Learning Structured Visual Concepts with Few-shot Supervision

Xuming He 何旭明

ShanghaiTech University

hexm@shanghaitech.edu.cn

Outline

- Introduction
  - Learning from very limited annotated data

- Background in few-shot learning
  - Few-shot classification
  - Meta-learning framework

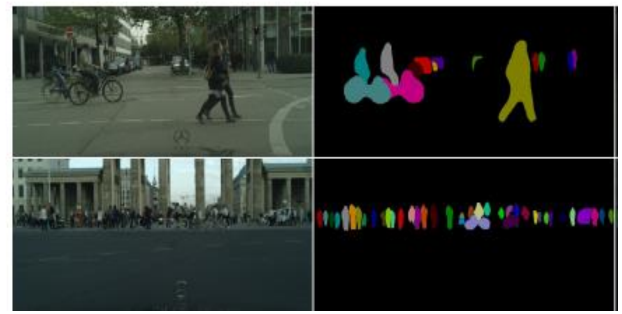
- Towards few-shot representation learning in vision tasks
  - Spatio-temporal patterns in videos [CVPR 2018]
  - Visual object & task representation [AAAI 2019]

- Summary and future directions

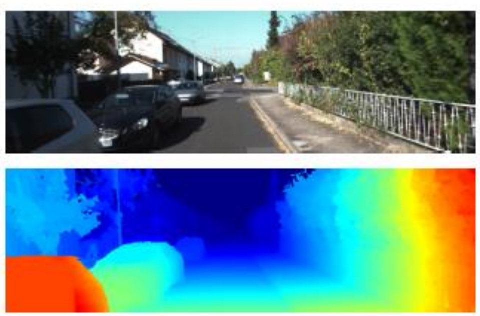
Introduction

- Data-driven visual scene understanding
  - Deep Neural Networks require large amount of annotated data

Semantic segmentation



Instance segmentation&detection



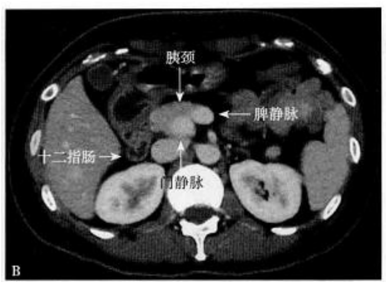
Depth estimation

…

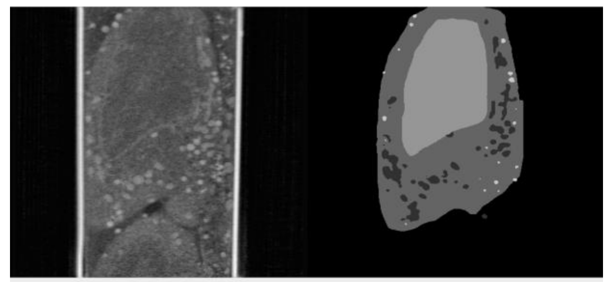
Image-level description

Real-world scenarios

- Data annotation is costly
  - Many specific domain and cross modality tasks



Medical image understanding
(image credit: 廖飞. 胰腺影像学. 2015.)

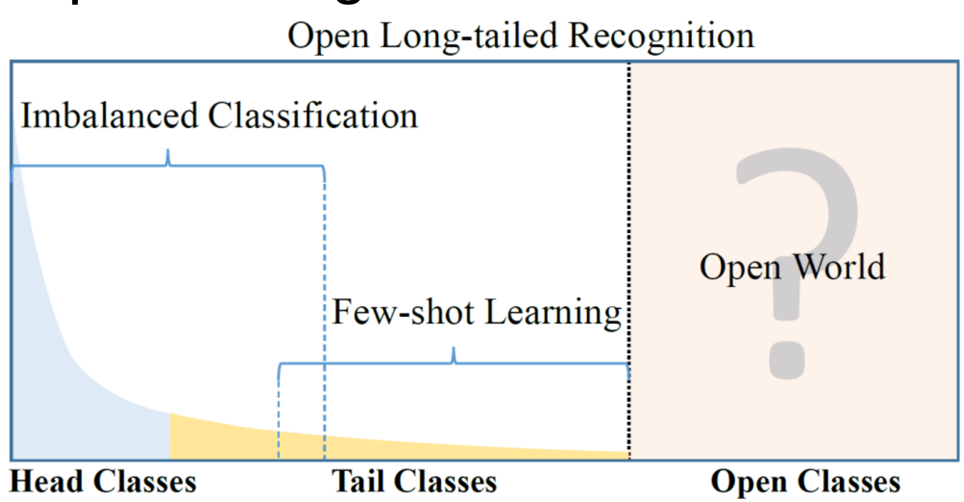


Biological image analysis
(Zhang and He, 2019)

A house cat laying on a couch beside a remote.

Vision & Language (MSCOCO)

- Visual concept learning in wild



(Liu et al CVPR 2019)

Challenges

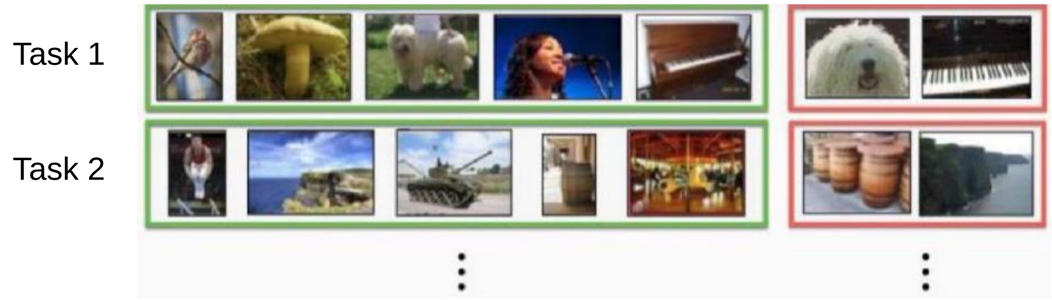
- Limitation in naïve transfer learning
  - Insufficient instance variations of novel classes
  - Fine-tuning usually fails given a few examples per class

Image Credit: Ravi & Larochelle et al 2017

- Human (child) performance is much better
  - How do we achieve such data efficiency?
  - What representations are used?
  - What are the underlying learning algorithms?

Main intuitions in few-shot learning

- Prior knowledge in different vision tasks
  - Similarity between visual categories
    - Feature representations, etc.
  - Similarity between visual recognition tasks
    - Learning a classifier, etc.



- Focusing on generic aspects of similar tasks
  - Generic visual representations
    - Not category-specific
  - Transferrable learning strategies
    - Very data-efficient

Outline

- Introduction
  - Learning from very limited annotated data
- Background in few-shot learning
  - Few-shot classification
  - Meta-learning framework
- Towards few-shot representation learning in vision tasks
  - Spatio-temporal patterns in videos [CVPR 2018]
  - Visual object & task representation [AAAI 2019]
- Summary and future directions

Few-shot learning problem

- Learning from (very) limited annotated data
- Typical setting:
  - Classification using a few training examples per visual category
  - Formally, given a small dataset $D_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$
    - N categories $y_i \in \mathcal{Y}, |\mathcal{Y}| = N,$
    - K shot: each class has K examples, or $L = N \times K$
  - The goal is to learn a model *F* parametrized by θ to minimize

$$E_{D_{test}}\left[\mathrm{loss}(y_i, F_\theta(x_i))\right]$$

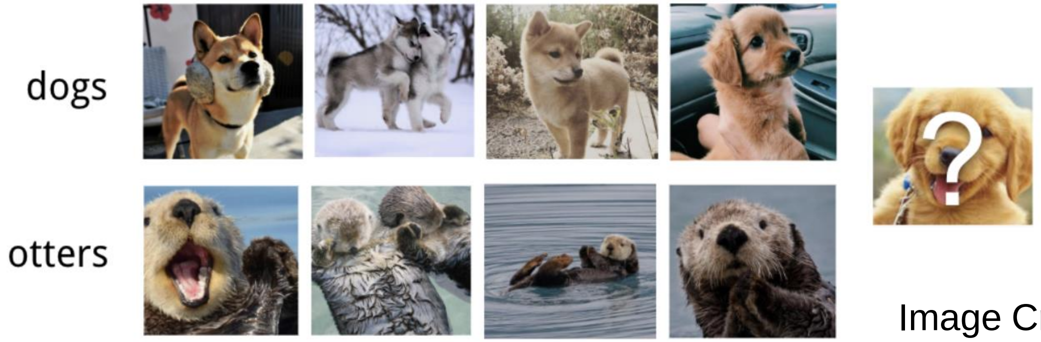


Image Credit: Weng, Lil-log, 2018

Few-shot learning problem

- For a single isolated task, this is difficult
  - But *if* we have access to many *similar* few-shot learning tasks, we can exploit such prior knowledge.
- Main idea is to consider task-level learning
  - Learn a representation shared by all those tasks
  - Learn an efficient classifier learning algorithm that can be applied to all the tasks

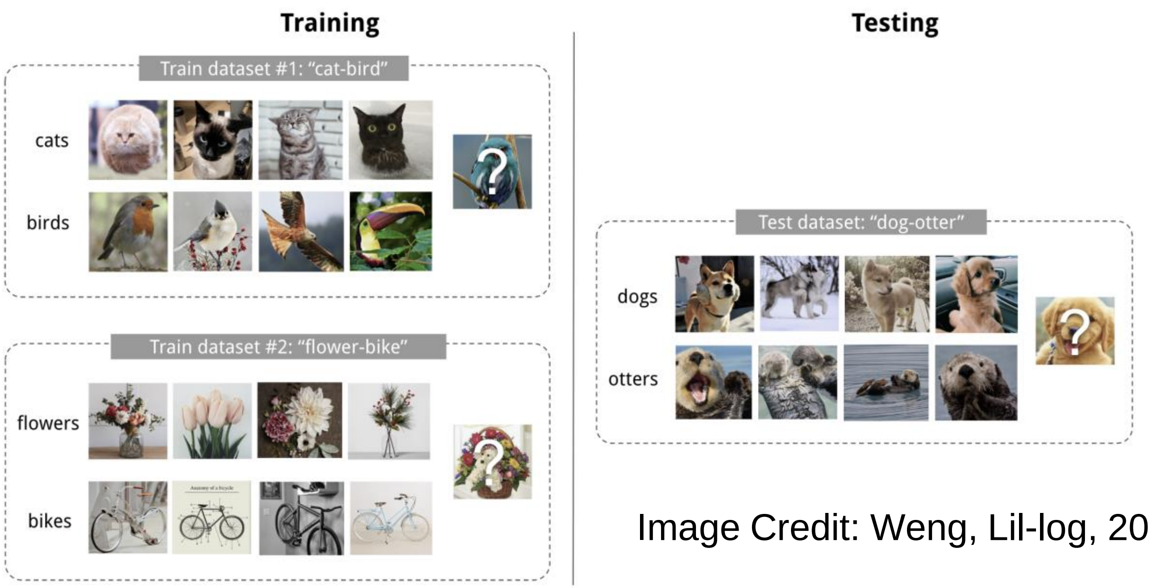


Image Credit: Weng, Lil-log, 2018

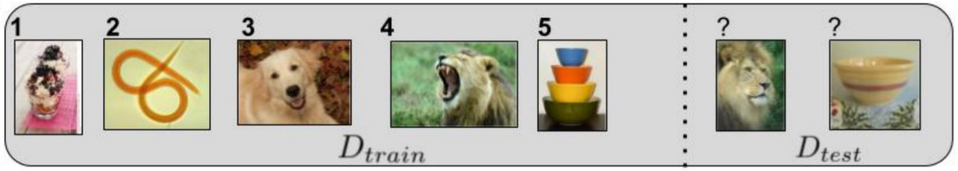
Meta-learning framework

- Problem formulation
  - Each few-shot classification problem as a task

    Each Task: $T \in \mathcal{T}$ $\qquad T \sim P(T)$

  - Each task (or an *episode*) consists of

    $$T = (D_{train}, D_{test}, \mathcal{Y}_T)$$



  - Task-train (*support*) set

    $$D_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L} \qquad \forall y_i \in \mathcal{Y}_T$$

