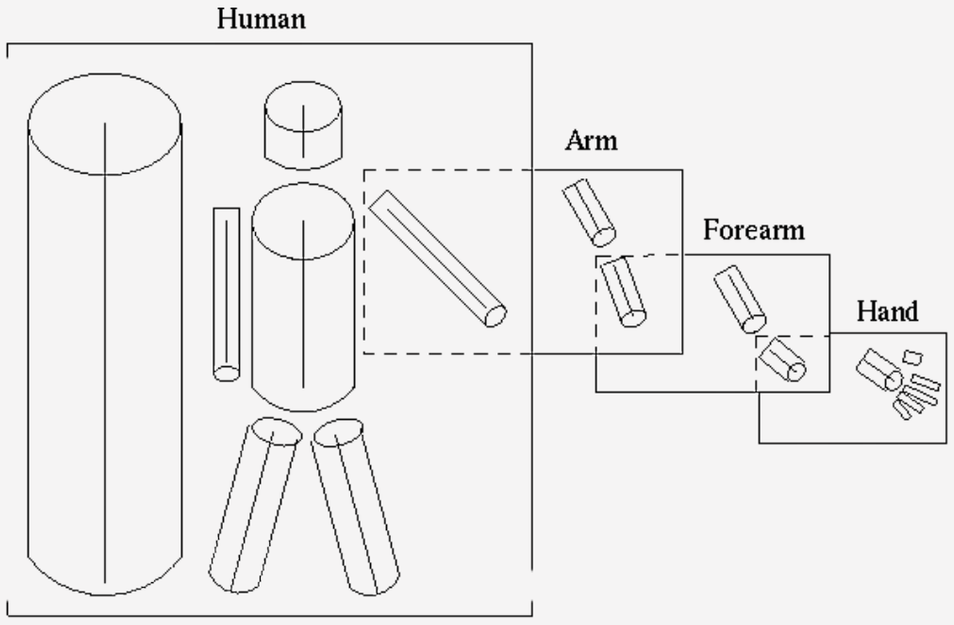
ML/DL based methods are necessary for many complex tasks.

Area of interest: Machine Learning

The need for ML/DL based methods



Classify → Category

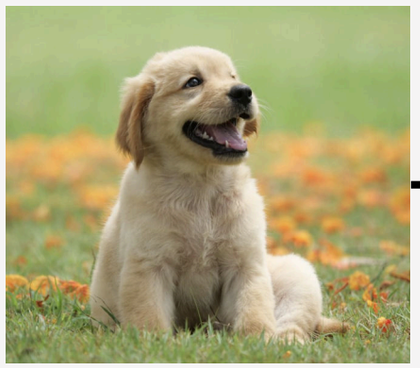


For example: image classification, we can try to engineer a solution ...

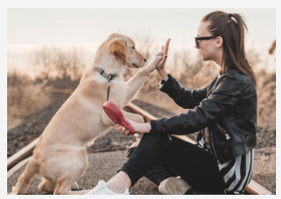
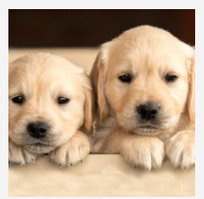
Marr's Theory: From primal sketch to 3-D models.

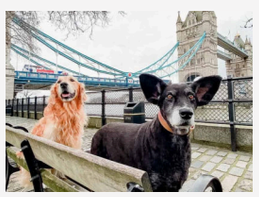
Area of interest: Machine Learning

The need for ML/DL based methods

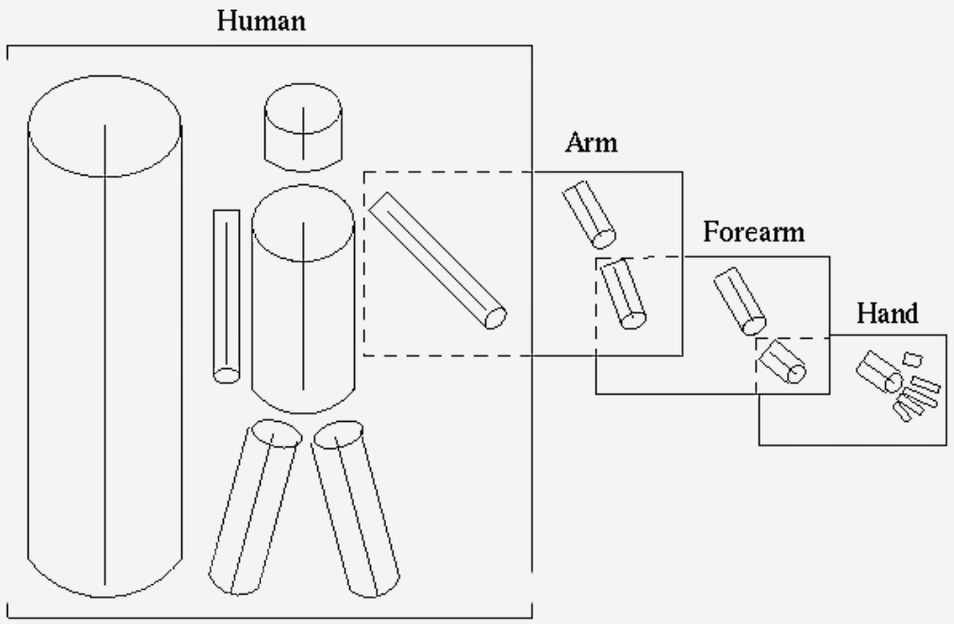


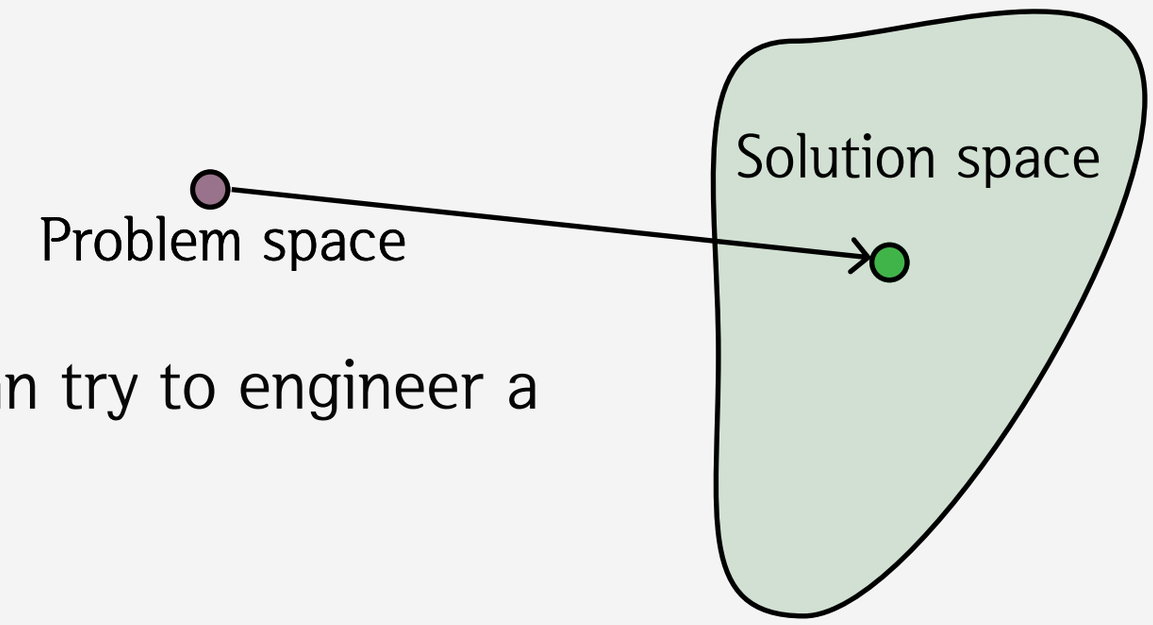
Classify → Category



but we have too many variations to consider

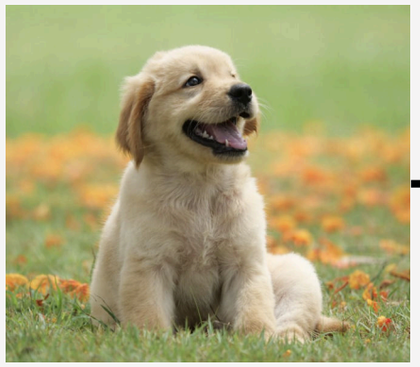




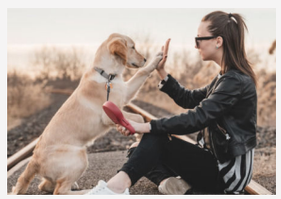
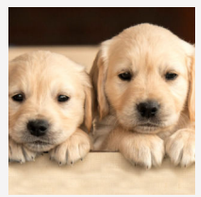
For example: image classification, we can try to engineer a solution ...

Area of interest: Machine Learning

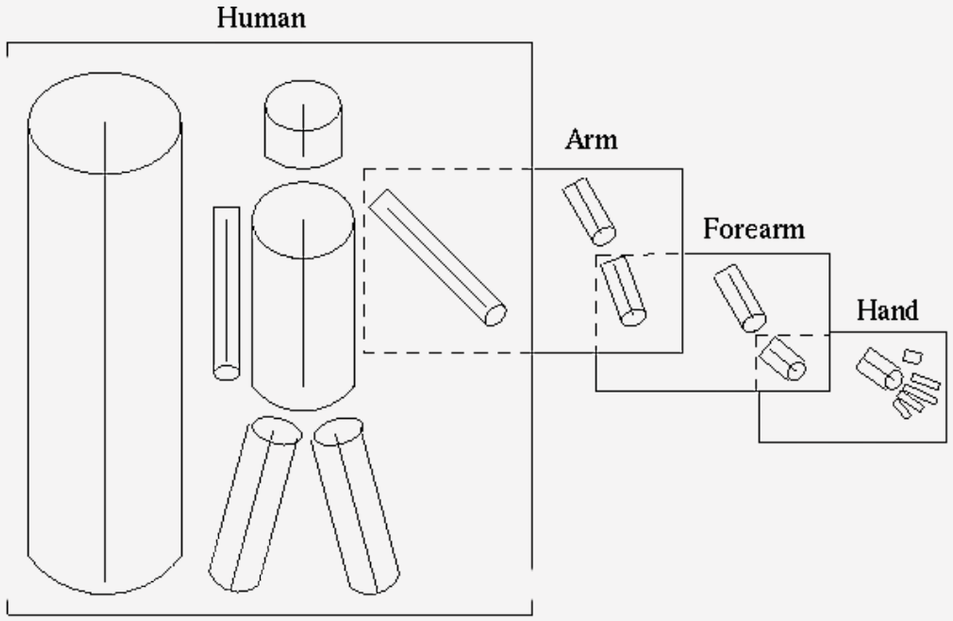
The need for ML/DL based methods



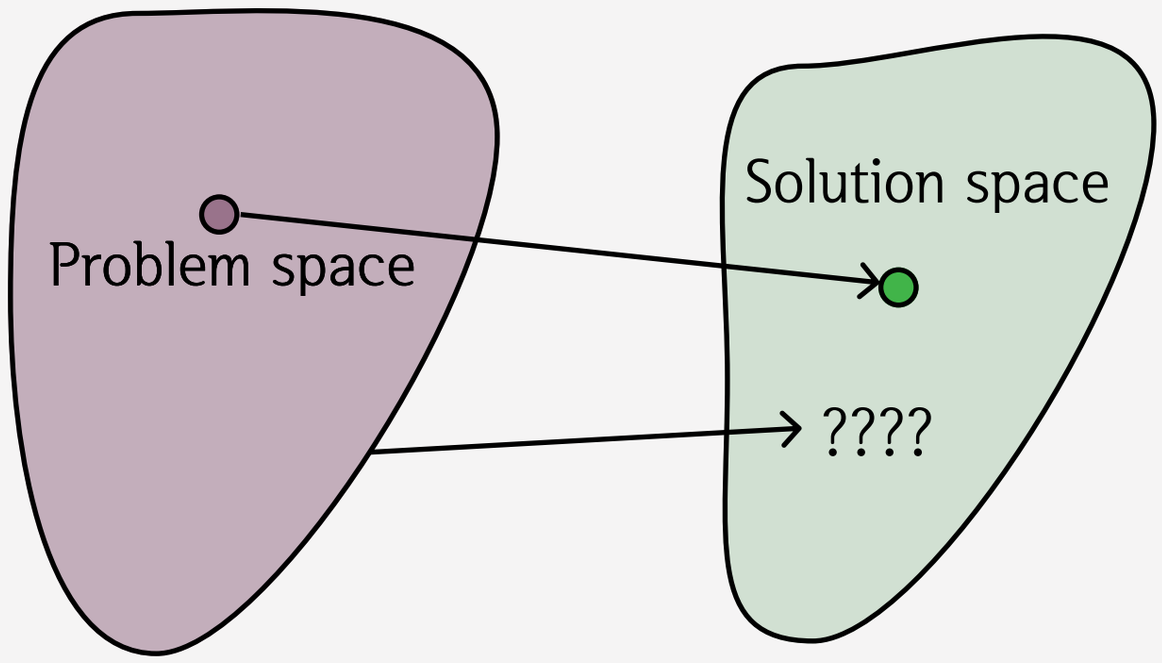
Classify → Category

but we have too many variations to consider.

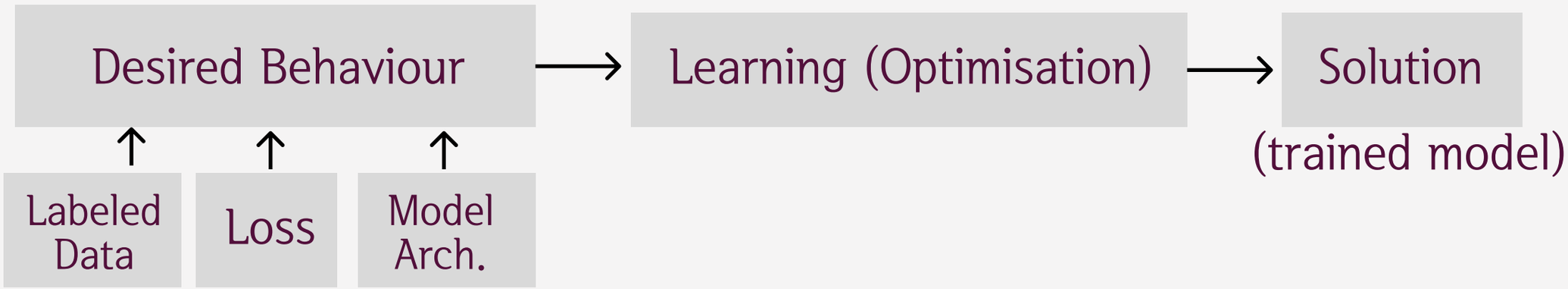


For example: image classification,
we can try to engineer a solution ...



ML/DL based solution

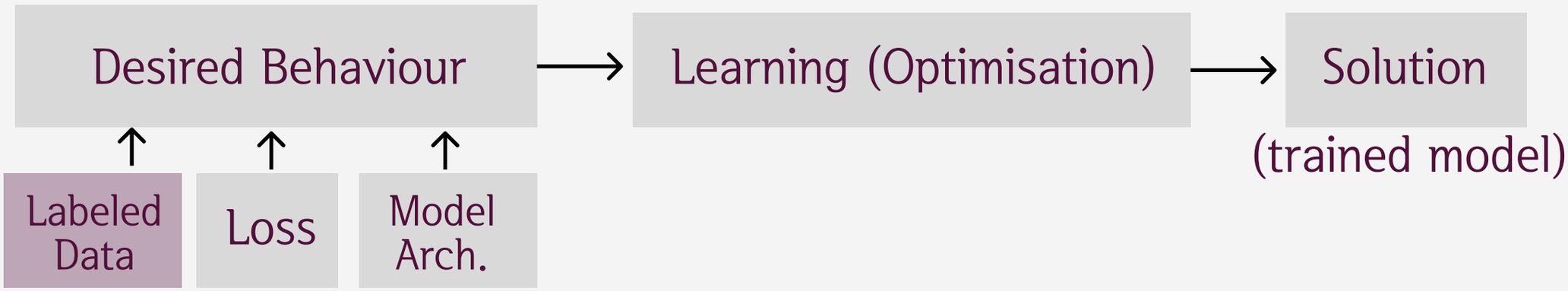
Optimise instead of conceive



Find the solution via optimisation.

ML/DL based solution

The drawback of such a SL-based pipeline

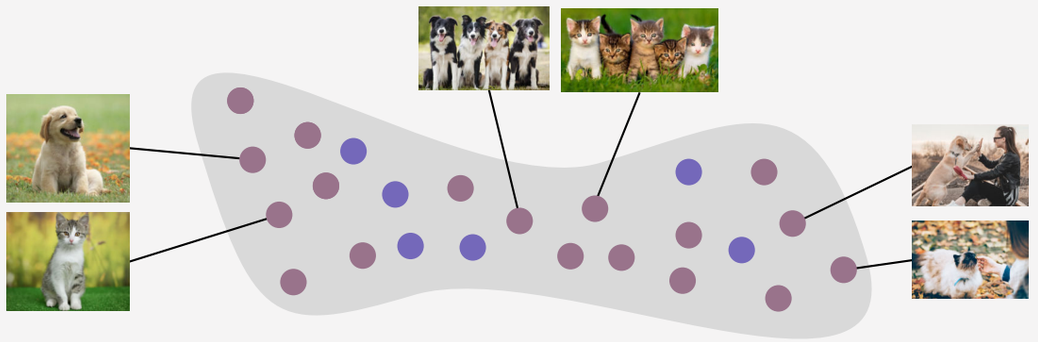


But this supervised learning based setup has many drawbacks.

Label-efficient DL: Motivations

Solving a given task: dog/cat classification with supervised learning

Data collection



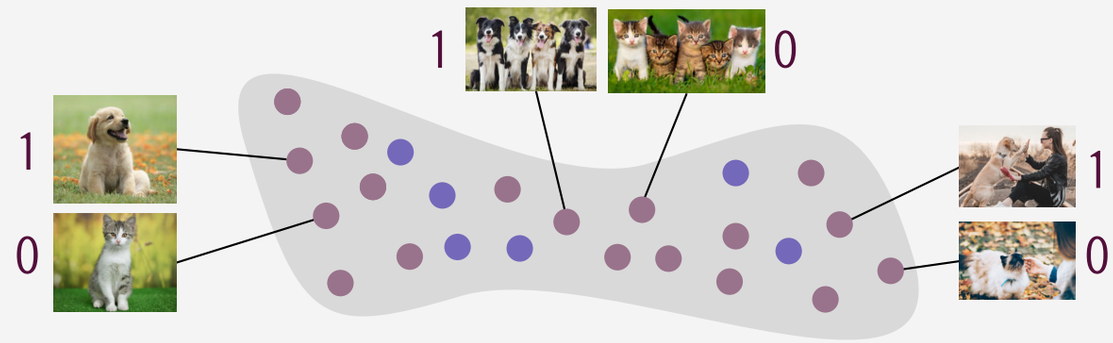
We start by collecting a diverse set of data points.

Label-efficient DL: Motivations

Solving a given task: dog/cat classification with supervised learning

Data collection

Data labelling



We label the collected data points with the desired classes.

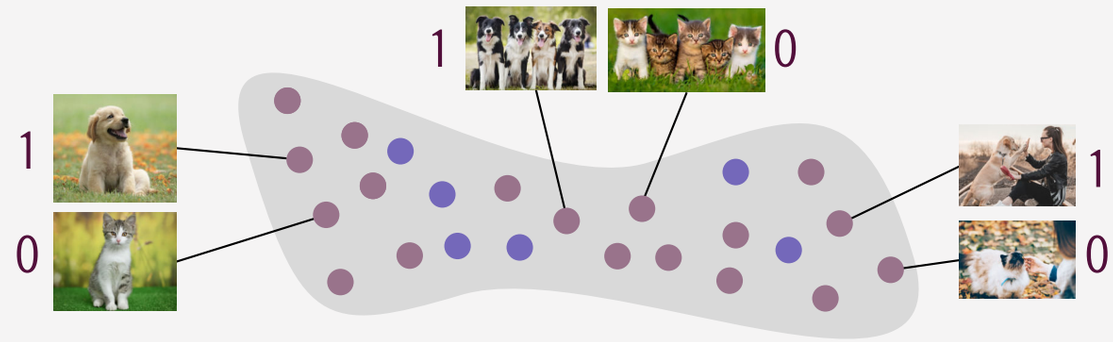
Label-efficient DL: Motivations

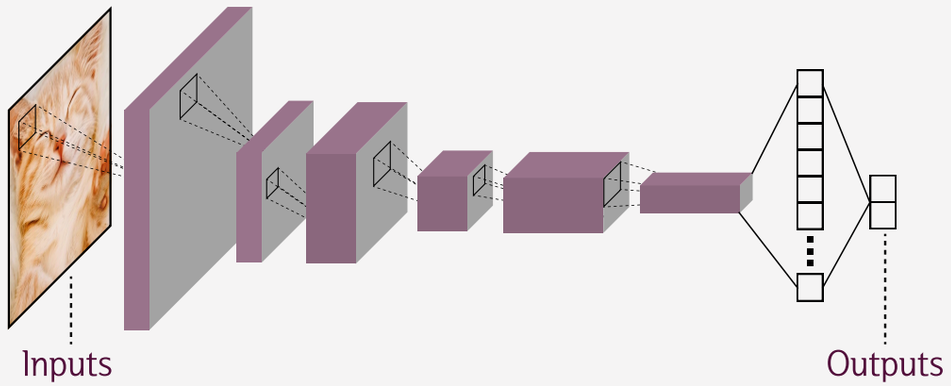
Solving a given task: dog/cat classification with supervised learning

Data collection

Data labelling

Selection/design





We define the neural architecture, loss functions, learning procedure ....

Label-efficient DL: Motivations

Solving a given task: dog/cat classification with supervised learning

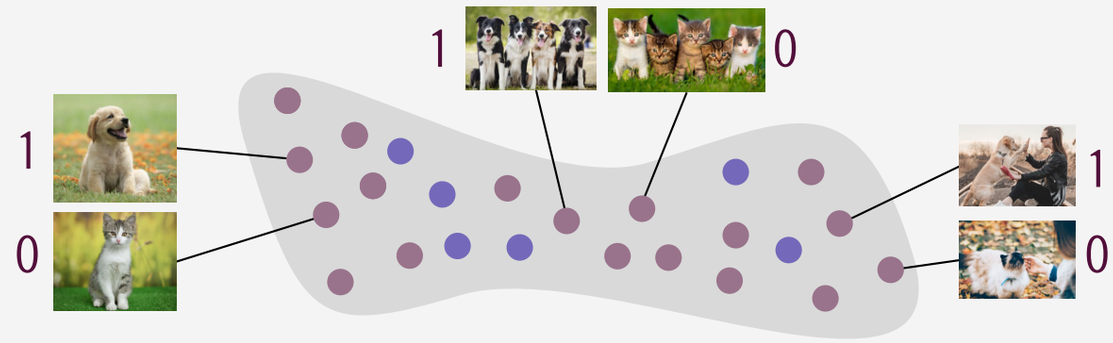
Data collection

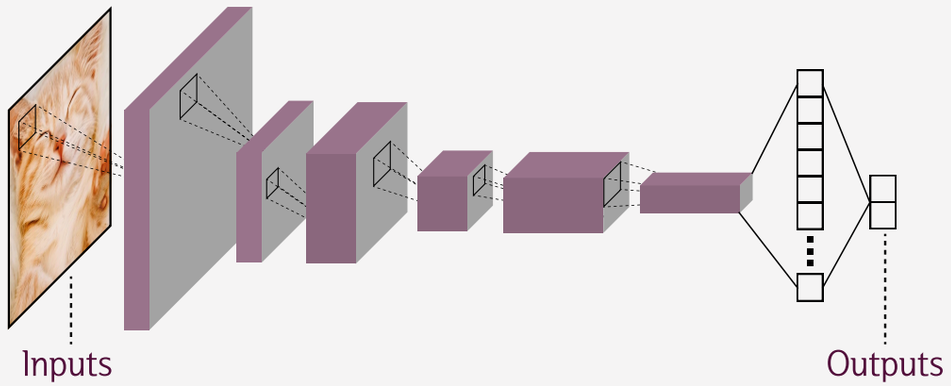
Data labelling

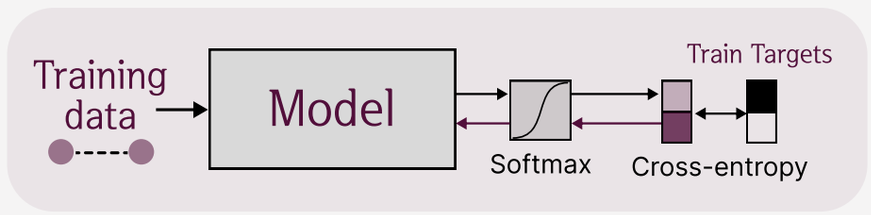
Selection/design

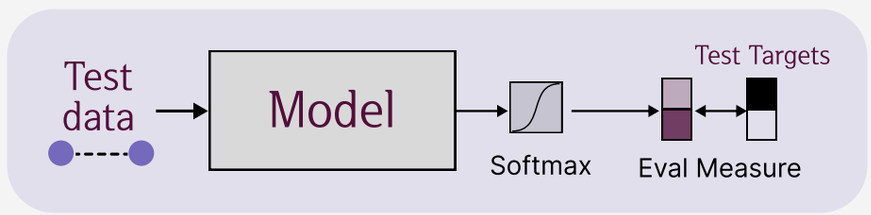
Training

Testing









Train the model to obtain the desired behaviour, then test it on held out data.

Label-efficient DL: Motivations

Solving a given task: dog/cat classification with supervised learning

Data collection

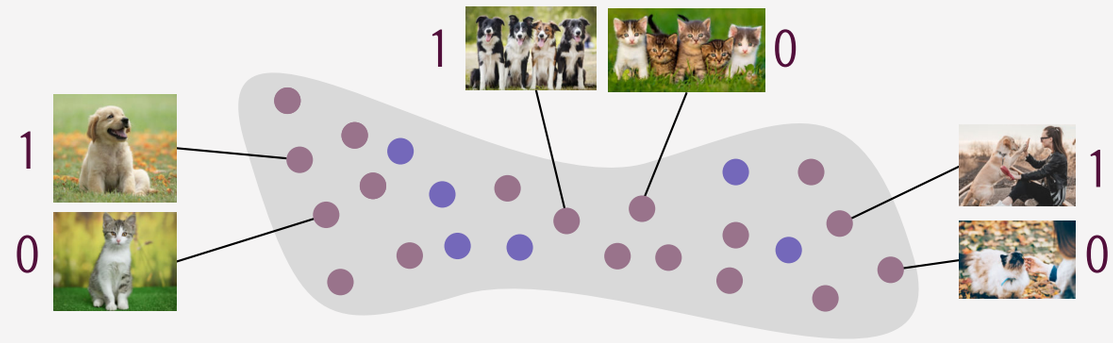
Data labelling

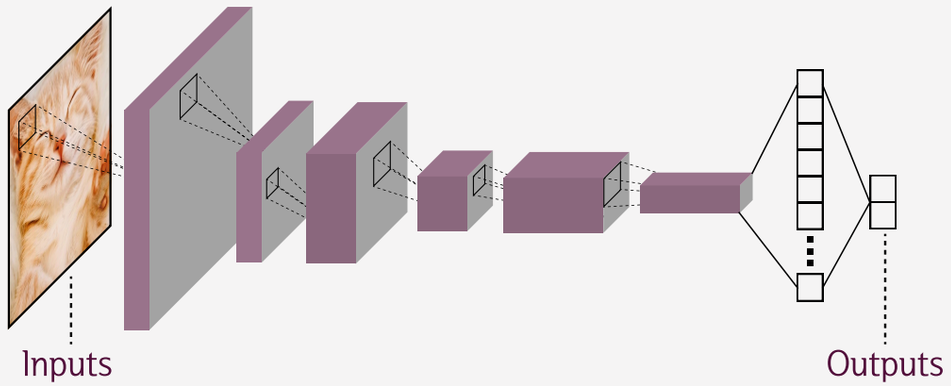
Selection/design

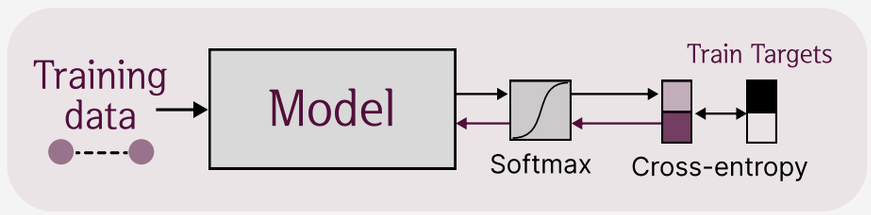
Training

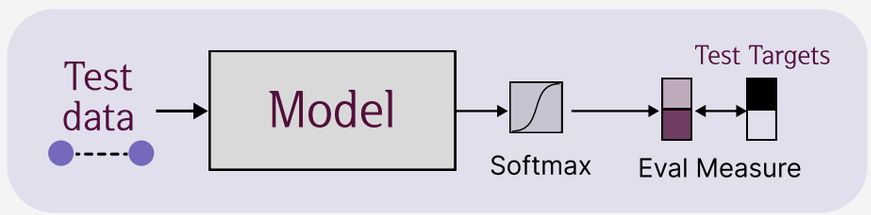
Testing

Deployment





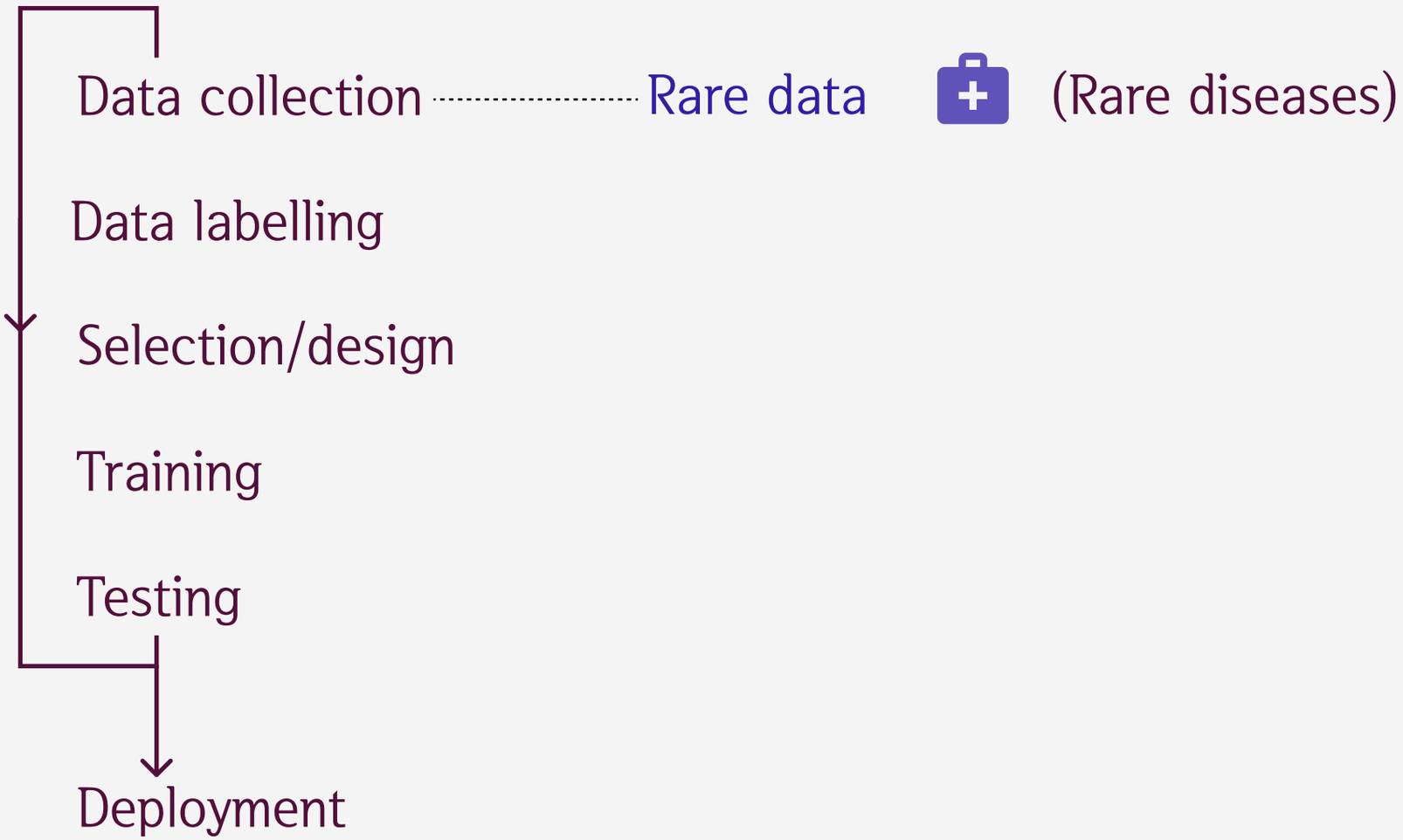




If the results are acceptable, deploy the model to solve the desired task.

Label-efficient DL: Motivations

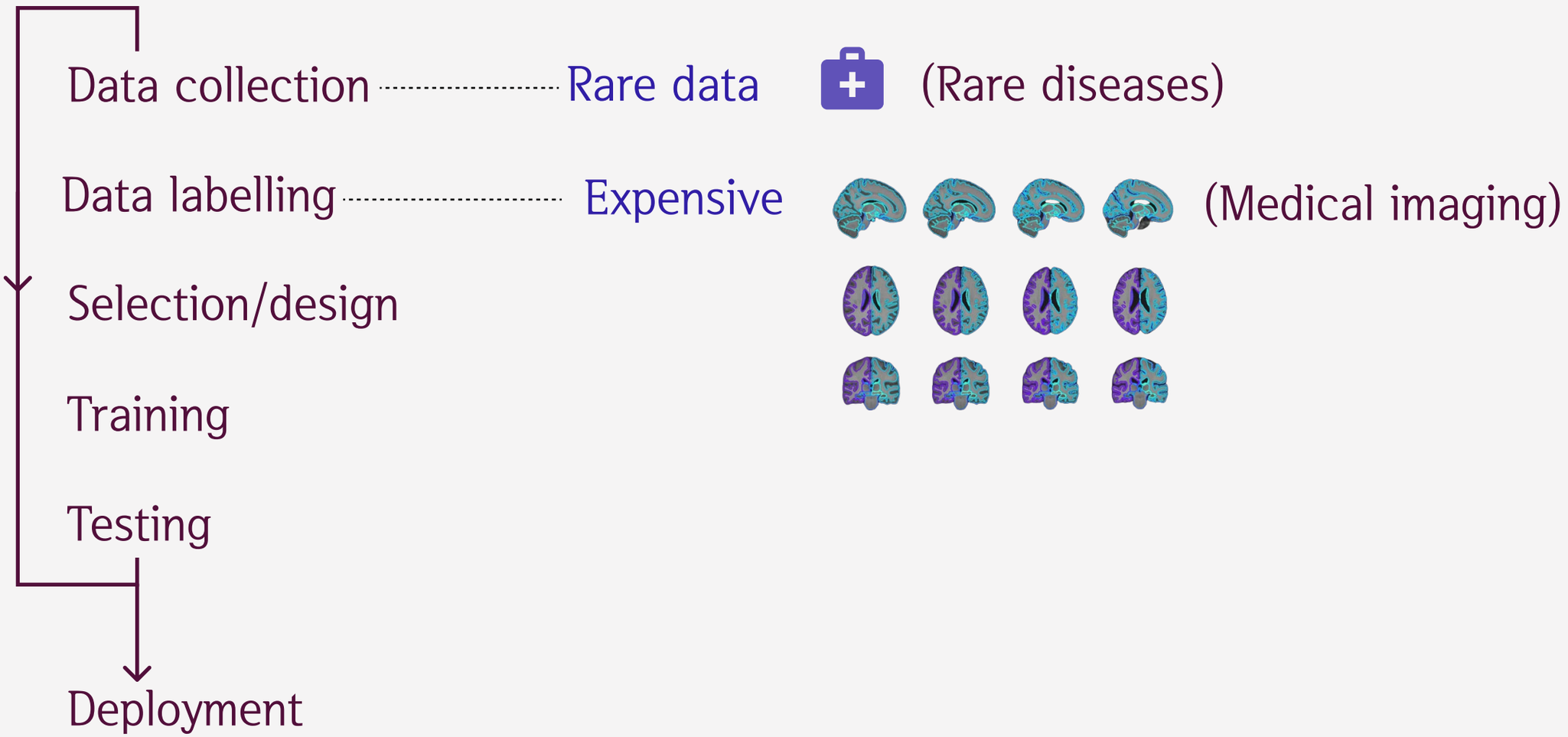
Limitations of supervised learning



But, ... data can be hard to acquire/collect.

Label-efficient DL: Motivations

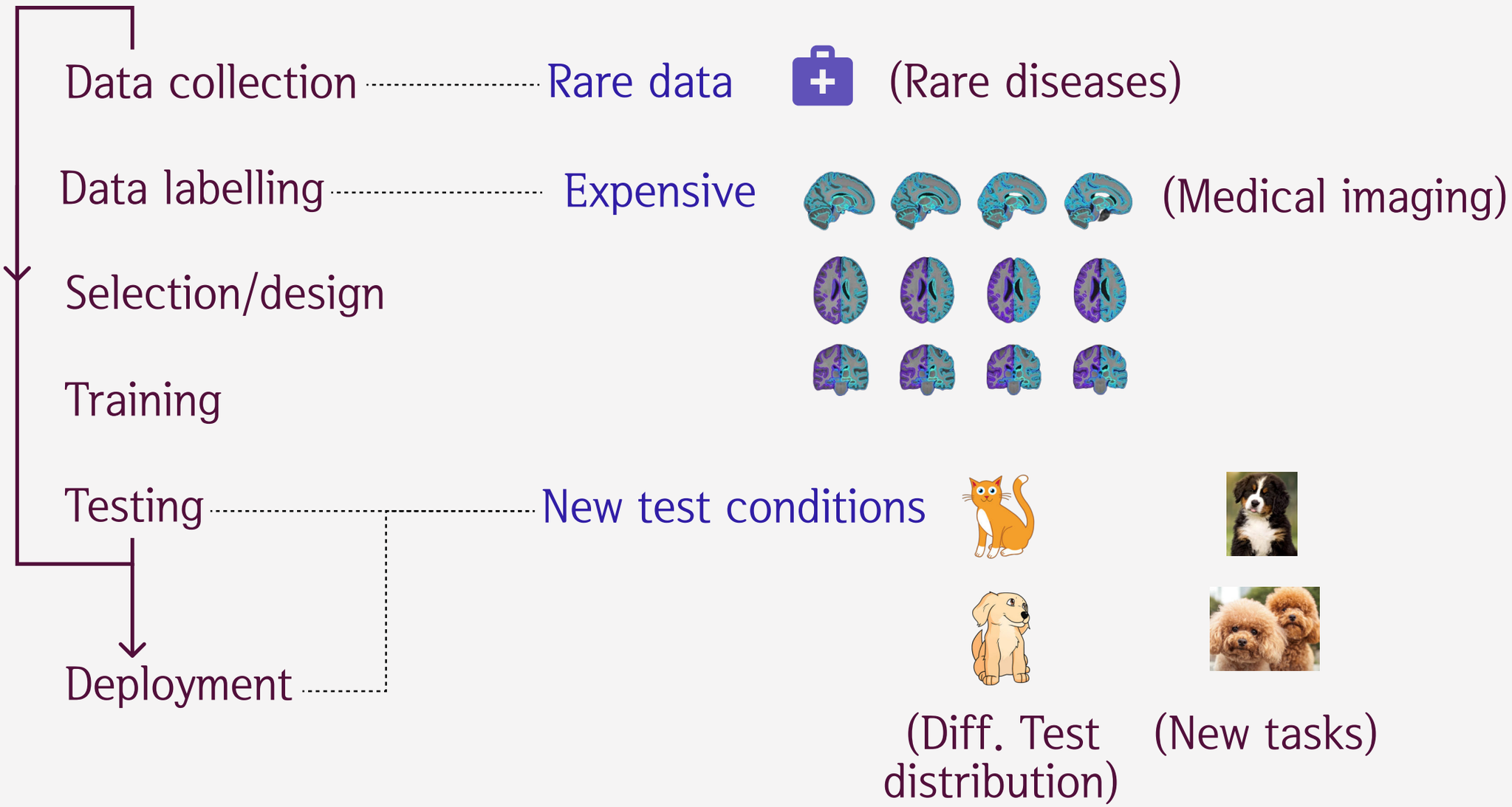
Limitations of supervised learning



But, ... data can also be expensive to annotate, requiring expert knowledge.

Label-efficient DL: Motivations

Limitations of supervised learning

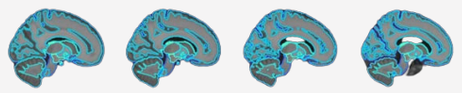


The test/deployment conditions can change rapidly.

Label-efficient DL: Motivations

Limitations of supervised learning

Data collection ..... Rare data  (Rare diseases)

Data labelling ..... Expensive  (Medical imaging)

Selection/design

Training

Testing ..... New test conditions  

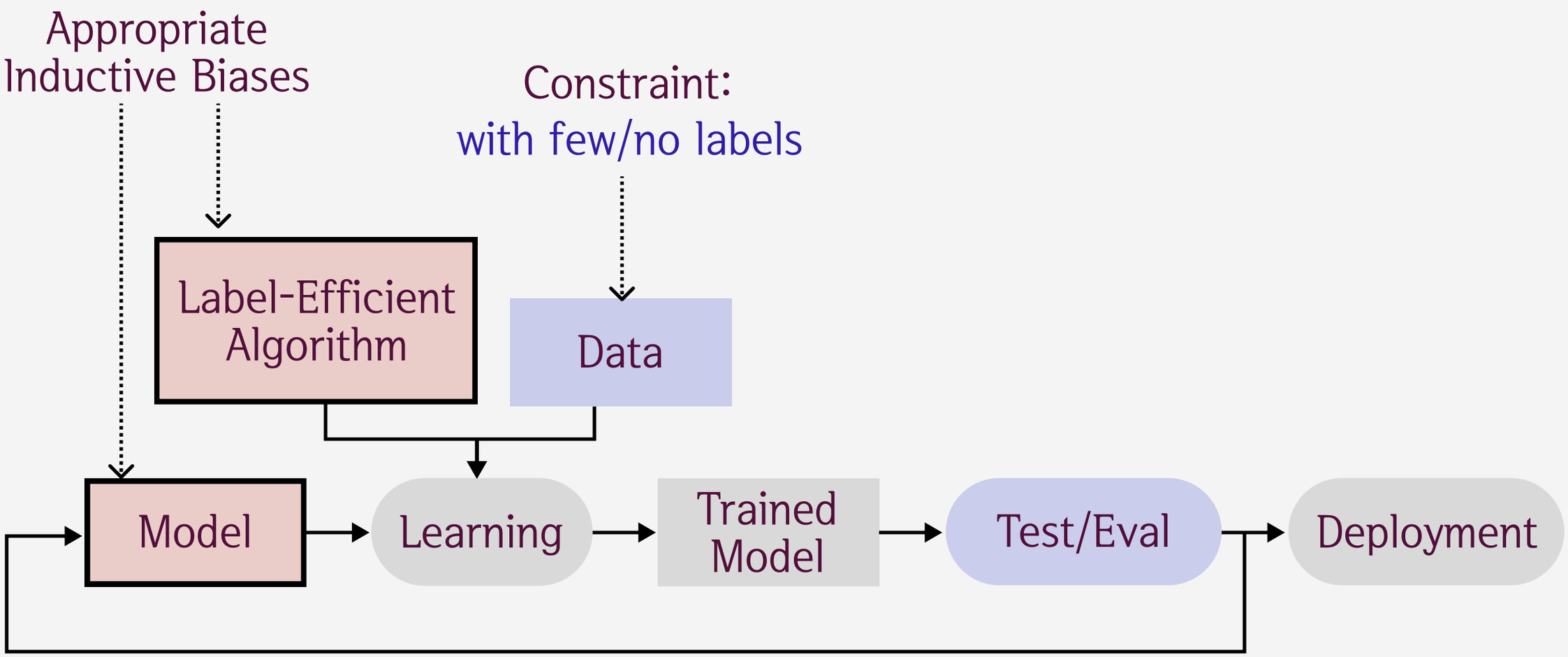
(Diff. Test distribution) (New tasks)

Deployment

-> These cases must be taken into consideration at the design/training stage.

Label-efficient DL: Motivations

Goal: develop label efficient DL methods



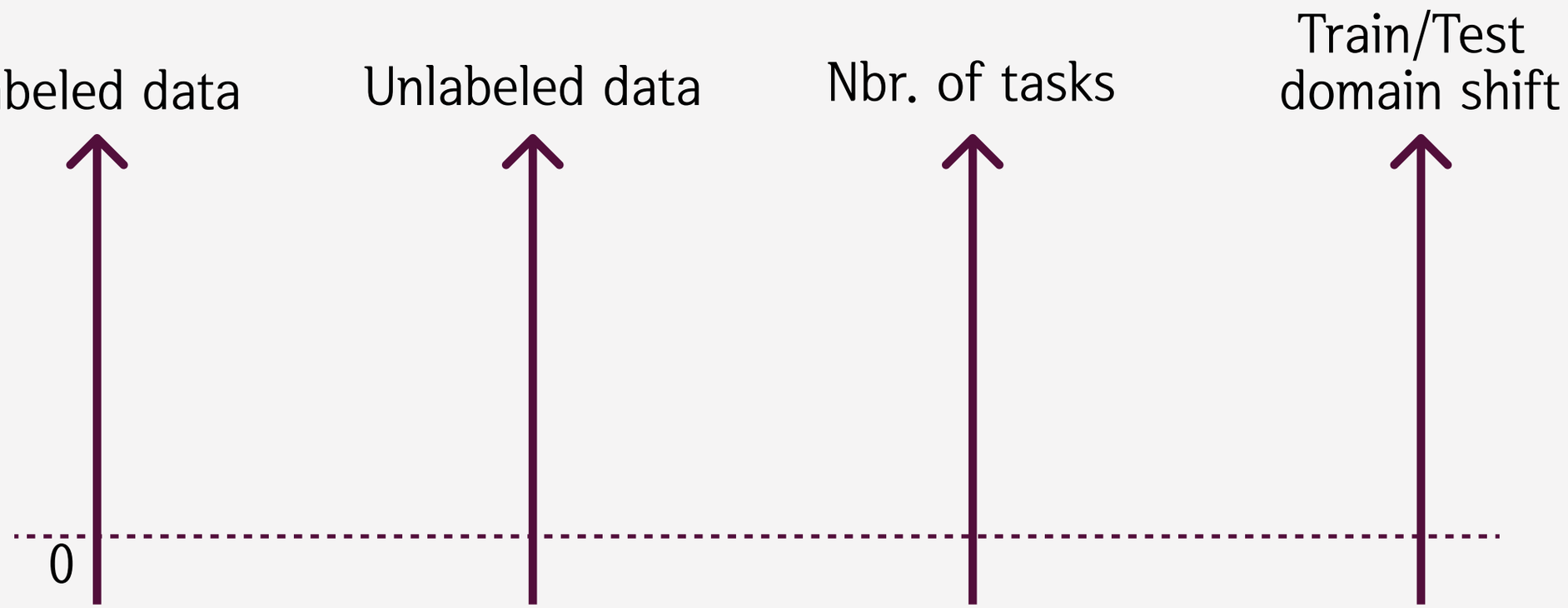
We focus on learning under the constraint of limited labels.

And set to define the appropriate inductive biases (learning procedure & model architecture) under this constraint.

How to define “label efficient” settings

Supervised learning as a starting point

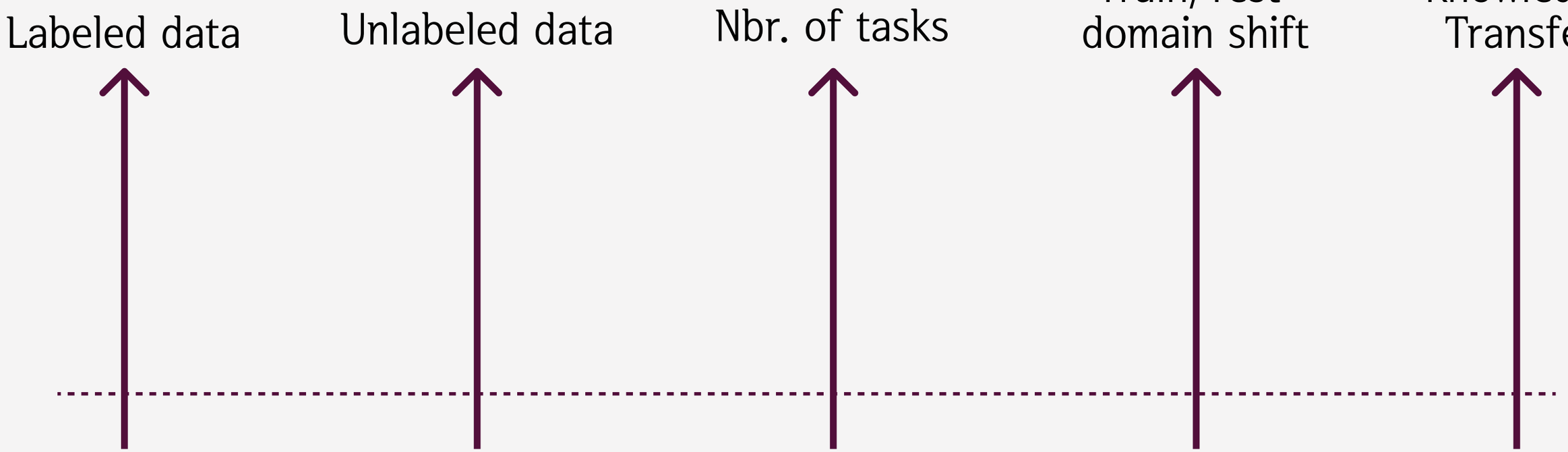
SL: Solve a single task using a large amount of labeled data, and test on similar data.



How to define “label efficient” settings

Supervised learning as a starting point

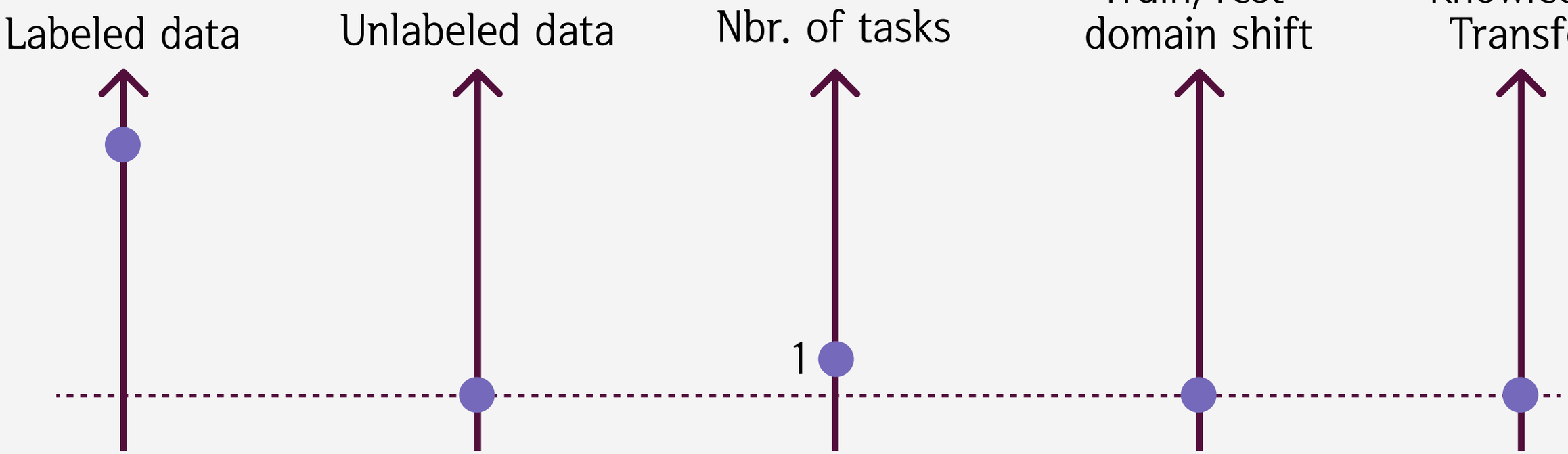
SL: Solve a single task using a large amount of labeled data, and test on similar data.



How to define “label efficient” settings

Supervised learning as a starting point

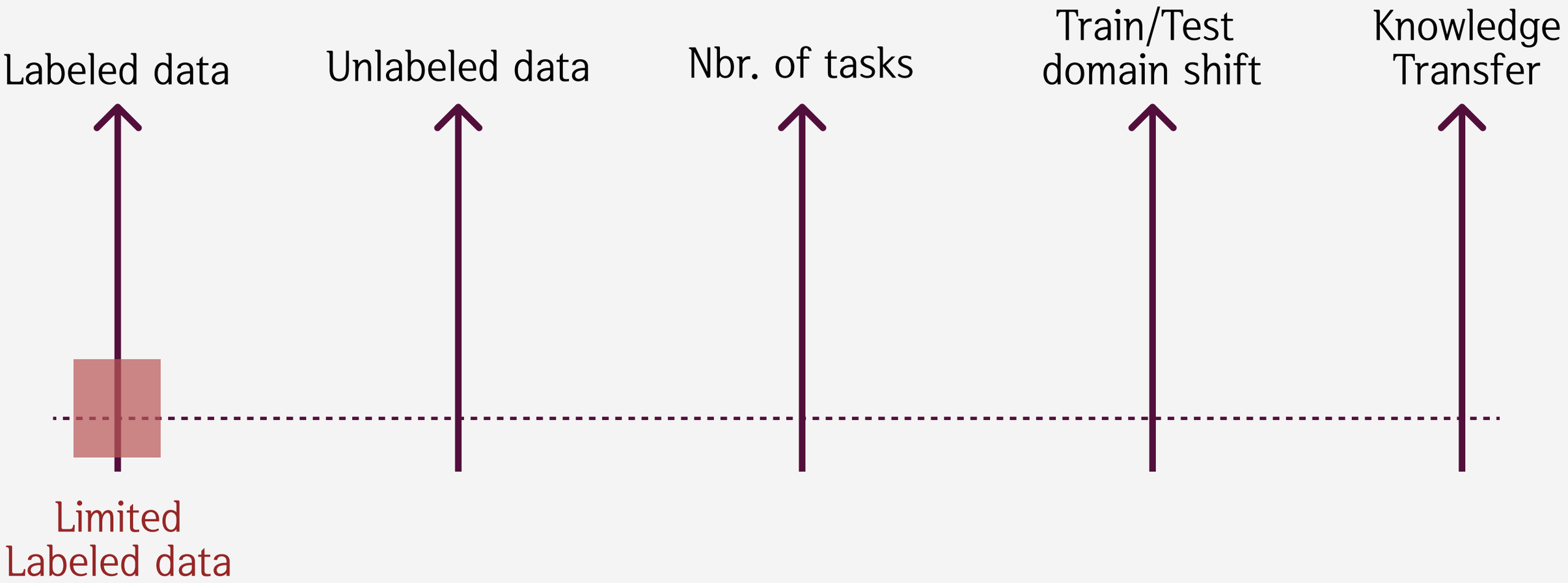
SL: Solve a single task using a large amount of labeled data, and test on similar data.



How to define “label efficient” settings

Label efficient DL: the many possible variations

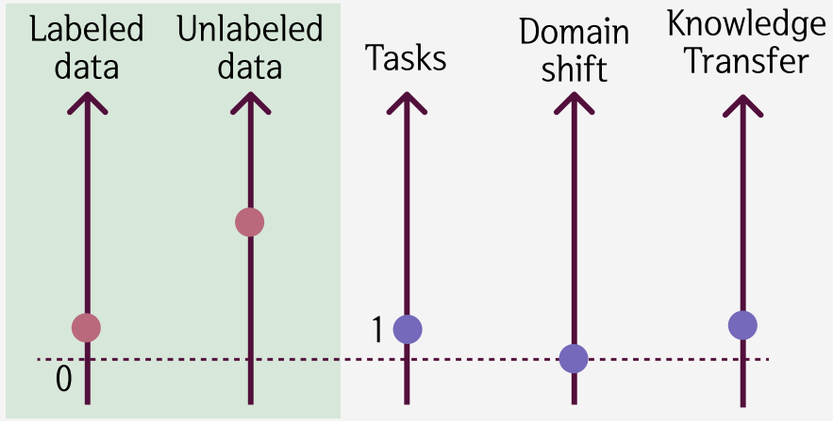
Only condition: use a limited amount of labels at train time.



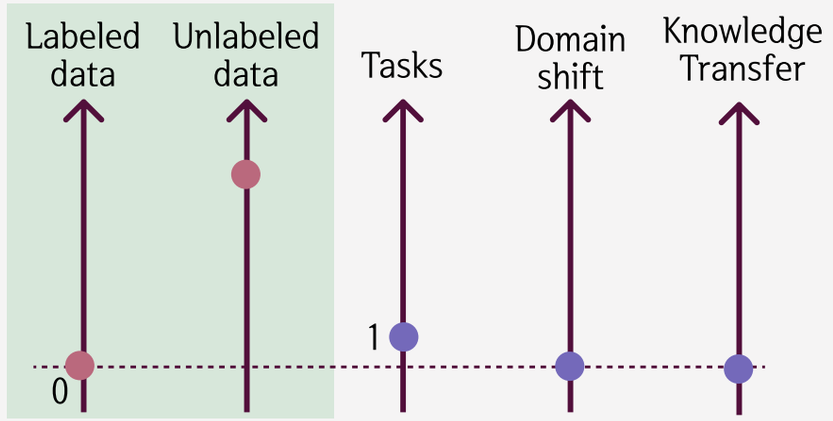
There is many possible variations that fall under the “label efficient” category.

How to define “label efficient” settings

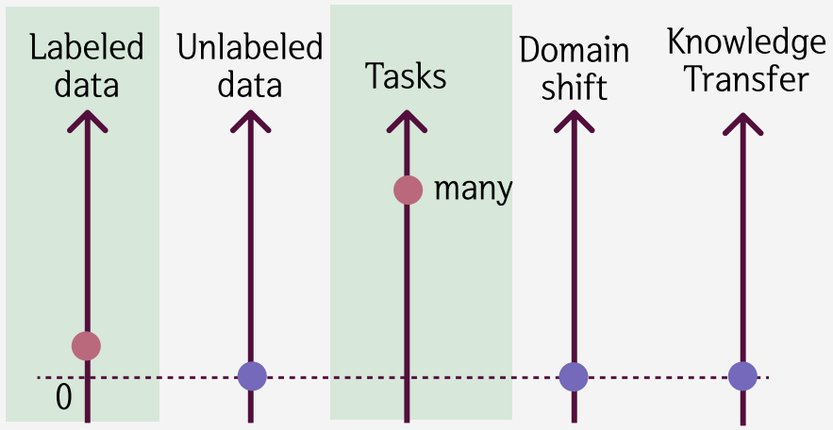
Considering pre-existing paradigms



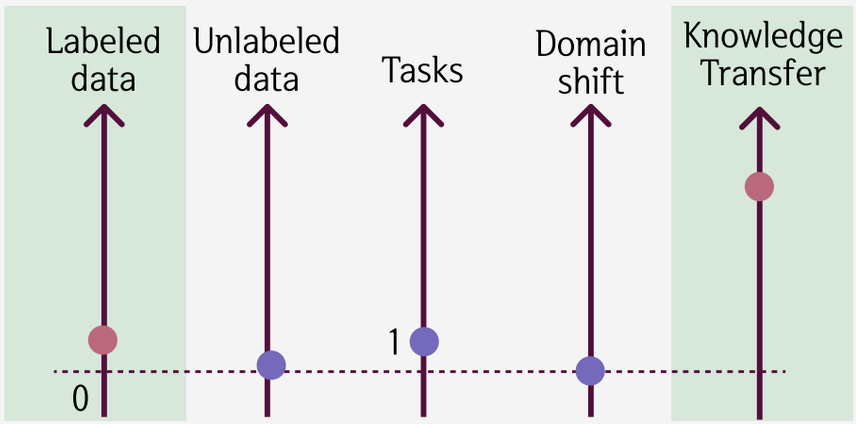
Semi-supervised Learning



Unsupervised Learning



Few-shot Learning



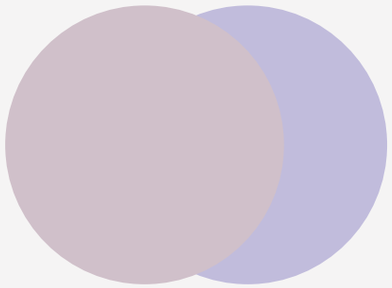
Few-shot Fine-tuning

-> We opt for popular and pre-existing paradigms.

