



Bridging Multi-Task Learning and Meta-Learning: *Towards Efficient Training and Effective Adaptation*



Haoxiang Wang
PhD Candidate
ECE, UIUC

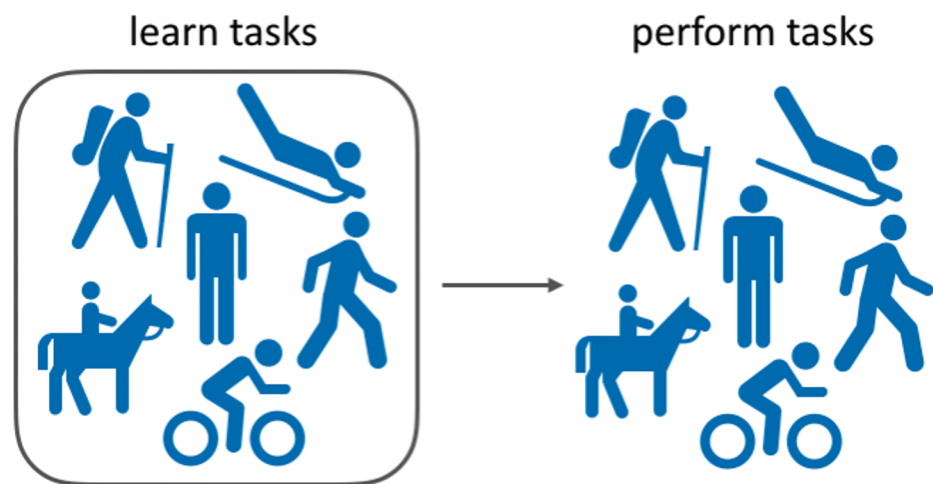


Han Zhao
Assistant Professor
Computer Science, UIUC



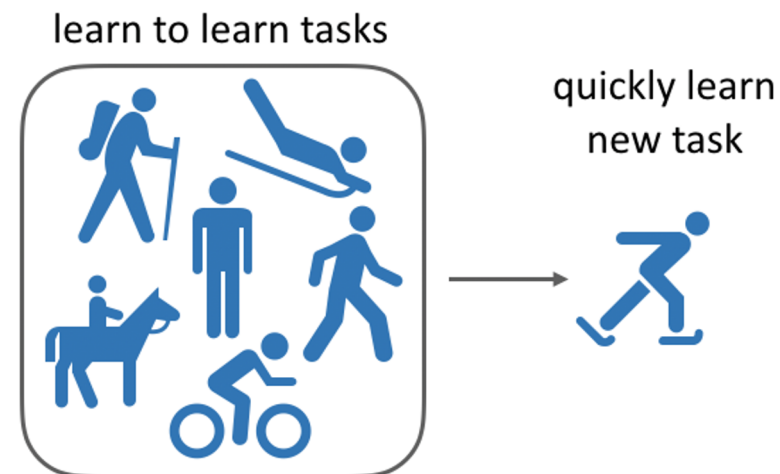
Bo Li
Assistant Professor
Computer Science, UIUC

Multi-Task Learning (MTL)