  - Task-test set (query) D_{test}
  - For each task, we adopt an learning algorithm A_ϕ
    - to learn its own classifier F_θ via $F_\theta = A_\phi(D_{train})$
    - to perform well on the task-test set D_{test}

Meta-learning formulation

- Key assumptions:
  - The learning algorithm A_ϕ is shared across tasks
  - We can sample many tasks to learn a good A_ϕ

- A meta-learning strategy
  - Input: meta-training set $\mathcal{D}_{meta-train} = \{(D^{(n)}_{train}, D^{(n)}_{test})\}_{n=1}^{N}$
  - Output: algorithm parameter ϕ^*
  - Objective: good performance on meta-test set

$$\mathcal{D}_{meta-test} = \{(D'^{(n)}_{train}, D'^{(n)}_{test})\}_{n=1}^{N'}$$

  - Minimizing the empirical loss on the meta-training set

$$\min_{\phi} E_{\mathcal{D}_{meta-train}} \left[\text{loss}(F_\theta^{(n)}, D^{(n)}_{test}) \right]$$

    - Each meta-train task $F_\theta^{(n)} = A_\phi(D^{(n)}_{train})$

Meta-learning formulation

- Analogy to standard supervised learning

Supervised-Learning	Train	Test	One data point
Meta-learning	Meta-training	Meta-testing	One task

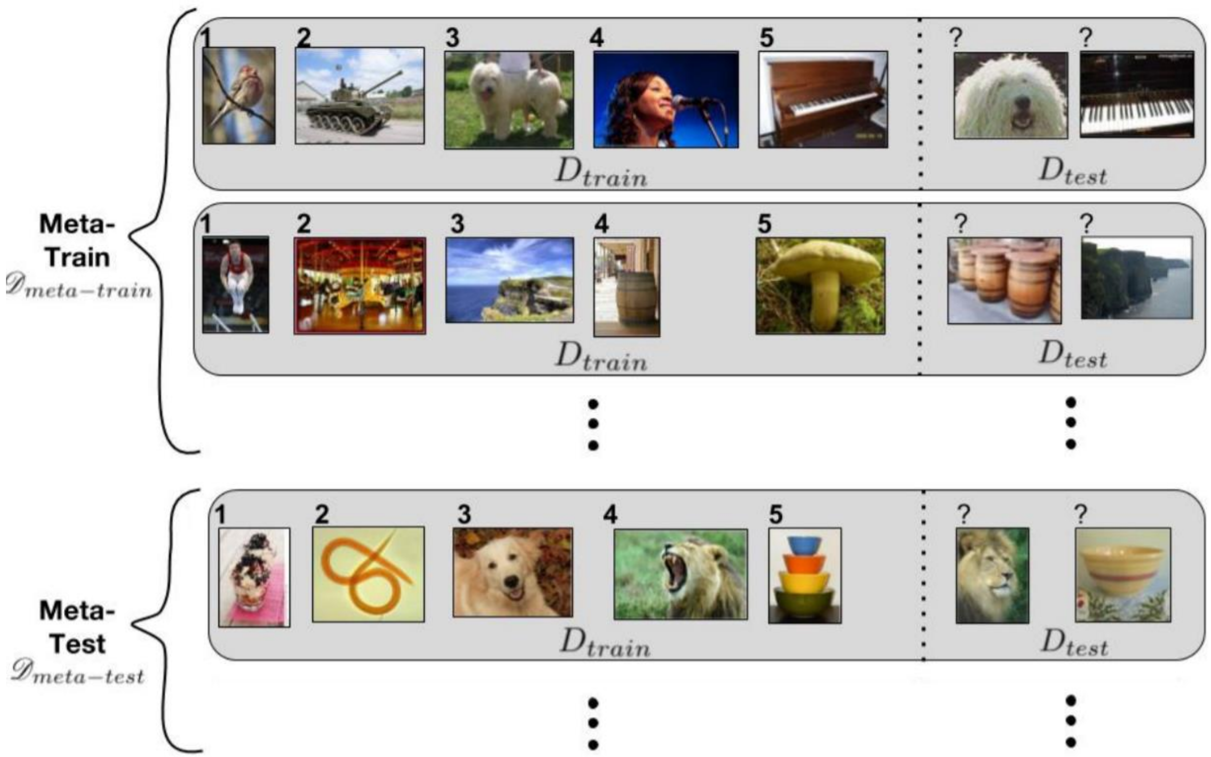
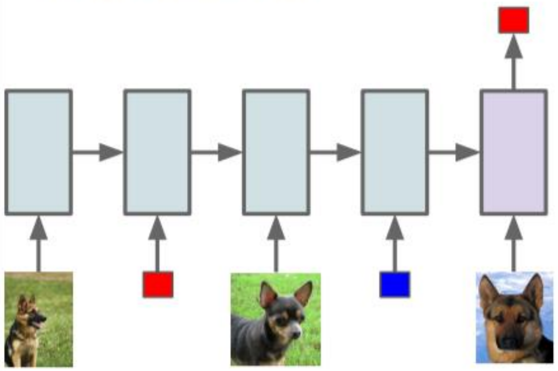


Image Credit: Ravi & Larochelle et al 2017

Overview of existing methods

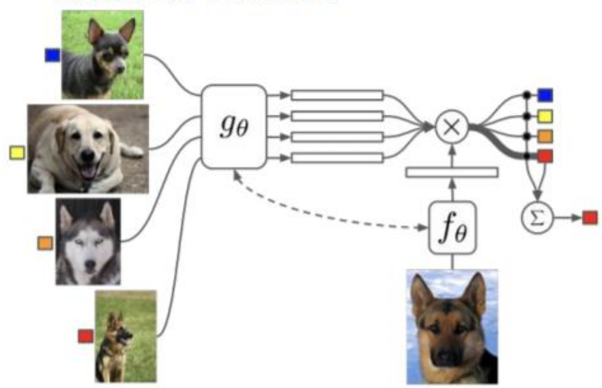
- Depending on the meta-learners used in few-shot tasks

Model Based



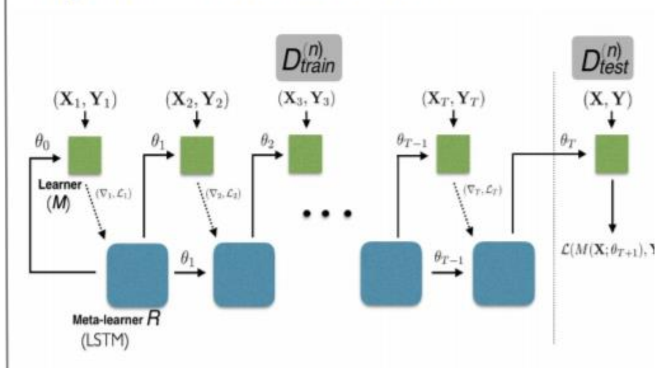
- Santoro et al. ’16
- Duan et al. ’17
- Wang et al. ‘17
- Munkhdalai & Yu ‘17
- Mishra et al. ‘17

Metric Based



- Koch ’15
- Vinyals et al. ‘16
- Snell et al. ‘17
- Shyam et al. ‘17
- Sung et al. ‘17

Optimization Based



- Schmidhuber ’87, ’92
- Bengio et al. ’90, ‘92
- Hochreiter et al. ’01
- Li & Malik ‘16
- Andrychowicz et al. ’16
- Ravi & Larochelle ‘17
- Finn et al. ‘17

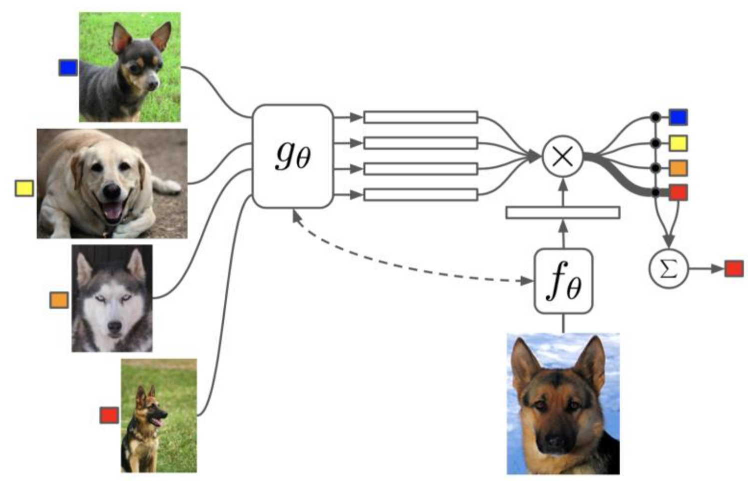
Slide Credit: Vinyals, NIPS 2017

Metric-based methods

- Basic idea: Learn a generic distance metric

$$P_\theta(y|\mathbf{x}, D_{train}) = \sum_{(\mathbf{x}_i, y_i) \in D_{train}} k_\theta(\mathbf{x}, \mathbf{x}_i) y_i$$

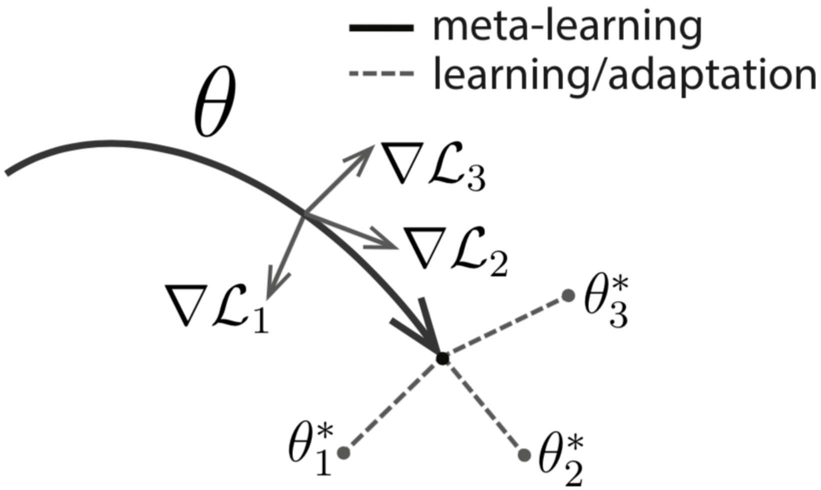
- Typical methods
  - Siamese network (Koch, Zemel & Salakhutdinov, 2015)
  - Matching network (Vinyals et al, 2016)
  - Relation network (Sung et al. 2018)
  - Prototypical network (Snell, Swersky & Zemel, 2017)



Optimization-based methods

- Basic idea: Adjust the optimization in model learning so that the model can effectively learn from a few examples

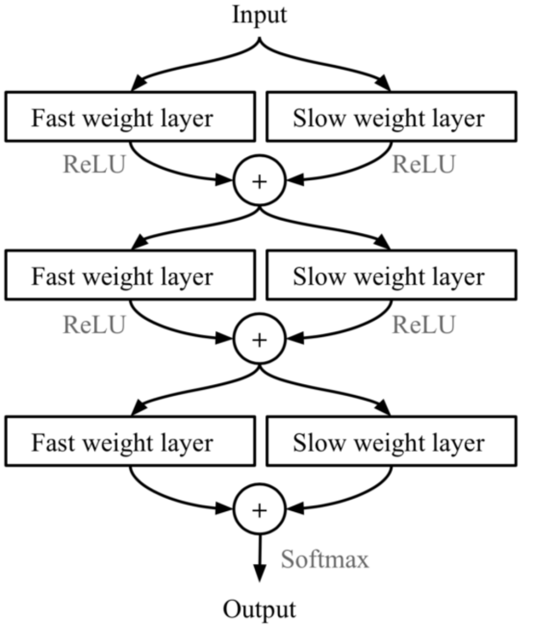
- Typical methods
  - LSTM meta-learner (Ravi & Larochelle, 2017)
  - MAML (Finn, et al. 2017)
  - Reptile (Nichol, Achiam & Schulman, 2018)



Model-based methods

- Basic idea: Building a neural network with specific architecture for fast learning

- Typical methods
  - Memory-augmented network (Santoro et al., 2016)
  - Meta networks (Munkhdalai & Yu, 2017)
  - SNAIL (Mishra et al., 2018)



Main limitations

- A global representation of inputs
  - Sensitive to nuisance parameters: background clutter, occlusions, etc.
- Mixed representation and predictor learning
  - Complex architecture, difficult to interpret
  - Sometimes slow convergence
- Focusing on classification tasks
  - Non-trivial to apply to other vision tasks: localization, segmentation, etc.