Label-efficient Learning paradigms

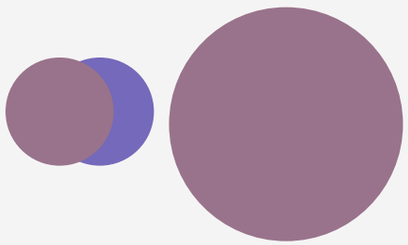
Supervised Learning

Labeled data



Inputs Labels

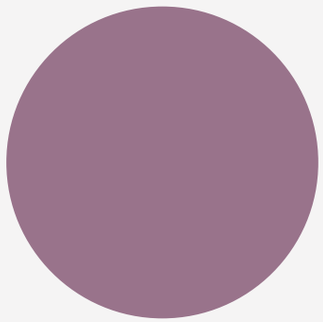
Semi-supervised Leaning



Labeled data

Unlabeled data

Unsupervised Learning

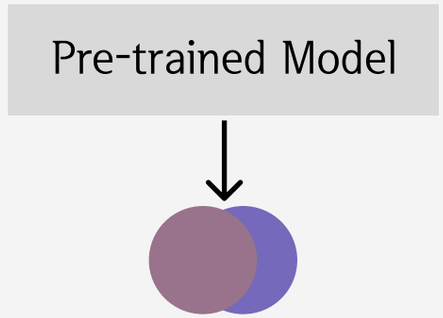


Unlabeled data

Few-show Learning

Per-task Labeled data

Few-Sample Fine-Tuning



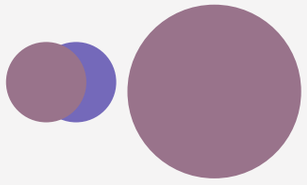
Labeled data

But for which applications/tasks?

Task for each contribution

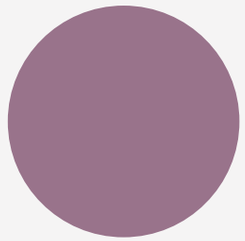
- Different Modalities: vision and language.
- Different Levels of Abstractions: image/document and instance/pixel level.

Semi-supervised Leaning



Labeled data Unlabeled data

Unsupervised Learning

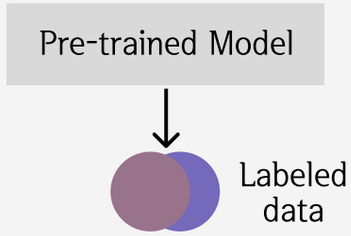


Unlabeled data

Few-show Learning

Per-task Labeled data

Few-Sample Fine-Tuning



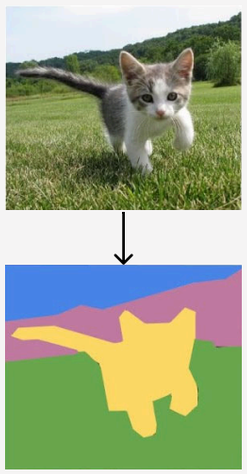


Image Segmentation

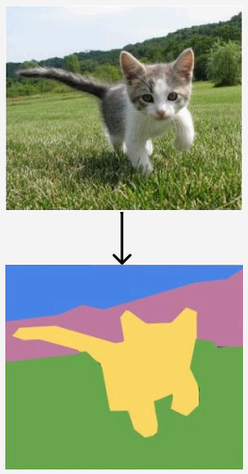


Image Segmentation

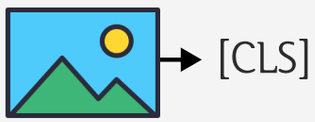
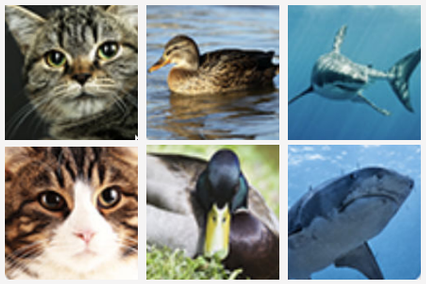
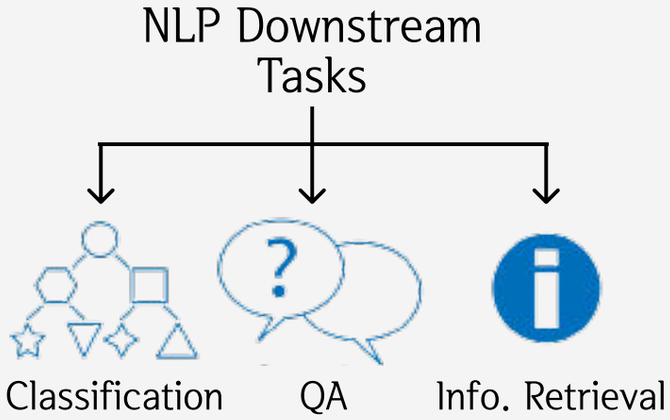


Image Classification

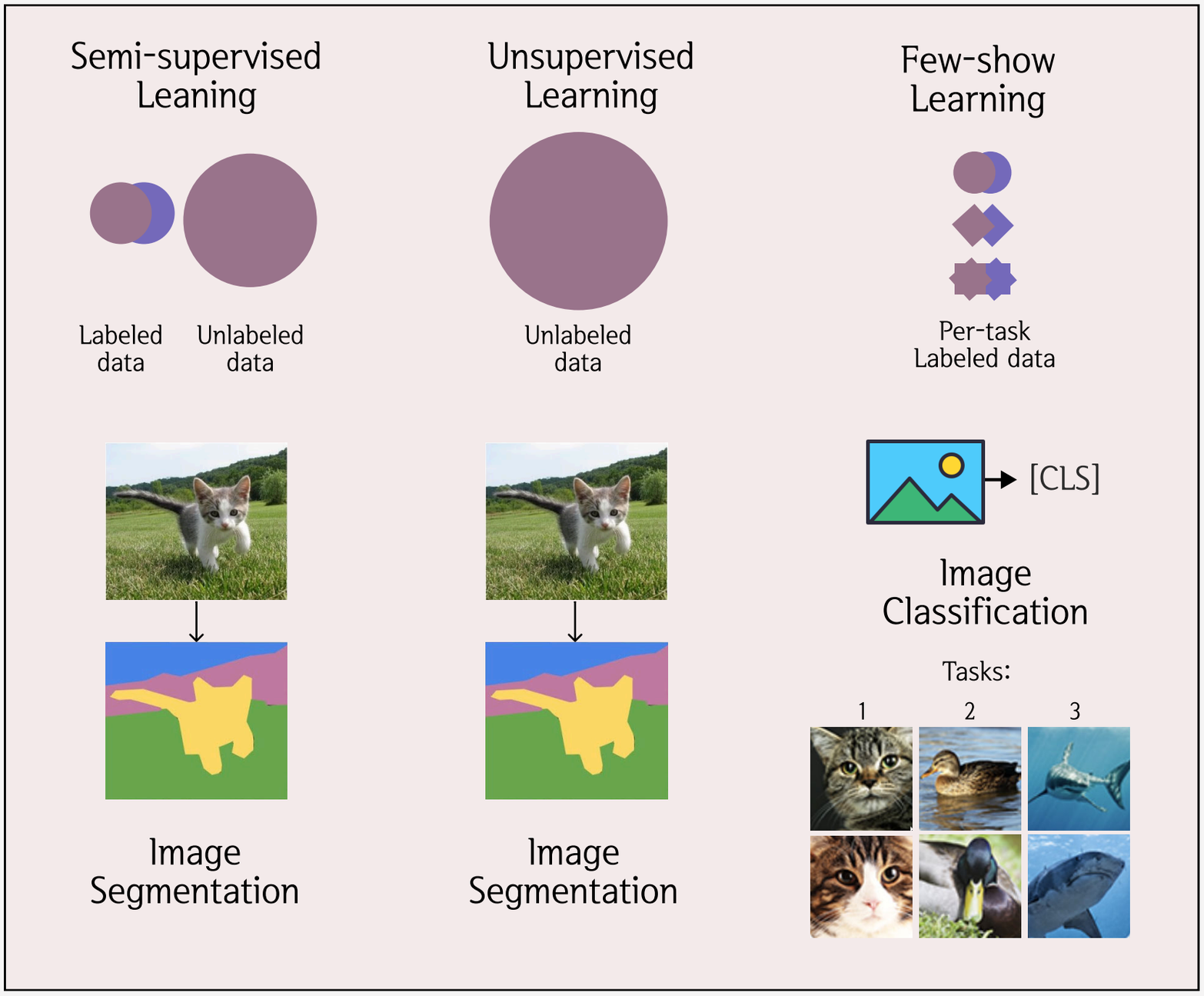
Tasks:

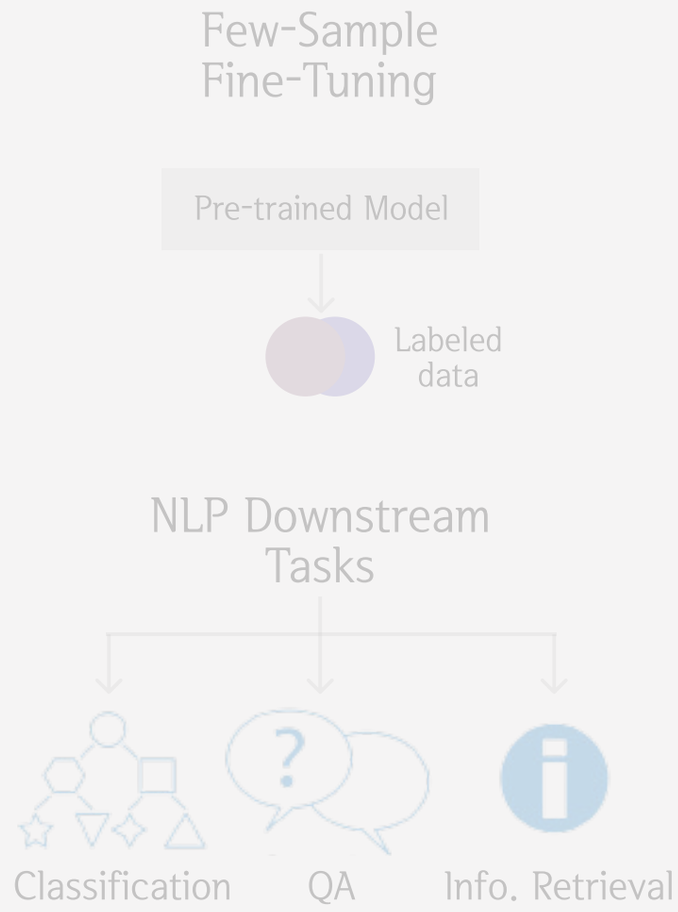




Task for each contribution

- Different Modalities: vision and language.
- Different Levels of Abstractions: image/document and instance/pixel level.

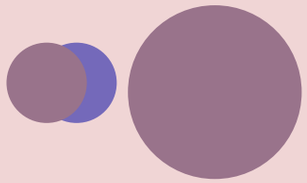




The focus of this presentation

Contributions

Semi-supervised Leaning

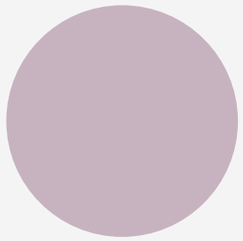


Labeled data

Unlabeled data

Image Segmentation

Unsupervised Learning



Unlabeled data

Image Segmentation

Few-show Learning

Per-task Labeled data

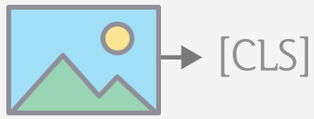
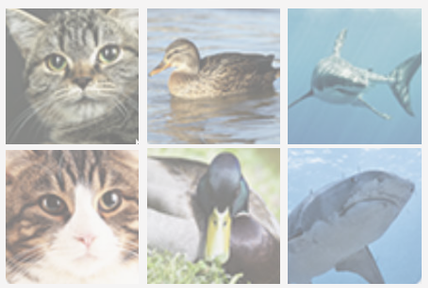


Image Classification

Tasks:

1 2 3

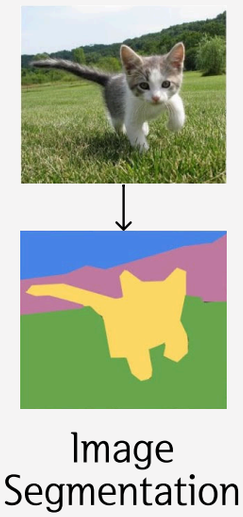


Contribution 1

Cross-consistency Training (CCT)

Paradigm:
Semi-supervised Learning

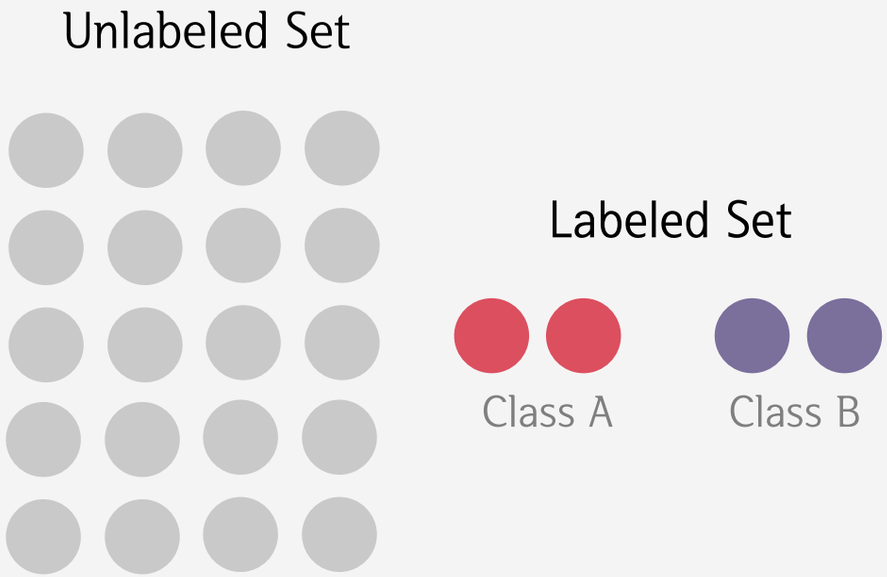
Task:
Image Segmentation



Paper: Semi-supervised semantic segmentation with cross-consistency training, CVPR 2020.
Code: https://github.com/yassouali/CCT

Cross-Consistency Training

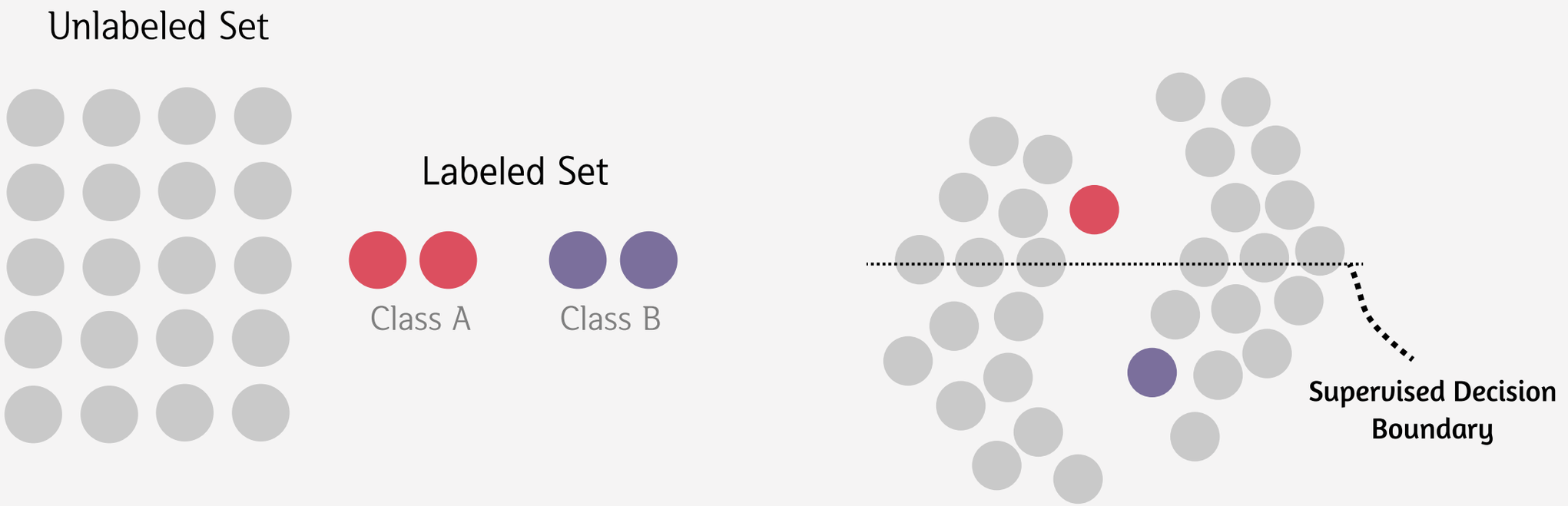
Introduction to Semi-Supervised Learning (SSL)



In SSL, we have two training subsets, a small labeled set, and a larger unlabeled set.

Cross-Consistency Training

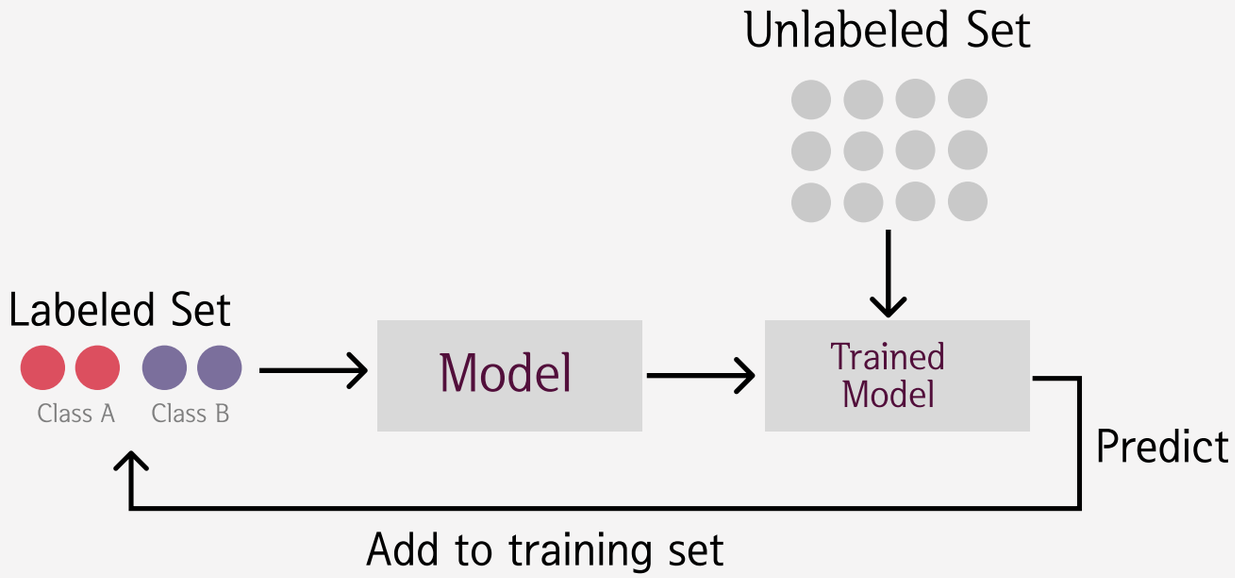
Introduction to Semi-Supervised Learning (SSL)



How can we obtain a better model than the one obtained using the labeled set only?

Cross-Consistency Training

Intro to SSL: Popular methods

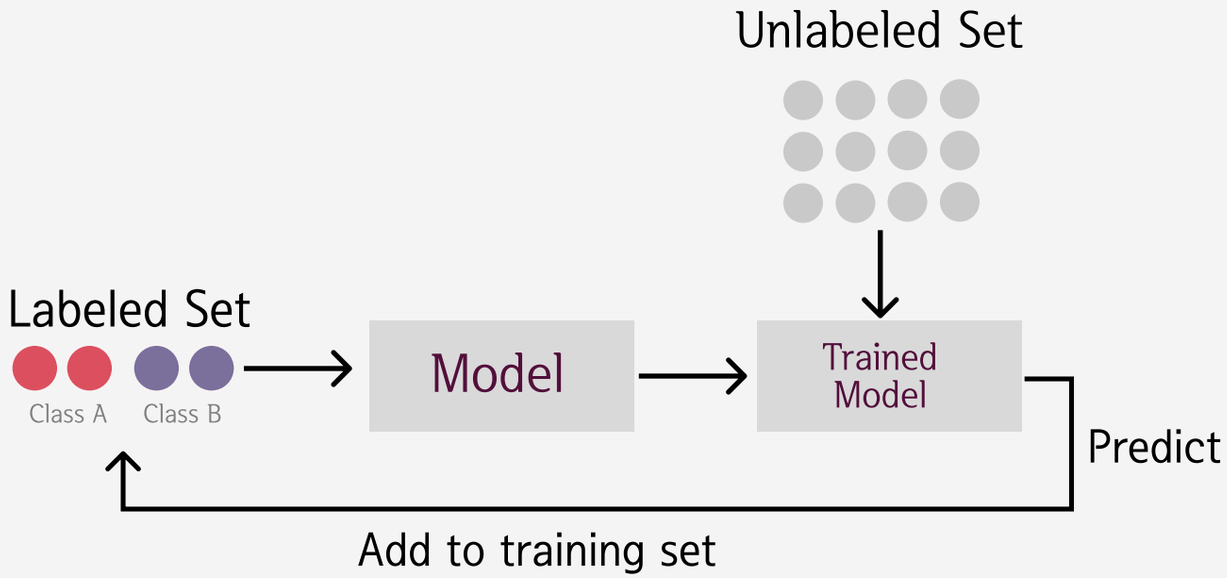


Self-training/Pseudo Labeling

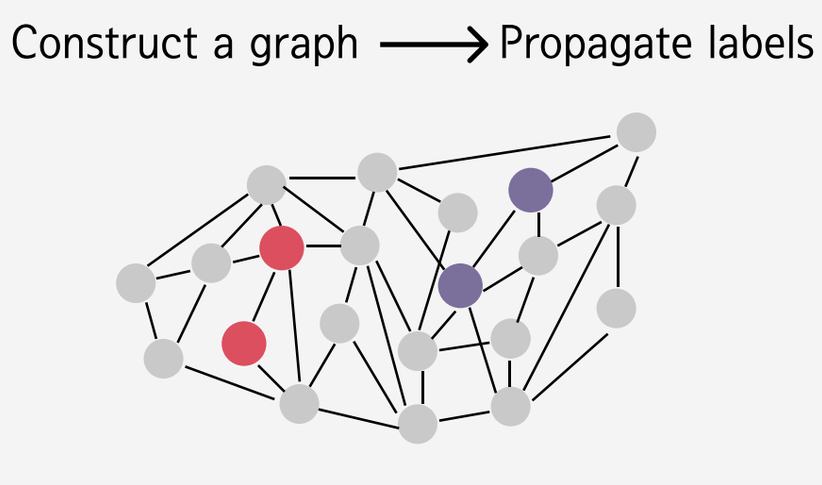
Pseudo-label: The simple and efficient SSL method for deep neural network. Lee et al. 2013.

Cross-Consistency Training

Intro to SSL: Popular methods



Self-training/Pseudo Labeling

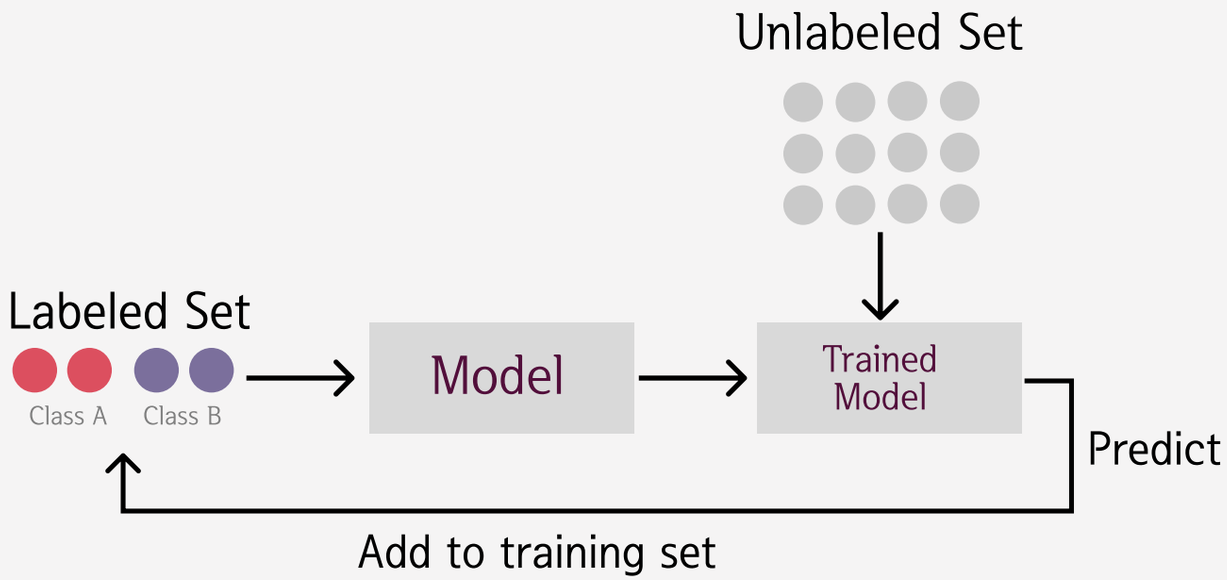


Label Propagation

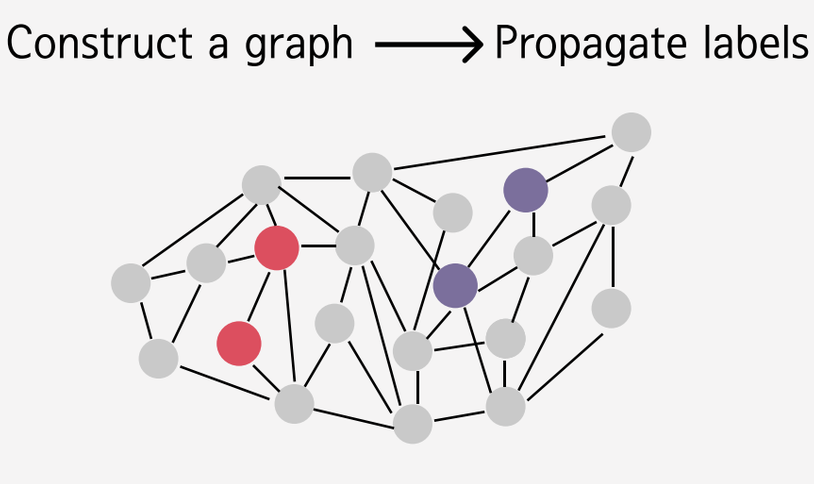
Semi-supervised classification with graph convolutional networks. Kipf et al, 2016.
Pseudo-label: The simple and efficient SSL method for deep neural network. Lee et al. 2013.

Cross-Consistency Training

Intro to SSL: Popular methods



Self-training/Pseudo Labeling



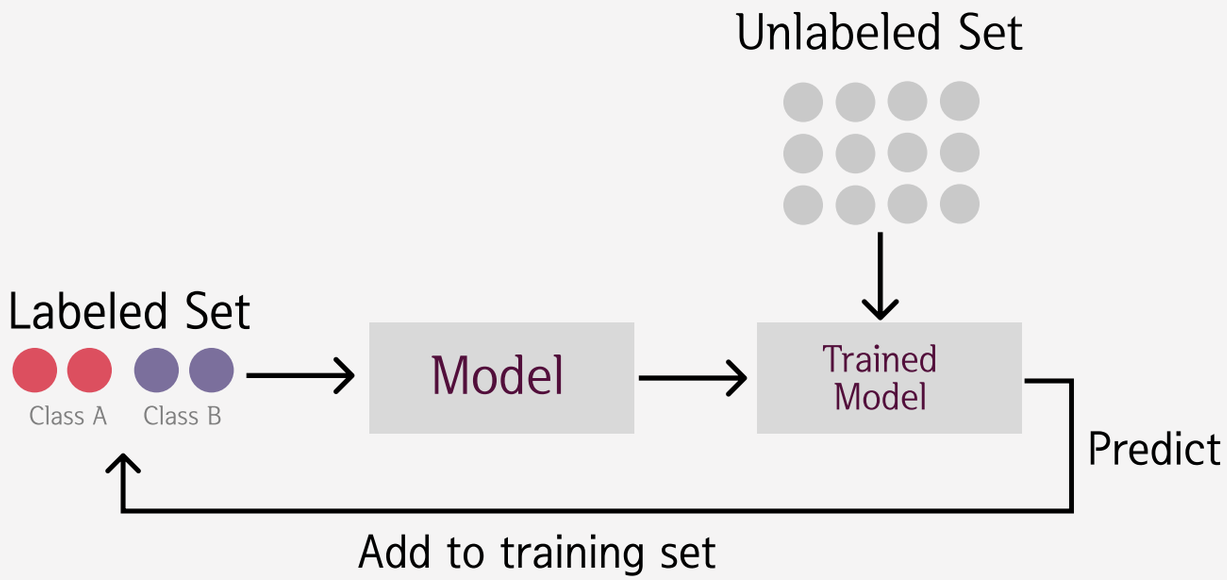
Label Propagation

Consistency Training/Regularization

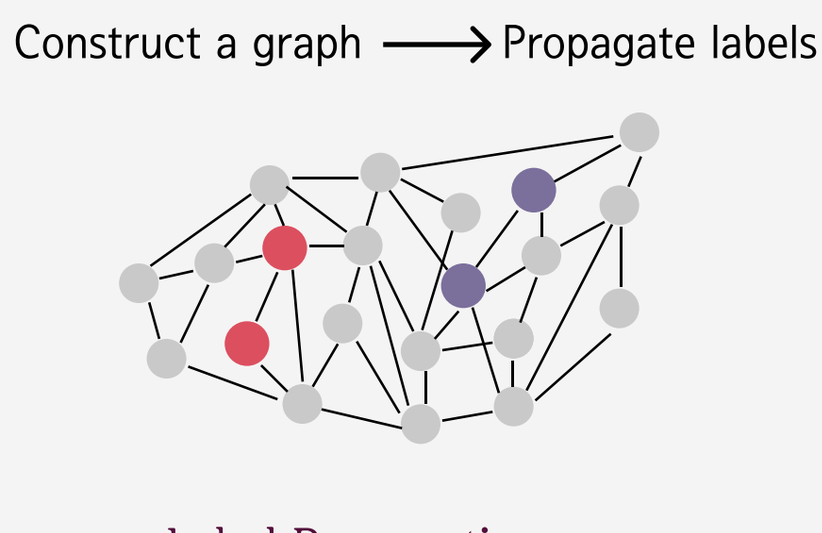
Semi-supervised classification with graph convolutional networks. Kipf et al, 2016.
Pseudo-label: The simple and efficient SSL method for deep neural network. Lee et al. 2013.
Mean teachers are better role models. Tarvainen et al, 2017

Cross-Consistency Training

Intro to SSL: Popular methods

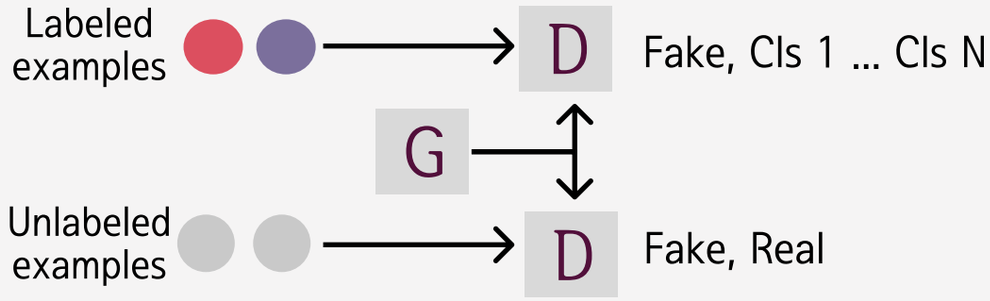


Self-training/Pseudo Labeling



Label Propagation

Consistency Training/Regularization

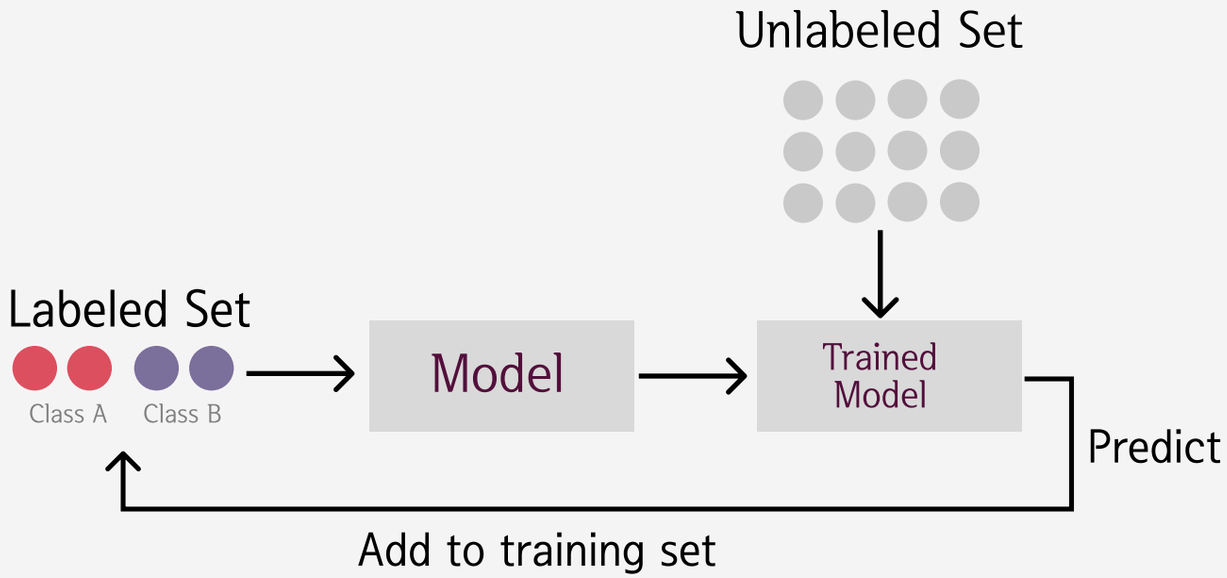


Generative Modelling for SSL

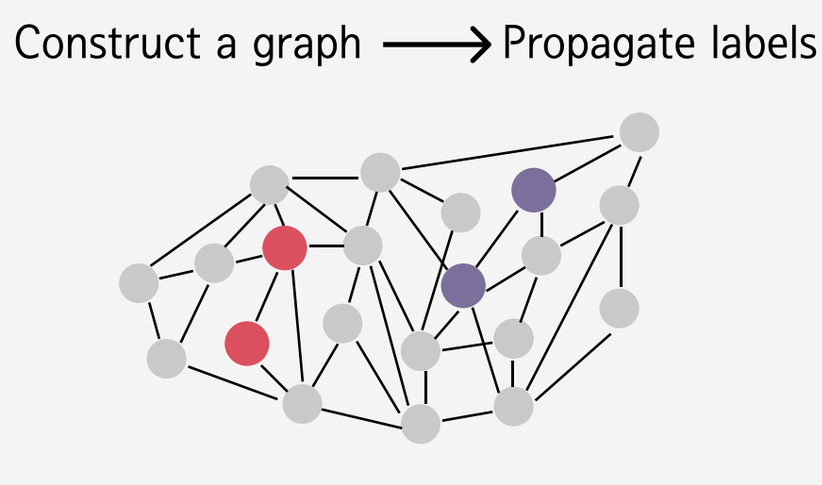
Semi-supervised classification with graph convolutional networks. Kipf et al, 2016.
Semi-Supervised Learning with Generative Adversarial Networks. Odena, 2017.
Pseudo-label: The simple and efficient SSL method for deep neural network. Lee et al. 2013.
Mean teachers are better role models. Tarvainen et al, 2017

Cross-Consistency Training

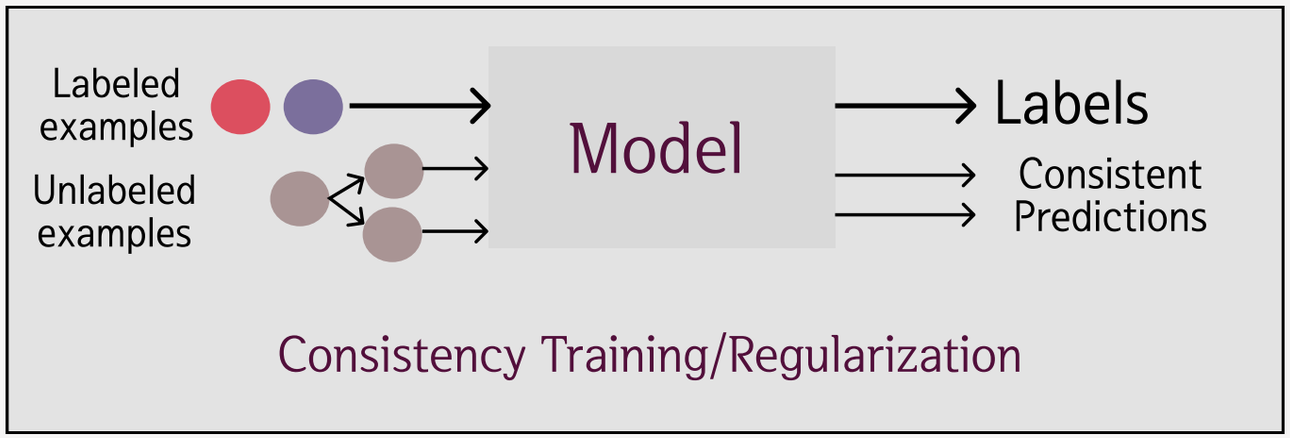
Intro to SSL: Popular methods



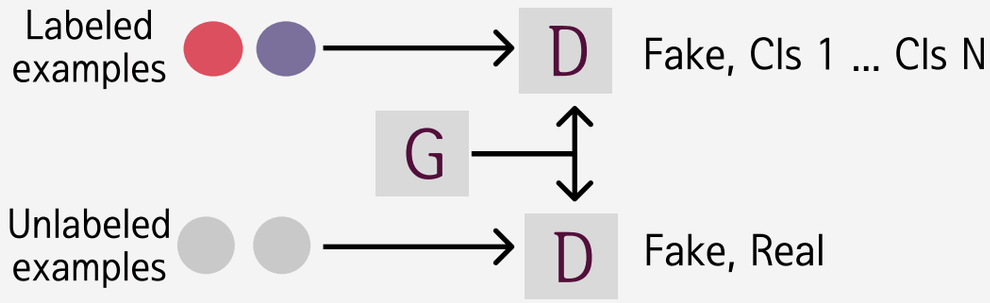
Self-training/Pseudo Labeling



Label Propagation



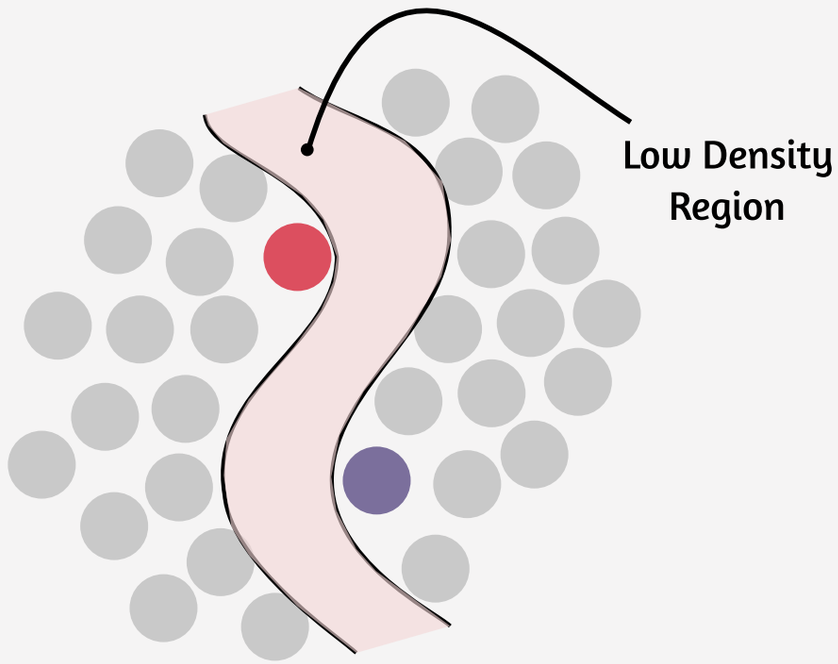
Consistency Training/Regularization



Generative Modelling for SSL

Cross-Consistency Training

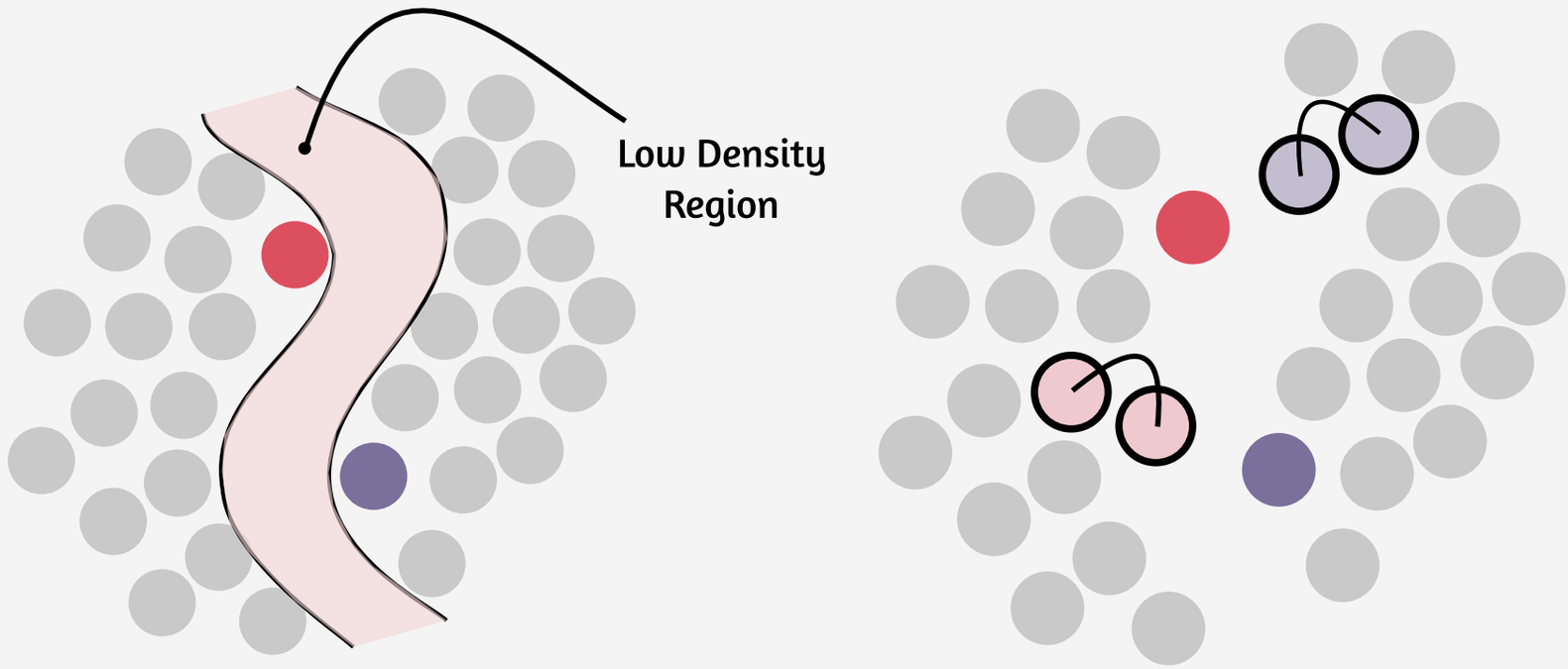
Introduction to Consistency Training



Cluster assumption: data naturally forms class relevant clusters or groups.

Cross-Consistency Training

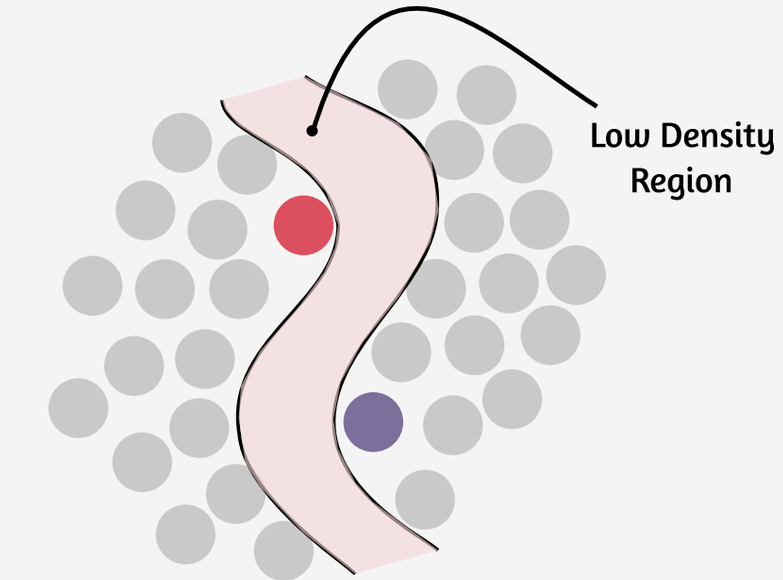
Introduction to Consistency Training

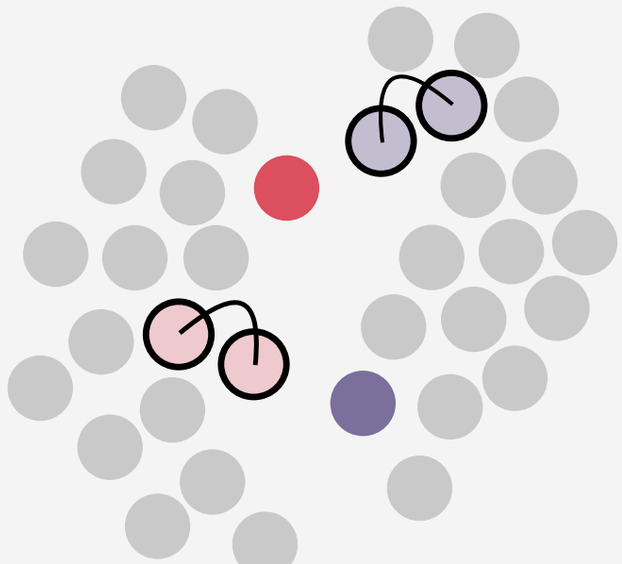


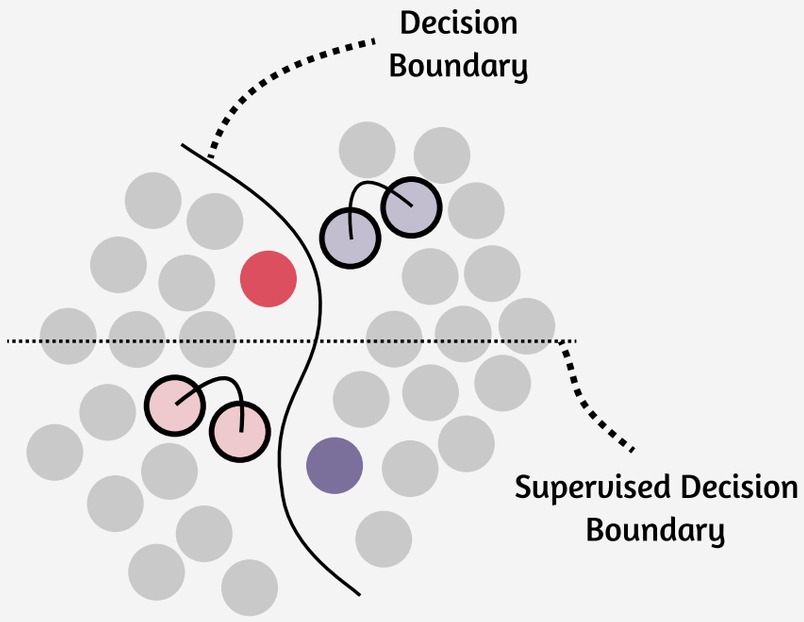
As a result, the class of a given input is preserved after a small perturbation.

Cross-Consistency Training

Introduction to Consistency Training







- model f
- unlabeled example x_u
- perturbation ϵ

Objective: $f(x_u) = f(x_u + \epsilon)$

$$L = L_s + L_u$$
$$L_u = MSE(f(x_u), f(x_u + \epsilon))$$

Consistency Training: force the model to have consistent prediction over the unlabeled data points, with and without a small perturbation.

Cross-Consistency Training

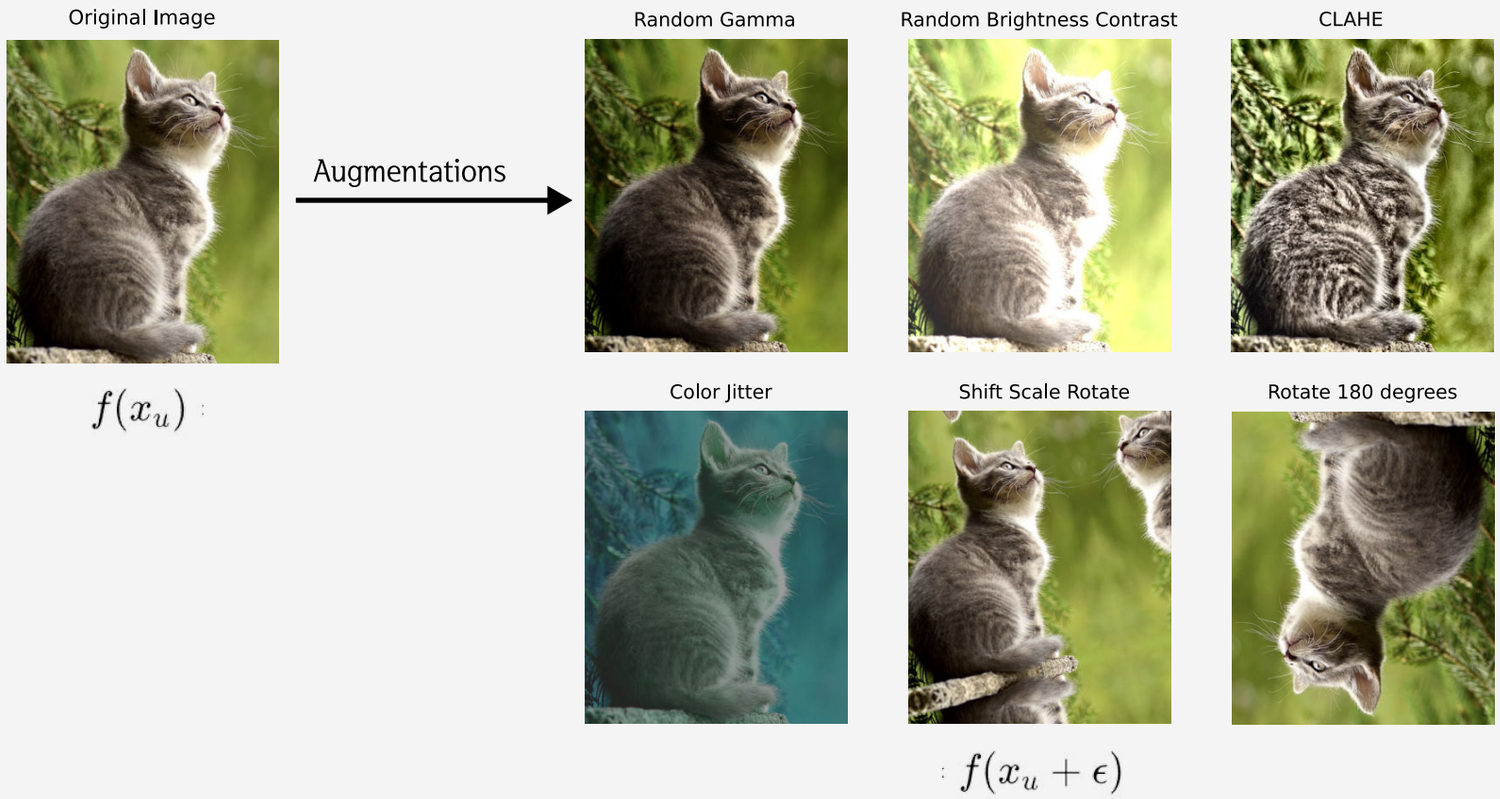
Cluster assumption at the image level

Original Image

Example: for the task of image classification, we can apply augmentations at the input level.

Cross-Consistency Training

Cluster assumption at the image level



And even with strong augmentation, the semantic content is preserved at the image level.

Cross-Consistency Training

Cluster assumption at the pixel level

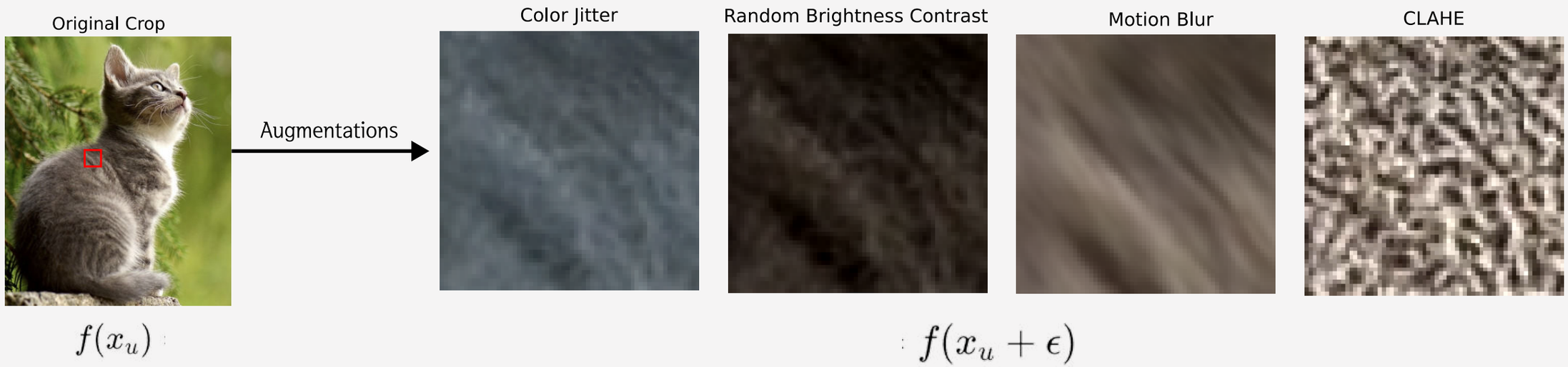
Original Crop

$f(x_u)$

But we are interested in pixel-wise classification, is the cluster assumption maintained at the pixel level?

Cross-Consistency Training

Cluster assumption at the pixel level



At the pixel, the texture/content changes with each augmentation.

For image segmentation:

- Augmentations need to be carefully chosen.
- Model needs to be carefully designed to account for large receptive fields.

Can we do better?

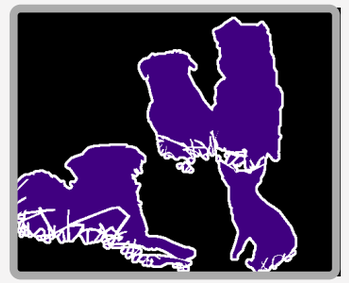
Cross-Consistency Training

Investigating the cluster assumption at the encoder's level

Input Image

$\mathcal{C}(x_{ij})$

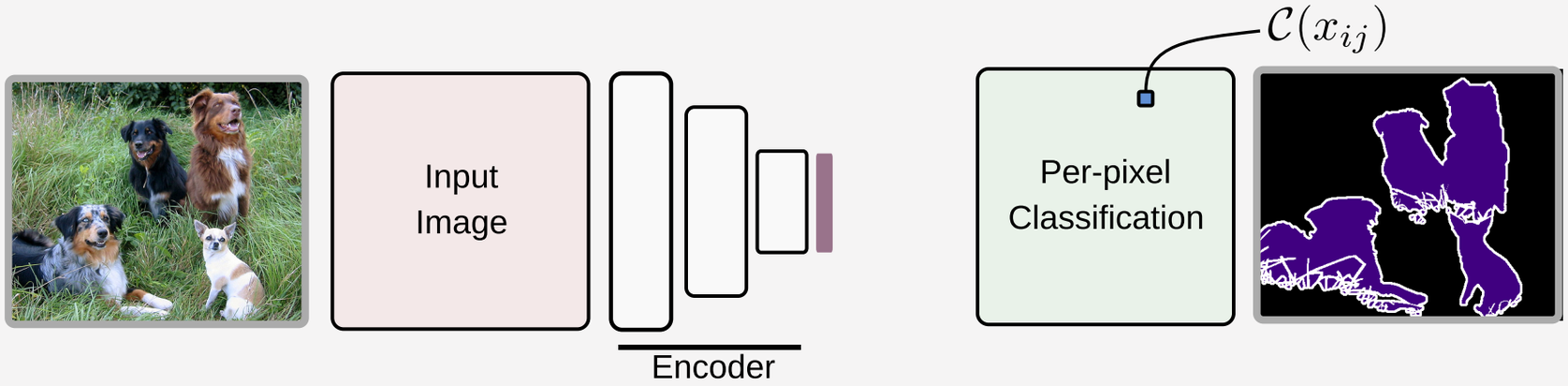
Per-pixel Classification



How about at the representation level?

Cross-Consistency Training

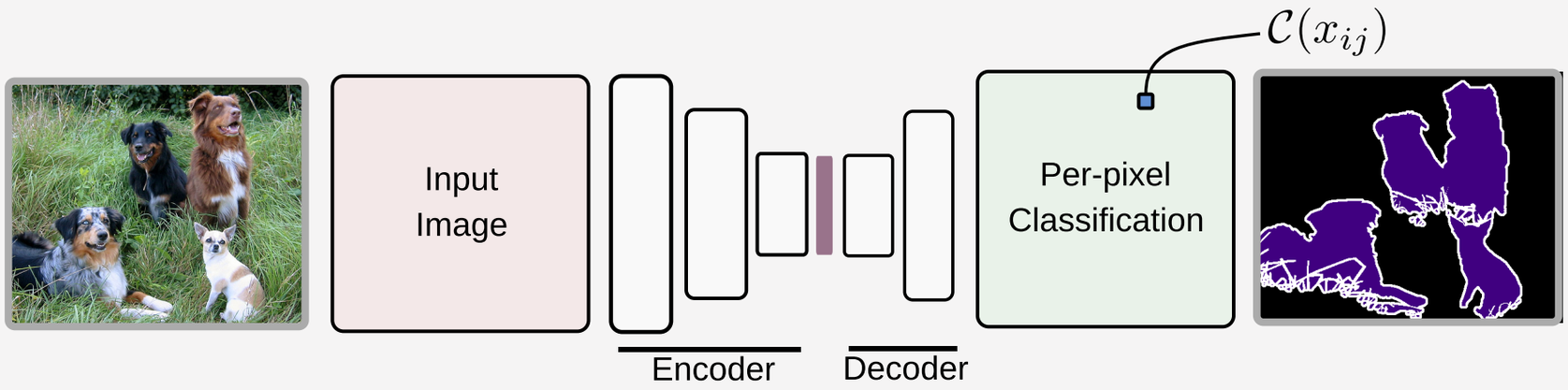
Investigating the cluster assumption at the encoder's level



How about at the representation level?

Cross-Consistency Training

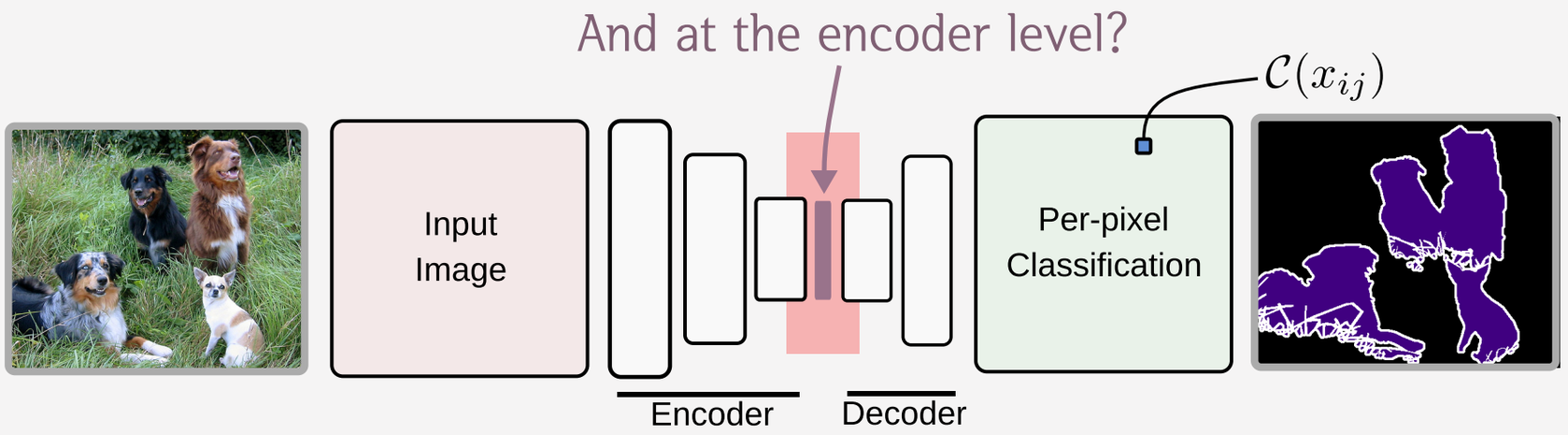
Investigating the cluster assumption at the encoder's level



How about at the representation level?