Setting: Test task = Training tasks
Goal: Be a master on a set of tasks

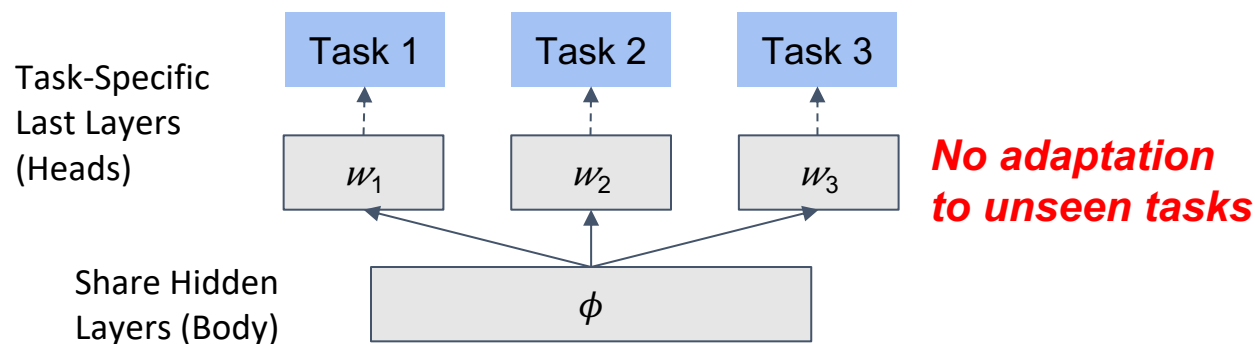
Meta-Learning



Setting: Test task \neq Training tasks
Goal: Adapt to an unseen task quickly.
Assumption: The test task has some shared knowledge (i.e., meta-knowledge) with the training tasks.

Multi-Task Learning (MTL)

Multi-Head Structure



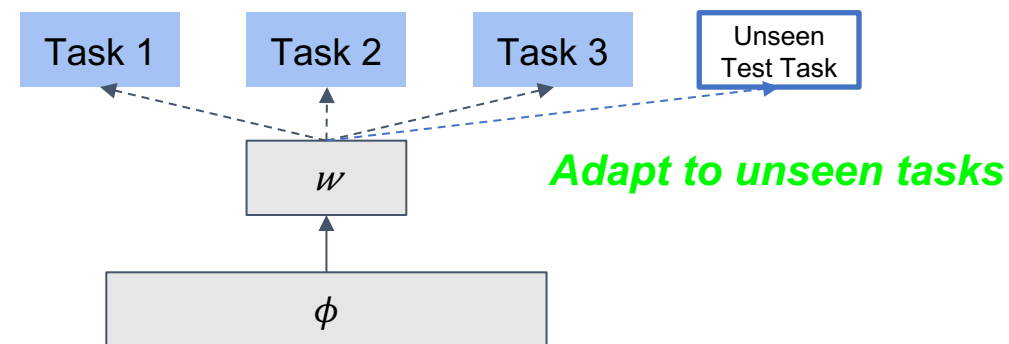
Training Objective:

1st-order optimization (a form of Empirical Risk Minimization)

→ **Efficient Training**

Gradient-Based Meta-Learning (GBML)

Single-Head Structure



Training Objective:

2nd order optimization (e.g., MAML, MetaOptNet, ANIL, iMAML)

→ **Expensive Training**

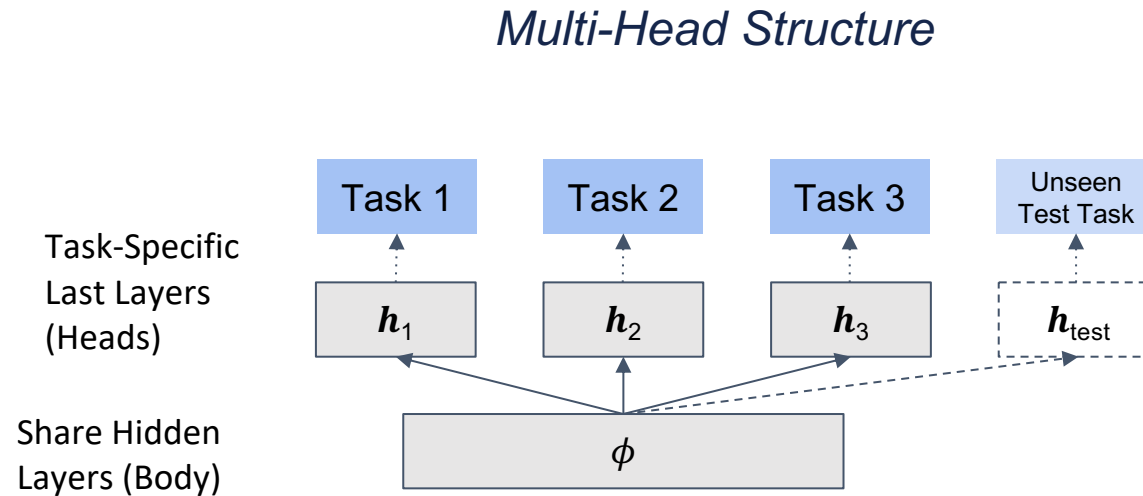
Motivation:

*Can we combine the best of both worlds from multi-task learning and meta-learning, i.e., **effective adaptation to unseen tasks** with **efficient training**?*

Our answer: **Yes!**

Contribution:

Our paper bridges *Multi-Task Learning* (MTL) and *Gradient-Based Meta-Learning* (GBML) by theoretical and empirical studies.



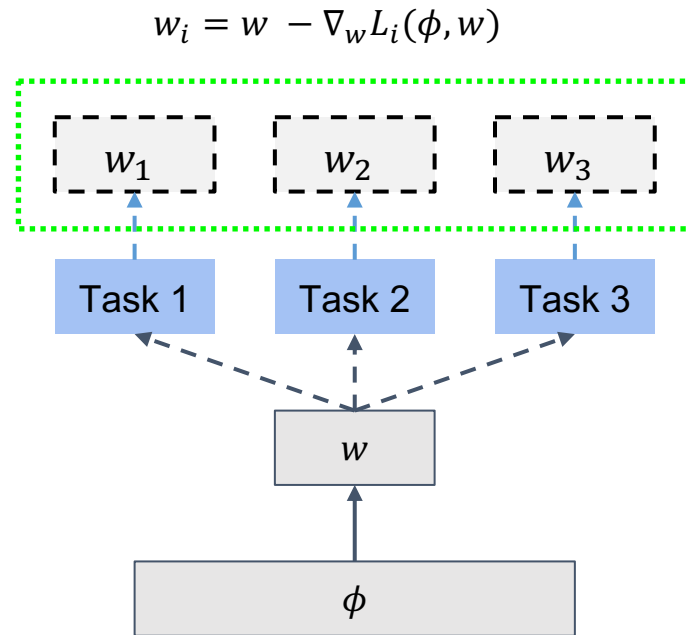
Fine-tuning: For a trained MTL model, we can adapt it to an unseen test task by

1. Randomly initialize a new head
2. Fine-tune the head on a few labelled data of the test task
3. Use the fine-tuned head for predictions on the new task

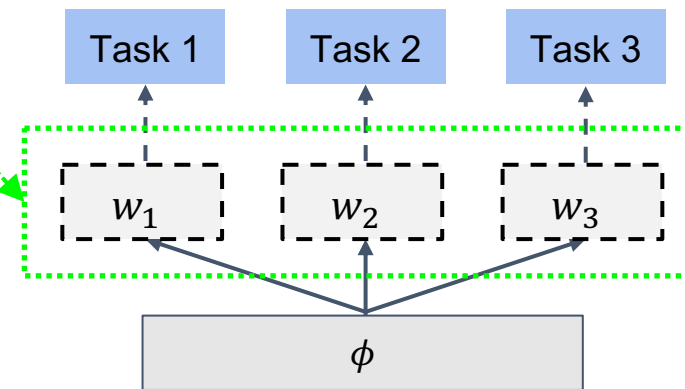
Gradient-Based Meta-Learning: Similarity to Multi-Task Learning



Step I: Obtain task-specific *transient heads* by gradient descent on each task



Step II: Use the *transient heads* to give predictions on each task.