Our proposed solutions

- Structure-aware data representation
  - Spatial/temporal representations for semantic objects/actions
- Decoupling representation and classifier learning
  - Improving representation learning
- Generalizing to other visual tasks
  - Instance localization and detection with few-shot learning

Outline

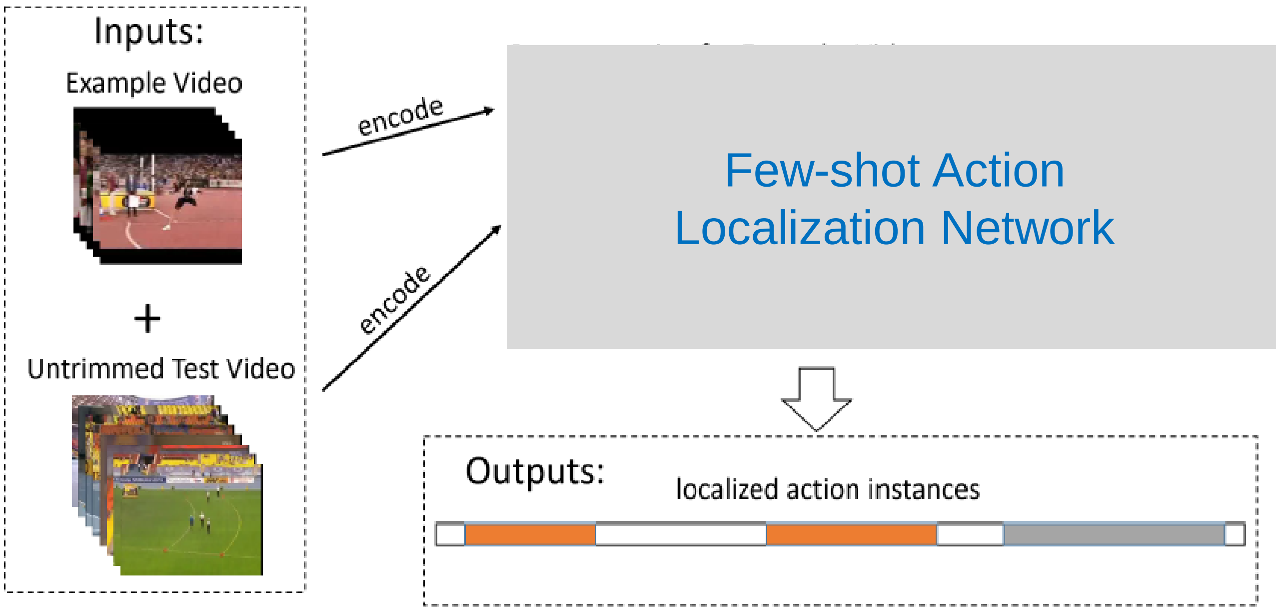
- Introduction
  - Learning from very limited annotated data
- Background in few-shot learning
  - Few-shot classification
  - Meta-learning framework
- Towards few-shot representation learning in vision tasks
  - **Spatio-temporal patterns in videos [CVPR 2018]**
  - Visual object & task representation [AAAI 2019]
- Summary and future directions

Temporal action localization

- Our goal: Jointly classify action instances and localize them in an untrimmed video
  - Important for detailed video understanding
  - Broad range of applications in video surveillance/analytics

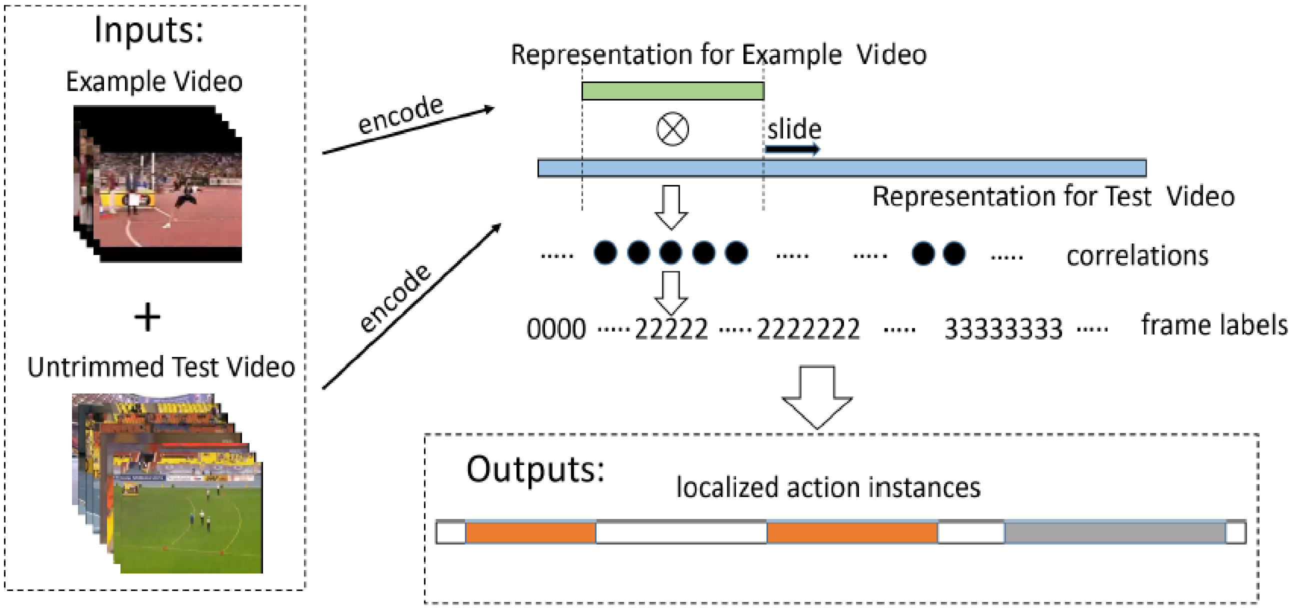
Our problem setting

- We conceptualize an example-based action localization strategy
  - Few-shot learning of action classes and
  - Being sensitive to action boundaries

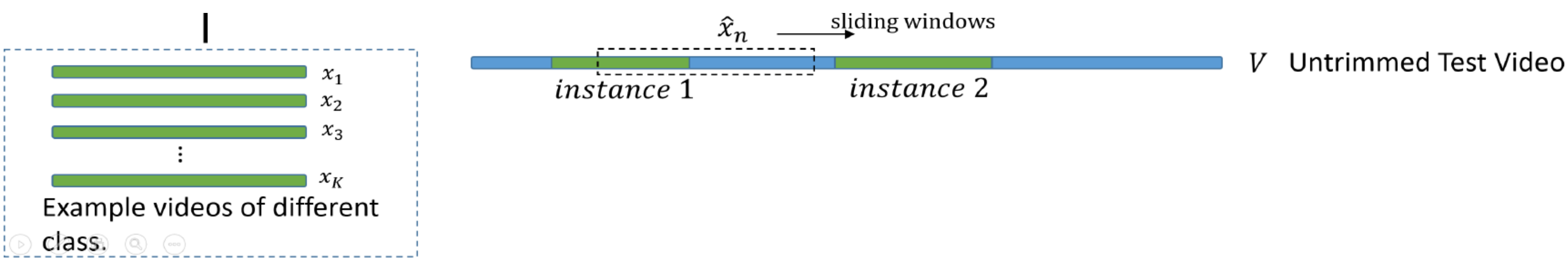


Main ideas

- Meta-learning problem formulation
  - Learning how to transfer the labels of a few action examples to a test video
    - Encode action instance into a structured representation
    - Learn to match (partial) action instances
    - Exploit the matching correlation scores

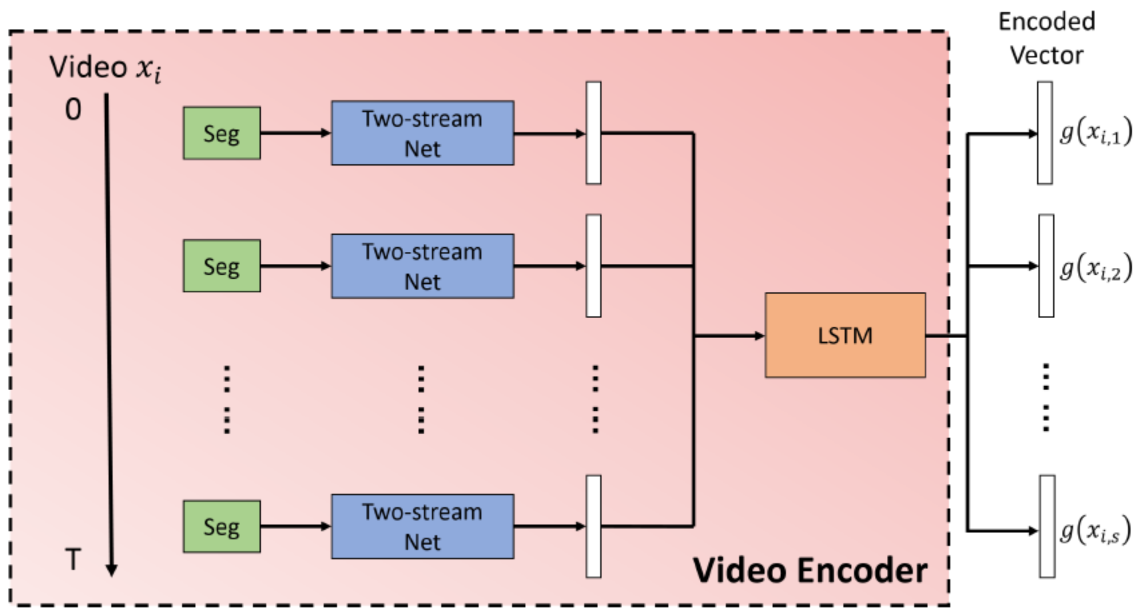


Overview of our method



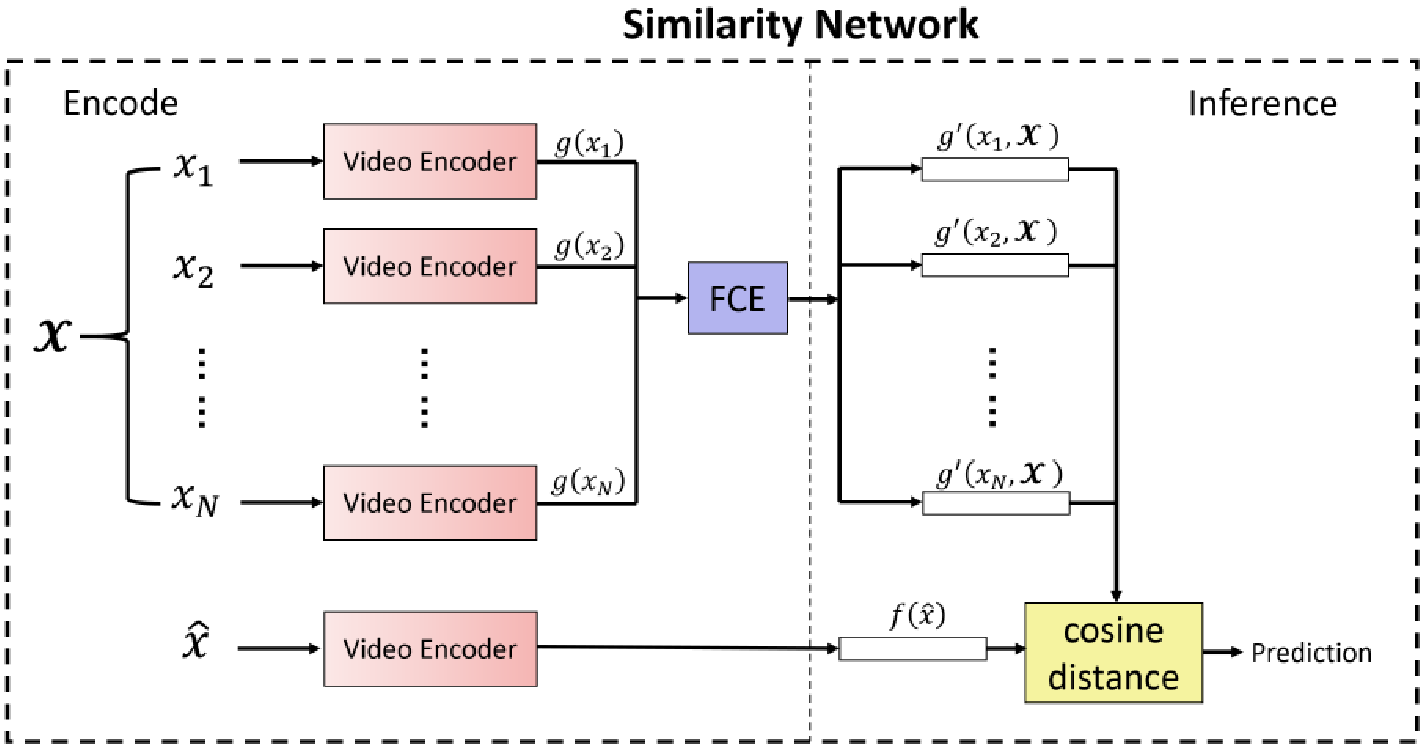
Video encoder network

- Embed an action video into a segment-based representation
  - Maintain its temporal structure
  - Allows partial matching between two actions