Cross-Consistency Training

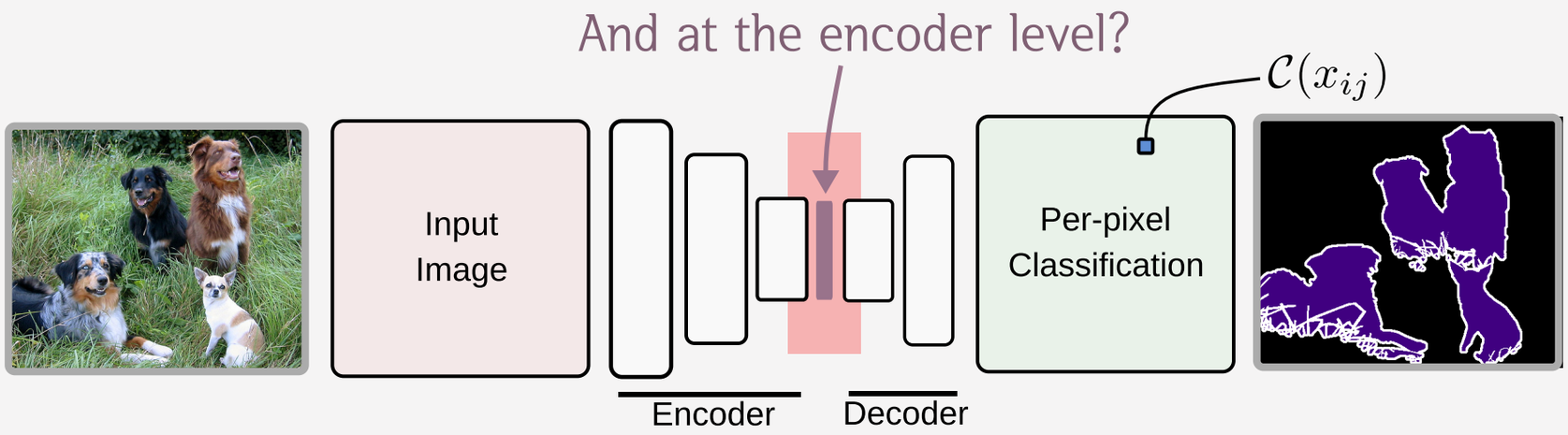
Investigating the cluster assumption at the encoder's level

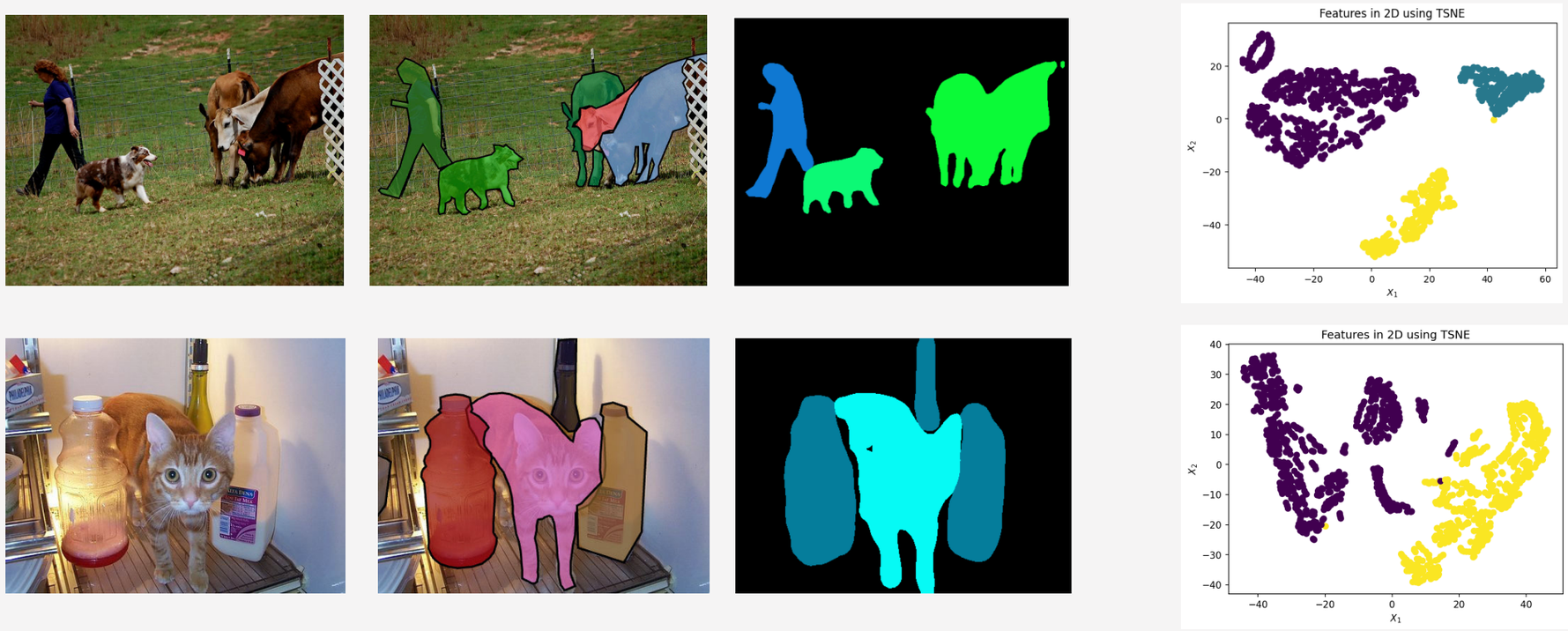


How about at the representation level?

Cross-Consistency Training

Investigating the cluster assumption at the encoder's level

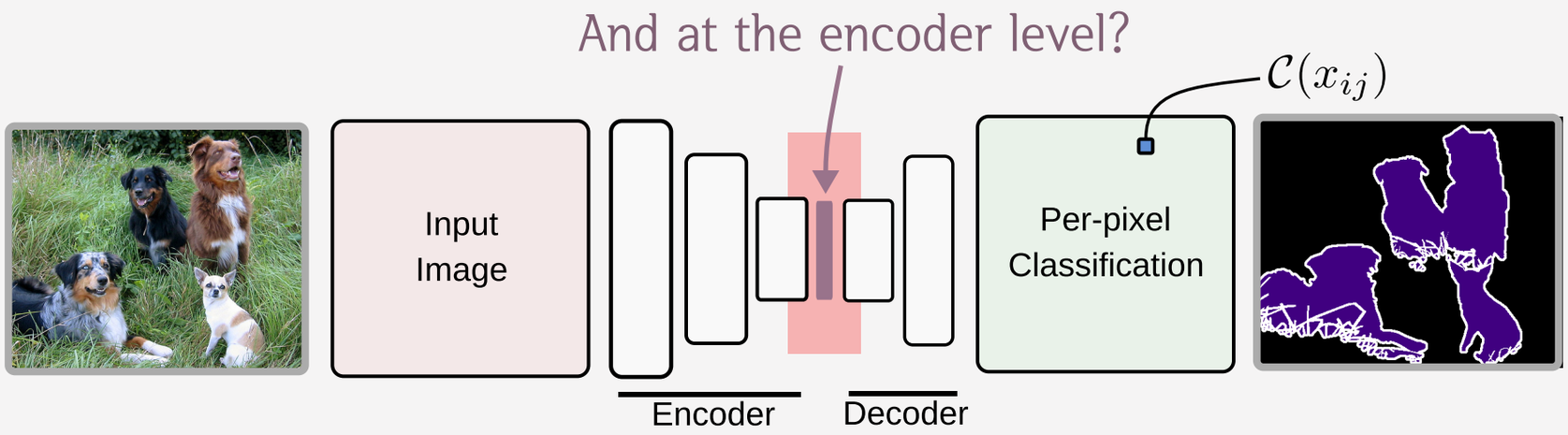


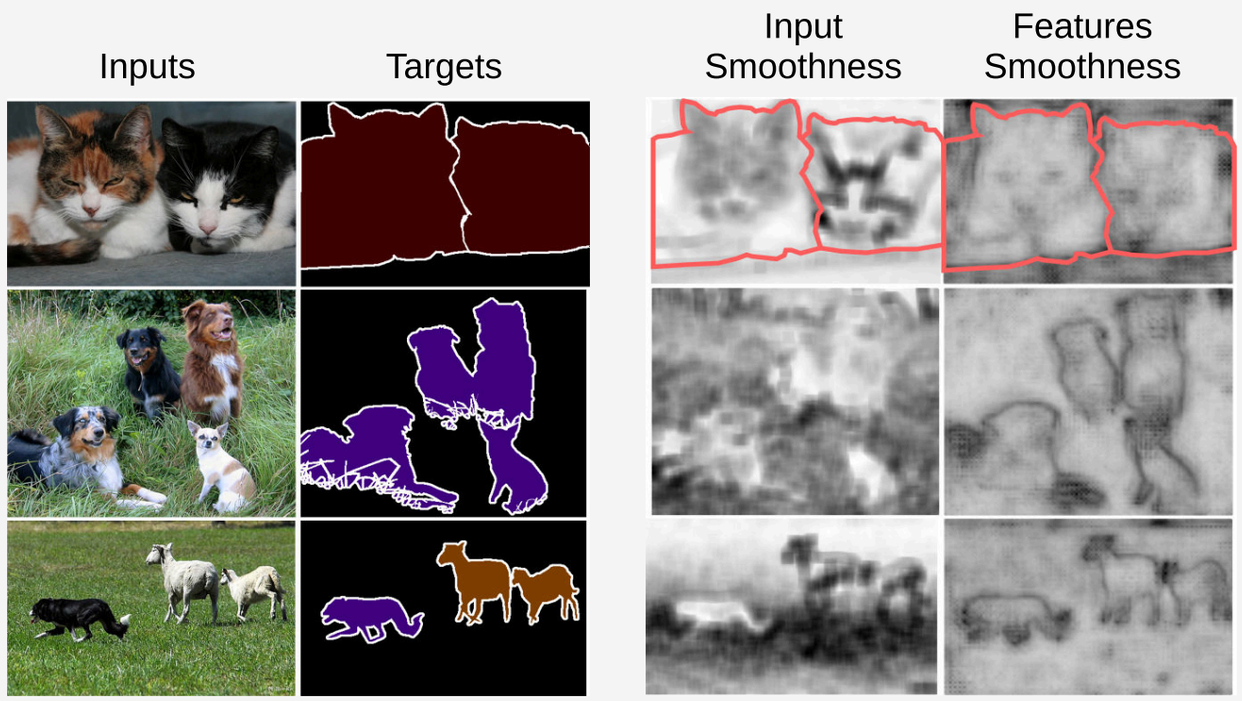


We observe that the encode produces class relevant clustered features.

Cross-Consistency Training

Investigating the cluster assumption at the encoder's level

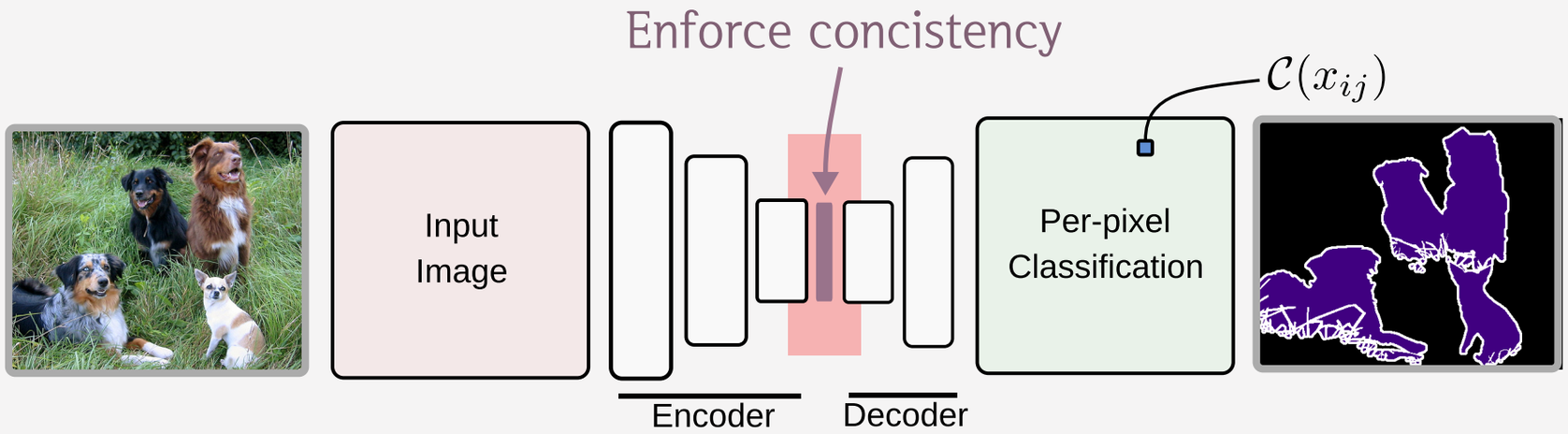




We observe that the encode produces class relevant clustered features.

Cross-Consistency Training

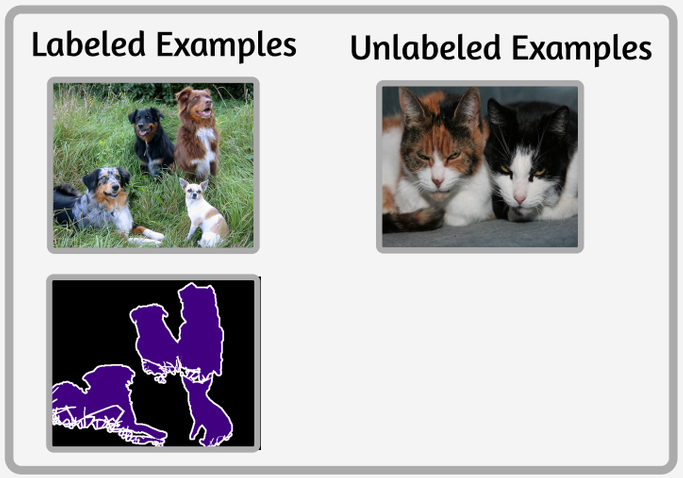
Investigating the cluster assumption at the encoder's level



-> Cross-consistency Training (CCT): enforce the consistency of predictions over the encoder's representations.

Cross-Consistency Training

CCT: step by step



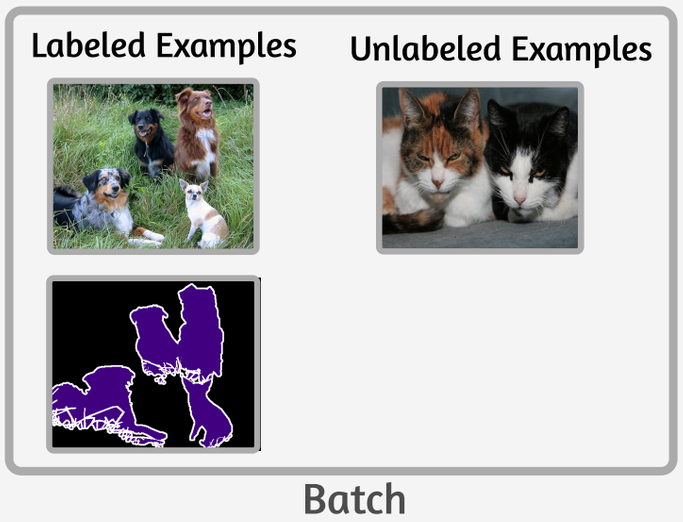
Input: x Outputs: y

Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$

Segmentation Network: f

Cross-Consistency Training

CCT: step by step

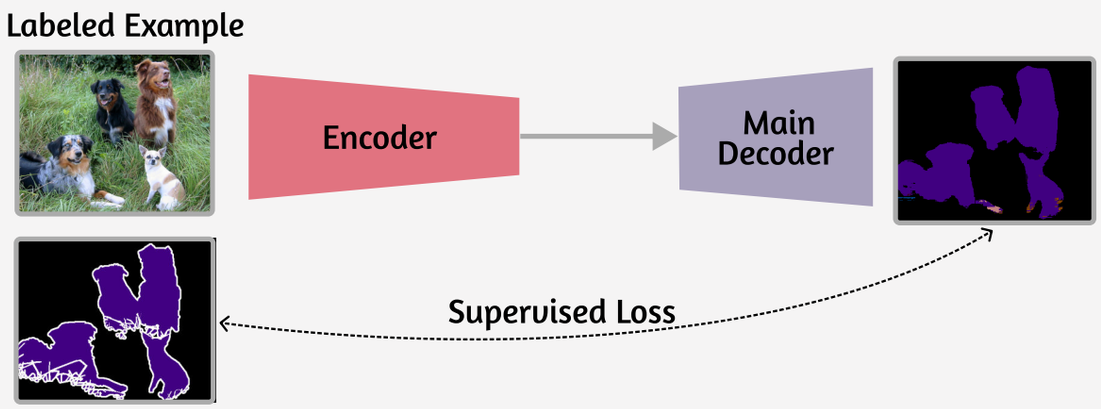


Cross-Entropy: $\mathbf{H}$
Input: x Outputs: y

Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$

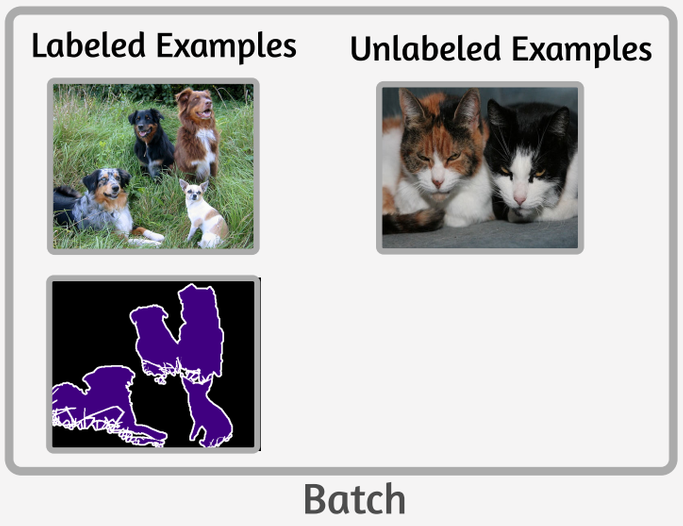
Segmentation Network: f

$$\text{Supervised Loss: } \mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}\left(y_i, f\left(\mathbf{x}_i^l\right)\right)$$



Cross-Consistency Training

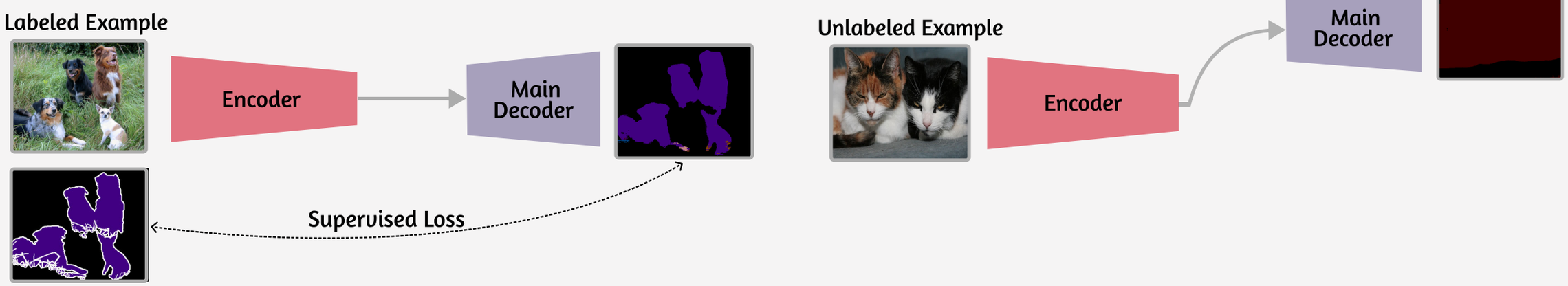
CCT: step by step



Cross-Entropy: $\mathbf{H}$
Input: x Outputs: y
Distance Measure: $\mathbf{d}$
Encoder's Output: $\mathbf{z}$

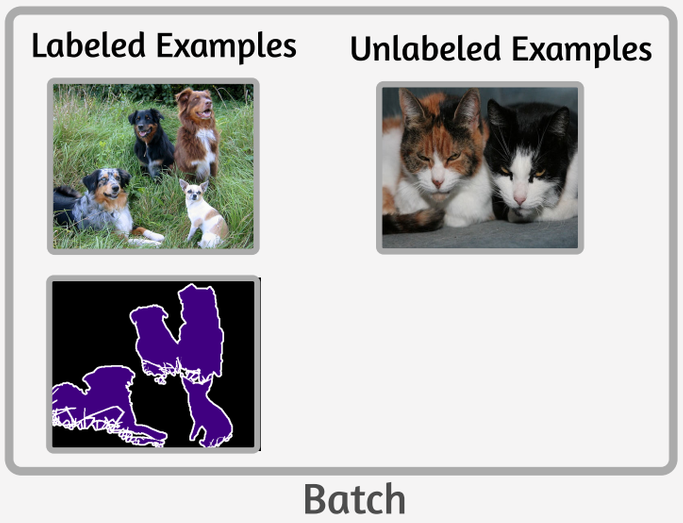
Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$
Segmentation Network: f
Aux. Decoders: g

$$\text{Supervised Loss: } \mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}\left(y_i, f\left(\mathbf{x}_i^l\right)\right)$$



Cross-Consistency Training

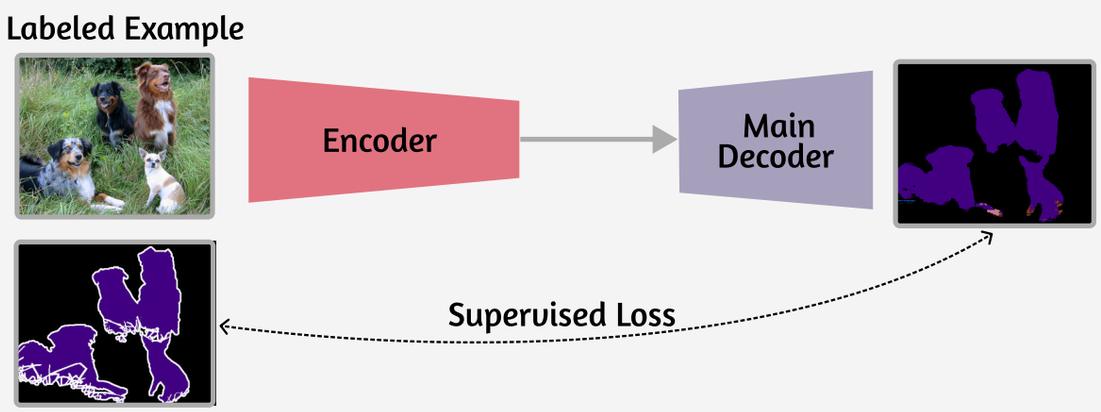
CCT: step by step



Cross-Entropy: $\mathbf{H}$
Input: x Outputs: y
Distance Measure: $\mathbf{d}$
Encoder's Output: $\mathbf{z}$

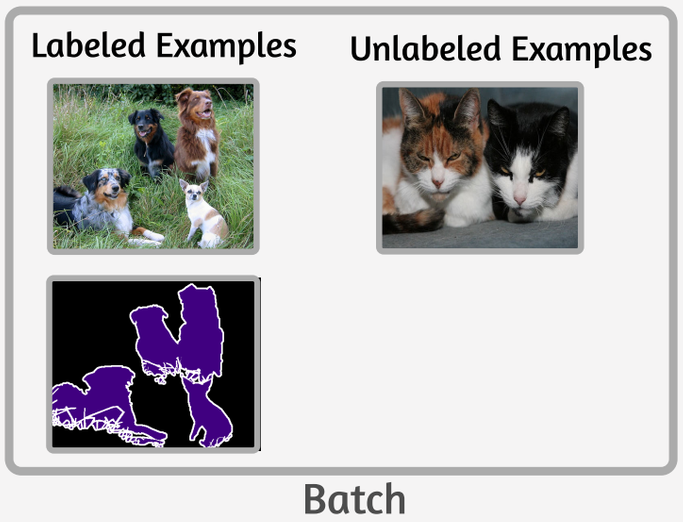
Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$
Segmentation Network: f
Aux. Decoders: g

$$\text{Supervised Loss: } \mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}\left(y_i, f\left(\mathbf{x}_i^l\right)\right)$$



Cross-Consistency Training

CCT: step by step

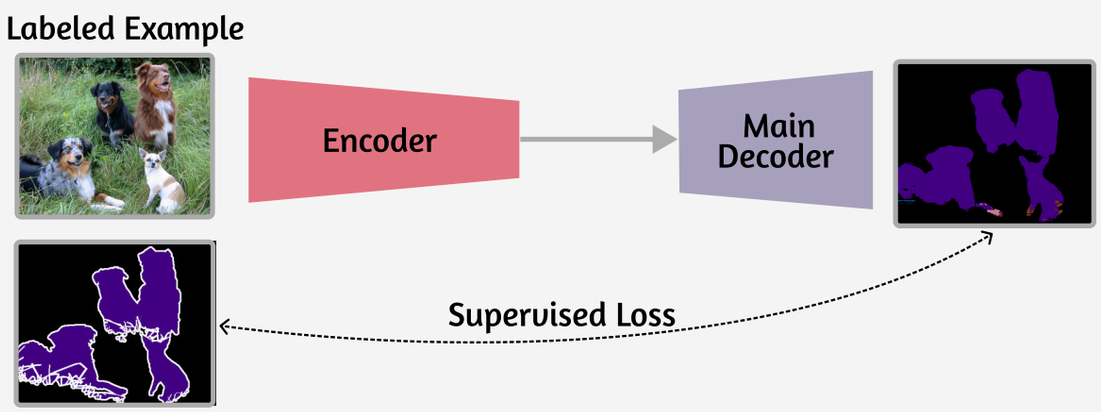


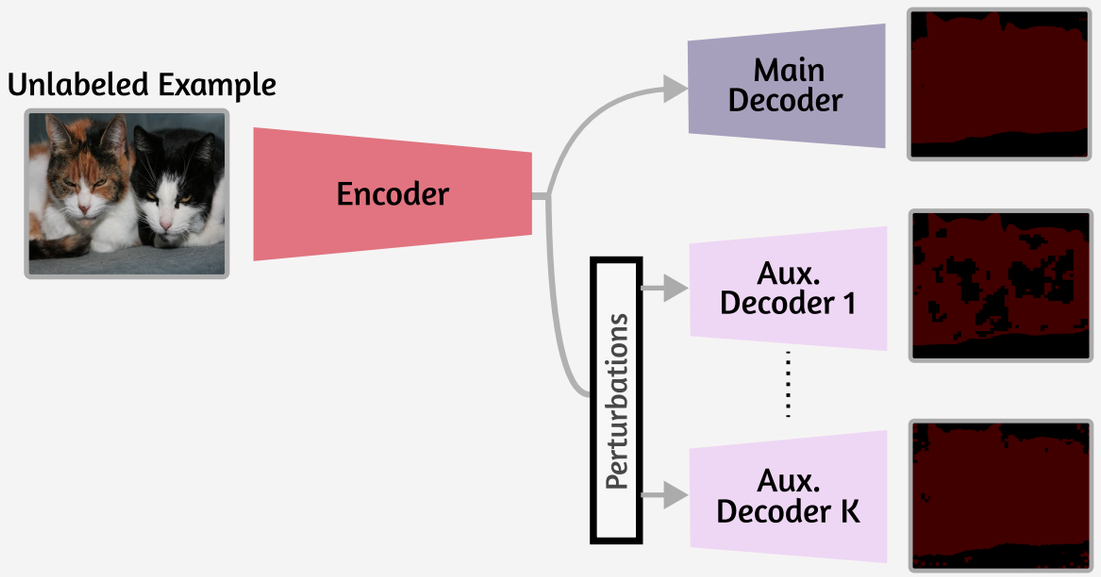
Cross-Entropy: $\mathbf{H}$
Input: x Outputs: y
Distance Measure: $\mathbf{d}$
Encoder's Output: $\mathbf{z}$

Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$
Segmentation Network: f
Aux. Decoders: g

$$\text{Supervised Loss: } \mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}\left(y_i, f\left(\mathbf{x}_i^l\right)\right)$$

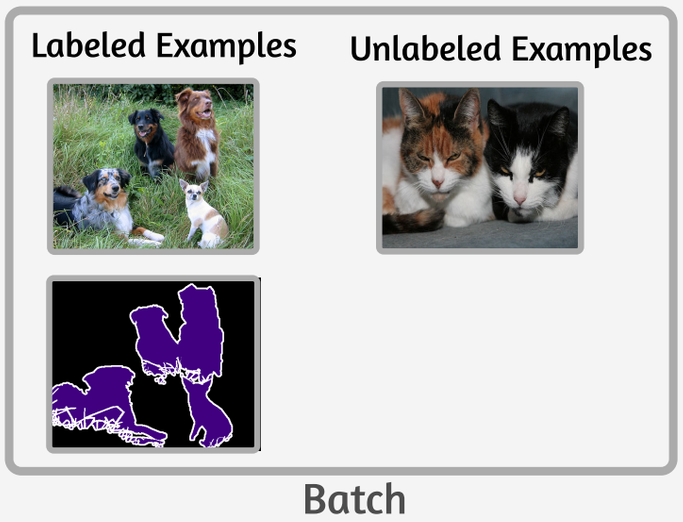
$$\text{Unsupervised Loss: } \mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \frac{1}{K} \sum_{\mathbf{x}_i^u \in \mathcal{D}_u} \sum_{k=1}^{K} \mathbf{d}\left(g\left(\mathbf{z}_i\right), g_a^k\left(\mathbf{z}_i\right)\right)$$





Cross-Consistency Training

CCT: step by step

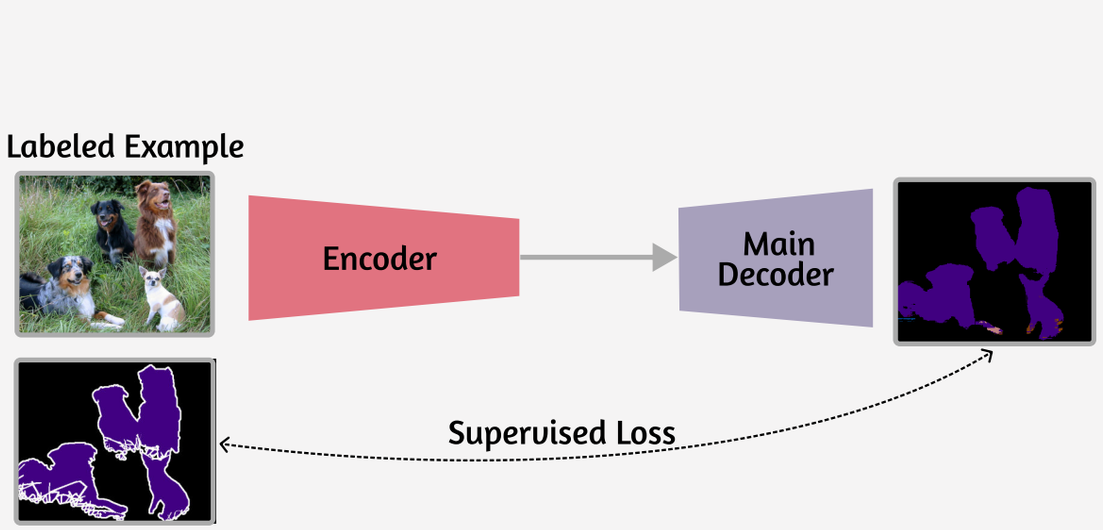


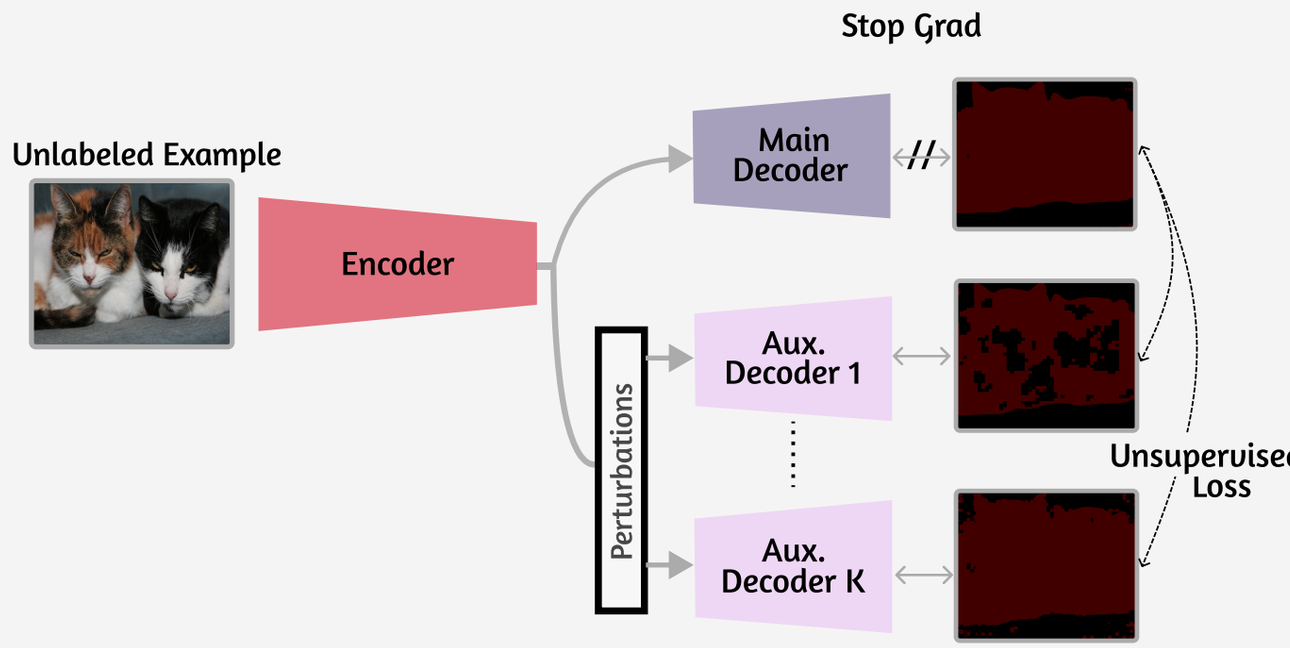
Cross-Entropy: $\mathbf{H}$
Input: x Outputs: y
Distance Measure: $\mathbf{d}$
Encoder's Output: $\mathbf{z}$

Unlabeled Set: $\mathcal{D}_u$
Labeled Set: $\mathcal{D}_l$
Segmentation Network: f
Aux. Decoders: g

$$\text{Supervised Loss: } \mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}\left(y_i, f\left(\mathbf{x}_i^l\right)\right)$$

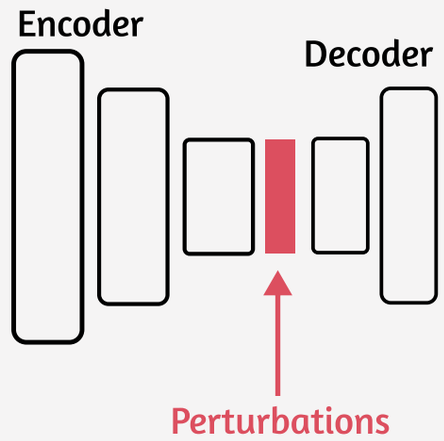
$$\text{Unsupervised Loss: } \mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \frac{1}{K} \sum_{\mathbf{x}_i^u \in \mathcal{D}_u} \sum_{k=1}^{K} \mathbf{d}\left(g\left(\mathbf{z}_i\right), g_a^k\left(\mathbf{z}_i\right)\right)$$





Cross-Consistency Training

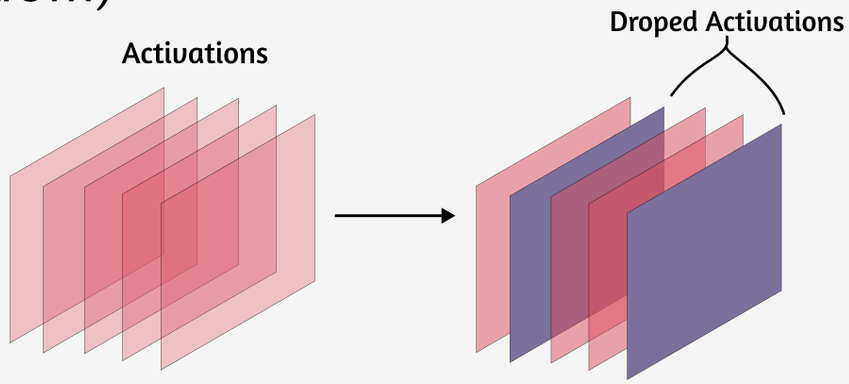
Perturbations at the encoder level



Cross-Consistency Training

Perturbations at the encoder level: Random

- Spatial Dropout (Random)

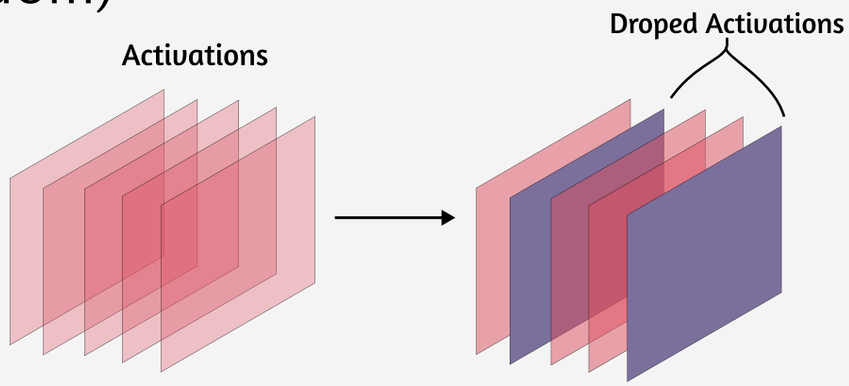


Randomly drop a portion of feature maps.

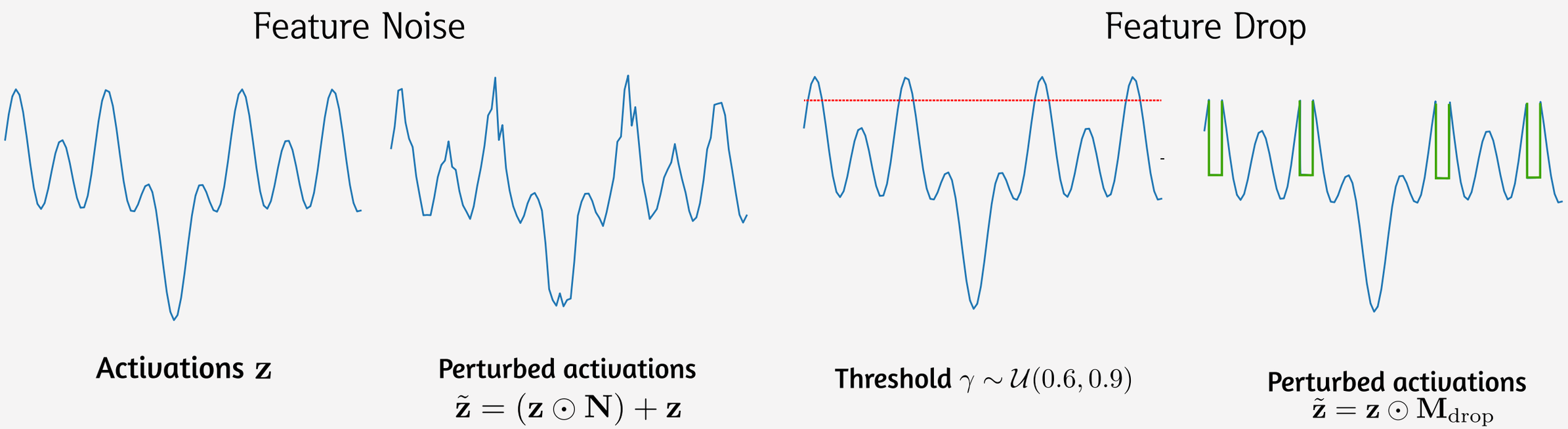
Cross-Consistency Training

Perturbations at the encoder level: random & feature-based

- Spatial Dropout (Random)



- Feature-Based Perturbations

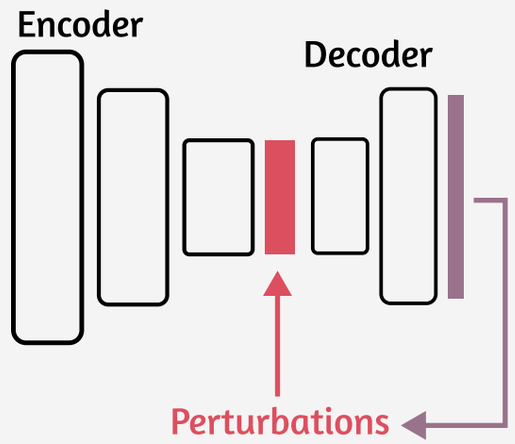


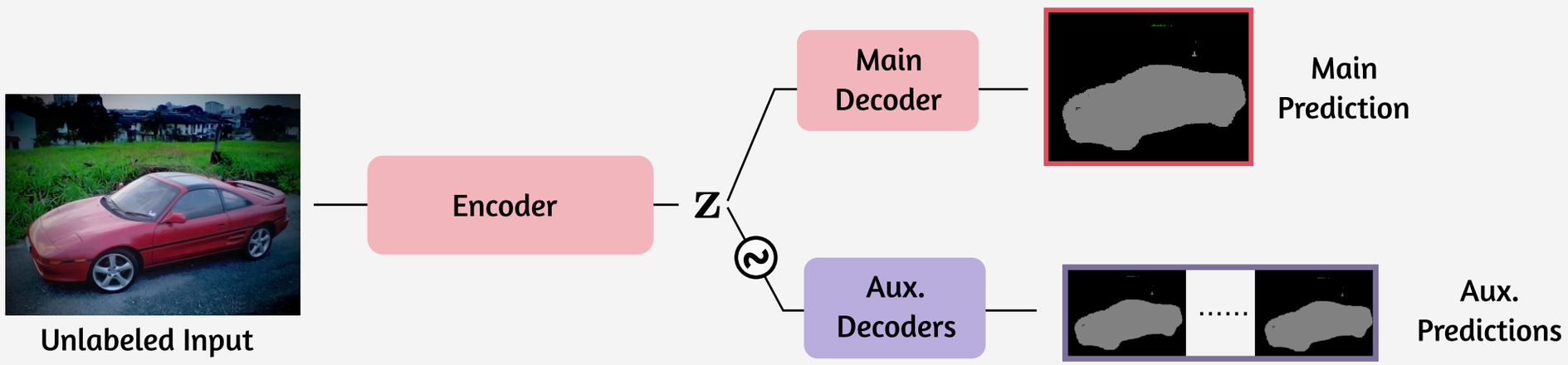
Perturb the activation based on their amplitude.

Cross-Consistency Training

Perturbations at the encoder level: prediction-based

- Prediction-Based Perturbations



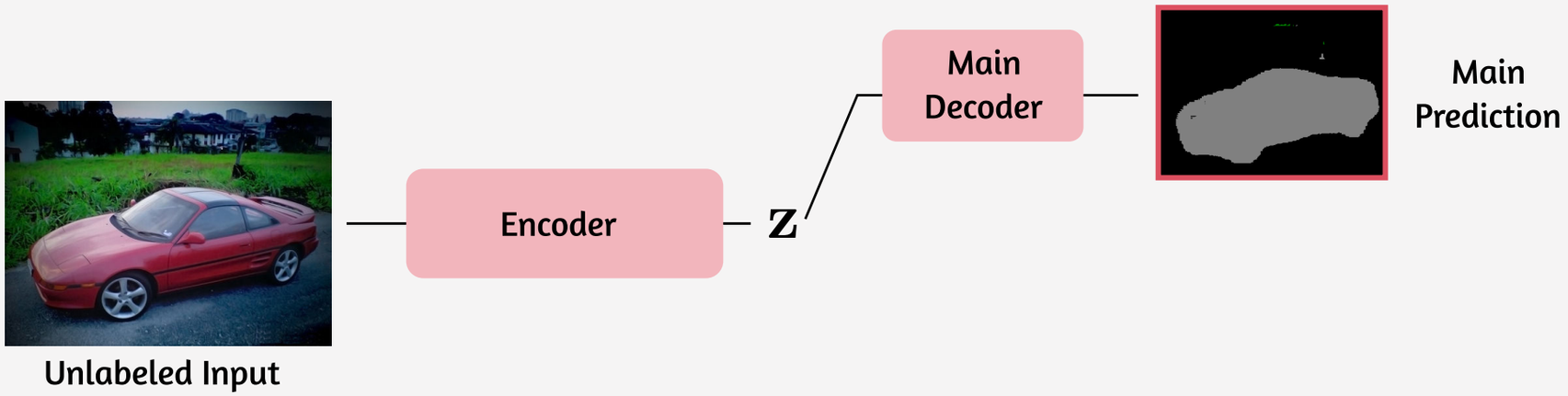


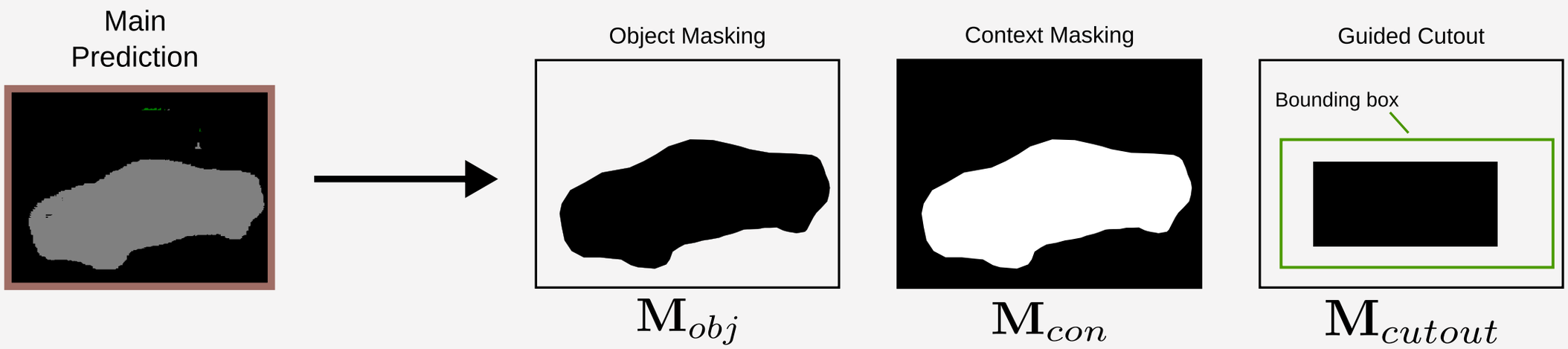
Perturb the encoder's representation based on the obtained predictions.

Cross-Consistency Training

Cross-Consistency Training: Perturbation

- Prediction-Based Perturbations



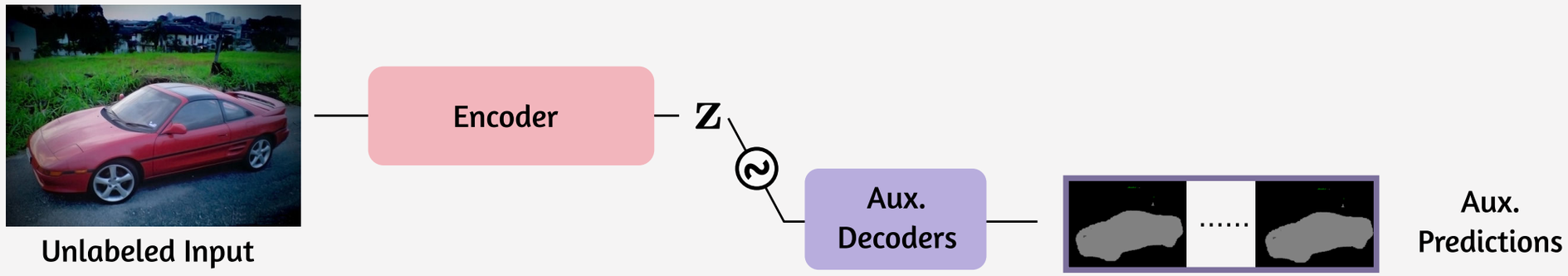


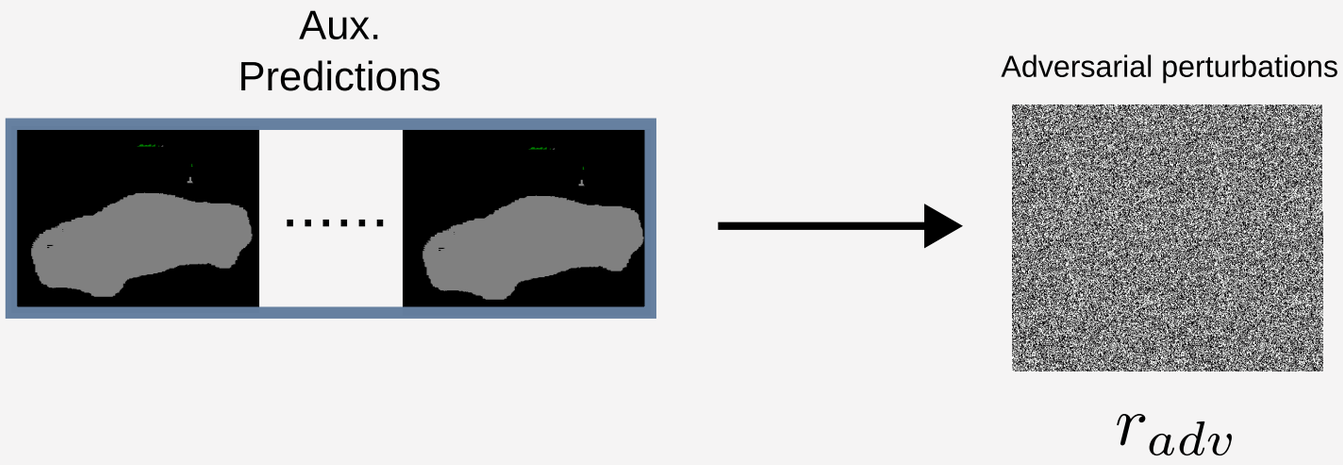
Perturb the encoder's representation based on the obtained predictions ... either the prediction of the main encoder.

Cross-Consistency Training

Cross-Consistency Training: Perturbation

- Prediction-Based Perturbations



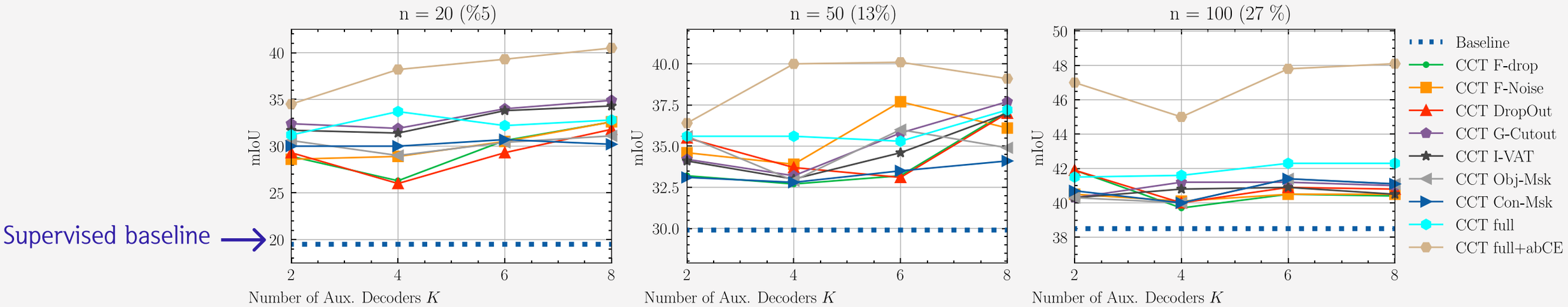


Perturb the encoder's representation based on the obtained predictions ... either the prediction of the main encoder, ... or that of the aux. decoders.

Cross-Consistency Training

Results: Effectiveness of CCT, or encoder-level consistency

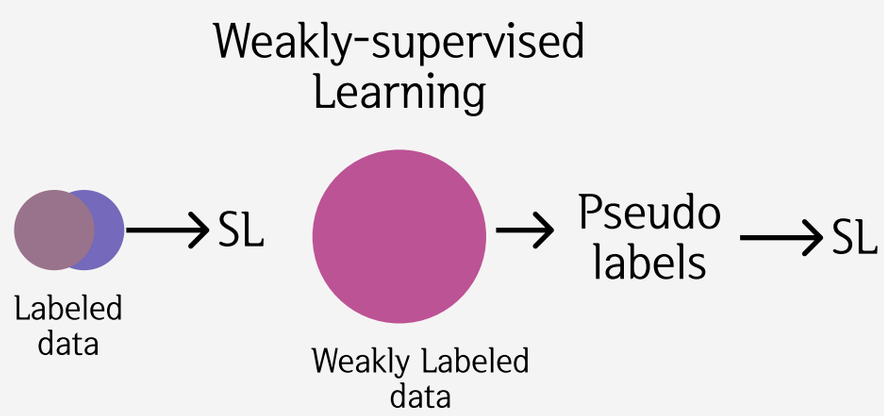
We compare the mIoU obtained with the proposed perturbations with a variable degree of supervision.

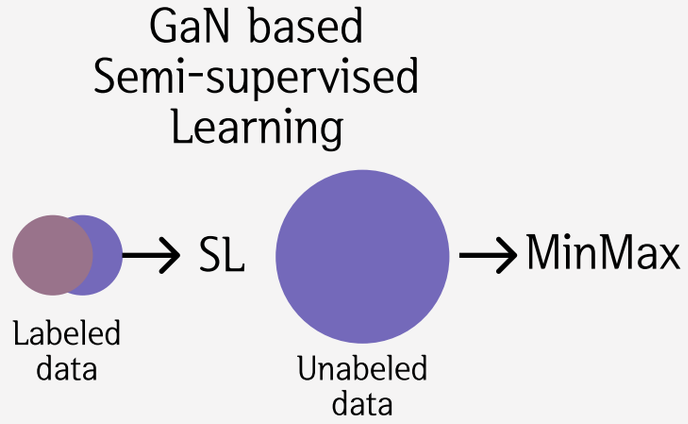


Obtained results confirm the effectiveness of enforcing the consistency over the hidden representations

Cross-Consistency Training

Results: Comparison with SOTA





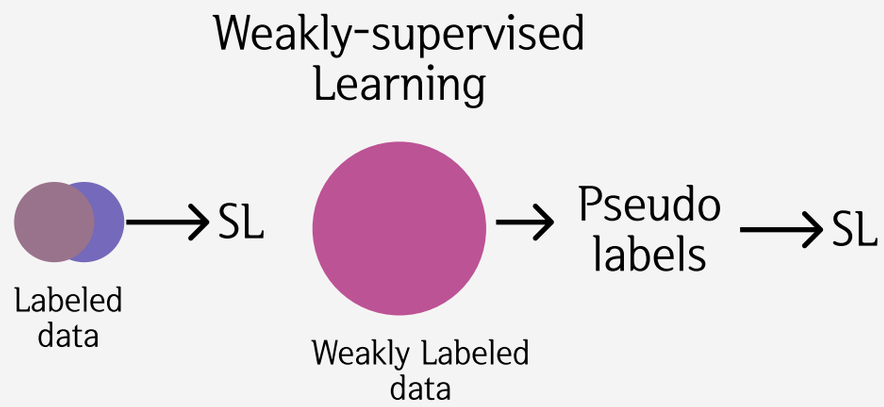
We compare CCT with SOTA SSL segmentation methods, consisting of two types of approaches:

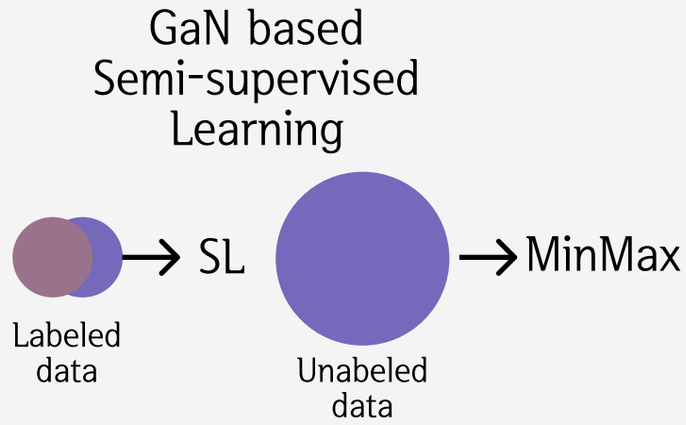
- Weakly-supervised methods.
- GaN based SSL methods.

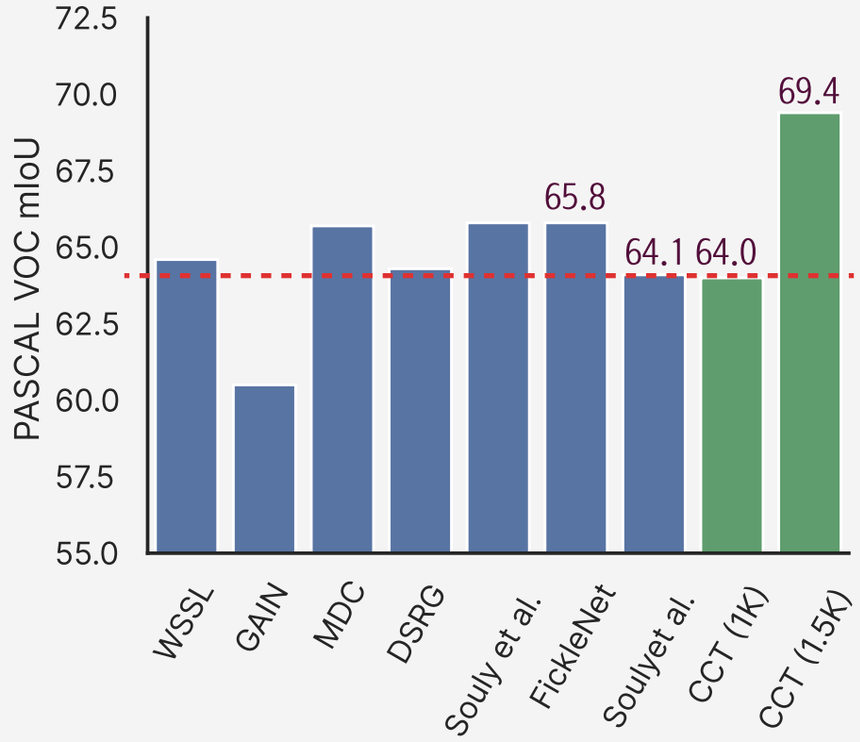
FickleNet: Weakly and Semi-supervised Semantic Image Segmentation. Lee et al. 2019.
Semi supervised semantic segmentation using generative adversarial network. Suly et al. 2017.
Adversarial learning for semi-supervised semantic segmentation. Hung at al, 2018.
Weakly-supervised semantic segmentation network with deep seeded region growing. Huang et al, 2018.

Cross-Consistency Training

Results: Comparison with SOTA



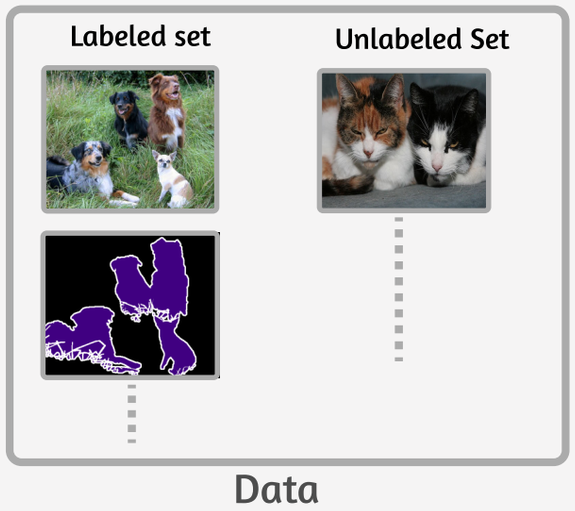




CCT shows notable improvements over previous methods.

Cross-Consistency Training

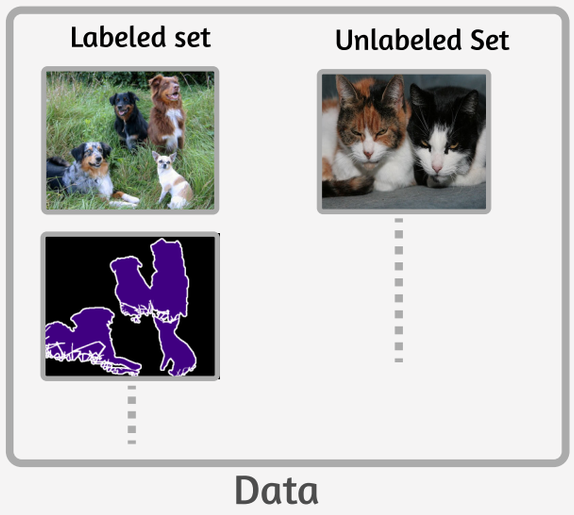
CCT with image-level labels

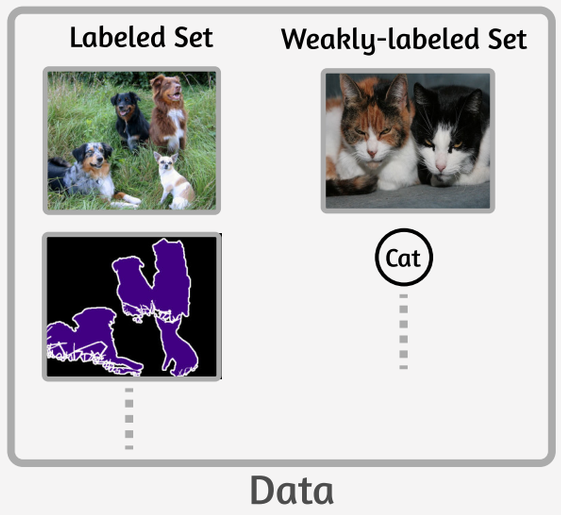


What if instead of unlabeled samples ....