Inner-Loop Optimized Layers	Early Stopping	l_2 Regularizer
Last Layer	ANIL (Raghu et al., 2020)	MetaOptNet (Lee et al., 2019b) R2D2 (Bertinetto et al., 2019)
All Layers	MAML (Finn et al., 2017)	iMAML (Rajeswaran et al., 2019) Meta-MinibatchProx (Zhou et al., 2019b)

Inner-Loop Optimized Layers	Early Stopping	ℓ_2 Regularizer
Last Layer	ANIL (Raghu et al., 2020)	MetaOptNet (Lee et al., 2019b) R2D2 (Bertinetto et al., 2019)
All Layers	MAML (Finn et al., 2017)	iMAML (Rajeswaran et al., 2019) Meta-MinibatchProx (Zhou et al., 2019b)

Equivalence: MTL and a **class of GBML** shares the **same optimization objective**

Difference: MTL uses **joint training**, while GBML adopts **bi-level optimization with regularization**

Closeness in the function space:

- We compare neural nets trained by ANIL (a MAML simplification) and MTL in an NTK-based meta-learning framework [1]
- We prove that, on any test task, the difference between predictions is upper bounded as

$$\|\text{ANIL prediction} - \text{MTL prediction}\|_2 \leq \mathcal{O}\left(\lambda\tau + \frac{1}{L}\right)$$

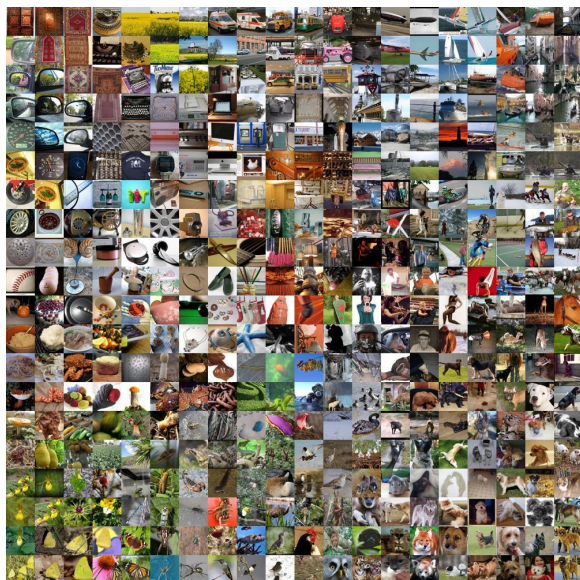
↖ λ, τ : Learning rates ↖ L : Network depth

Benchmarks: 4 popular few-shot learning datasets, extracted from ImageNet and CIFAR-100.

