CVPR 2023 Few-Shot Learning Tutorial
Part II: Meta-Learning

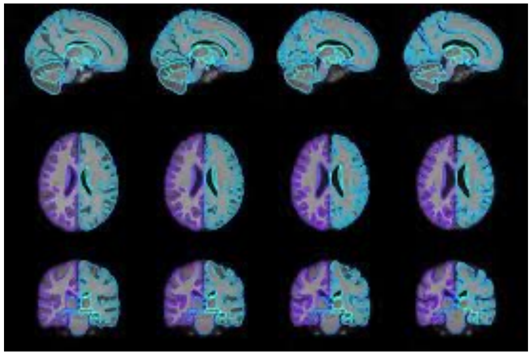
Timothy Hospedales

University of Edinburgh, UK

Samsung AI Center, Cambridge, UK

Why few-shot learning?

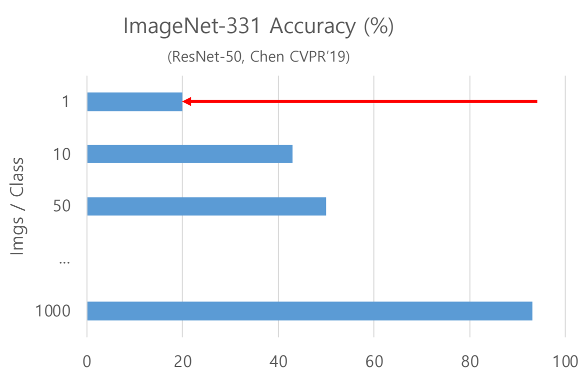
Expensive to Annotate Data
(e.g., medical)



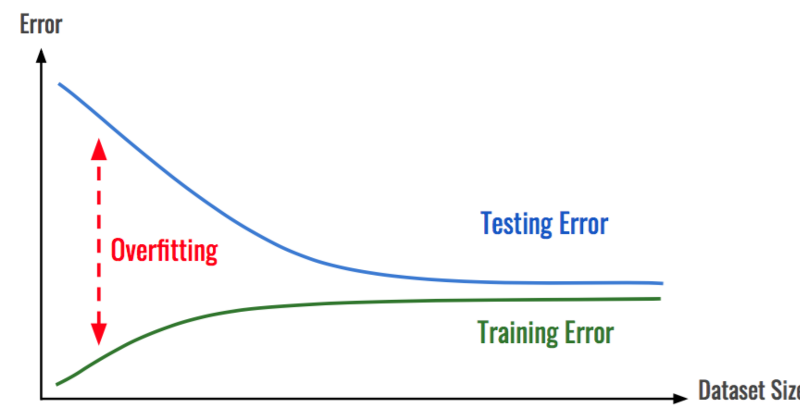
Emerging Categories (e.g., New brands or products)

Rare Concepts (e.g., Endangered species)

Why is FSL Hard?



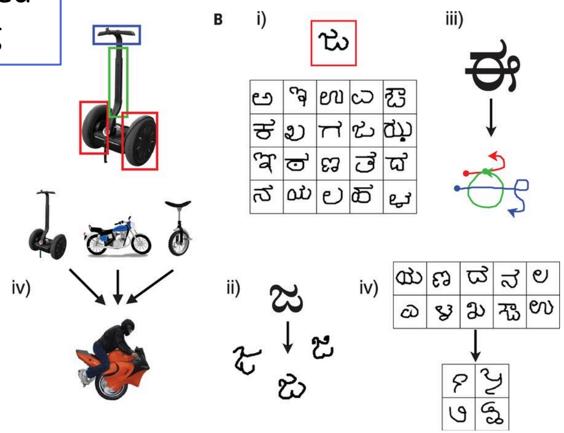
Performance drops dramatically in low data regime



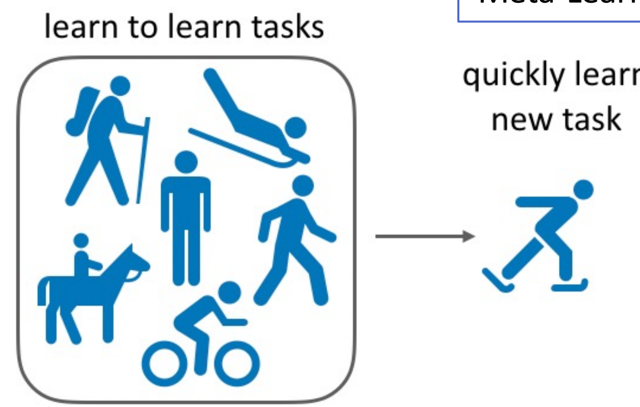
.... thanks to overfitting.

Solutions to FSL all involve borrowing related data from elsewhere….

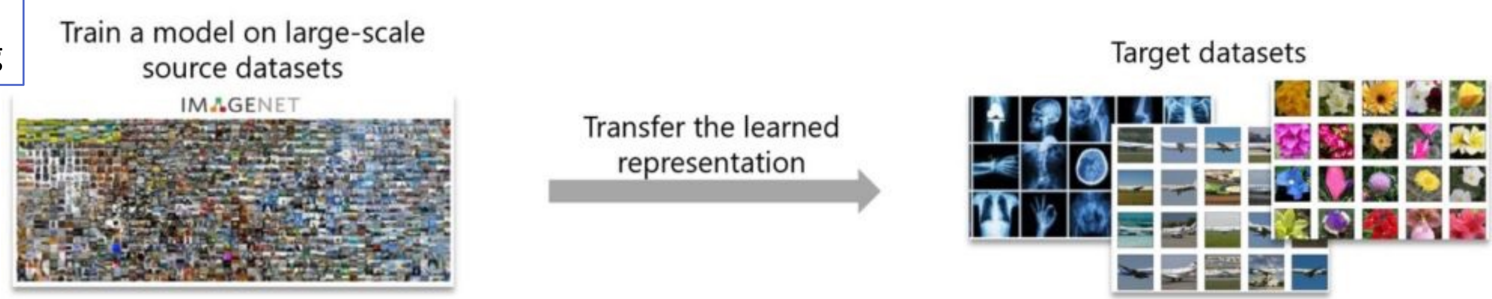
Part Based Learning



Meta-Learning



Transfer Learning

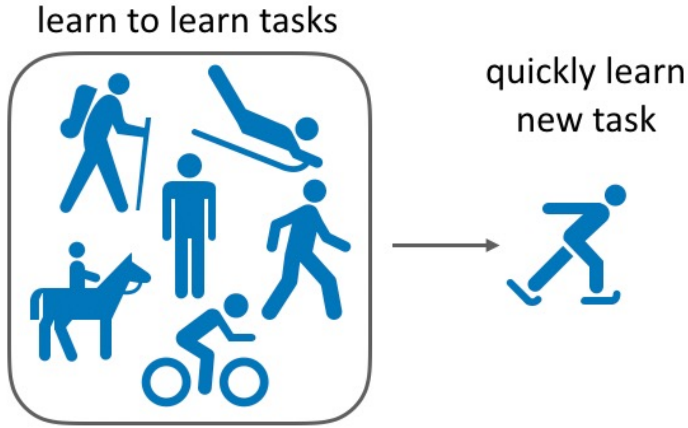


Lake, Human-level concept learning through probabilistic program induction, Science 2015;
Salman, Do Adversarially Robust ImageNet Models Transfer Better?, NeurIPS 2020. Yu, Meta-world, CoRL 2019

Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

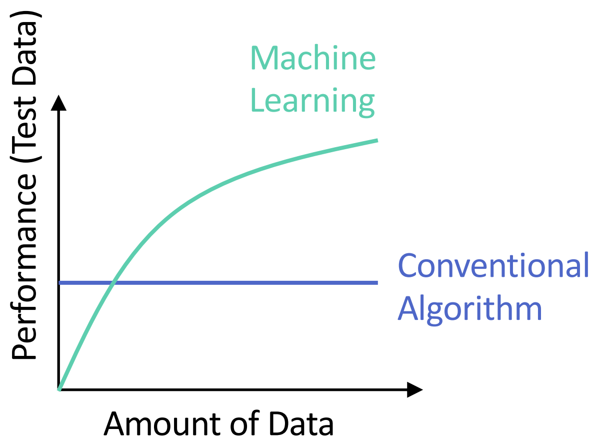
Meta Learning and Learning-to-Learn

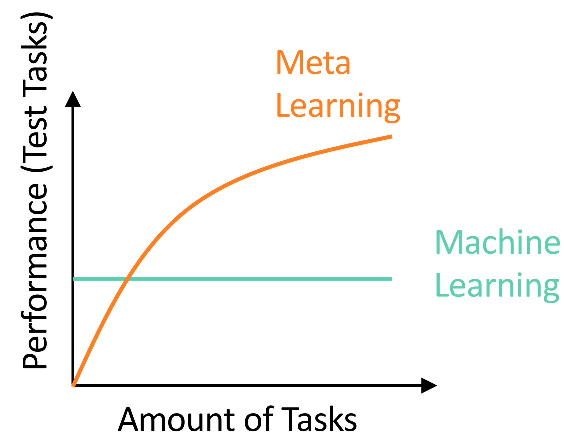


	Past: Shallow Learning	Current: Deep Learning	Future: Deep Meta Learning
Classifier	Learned	Learned	Learned
Feature	Hand-Crafted	Learned	Learned
Learning Algorithm EG: Architecture, Hyper-params, Optimiser, etc	Hand-crafted	Hand-crafted	Learned

Hospedales et al, Meta-Learning in Neural Networks: A Survey, IEEE T-PAMI 2021

Defining Learning-to-Learn

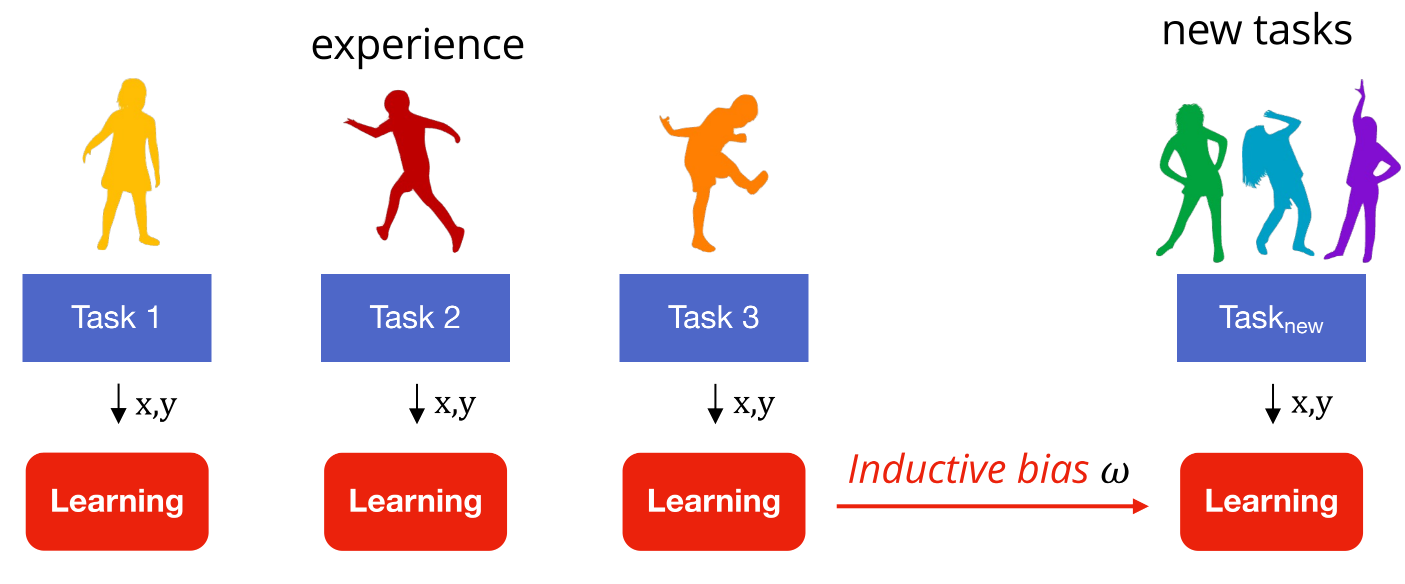




- Machine Learning Definition (Mitchell, 1993):
  - Given: Task *T*, experience *E*~*T*, performance measure *P*.
  - A program learns if performance at *T* wrt *P* improves with amount of experience *E*.