Cross-Consistency Training

CCT with image-level labels

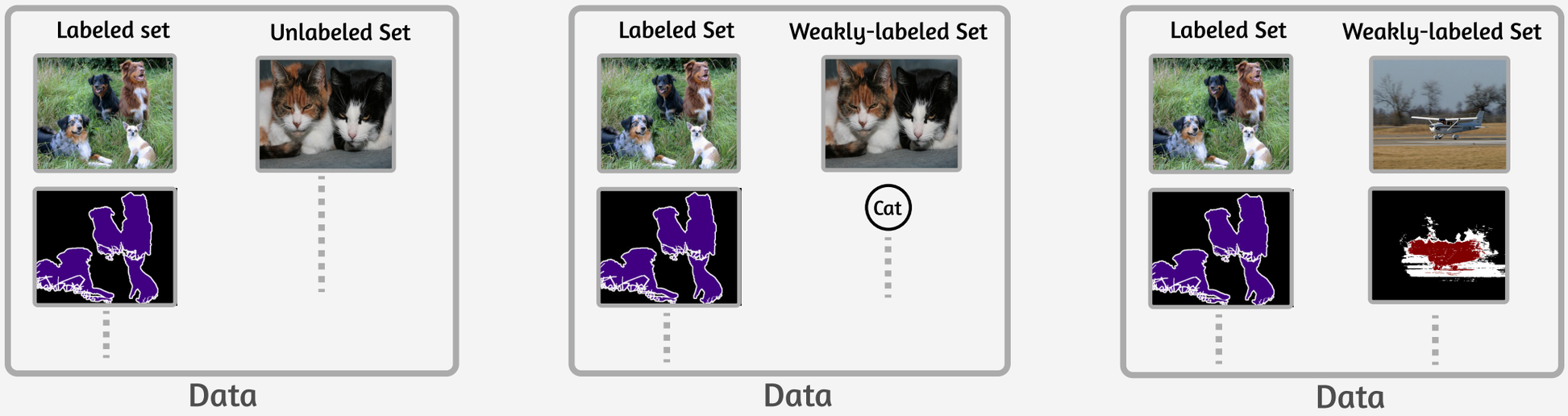




What if instead of unlabeled samples .... we had access to some form of weak labels, such as image level labels (classes).

Cross-Consistency Training

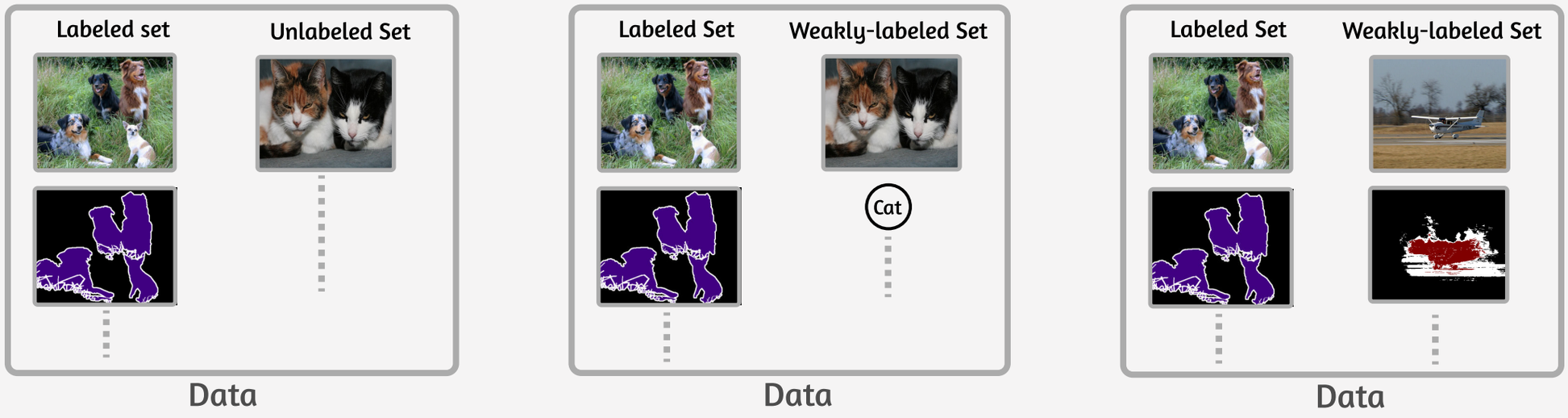
CCT with image-level labels

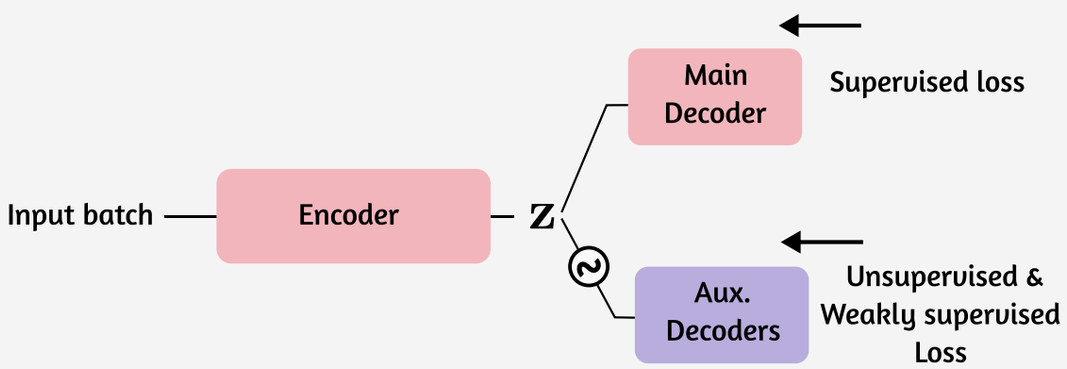


The weak labels can be used to generate pseudo labels.

Cross-Consistency Training

CCT with image-level labels

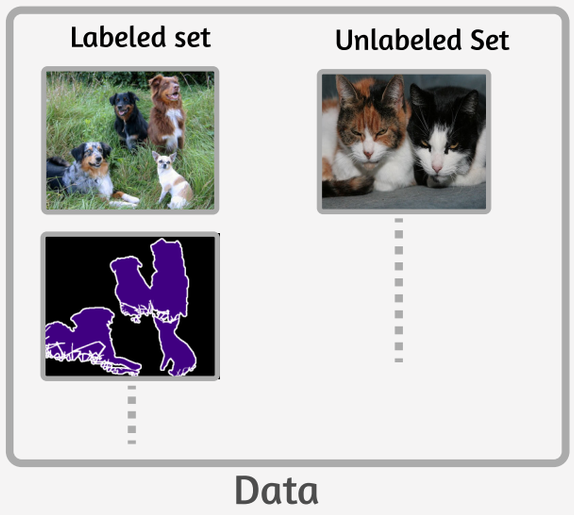


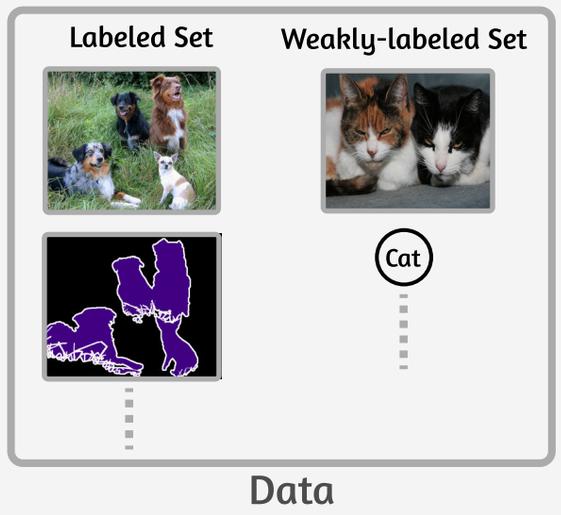


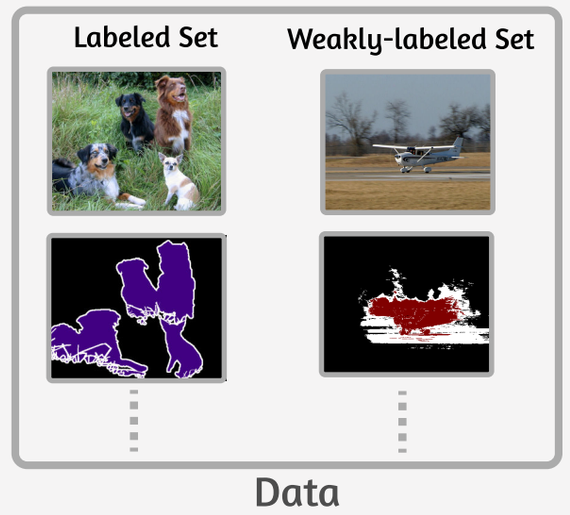
The weak labels can be used to generate pseudo labels, which can then be used to train the aux. decoder.

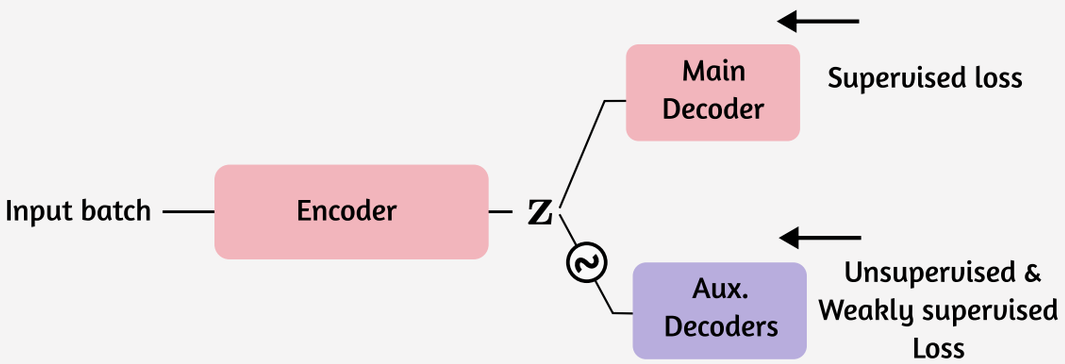
Cross-Consistency Training

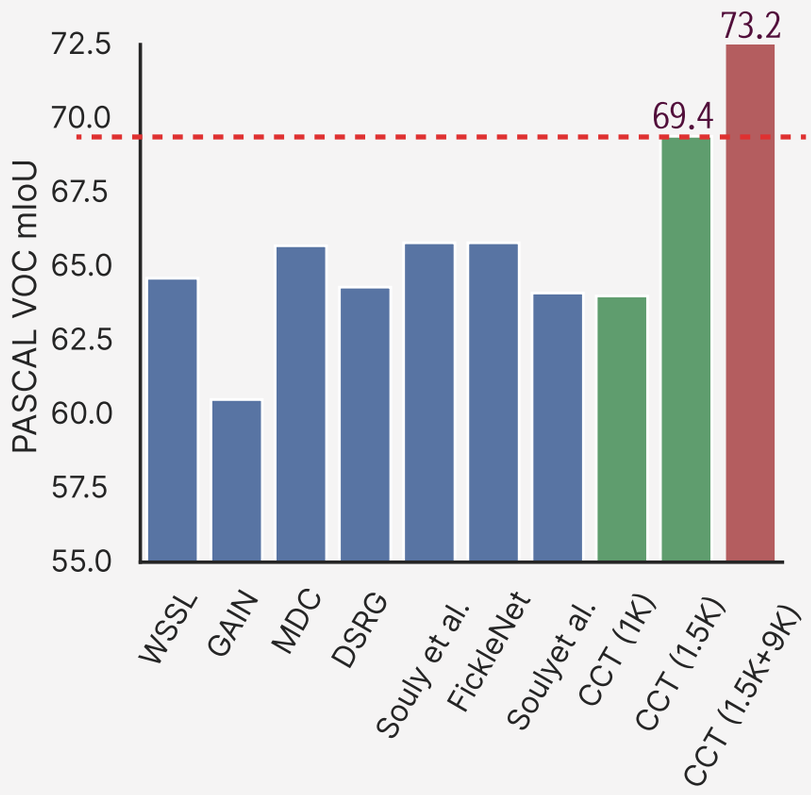
CCT with image-level labels









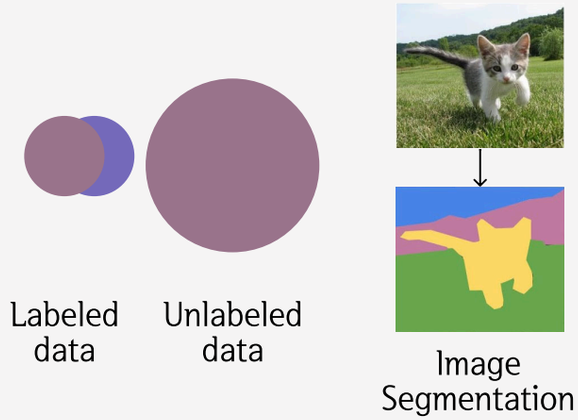


Contribution 1: Cross-Consistency Training (CCT)

Conclusion

We considered:

- Semi-supervised Learning
- Image Segmentation



We introduced CCT, a novel consistency-based approach for semi-supervised image segmentation.

Contribution 1: Cross-Consistency Training (CCT)

Conclusion

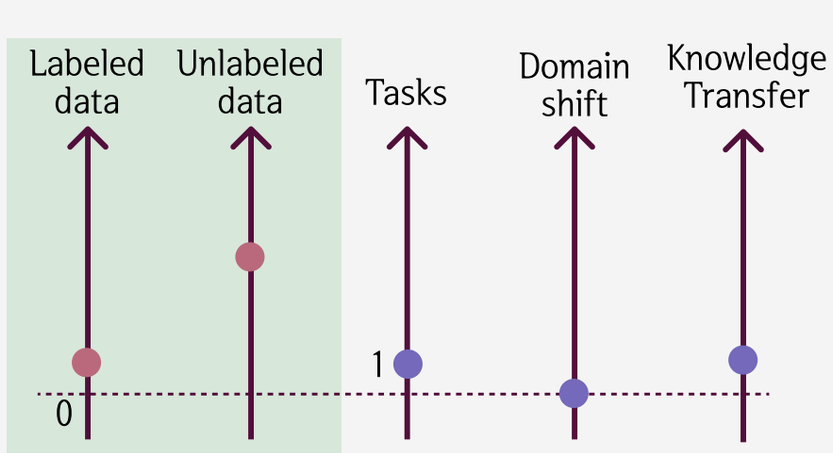
We considered:

- Semi-supervised Learning
- Image Segmentation

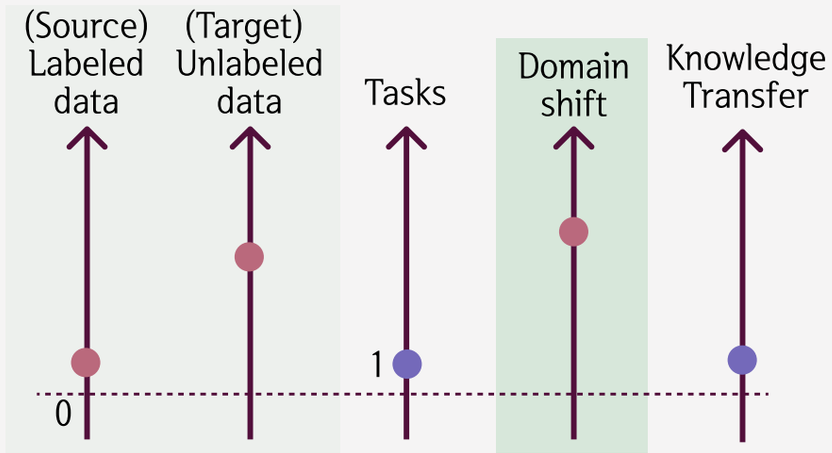
We introduced CCT, a novel consistency-based approach for semi-supervised image segmentation.

Additional details:

- Training on multiple domains.
- Training under a distribution shift (UDA).



Semi-supervised Learning



Unsupervised Domain Adaptation

Contribution 1: Cross-Consistency Training (CCT)

Conclusion

Limitations:

- Computation: multiple aux. decoder forward passes.
- Applicability: we only consider image segmentation.
- Perturbation: what are the optimal perturbations & at what level, is there a big dependence on them?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contribution 1: Cross-Consistency Training (CCT)

Conclusion

Limitations:

- Computation: multiple aux. decoder forward passes.
- Applicability: we only consider image segmentation.
- Perturbation: what are the optimal perturbations & at what level, is there a big dependence on them?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contribution 1: Cross-Consistency Training (CCT)

Conclusion

Limitations:

- Computation: multiple aux. decoder forward passes.
- Applicability: we only consider image segmentation.
- Perturbation: what are the optimal perturbations & at what level, is there a big dependence on them?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contribution 1: Cross-Consistency Training (CCT)

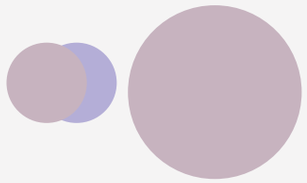
Conclusion

Limitations:

- Computation: multiple aux. decoder forward passes.
- Applicability: we only consider image segmentation.
- Perturbation: what are the optimal perturbations & at what level, is there a big dependence on them?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contributions

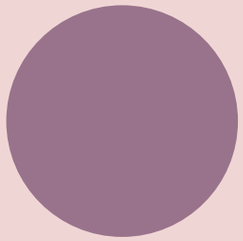
Semi-supervised Leaning



Labeled data Unlabeled data

Image Segmentation

Unsupervised Learning



Unlabeled data

Image Segmentation

Few-show Learning

Per-task Labeled data

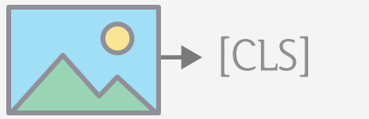
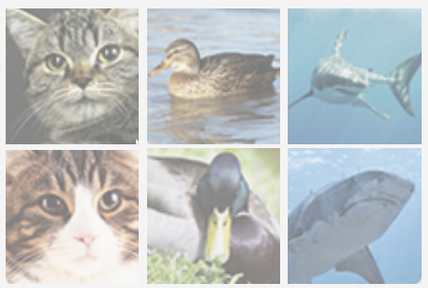


Image Classification

Tasks:

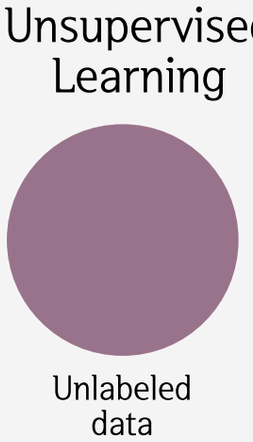
1 2 3



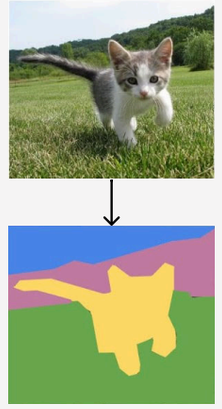
Contribution 2

Autoregressive Image Segmentation

Paradigm:
Unsupervised Learning



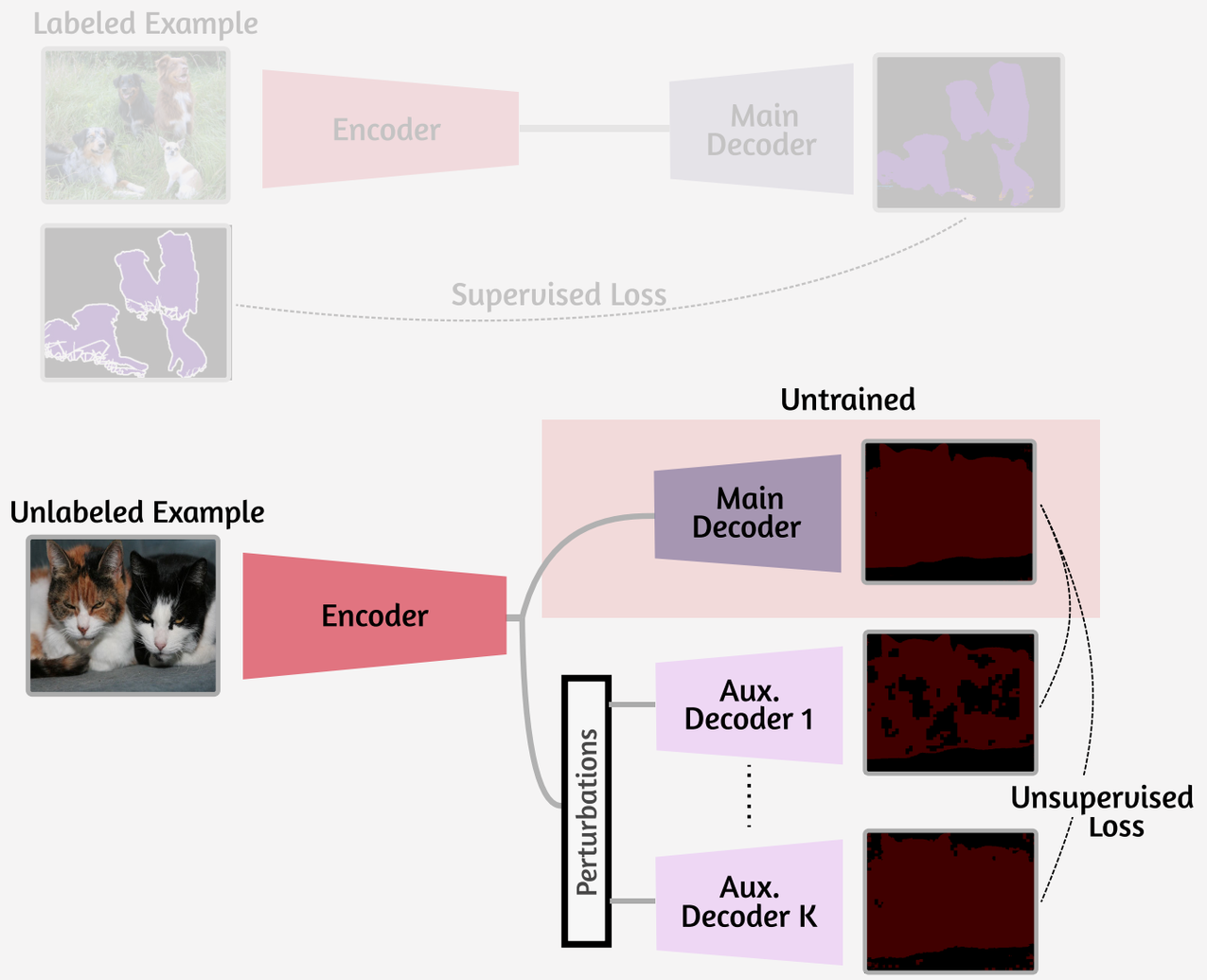
Task:
Image Segmentation



Autoregressive Unsupervised Image Segmentation. ECCV 2020.
Project: https://yassouali.github.io/autoreg_seg/

Autoregressive Image Segmentation

CCT as a starting point

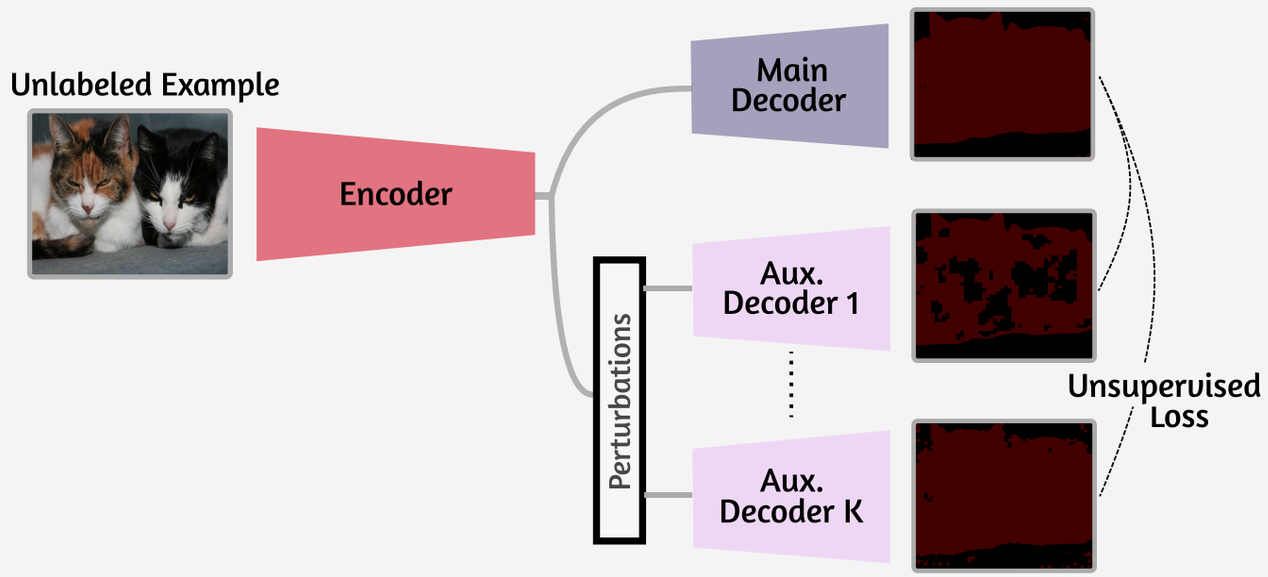


- No supervised loss, so the main decoder is not trained.
- Training only using the unsupervised loss.

Autoregressive Image Segmentation

CCT as a starting point

Unlabeled Set

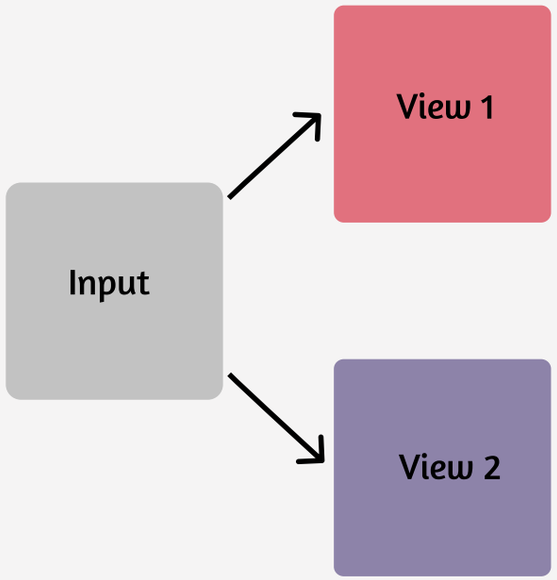


$$\text{Unsupervised Loss: } \mathcal{L}_u = \frac{1}{|\mathcal{D}_u|}\frac{1}{K}\sum_{\mathbf{x}_i^u \in \mathcal{D}_u}\sum_{k=1}^{K}\mathbf{d}\left(g\left(\mathbf{z}_i\right), g_a^k\left(\mathbf{z}_i\right)\right)$$

- A non-degenerate loss.
- A new method to generate the augmentation/ views.

Autoregressive Image Segmentation

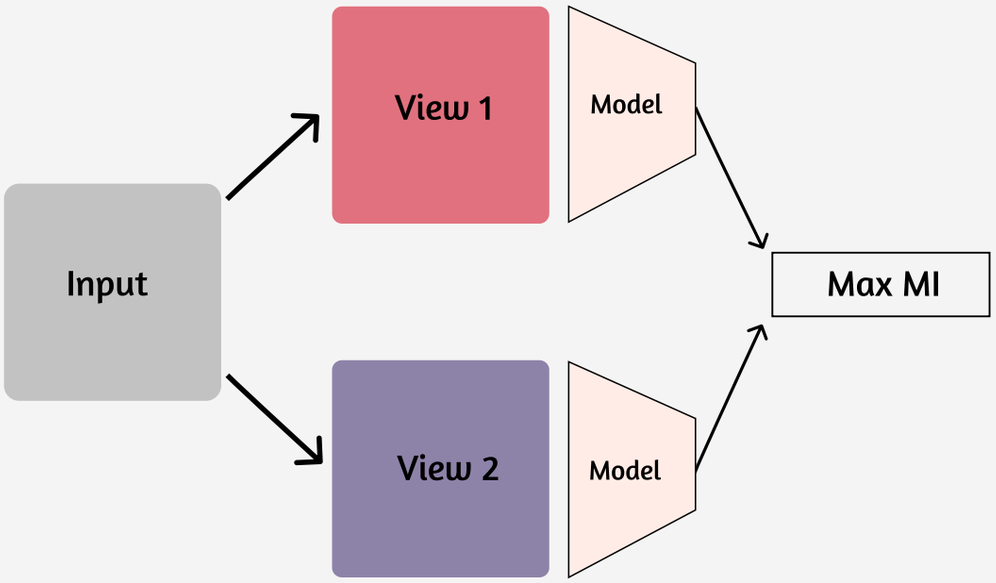
Loss function: Mutual Information Maximisation



First, we generate two views of a given input image.

Autoregressive Image Segmentation

Loss function: Mutual Information Maximisation

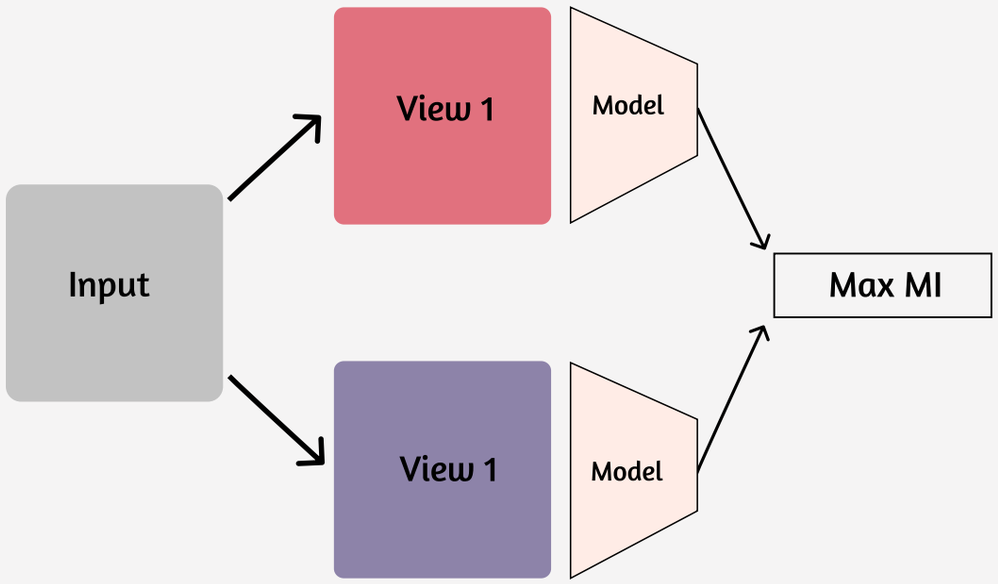


We obtain two set of predictions using our model, then we maximise the MI between them.

On Mutual Information Maximization for Representation Learning. Tschannen et al, 2019.
Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.

Autoregressive Image Segmentation

Loss function: Mutual Information Maximisation



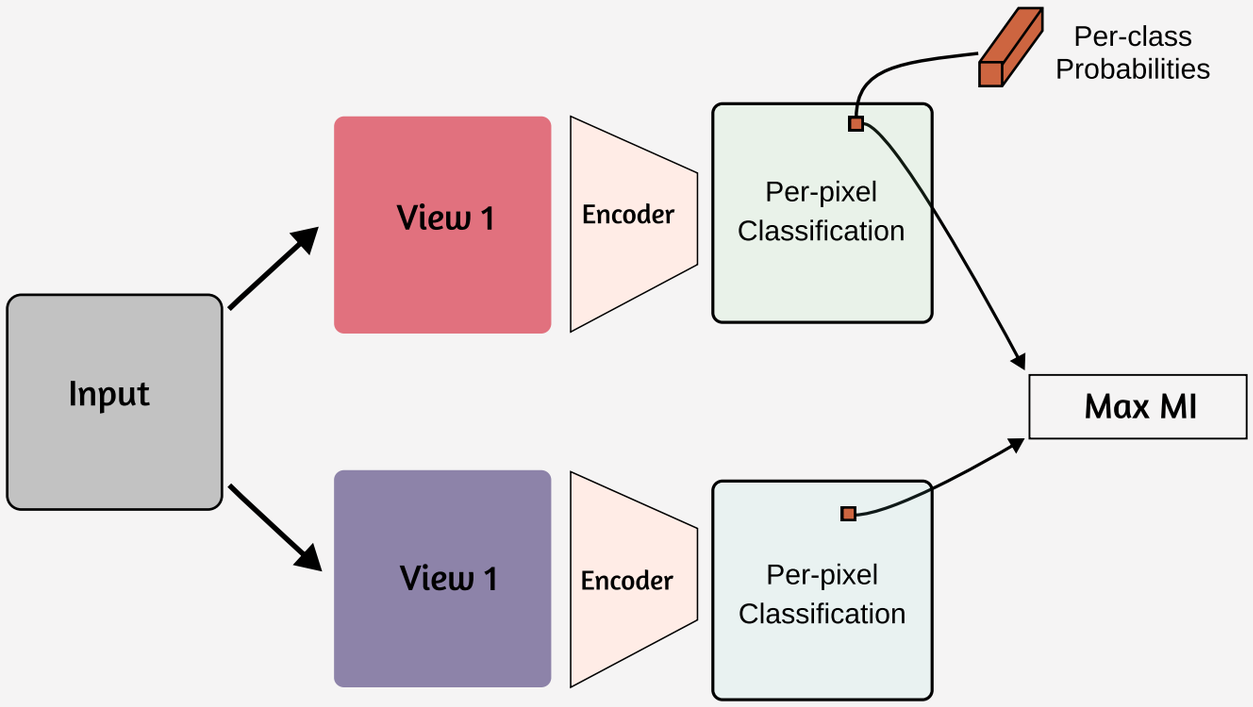
$$MI = \underbrace{Entropy}_{\text{Pushes for degenerate solutions}} - \underbrace{Cond.\ Entropy}_{\text{Pushes for trivial solutions}}$$

As a result, we avoid degenerate solutions due to the robustness of MI.

On Mutual Information Maximization for Representation Learning. Tschannen et al, 2019.
Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.

Autoregressive Image Segmentation

Loss function: Mutual Information Maximisation



$$\max I(\mathbf{y}; \mathbf{y}')$$

Maximizing the extact value
for clustering

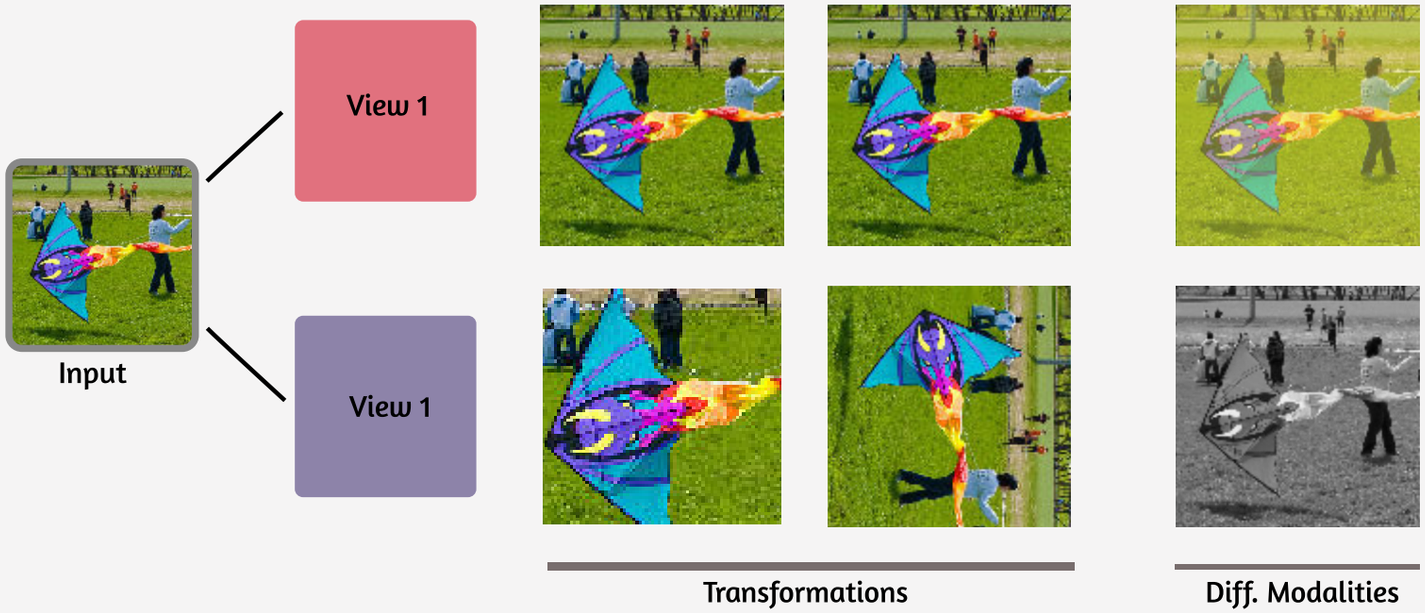
$$\mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\mathbb{E}_{p(\mathbf{y},\mathbf{y}')}\log\frac{p(\mathbf{y},\mathbf{y}')}{p(\mathbf{y})p(\mathbf{y}')}\right]$$

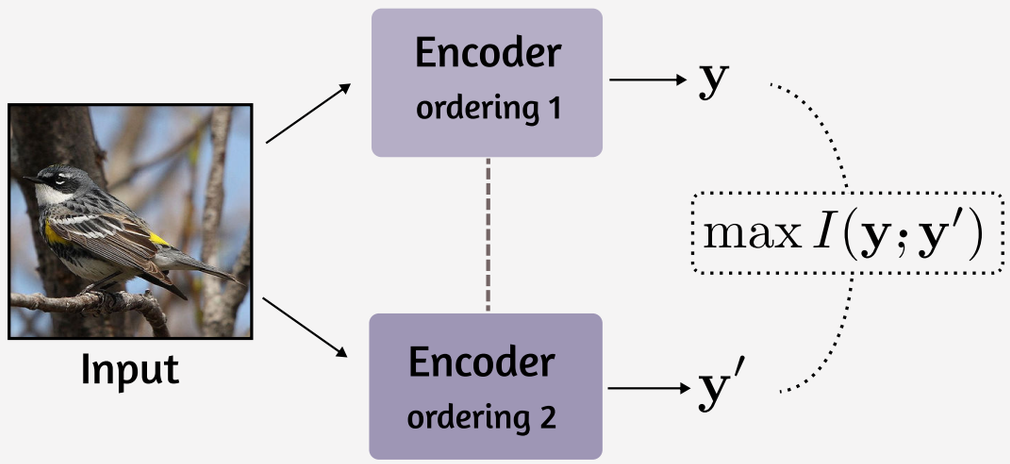
For image segmentation, this is applied in a per-pixel manner.

On Mutual Information Maximization for Representation Learning. Tschannen et al, 2019.
Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.
Representation Learning with Contrastive Predictive Coding. Oord et al, 2019.

Autoregressive Image Segmentation

Using the model as a view generator

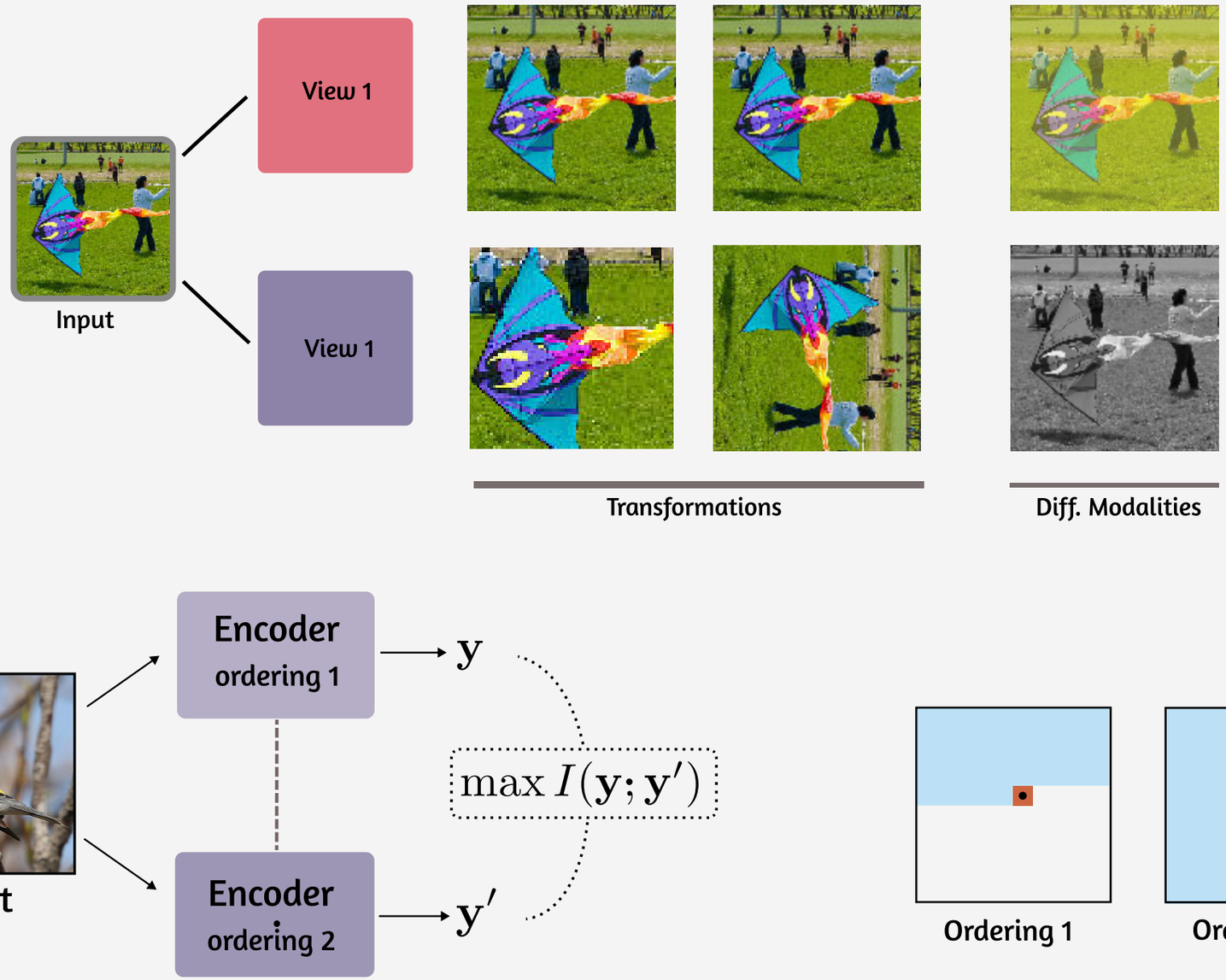




Instead of using standard image augmentations, we propose a view generation method rooted at the pixel level.

Autoregressive Image Segmentation

Using the model as a view generator

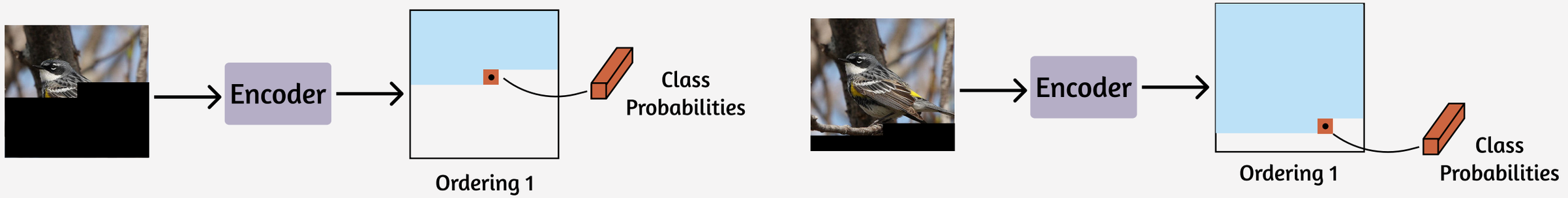


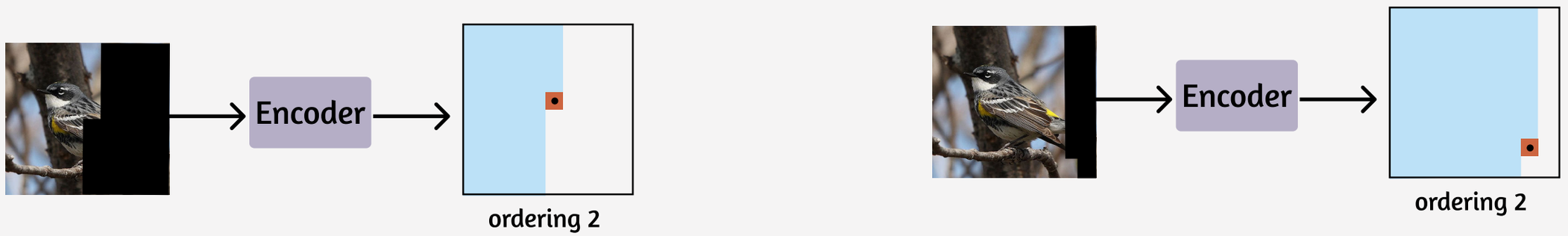
The model itself will be leveraged to generate different forms of output-input dependencies.

Pixel Recurrent Neural Networks. 2016. Van den Oord et al.
Conditional Image Generation with Pixel CNN Decoders. 2016. Van den Oord et al.

Autoregressive Image Segmentation

Views generation using masked convolutions





Each pixel will have its corresponding dependencies ... but how to generate them efficiently?

Autoregressive Image Segmentation

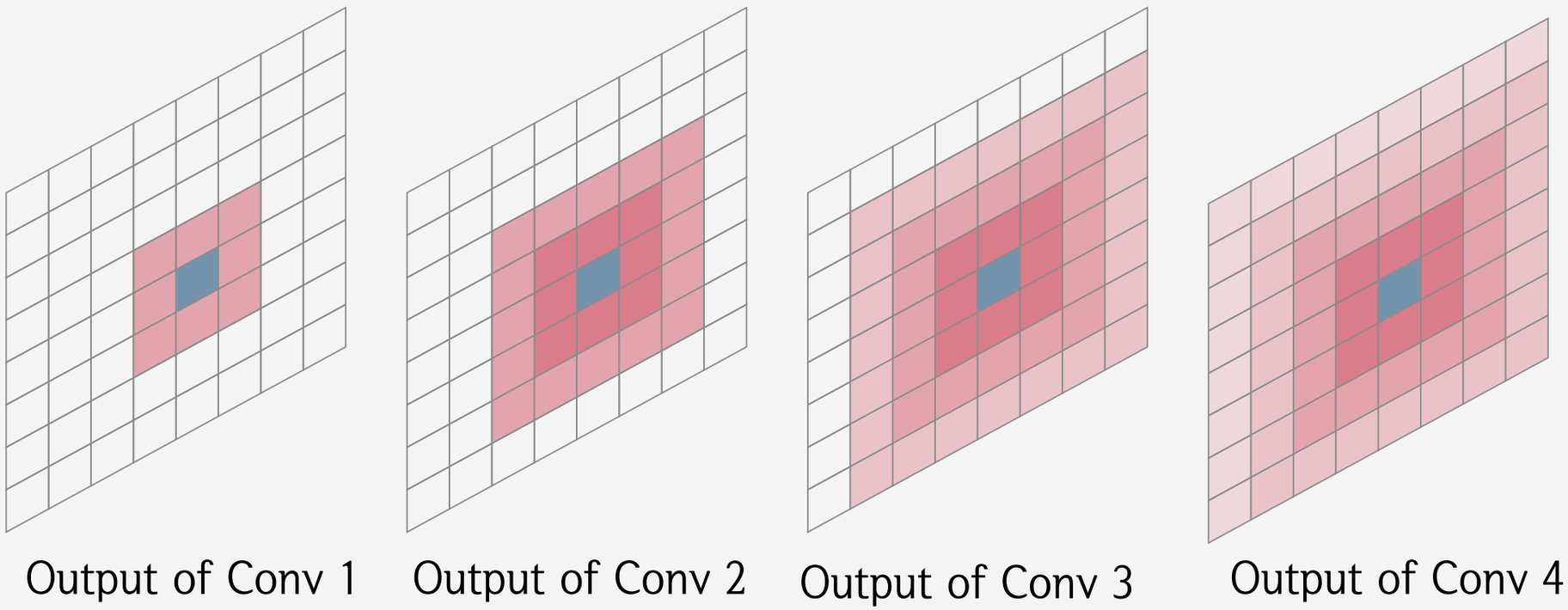
Views generation using masked convolutions

To efficiently obtain the desired dependencies, we use masked convolutions.

Autoregressive Image Segmentation

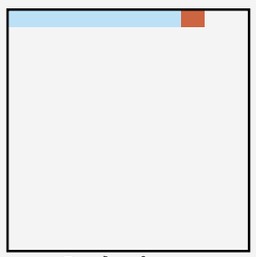
Views generation using masked convolutions

Receptive field of a standard convolution

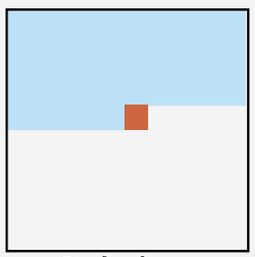


Autoregressive Image Segmentation

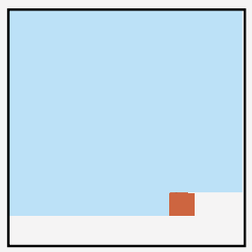
Views generation using masked convolutions



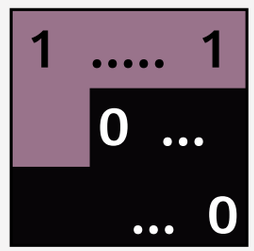
Ordering 1



Ordering 1

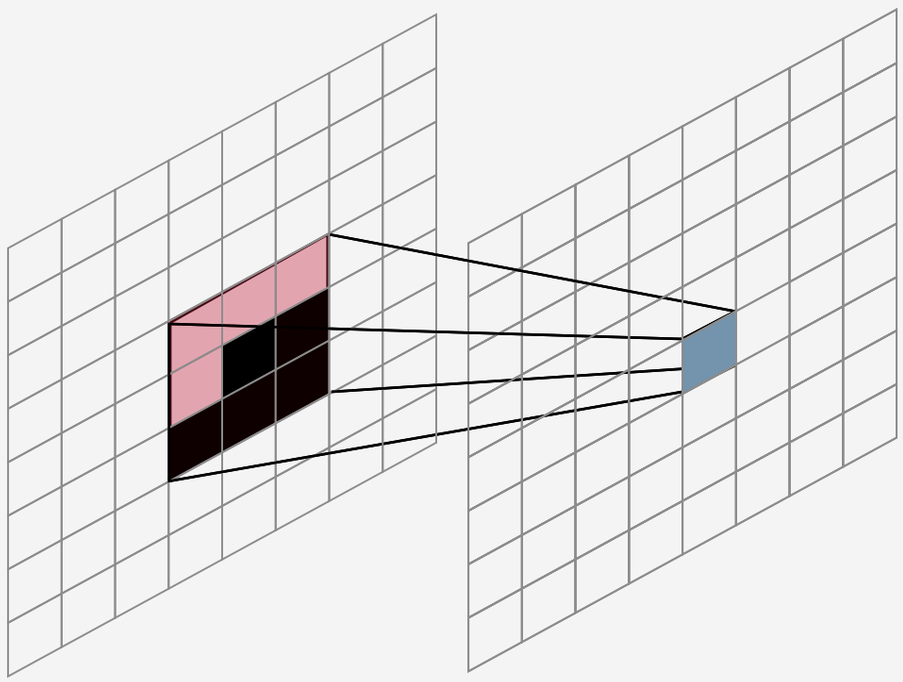


Ordering 1



Masked Convolutions

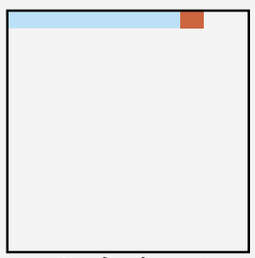
Receptive field of a masked convolution



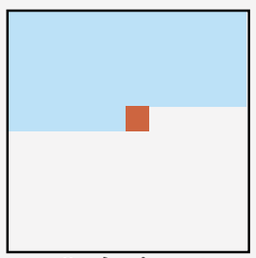
First application of a masked conv

Autoregressive Image Segmentation

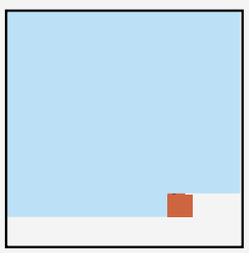
Views generation using masked convolutions



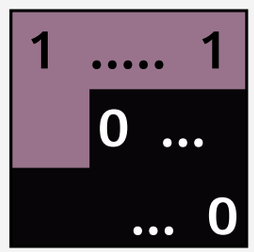
Ordering 1



Ordering 1

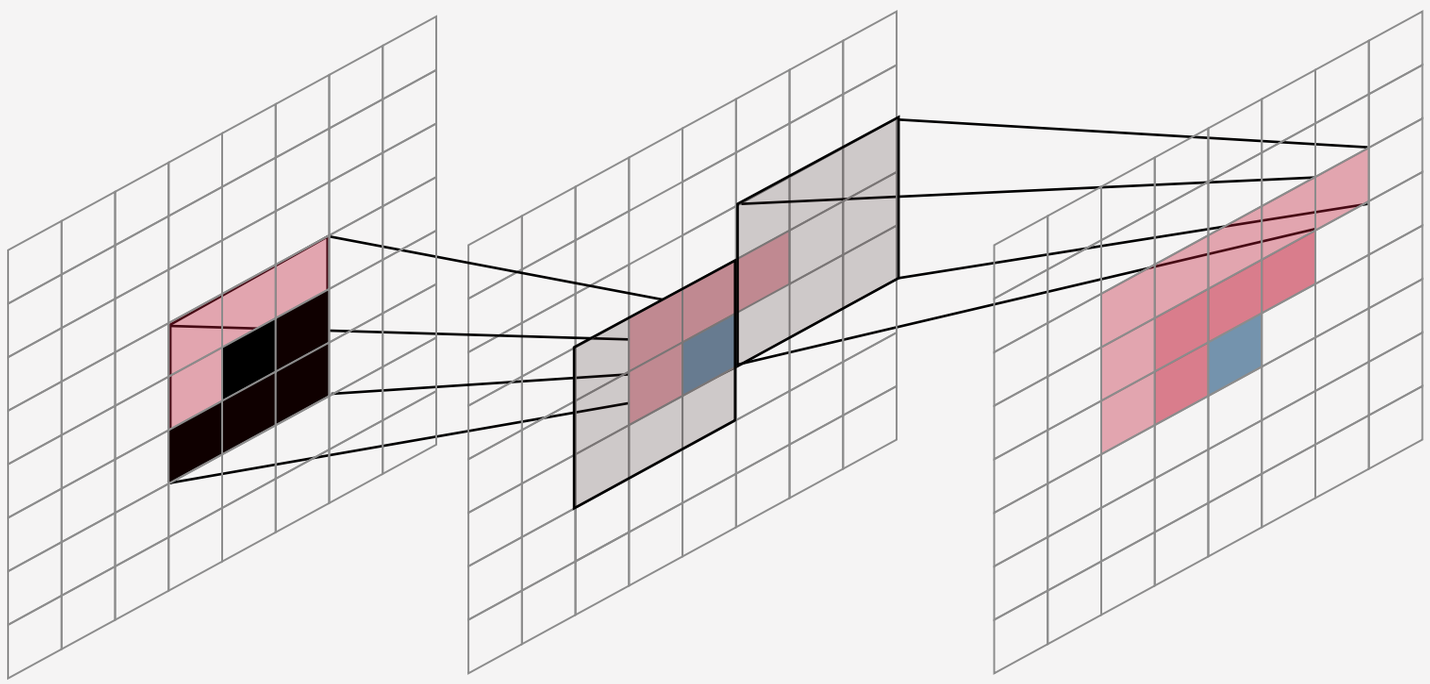


Ordering 1



Masked Convolutions

Receptive field of a masked convolution

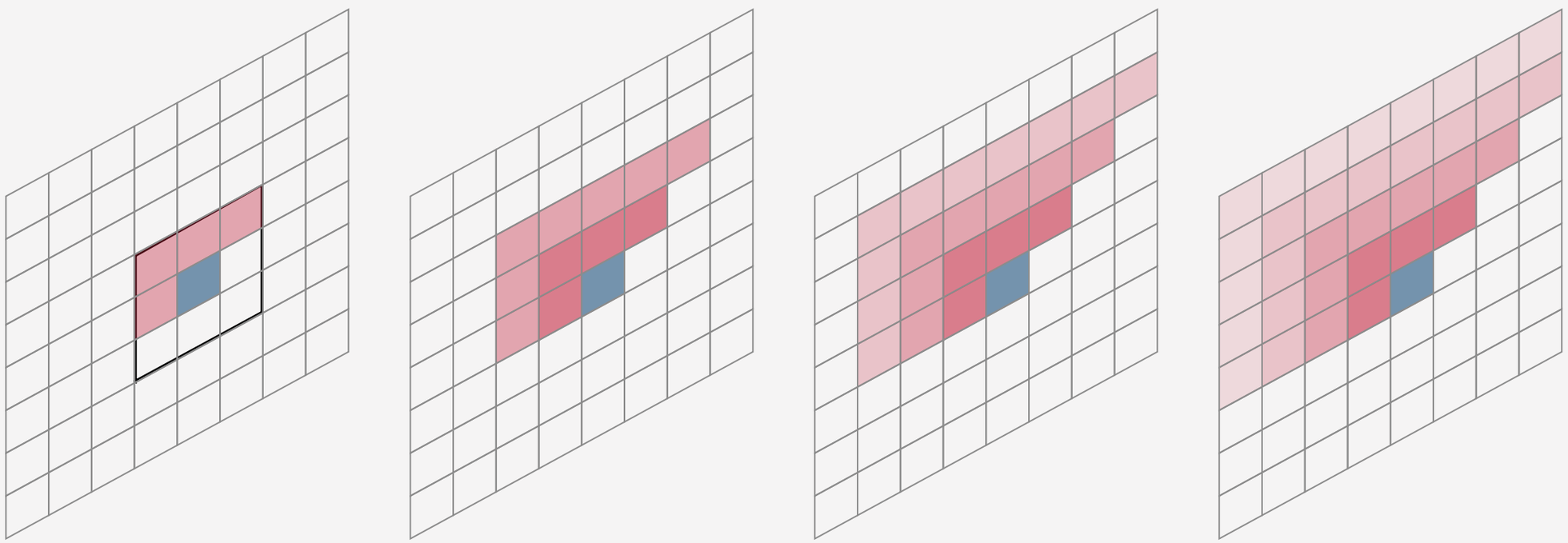


Second application of a masked conv

Autoregressive Image Segmentation

Views generation using masked convolutions

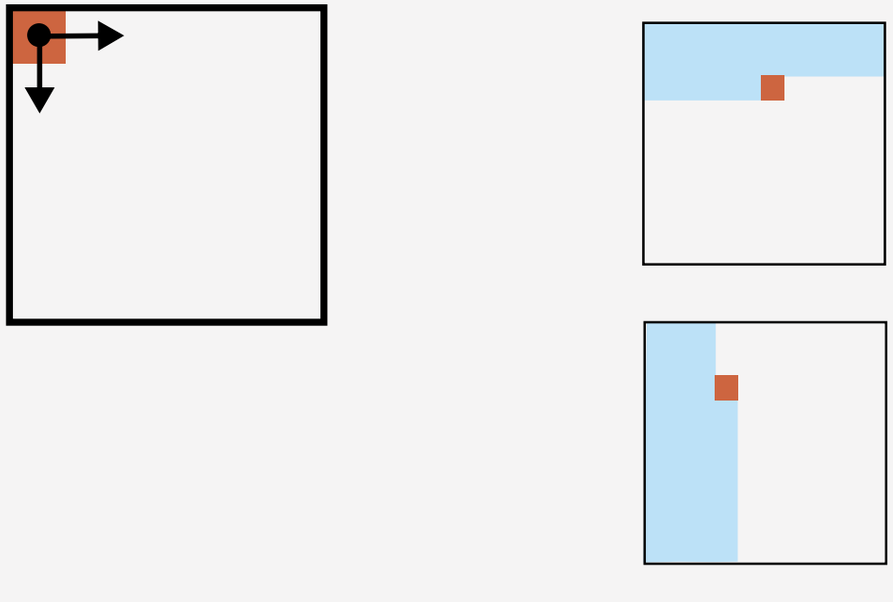
Receptive field of a masked convolution



Output of Conv 1 Output of Conv 2 Output of Conv 3 Output of Conv 4

Autoregressive Image Segmentation

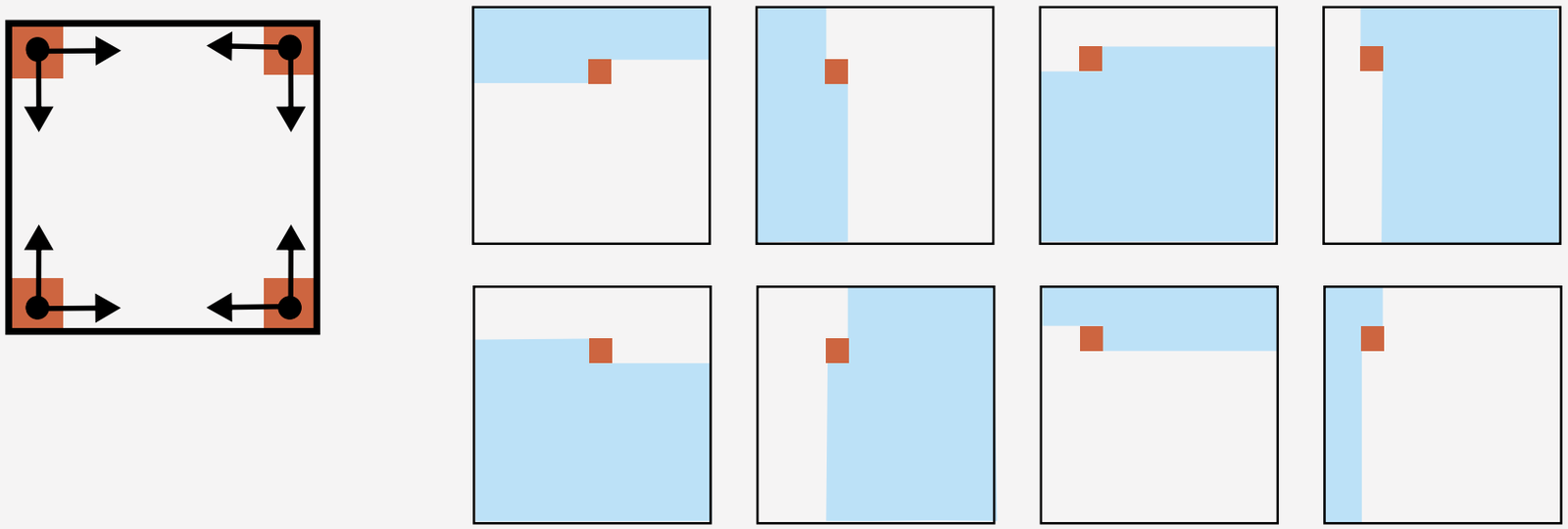
The set of possible views



Instead of just two orderings to generate two two views and compute the loss...

