JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Meta Omnium: A Benchmark for General-Purpose Learning-to-Learn

Ondrej Bohdal[1,*], Yinbing Tian[1,2,*], Yongshuo Zong[1], Ruchika Chavhan[1], Da Li[3], Henry Gouk[1], Li Guo[2], Timothy Hospedales[1,3]

Session and poster ID: TUE-PM-341

THE UNIVERSITY of EDINBURGH informatics 1

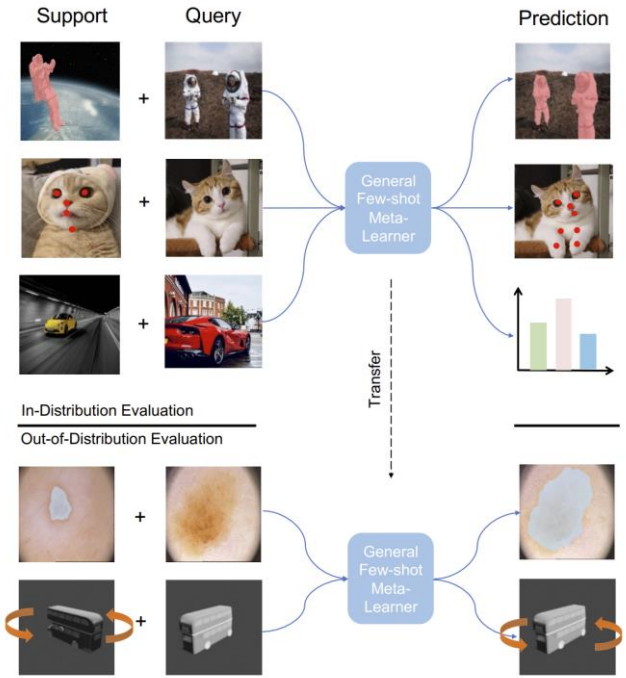
2

Samsung Research 3

* Joint-first authors

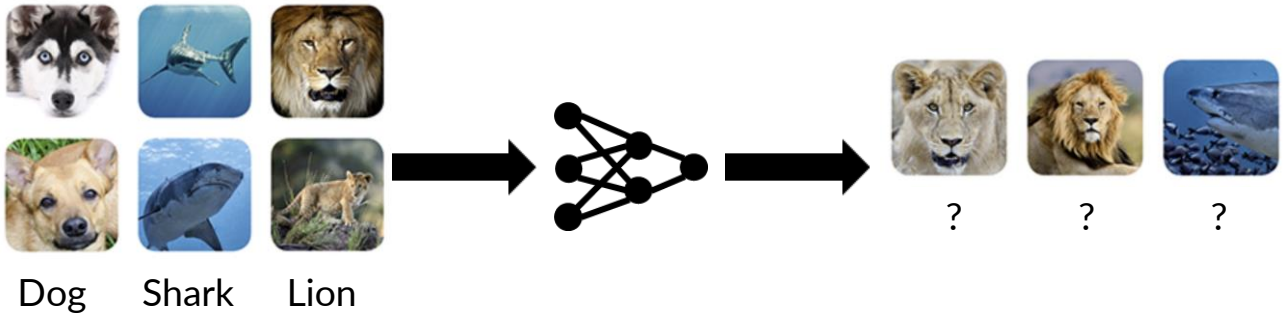
Meta Omnium Overview

- Multi-task few-shot learning benchmark for evaluating generalization across diverse computer vision task types
- Challenging yet lightweight
- Clear hyperparameter tuning and model selection protocol
  - Fair comparison across current and future few-shot learning algorithms
- Includes multi-task extensions of the most popular few-shot learning approaches
  - Analysis of their ability to generalize across tasks and to transfer knowledge between them



Few-Shot Learning

- Learn a new concept from only a small number of examples
  - E.g. learn to classify images into three new classes after seeing two examples of each



- Currently done separately for classification, segmentation, regression, …
- *Can we meta-learn one general-purpose model that can adapt to all of them?*

