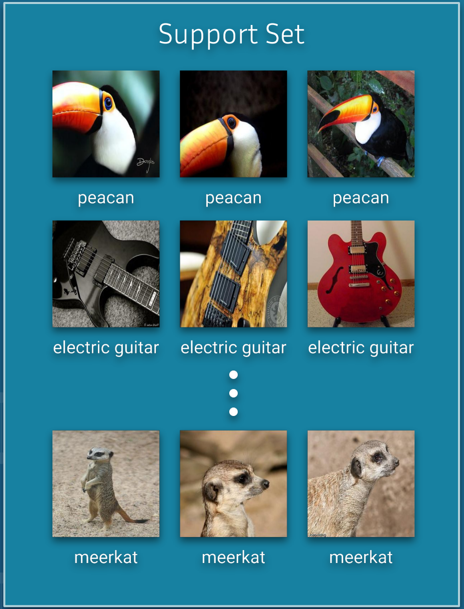
Few shot learning by features adaptation with Graph Neural Networks

Armand Nicolicioiu

Andrei Nicolicioiu

Problem setup

- Train a classifier using only a few examples per class



Problem setup

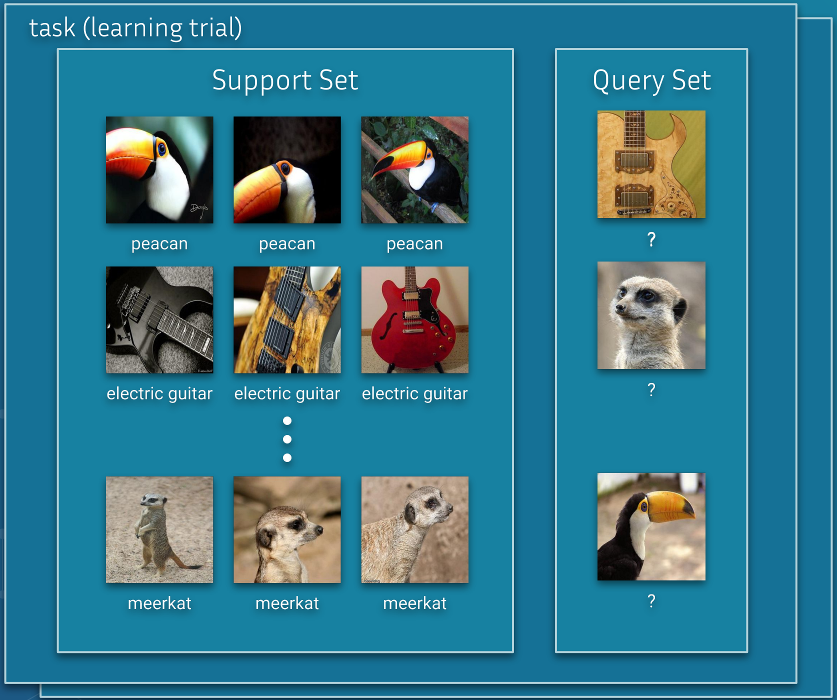
- Train a classifier using only a few examples per class