Autoregressive Image Segmentation

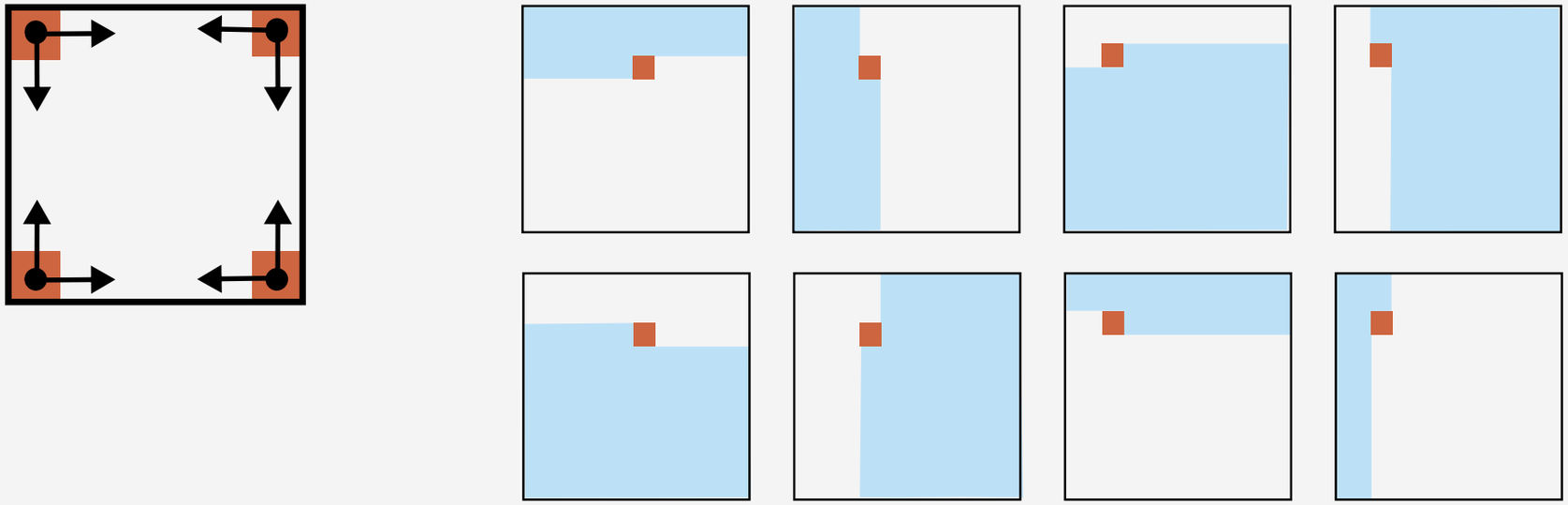
The set of possible views



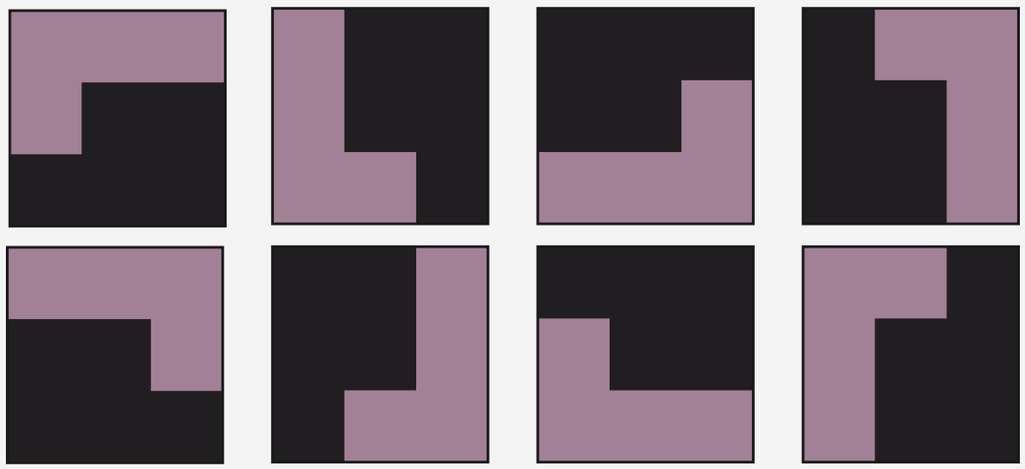
We use propose to use possible 8 orderings = 4 corners x 2 directions.

Autoregressive Image Segmentation

The set of possible views



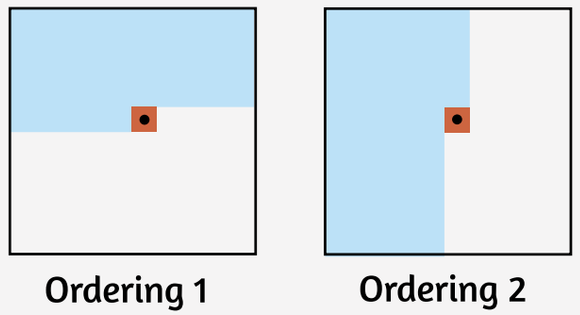
Convolution masks



And each ordering is generated using its corresponding masks.

Autoregressive Image Segmentation

Summary: Autoregressive Image Segmentation

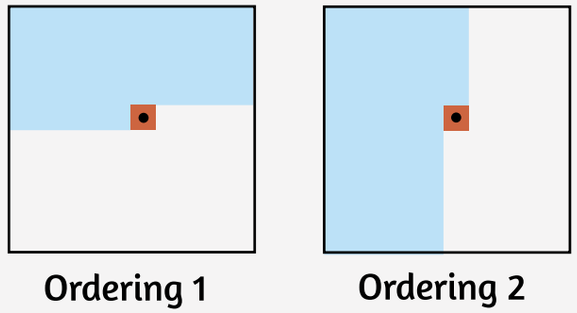


Training:

1- Sample two valid orderings

Autoregressive Image Segmentation

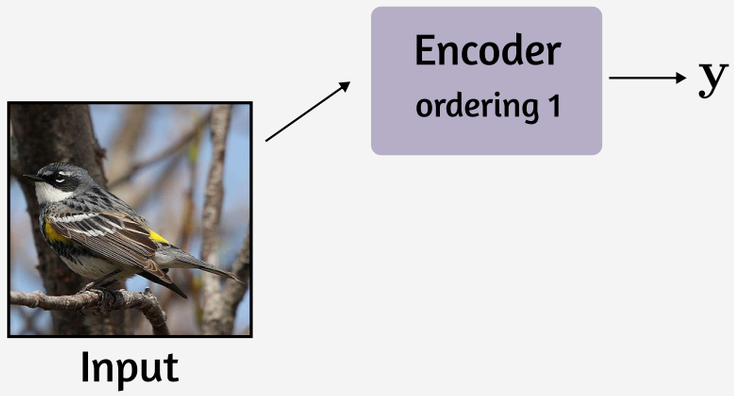
Summary: Autoregressive Image Segmentation



Training:

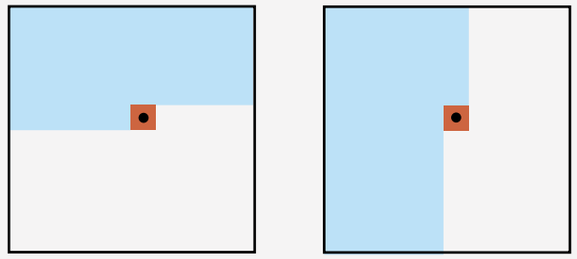
1- Sample two valid orderings

2- Compute the output corresponding to the first view



Autoregressive Image Segmentation

Autoregressive Image Segmentation

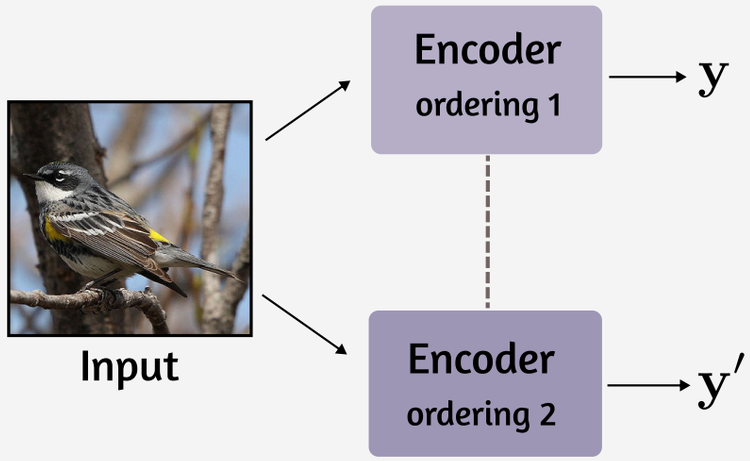


Training:

1- Sample two valid orderings

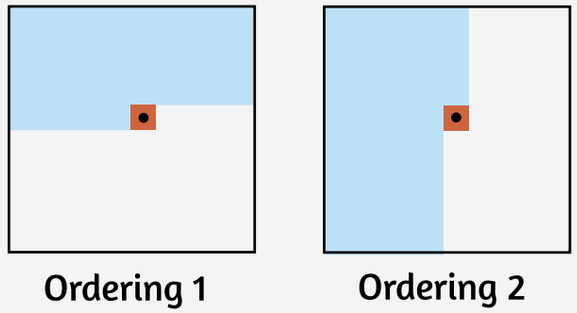
2- Compute the output corresponding to the first view

3. Compute the output corresponding to the second view



Autoregressive Image Segmentation

Autoregressive Image Segmentation



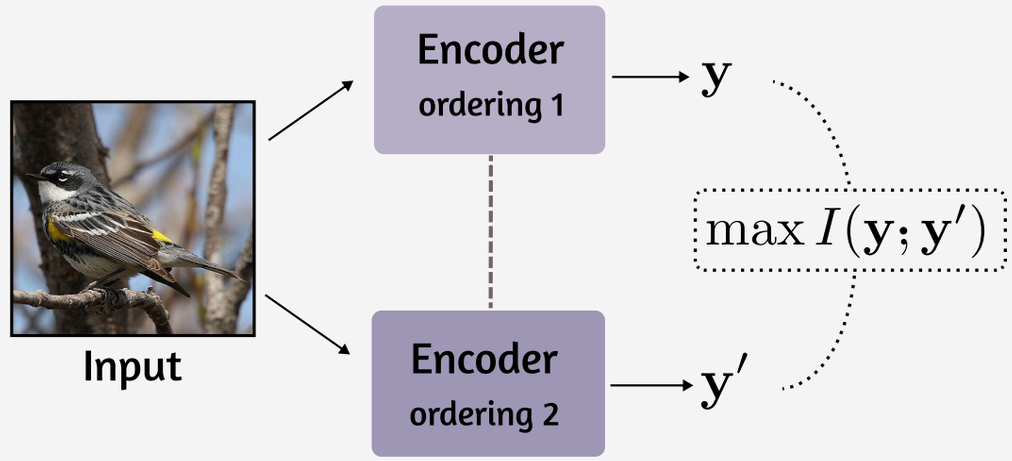
Training:

1- Sample two valid orderings

2- Compute the output corresponding to the first view

3. Compute the output corresponding to the second view

4- Compute the loss



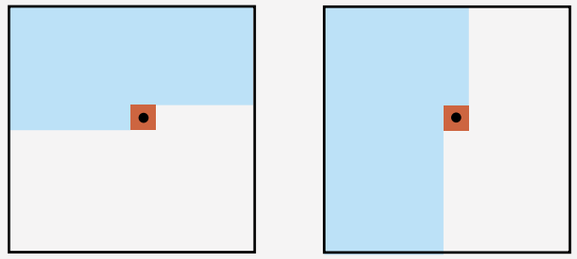
$$\max I(\mathbf{y};\mathbf{y}') \longleftrightarrow$$

Maximizing the extact value for clustering

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\mathbb{E}_{p(\mathbf{y},\mathbf{y}')}\log\frac{p(\mathbf{y},\mathbf{y}')}{p(\mathbf{y})p(\mathbf{y}')}\right]$$

Autoregressive Image Segmentation

Autoregressive Image Segmentation



Training:

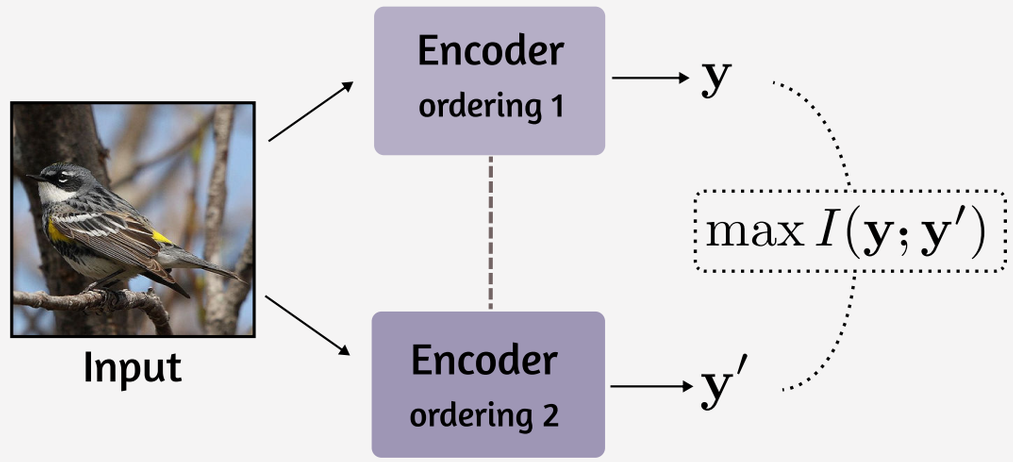
1- Sample two valid orderings

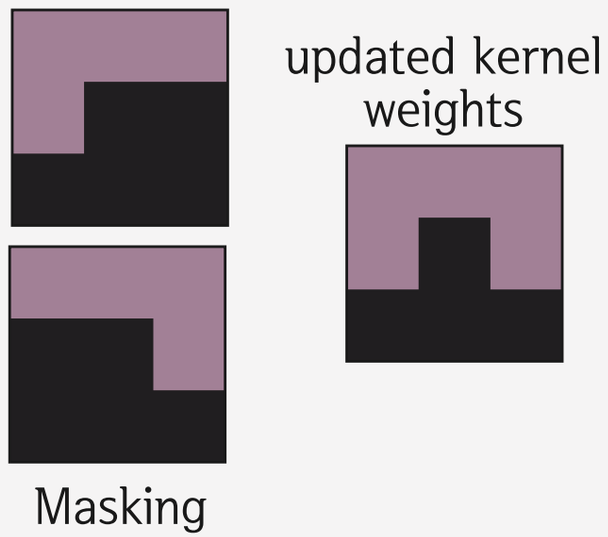
2- Compute the output corresponding to the first view

3. Compute the output corresponding to the second view

4- Compute the loss

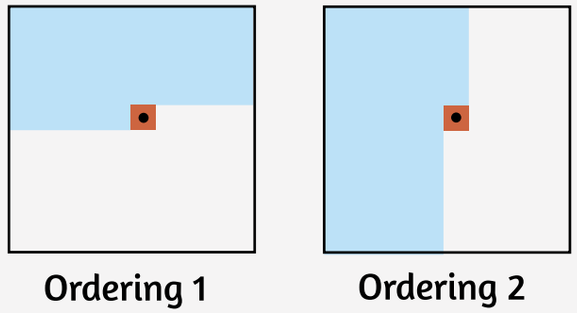
5- Backpropagate and only update the unmasked weights, masked weights remain unchanged





Autoregressive Image Segmentation

Autoregressive Image Segmentation



Training:

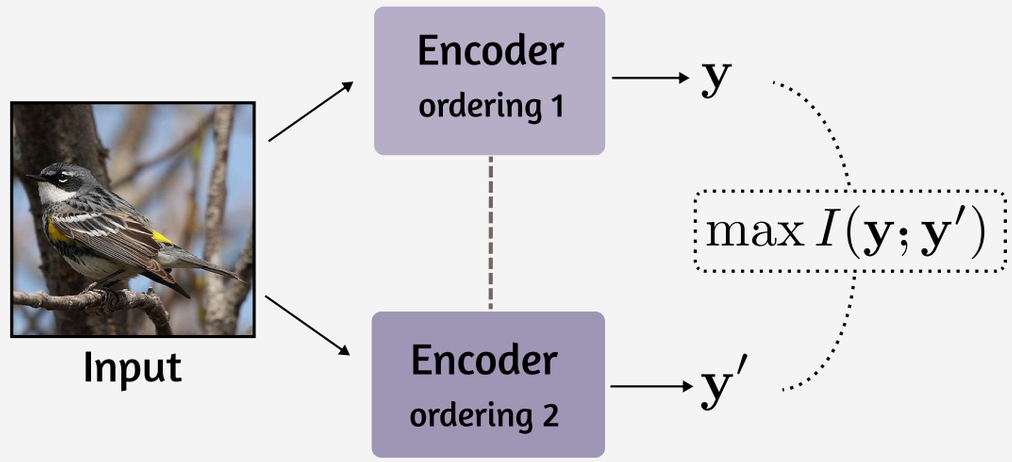
1- Sample two valid orderings

2- Compute the output corresponding to the first view

3. Compute the output corresponding to the second view

4- Compute the loss

5- Backpropagate and only update the unmasked weights, masked weights remain unchanged



Inference:

Remove making & proceed as in standard image segmentation models.

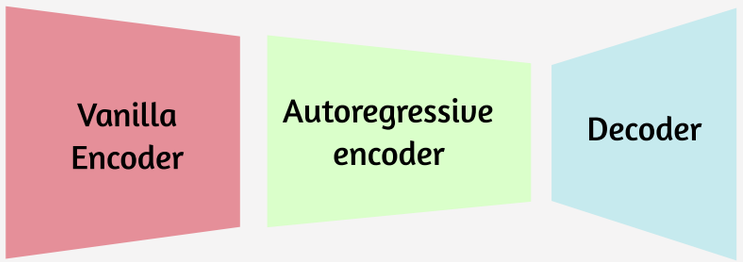
For evaluation:

- Find best one-to-one matching between predictions & ground truth.
- Compute the pixel wise accuracy.

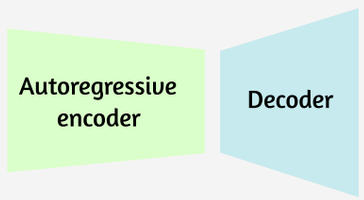
Autoregressive Image Segmentation

Results: Ablations of the model architecture

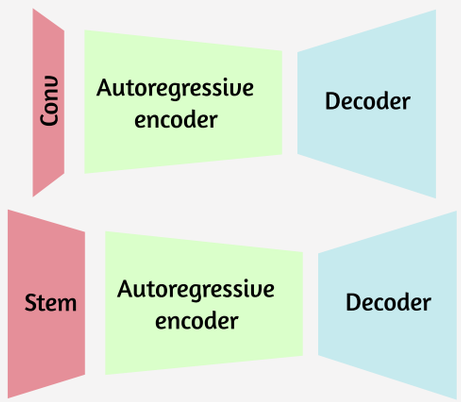
A general purpose model design



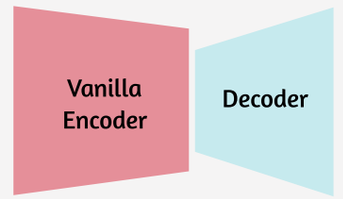
F1: apply masking over the inputs

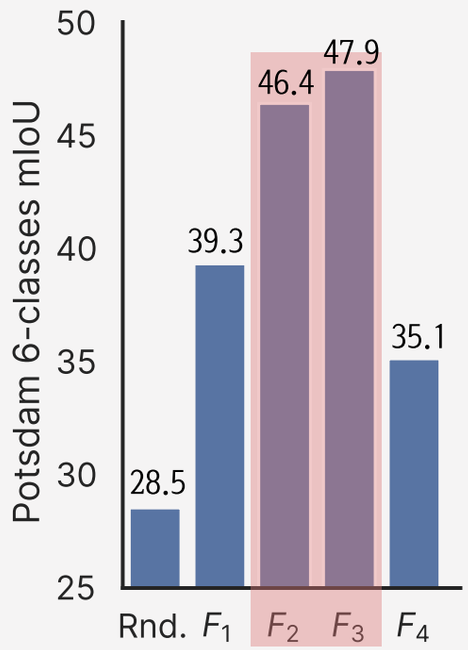


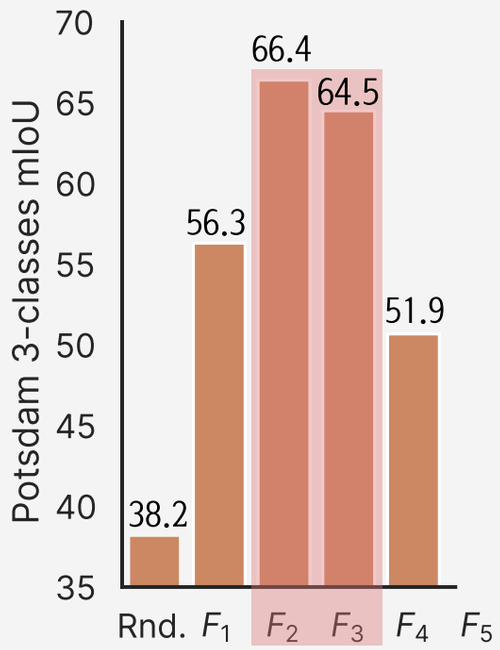
F2 & F3: apply over the features



F4: a standard encoder-decoder model



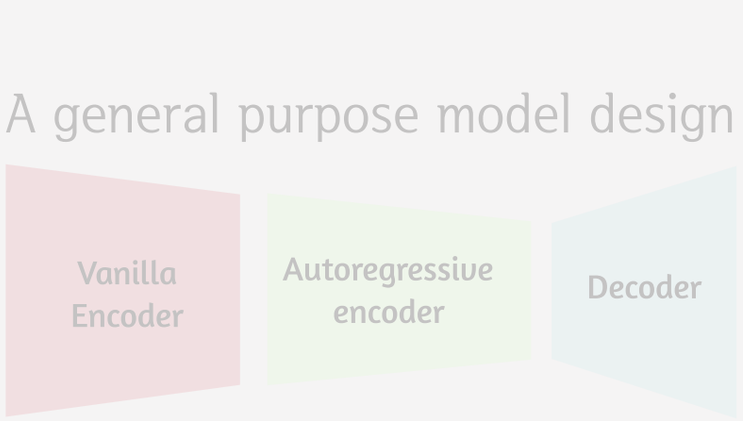


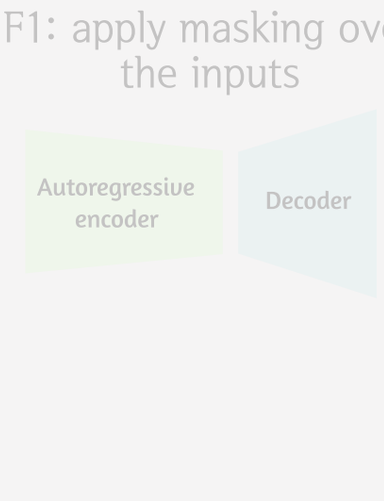


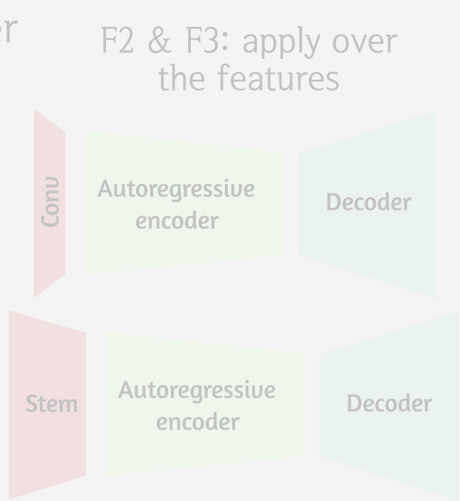
The best results are obtained when applying the orderings over the features instead of the inputs directly.

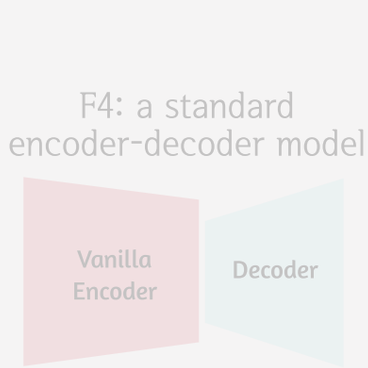
Autoregressive Image Segmentation

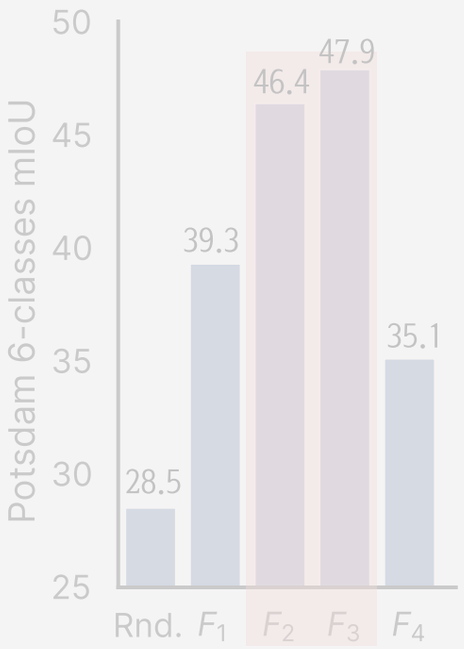
Results: Ablations of the effect of number of views

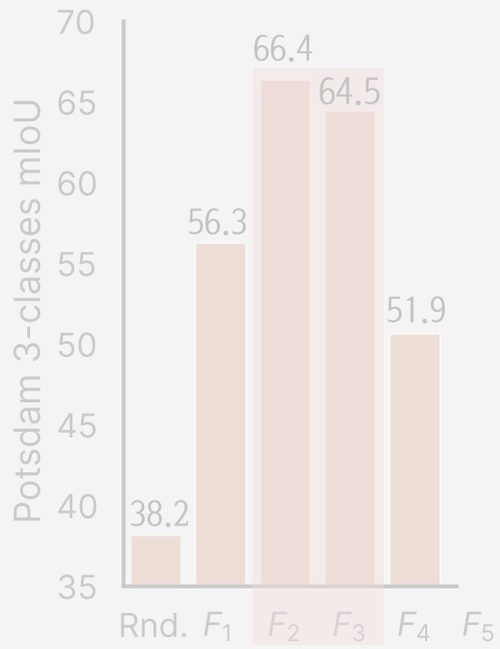


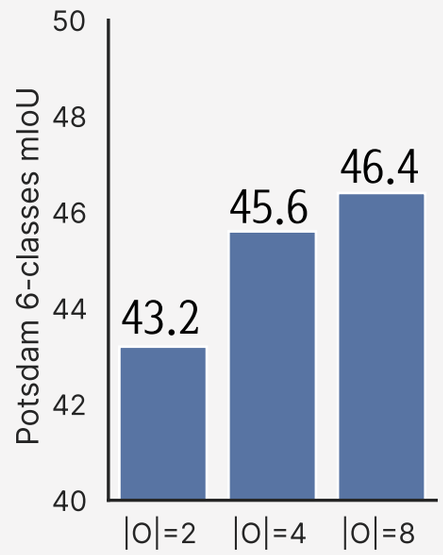


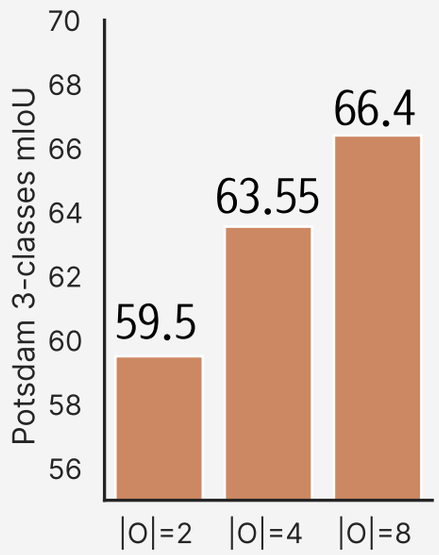








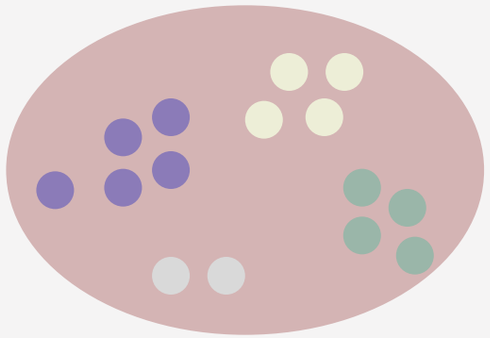




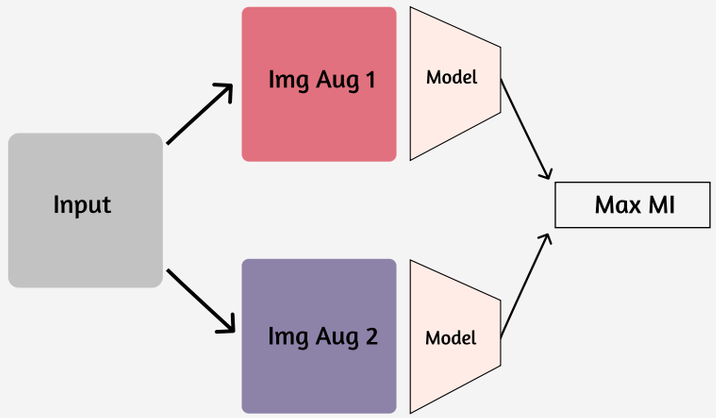
Increasing the number of views used results in better performances.

Autoregressive Image Segmentation

Results: Comparison with SOTA



Clustering Based methods

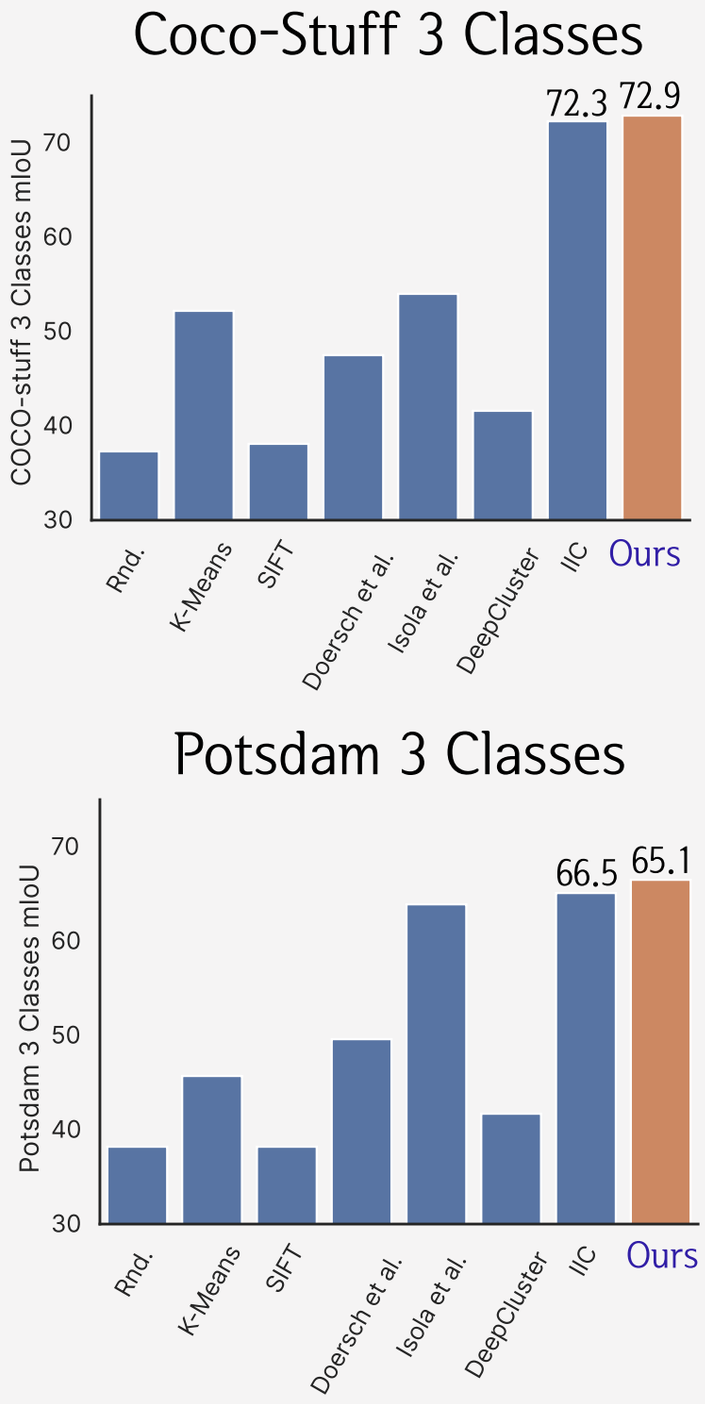


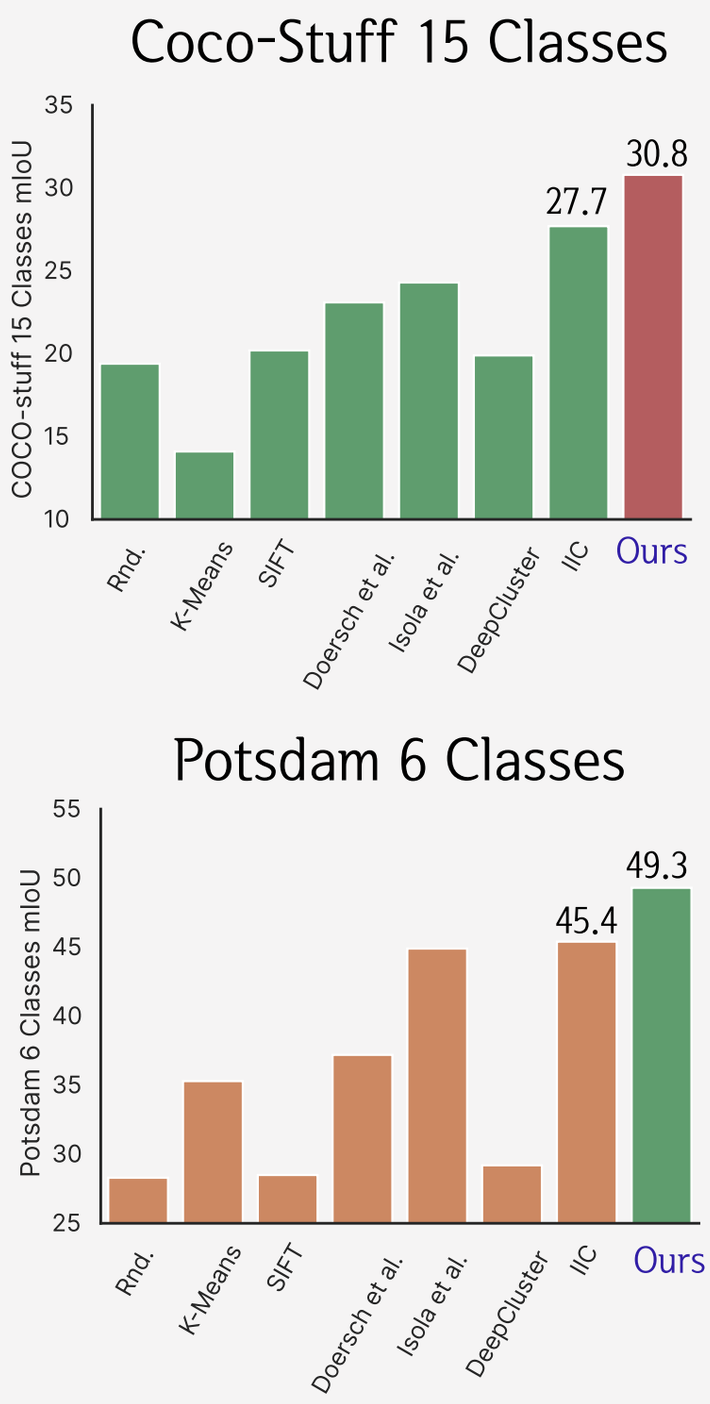
MI maximisation
with standard img. aug.

Learning visual groups from co-occurrences in space and time. Isola et al, 2016.
Deep clustering for unsupervised learning of visual features. Caron et al, 2018.
Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.

Autoregressive Image Segmentation

Results: Comparison with SOTA





AC outperforms previous methods, especially on harder tasks (6 & 15 classes)

Autoregressive Image Segmentation

Results: Obtained segmentation maps

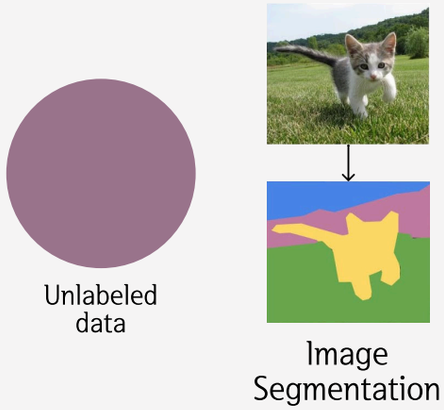
The segmentation maps are semantically meaningful and consistent across images.

Contribution2: Autoregressive Image Segmentation

Conclusion

We considered:

- Unsupervised Learning
- Image Segmentation



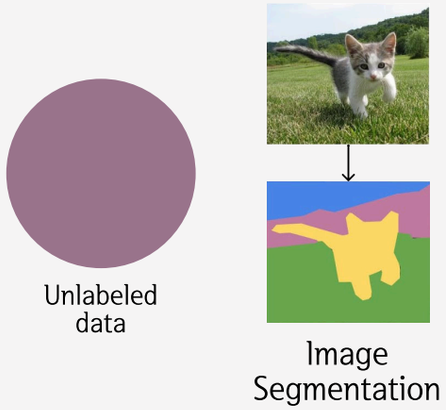
We introduced Autoregressive image segmentation, a novel view generation method for pixel wise tasks in an unsupervised setting.

Contribution2: Autoregressive Image Segmentation

Conclusion

We considered:

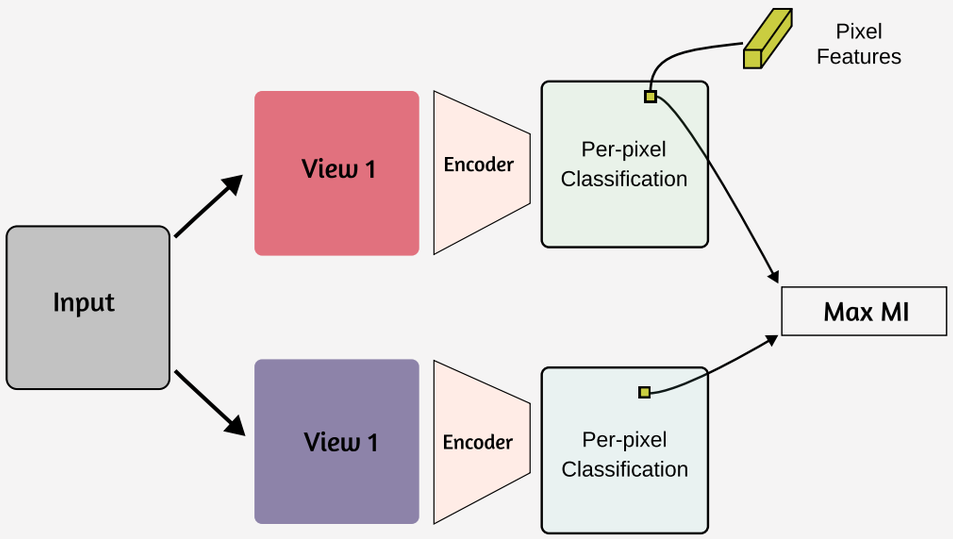
- Unsupervised Learning
- Image Segmentation

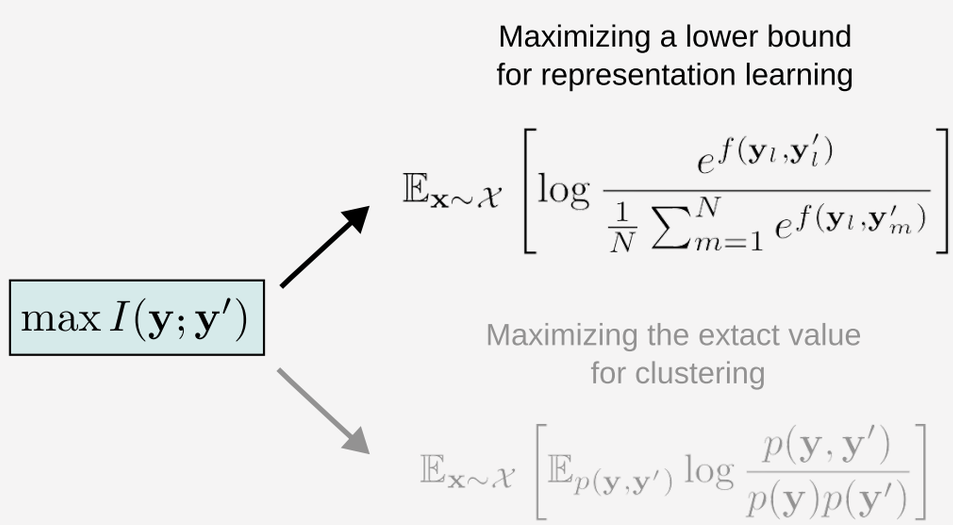


We introduced Autoregressive image segmentation, a novel view generation method for pixel wise tasks in an unsupervised setting.

Additional details:

- A new masking procedure, with optional attention mechanism.
- Representation Learning.





Contribution2: Autoregressive Image Segmentation

Conclusion

Limitations:

- Importance of structured orderings: can we use simpler methods like dropout or random orderings?
- Applicability: we limited ourselves to simple settings, and clean dataset, but does this work with corrupted data and challenging settings?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contribution2: Autoregressive Image Segmentation

Conclusion

Limitations:

- Importance of structured orderings: can we use simpler methods like dropout or random orderings?
- Applicability: we limited ourselves to simple settings, and clean dataset, but does this work with corrupted data and challenging settings?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contribution2: Autoregressive Image Segmentation

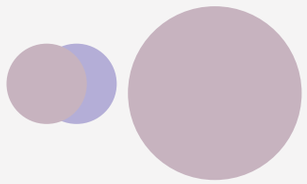
Conclusion

Limitations:

- Importance of structured orderings: can we use simpler methods like dropout or random orderings?
- Applicability: we limited ourselves to simple settings, and clean dataset, but does this work with corrupted data and challenging settings?
- Scalability: can we scale this method with a large amount of unlabeled data or does it saturate?

Contributions

Semi-supervised Leaning



Labeled data Unlabeled data

Image Segmentation

Unsupervised Learning

Unlabeled data

Image Segmentation

Few-show Learning

Per-task Labeled data

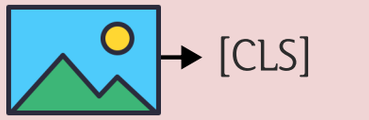


Image Classification

Tasks:

Contribution 3

Spatial Contrastive Learning

Paradigm:
Few-show Learning

Per-task
Labeled data

Task:
Image Classification

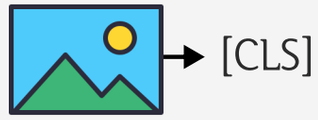


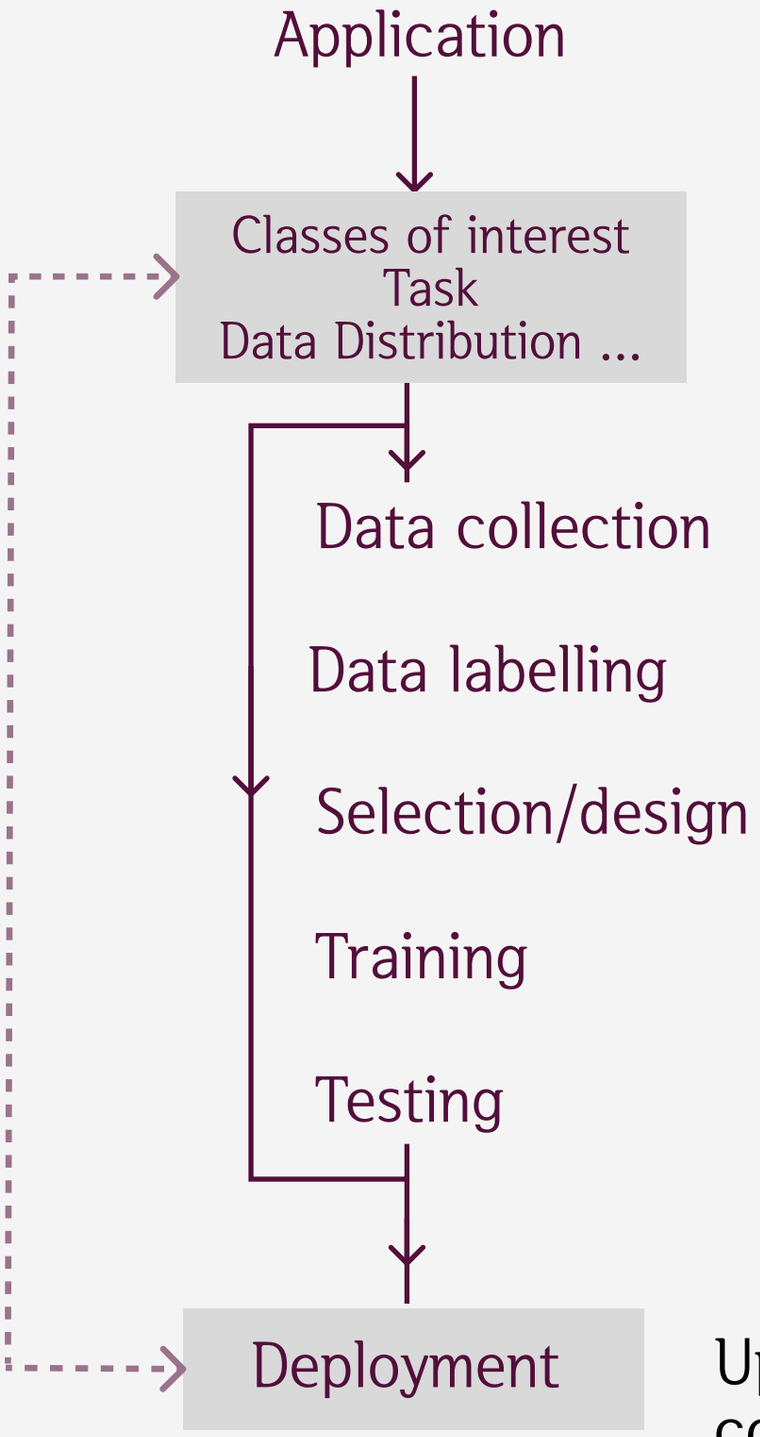
Image
Classification

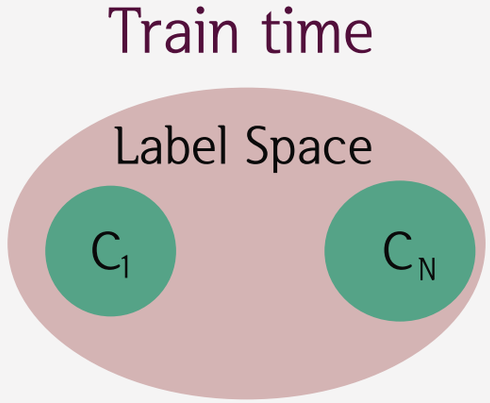
Tasks:

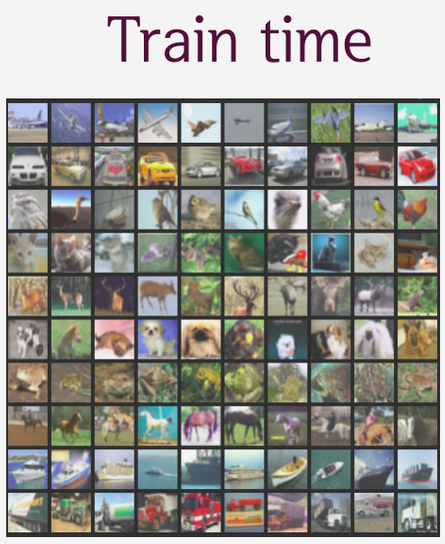
Spatial Contrastive Learning for Few-Shot Classification. ECML 2021.
Code: https://github.com/yassouali/SCL

Spatial Contrastive Learning

Introduction



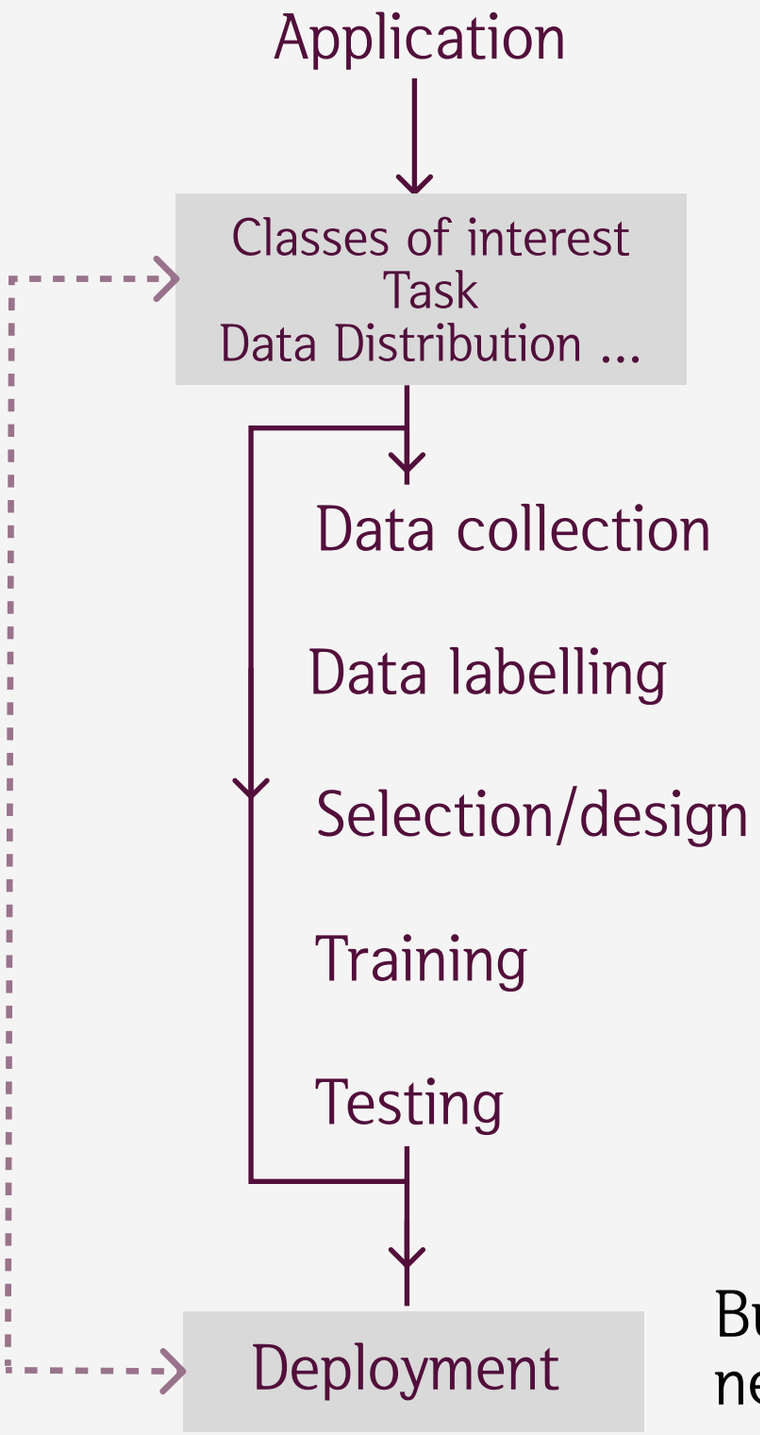


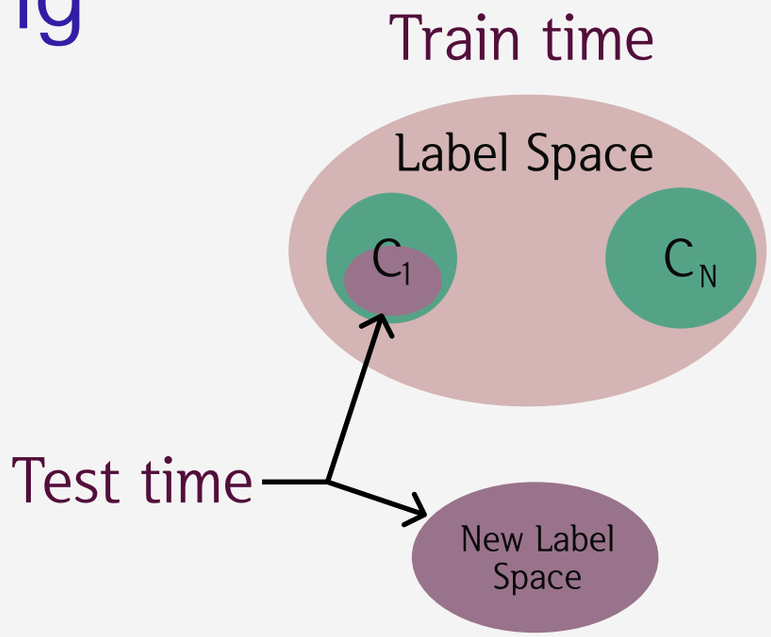


Up until now, we considered that deployment conditions are known during the development stage.

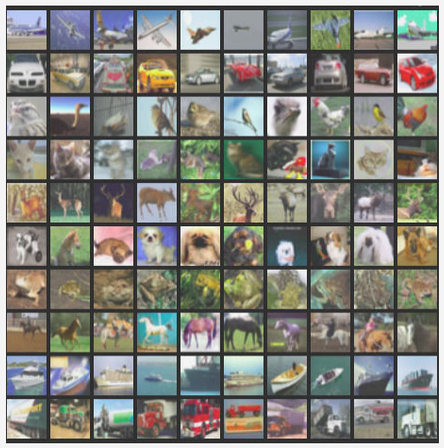
Spatial Contrastive Learning

Introduction





Train time



ImageNet

Test time

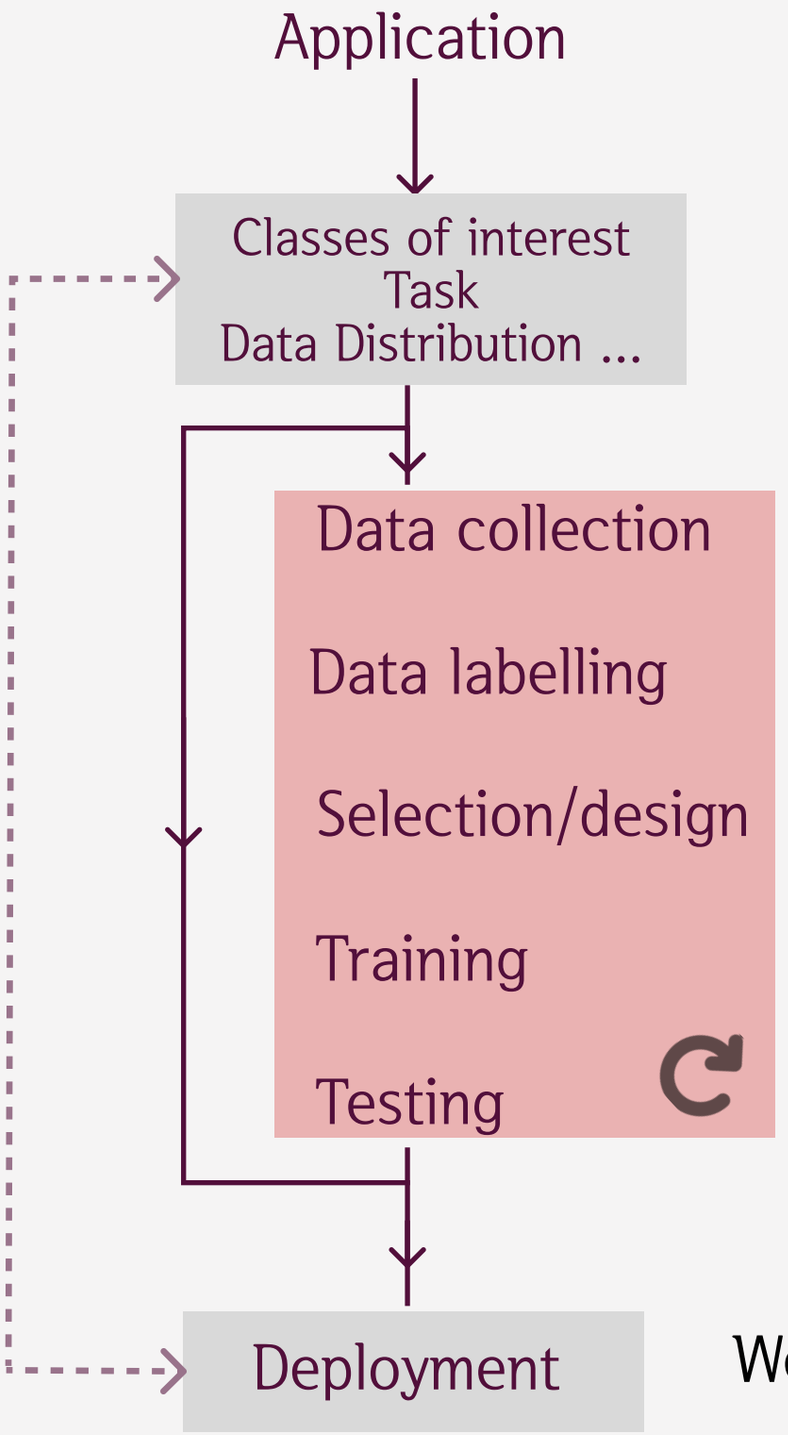
Fine-grained classification

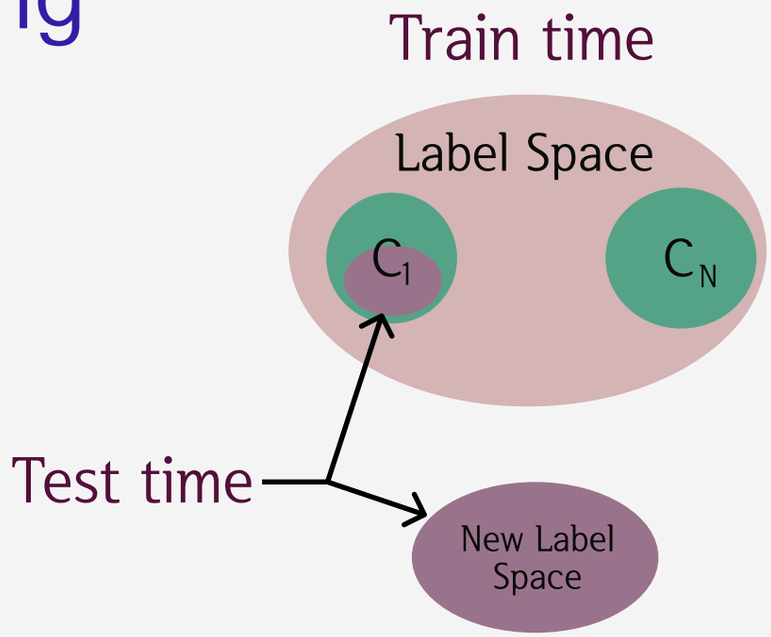
Different classification type

But what if they changed? eg, we want to solve a new, but related task.