- Learning to Learn Definition (Thrun, 1998)
  - Given: Tasks *T* from a task distribution *T*~*D*, experience of each task *E*~*T*, performance measure *P*.
  - A program learns-to-learn if performance at tasks *T* wrt *P* improves with amount of experience *E* and with number of tasks *T*.

Learning-to-Learn aka Meta-Learning

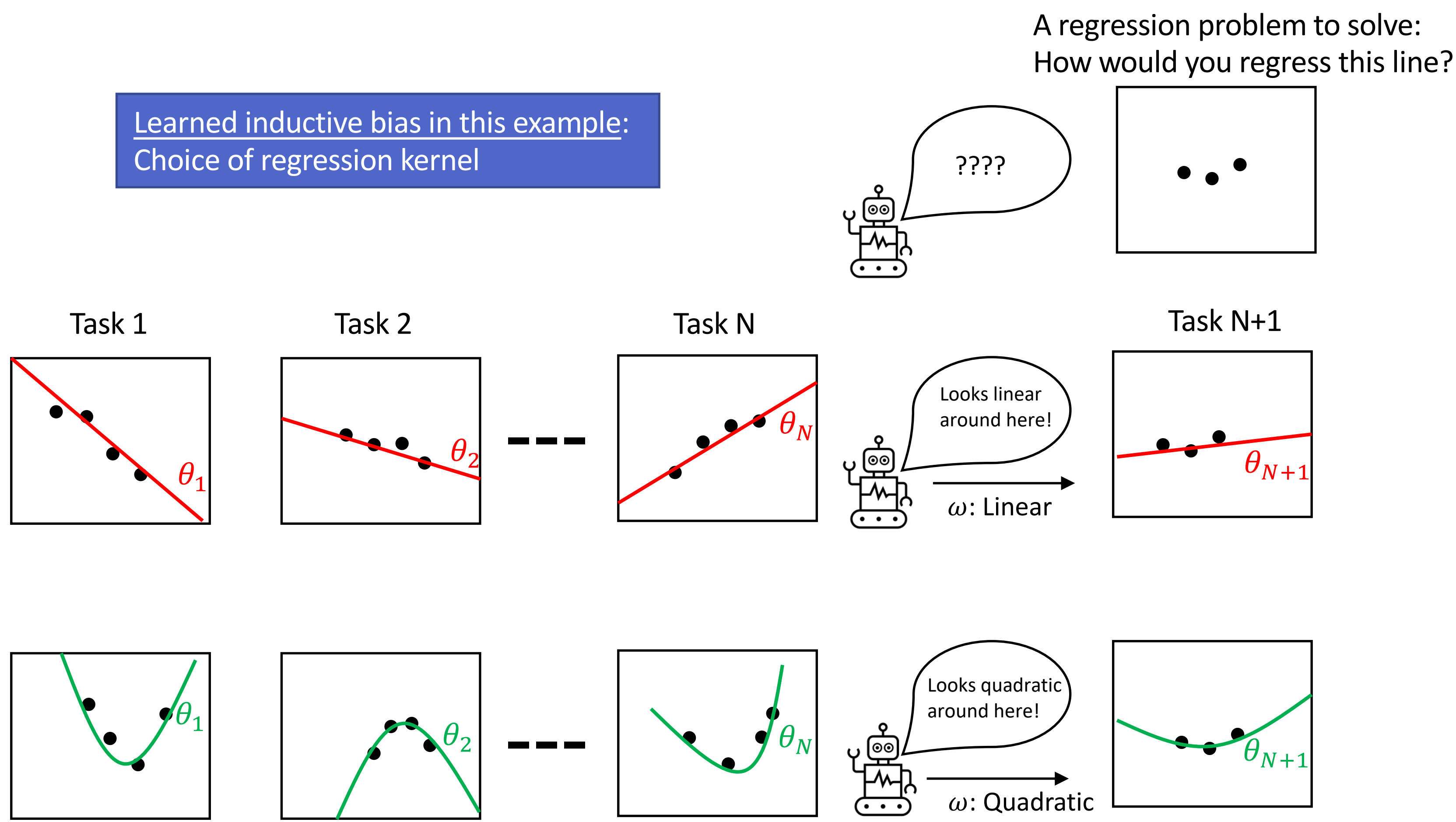


Few-Shot Meta-Learning: Learn the inductive bias that leads to success with small training sets.

What can we (meta-)learn and transfer? Priors, representations, optimizers, hyperparameters,…

[Adapted from Drori & Vanschoren AAAI'21 Tutorial]

A Minimal Example of Human Meta-Learning



Probabilistic View

- Supervised Learning (from scratch). $D = \{(x_i, y_i)\}$

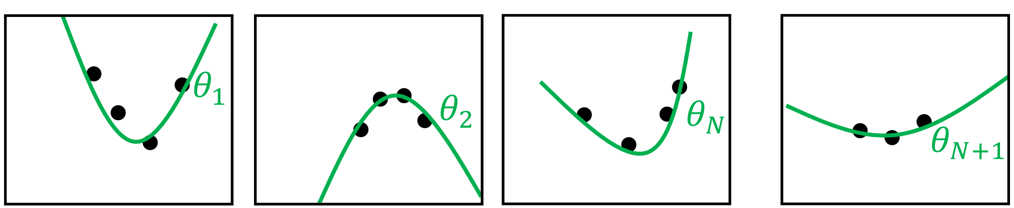
$$\text{argmax}_\theta p(\theta|D) = \text{argmax}_\theta \sum_i \log p(y_i|x_i,\theta) + \log p(\theta)$$

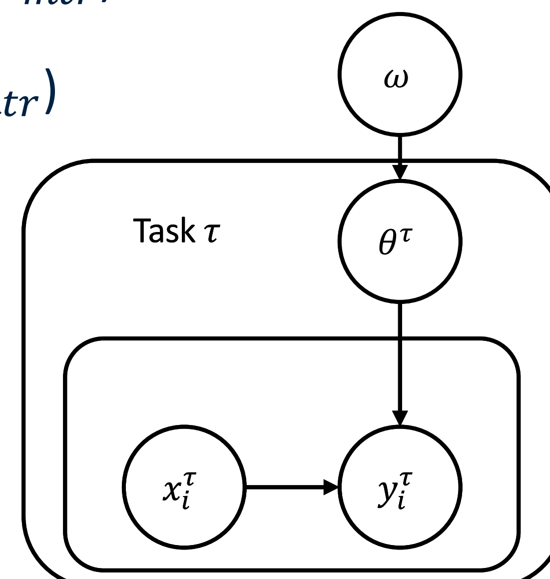
- If there are also related tasks $\mathcal{D}_{mtr} = \{D_\tau\}$: $\text{argmax}_\theta p(\theta|D, \mathcal{D}_{mtr})$

$$\log p(\theta|D,\mathcal{D}_{mtr}) = \log \int_\omega p(\theta|D,\omega)p(\omega|\mathcal{D}_{mtr})\,d\omega$$

$$\approx \log p(\theta|D,\omega^*) + \log p(\omega^*|\mathcal{D}_{mtr})$$

where $\omega^* = \underbrace{\text{argmax}_\omega \log p(\omega|\mathcal{D}_{mtr})}_{\text{Meta-learning}}$





Probabilistic View

- Meta-Train: $\omega^* = \text{argmax}_\omega \log p(\omega|\mathcal{D}_{mtr}) = \text{argmax}_\omega \sum_\tau \log p(\omega|D_\tau)$

- Meta-Test: $\theta^* = \text{argmax}_\theta \log p(\theta|D,\omega^*) = A_{\omega^*}(D)$

Summarize the learning algorithm as a function

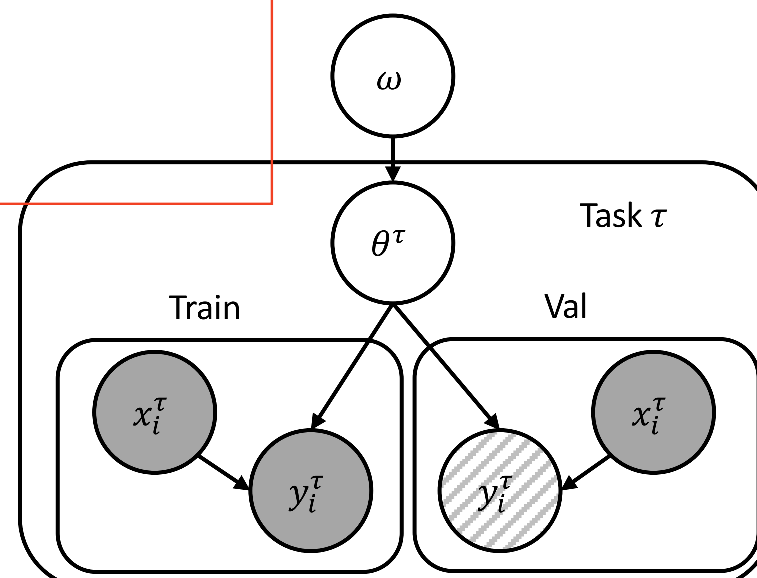
Important #1:

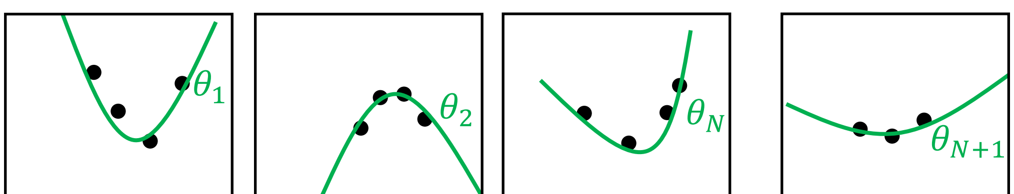
Learn ω so that we generalize from D_τ^{tr} to D_τ^{va}

$$\omega^* = \text{argmax}_\omega \sum_\tau \log p(\theta_\tau|D_\tau^{va})$$

$$\text{s.t. } \theta_\tau = \mathcal{A}_\omega(D_\tau^{tr})$$

Implies this graphical model:





Compare:

(Meta) optimise for overfitting

$$\omega^* = \text{argmax}_\omega \sum_\tau \log p(\theta_\tau | D_\tau^{tr}) \\ \text{s.t.} \theta_\tau = \mathcal{A}_\omega(D_\tau^{tr})$$

(Meta) optimise for generalisation

$$\omega^* = \text{argmax}_\omega \sum_\tau \log p(\theta_\tau | D_\tau^{va}) \\ \text{s.t.} \theta_\tau = \mathcal{A}_\omega(D_\tau^{tr})$$

Important #2:
If the auxiliary train sets are small...
Meta-optimize for generalisation after FSL!