Comparison of Existing Benchmarks

Dataset	Num Tasks	Num Domains	Num Imgs	Categories	Size	Lightweight	Multi-Task	Multi-Domain
Omniglot [30]	1	1	32K	1623	148MB	✓	✗	✗
miniImageNet [66]	1	1	60K	100	1GB	✓	✗	✗
Meta-Dataset [63]	1	7∼10	53M	43∼1500	210GB	✗	✗	✓
VTAB [80]	1	3∼19	2.2M	2∼397	100GB	✗	✗	✓
FSS1000 [37]	1	1	10000	1000	670MB	✓	✗	✗
Meta-Album [65]	1	10∼40	1.5M	19∼706	15GB	✓	✗	✓
Meta Omnium	4	21	160K	2∼706	3.1GB	✓	✓	✓

As part of Meta Omnium:

- Task types:
  - Classification, Segmentation, Keypoint estimation, Regression
- Domains:
  - Birds, cars, microscopy, remote sensing, natural images, medical images, synthetic images, ...

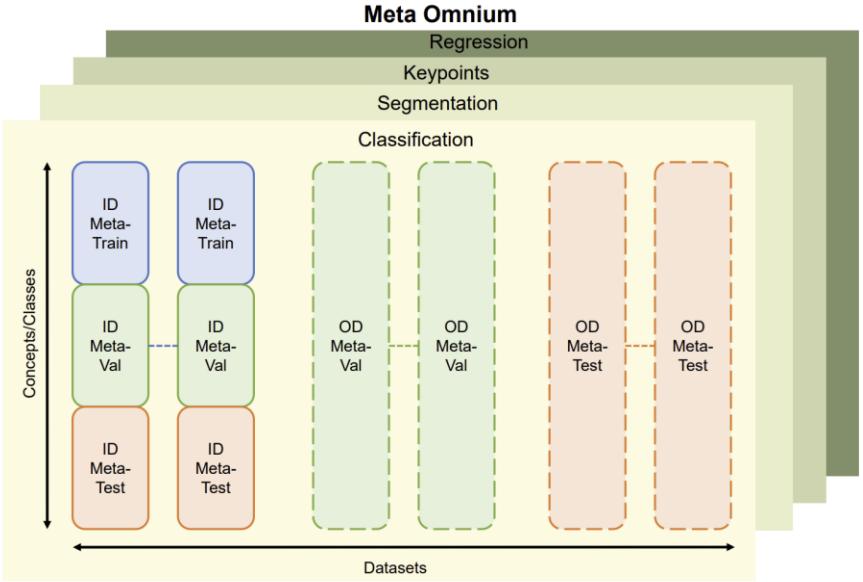
Benchmark Structure

Scenarios:

- Single/multi-task meta-learning
- In/out-distribution generalization

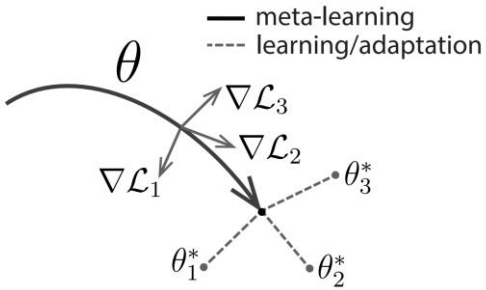
Task types:

- Classification
- Segmentation
- Keypoint estimation
- Regression - out-of-task

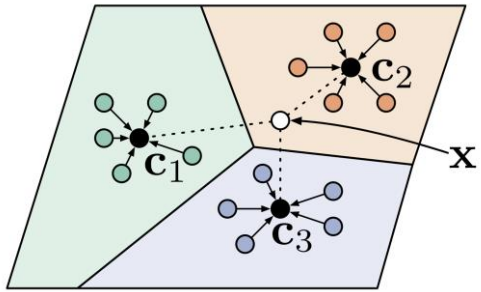


Approaches

- Meta-learning:
  - MAML, Proto-MAML, Meta-Curvature, ProtoNet, deep differentiable ridge-regression (DDRR)
- Baselines:
  - Fine-tuning of prototypes, fine-tuning, linear readout (all transfer learning), training from scratch
- Extended to support various task types
  - E.g. ProtoNet acts as a simple Gaussian kernel-regression model for certain tasks



MAML



ProtoNet

Hyperparameter Optimization (HPO)