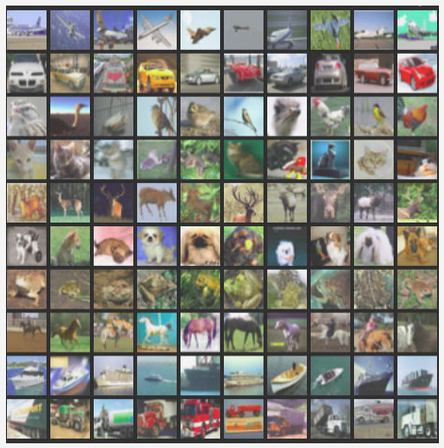
Spatial Contrastive Learning

Introduction



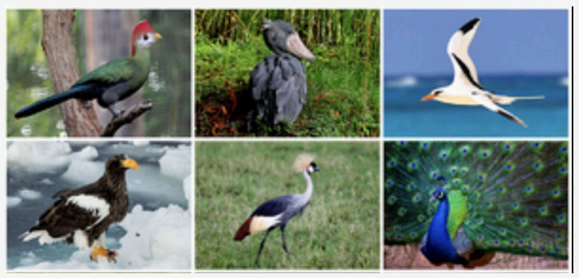


Train time



ImageNet

Test time



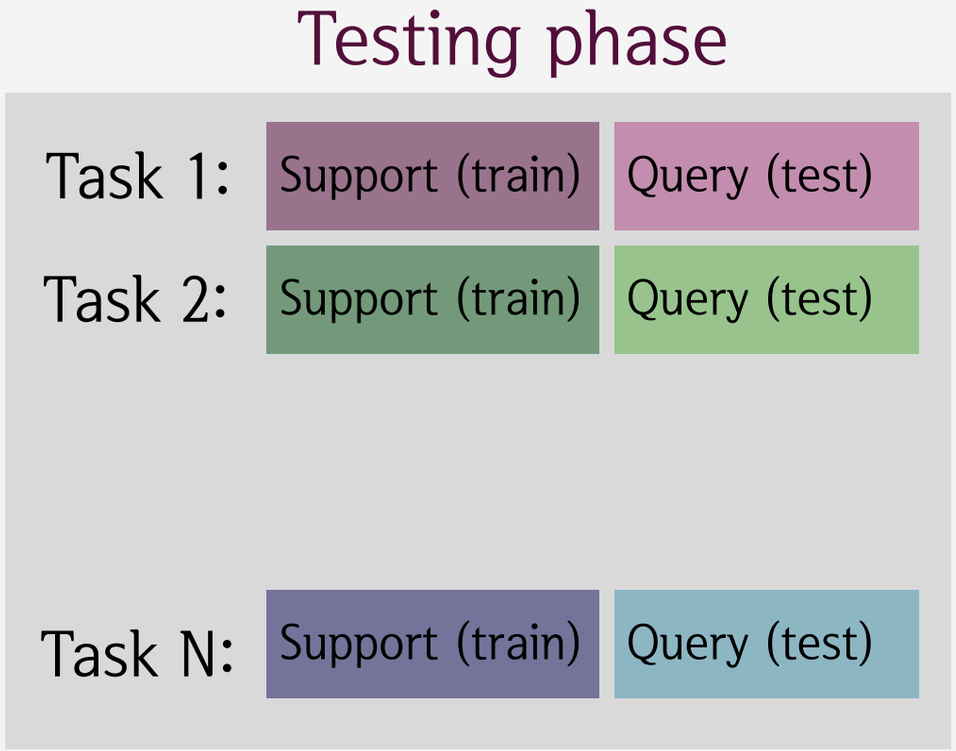
Fine-grained classification

Different classification type

We need to redo the whole pipeline.

Spatial Contrastive Learning

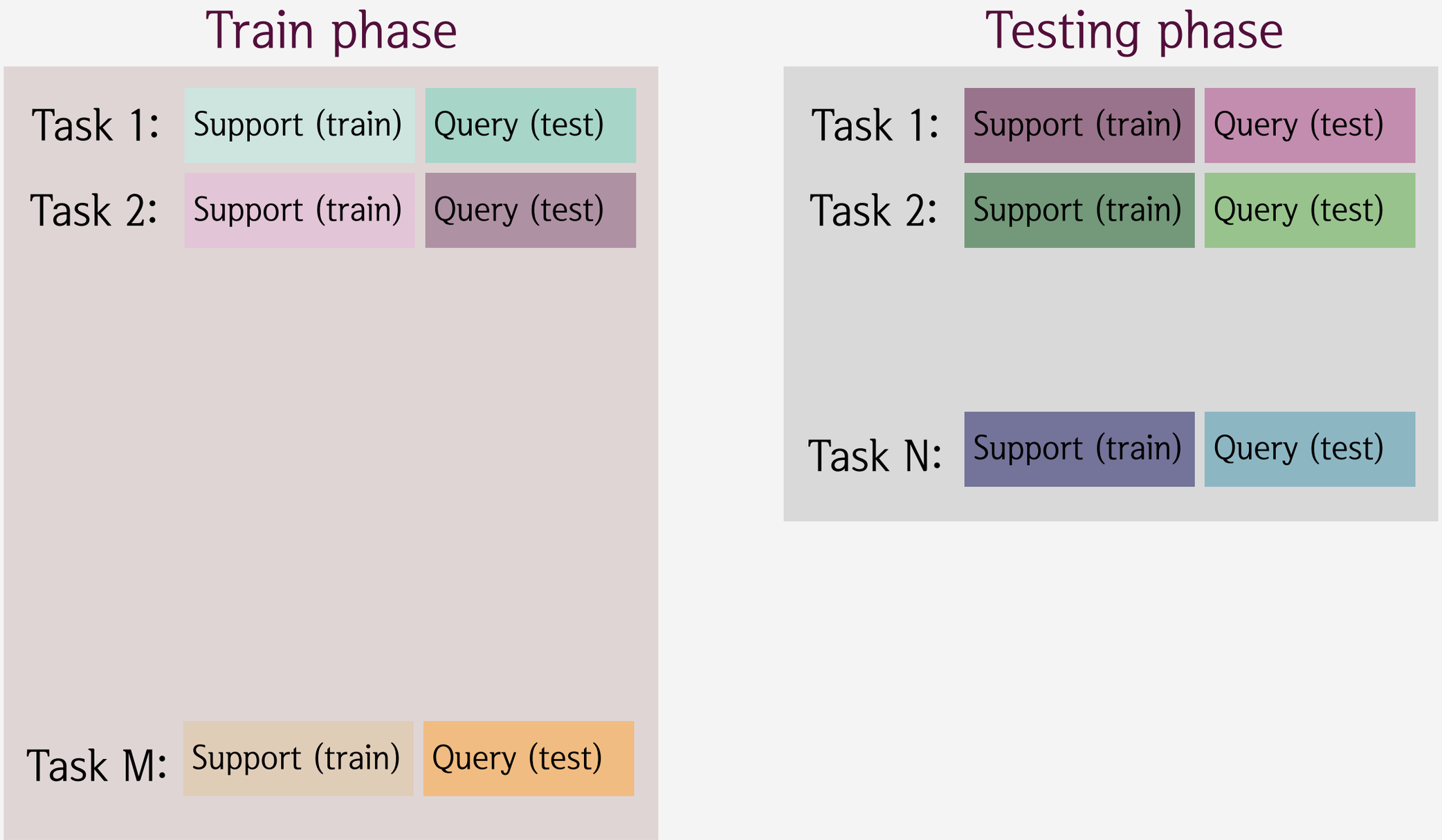
Introduction to few-shot learning



Few-shot learning tries to simulate a setup with similar test conditions.

Spatial Contrastive Learning

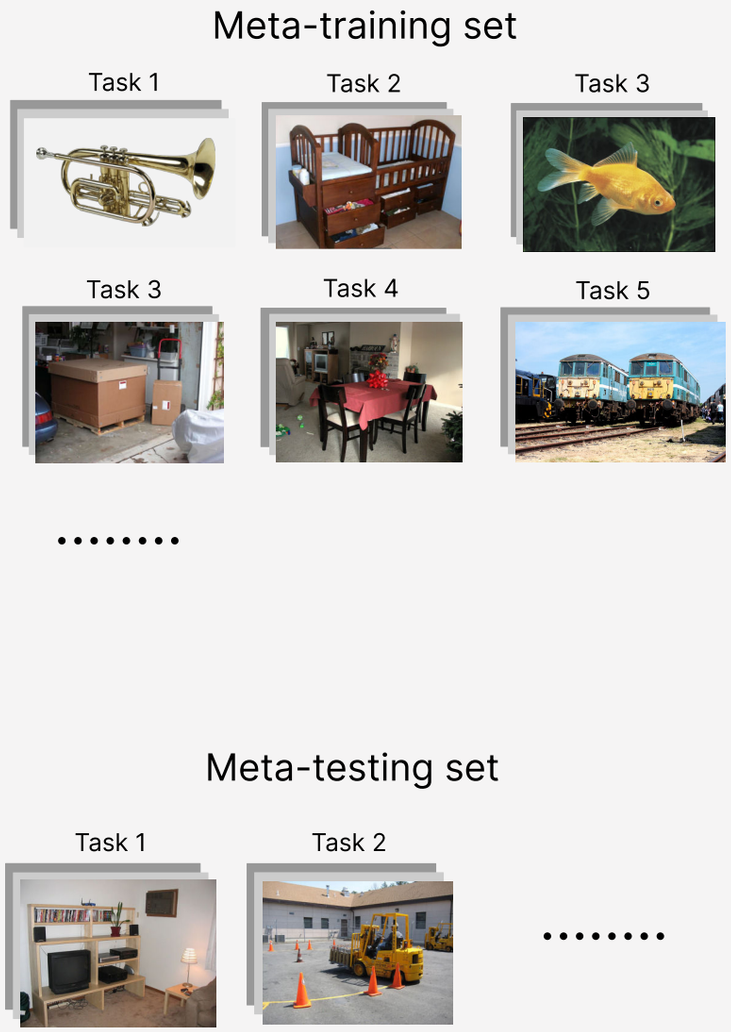
Introduction to few-shot learning



And a model is trained for fast adaptability using a similarly constructed (meta) train set.

Spatial Contrastive Learning

Meta-training and meta-testing for few-shot image classification

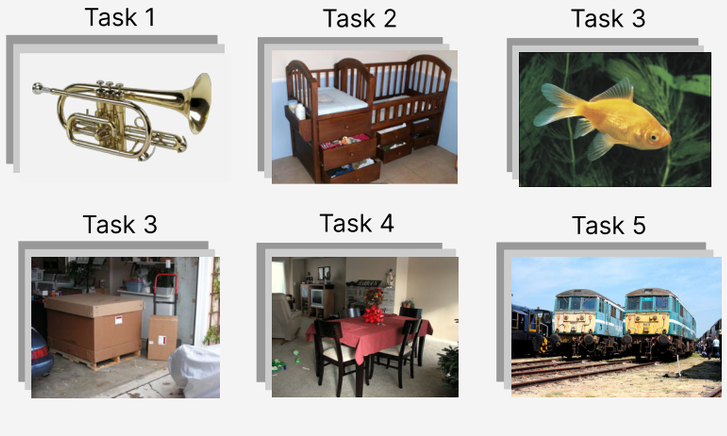


For few-shot classification, each task is a classification task.

Spatial Contrastive Learning

Meta-training and meta-testing for few-shot image classification

Meta-training set

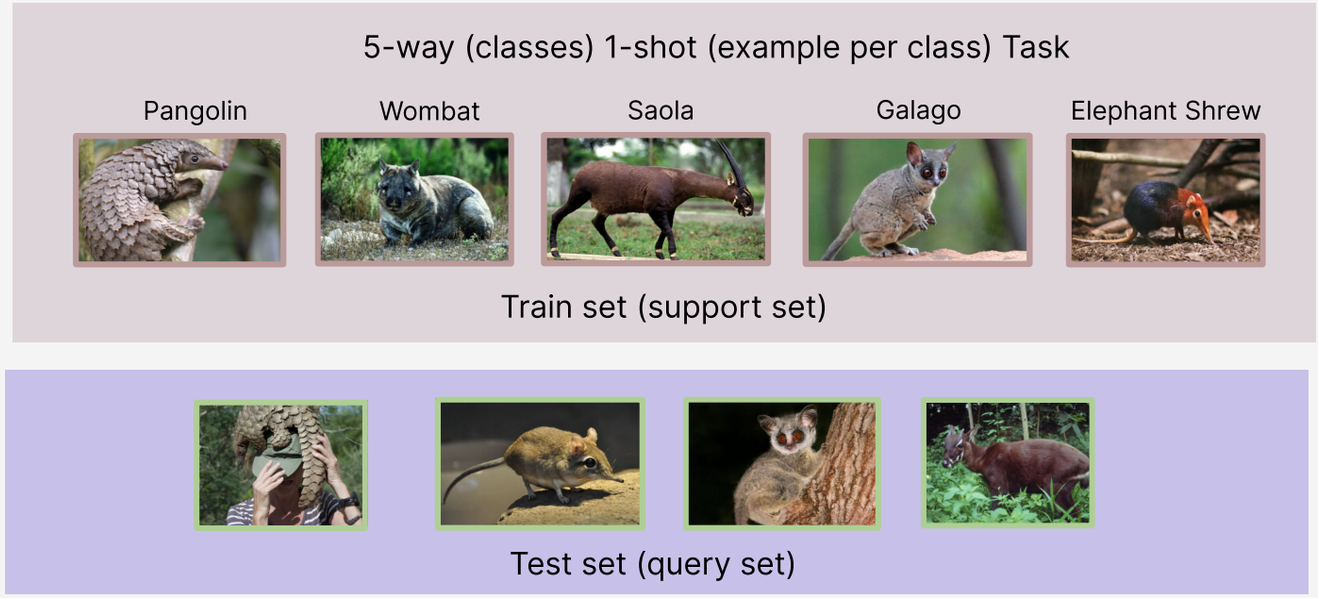


.....

Meta-testing set

.....

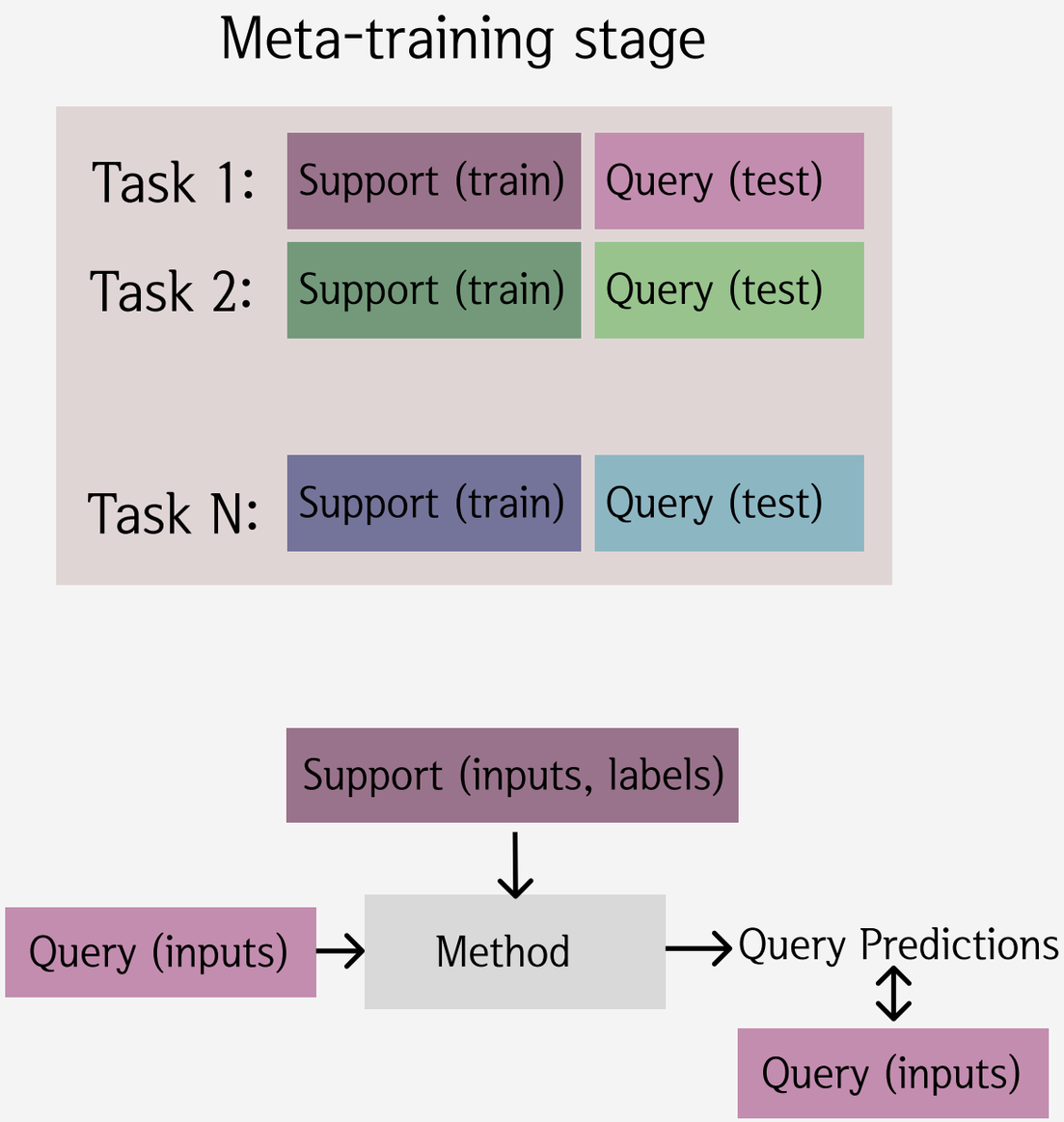
Task i: K-shot C-way



For few-shot classification, each task is a classification task, and is defined by the number of classes and the number of train examples per class.

Spatial Contrastive Learning

Few-shot image classification methods: meta learning

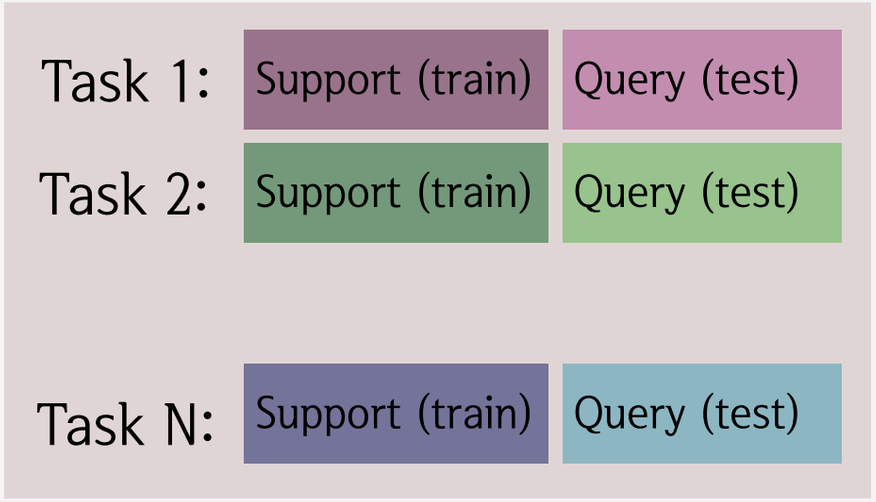


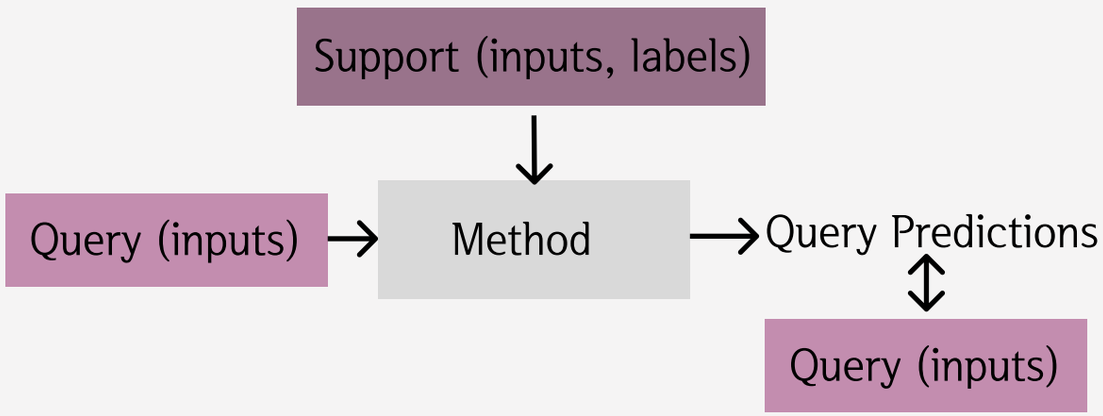
For meta-learning based approaches: query set predictions are explicitly conditioned on its support set.

Spatial Contrastive Learning

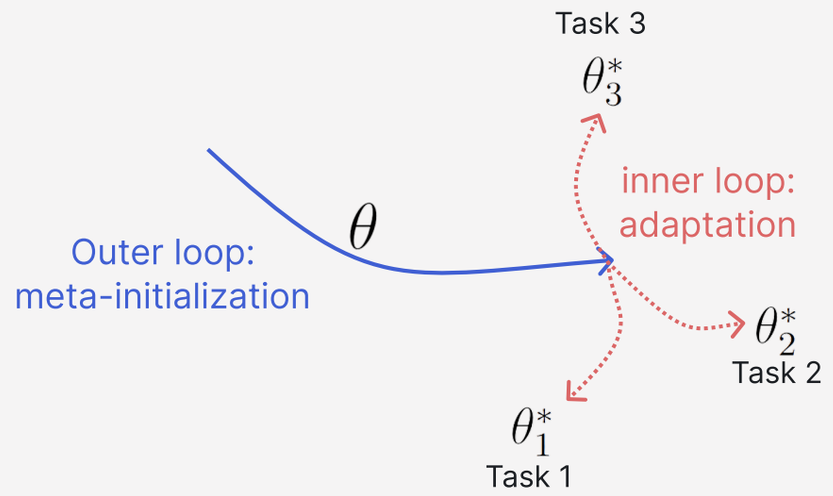
Few-shot image classification methods: meta learning

Meta-training stage

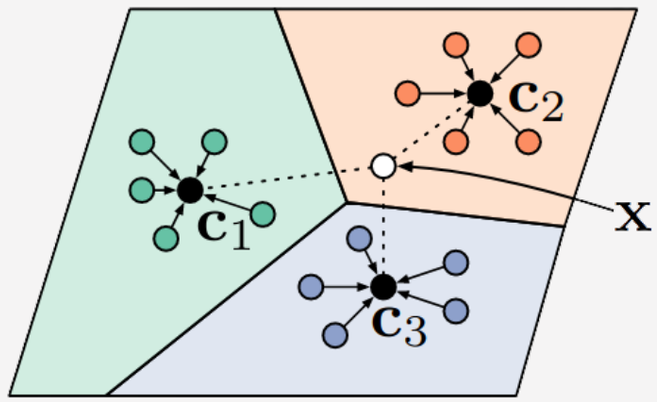




Optimisation based methods, e.g: MAML



Metric learning based methods, e.g: ProtoNets

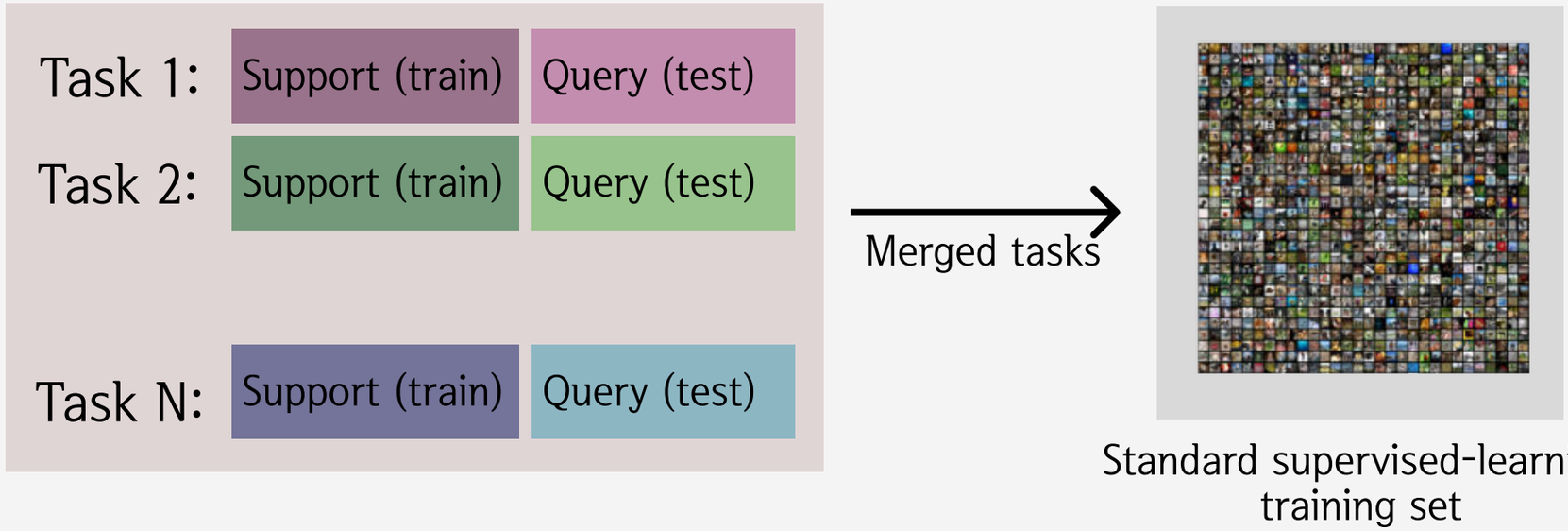


Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. Finn et al, 2017.
Prototypical Networks for Few-shot Learning. Snell el al, 2017.

Spatial Contrastive Learning

Few-shot image classification methods: transfer learning

Transforming the training stage into standard classification setup.

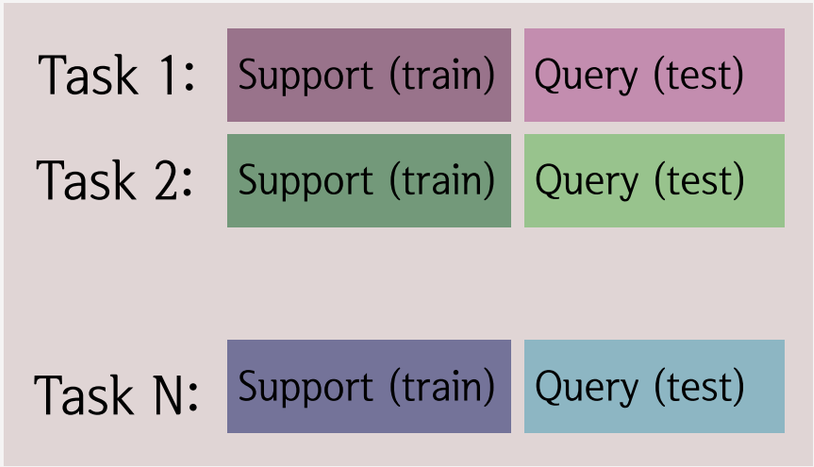


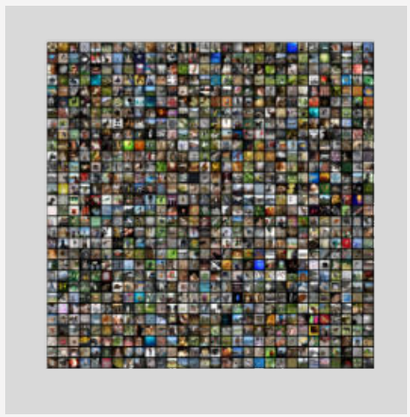
For transfer learning methods: first merge all training tasks into a single training set.

Spatial Contrastive Learning

Few-shot image classification methods: transfer learning

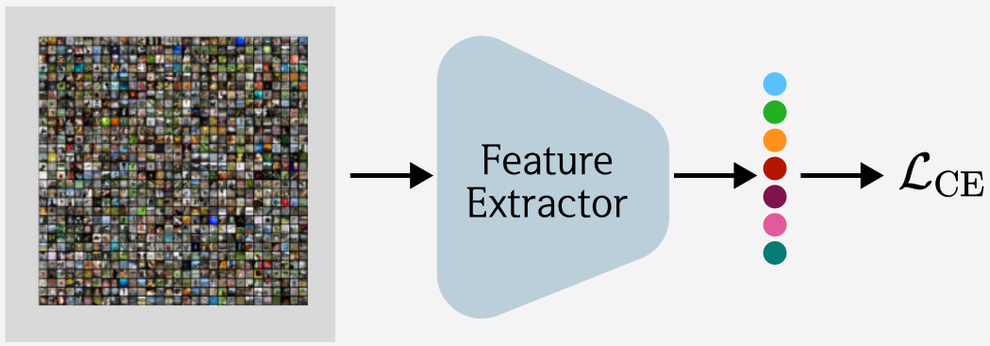
Transforming the training stage into standard classification setup.



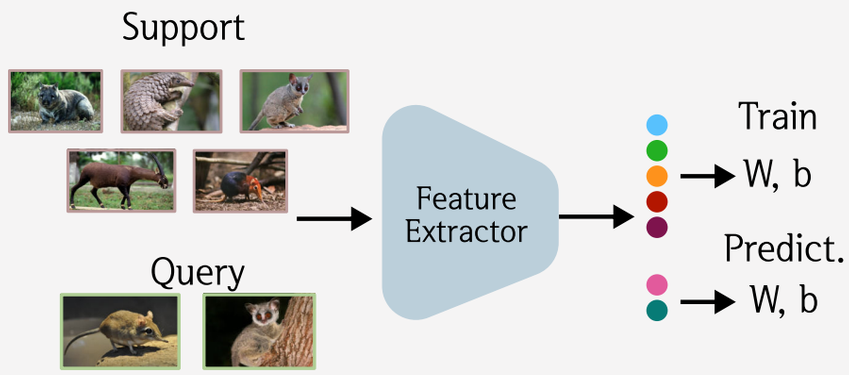


Standard supervised-learning training set

Training Phase



Testing Phase (per-task)

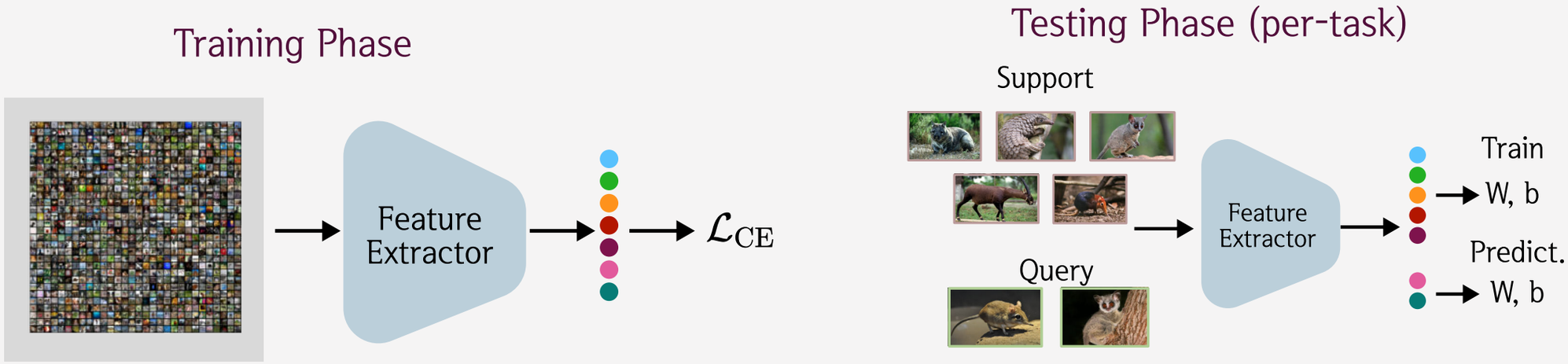


Then train a feature extractor on the full (merged) train set, then at test time, only learn a linear classifier for each task on top of the frozen features.

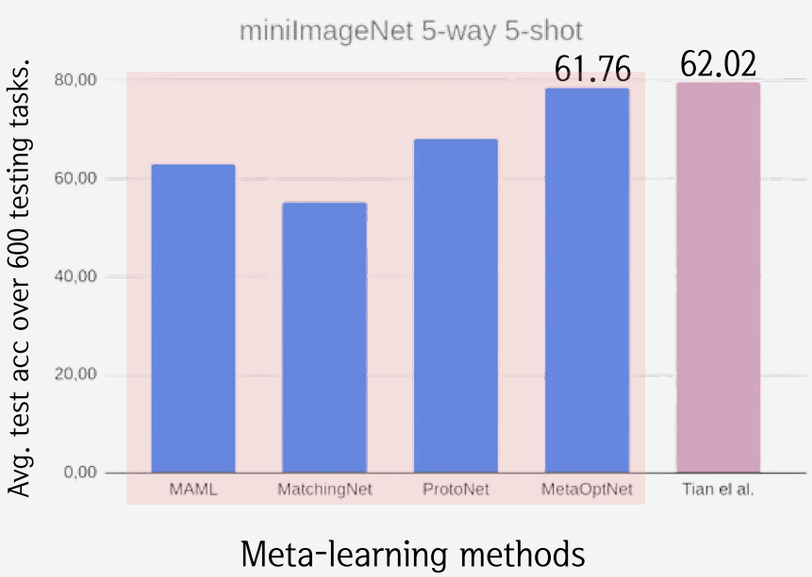
Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?. Tian et al. 2020.
A Closer Look at Few-shot Classification. Chen et al, 2019.

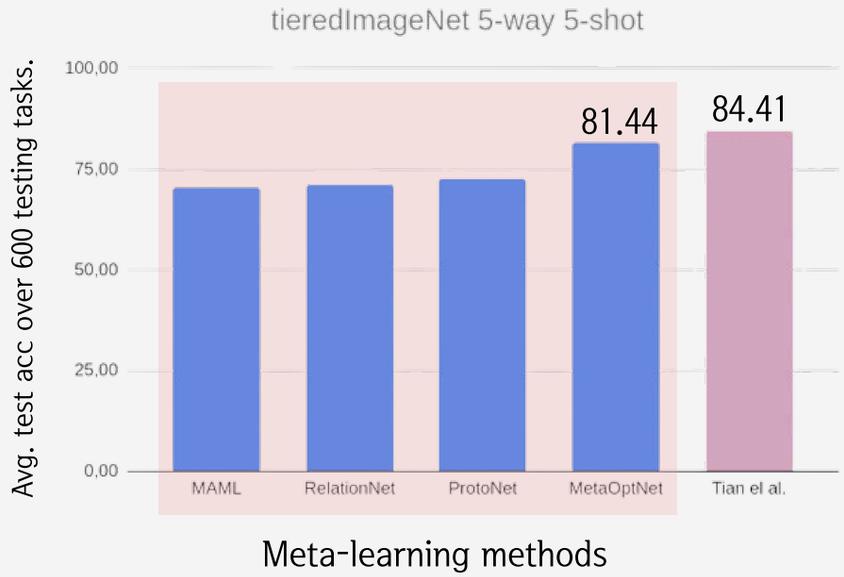
Spatial Contrastive Learning

Few-shot image classification methods: transfer learning



While simple, the transfer learning baseline remains very competitive with meta-learning methods, eg:



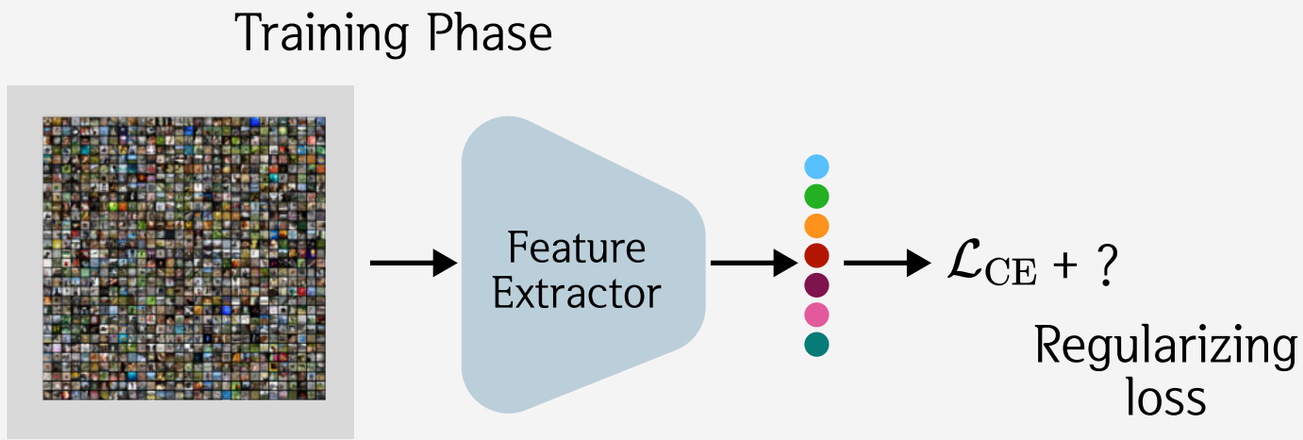


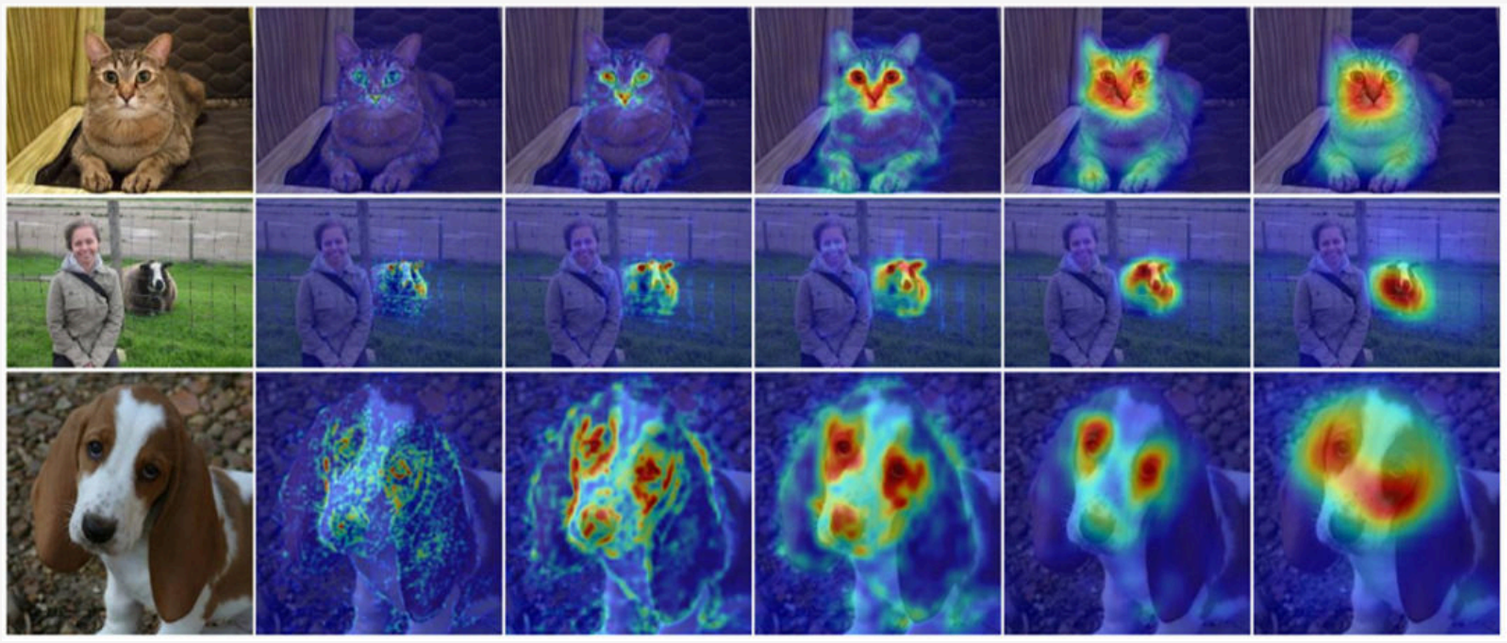
-> Learning good representation is important.

Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?. Tian et al. 2020.
A Closer Look at Few-shot Classification. Chen et al, 2019.

Spatial Contrastive Learning

Learning better representation



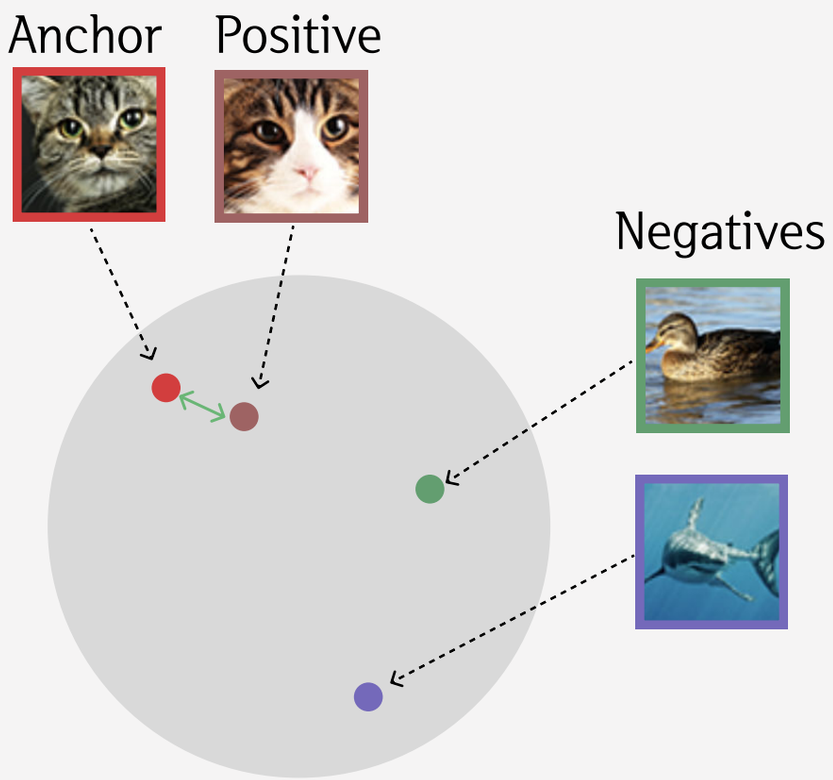


Given the importance of the quality of the learned representations for test-time adaptation, can we train using a better objective?

Generalized cross entropy loss for training deep neural networks with noisy labels. Zhang et al, 2018.
Learning imbalanced datasets with label-distribution-aware margin loss. Cao et al, 2019.
LayerCAM: Exploring Hierarchical Class Activation Maps for Localization. Jiang et al, 2021.

Spatial Contrastive Learning

Introduction Contrastive Learning



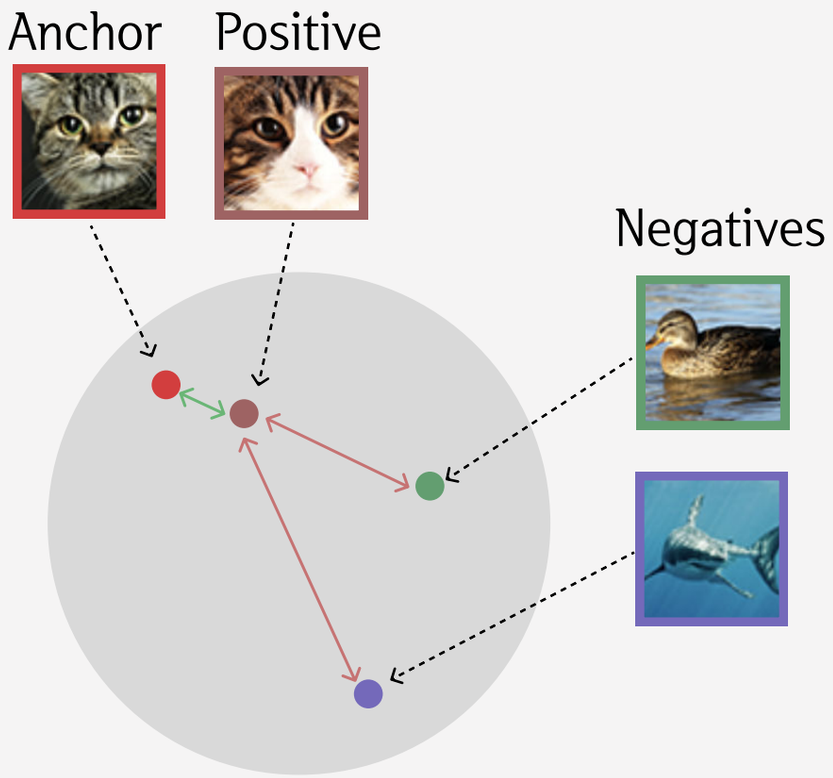
- Pull together similar inputs:
  - Anchor & Positive

How about a contrastive learning based loss?

Supervised Contrastive Learning. Khosla et al. 2020.
A Simple Framework for Contrastive Learning of Visual Representations. Chen at al, 2020.

Spatial Contrastive Learning

Introduction Contrastive Learning



- Pull together similar inputs:
  - Anchor & Positive
- Push away dissimilar inputs:
  - Anchor & Negatives

$$-\log \frac{\exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{i \neq k} \cdot \exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_k)/\tau)}$$

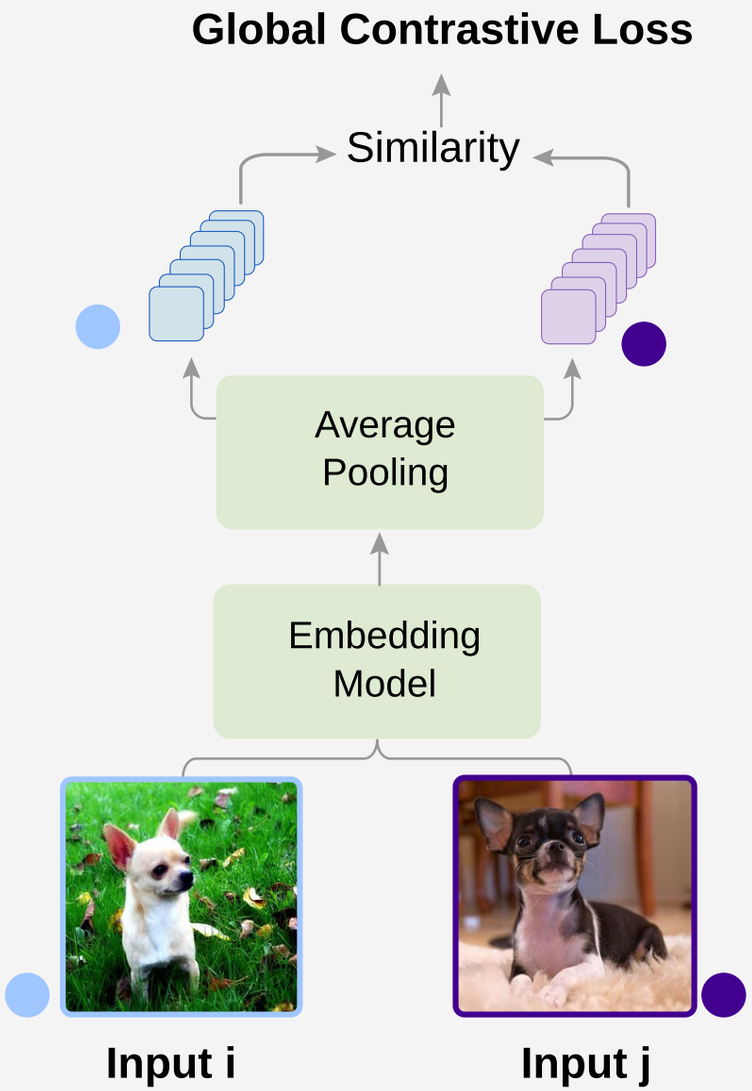
- Positives: augment or labels.
- Negatives: at random or different labels.

How about a contrastive learning based loss?

Supervised Contrastive Learning. Khosla et al. 2020.
A Simple Framework for Contrastive Learning of Visual Representations. Chen at al, 2020.

Spatial Contrastive Learning

A Closer look into contrastive learning

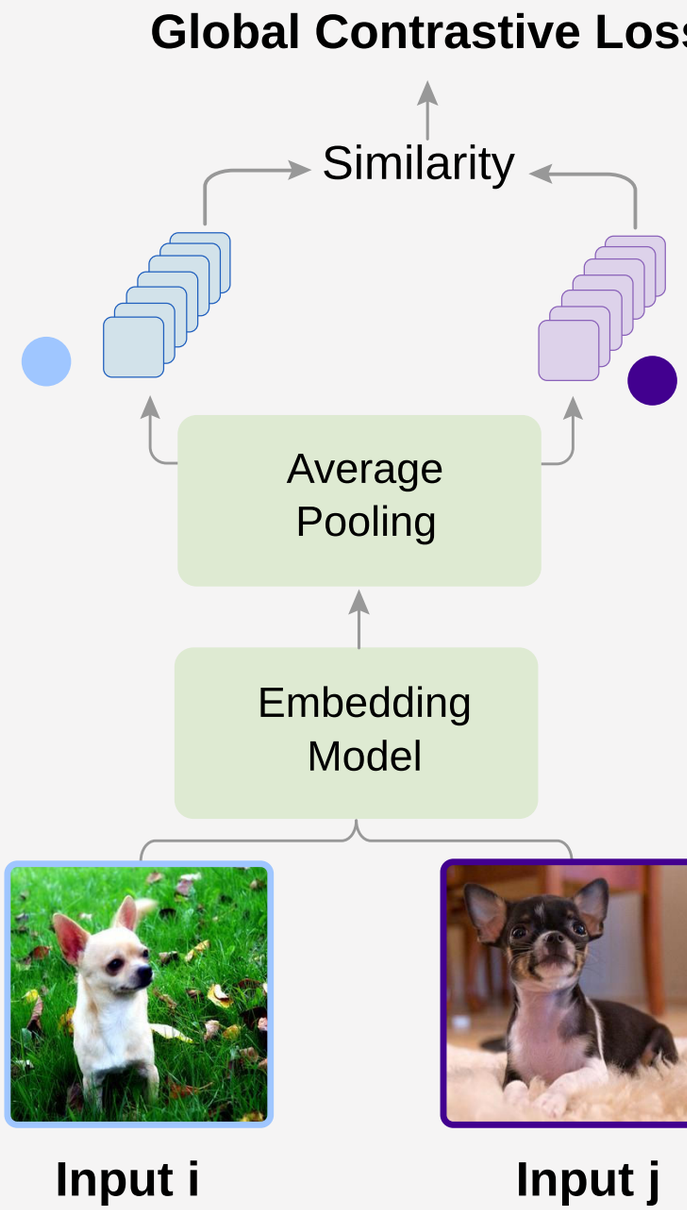


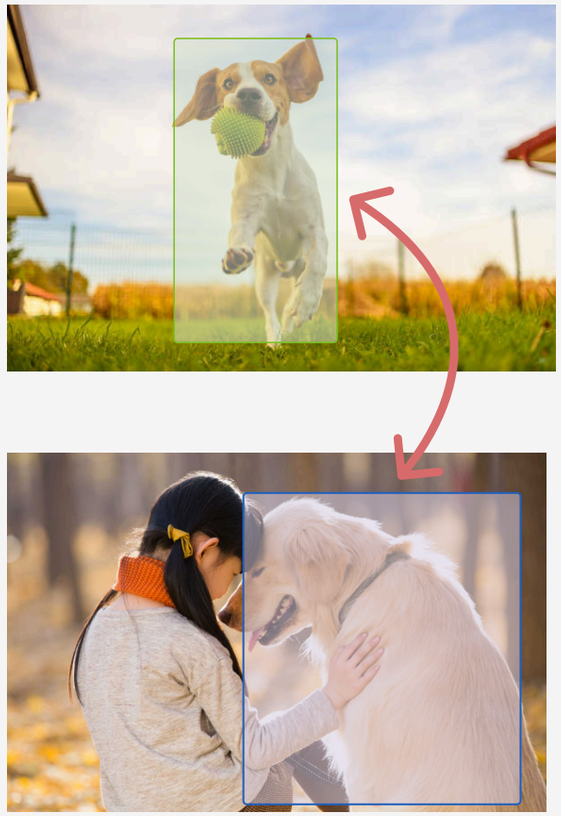
$$-\log \frac{\exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{i \neq k} \cdot \exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_k)/\tau)}$$

In contrastive learning, the similarity score is based on the global averaged features.

Spatial Contrastive Learning

A better objective: Contrastive Learning





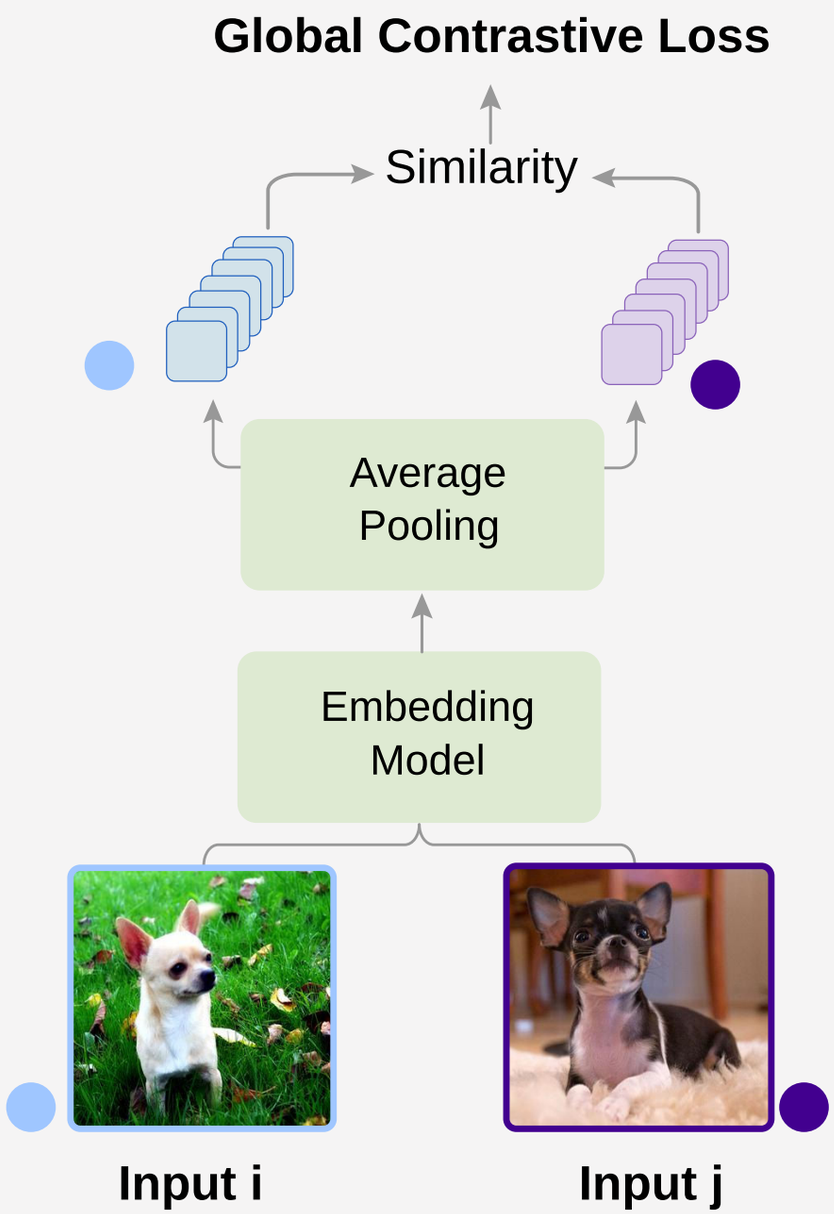
We might align irrelevant objects:

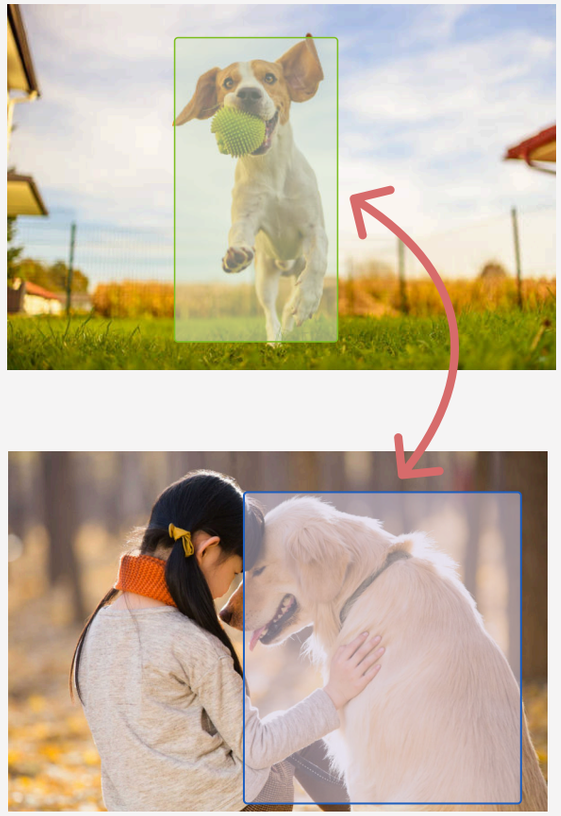
- Dog & person
- Dog & grass

But this can result in a misalignment of features during training.

Spatial Contrastive Learning

A better objective: Contrastive Learning





We might align irrelevant objects:

- Dog & person
- Dog & grass

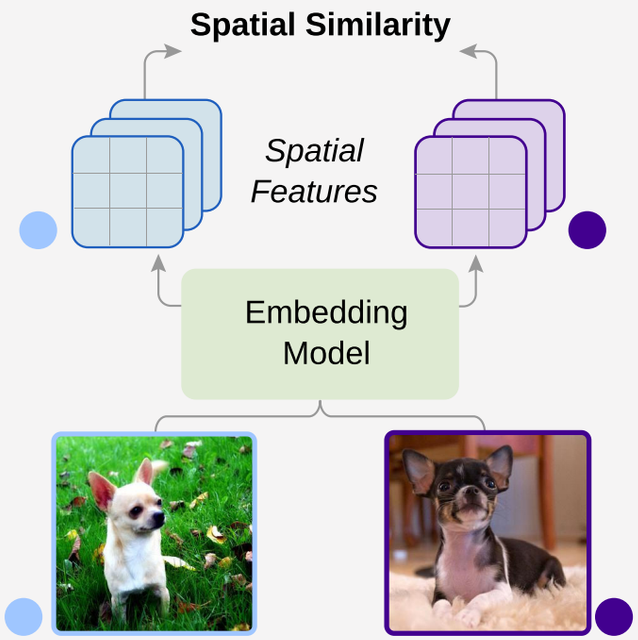
In contrastive learning, this is avoided by:

- Training for many epochs.
- Using very large batches (or a queue)
- Training on large datasets.

How to adjust the objective for a few-shot appropriate contrastive loss?

Spatial Contrastive Learning

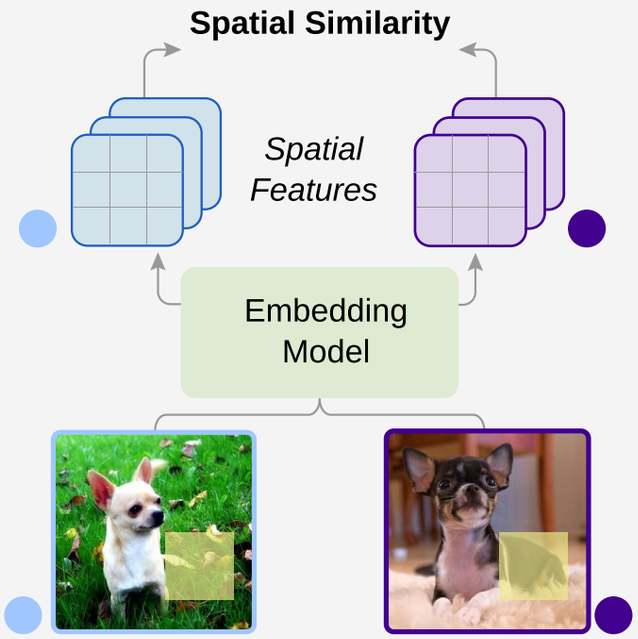
Using the spatial features for similarity computation



Can we use the spatial features directly without an avg. pool.