Optimization View: Bilevel Optimization

How to Meta-Learn?

- Second-order Gradient
- Implicit Gradient
- Evolution
- …

Why Optimize?

- Generalisation Accuracy + Data-Efficiency
- …

What to Meta-Learn?

- Bayesian Prior
- Architecture (NAS)
- Optimiser
- …

Meta-Training

Tasks D_τ

Outer Loop: $$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

Split each task into train & val. Aka: Query/Support.

Inner Loop: $$\theta_\tau^* = \mathcal{A}(D_\tau^{tr}, \omega) = \arg\min_\theta \mathcal{L}(D_{\tau,}^{tr}; \theta_\tau, \omega)$$

Encapsulate training algorithm $\mathcal{A}$

ω^*: How to learn?

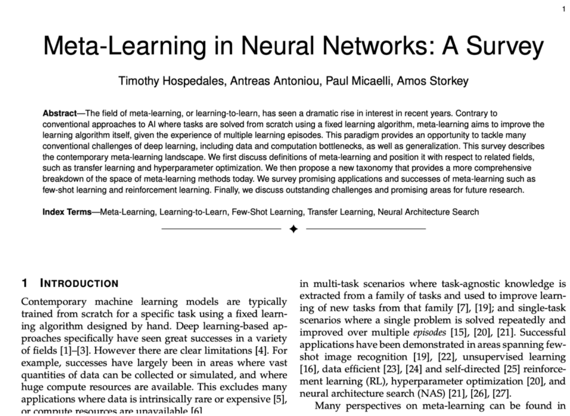
Meta-Testing

New Task D_{new}

Learning: $$\theta_{new}^* = \mathcal{A}(D_{new}, \omega^*) = \arg\min_\theta \mathcal{L}(D_{new,}; \theta_{new}, \omega^*)$$

Inference: $$y' = f_{\theta_{new}^*}(x')$$

Outline



Meta-Learning in Neural Networks: A Survey

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, Amos Storkey

Abstract—The field of meta-learning, or learning-to-learn, has seen a dramatic rise in interest in recent years. Contrary to conventional approaches to AI where tasks are solved from scratch using a fixed learning algorithm, meta-learning aims to improve the learning algorithm itself, given the experience of multiple learning episodes. This paradigm provides an opportunity to tackle many conventional challenges of deep learning, including data and computation bottlenecks, as well as generalization. This survey describes the contemporary meta-learning landscape. We first discuss definitions of meta-learning and position it with respect to related fields, such as transfer learning and hyperparameter optimization. We then propose a new taxonomy that provides a more comprehensive breakdown of the space of meta-learning methods today. We survey promising applications and successes of meta-learning such as few-shot learning and reinforcement learning. Finally, we discuss outstanding challenges and promising areas for future research.

Index Terms—Meta-Learning, Learning-to-Learn, Few-Shot Learning, Transfer Learning, Neural Architecture Search

1 INTRODUCTION

Contemporary machine learning models are typically trained from scratch for a specific task using a fixed learning algorithm designed by hand. Deep learning-based approaches specifically have seen great successes in a variety of fields [1]–[3]. However there are clear limitations [4]. For example, successes have largely been in areas where vast quantities of data can be collected or simulated, and where huge compute resources are available. This excludes many applications where data is intrinsically rare or expensive [5], or compute resources are unavailable [6]

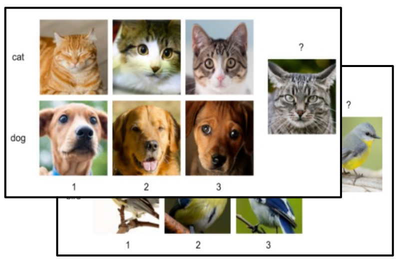
in multi-task scenarios where task-agnostic knowledge is extracted from a family of tasks and used to improve learning of new tasks from that family [7], [19]; and single-task scenarios where a single problem is solved repeatedly and improved over multiple *episodes* [15], [20], [21]. Successful applications have been demonstrated in areas spanning few-shot image recognition [19], [22], unsupervised learning [16], data efficient [23], [24] and self-directed [25] reinforcement learning (RL), hyperparameter optimization [20], and neural architecture search (NAS) [21], [26], [27].

Many perspectives on meta-learning can be found in

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

Hospedales et al, Meta-Learning in Neural Networks: A Survey, IEEE T-PAMI 2021

Few-Shot Meta-Learning: Summary



Meta-train

Val set
Aka: "query"

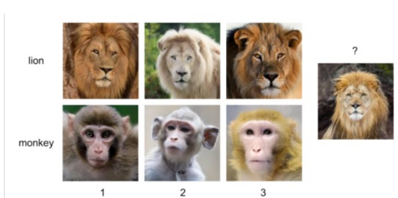
Few-shot train set
Aka: "Support"

$$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}^{meta}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

$$\theta_\tau^* = \mathcal{A}(D_\tau^{tr}, \omega) = \arg\min_{\theta} \mathcal{L}(D_{\tau,}^{tr}; \theta_\tau, \omega)$$

Suggests amortised learner

Suggests iterative gradient descent –based learner



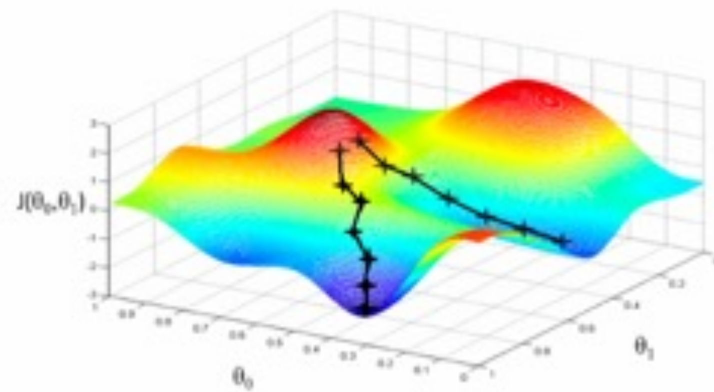
Meta-Test

$$\theta_{new}^* = \mathcal{A}(D_{new}^{tr}, \omega^*) = \arg\min_{\theta} \mathcal{L}(D_{new,}^{tr}; \theta_{new}, \omega^*)$$

$$y_\tau' = f_{\theta_{new}^*}(x_\tau')$$

MAML: Context

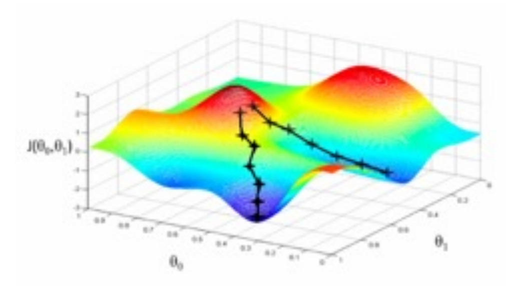
- In non-convex optimization, the final local minima depends on the starting point.
  - Few-shot regime: Minima found likely to be poor.



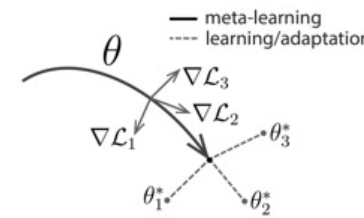
- MAML: Can we find a starting point that leads to good generalization accuracy, even with small training data?

Model Agnostic Meta-Learning

Connection of GBML to HPO: Scale to Millions of Hyperparameters!



- Setup:
  - Goal: Generalisation after few-shot learning (small D^{tr})
  - Meta representation: ω:= initial parameters θ^0.
  - Meta optimizer: Gradient.
  - => Learn an initial condition θ^0 such that few-step/few-shot fine-tuning from i.c. θ^0 works well.



Meta-Train

Outer Loop: $$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

Inner Loop: $$\theta_\tau^* = \arg\min_{\theta} \mathcal{L}(D_{\tau,}^{tr}; \theta_\tau, \omega) = \underbrace{\omega - \alpha \nabla_\theta \mathcal{L}(D_\tau^{tr}; \theta_\tau)}$$

Deploy/ Meta-Test: $$\theta_{new}^* = \omega^* - \alpha \nabla_\theta \mathcal{L}(D_{new}^{tr}; \theta_{new})$$

Assume the inner loop can be solved with one (or few) gradient-descent steps if given a good initial condition ω

Finn, ICML'17, Model Agnostic Meta-Learning

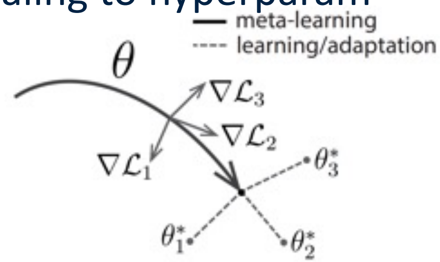
GBML Trends: Efficiency / Optimizer / Meta-Params

► GBML is still expensive.

- Cost: (1) High order gradients, (2) Store compute graph for default reverse mode differentiation (memory proportional to number of inner steps).

FSL: Annoying, Not fatal

- Huge amount of ongoing work trying to make gradient-based meta-learning faster & more scalable:
  - First order approximations [Reptile, Nichol arXiv'18,FOMAML Finn ICML'17]
  - Forward mode differentiation [Franceschi ICML'17, Micaelli NeurIPS'21]
    - Constant memory but worsen scaling to hyperparam dimension
  - Implicit Gradient [Rajeswaran NeurIPS'19; Lorraine AISTATS'21]
    - Constant memory but require inner convergence
  - Evolution [ES-MAML, Song ICML'20; EvoGrad Bohdal, NeurIPS'21]
    - Avoid second order gradient & constant memory, but worsen scaling to hyperparam dimension
  - Hyper Distillation [Lee, ICLR'22]
    - Alleviate second order gradient



GBML Trends: Efficiency / Optimizer / Meta-Params

▶ Meta-Learning Aspects of the Inner Loop Optimizer

Inner Loop of optimizer learning algorithms

Growing space of meta-parameters ω to learn:

- MAML: $\theta \leftarrow \theta_0^{\omega} - \beta \nabla_\theta L(\theta)$ — $|\omega| = |\theta|$
- MetaSGD: $\theta \leftarrow \theta_0^{\omega} - \beta \mathrm{diag}(\omega) \nabla_\theta L(\theta)$ — Elementwise learning rate: $|\omega| = 2|\theta|$
- Sparse MAML: $\theta \leftarrow \theta_0^{\omega} - \beta I_{\omega>0} \nabla_\theta L(\theta)$ — Elementwise sparse updates: $|\omega| = 2|\theta|$
- MetaCurve/MetaMD: $\theta \leftarrow \theta_0^{\omega} - \beta \mathrm{P}(\omega) \nabla_\theta L(\theta)$ — Preconditioning matrix, $|\omega| = |\theta| + |\theta|^2$
- LEO/MMAML : $\theta \leftarrow g_\omega(D_{trn}) - \beta \nabla_\theta L(\theta)$ — Initialization network, $|\omega| < |\theta|$
- Neural Optimizers: $\theta \leftarrow NN_\omega(\nabla_\theta L(\theta), \theta)$ — Neural Optimizer $|\omega| \ll |\theta|$ or $|\theta| \ll |\omega|$

Number of meta-parameters to learn.

Finn, ICML'17, Model Agnostic Meta-Learning; Li, arXiv'17 MetaSGD; Park, NeurIPS'19, Meta-Curvature; Gao, arXiv'22, MetaMD
Rusu ICLR'19 Latent Embedding Optimization; Rajeswaran NeurIPS'19, Meta Learning with Implicit Gradient,
Stuhmer, arXiv'21, Constrained Adaptation for Meta-Learning; Ravi ICLR'17, Optimization as a Model for Few-Shot Learning

GBML Trends: Efficiency / Optimizer / Meta-Params

▶ Recap: How MAML avoids overfitting?