Similarity Network

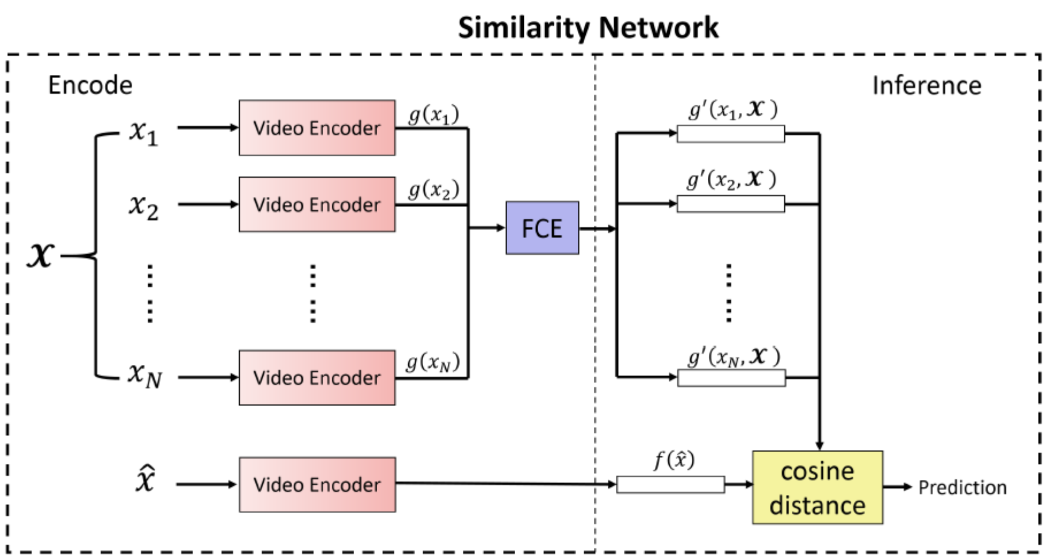
- Generate a matching score between labeled examples (support set) and a test window



Similarity Network

- Full context embedding (FCE)
  - Capture context of the entire support set and enrich the action representations

$$g'(x_i, \mathcal{X}) = \overrightarrow{h}_i + \overleftarrow{h}_i + g(x_i)$$
$$\overrightarrow{h}_i, \overrightarrow{c}_i = \mathrm{LSTM}(g(x_i), \overrightarrow{h}_{i-1}, \overrightarrow{c}_{i-1})$$
$$\overleftarrow{h}_i, \overleftarrow{c}_i = \mathrm{LSTM}(g(x_i), \overleftarrow{h}_{i+1}, \overleftarrow{c}_{i+1})$$

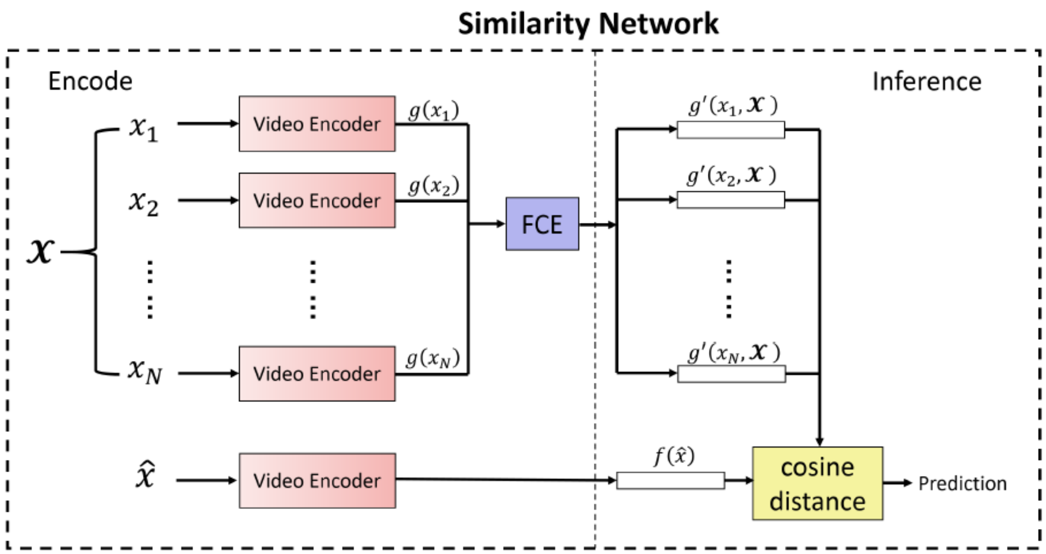


Similarity Network

- Similarity scores
  - Cosine distance between two action instances

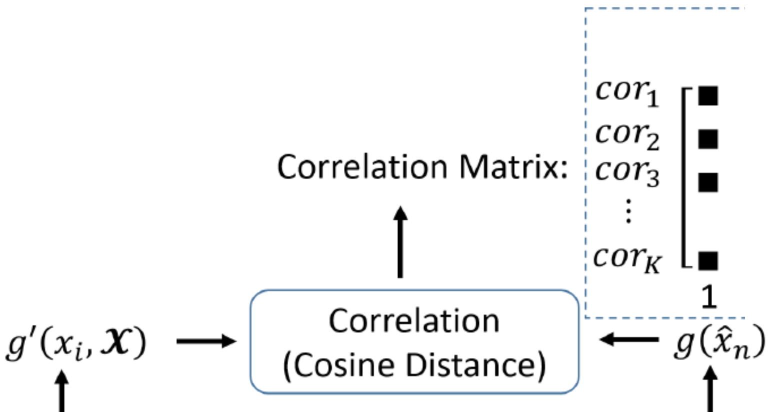
$$d(\hat{x}, x_i) = \frac{g(\hat{x})^T g'(x_i, \mathcal{X})}{|g(\hat{x})| \cdot |g'(x_i, \mathcal{X})|}$$

  - Nearest neighbor for classification, but what about localization?

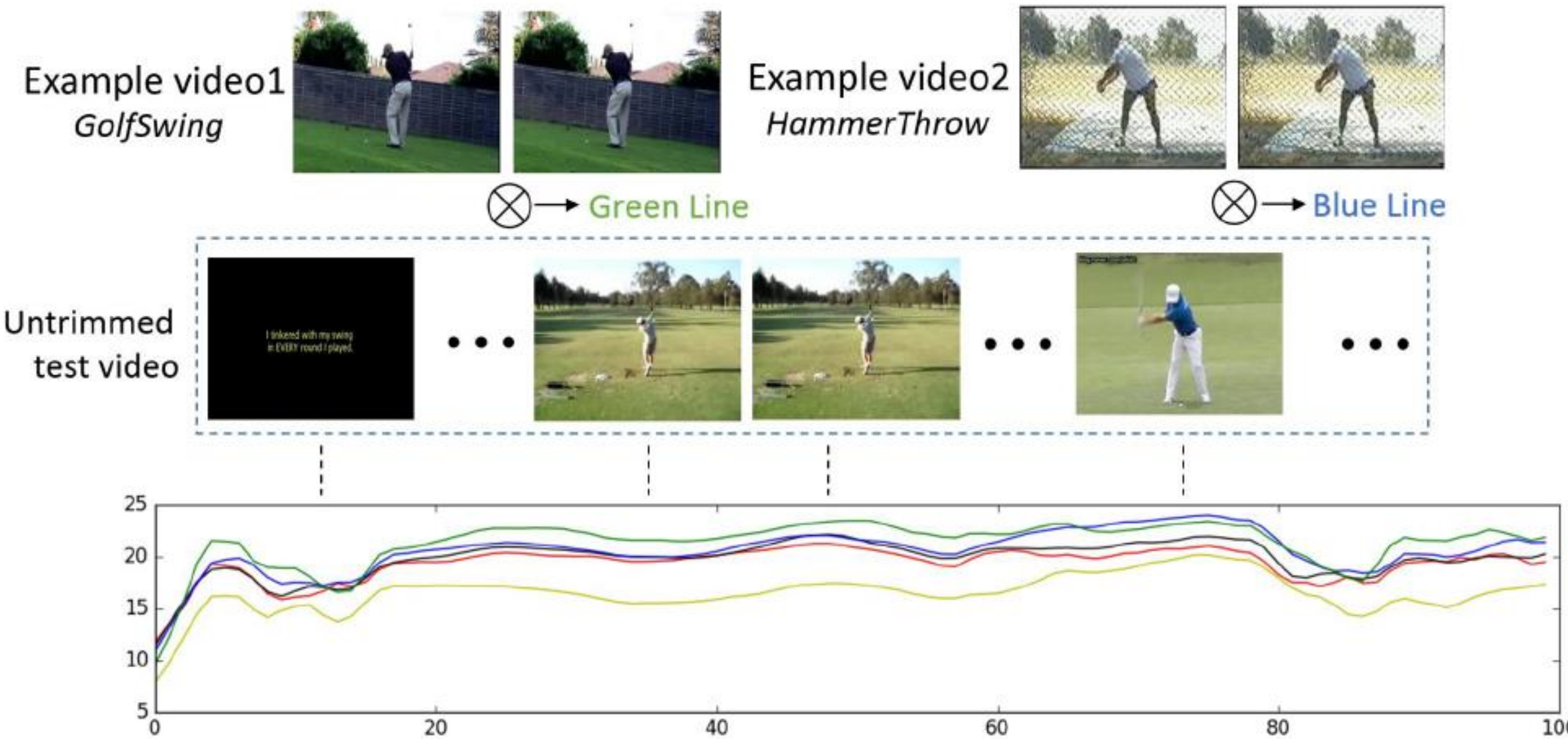


Labeling network

- Cache correlation scores for sliding windows
- Exploit patterns in the score matrix to predict the locations



Matching examples



Matching score trajectories

Meta-learning strategy

- Meta-training phase
  - Meta-training set $\mathcal{T}_{meta-train} = \{\mathcal{X}, \hat{\mathcal{X}}, \mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}, \theta)\}$
  - Task-train (support set) $\mathcal{X} = \{x_i, y_i\}$
  - Task-test (query) $\hat{\mathcal{X}} = \{\hat{x}_j, \hat{y}_j\}$
  - Loss function $\mathcal{L}$
- Our loss function
  - Localization loss: foreground vs background (cross entropy)
  - Classification loss: action class (log loss)

$$L = \mathbb{E}_{\mathcal{T} \sim \mathcal{T}_{meta-train}}[L_{loc} + L_{cls}]$$

  - Ranking loss: replacing localization loss to encourage partial alignment

Experimental evaluation

- Few-shot performance summary
  - ~80 classes for meta-training and ~20 for meta-test

Fully supervised	mAP	Few-shot	mAP
Heilbron *et al.* [5]	13.5	Ours@1	13.6
Yeung *et al.* [49]	17.1	Ours@5	14.0
Yuan *et al.* [50]	17.8	Ours@15	14.7
S-CNN [35]	19.0	CDC@1	6.4
S-CNN + SST [4]	23.0	CDC@5	6.5
CDC [34]	23.3	CDC@15	6.8