mini-ImageNet tiered-ImageNet



ImageNet

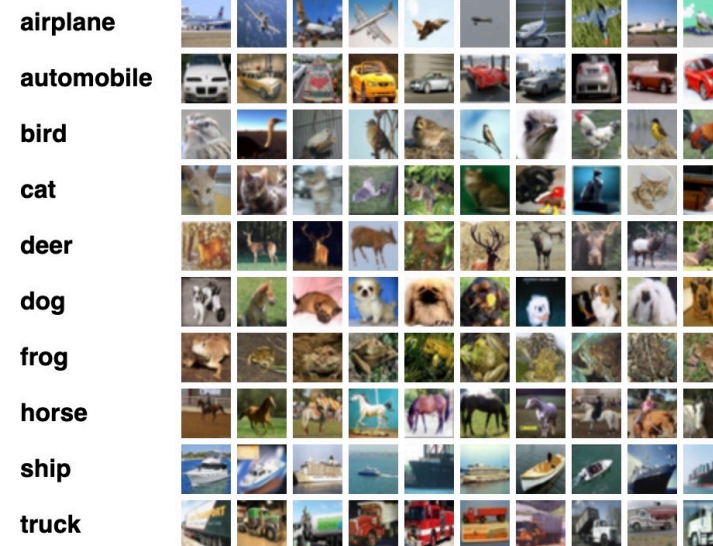


CIFAR-FS

FC-100



CIFAR-100

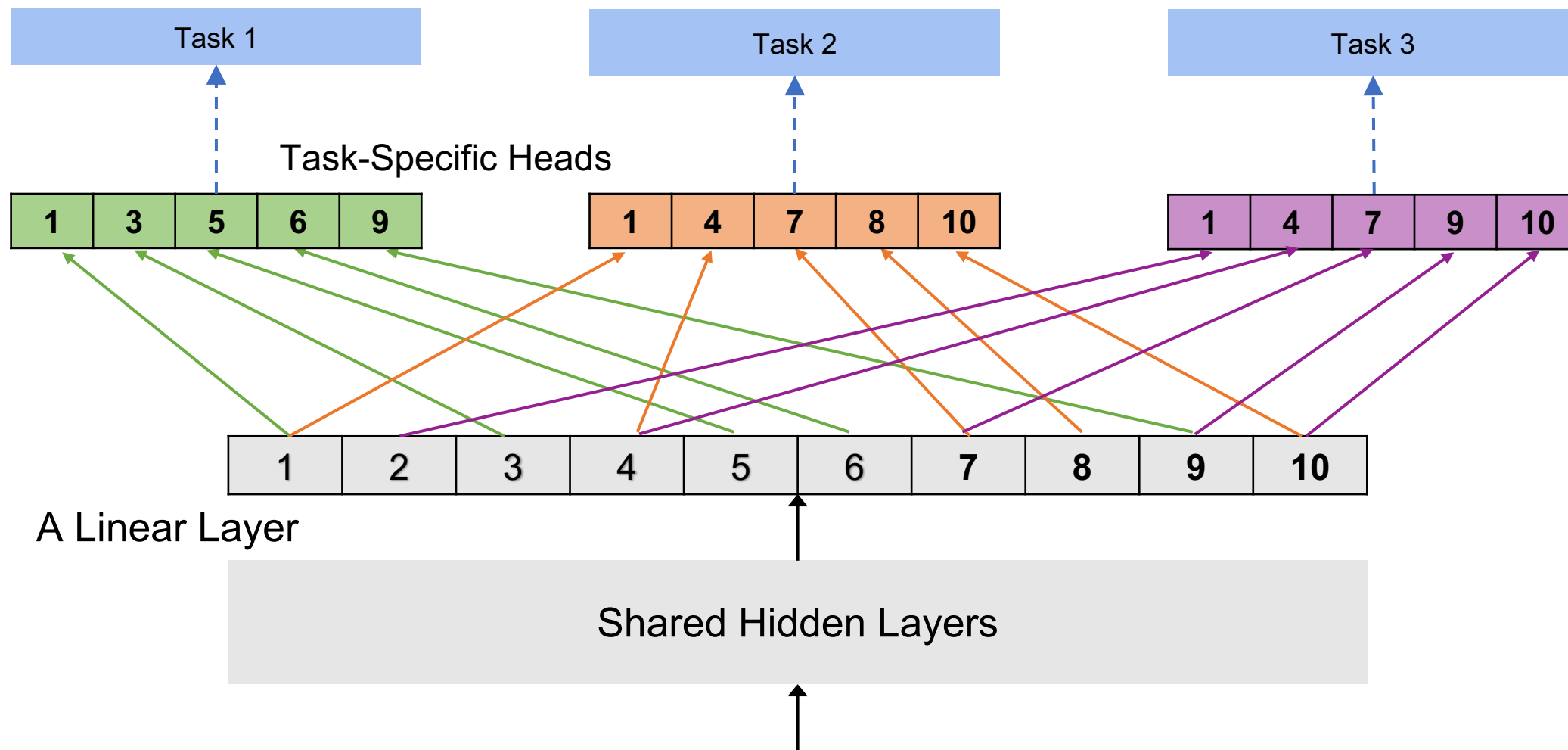


Remarks: The number of unique training tasks is quite large (due to combinatorial explosion), e.g., it's 4.3 billions for tiered-ImageNet. Thus, we cannot afford an individual head for each training task.