MAML inner loop:

$$\theta_1 \leftarrow \theta_0 - \beta \nabla_\theta L\left(D_{fsl}^{tr}\right)$$

$$....$$

$$\theta_K \leftarrow \theta_{k-1} - \beta \nabla_\theta L(D_{fsl}^{tr})$$

Reduced overfitting because:

1. We meta-learned an initial condition $\omega = \theta_0$ that leads to good generalization.
2. We only take a small number of gradient steps K.

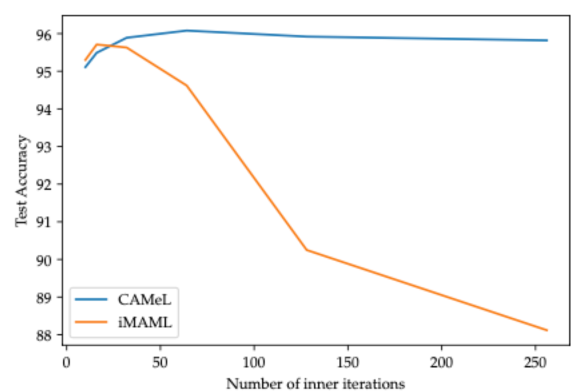
=> θ_0 is dealt with elegantly by meta-learning, but K is still a heuristic.

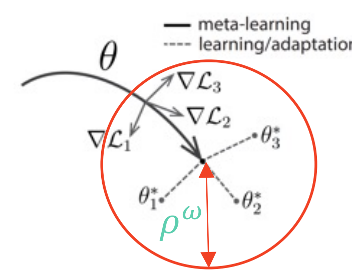
GBML Trends: Efficiency / Optimizer / Meta-Params ►CAMEL: Constrained Meta-Learning

Regularising MAML to improve few-shot reliability

- MAML: $\theta \leftarrow \theta_0^\omega - \beta \nabla_\theta L(\theta)$ Regularize by limiting steps to K=1,2,3. But can't meta-learn K ☹
- iMAML: $\theta \leftarrow \theta_0^\omega - \beta \nabla_\theta (L(\theta) + \lambda ||\theta - \theta_0^\omega||^2)$ Regularize by limiting steps and weight decay But can't (efficiently) meta-learn λ ☹
- CAMEL: $\theta \leftarrow \text{project}_{|\theta - \theta_0^\omega| < \rho^\omega}(\theta_0^\omega - \beta \nabla_\theta L(\theta))$ Regularize by constraining net update size Can efficiently meta-learn ρ^ω ☺

Removes a very tricky hyperparameter!

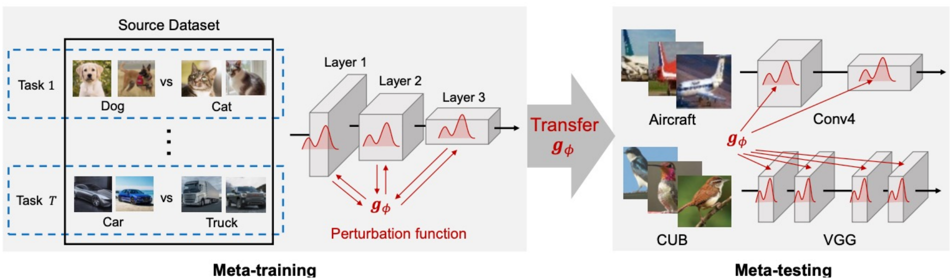


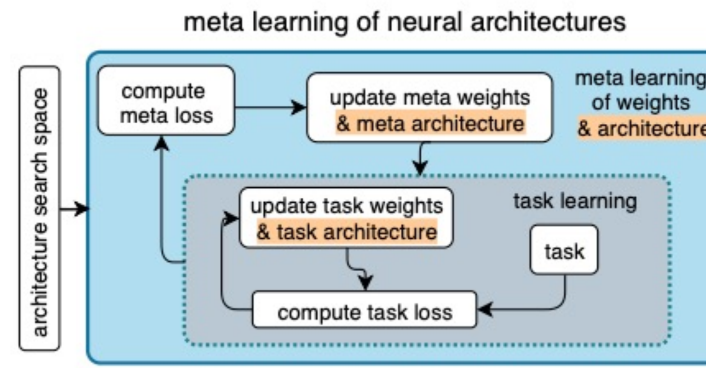


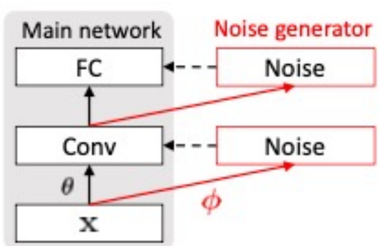
[Stuhmer, Gouk, Hospedales, arXiv'21, Constrained Adaptation for Meta-Learning]

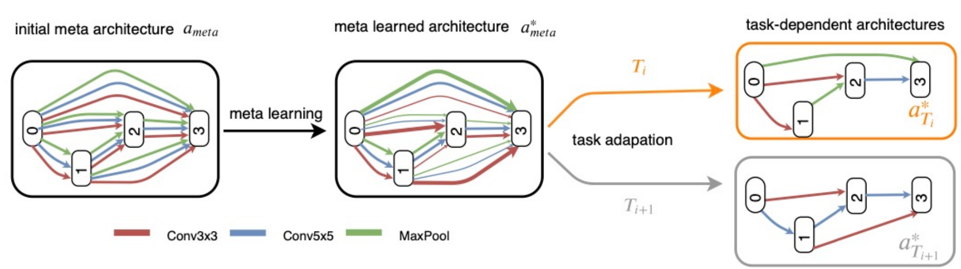
GBML Trends: Efficiency / Optimizer / Meta-Params

▶ Learning Other Meta-Params





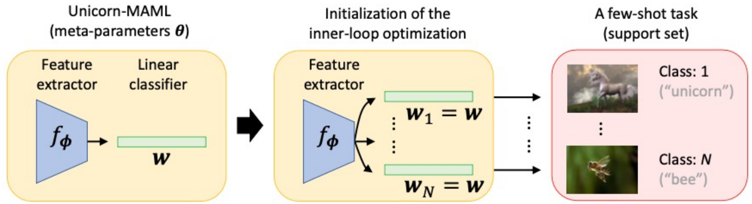




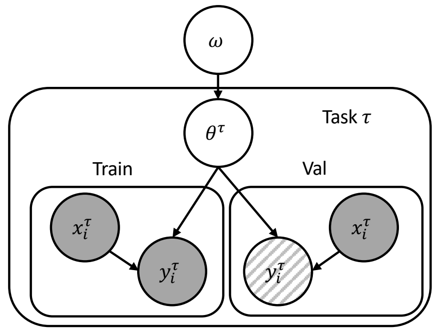
Lee, ICLR'20, Meta Dropout: Learning to Perturb. Elsken, CVPR'20, Meta-Learning NAS for FSL

Two State of the Art Few-Shot GBML

- Unicorn-MAML [Ye, How to Train Your MAML, ICLR'22]:
  - Good for classic MAML: (1) Sufficient inner loop steps, (2) care with different role of feature extractor + classifier.
  - => Beats a lot of prior SotA!



- Meta-NIW [Kim & Hospedales, A Hierarchical Bayesian Model for Deep Few-Shot Meta Learning, arXiv'23]:
  - Variational BNN solution to the canonical graphical model:
  - => Conjugate updates. No storing compute graph: Fast ☺.
  - => Uniquely scales MAML up to VIT backbones! ☺
  - => Excellent results on classification, regression, calibration.



Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

Is there any theory for few-shot meta-learning?

- Q: Can we guarantee generalization even in FSL scenario?
- Q: How can we know if the meta-train set and/or the meta-test support set are large enough that ω should generalize to new meta-test tasks?
- Q: Can any theory meaningfully apply to deep learning?

Stuhmer, Gouk, Hospedales, arXiv'21, CAMeL: Constrained Adaptation for Meta-Learning
Rothfuss, ICML'21, PACOH: Bayes-optimal meta-learning with PAC-guarantees
Kim & Hospedales, arXiv'23, A Hierarchical Bayesian Model for Deep Few-Shot Meta Learning

Theory For Few-Shot Meta-Learning?

Guaranteed Test Error <= Empirical Train Error + Deep neural net complexity

$$\mathbb{E}_{(\vec{x},y)}[\mathcal{L}(f(\vec{x}),y)] \le \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(f(\vec{x}_i),y_i) + \frac{4\sqrt{\log(2d)}cX\sum_{j=1}^{L}\frac{D_j}{B_j}\prod_{j=1}^{L}2B_j}{\sqrt{m}}$$

Exponential in Num Layers ☹

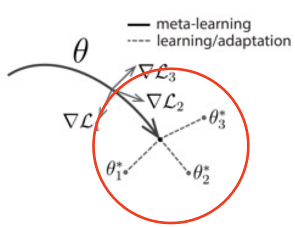
Weight Norms

Num Data

Standard Deep Learning Theory

Guaranteed Test Error <= Empirical Train Error + Deep neural net complexity

(Task Overfit) + (Meta Overfit)



$$L_{\mathcal{E}}(\mathcal{H}_{\boldsymbol{\theta}^{(0)},\boldsymbol{\rho}}) \le \hat{L}_{\mathcal{E}}(\mathcal{H}_{\boldsymbol{\theta}^{(0)},\boldsymbol{\rho}}) + \frac{\Omega_1(\boldsymbol{\rho},\boldsymbol{\tau})}{\sqrt{m}} + \frac{\Omega_2(\boldsymbol{\rho},\boldsymbol{\tau})}{\sqrt{n}}$$

Instances per task

Num Tasks

Distance ρ allowed to move (in weight space) by gradient descent from initialization

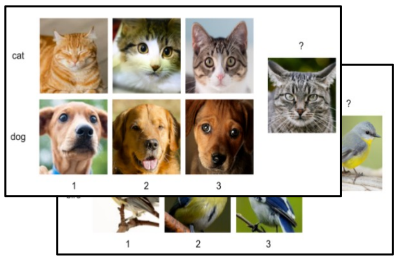
Deep Meta-Learning Theory

Gouk, ICLR'20, Distance Based Regularisation; Stuhmer arXiv'21, CAMeL: Constrained Adaptation for Meta-Learning

Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

Few-Shot Meta Learning: Gradient vs Amortized



Meta-train

Val set
Aka: “query”

Few-shot train set
Aka: “Support”

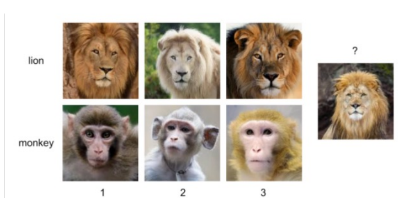
$$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}^{meta}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

$$\theta_\tau^* = \mathcal{A}(D_\tau^{tr}, \omega) = \arg\min_{\theta} \mathcal{L}(D_{\tau,}^{tr}; \theta_\tau, \omega)$$

Amortised Learning:
- Pay an up front cost for meta-learning, but amortise it over faster learning for many meta-test tasks. Here: Faster=feed-forward.