- Enable fair comparison between methods
- Separately for single-task and multi-task scenarios
- Multi-Objective TPE method from the Optuna library
  - Sample efficient method - 30 candidates are sampled (lightweight HPO)
- Hyperparameters include the meta-learning rate and optimizer, momentum, and various method-specific hyperparameters

Main Findings

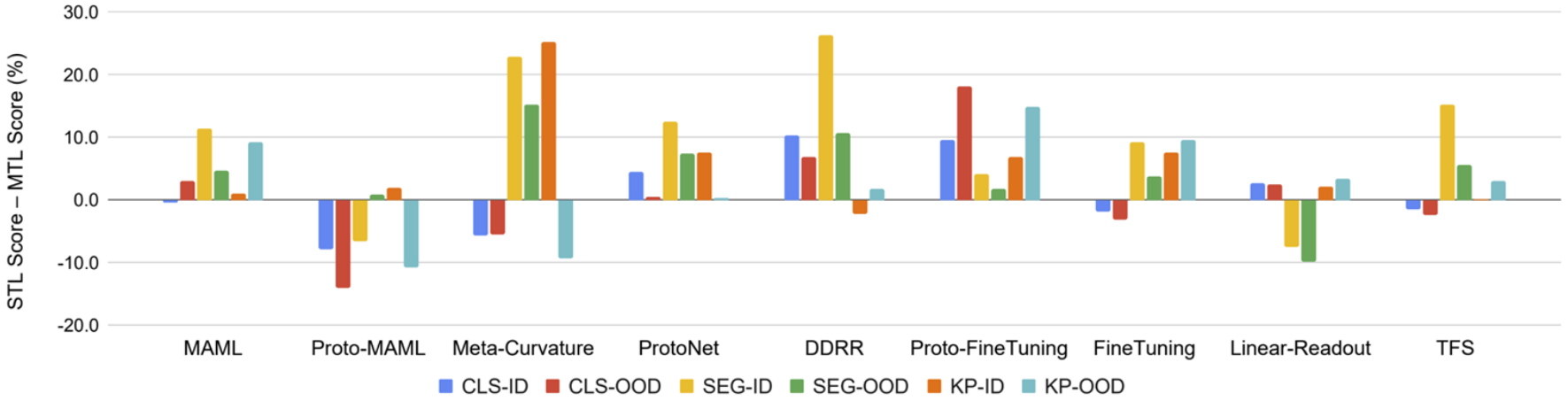
- Single-task:
  - ProtoNet is the most versatile meta-learner
  - ProtoNet is also the most robust to out-of-distribution episodes
- Multi-task:
  - ProtoNet and ProtoMAML are the best in this more challenging setting
- Best meta-learners are clearly better than transfer learning for both single and multi-task scenarios

		Classification		**Segmentation**		**Keypoints**		**Average Rank**		
		ID	OOD	ID	OOD	ID	OOD	ID	OOD	AVG
Single-Task	MAML	58.7	61.6	54.7	42.1	25.4	33.0	4.3	3.3	3.8
	Proto-MAML	50.5	49.7	46.4	44.1	23.6	22.5	6.0	6.3	6.2
	Meta-Curvature	64.8	61.4	65.6	49.8	43.5	16.0	2.0	4.3	3.2
	ProtoNet	70.4	59.4	75.8	57.2	27.8	33.3	**1.3**	**1.7**	**1.5**
	DDRR	63.1	58.7	66.7	48.0	20.5	31.9	4.7	3.7	4.2
	Proto-FineTuning	50.8	50.7	60.0	43.4	21.3	33.1	5.3	4.3	4.8
	FineTuning	42.3	48.2	50.5	40.0	25.7	30.0	5.7	6.7	6.2
	Linear-Readout	48.6	53.4	34.0	22.7	22.1	26.9	7.3	6.7	7.0
	TFS	31.5	42.0	42.8	37.6	21.0	26.0	8.3	8.0	8.2
Multi-Task	MAML	59.1	58.5	43.3	37.4	24.3	23.9	**2.7**	4.7	3.7
	Proto-MAML	58.5	63.7	53.0	43.2	21.6	33.3	3.0	**1.7**	**2.3**
	Meta-Curvature	70.4	66.9	42.6	34.5	18.2	25.3	4.3	4.7	4.5
	ProtoNet	65.9	58.8	63.3	49.7	20.1	33.0	**2.7**	2.0	**2.3**
	DDRR	52.8	51.9	40.4	37.3	22.8	30.1	5.0	4.7	4.8
	Proto-FineTuning	52.4	53.2	44.8	37.8	21.2	30.0	4.3	4.0	4.2
	FineTuning	44.1	51.2	41.3	36.1	18.1	20.5	7.7	7.0	7.3
	Linear-Readout	46.0	50.9	41.5	32.6	19.9	23.5	6.3	8.0	7.2
	TFS	21.9	23.8	38.7	35.8	14.1	11.0	9.0	8.3	8.7