Problem setup

- Train a classifier using only a few examples per class

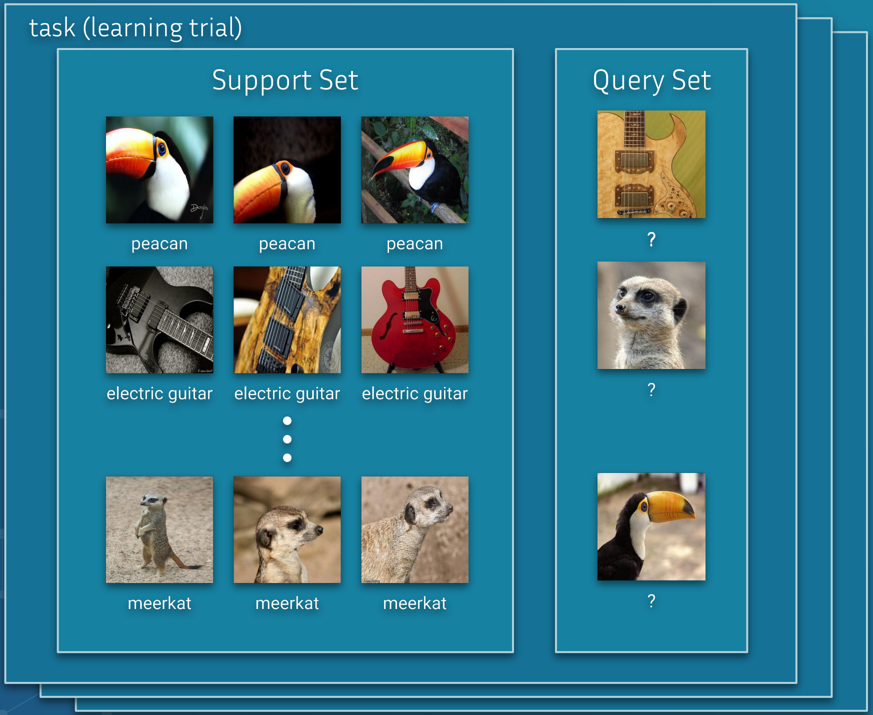
Problem setup

- Train a classifier using only a few examples per class

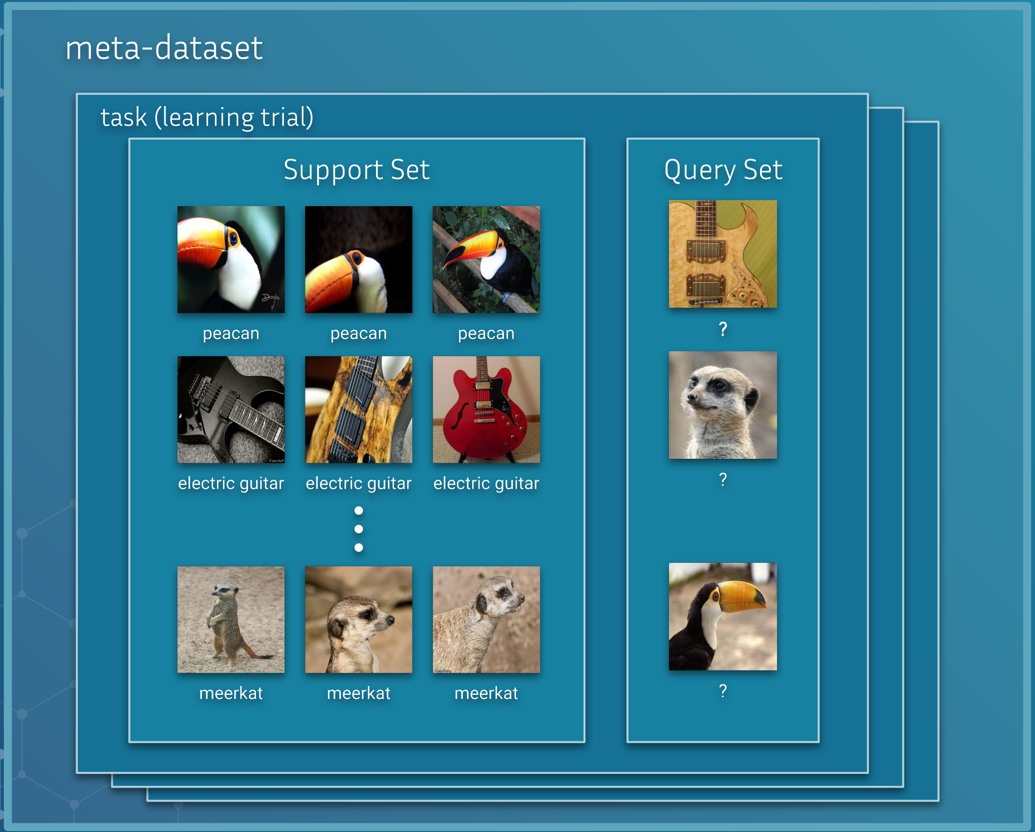


Problem setup

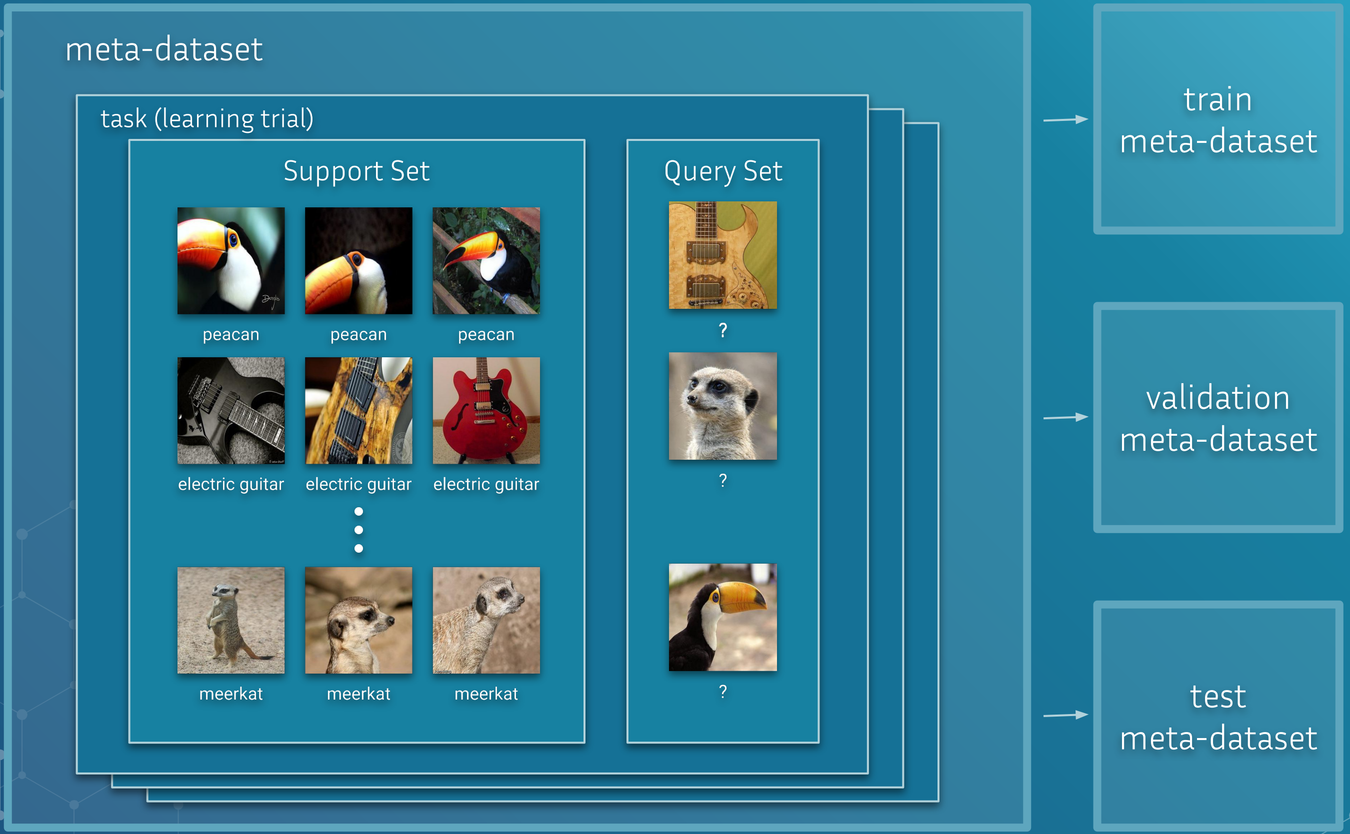
- Train a classifier using only a few examples per class



Problem setup



Problem setup



Existing methods

- Non parametric
  - Matching Networks (Vinyals et al., 2016)
  - Prototypical Networks (Snell et al., 2017)
- Optimization based
  - MAML (Finn et al., 2017)
  - CAVIA (Zintgraf et al., 2018)
- Hybrid:
  - CAML (Jiang et al., 2019)
  - Triantafillou et al. (2020)

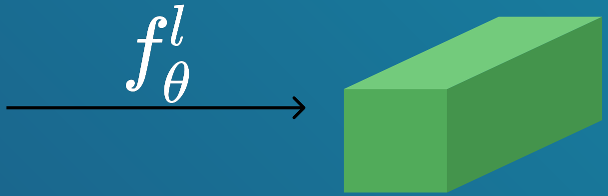
Our method

- Intuition
  - It’s easy to overfit the few samples
  - The context is very important

- Proposed solution
  - **Adapt** the activations by info explicitly extracted from the support set

$$\hat{f}_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) \odot \gamma + \beta$$