Suggests amortised learner

Suggests iterative gradient descent –based learner



Meta-Test

$$\theta_{new}^* = \mathcal{A}(D_{new}^{tr}, \omega^*) = \arg\min_{\theta} \mathcal{L}(D_{new,}^{tr}; \theta_{new}, \omega^*)$$

$$y_\tau' = f_{\theta_{new}^*}(x_\tau')$$

Prototypical Network

- Background: Nearest-centroid classifier (NCC)

Train:

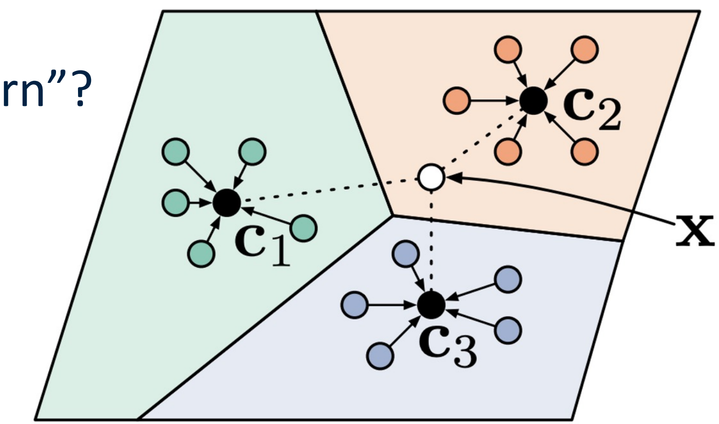
$$c_k(S_k) = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} x_i$$

Test:

$$p(y = k|x) \propto \exp(-\|x - c_k\|^2)$$

- Q: What part of NCC classifier says "how to learn"?
  - A: Distance metric!

$$p(y = k|x) \propto \exp(D_\omega(x, c_k))$$

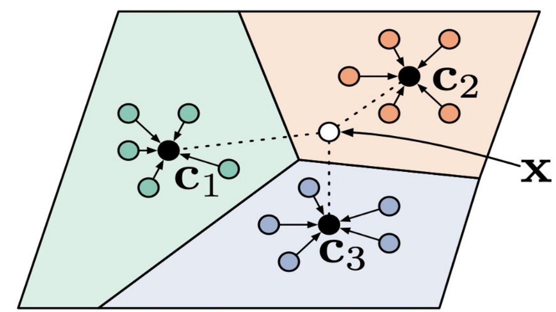


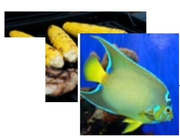
Snell et al, NIPS'17, Prototypical Networks for Few Shot Learning

Prototypical Network

- Learning: A deep "Prototype" per class: $\mathcal{A}(D,\omega)$: $\boldsymbol{\theta}_k = \frac{1}{|D_k|} \sum_{(\boldsymbol{x}_i, y_i) \in D_k} f_\omega(\boldsymbol{x}_i)$
- Classify with: $p(y=k|x) \propto \exp(-\|f_\omega(\boldsymbol{x}_i) - \boldsymbol{\theta}_k\|^2)$
- Meta-Learn by: $\min_\omega \sum_{D_\tau^v, D_\tau^t = D_\tau} \mathcal{L}^{meta}(D_\tau^v; \mathcal{A}(D_\tau^t, \omega))$

ω: How shall we represent the inputs
Before measuring Euclidean distance?





CNN f_ω → $\{\boldsymbol{\theta}_k\}_{k=1}^K$

CNN f_ω → $\exp(-\|f_\omega(\boldsymbol{x}_i) - \boldsymbol{\theta}_k\|^2)$ → [1, 0, 0, 0, 0]

Snell et al, NIPS'17, Prototypical Networks for Few Shot Learning

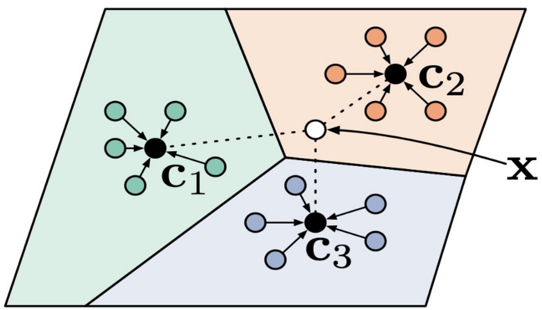
AML Trends: Metrics / Dyn. Feats. / Joint Inference

▶ Improved Distance Metrics

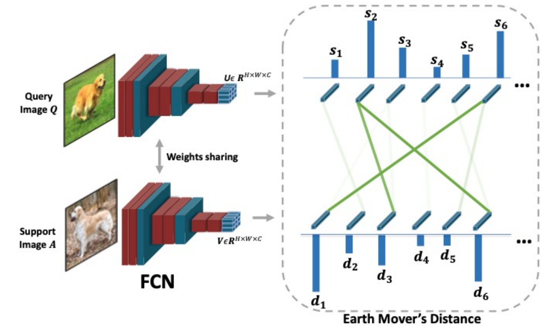
$p(y = k|x) \propto \exp(g_\omega(f_\omega(\boldsymbol{x}_i), f_\omega(S_k)))$

ProtoNet: Deep Embedding + Euclidean DIstance

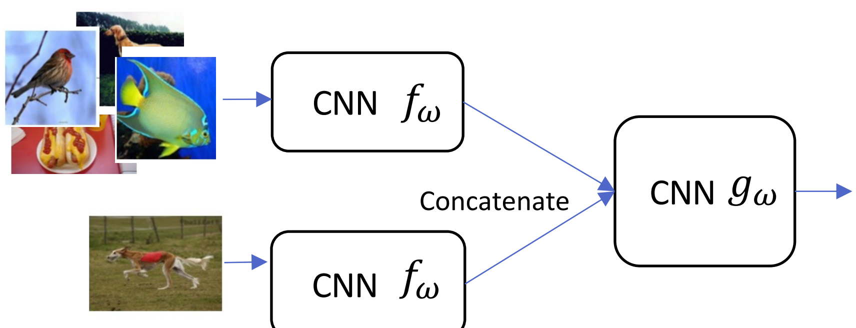
$$p(y = k|x) \propto \exp(-\|f_\omega(\boldsymbol{x}_i) - \boldsymbol{\theta}_k\|^2)$$



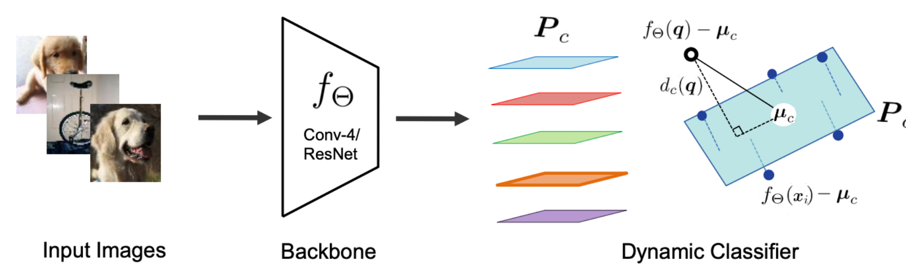
DeepEMD: Deep Embedding + Earth Movers Distance



RelationNet: Deep Embedding + Neural Distance



SubspaceNet: Deep Embedding + Point-to-Plane Distance

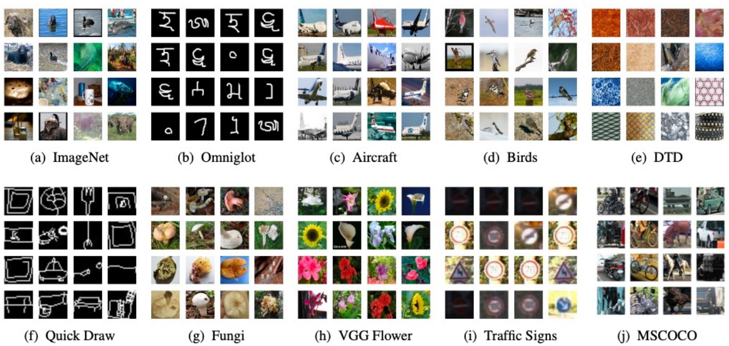


Snell, NIPS'17, ProtoNet. Sung CVPR'18 RelationNet. Zhang CVPR'20 DeepEMD. Simon, CVPR'20, DeepSubspaceNet.

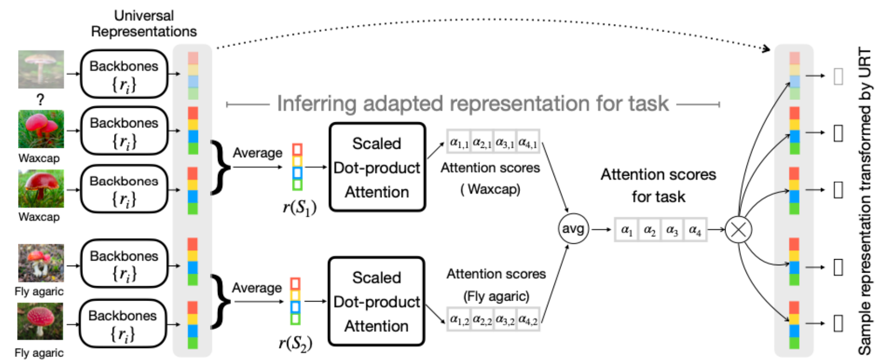
AML Trends: Metrics / Dynamic Feats. / Joint Inf.

▶ Feature Extractor Conditioned on Support Set

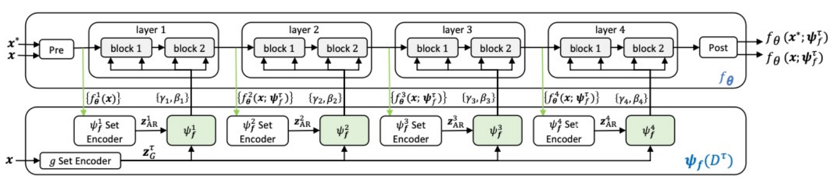
Inspiration: Meta-Dataset benchmark
▶ Distribution shift makes pre-trained features sub-optimal



Universal Representation Transformer



CNAPS Adaptive Feature Extractor

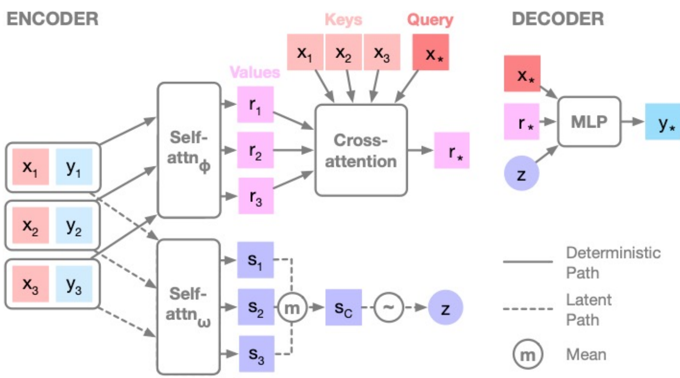


Triantafillou, ICLR'20, Meta Dataset. Requeima, NeurIPS'19, CNAPS. Liu, ICLR'20, Universal Representation Transformer.

AML Trends: Metrics / Dyn. Feats. / Joint Inference

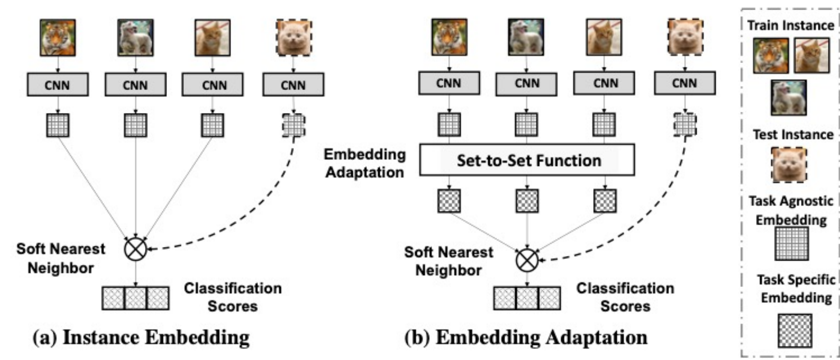
▶ Reason jointly about the query + supports.