Thumos14

	mAP@0.5	Average mAP
TCN [8]	37.4	23.5
R-C3D [48]	-	26.8
Wang *et al.* [26]	42.2	14.8
Lin *et al.* [27]	48.9	32.2
Xiong *et al.* [47]	41.1	24.8
CDC [34]	43.8	22.7
Ours@1	22.3	9.8
Ours@5	23.1	10.0
CDC@1	8.2	2.4
CDC@5	8.6	2.5

ActivityNet

Ablative Study

- Effect of the similarity net

	mAP
Our encoder + Binary classifier	6.0
Our encoder + Similarity net	13.6

- Effect of temporal structure

Alignment Scheme	mAP	accuracy
No alignment	7.8	-
TS, w/o ranking loss	12.9	57.5
TS, with ranking loss	13.6	57.4

Outline

- Introduction
  - Learning from very limited annotated data
- Background in few-shot learning
  - Few-shot classification
  - Meta-learning framework
- Towards few-shot representation learning in vision tasks
  - Spatio-temporal patterns in videos [CVPR 2018]
  - **Visual object & task representation [AAAI 2019]**
- Summary and future directions

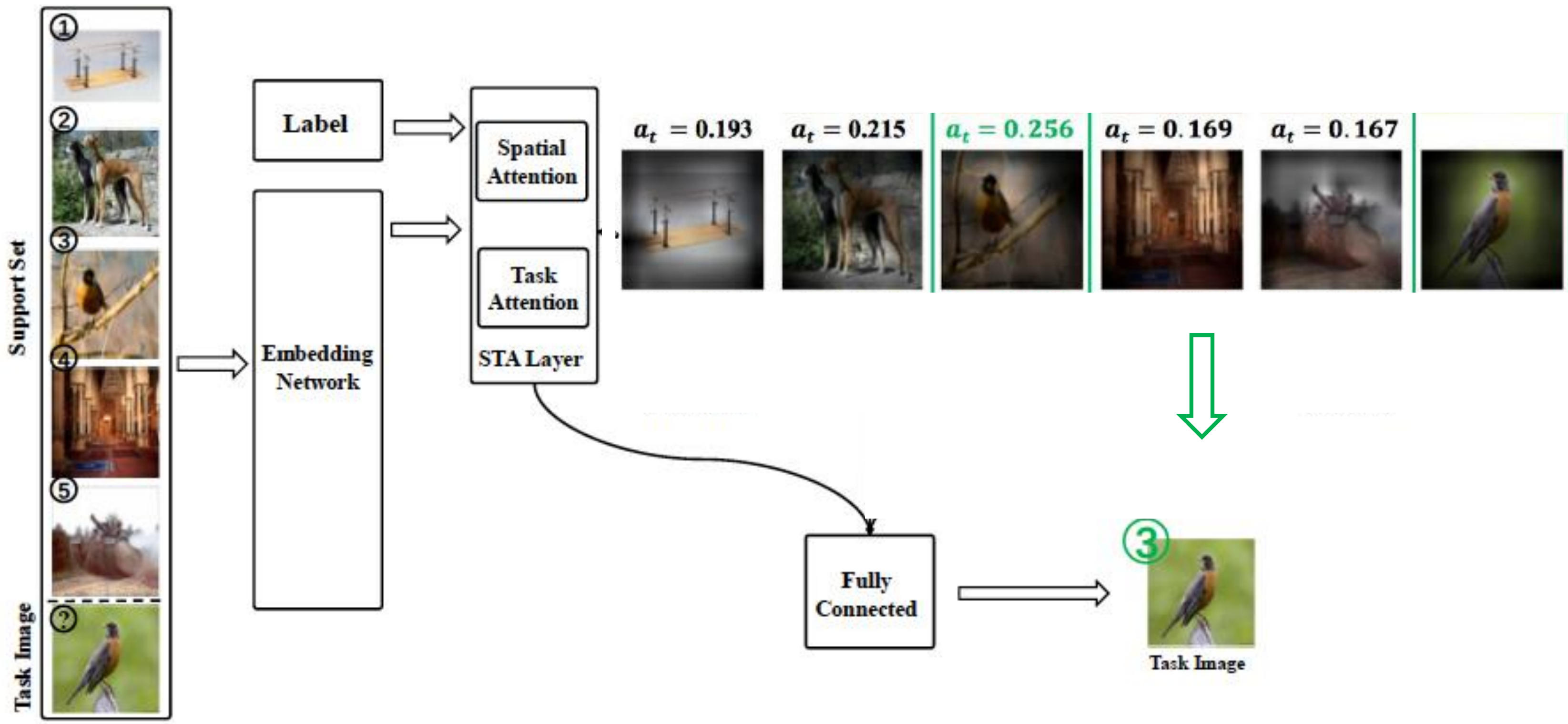
Task: Few-shot image classification

- Our goal: An efficient modular meta-learner for visual concepts
  - A better image representation
  - An easy-to-interpret encoding method for support set

Image Credit: Ravi & Larochelle et al 2017

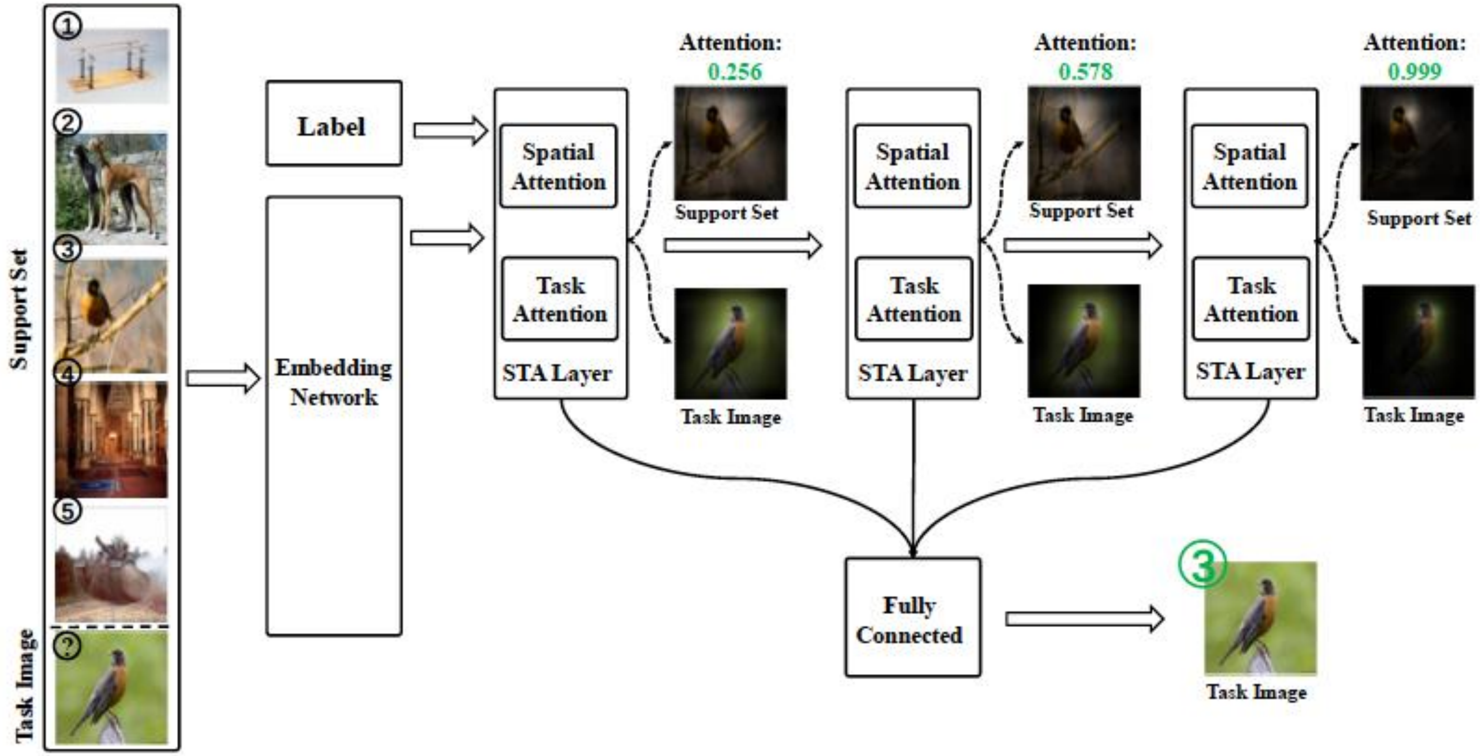
Main idea

- Exploiting attention mechanism in representation learning
  - Spatial attention to localize the foreground object
  - Task attention to encode the task context for label prediction



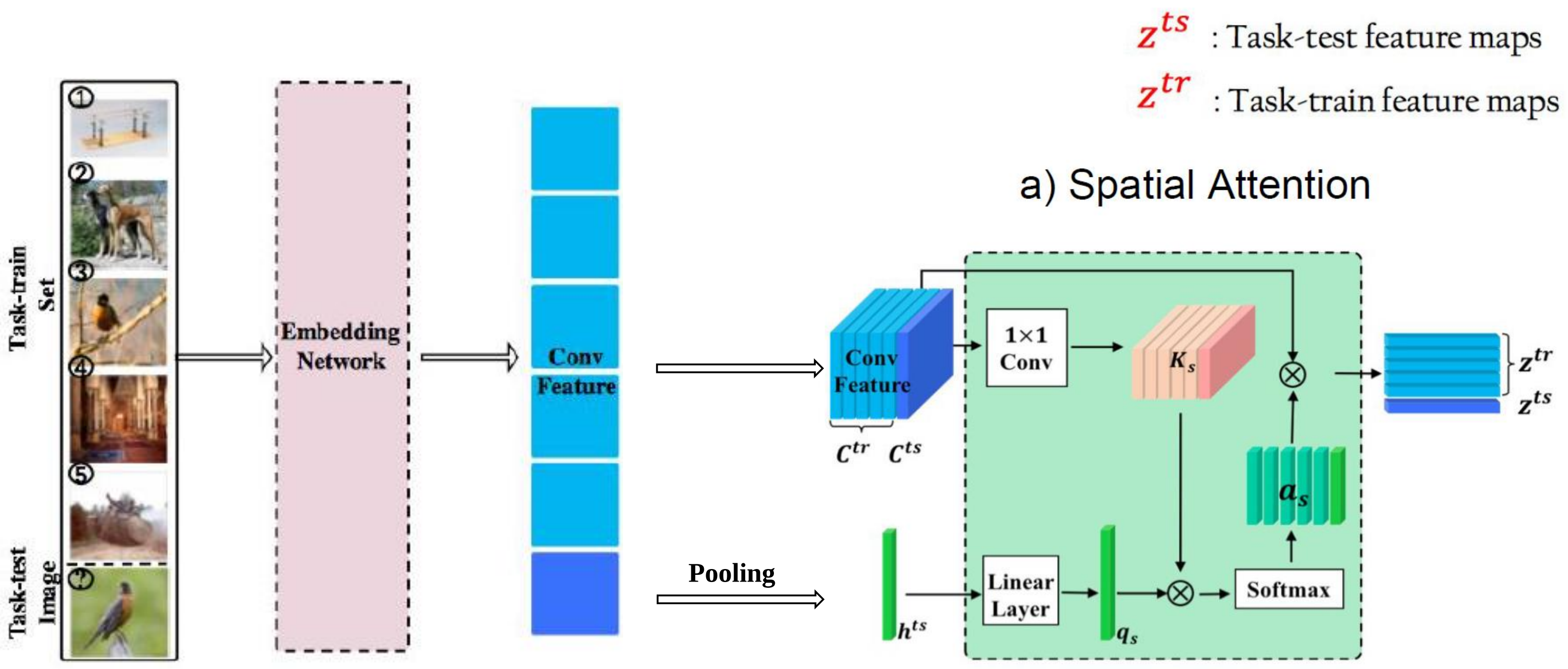
Main idea

- Exploiting attention mechanism in representation learning
  - Recurrent attention to refine the representation



