LGM-Net: Learning to Generate Matching Networks for Few-Shot Learning

Huaiyu Li, Weiming Dong, Xing Mei,

Chongyang Ma, Feiyue Huang, Bao-Gang Hu

ICML, 2019

Presenter: Huaiyu Li

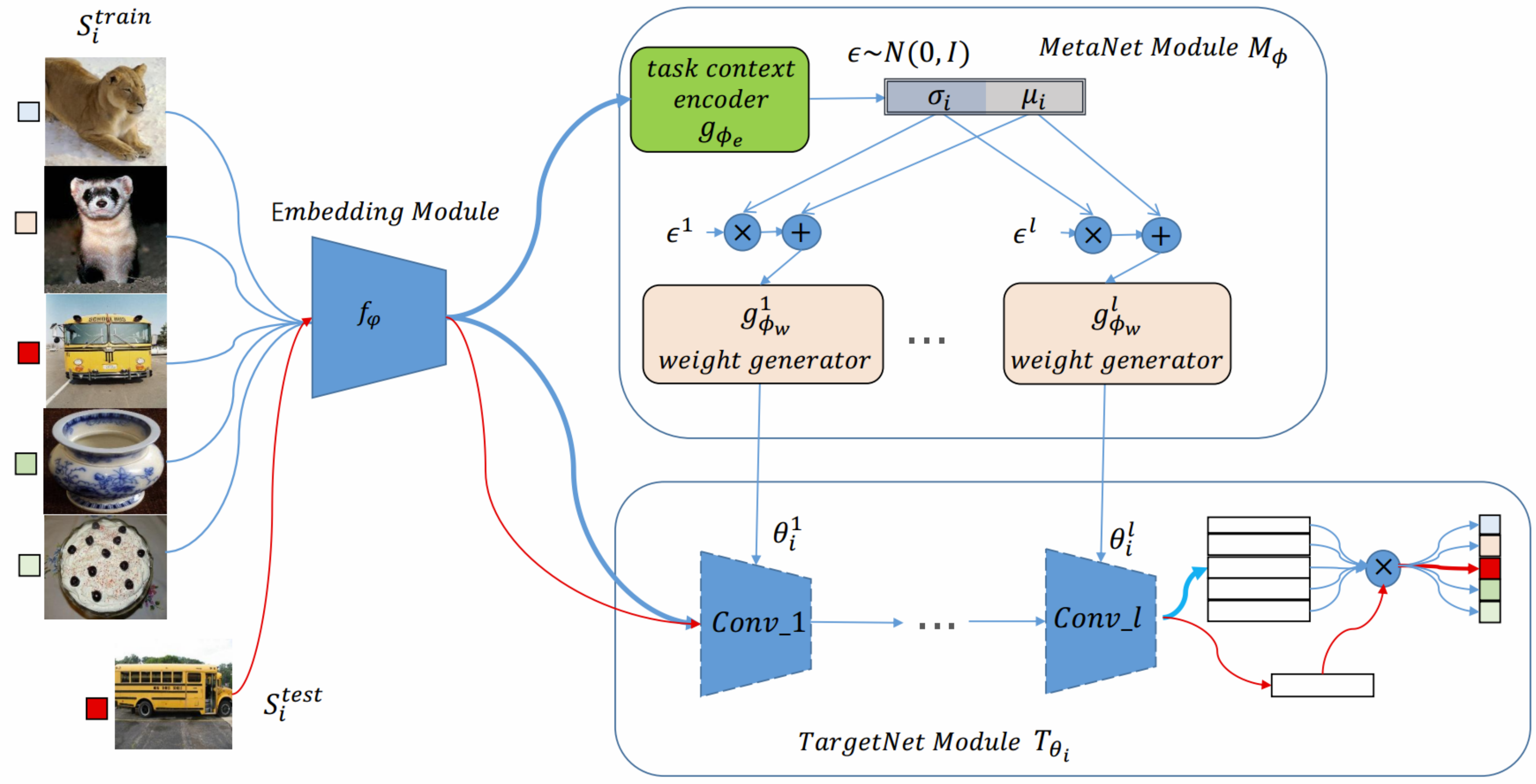
Email: lihuaiyu2014@ia.ac.cn

Motivation

- Training a DNN with SGD algorithm from random initialization
  - Overfitting when training data is scarce
  - Fitting well when training data is sufficient
  - Weights determine DNN functionality
  - Functional weights as a conditional distribution $P(\theta|S^{train})$