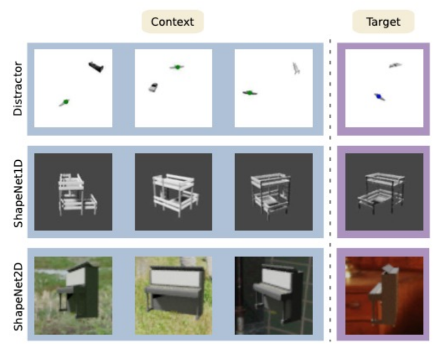
Neural Processes



Recently evaluated as SotA for the less studied few-shot regression!
+ Good at uncertainty estimation.

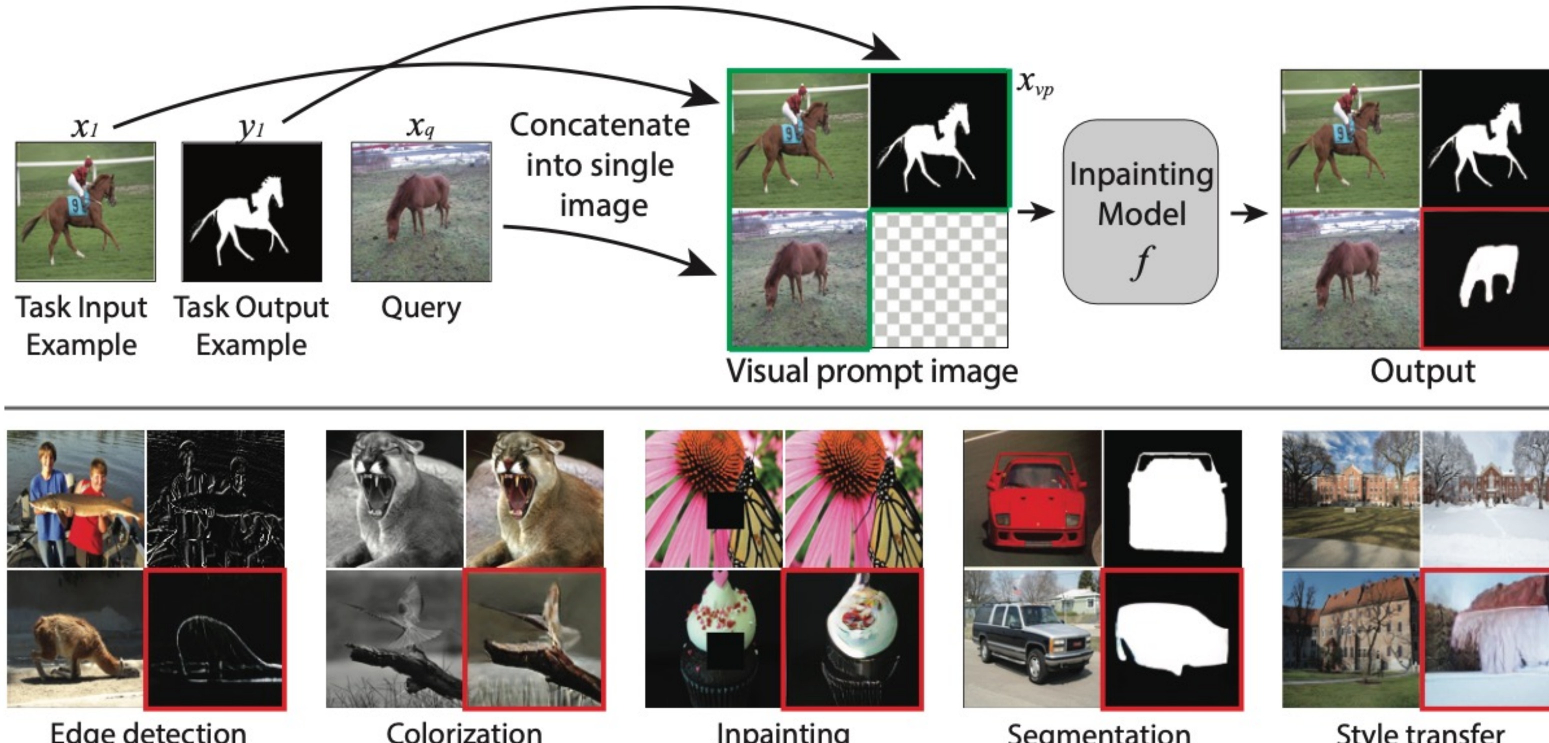
FEAT





Garnelo, ICML'18, Conditional Neural Processes. Kim, ICLR'19, Attentive neural processes
Ye, CVPR'20, Few-shot learning … with set-to-set functions. Gao, CVPR'22, What matters for meta-learning regression tasks?

A State of the Art Amortised FSL

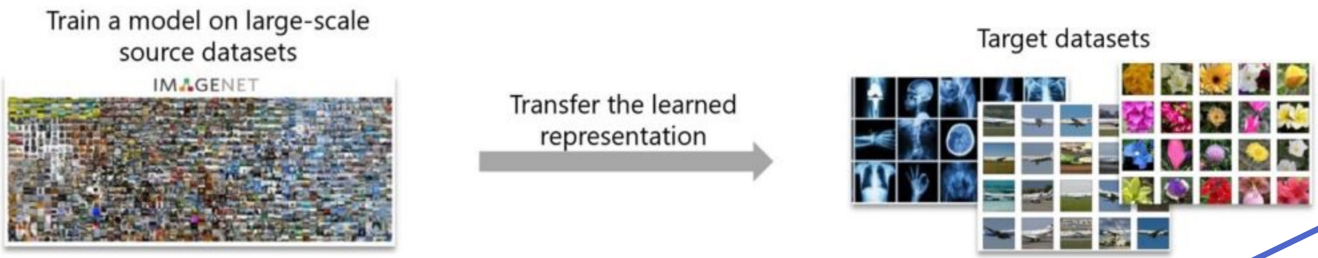


Bar, NeurIPS'22, Visual prompting via image inpainting

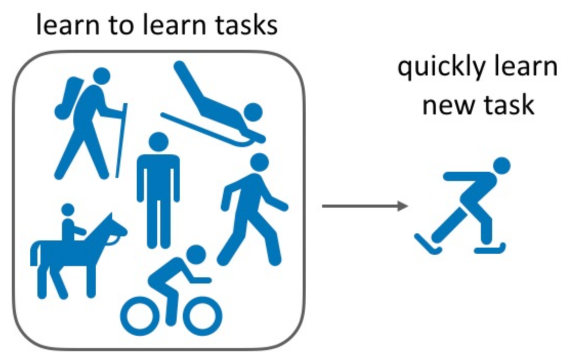
Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & "In-context learning"
- Applications
- Challenges & Outlook

An ongoing debate….



Is meta-learning worth it, or transfer learning is as good or better?



Is meta-learning useful for few-shot recognition: No?

- ANIL [ICLR-20]: Meta-test adaptation in MAML-like methods doesn’t help. They just learn a good feature. Then you can use NCC.
- Unravelling [ICML-20]: Meta-training in MetaOptNet/R2D2 learns a good feature (MAML doesn’t). But this can be replicated in classical training with an appropriate extra loss term.
- CloserLook [ICLR-19], SimpleShot [arXiv-19], Manifold Charting [WACV-20], Rethinking FSL [ECCV’20]: No. Pre-train followed by linear/NCC works well.
- FT [ICLR’22], TSA [CVPR’22], PMF [CVPR’22], FiT [ICLR’23]: No, pre-train followed by fine-tuning is all you need.

Is meta-learning useful for few-shot recognition: Yes?

- BOIL [ICLR-21]: Contrary to the claim of ANIL, representation adaptation of MAML does help.
- Unicorn [ICLR-22]: Properly tuned MAML works great.

....Which group to believe?....

► Idea: Develop meta-learners which are agnostic to choice of feature extractor / feature extractor initialization.

► If they help, meta-learning is at least complementary to transfer learning.

- MetaQDA [ICCV-21]: Yes. Meta-learning is complementary to pre-trained features in fixed feature condition!
- NFTS [arXiv-23]: Yes. Meta-learning can answer the question ”how to fine-tune?”!

Shallow Bayesian Meta-Learning

Setup:
Given a fixed pre-trained feature $f(x)$ and target dataset $\mathrm{D} = \{f(x), y\}$.
Meta-learn shallow classifier $g_\theta(\cdot)$, so that $g_\theta(f(x))$ performs well even with few training examples for g_θ.

How?
Learn a Bayesian prior on θ.

Training

Learning by Bayesian Inference

Support Set

Prior over classifier parameters

$$p(\theta|D_s, \omega) \propto p(D_s|\theta)p(\theta|\omega)$$

Quadratic Discriminant Analysis → Bayesian QDA

$$p(y|x,\theta) \propto \exp\left(\left(x-\mu_y\right)^T \Sigma_y \left(x-\mu_y\right)\right)$$

$$\theta_y = \mu_y, \Sigma_y$$

Everything is tractable if $p(\theta|\omega)$ is normal inverse wishart!

Inference

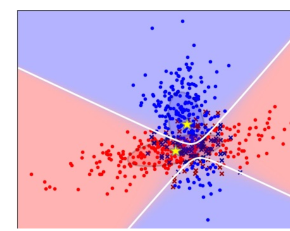
Recognize final query set by integrating out parameters.

$$p\left(D_q|D_s,\omega\right) = \int \prod_i p(x_i, y_i|\theta)\, p(\theta|D_s,\omega)d\theta$$

Meta-Learning

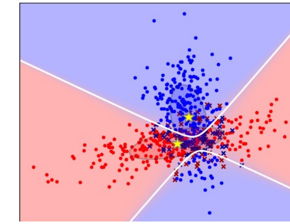
Episodic training of the parameter prior ω

$$\min_{\omega} E_{D_s\, D_q} - \log(D_q|D_s; \omega)$$



[Zhang, ICCV'21, MetaQDA]

Shallow Bayesian Meta-Learning with MetaQDA



BayesianQDA: Recognize query set by Bayesian inference on Gaussians.
Integrate out their unknown means & covariances:

$$p\left(D_q \middle| D_s, \omega\right) = \int p(x, y|\theta) p(\theta|D_s, \omega) d\theta$$

$$p(y|x, \theta) \propto \exp\left(\left(x - \mu_y\right)^T \Sigma_y \left(x - \mu_y\right)\right)$$
$$\theta = \mu, \Sigma$$

✓ Closed form solution for classifier posterior given prior and support set
(By careful choice of inverse-Wishart conjugate prior $p(\theta|\omega)$)

✓ Closed form solution for inference of query given support + prior.
(Approximate and v. fast, or exact and fast via student-t posterior)

✓ Train the optimal inverse-Wishart prior ω by gradient during meta-train.

✓ Accurate: More powerful than a linear classifier, but avoids overfitting thanks to meta-learned prior!

EG: +4% over MetaOptNet.

✓ Well calibrated probabilities..

[Zhang, ICCV'21, MetaQDA]

Neural Fine-Tuning Search

Freeze & Insert adapters

... but where?