Spatial Contrastive Learning

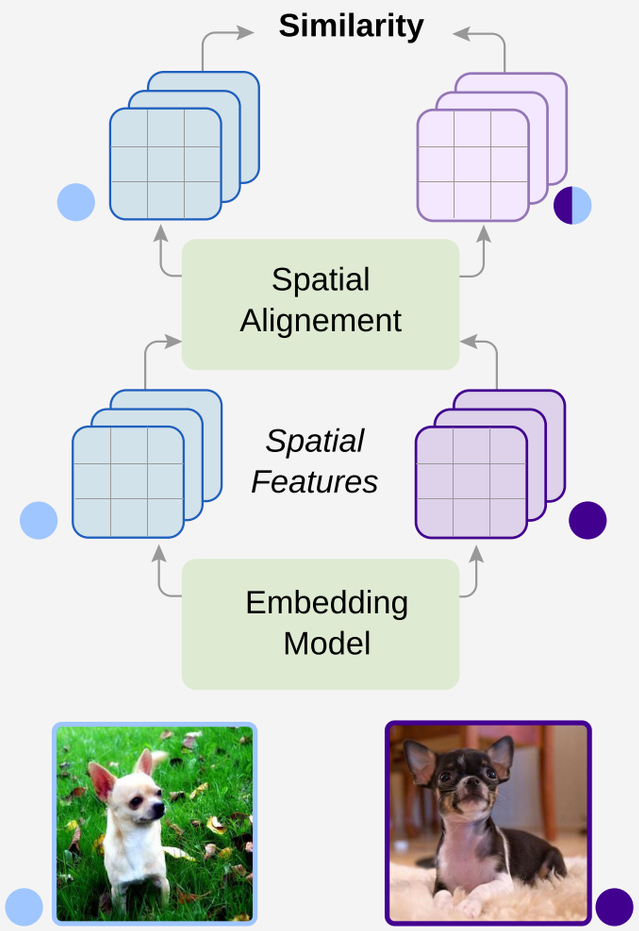
The need for spatial alignment



To use the spatial features first, we first need to align them.

Spatial Contrastive Learning

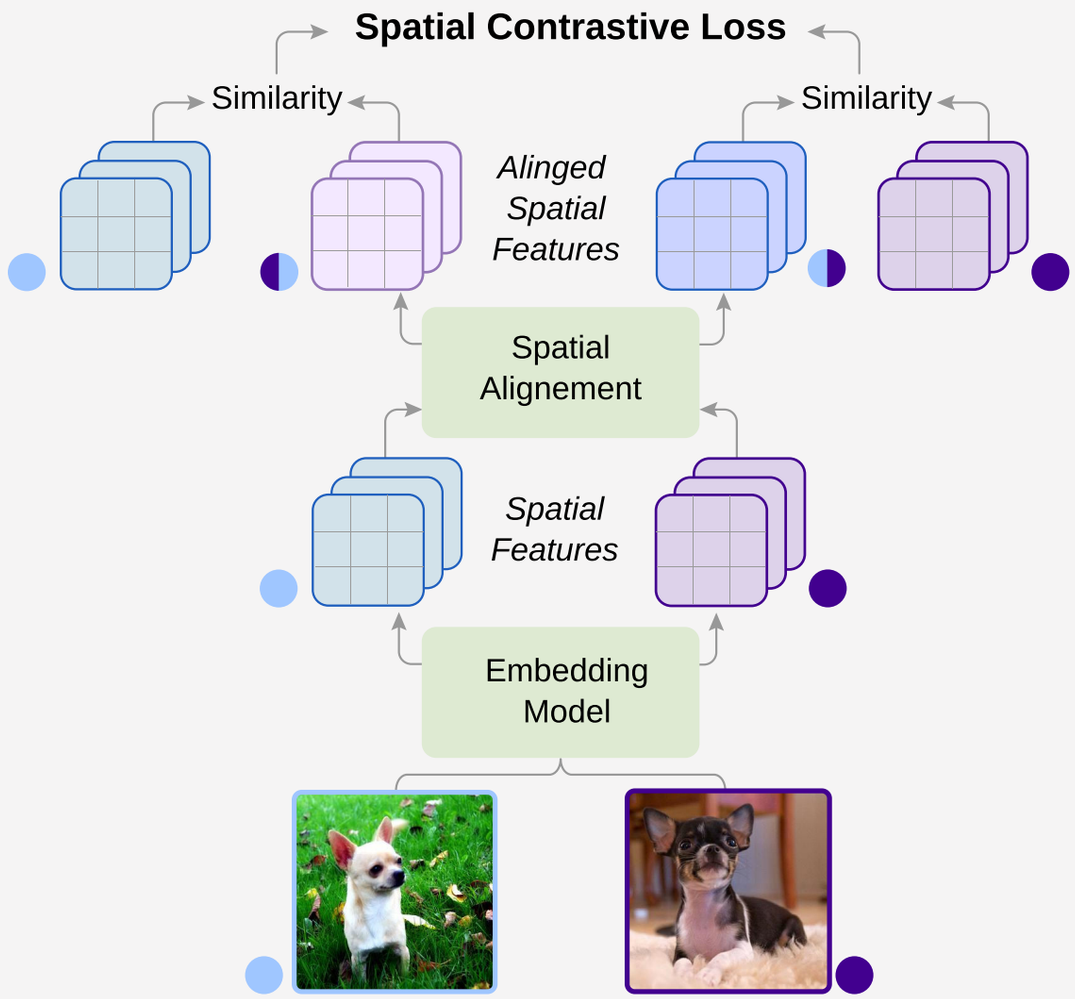
Spatial alignment & similarity computation



So, first, we need to align the positive with the anchor before computing the similarity.

Spatial Contrastive Learning

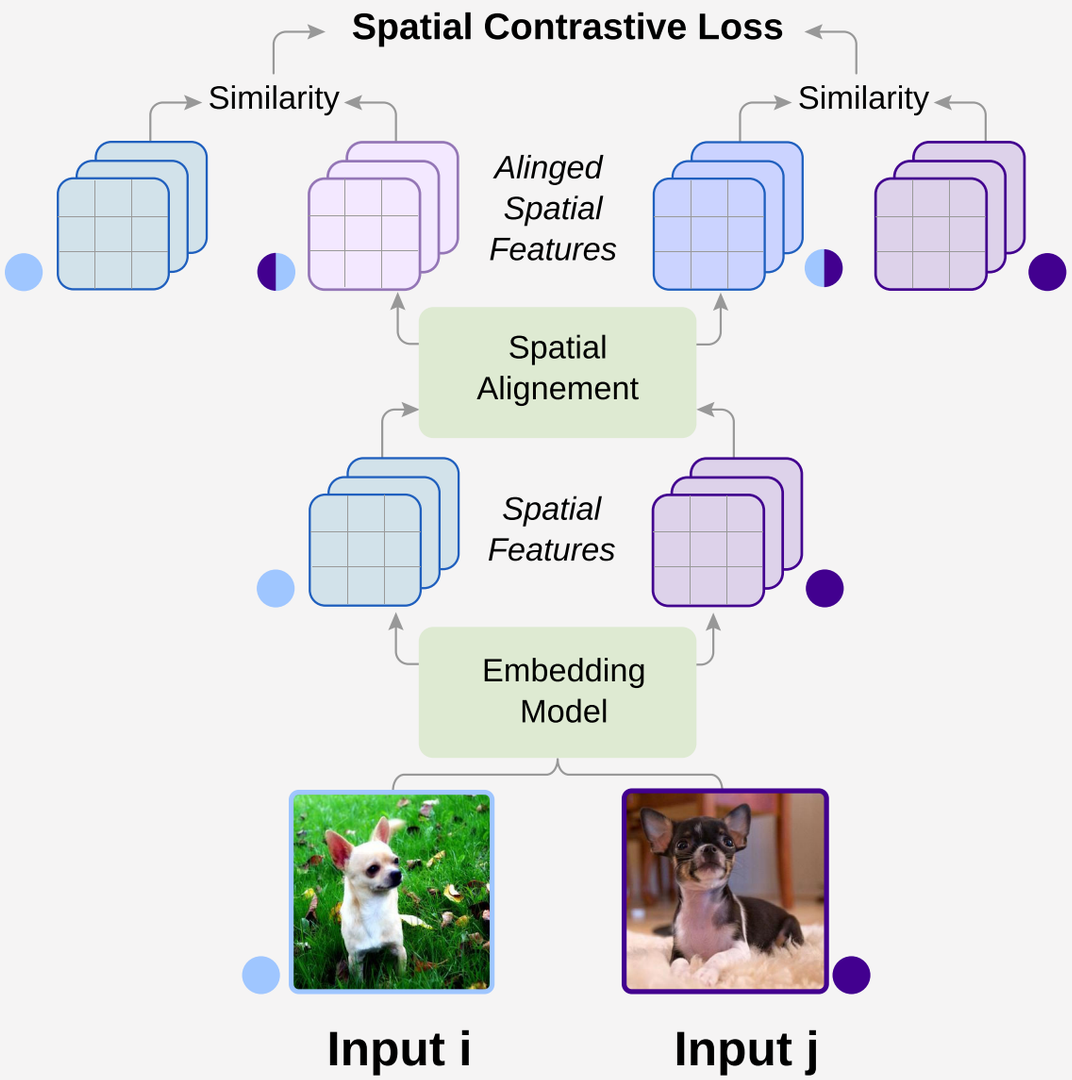
Spatial alignment & similarity computation

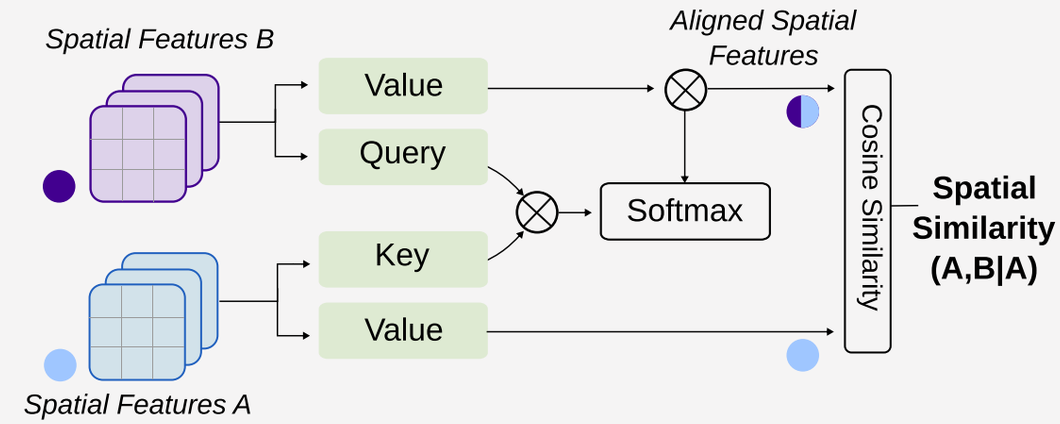


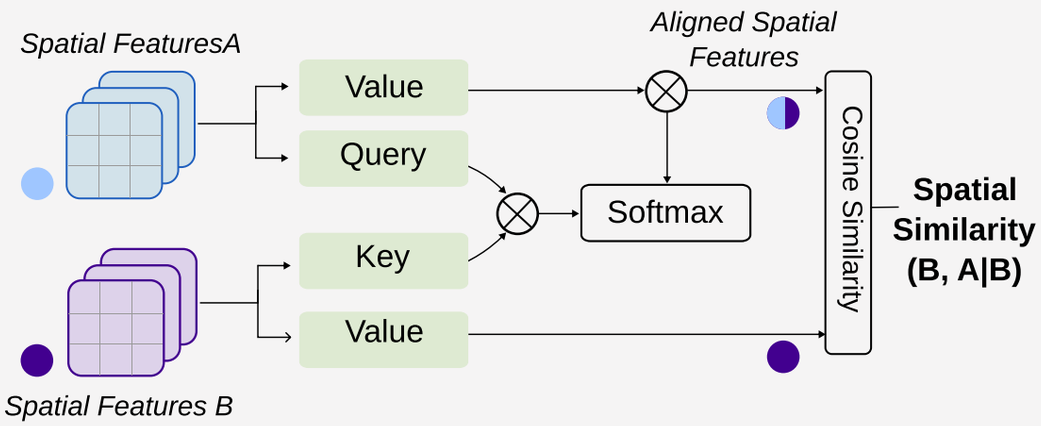
Make the loss symmetric: we compute both alignments, positive to anchor and anchor to positive.

Spatial Contrastive Learning

Spatial alignment & similarity computation







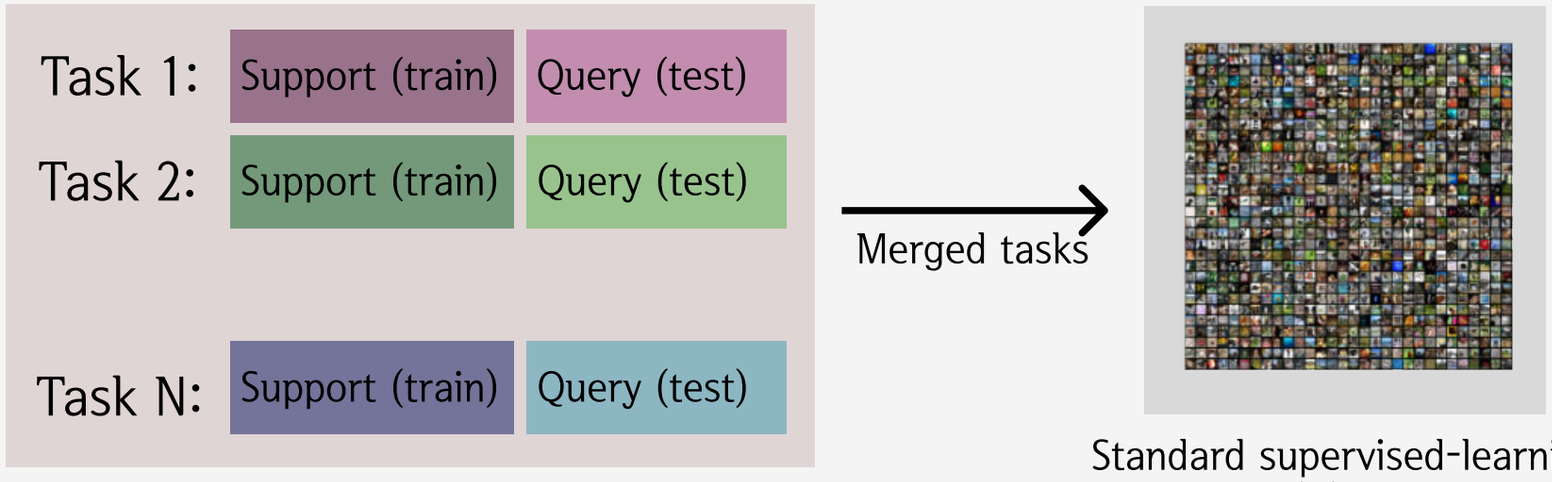
Spatial Similarity (A, B) = Spatial Similarity (A, B|A) + Spatial Similarity (B, A|B)

$$-\log \frac{\exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{i \neq k} \cdot \exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_k)/\tau)}$$

Make the loss symmetric: we compute both alignments, positive to anchor and anchor to positive ... via attention.

Spatial Contrastive Learning

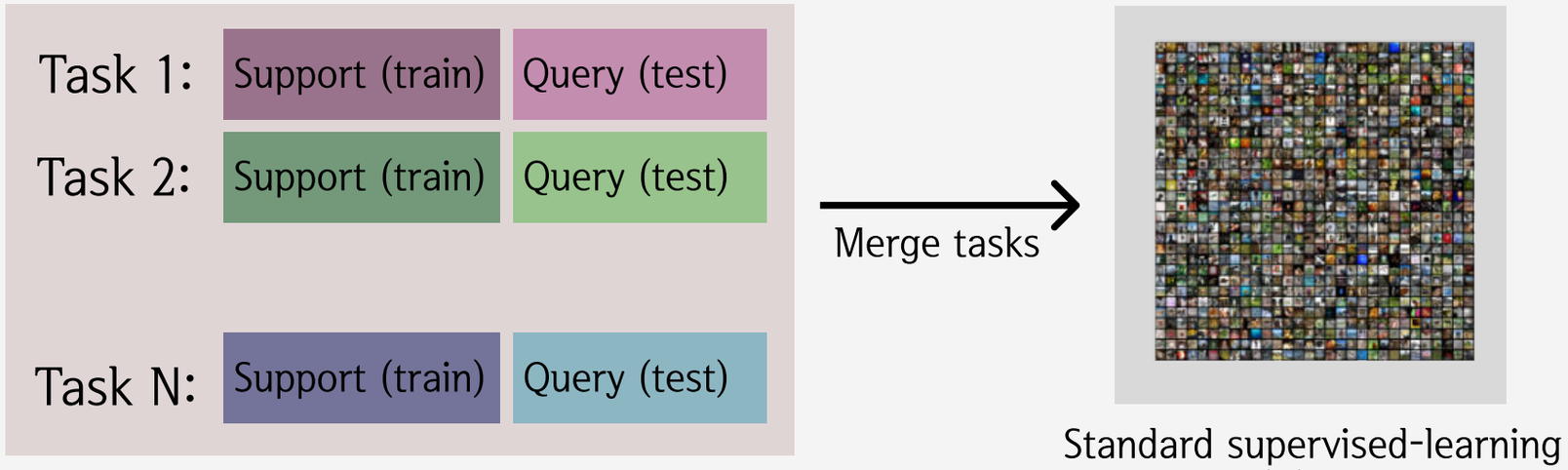
SCL for transfer learning based few-shot classification

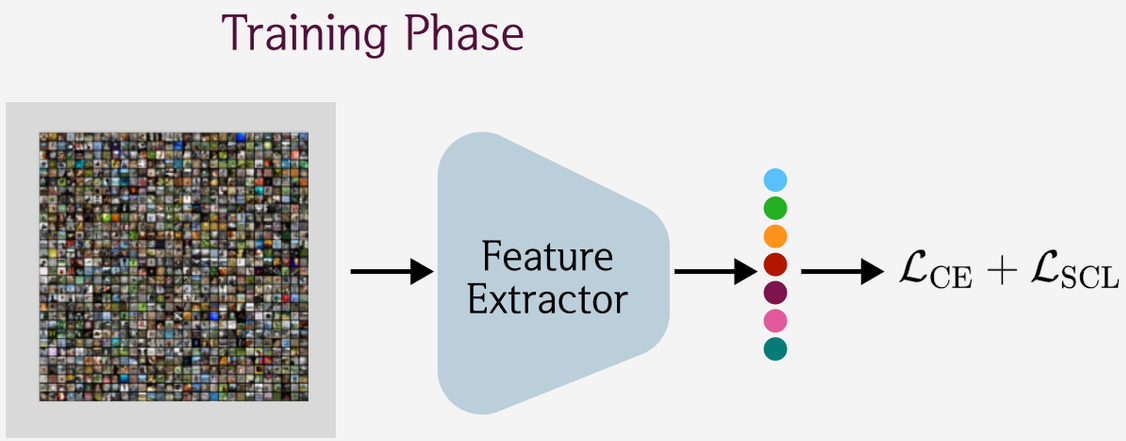


So first, we merge all of the training tasks into a single training set.

Spatial Contrastive Learning

SCL for transfer learning based few-shot classification

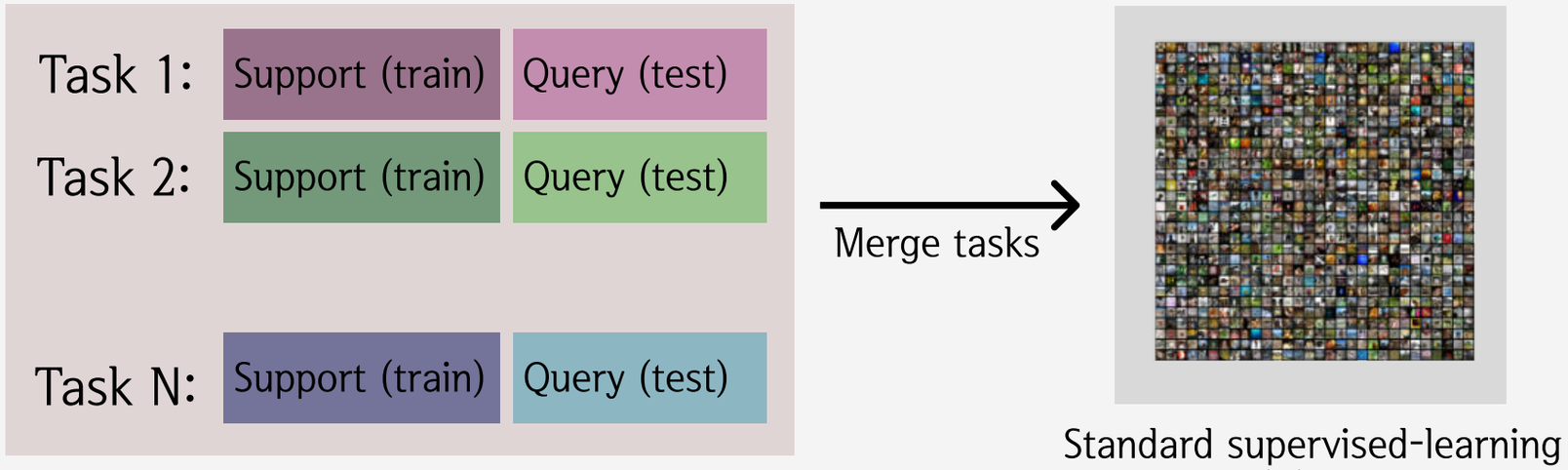


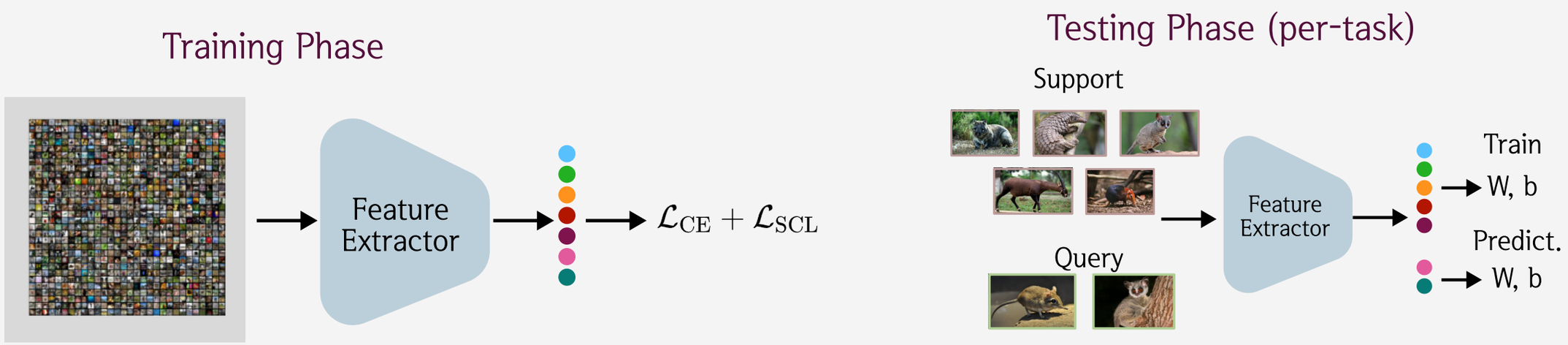


We train the feature extractor on the merged set using the CE loss, and the proposed SCL loss.

Spatial Contrastive Learning

SCL for transfer learning based few-shot classification





At test time, we train a linear classifier on the extracted & L2 normalised features of the support, and use it to predict on the query set.

For eval, we report the avg. accuracy over 600 testing tasks.

Spatial Contrastive Learning

Results: the effects of spatial contrastive loss

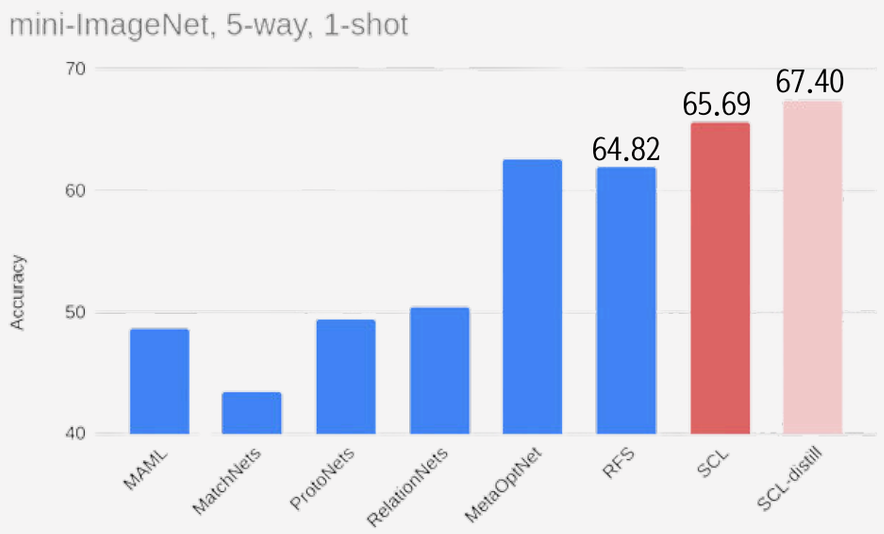
Loss Function	Aug.	*mini*-ImageNet, 5-way 1-shot	*mini*-ImageNet, 5-way 5-shot	CIFAR-CS, 5-way 1-shot	CIFAR-CS, 5-way 5-shot
CE		61.8 ± 0.7	79.7 ± 0.6	71.3 ± 0.9	86.1 ± 0.6
CE	✓	61.8 ± 0.8	78.6 ± 0.5	71.9 ± 0.9	86.3 ± 0.5
CE + SS-GC	✓	62.7 ± 0.7	81.0 ± 0.6	70.9 ± 0.9	84.5 ± 0.6
CE + SS-SC	✓	64.0 ± 0.8	81.5 ± 0.5	72.1 ± 0.8	86.2 ± 0.6
CE + SS-GC + SS-SC	✓	62.8 ± 0.8	81.1 ± 0.6	69.0 ± 0.9	85.0 ± 0.6
CE + GC	✓	65.0 ± 0.8	81.6 ± 0.5	74.0 ± 0.8	87.3 ± 0.6
CE + SC	✓	$\mathbf{65.7 \pm 0.8}$	$\mathbf{82.5 \pm 0.5}$	75.0 ± 0.9	87.4 ± 0.6
CE + GC + SC	✓	65.0 ± 0.8	81.3 ± 0.5	$\mathbf{76.0 \pm 0.7}$	$\mathbf{87.5 \pm 0.5}$

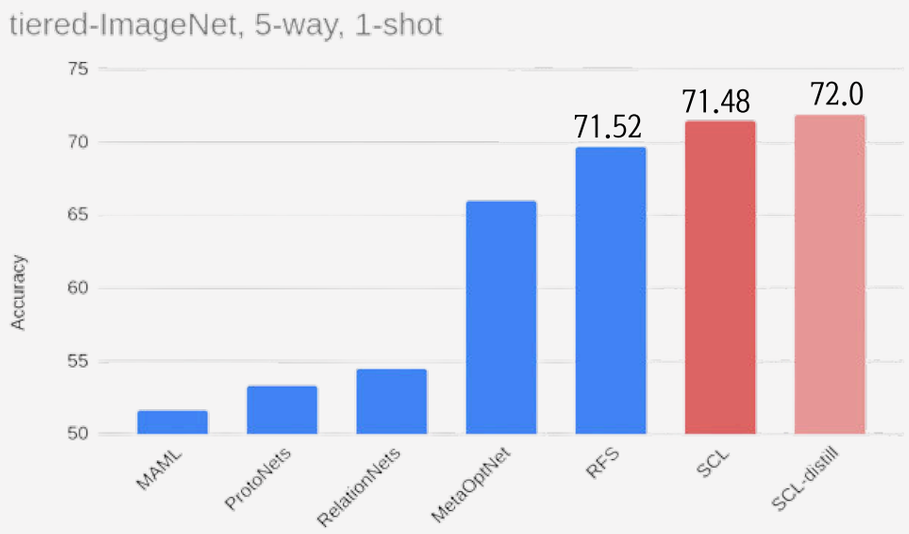
Spatial Contrastive Learning

Results: the effects of spatial contrastive loss

Loss Function	Aug.	*mini*-ImageNet, 5-way 1-shot	*mini*-ImageNet, 5-way 5-shot	CIFAR-CS, 5-way 1-shot	CIFAR-CS, 5-way 5-shot
CE		61.8 ± 0.7	79.7 ± 0.6	71.3 ± 0.9	86.1 ± 0.6
CE	✓	61.8 ± 0.8	78.6 ± 0.5	71.9 ± 0.9	86.3 ± 0.5
CE + SS-GC	✓	62.7 ± 0.7	81.0 ± 0.6	70.9 ± 0.9	84.5 ± 0.6
CE + SS-SC	✓	64.0 ± 0.8	81.5 ± 0.5	72.1 ± 0.8	86.2 ± 0.6
CE + SS-GC + SS-SC	✓	62.8 ± 0.8	81.1 ± 0.6	69.0 ± 0.9	85.0 ± 0.6
CE + GC	✓	65.0 ± 0.8	81.6 ± 0.5	74.0 ± 0.8	87.3 ± 0.6
CE + SC	✓	$\mathbf{65.7 \pm 0.8}$	$\mathbf{82.5 \pm 0.5}$	75.0 ± 0.9	87.4 ± 0.6
CE + GC + SC	✓	65.0 ± 0.8	81.3 ± 0.5	$\mathbf{76.0 \pm 0.7}$	$\mathbf{87.5 \pm 0.5}$

Results: Comparison with SOTA





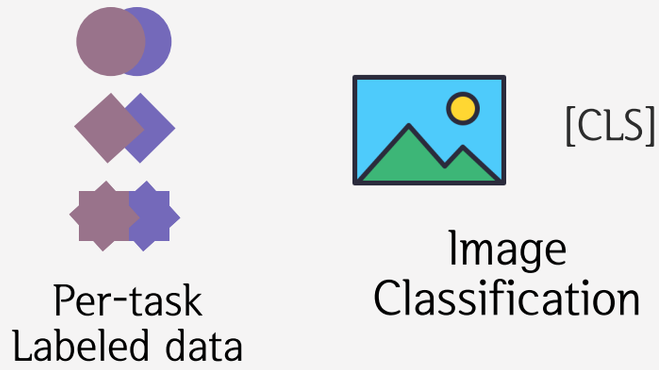
The usage of SCL during the training phase as a data dependent regulariser results in overall better performances.

Contribution 3: Spatial Contrastive Learning

Conclusion

We considered:

- Few-shot Learning
- Image Classification



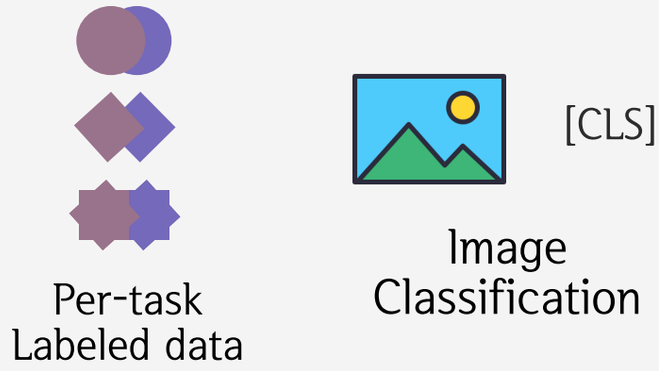
We introduced Spatial contrastive learning, a novel contrastive loss that uses the spatial features to learn better representations for test-time adaptation.

Contribution 3: Spatial Contrastive Learning

Conclusion

We considered:

- Few-shot Learning
- Image Classification



We introduced Spatial contrastive learning, a novel contrastive loss that uses the spatial features to learn better representations for test-time adaptation.

Additional details:

- A feature distillation loss based on SCL for better results.
- Extension to ProtoNet, a metric learning few-shot approach.

Contribution 3: Spatial Contrastive Learning

Conclusion

Limitation:

- Importance of contrastive loss: the loss plays a role of a regulariser, can we use other types of regularisation that are more computationally efficient?
- Applicability: we limited our selves to few-shot learning, but the proposed method is general and can be used for self-supervised learning and other tasks.

Contribution 3: Spatial Contrastive Learning

Conclusion

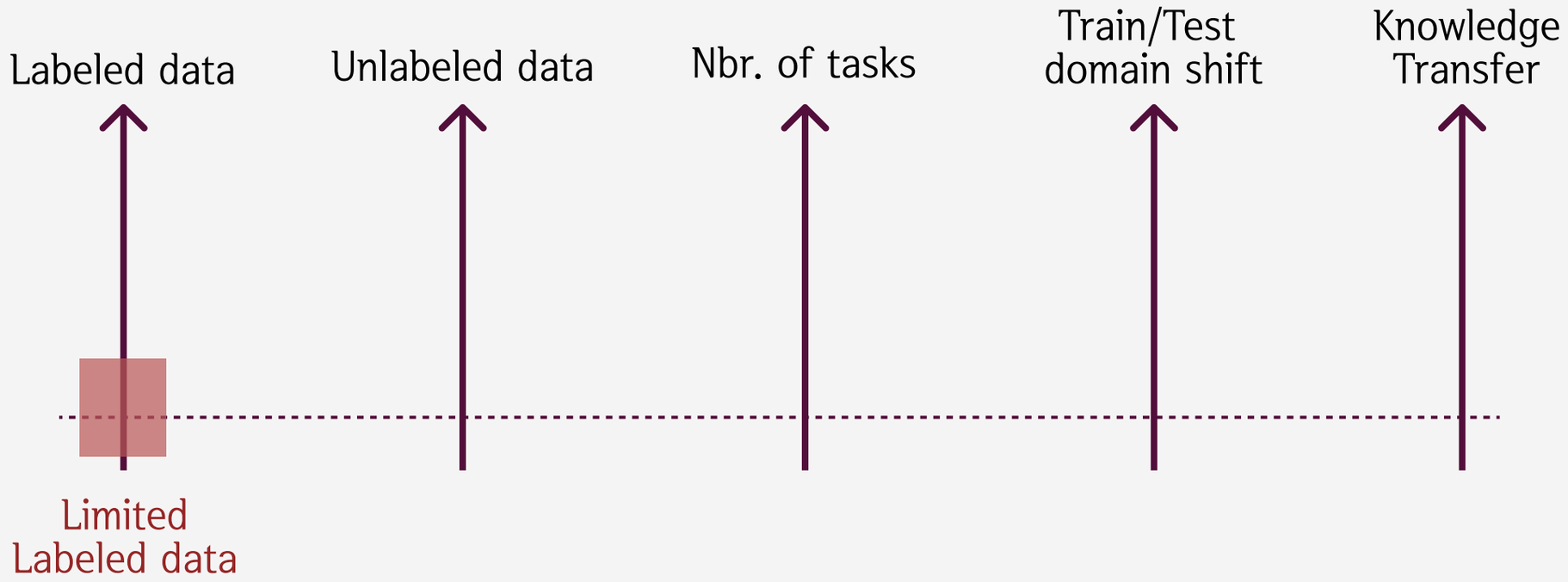
Limitation:

- Importance of contrastive loss: the loss plays a role of a regulariser, can we use other types of regularisation that are more computationally efficient?
- Applicability: we limited our selves to few-shot learning, but the proposed method is general and can be used for self-supervised learning and other tasks.

Conclusion

Contributions

We considered the problem of learning under the constrain of limited labels.

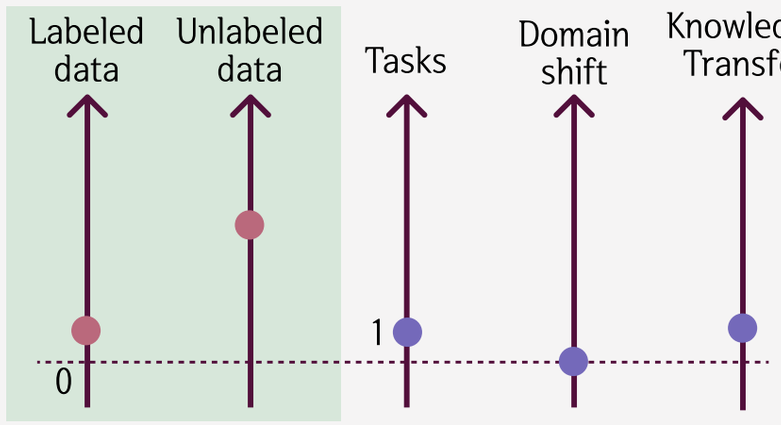


Conclusion

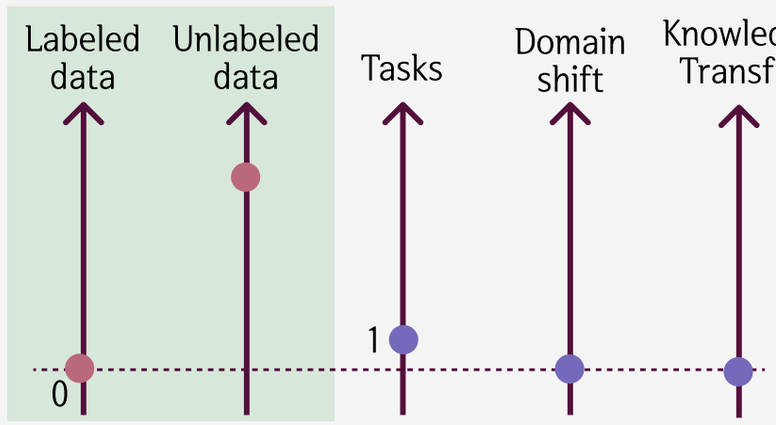
Contributions

We considered the problem of learning under the constrain of limited labels.

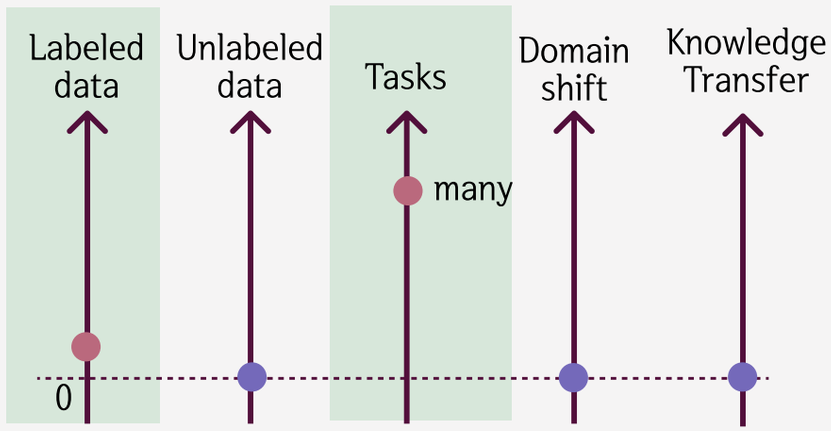
We focused on pre-existing paradigms that deal with a similar setting.



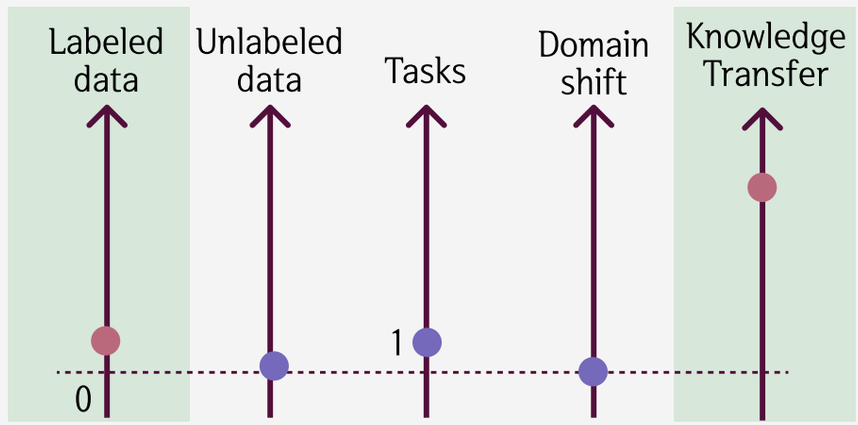
Semi-supervised Learning



Unsupervised Learning



Few-shot Learning



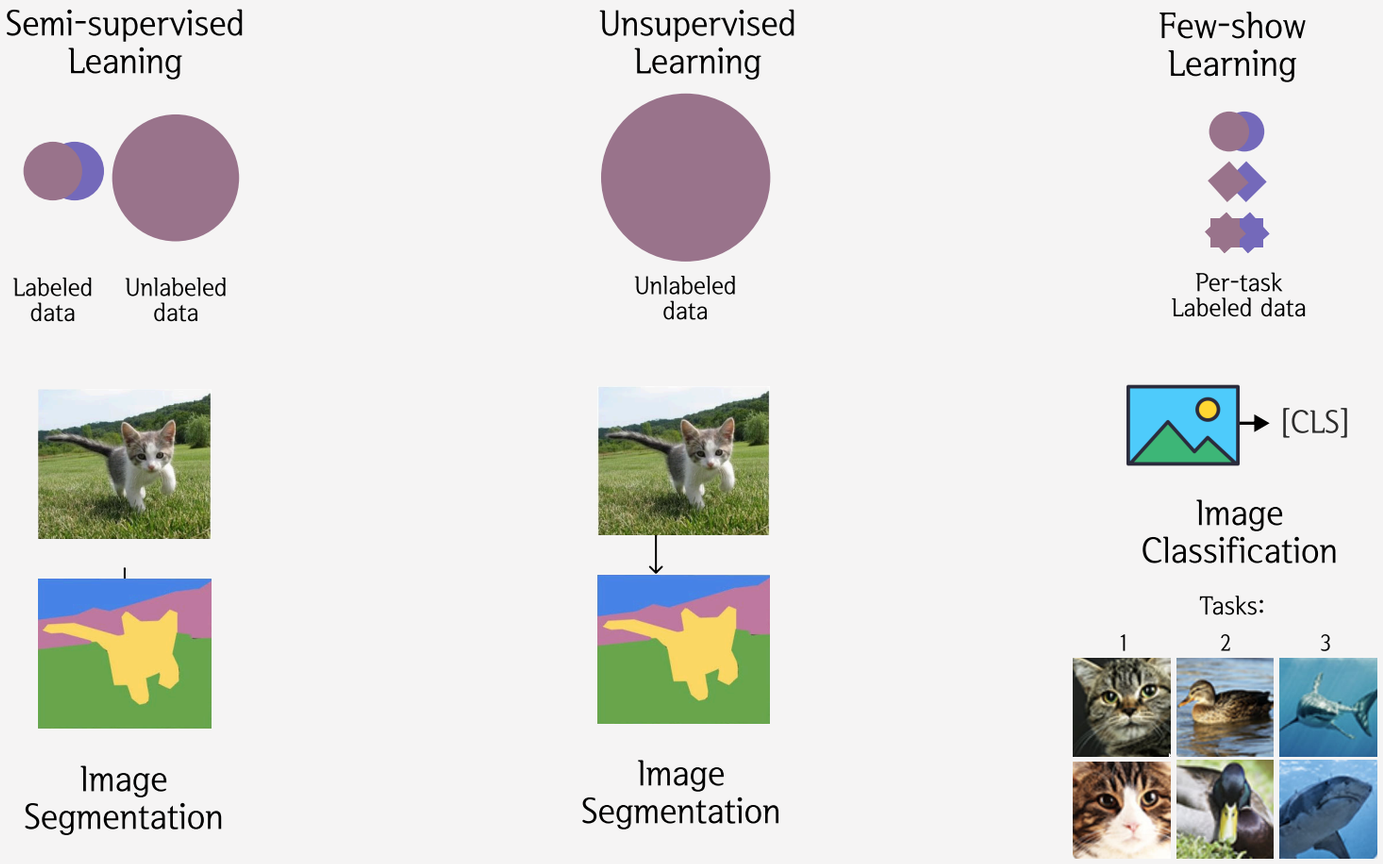
Few-shot Fine-tuning

Conclusion

Contributions

We presented:

- CCT a method for semi-supervised image segmentation.
- Autoregressive Segmentation for unsupervised image segmentation.
- Spatial Contrastive Learning (SCL) for few-shot image classification.



Conclusion

Publications

These contributions resulted in the following publications:

- Semi-supervised semantic segmentation with cross-consistency training, Ouali et al. CVPR 2020.
- Autoregressive unsupervised image segmentation, Ouali et al. ECCV 2020.
- Target Consistency for Domain Adaptation: when Robustness meets Transferability, Ouali et al. CaP 2021.
- Spatial contrastive learning for few-shot classification, Ouali et al. ECML-PKDD 2021.

And the following survey paper:

- An overview of deep semi-supervised learning, Ouali et al. arXiv.

Conclusion

Perspectives: drawbacks & possible extensions of the proposed methods

The proposed methods were:

- Uni-modal
- Uni-task
- Considered pre-existing label-efficient settings

Conclusion

Perspectives: drawbacks & possible extensions of the proposed methods

The proposed methods were:

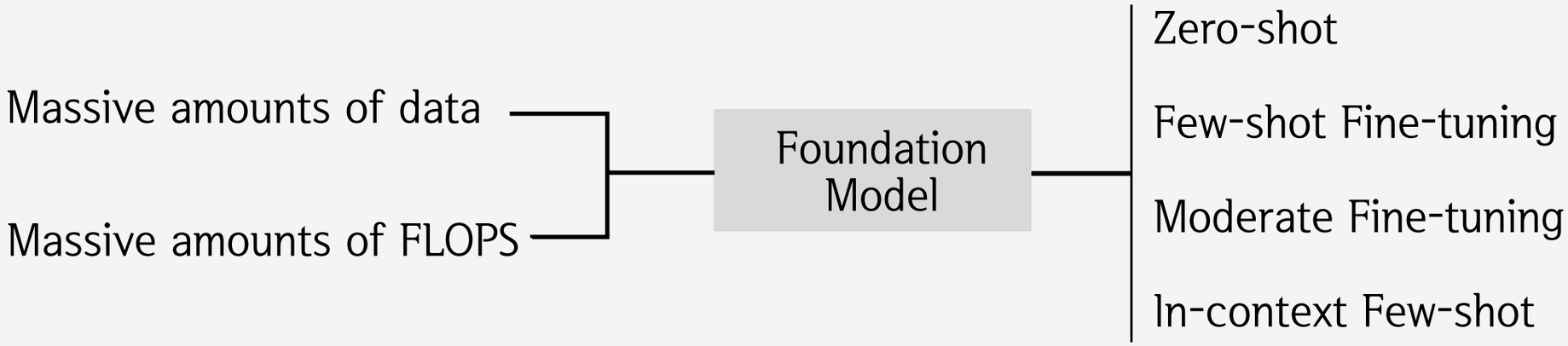
- Uni-modal
- Uni-task
- Considered pre-existing label-efficient settings

Possible extension:

- Train using multiple modalities at once.
- Solve multiple tasks at once.
- Proposes novel & flexible learning:
  - Show fast adaption (few-shot)
  - Use unlabeled data as the main source of supervision (unsupervised learning)
  - Leverage labeled data if available (optional semi-supersion)
  - Learn new tasks without forgetting (continual and incremental learning)
  - Request annotations (active learning)

Conclusion

Long term perspectives



Vision-language models: CLIP, ALIGN, BLIP.
Object detection: GLIP, Owl-ViT.
Large Language Models: LLaMa, GPT-x, PaML, Chinchilla.
Image Segmentation: Segment Everything.

GPT-3 training data[1]:9

Dataset	# tokens	Proportion within training
Common Crawl	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

LAION-5B: A NEW ERA OF OPEN LARGE-SCALE MULTI-MODAL DATASETS

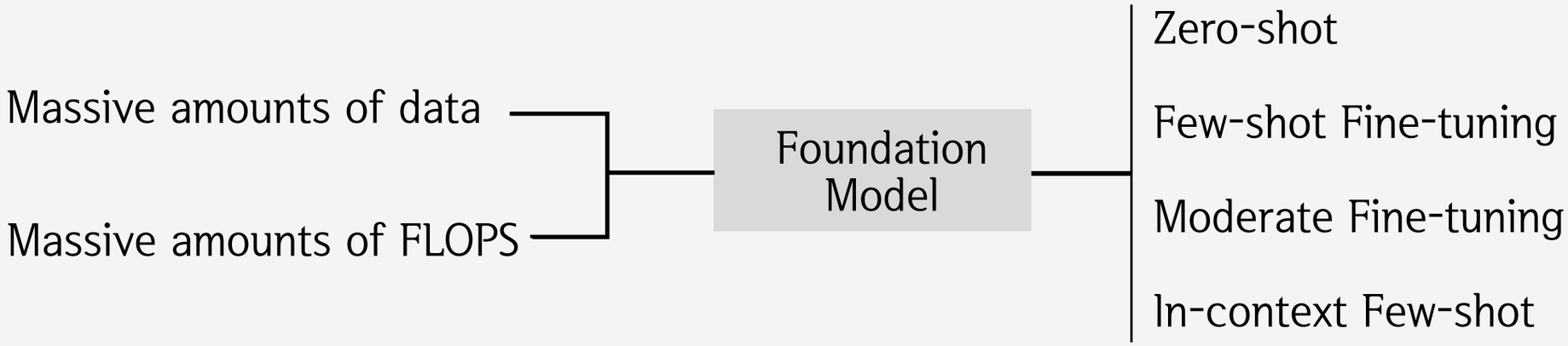
The Pile *An 800GB Dataset of Diverse Text for Language Modeling*

What is the Pile?

The Pile is a **825 GiB** diverse, open source language modelling data set that consists of 22 smaller, high-quality datasets combined together.

Conclusion

Long term perspectives



Instead of focusing on label Efficiency only.

We switch to:

- Data Efficiency:
  - Pre-training with reduced amount of data.
  - Fine-tuning with limited amount of labels.
- Computational & Parameter Efficiency:
  - Training/Fine-tuning with reduced FLOPs & parameters.
  - Fine-tuning/Adaption with reduced FLOPs & parameters.
  - Deployment with reduced FLOPs & parameters.

Ph.D. Thesis Defense

Learning with Limited Labels

Yassine, Céline, Myriam

Thank You!

References

Semi-supervised classification with graph convolutional networks. Kipf et al, 2016.

Semi-Supervised Learning with Generative Adversarial Networks. Odena et al, 2017.

Pseudo-label: The simple and efficient SSL method for deep neural network. Lee et al. 2013.

Mean teachers are better role models. Tarvainen et al, 2017.

Marr's Theory: From primal sketch to 3-D models.

FickleNet: Weakly and Semi-supervised Semantic Image Segmentation. Lee et al. 2019.

Semi supervised semantic segmentation using generative adversarial network. Suly et al. 2017.

Adversarial learning for semi-supervised semantic segmentation. Hung at al. 2018.

Weakly-supervised semantic segmentation network with deep seeded region growing. Huang1 et al, 2018.

On Mutual Information Maximization for Representation Learning. Tschannen et al, 2019.

Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.

Representation Learning with Contrastive Predictive Coding. Oord et al, 2019.

Learning visual groups from co-occurrences in space and time. Isola et al, 2016.

Deep clustering for unsupervised learning of visual features. Caron et al, 2018.

Invariant Information Clustering for Unsupervised Image Classification and Segmentation. Ji et al, 2020.

Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?. Tian et al. 2020.

A Closer Look at Few-shot Classification. Chen et al, 2019.

Parameter-Efficient Transfer Learning for NLP. Houlsby et al, 2019.

Prefix-Tuning: Optimizing Continuous Prompts for Generation. Li et al, 2021.

LoRA: Low-Rank Adaptation of Large Language Models. Hu et al. 2021.

BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. Zaken et al, 2021.

Appendix

Semi-supervised Leaning

Labeled data Unlabeled data

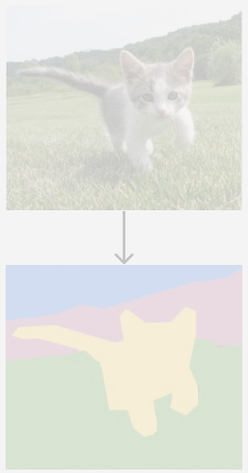


Image Segmentation

Unsupervised Learning

Unlabeled data

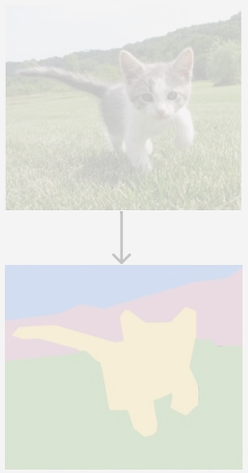


Image Segmentation

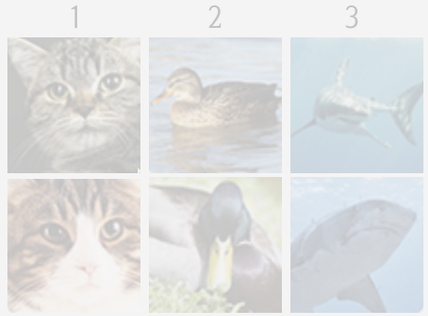
Few-show Learning

Per-task Labeled data

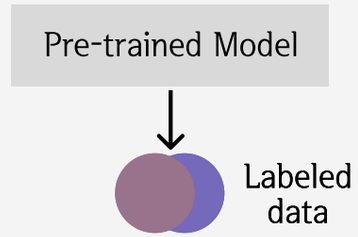
Image Classification

Tasks:

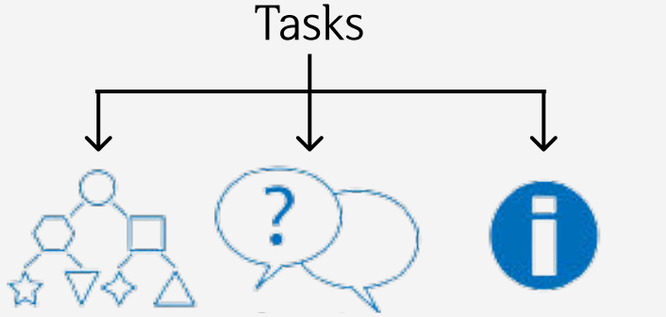
1 2 3



Few-Sample Fine-Tuning



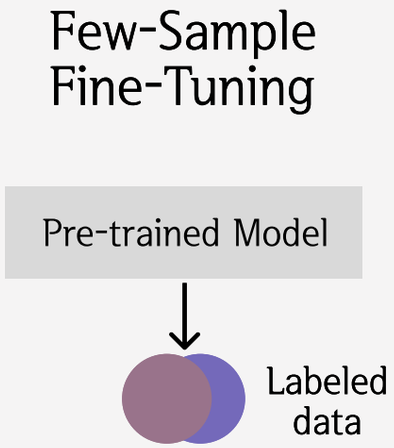
NLP Downstream Tasks



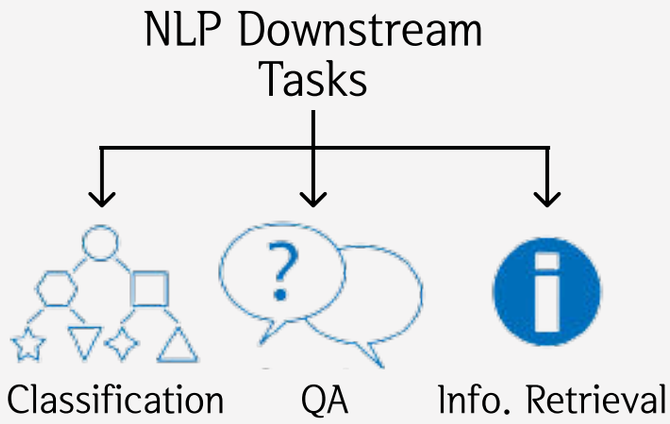
Contribution 4

Relative Bias Fine-tuning

Paradigm:
Few-sample Fine-tuning

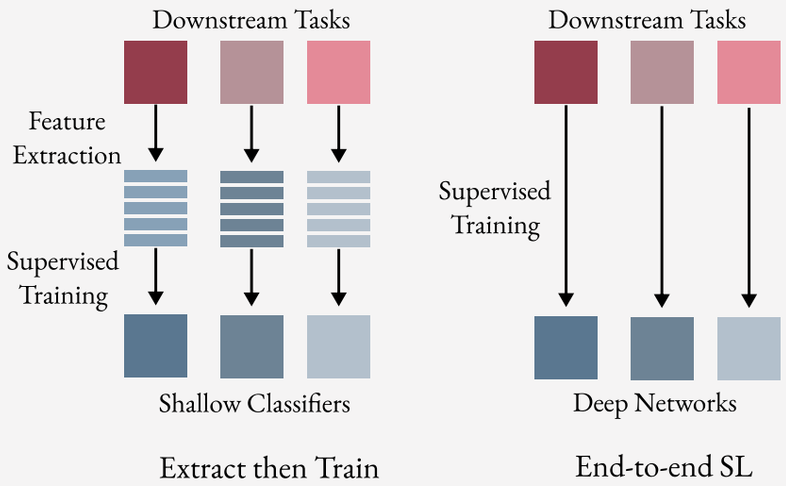


Task:
NLP classification type downstream tasks



Relative Bias Fine-tuning

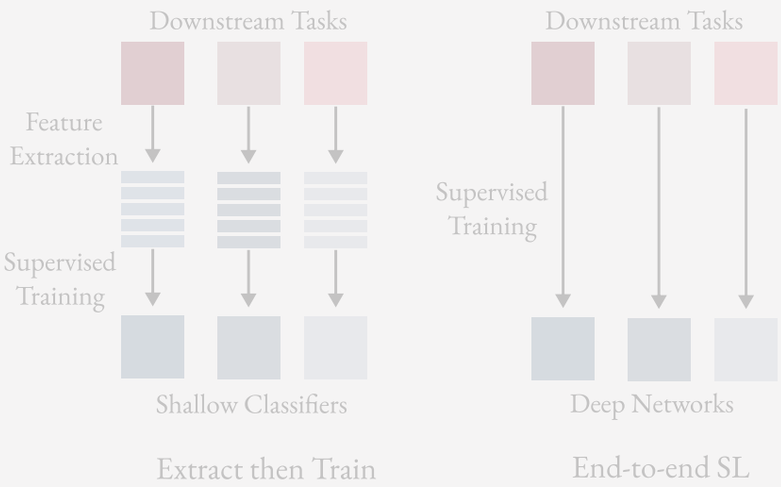
Introduction to the pre-train then fine-tune paradigm

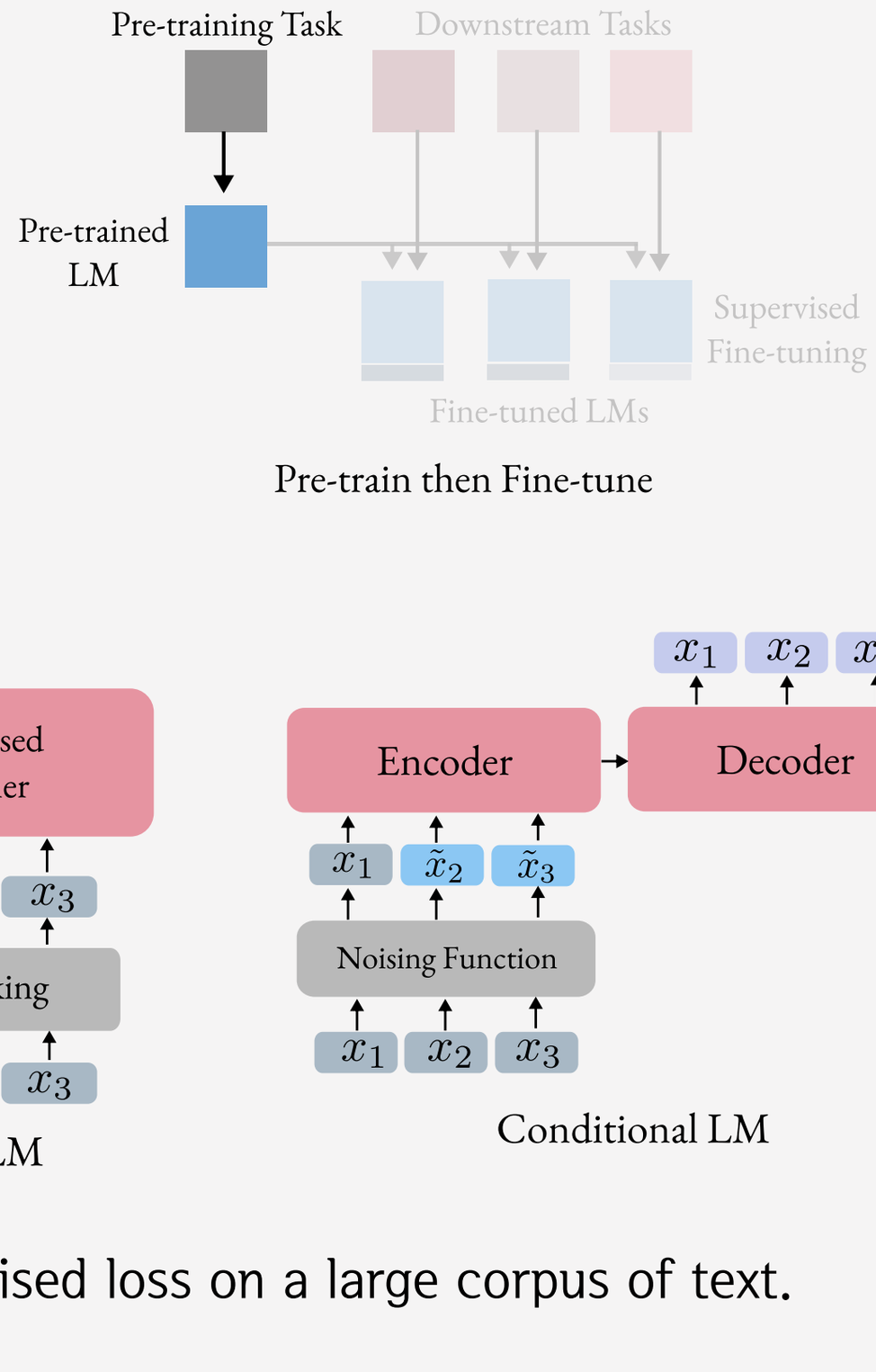


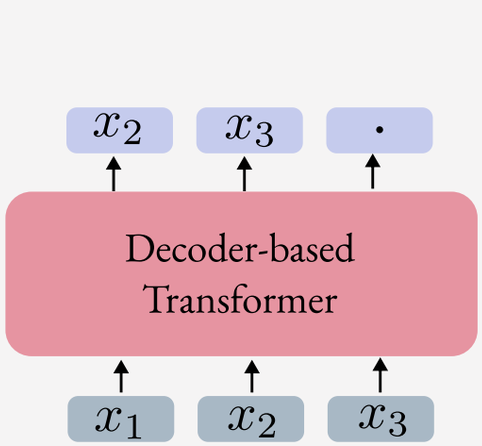
Instead of training a task specific model to solve each downstream task.