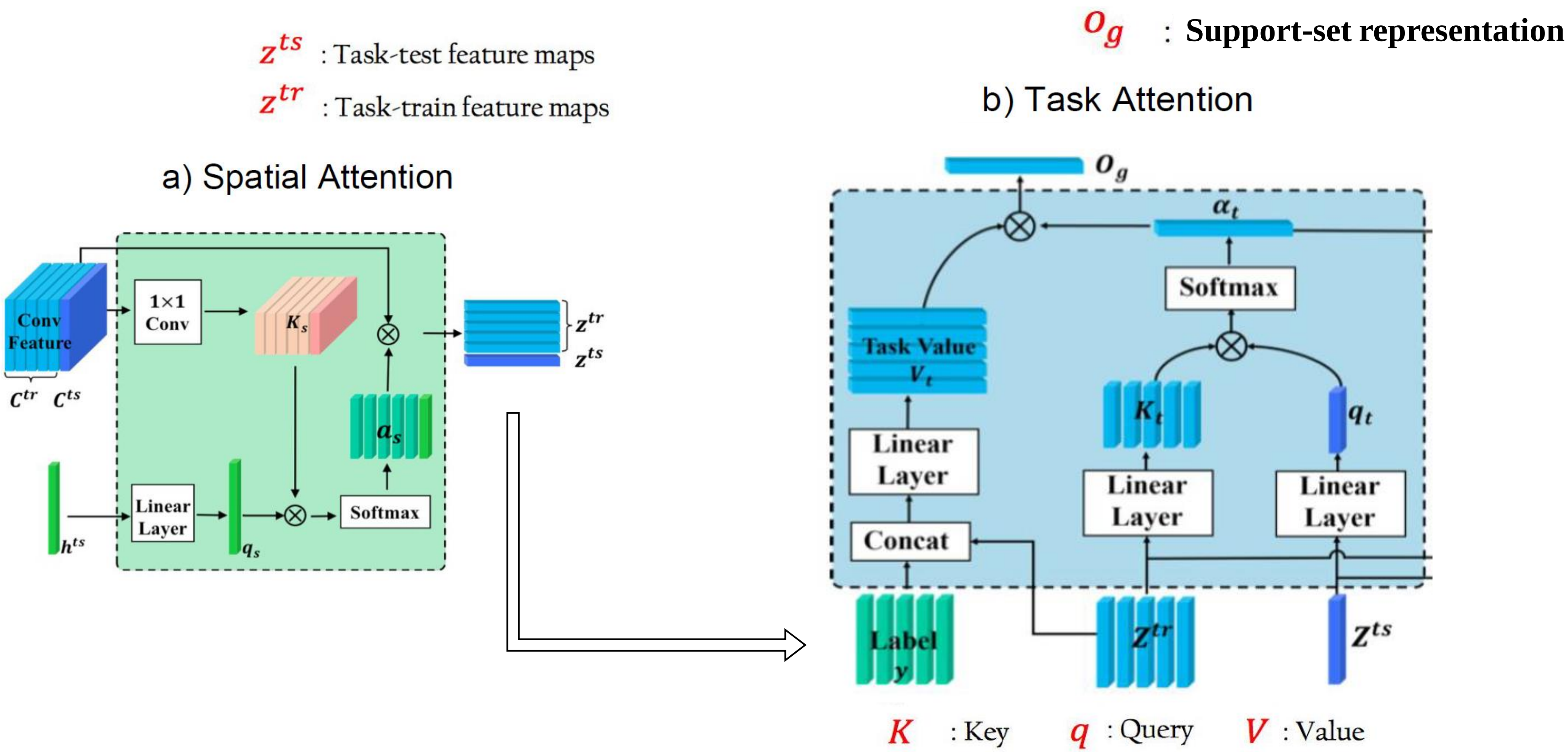
Dual-attention structure

- Spatial attention
  - Extracting relevant features on Conv-feature maps
  - Using test image feature as query



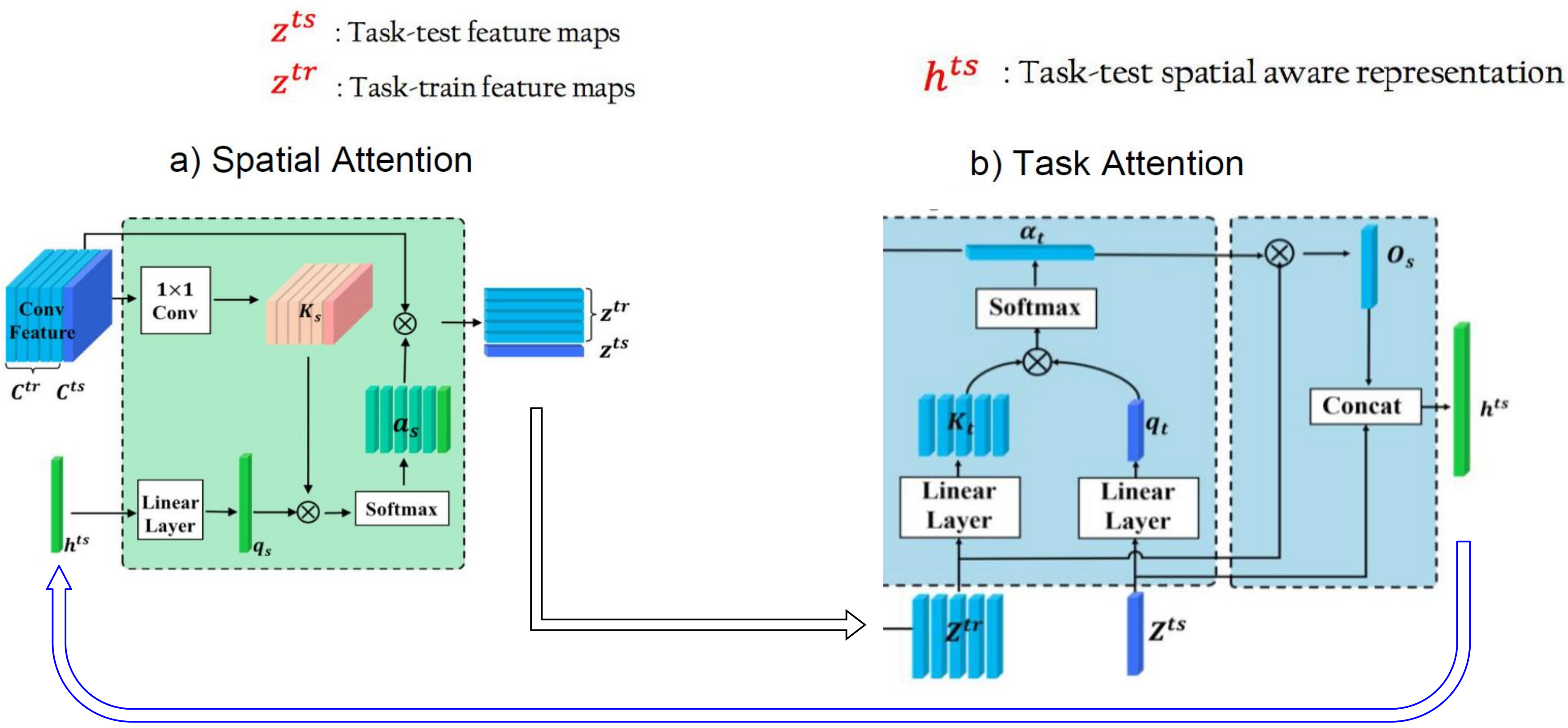
Dual-attention structure

- Task attention
  - Encoding the support set by selecting relevant training examples



Dual-attention structure

- Recurrent attention
  - Refining task-test (query image) features with support set