Selective Fine-tuning

<u>Recent SotA on Meta-Dataset</u>:

- FLUTE (ICML'21)
- PMF (CVPR'22)
- FiT (ICLR'23)
- TSA (CVPR'22)

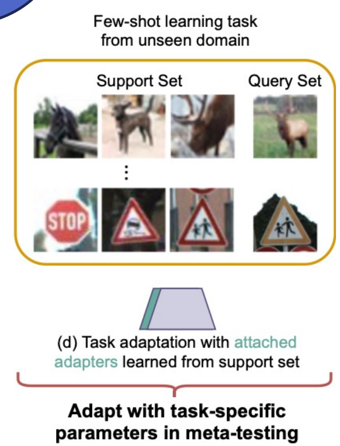
<u>Key Idea</u>:
Careful adaptation of pre-trained features

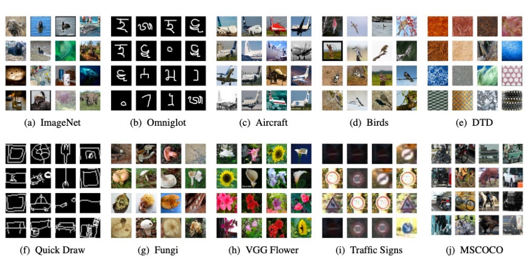
Evolutionary search

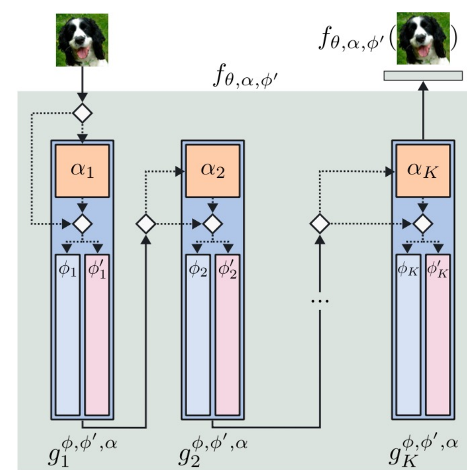
ω: Binary adaptation mask

$$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

$$\theta^* = \mathcal{A}(D_\tau^{tr}, \omega) = \arg\min_{\theta} \mathcal{L}(D_\tau^{tr}; \theta, \omega)$$

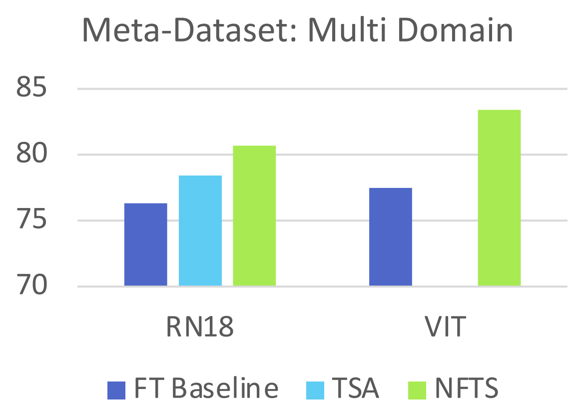






[Eustratiadis, arXiv'23, Neural Fine Tuning Search]

Neural Fine-Tuning Search: Results



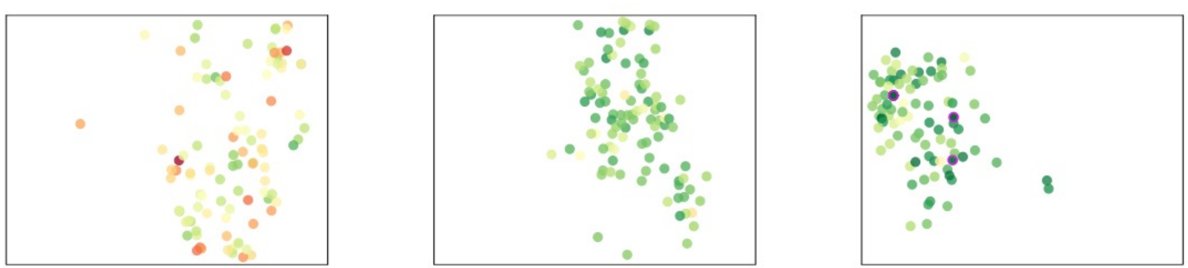
Evolutionary search

ω: Binary adaptation mask

$$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

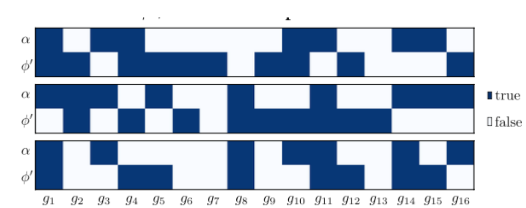
$$\theta^* = \mathcal{A}(D_\tau^{tr}, \omega) = \arg\min_{\theta} \mathcal{L}(D_{\tau}^{tr}; \theta, \omega)$$

Fitness (Accuracy) of each fine-tuning mask



Generation 1 Generation 5 Generation 15

Final Masks

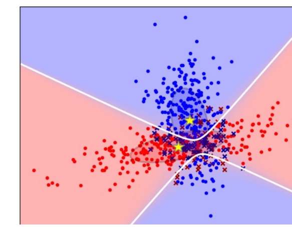


(b) Top 3 performing paths subject to diversity constraint.

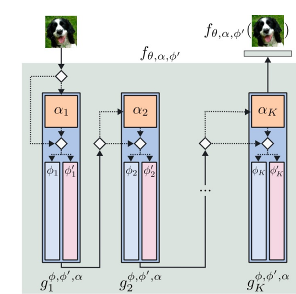
[Eustratiadis, arXiv'23, Neural Fine Tuning Search]

Is meta-learning useful for few-shot recognition? Conclusion: Yes!

- MetaQDA [ICCV-21]: Yes. Meta-learning a prior on the classifier layer, is complementary to any choice of fixed feature extractor!



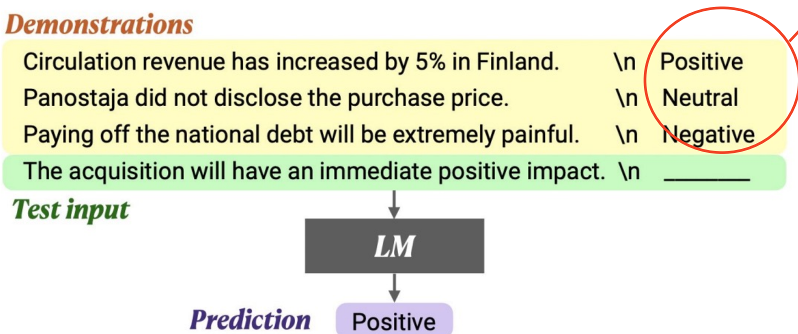
- NFTS [arXiv-23]: Yes. Meta-learning ”how to fine-tune?” is complementary to any choice of initial feature extractor!



Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context” learning
- Applications
- Challenges & Outlook

Classic ICL is emergent….
….But explicit meta-learning seems to be better



Training on vast number of prior sentence completions.
…. Leads to emergent in-context learning.

Very reminiscent of our amortised meta-learner…

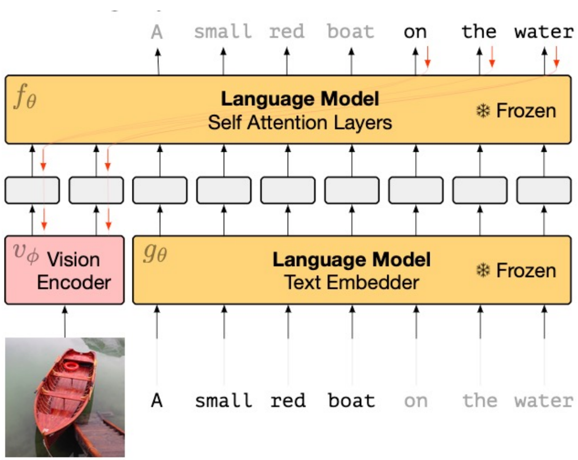
$$\min_{\omega} \sum_{(D_\tau^{va}, D_\tau^{tr}) \in \mathcal{D}} \mathcal{L}(D_\tau^{va}; \mathcal{A}(D_\tau^{tr}, \omega))$$

Actually training as meta-learning substantially improves emergent ICL (GPT2) in head-to-head comparison.
[Min, Meta ICL, ACL'22]

Brown, NeurIPS'20, Language models are few-shot learners. Min, ACL'22, MetaICL: Learning to Learn In Context

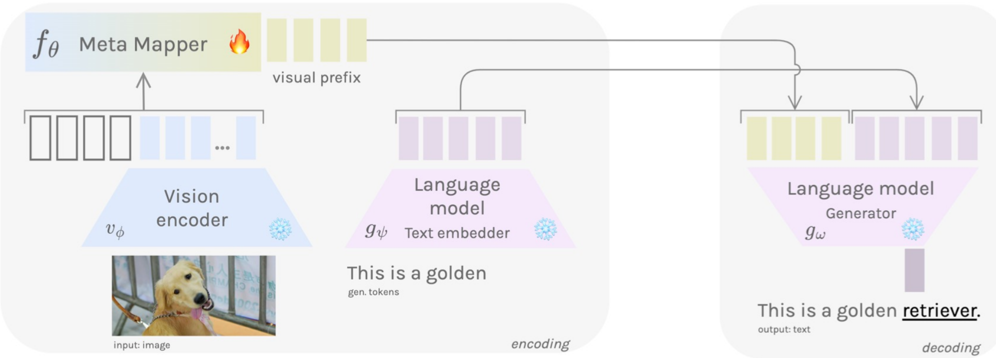
Leveraging (emergent) ICL for few-shot vision…

<u>Setup</u>:
1. Align vision encoder & language decoder by training a captioning objective.
2. Exploit language model's emergent ability to perform amortised in-context learning.



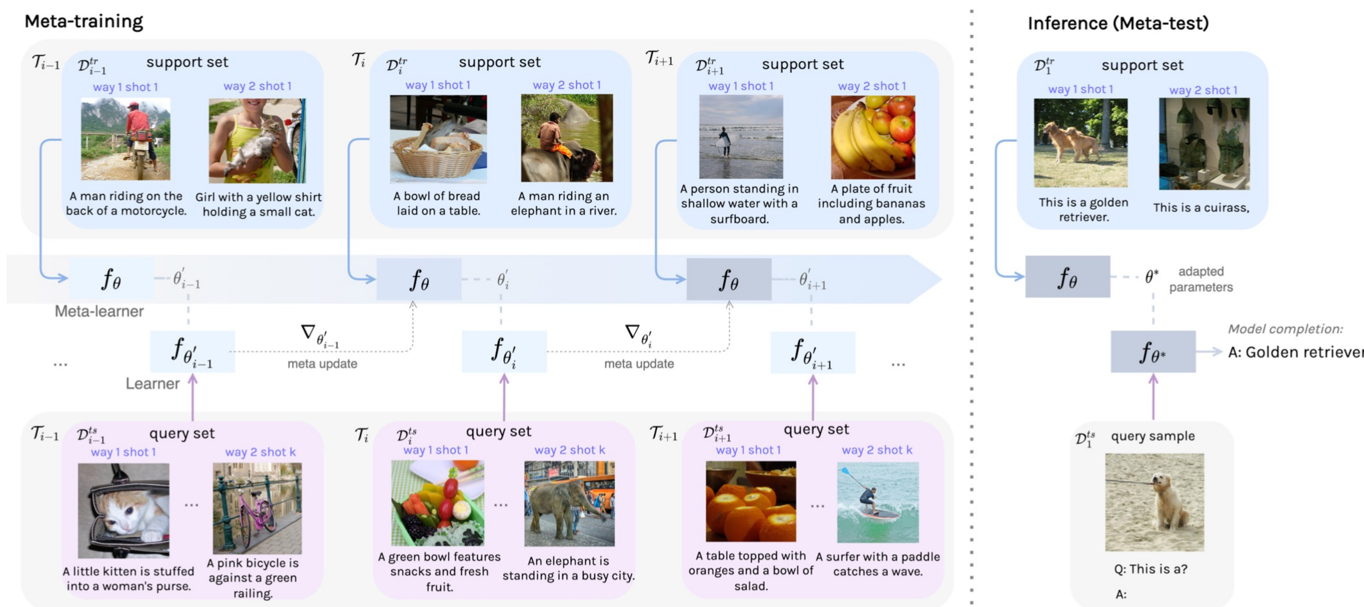
Tsimpoukelli, NeurIPS'21, Multimodal few-shot learning with frozen language models

Leveraging (meta) ICL for few-shot vision…



<u>Setup</u>: Align vision encoder & language decoder by training a "meta mapper".