Memory-Efficiency Implementation of MTL Heads



Example: 5-way few-shot classification; Each task has 5 task-specific classes drawn from 10 base classes.



MetaOptNet: A state-of-the-art gradient-based meta-learning algorithm.

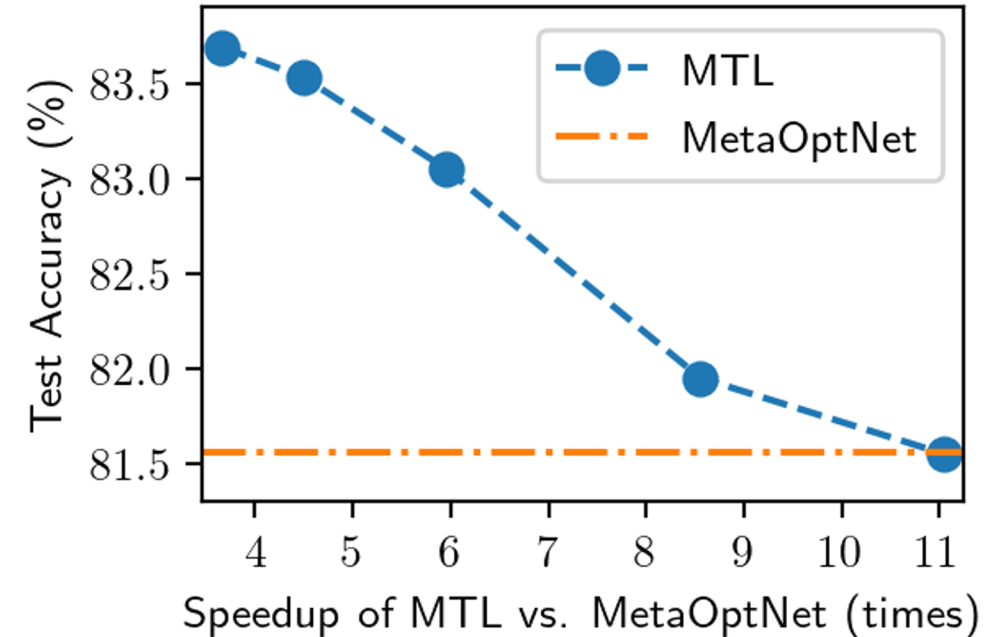
MTL-ours: Our memory-efficient implementation of multi-task learning.

Algorithm	Architecture	mini-ImageNet		tiered-ImageNet		CIFAR-FS		FC100	
		1-shot (%)	5-shot (%)	1-shot (%)	5-shot (%)	1-shot (%)	5-shot (%)	1-shot (%)	5-shot (%)
MAML [Finn et al., 2017a]	CNN-4	48.70 ± 1.84	63.11 ± 0.92						
MetaOptNet [Lee et al., 2019]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53	72.0 ± 0.7	84.2 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
MTL-ours [Wang et al., 2021]	ResNet-12	59.84 ± 0.22	77.72 ± 0.09	67.11 ± 0.12	83.69 ± 0.02	69.5 ± 0.3	84.1 ± 0.1	42.4 ± 0.2	57.7 ± 0.3

Multi-task learning can match the SOTA of gradient-based meta-learning on few-shot learning benchmarks!

	Test Accuracy	GPU Hours
MetaOptNet	78.63%	85.6 hrs
MTL	77.72%	3.7 hrs

Mini-ImageNet (5-way 5 shot)



tiered-ImageNet (5-way 5 shot)

Multi-task learning can be more than 10x faster, since it does not use any 2nd order optimization.

Takeaway: *We can combine the benefits of multi-task learning and meta-learning, i.e., **effective adaptation to unseen tasks** with **efficient training**.*

Code: <https://github.com/Al-secure/multi-task-learning>

Contact Information:

- Haoxiang Wang: hwang264@illinois.edu
- Han Zhao: hanzhao@illinois.edu
- Bo Li: lbo@illinois.edu