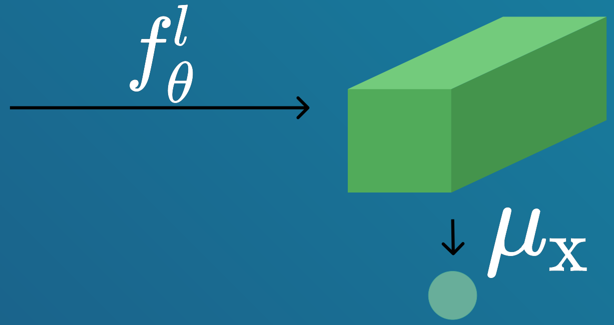
Our method



- Extract query features

$$f_\theta(\mathbf{x})$$

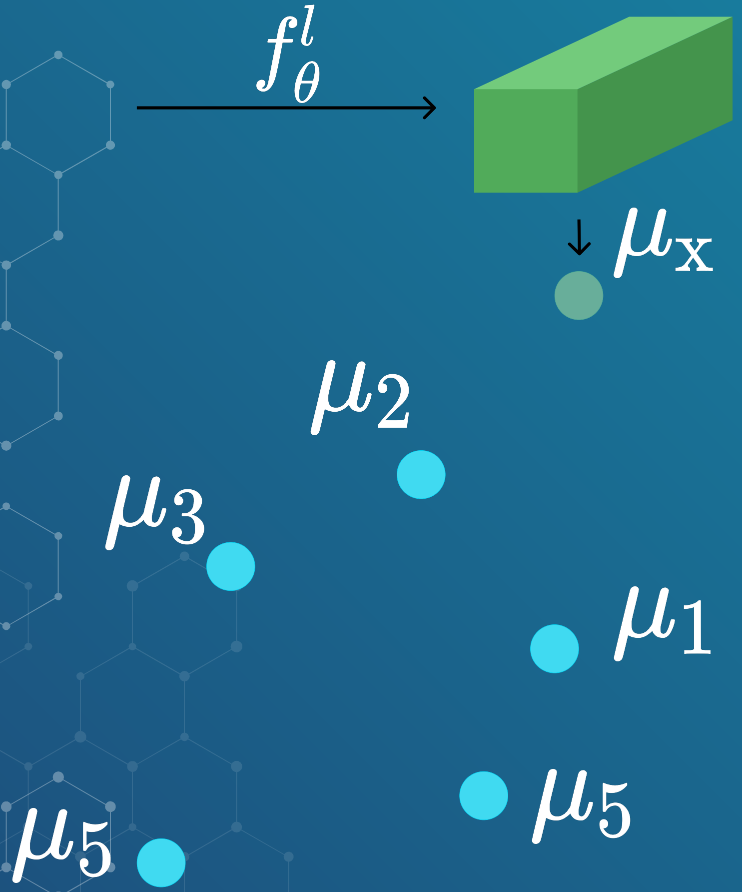
Our method



- Form query prototype

$$\mu_x$$

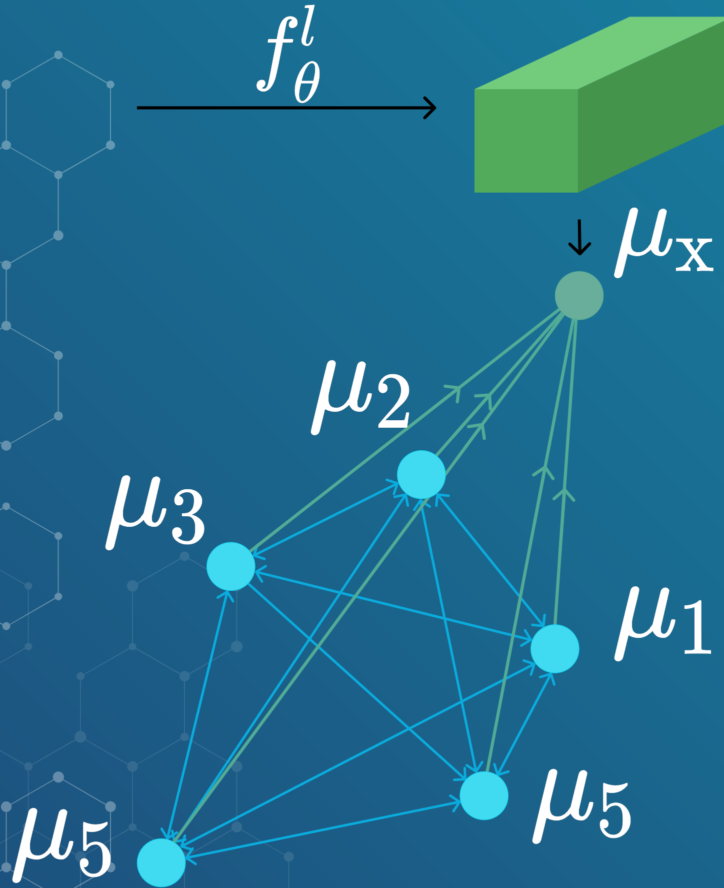
Our method



- Extract support prototypes

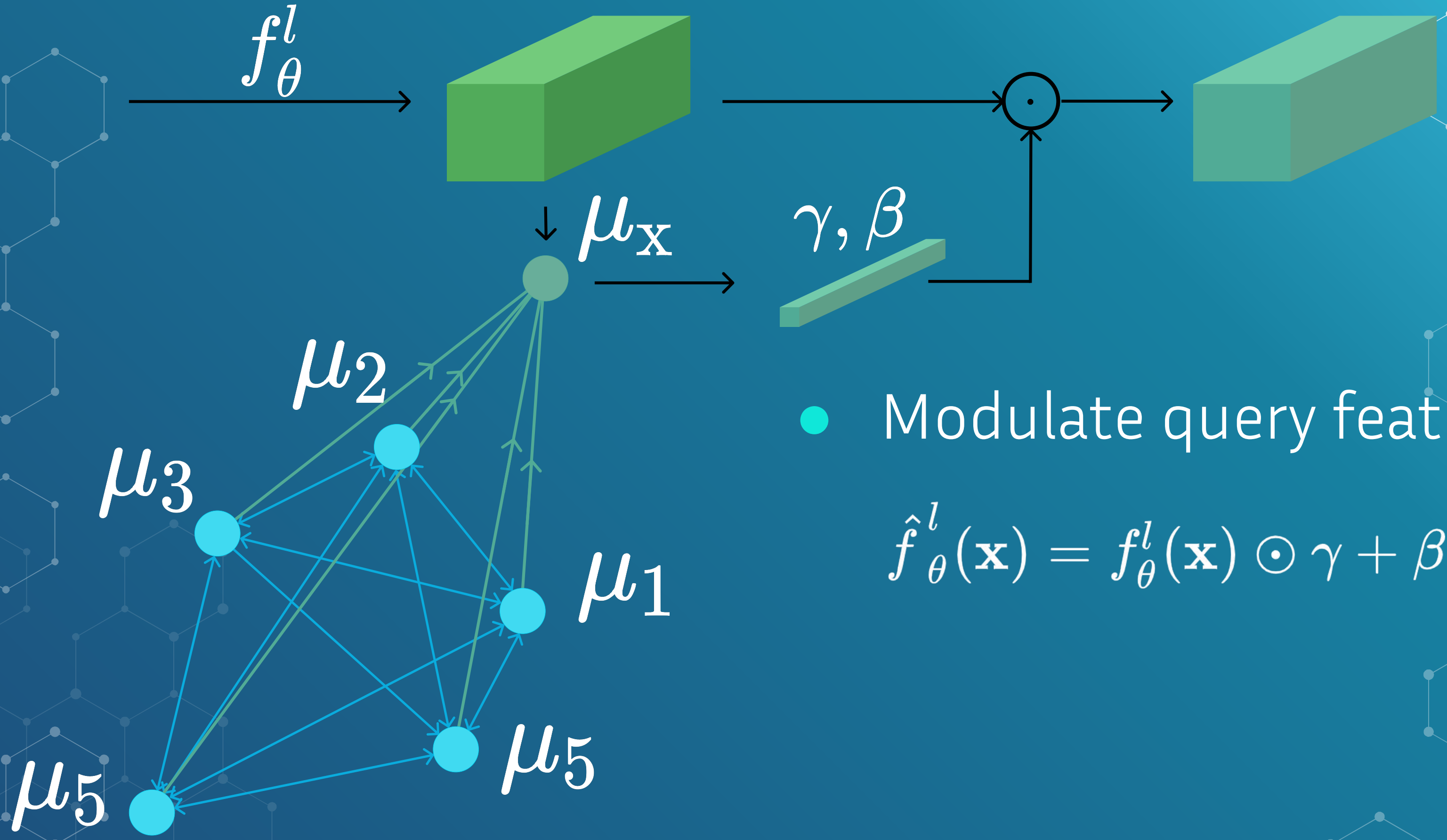
$$\mu_i$$

Our method



- Process the prototypes with GNN

Our method



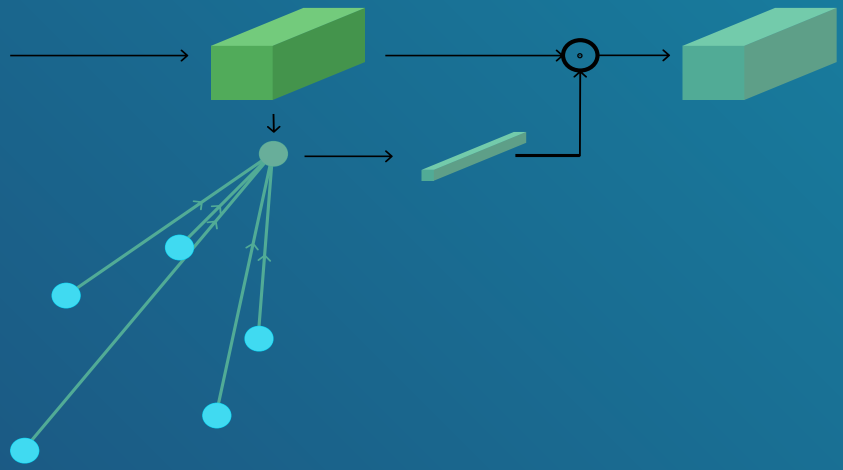
- Modulate query features

$$\hat{f}_\theta^l(\mathbf{x}) = f_\theta^l(\mathbf{x}) \odot \gamma + \beta$$

Our method

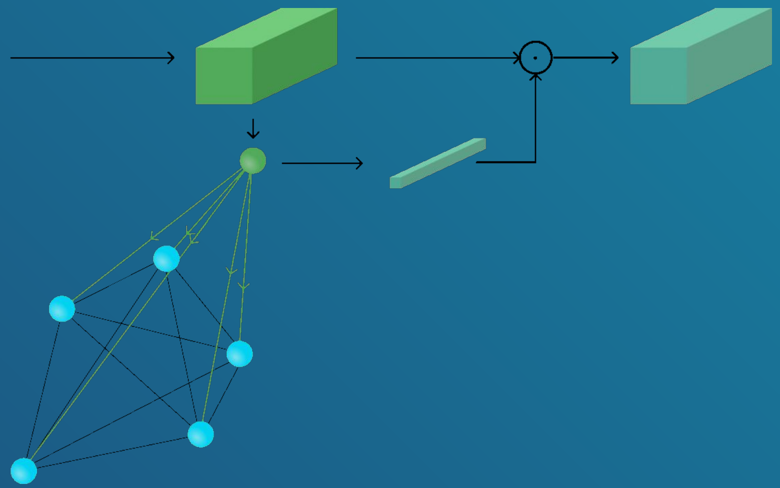
- Support-Attention
  - Send messages from the prototypes of the support set to the current sample