- Can we directly obtain functional weights of a DNN for a few-shot learning task ?
  - Let's learn a neural network M to directly generate the weights θ for a neural network T from just a few training samples.
  - e. g. $\theta = M(S^{train})$

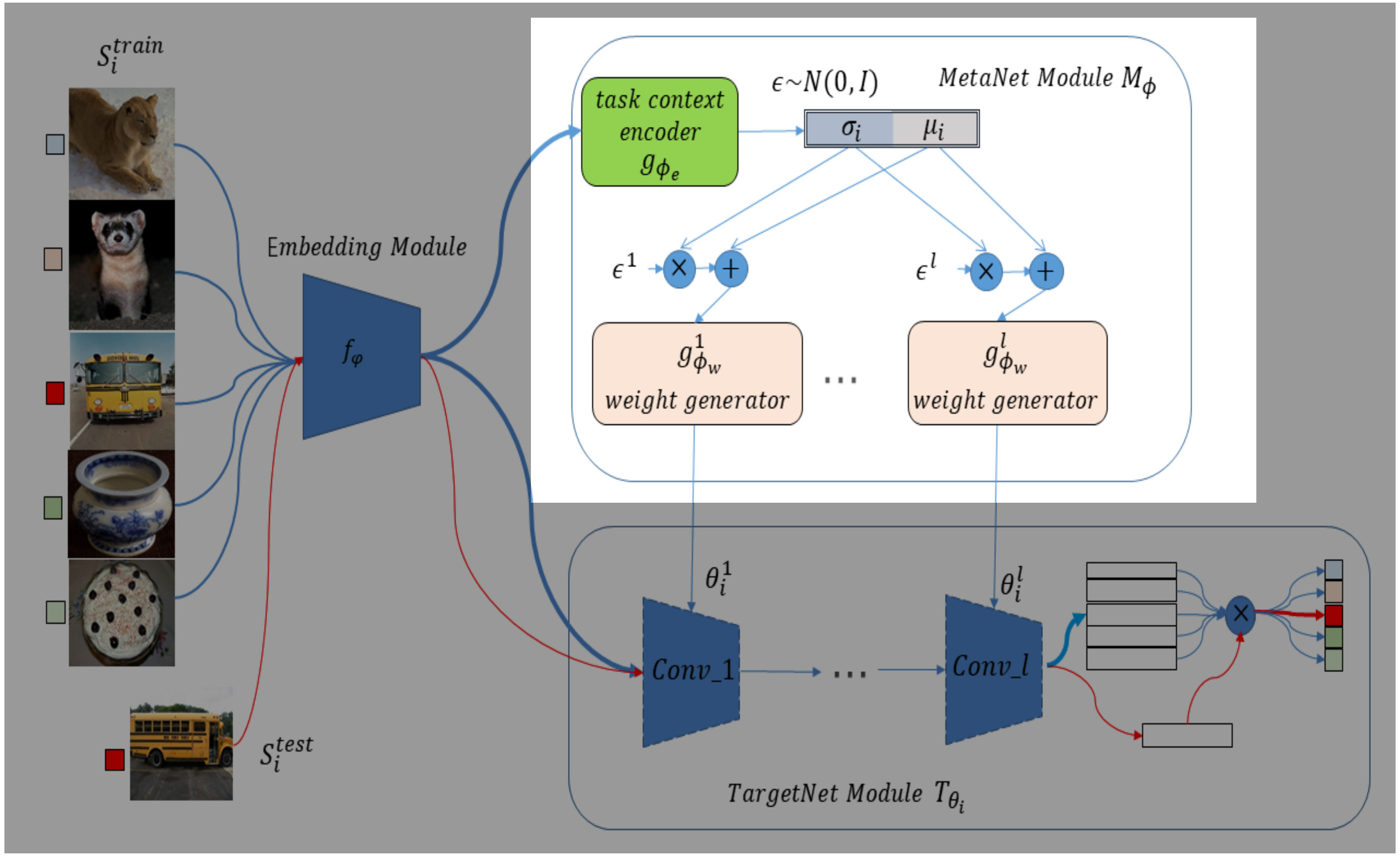
Approach



The architecture of our LGM-Net for few-shot learning on 5-way 1-shot classification problems.