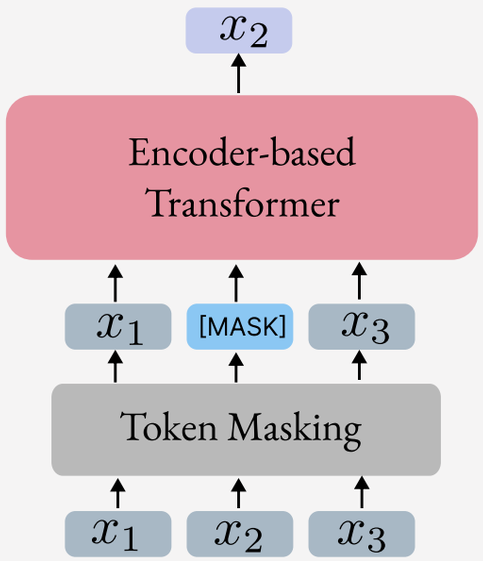
Relative Bias Fine-tuning

Introduction to the pre-train then fine-tune paradigm





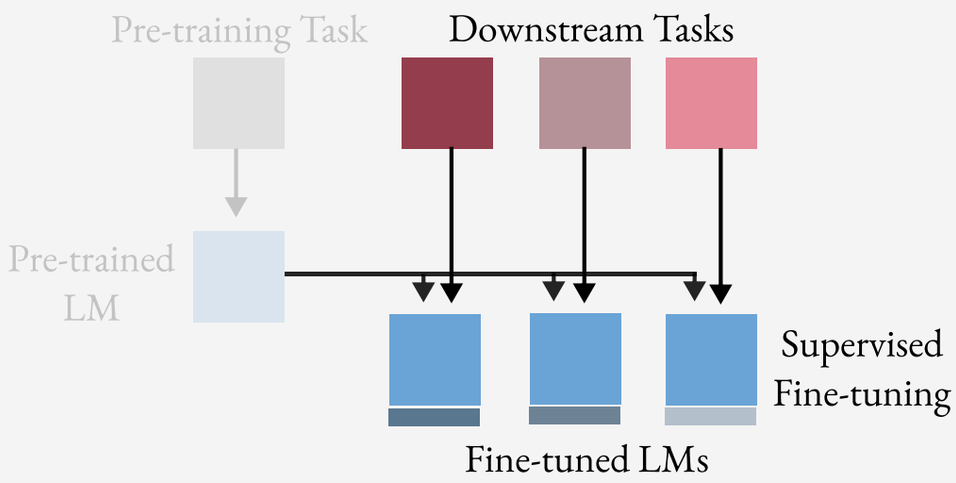




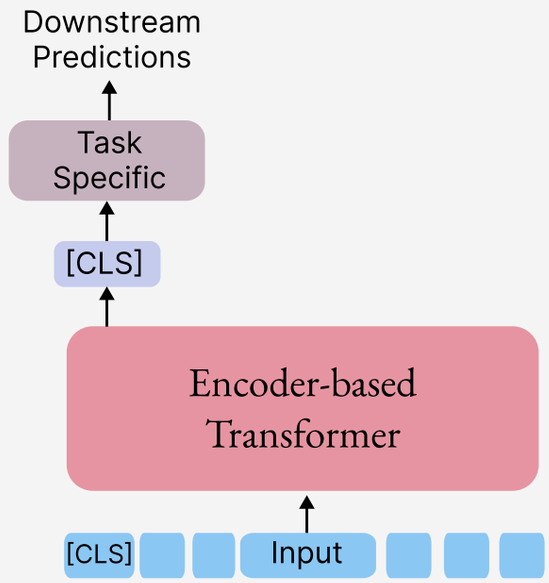
First, we pre-train a LM using a self-supervised loss on a large corpus of text.

Relative Bias Fine-tuning

Introduction to the pre-train then fine-tune paradigm



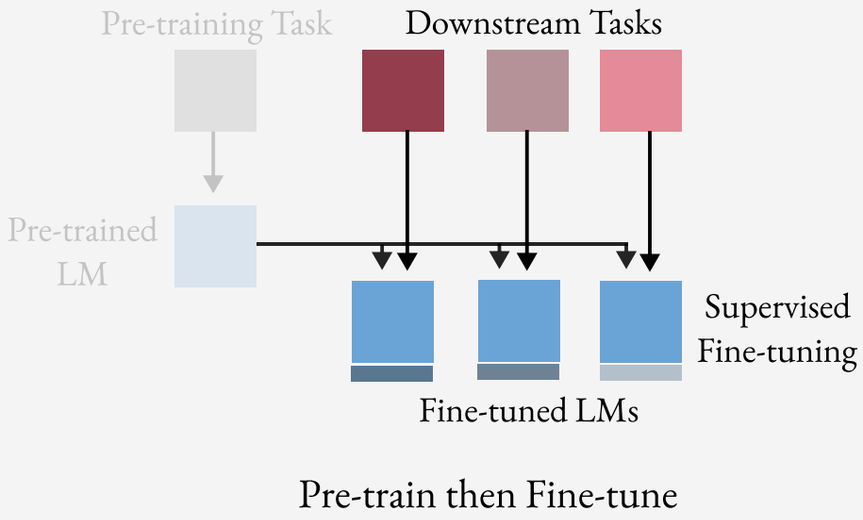
Pre-train then Fine-tune



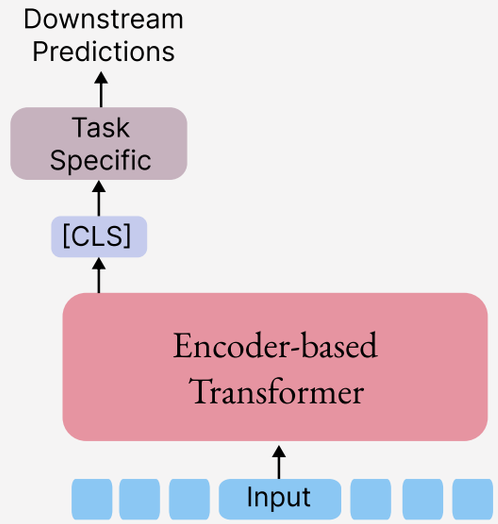
Then, for a given downstream task, we add a a task specific module on top of the pre-train LM for task adaptation.

Relative Bias Fine-tuning

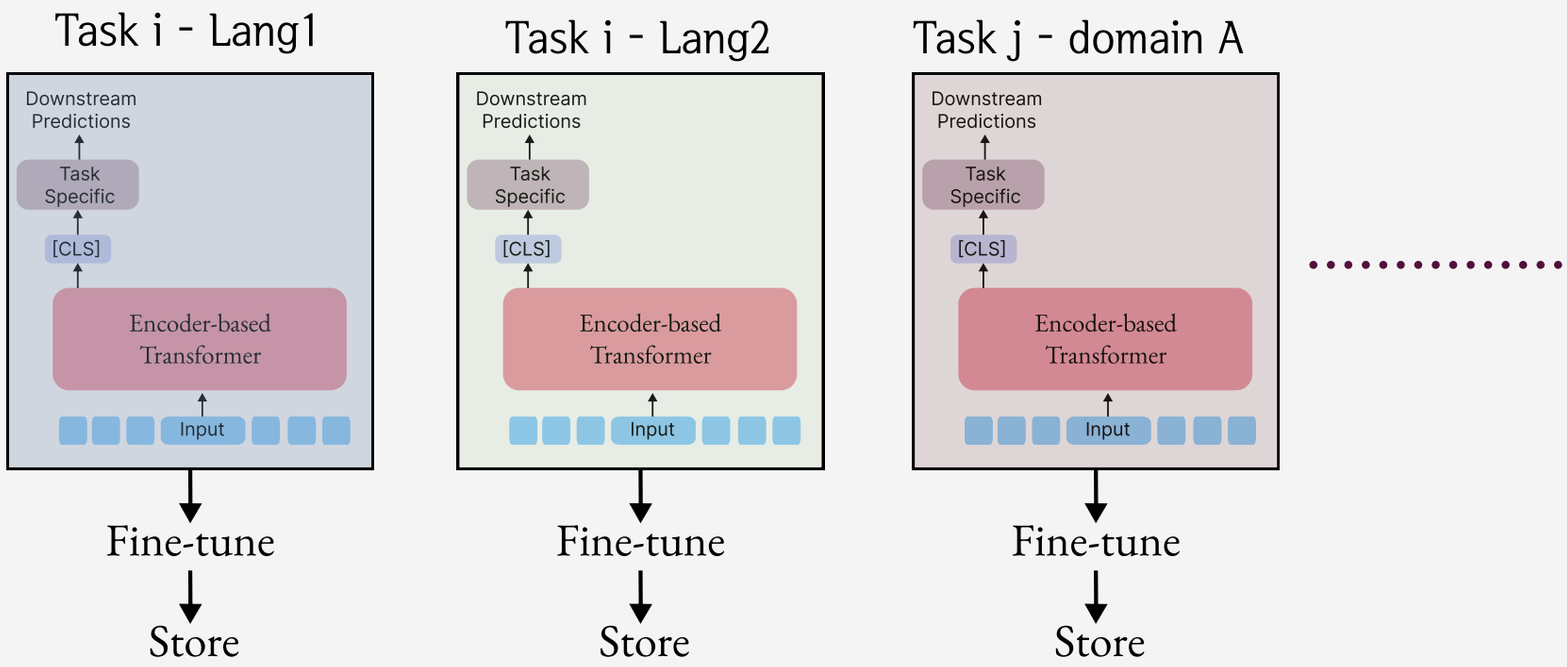
Introduction to the pre-train then fine-tune paradigm



Pre-train then Fine-tune



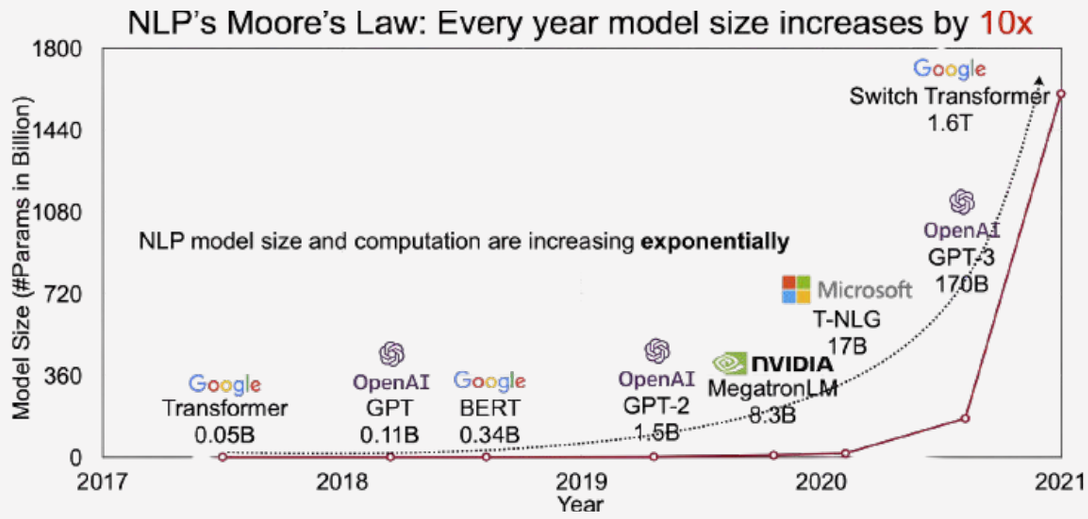
For N downstream tasks: N fine-tuning steps & N models



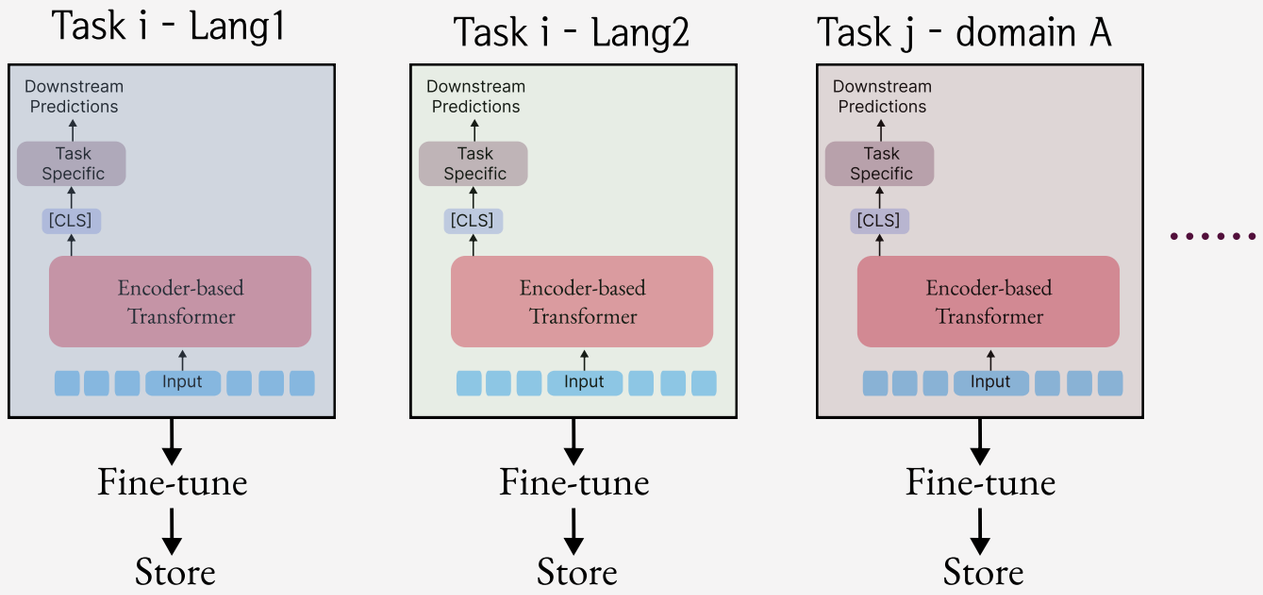
But this can be expensive, both for fine-tuning and for storing all of the task specific models.

Relative Bias Fine-tuning

Introduction to the pre-train then fine-tune paradigm



For N downstream tasks: N fine-tuning steps & N models



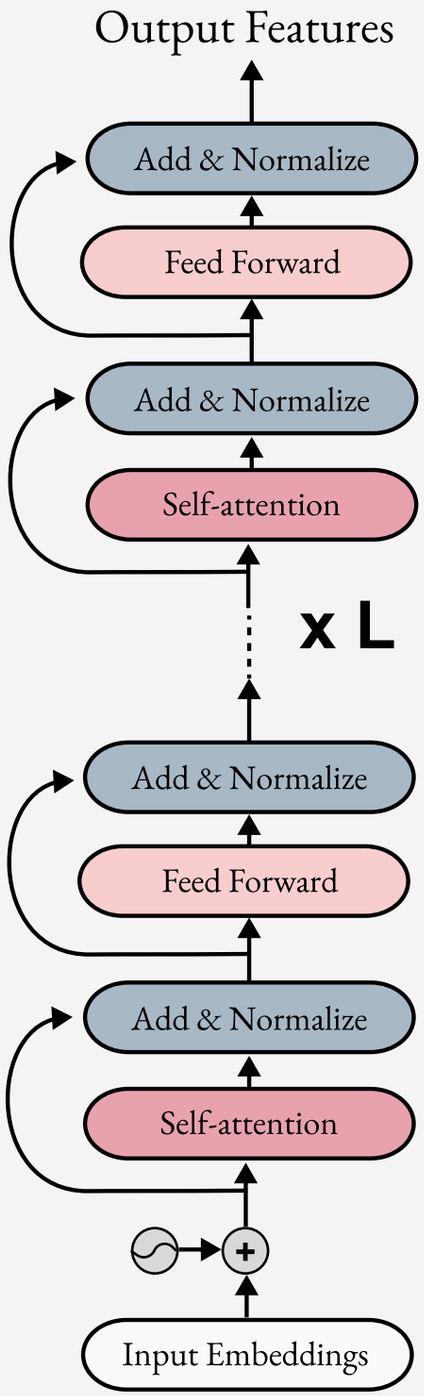
- Expensive Fine-tuning.
- Large amounts of storage requires.

And this can be be expensive, both for fine-tuning and for stroring all of the task specific models.

Image source: https://hanlab.mit.edu/projects/efficientnlp_old/

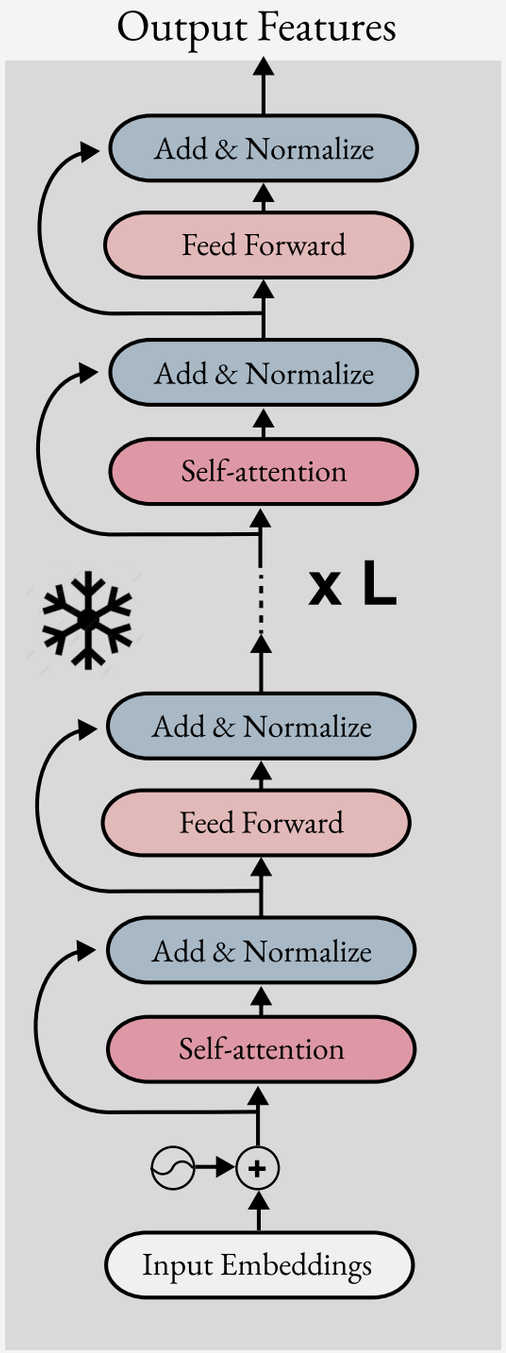
Relative Bias Fine-tuning

Introduction to Parameter Efficient Fine-tuning



Relative Bias Fine-tuning

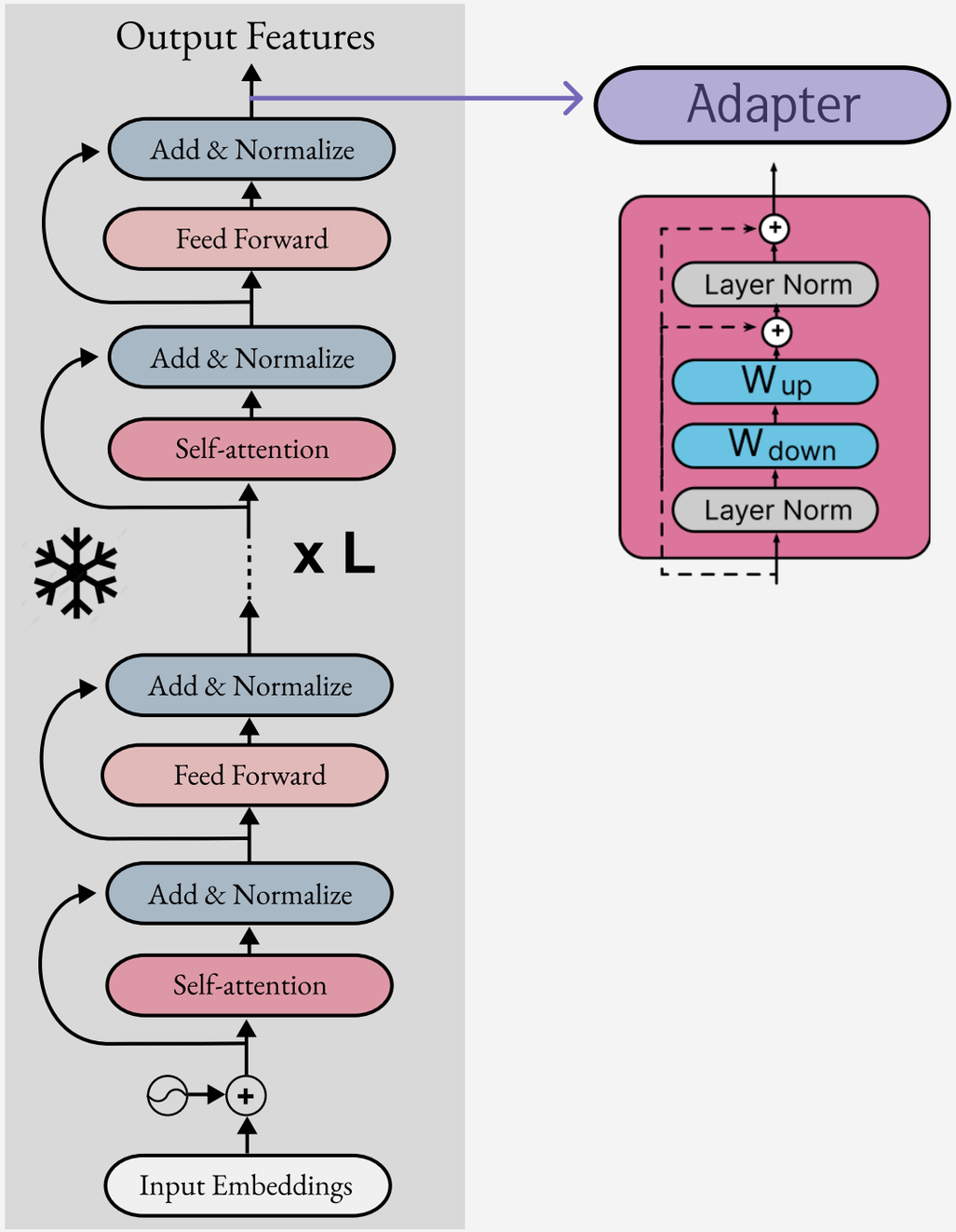
Introduction to Parameter Efficient Fine-tuning



In PEFT, we first freeze the original pre-trained model.

Relative Bias Fine-tuning

Introduction to Parameter Efficient Fine-tuning

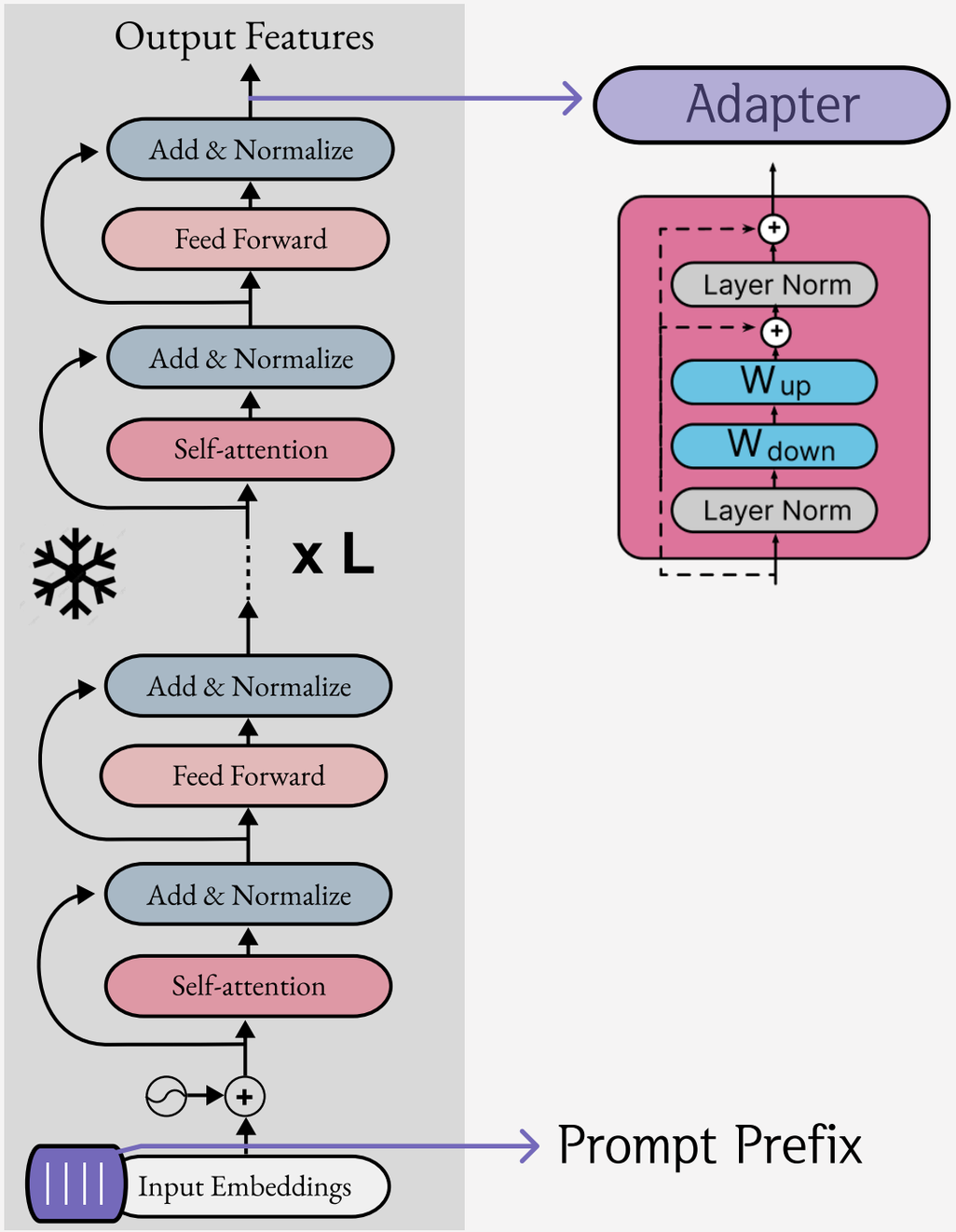


Then we add light & per-task modules.

Parameter-Efficient Transfer Learning for NLP. Houlsby et al, 2019.

Relative Bias Fine-tuning

Introduction to Parameter Efficient Fine-tuning

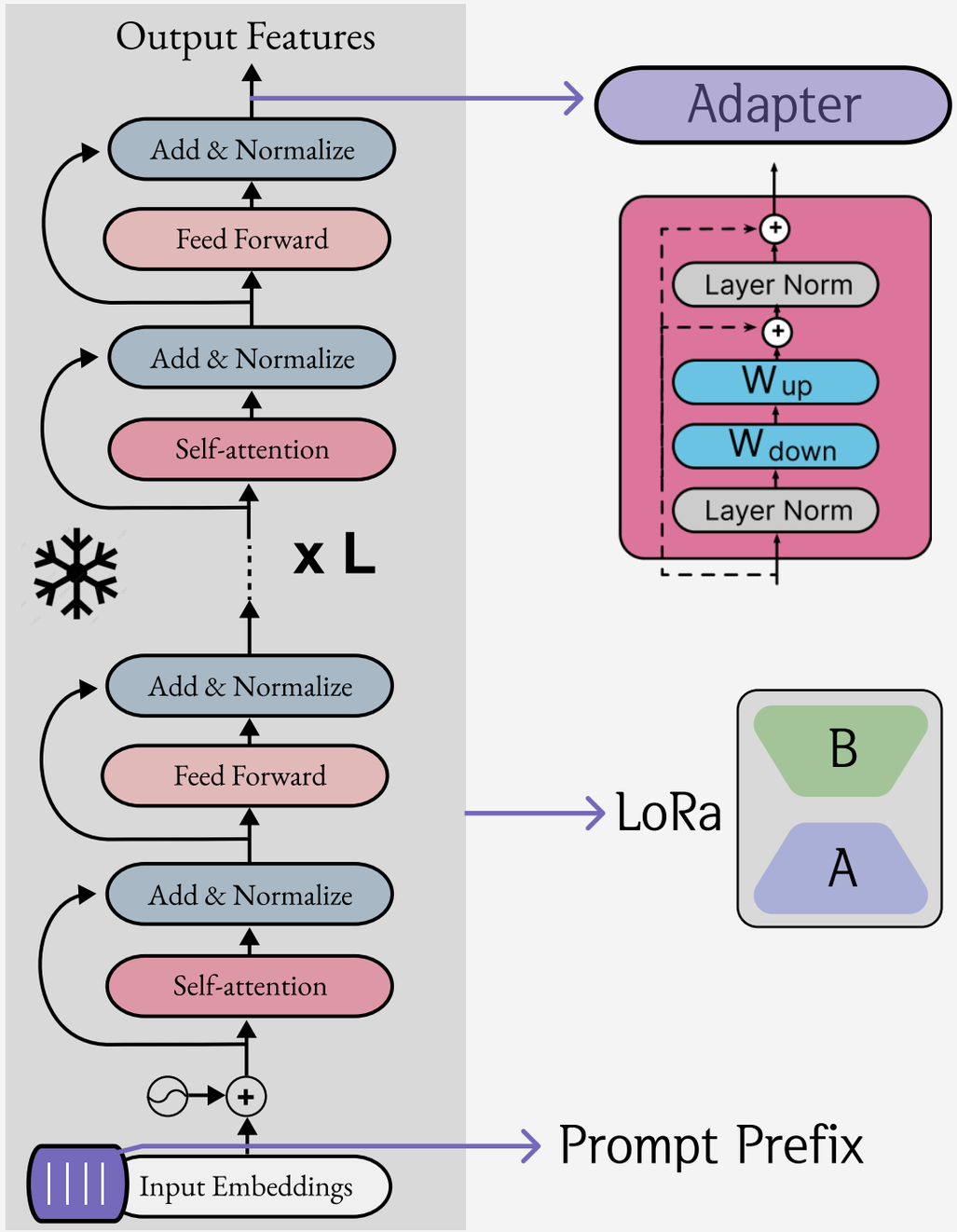


Then we add light & per-task modules.

Parameter-Efficient Transfer Learning for NLP. Houlsby et al, 2019.
Prefix-Tuning: Optimizing Continuous Prompts for Generation. Li et al, 2021.

Relative Bias Fine-tuning

Introduction to Parameter Efficient Fine-tuning

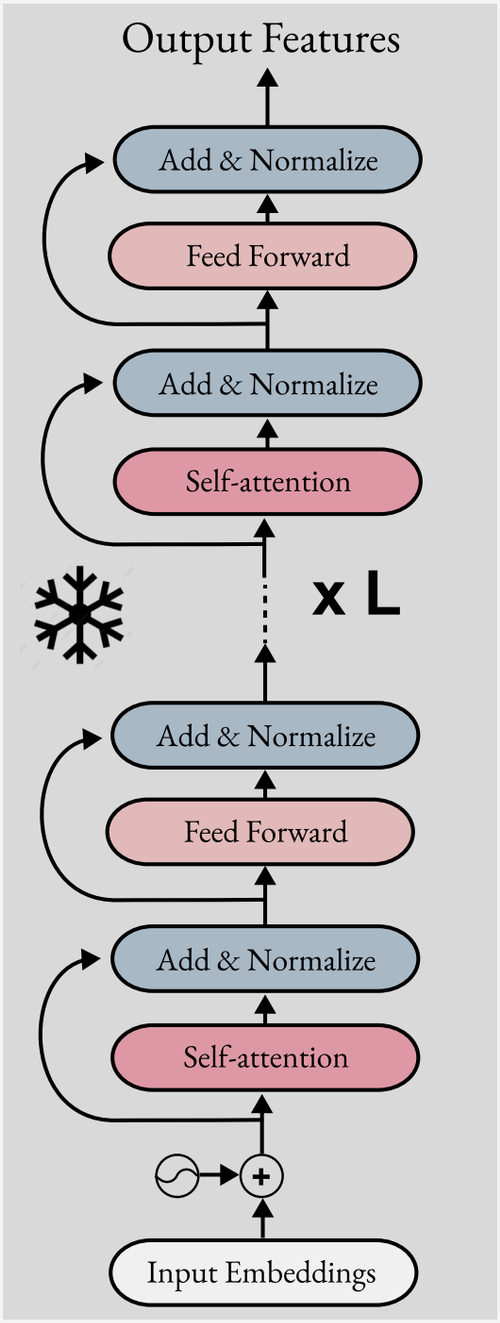


Then we add light & per-task modules.

Parameter-Efficient Transfer Learning for NLP. Houlsby et al, 2019.
Prefix-Tuning: Optimizing Continuous Prompts for Generation. Li et al, 2021.
LoRA: Low-Rank Adaptation of Large Language Models. Hu et al. 2021.

Relative Bias Fine-tuning

Starting for Bit-Fit: Fine-tune the biases only



Biases of the attention projection layers.

$$\mathbf{q} = \mathbf{z}\mathbf{W}_q + \mathbf{b}_q$$
$$\mathbf{k} = \mathbf{z}\mathbf{W}_k + \mathbf{b}_k$$
$$\mathbf{v} = \mathbf{z}\mathbf{W}_v + \mathbf{b}_v$$

Biases of the attention output layers.

$$\mathbf{h}_2 = \mathbf{h}_1\mathbf{W}_o + \mathbf{b}_o$$
$$\mathbf{h}_3 = \mathbf{g}_{\text{LN}_1} \odot \frac{(\mathbf{h}_2 + \mathbf{z}) - \mu}{\sigma} + \mathbf{b}_{\text{LN}_1}$$

Biases of the feed-forward layers.

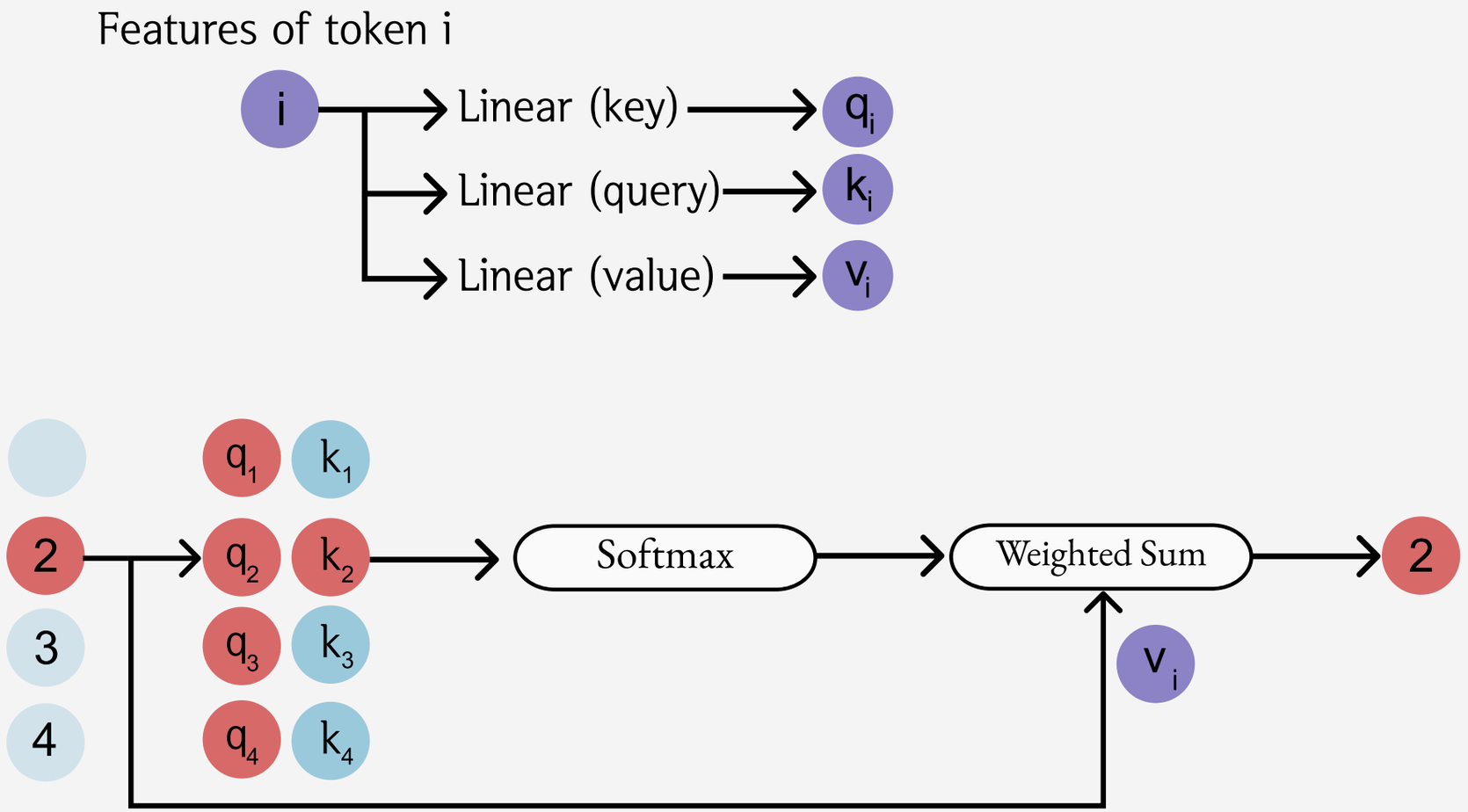
$$\mathbf{h}_4 = \text{GELU}(\mathbf{h}_3\mathbf{W}_{\text{FF}_1} + \mathbf{b}_{\text{FF}_1})$$
$$\mathbf{h}_5 = \mathbf{h}_4\mathbf{W}_{\text{FF}_2} + \mathbf{b}_{\text{FF}_2}$$
$$\mathbf{z}' = \mathbf{g}_{\text{LN}_2} \odot \frac{(\mathbf{h}_5 + \mathbf{h}_3) - \mu}{\sigma} + \mathbf{b}_{\text{LN}_2}$$

BitFit: only fine-tune the biases of all the model (<1% of the number of parameters).

BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. Zaken et al, 2021.

Relative Bias Fine-tuning

Bit-Fit analysis: the key bias term



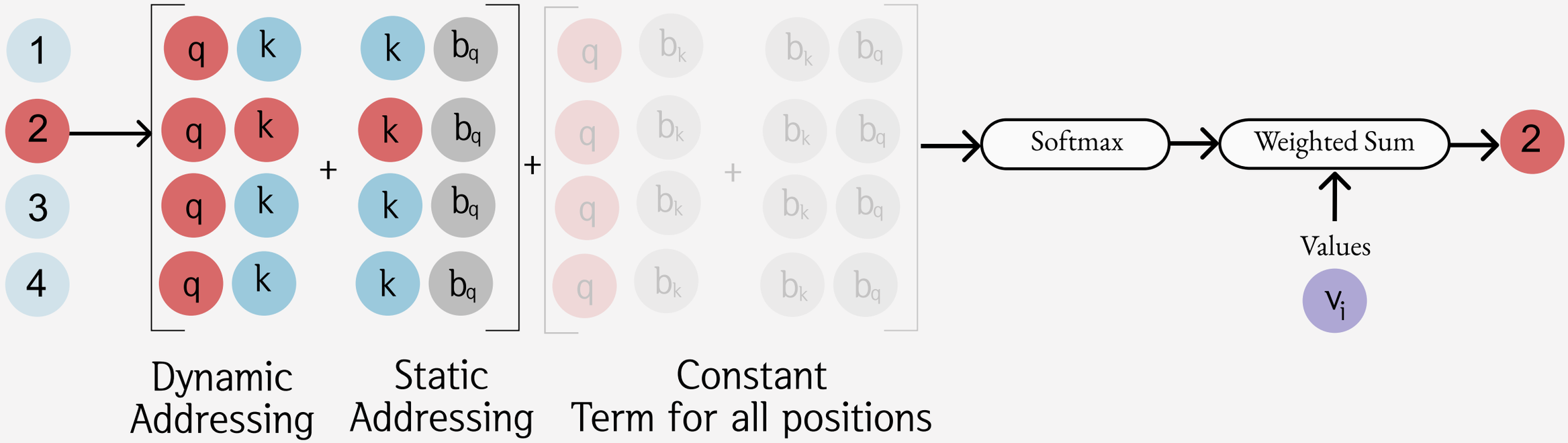
Attention computation without bias term in the projection layers.

Relative Bias Fine-tuning

Bit-Fit analysis: the key bias term

Attention computation without bias term in the projection layers.

$$\begin{aligned}\mathbf{q}\mathbf{k}^\top &= (\mathbf{z}\mathbf{W}_q + \mathbf{b}_q)(\mathbf{z}\mathbf{W}_k + \mathbf{b}_k)^\top \\ &= \underbrace{\mathbf{z}\mathbf{W}_q\mathbf{W}_k^\top\mathbf{z}^\top}_{1} + \underbrace{\mathbf{b}_q\mathbf{W}_k^\top\mathbf{z}^\top}_{2} + \underbrace{\mathbf{z}\mathbf{W}_q\mathbf{b}_k^\top}_{3} + \underbrace{\mathbf{b}_q\mathbf{b}_k^\top}_{4}\end{aligned}$$



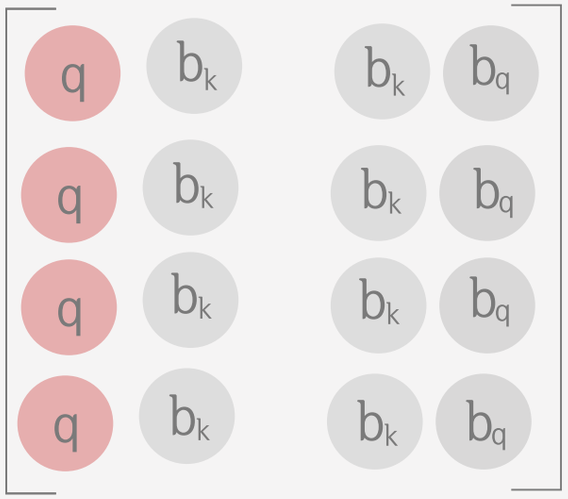
$$\text{softmax}(\mathbf{a} + \text{ const }) = \text{softmax}(\mathbf{a}), \forall \text{ const}$$

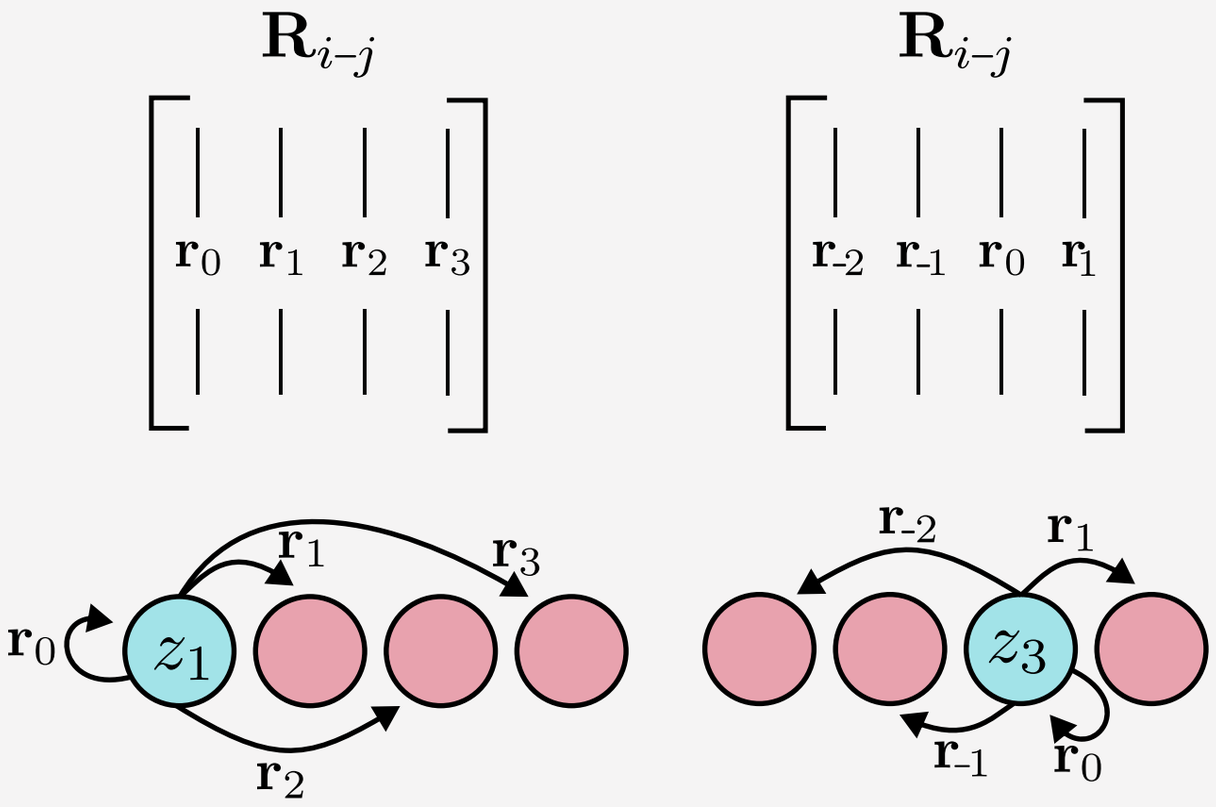
-> The key bias term is not used/learned.

Relative Bias Fine-tuning

RelBitFit: Replacing the key bias terms with relative biases

$$\mathbf{q}\mathbf{k}^\top = \mathbf{z}\mathbf{W}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{b}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{z}\mathbf{W}_q\mathbf{R}_{i-j} + \mathbf{s}_{i-j}$$





Replace the key bias related terms with relative biases.

Relative Bias Fine-tuning

Reducing the parameter count of RelBitFit

Method	Num. of trainable params.	% of the model's total params.
Full Fine-tuning	$\approx 124 \times 10^6$	100%
BitFit	$\approx 0.7 \times 10^6$	0.55%
RelBitFit	$\approx 1.8 \times 10^6$	1.5% + 157%

Relative Bias Fine-tuning

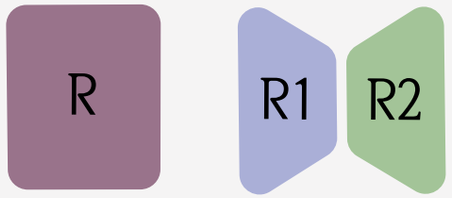
Reducing the parameter count of RelBitFit

Attention Operation:

- Parameter Tying (PT): learn single joint parameters R and S for the whole model.

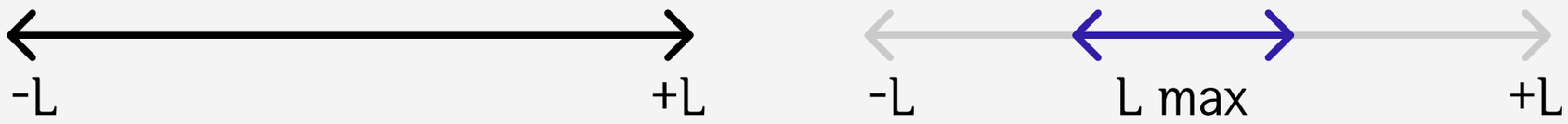
RelBitFit with PT	$\approx 0.78 \times 10^6$	0.63%

- Low Rank (LR): Decompose R into two low rank matrices.



RelBitFit with LR ($d' = 64$)	$\approx 0.76 \times 10^6$	0.8%

- Clipping the Relative Distances (Clip.): Consider a smaller window of relative distances.



RelBitFit with Clip. ($L_{\max} = 32$)	$\approx 0.98 \times 10^6$	0.8%

Relative Bias Fine-tuning

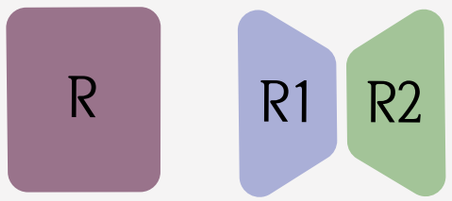
Reducing the parameter count of RelBitFit

Attention Operation:

- Parameter Tying (PT): learn single joint parameters R and S for the whole model.

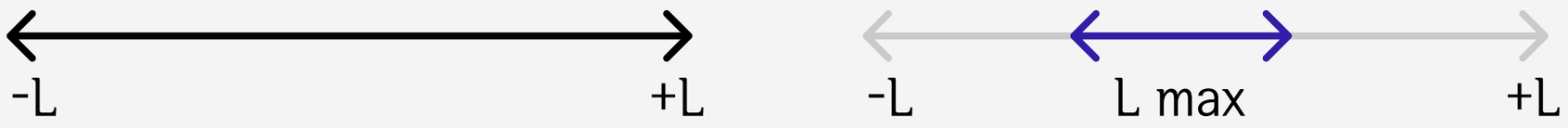
RelBitFit with PT	$\approx 0.78 \times 10^6$	0.63%

- Low Rank (LR): Decompose R into two low rank matrices.



RelBitFit with LR ($d' = 64$)	$\approx 0.76 \times 10^6$	0.8%

- Clipping the Relative Distances (Clip.): Consider a smaller window of relative distances.



RelBitFit with Clip. ($L_{\max} = 32$)	$\approx 0.98 \times 10^6$	0.8%

Method	Num. of trainable params.	% of the model's total params.
Full Fine-tuning	$\approx 124 \times 10^6$	100%
BitFit	$\approx 0.7 \times 10^6$	0.55%
RelBitFit with PT+LR+Clip.($L_{\max} = 32$)	$\approx 0.73 \times 10^6$	0.59%

Relative Bias Fine-tuning

Results: GLUE benchmark

Dataset	Task	Domain	#Train	#Classes
RTE	Textual Entailment	News/Wikipedia	2.5k	2
MRPC	Paraphrase	News	3.7k	2
STS-B	Textual Similarity	News/Others	7k	Reg.
CoLA	Grammatical Correctness	Linguistic Publications	8.5k	2
SST-2	Sentiment Analysis	Movie Reviews	67k	2
QNLI	Question Answering/Textual Entailment	Wikipedia	105k	2
QQP	Question Answering/Semantic Equivalence	Quora	364k	2
MNLI	Textual Entailment	Multi-domain	393k	3

Dataset	Metric
QNLI	Accuracy
SST-2	Accuracy
MNLI	Matched Accuracy and Mismatched Accuracy
CoLA	Matthews Correlation
MRPC	F1
STS-B	Spearman Correlation
RTE	Accuracy
QQP	F1

Relative Bias Fine-tuning

Results: GLUE benchmark

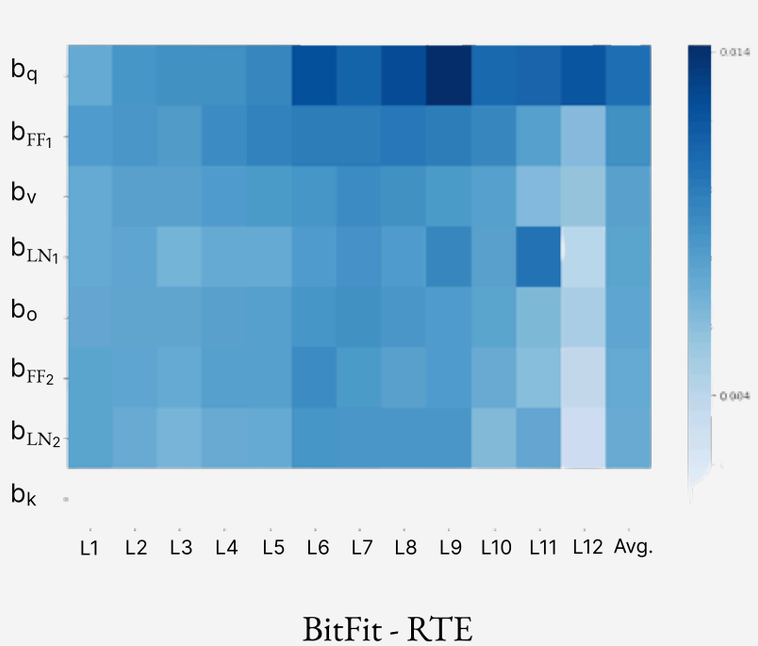
Method	QNLI	SST-2	MNLI_m	MNLI_mm	CoLA	MRPC	STS-B	RTE	QQP	Avg.
Size of training set: $N = 2000$										
Full Fine-tuning	50.5	58.9	35.4	35.2	N/A	81.2	58.6	52.7	63.2	54.5
BitFit	83.5	91.6	74.0	75.4	51.1	91.1	88.9	71.8	80.4	78.6
RelBitFit	**84.8**	**91.7**	**75.1**	**76.4**	**55.7**	**92.1**	**89.4**	**74.4**	**80.6**	**80.0**
Size of training set: $N = 1000$										
Full Fine-tuning	50.5	50.9	35.4	35.2	N/A	81.2	83.0	52.7	66.9	57.0
BitFit	81.5	91.3	70.4	71.7	49.3	88.2	86.5	**66.8**	78.4	76.0
RelBitFit	**82.7**	**91.5**	**71.6**	**72.1**	**51.3**	**90.5**	**87.7**	**66.8**	**79.2**	**77.0**

Method	QNLI	SST-2	MNLI_m	MNLI_mm	Avg.
Size of training set: $N = 500$					
Full Fine-tuning	52.1	72.6	35.4	35.2	48.8
BitFit	78.4	89.7	63.9	64.5	74.1
RelBitFit	**80.7**	**90.8**	**64.1**	**64.6**	**75.0**
Size of training set: $N = 100$					
Full Fine-tuning	50.5	79.6	35.4	35.2	50.2
BitFit	56.4	84.3	35.3	35.2	52.8
RelBitFit	**64.3**	**84.4**	**36.4**	**36.2**	**55.3**

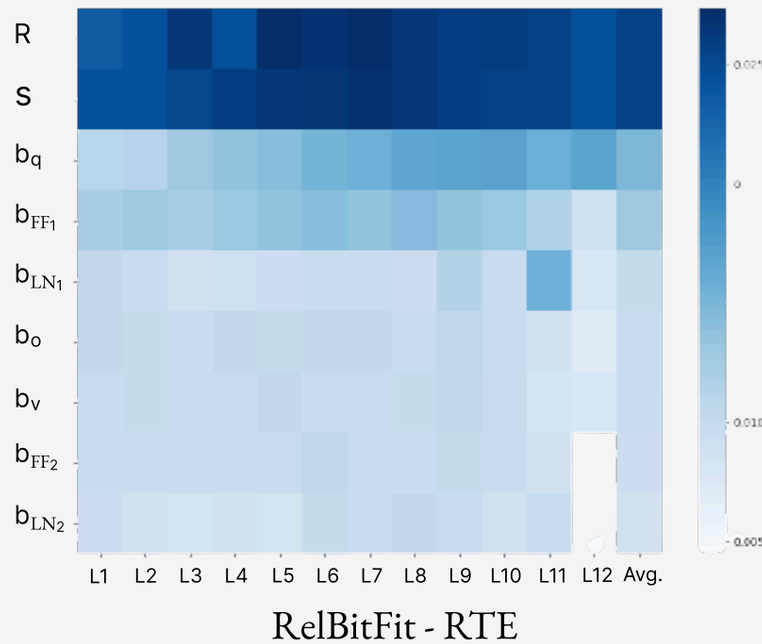
While fine-tuning the whole model results in overfitting in low label settings, RelBitFit performs better than BitFit on all of the GLUE tasks.

Relative Bias Fine-tuning

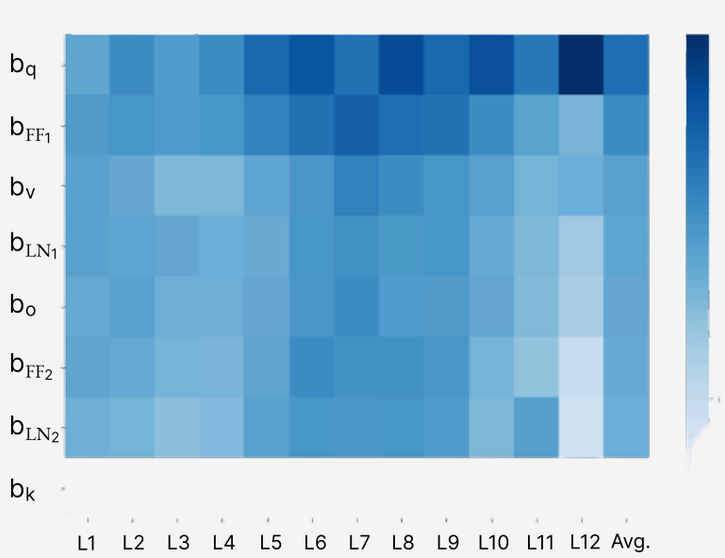
Results: Change in the biases values



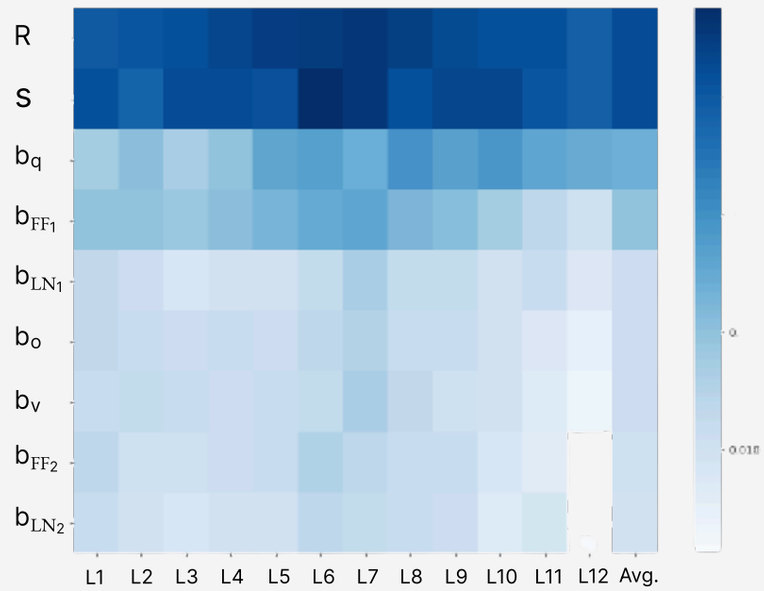
BitFit - RTE



RelBitFit - RTE



BitFit - MRPC



RelBitFit - MRPC

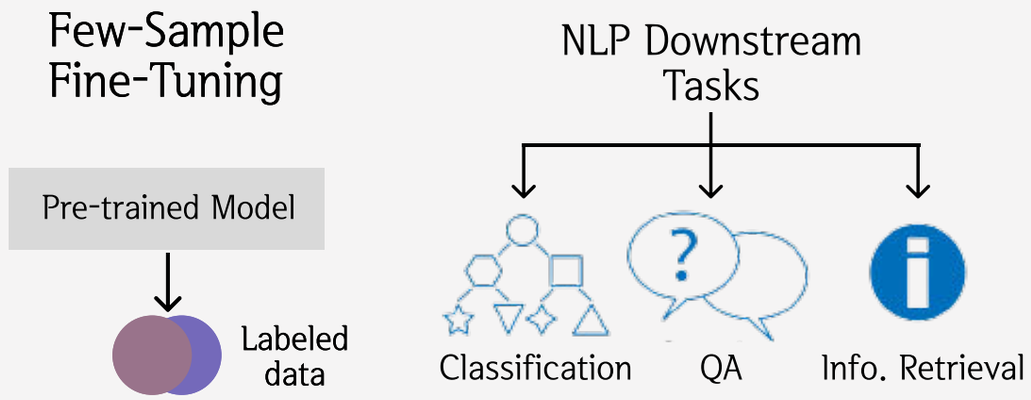
By showing the bias term changes before and after fine-tuning, we see that the relative biases are learned, and play a role in the fine-tuned model.

Contribution 4: Relative Bias Fine-tuning

Conclusion

We considered:

- Few-sample Fine-tuning
- NLP downstream tasks



We introduced a variant of BitFit for parameters efficient fine-tuning.

Contribution 4: Relative Bias Fine-tuning

Conclusion

Limitation:

- Limited to bias fine-tuning, and for some down-stream tasks, we might require more capacity.
- Limited applicability: the presented results were limited to GLUE benchmark, of which the majority are classification tasks. Its effectiveness for other tasks like generation is unknown.

Contribution 4: Relative Bias Fine-tuning

Conclusion

Limitation:

- Limited to bias fine-tuning, and for some down-stream tasks, we might require more capacity.
- Limited applicability: the presented results were limited to GLUE benchmark, of which the majority are classification tasks. Its effectiveness for other tasks like generation is unknown.