Average performance across the datasets within each task type. Larger value is better.

Analysis of Single-Task vs Multi-Task Learning

- Single-task learning (STL) outperforms the multi-task learning (MTL) condition
- The difficulty of learning from heterogeneous tasks outweighs the benefit of the extra available multi-task data



Analysis of the differences in scores between STL and MTL for different methods.

Generalization to Held-Out Task Types

- Meta-learners MAML, ProtoNets and DDRR perform the best, while training from scratch (TFS) performs the worst
  - Suggests meta-learners can generalize even to new task types
- In several cases the results were not better than predicting the mean, so learning-to-learn of completely new task families is an open challenge

MAML	PMAML	MC	PN	DDRR	PFT	FT	LR	TFS
3.3	6.5	4.8	3.5	3.5	3.8	5.8	4.3	8.5

Average ranking of the different methods
across four out-of-task regression datasets.

Does External Pre-Training Help?

- Initialize from ImageNet1k pre-trained model before meta-training
- Pre-training is not necessarily helpful in the considered multi-task setting

Method	Pretrain	Cls.		Segm.		Keyp.	
		ID	OOD	ID	OOD	ID	OOD
Proto-MAML	✗	58.5	63.7	53.0	43.2	21.6	33.3
ProtoNet	✗	66.0	58.8	63.3	49.7	20.1	33.0
Proto-MAML	✓	63.9	62.7	56.2	45.3	21.8	33.3
ProtoNet	✓	63.5	58.6	62.0	49.0	20.1	33.1

Impact of external pre-training in multi-task few-shot learning.

Analysis of Runtimes

- Despite the ambitious goal of our benchmark, experiments on Meta Omnium are lightweight
- ProtoNet is the fastest approach, as well as the best-performing one
- Fine-tuning and training from scratch are expensive during the test time as they use backpropagation with a larger number of steps

Method	Train Time	Val Time	Test Time	Total Time
MAML	1.8h	1.9h	0.9h	5.0h
Proto-MAML	1.9h	1.9h	0.9h	5.1h
Meta-Curvature	3.4h	2.6h	1.3h	7.6h
ProtoNet	0.8h	0.4h	0.2h	1.8h
DDRR	1.4h	0.6h	0.3h	2.7h
Proto-FineTuning	1.7h	4.5h	2.3h	8.9h
FineTuning	1.5h	8.1h	4.9h	14.9h
Linear-Readout	1.2h	5.1h	2.8h	9.6h
TFS	0.0h	0.8h	6.2h	7.0h

Analysis of times needed by different algorithms in the multi-task setting (using one NVIDIA 1080 Ti GPU and 4 CPUs).

Example Uses of Meta Omnium

- Developing new multi-task few-shot learning approaches
- Studying multi-task optimization in meta-learning
- Studying hyperparameter optimization for meta-learning
- Developing validation strategies in meta-learning
  - Using in-domain vs out-domain validation sets
- Studying the benefit of task-specific decoders and external data

Conclusion

- First multi-task few-shot meta-learning benchmark for computer vision
- Challenging in multiple highly topical ways:
  - Learning on heterogeneous task distributions
  - Evaluating generalization to out-of-distribution datasets
  - Learning-to-learn and transfer knowledge across tasks with heterogeneous output spaces
- Lightweight enough to be of broad interest and use for driving future research
- Project website: https://edi-meta-learning.github.io/meta-omnium/
  - Includes links to the paper, code and data