- TargetNet Module(base-learner)
  - A neural network with fixed architecture for classification
- MetaNet Module(meta-learner)
  - Encoding training samples and generating functional weights for TargetNet
- Embedding Module
  - Learnable neural network to extract low dimensional features

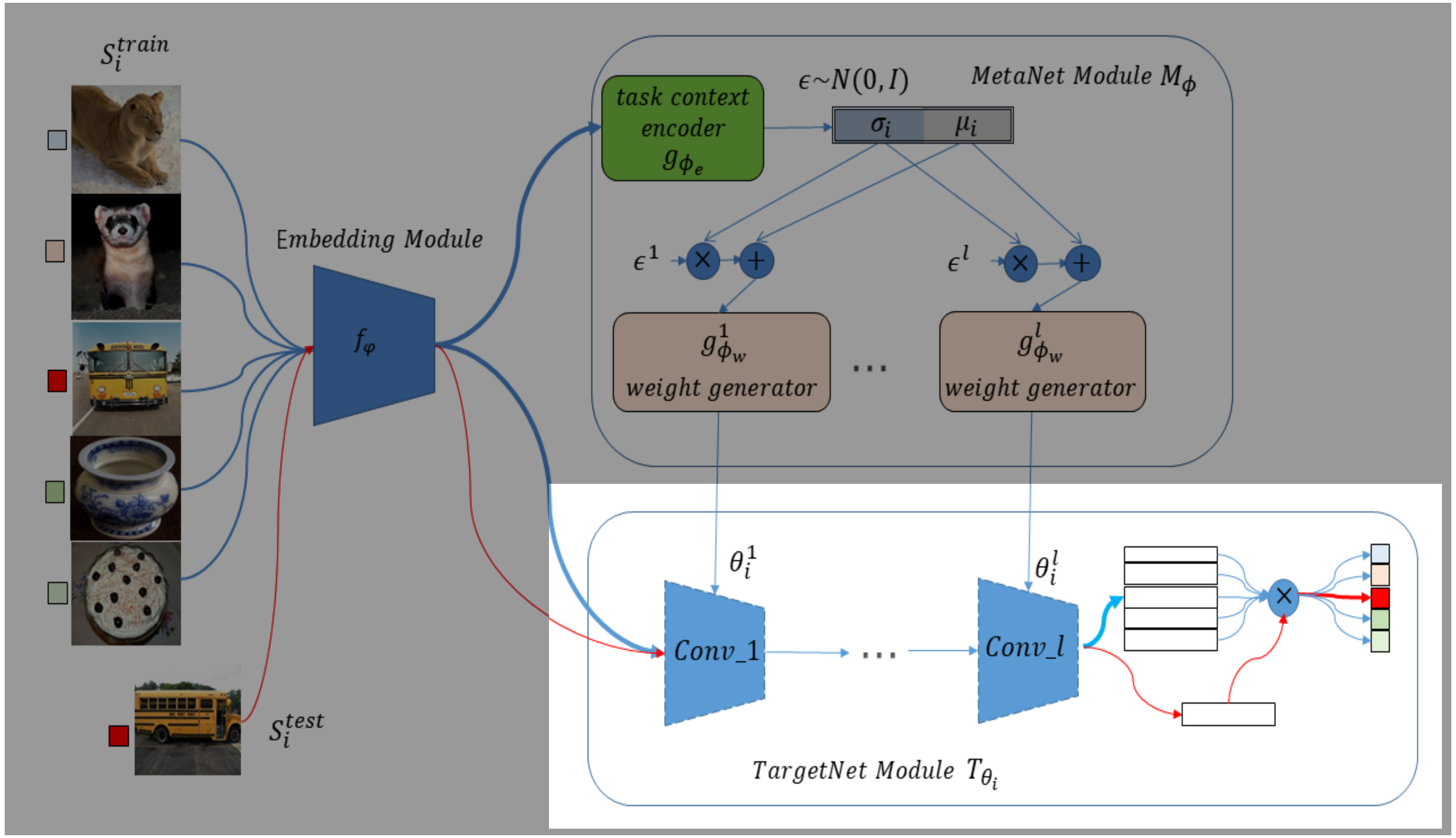
MetaNet Module(meta-learner)