$$\gamma, \beta = \left[\mu_{\mathbf{x}} \middle| \mathrm{softmax}\left(\frac{(\mu_{\mathbf{x}} W_q)(P W_k)^T}{\sqrt{C}} \right) (P W_v) \right] W$$



Our method

- Support-Graph-Attention
  1. support set $\longrightarrow$ $\mu_{\mathbf{x}}$
  2. resulting node $\longrightarrow$ all elements of the support set
  3. update resulting nodes by self attention
  4. generate y and b with the previous mechanism



Results

- Our implementations

Model	ConvNet-4-32	ConvNet-4-128
MAML	47.41	48.29
CAVIA	46.01	49.44
Proto-Nets	49.09	51.33
Our Inner Att	48.04	49.81
Out Inner Graph	46.72	49.20
Our Proto Graph	50.23	52.38

Model	Backbone	5-way 1-shot
Matching Nets [Vinyals et al., 2016]	ConvNet-4-32	43.56 ± 0.84
Proto Nets [Snell et al., 2017]	ConvNet-4-32	48.70 ± 1.84
MAML [Finn et al., 2017]	ConvNet-4-32	48.07 ± 1.75
Cavia [Zintgraf et al., 2018]	ConvNet-4-128 4	49.84 ± 0.68
GNN [Satorras and Estrach, 2018]	64-96-128-256	50.33 ± 0.36
LEO [Rusu et al., 2019]	WRN-28-10	61.76 ± 0.08
SNAIL [Mishra et al., 2018]	ResNet-12	55.71 ± 0.99
MetaOptNet [Lee et al., 2019]	ResNet-12	62.64 ± 0.61
Ours	ConvNet-4-32	50.23
Ours	ConvNet-4-128	52.38

Conclusion

- Proposed an architecture that is biased towards modelling relation between classes with a GNN
- Modulating features with respect to the support set using graph models improves performance

Thank you for watching!