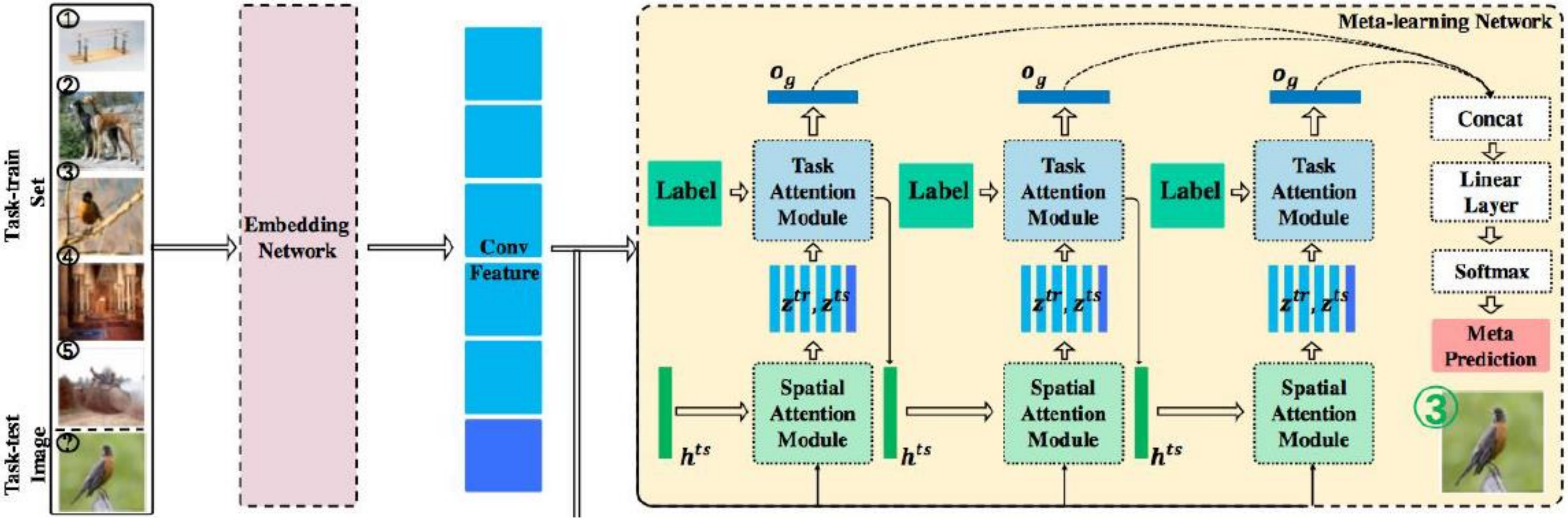
Network architecture

z^{tr} : Task−train feature maps

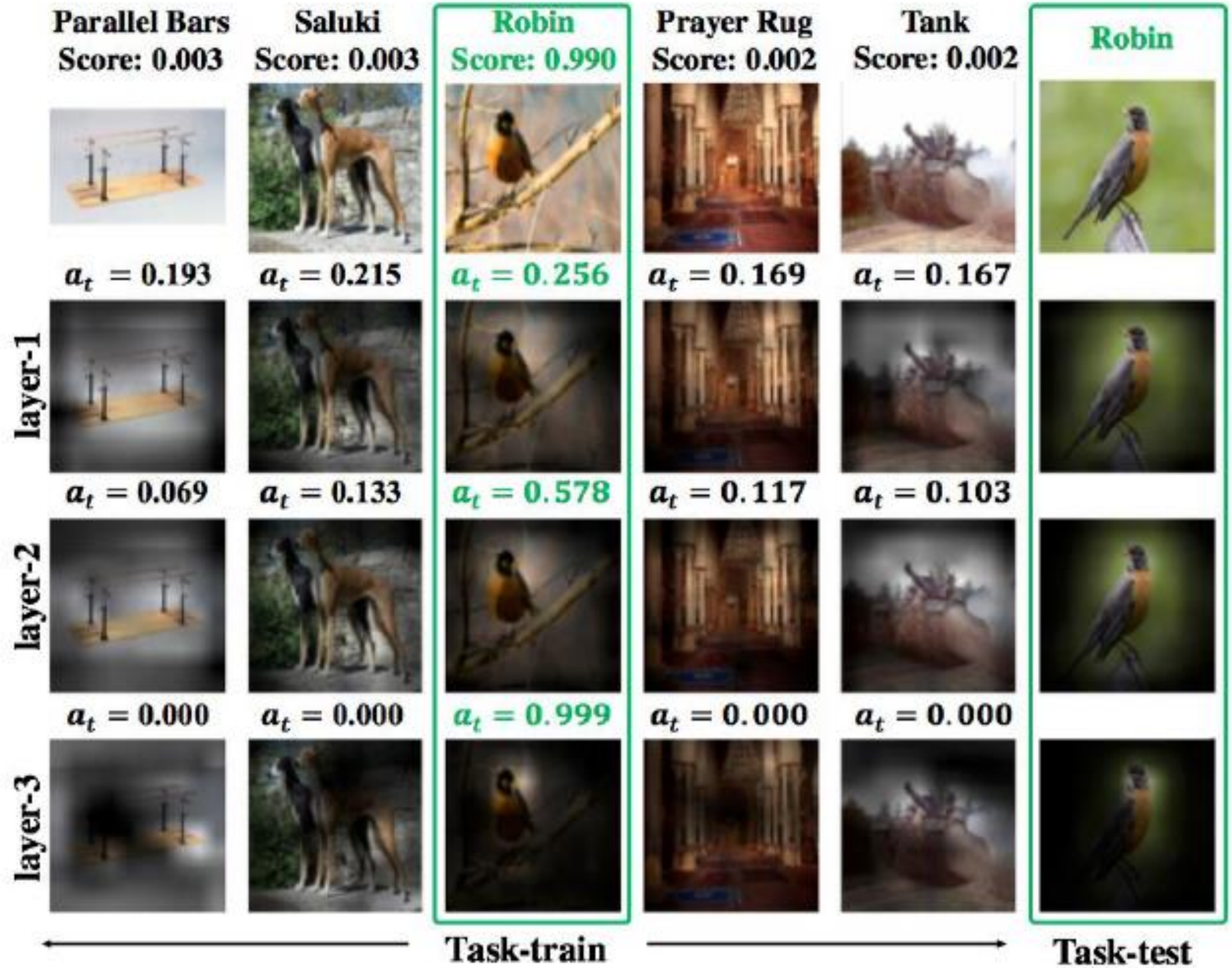
z^{ts} : Task−test feature maps

o_g : Task−test label prediction score

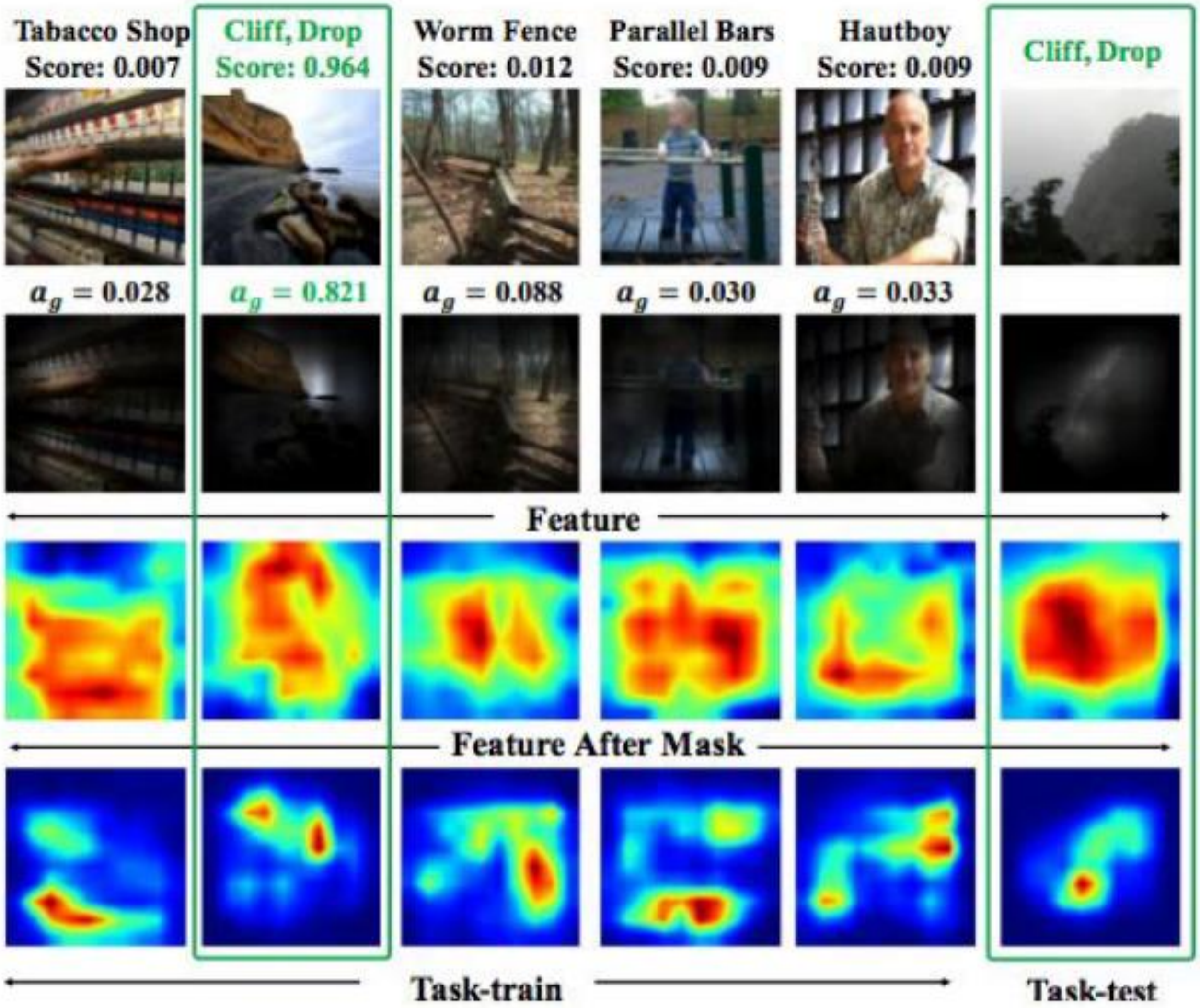
h^{ts} : Task−test spatial aware representation



Example results



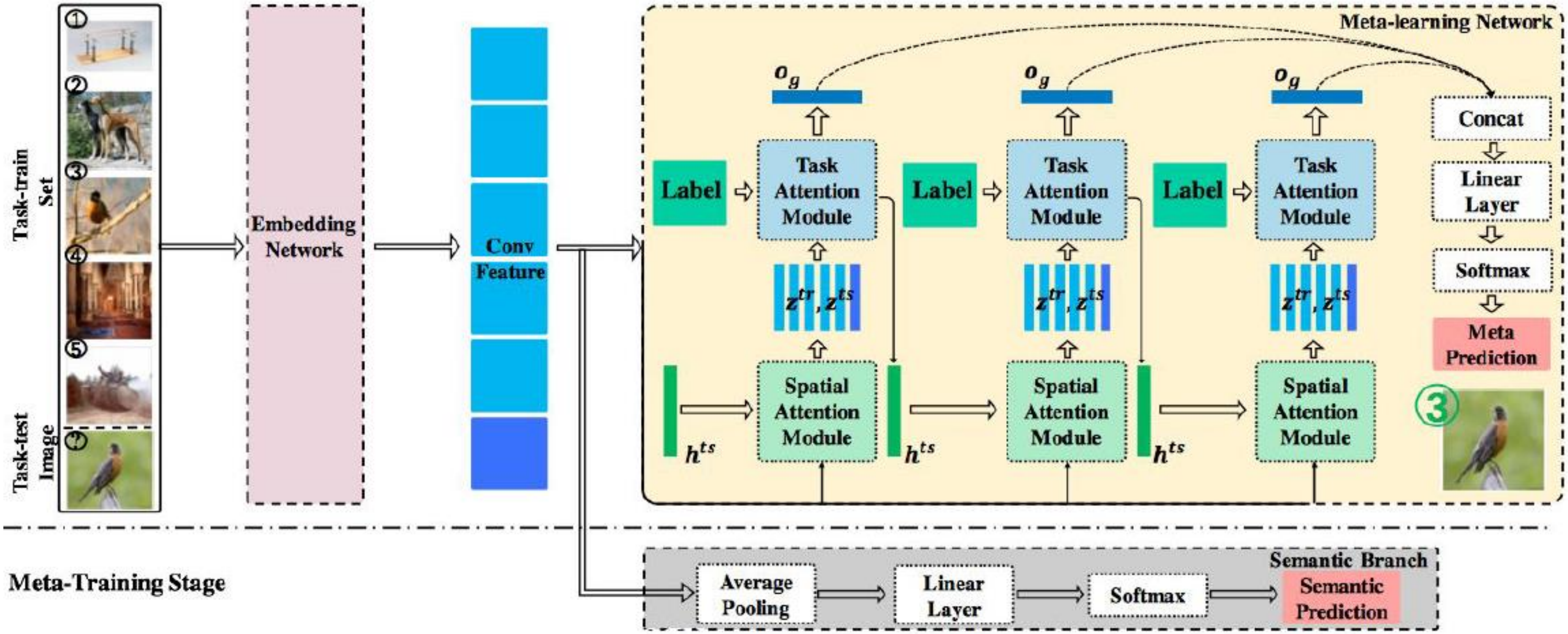
Example results



A hybrid loss function

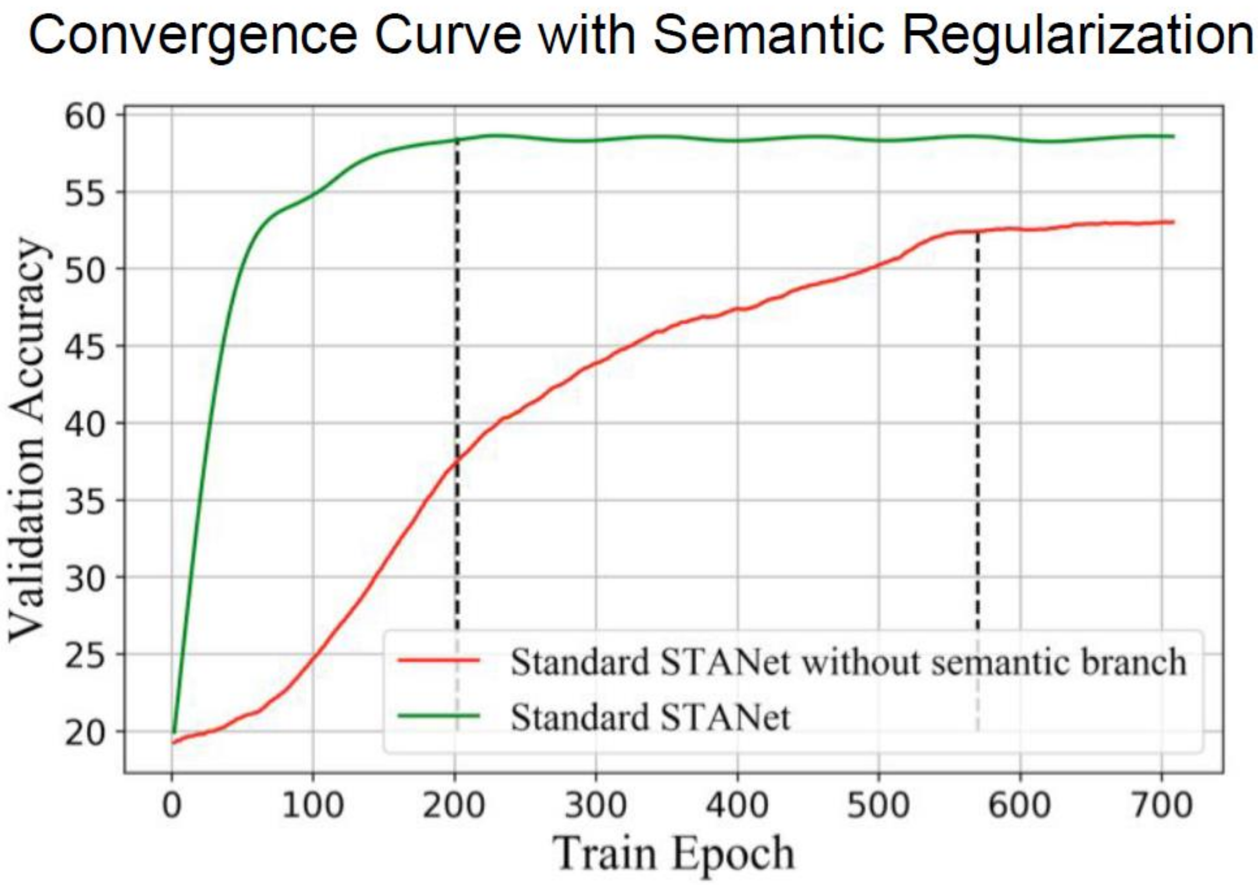
- Standard meta-learning loss + global classification loss

$$\boxed{L_{full} = L_{task}(\Theta) + \lambda L_{sem}(\Theta)}$$



Validation of hybrid loss

- We train our models from scratch (no pre-training)