The architecture of our LGM-Net for few-shot learning on 5-way 1-shot classification problems.

- Task context encoder
  - To produce fixed-sized task context features
- Weight generator
  - To produce the weights of TargetNet based on task context features
  - With weight normalization on the generated weights

TargetNet Module(base-learner)



The architecture of our LGM-Net for few-shot learning on 5-way 1-shot classification problems.

- Use matching networks as the computing structure of TargetNet
- The weights of each layer are generated by MetaNet

Learning Algorithm

Algorithm 1 The training algorithm of LGM-Net for N-way K-shot problems

---

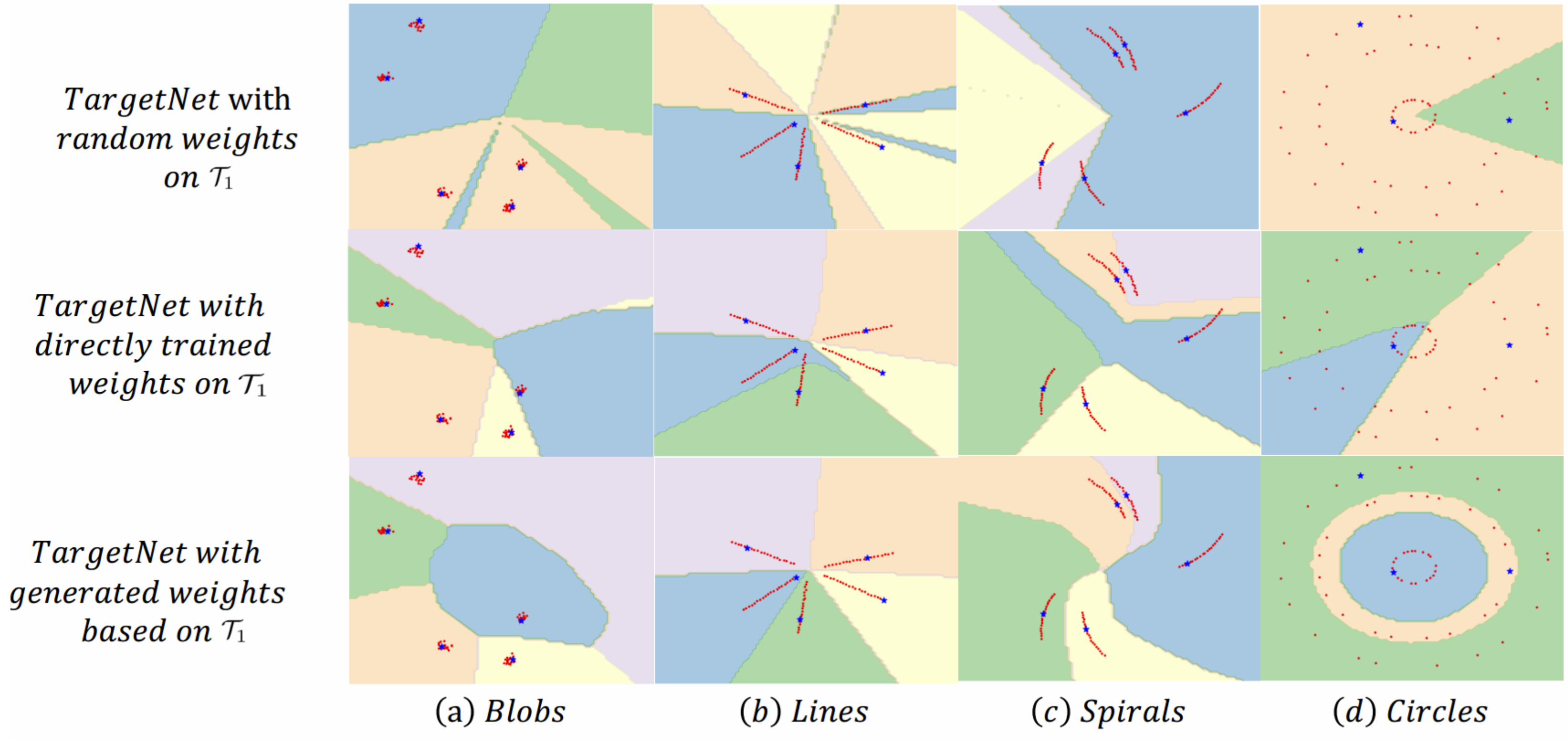
Required: Meta training dataset $D^{meta\text{-}train}$
Required: MetaNet M with parameters ϕ, TargetNet computational structure T with parameter placeholder θ.
Randomly initialize ϕ
while not converged **do**
  Sample a N-way K-shot task batch $\mathcal{T}^{batch}$ from $D^{meta\text{-}train}$
  for all the task instances in a batch **do**
    Divide a task instance as $(S_i^{train}, S_i^{test}) = \mathcal{T}_i$
    Sample a functional weights point $\hat{\theta}$ for TargetNet from $M(S_i^{train})$
    Assign generated weights $\hat{\theta}$ to TargetNet placeholder weights θ
    Compute TargetNet test loss for this task on S_i^{test} as $\mathcal{L}_{\mathcal{T}_i}$
  end for
  Compute batch loss $\mathcal{L}_{\mathcal{T}^{batch}} = \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}$
  Update ϕ using $\nabla_\phi \mathcal{L}_{\mathcal{T}^{batch}}$
end while

- Few-shot classification task episodic training
- Intertask normalization
  - To incorporate information across tasks in a task batch

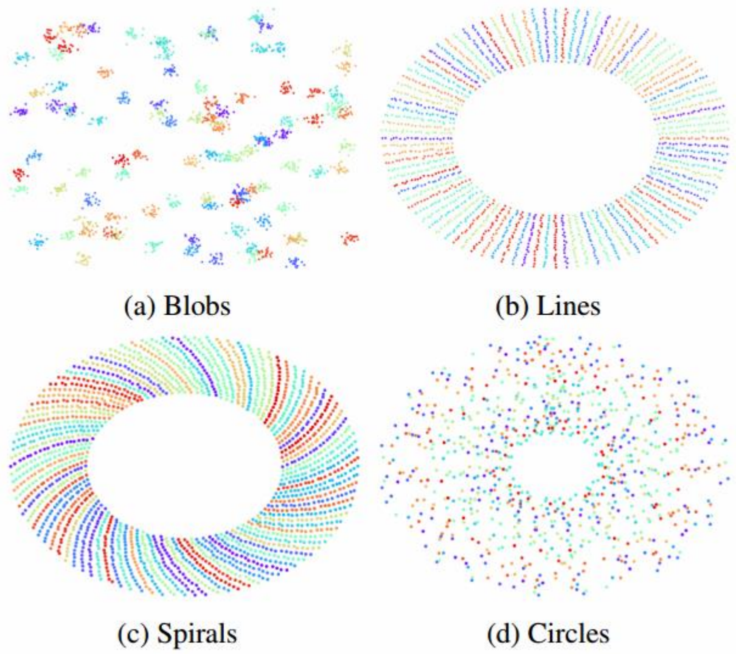
Comparison

- Current meta-learning approaches:
  - Learning an initialization (Finn et al. 2017, ICML)
  - Learning an optimizer (Ravi & Larochelle. 2017, ICLR)
  - Learning a metric mapping function (Vinyals et al. 2016, NIPS)
  - others
- Our approach
  - Learning a conditional weight generator
- Advantages:
  - Neural weights are dynamically adapted to unseen tasks
  - Further fine-tuning is unnecessary

Results on Synthetic Datasets



Comparing the decision boundary of TargetNet with different weights



The weights generated by MetaNet contain prior knowledge for solving unseen tasks.

Evaluation

Table 1. Mean accuracy of our LGM-Net and state-of-the-art methods on Omniglot dataset.

Model	5-way 1-shot	5-way 5-shot	20-way 1-shot	20-way 5-shot
Siamese Net (Koch et al., 2015)	97.3%	98.4%	88.1%	97.0%
Neural Statistician (Harrison Edwards, 2017)	98.1%	99.5%	93.2%	98.1%
Meta Nets (Munkhdalai & Yu, 2017)	99.0%	-	97.0%	-
Prototypical Nets (Snell et al., 2017)	98.8%	99.7%	96.0%	98.9%
MAML (Finn et al., 2017)	98.7%	**99.9%**	95.8%	98.9%
Meta-SGD (Li et al., 2017)	99.5%	**99.9%**	95.9%	99.0%
Relation Net (Sung et al., 2018)	**99.6%**	99.8%	**97.6%**	**99.1%**
Matching networks (Vinyals et al., 2016)	98.1%	98.9%	93.8%	98.5%
LGM-Net (Ours)	99.0%	99.4%	96.5%	98.5%

Table 2. Mean accuracy ± 95% confidence intervals of our LGM-Net and state-of-the-art methods on *mini*ImageNet dataset.

Model	5-way 1-shot	5-way 5-shot	20-way 1-shot
Matching networks (Vinyals et al., 2016)	43.56±0.84%	55.31±0.73%	17.31±0.22%
Meta-LSTM (Ravi & Larochelle, 2017)	43.44±0.77%	60.60±0.71%	16.70±0.23%
MetaNet (Munkhdalai & Yu, 2017)	49.21±0.96%	-	-
Prototypical Nets (Snell et al., 2017)	49.42±0.78%	68.20±0.66%	
MAML (Finn et al., 2017)	48.70±1.84%	63.11±0.92%	16.49±0.58%
Meta-SGD (Li et al., 2017)	50.47±1.87%	64.03±0.94%	17.56±0.64%
Relation Net (Sung et al., 2018)	51.38±0.82%	67.07±0.69%	-
REPTILE (Nichol & Schulman, 2018)	49.97±0.32%	65.99 ± 0.58%	-
SNAIL (Mishra et al., 2018)	55.71±0.99%	65.99 ± 0.58%	-
(Gidaris & Komodakis, 2018)	56.20±0.86%	73.00 ± 0.64%	-
LEO(Rusu et al., 2019)	61.76±0.08%	**77.59± 0.12%**	-
LGM-Net (Ours)	**69.13±0.35%**	**71.18±0.68%**	**26.14±0.34%**

- Competitive performance on Omniglot
- STOA 1-shot learning performance on mini-ImageNet
- Ablation Study
  - Task context encoder and intertask normalization are important.

At the poster:

additional details, experiments and discussions

[Tue Jun 11th 06:30—09:00 PM @Pacific Ballroom#10]

Thanks!