Quantitative results

- MiniImageNet:
  - 80 classes for meta-training and 20 for meta-test
  - Roughly 100K tasks for training and 1K for test

Method	# Params	Feature Extractor	5-way Accuracy	
			1-shot	5-shot
Matching Net (Vinyals et al. 2016)	0.1M	Conv64	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
Prototypical Net(Snell et al. 2017)	0.1M	Conv64	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
MAML (Finn et al. 2017)	0.1M	Conv64	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
RelationNet(Sung et al. 2018)	0.23M	Conv64	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
(Gidaris et al. 2018)	0.24M	Conv64	$56.20 \pm 0.86\%$	$72.81 \pm 0.62\%$
GNN (Satorras et al. 2018)	1.6M	Conv64	$50.33 \pm 0.36\%$	$66.41 \pm 0.63\%$
STANet-S(1-Layer)	0.24M	Conv64	$50.38 \pm 0.65\%$	$65.67 \pm 0.66\%$
STANet-S(3-Layer)	0.24M	Conv64	$\mathbf{53.11 \pm 0.60\%}$	$\mathbf{67.16 \pm 0.66\%}$
SNAIL (Mishra et al. 2018)	6.1M	ResNet-12	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$
(Gidaris et al. 2018)	2.6M	ResNet-12	$55.45 \pm 0.86\%$	$70.13 \pm 0.68\%$
(Qiao et al. 2018)	40.5M	WRN-28	$\mathbf{59.60 \pm 0.41\%}$	$\mathbf{73.74 \pm 0.19\%}$
STANet(1-Layer)	2.6M	ResNet-12	$57.25 \pm 0.40\%$	$69.45 \pm 0.50\%$
STANet(3-Layer)	2.6M	ResNet-12	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

Quantitative results

- MiniImageNet:
  - 80 classes for meta-training and 20 for meta-test
  - Roughly 100K tasks for training and 1K for test

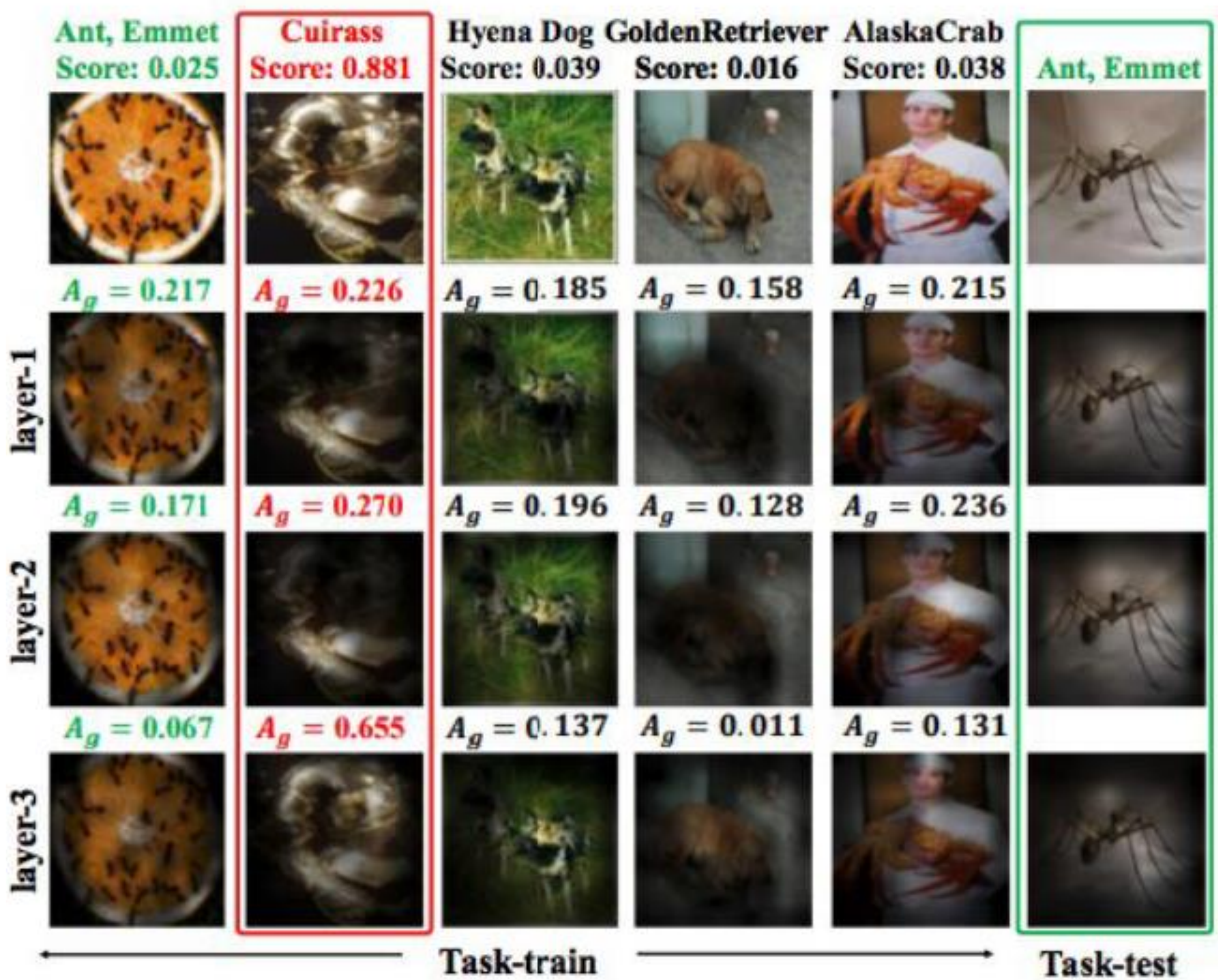
b) Ablation Study

Components			5-way(Normal)	
SR.	SA.	TA.	1-shot	5-shot
✗	Uniform	✓	$53.41 \pm 0.61\%$	$64.32 \pm 0.57\%$
✗	Gaussian	✓	$54.29 \pm 0.66\%$	$65.41 \pm 0.55\%$
✗	✓	✓	$55.52 \pm 0.64\%$	$66.75 \pm 0.62\%$
✓	✓	✓	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

SR: Semantic Regularization,
SA: Spatial Attention
TA: Task Attention

Failure cases

- Large variations in scale/viewpoint



Research questions I

- Task similarity
  - A new benchmark: Meta-CIFAR100

Task Difficulty Level:

- Easy
  - Share superclass between meta-train and meta-test
- Hard
  - Totally different superclass between meta-train and meta-test
- Moderate
  - Part-shared, part-different

To study:

1. How related the test tasks are to the training tasks
2. How much semantic knowledge can be transferred.

Superclass
aquatic mammals
fish
flowers
food containers
fruit and vegetables
household electrical devices
household furniture
insects
large carnivores
large man–made outdoor things
large natural outdoor scenes
large omnivores and herbivores
medium–sized mammals
non–insect invertebrates
people
reptiles
small mammals
trees
vehicles 1
vehicles 2

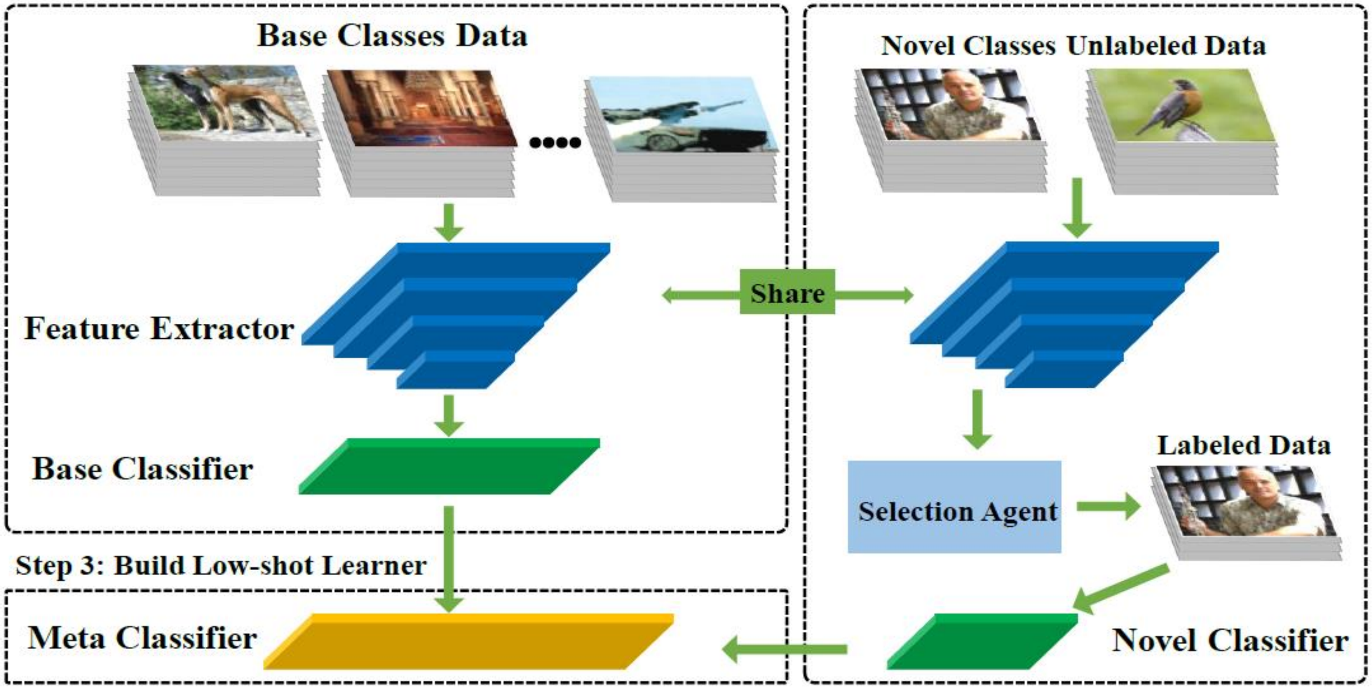
Research questions I

- Preliminary results on Meta-CIFAR100
  - Task similarity plays a key role in few-shot performance

Method	# Params	Feature Extractor	Easy		Moderate		Hard	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML (Finn et al. 2017)	0.1M	Conv64	$55.17 \pm 1.90\%$	$74.12 \pm 0.86\%$	$46.30 \pm 1.90\%$	$58.49 \pm 0.94\%$	$38.46 \pm 1.83\%$	$52.17 \pm 0.89\%$
STANet(1-layer)	0.1M	Conv64	$\mathbf{65.51 \pm 0.44\%}$	$\mathbf{78.31 \pm 0.38\%}$	$\mathbf{56.62 \pm 0.46\%}$	$\mathbf{67.38 \pm 0.51\%}$	$\mathbf{42.94 \pm 0.41\%}$	$\mathbf{54.91 \pm 0.38\%}$
STANet(3-layer)	0.1M	Conv64	$\mathbf{66.11 \pm 0.42\%}$	$\mathbf{78.54 \pm 0.51\%}$	$\mathbf{57.31 \pm 0.39\%}$	$\mathbf{68.71 \pm 0.43\%}$	$\mathbf{42.98 \pm 0.43\%}$	$\mathbf{55.23 \pm 0.42\%}$

Research questions II

- From few-shot to low-shot learning
  - Novel classifier: incremental few-shot learning
  - How do we exploit unlabeled data?



Summary and future directions

- Few-shot visual concept learning
  - Structured representation is important
  - Modularized, interpretable network design
  - Extension to multiple vision tasks

- Future directions
  - Studying impact of different task distributions
  - Connecting few-shot learning to continual learning
  - Exploring few-shot learning in real-world applications

Acknowledgement

- PhD students
  - Hongtao Yang @ANU
  - Songyang Zhang and Shipeng Yan @ShanghtaiTech

Thank You & Question!