1. Meta-Train: Explicitly learn mapper initialization many episodes (CF: MAML).
2. Meta-Test: Fine-tune mapper on support set and infer query set.



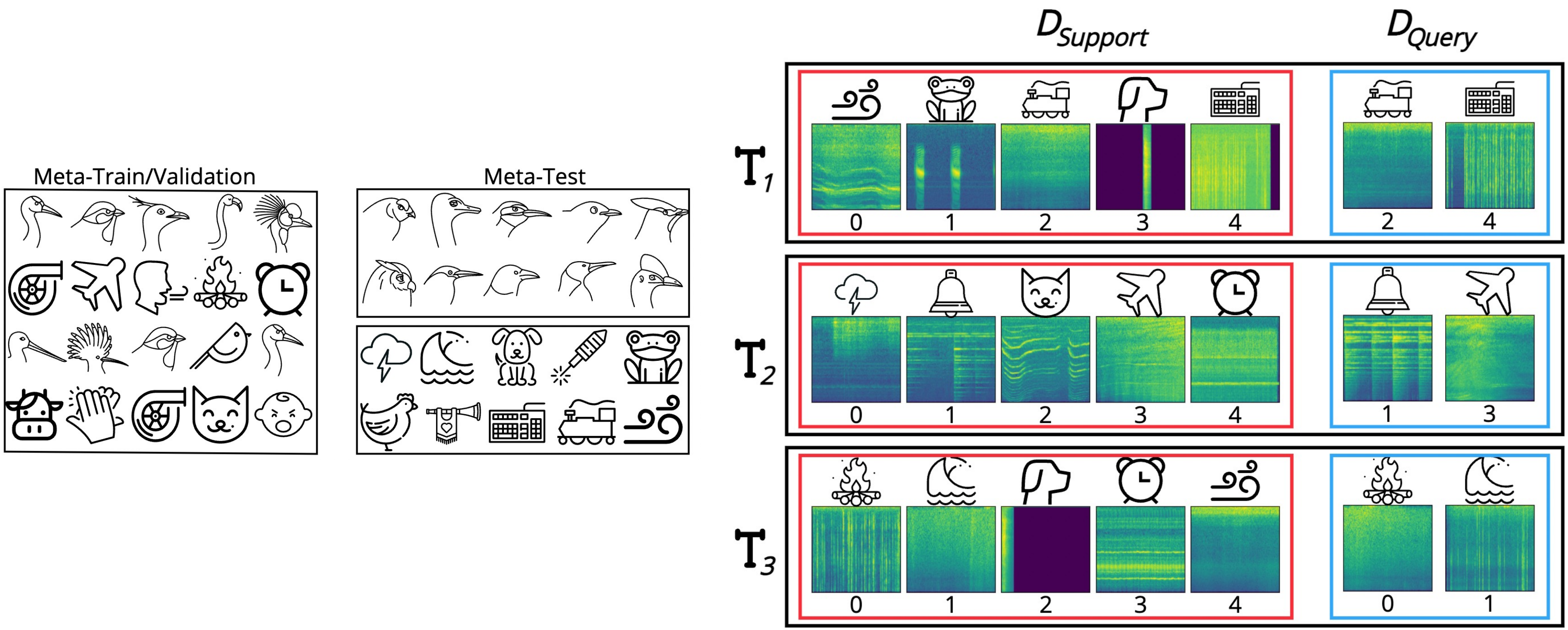
Najdenkoska, ICLR'23, Meta Learning to Bridge Vision and Language Models for Multimodal Few-Shot Learning

Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

Meta Audio

A Few-Shot Audio Classification Benchmark



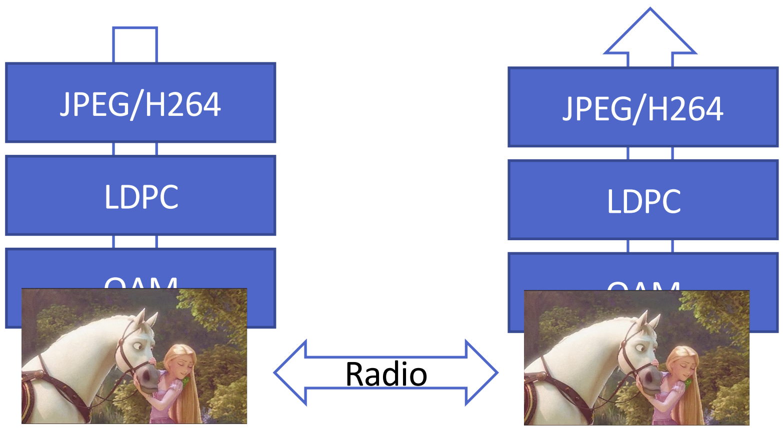
Heggan, ICANN'22, MetaAudio: A Few-Shot Audio Classification Benchmark
https://cheggan.github.io/posts/2022/04/MetaAudio_blog/

MetaAudio: Results

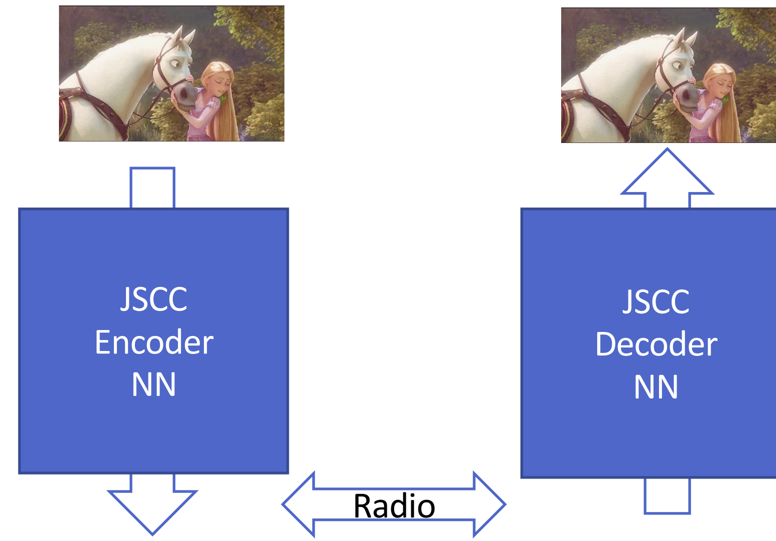
- Modern gradient-based few-shot learners (meta-curvature) are in the lead. Amortised learners are behind.
  - (Unlike vision).
- Supervised pre-training is far-behind.
  - => Don’t overfit your conclusions to popular benchmarks!

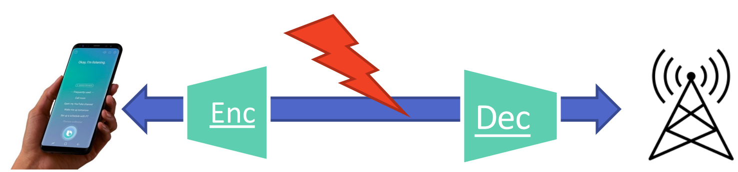
Dataset	FO-MAML	FO-Meta-Curvature	ProtoNets	SimpleShot CL2N	Meta_baseline
ESC-50	74.66 ± 0.42	**76.17 ± 0.41**	68.83 ± 0.38	68.82 ± 0.39	71.72 ± 0.38
NSynth	93.85 ± 0.24	**96.47 ± 0.19**	95.23 ± 0.19	90.04 ± 0.27	90.74 ± 0.25
FSDKaggle18	**43.45 ± 0.46**	43.18 ± 0.45	39.44 ± 0.44	42.03 ± 0.42	40.27 ± 0.44
VoxCeleb1	60.89 ± 0.45	**63.85 ± 0.44**	59.64 ± 0.44	48.50 ± 0.42	55.54 ± 0.42
BirdCLEF 2020 (Pruned)	56.26 ± 0.45	**61.34 ± 0.46**	56.11 ± 0.46	57.66 ± 0.43	57.28 ± 0.41
Avg Algorithm Rank	2.4	**1.2**	3.8	4.0	3.6

Comms is trending toward DL...

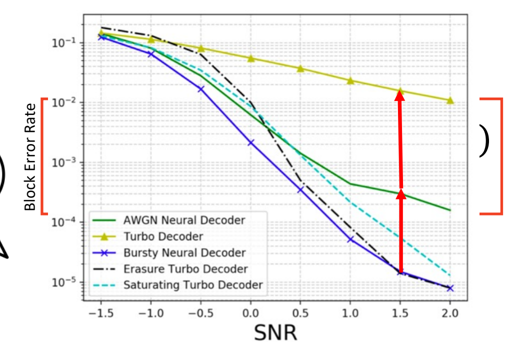


Standard





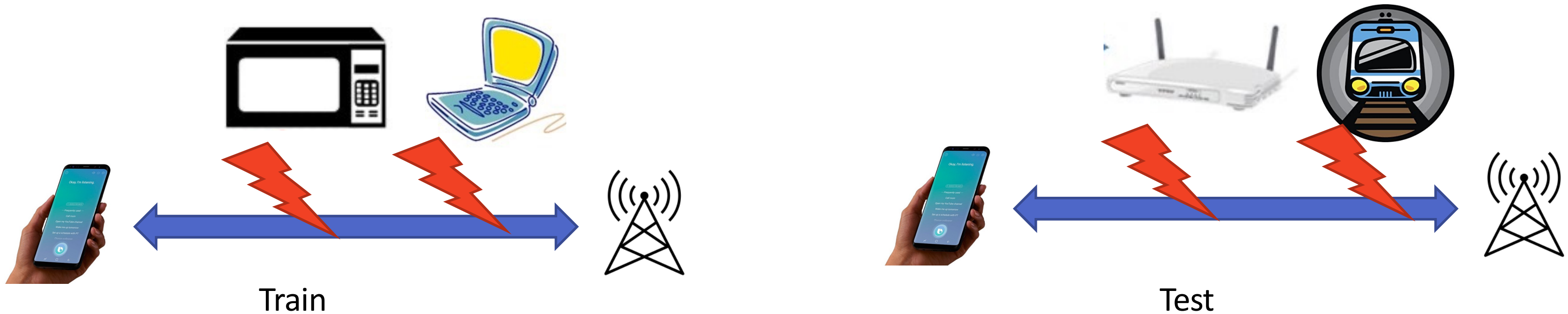
Decoding Error Rate



[Kim et al; Deepcode..., NeurIPS'18; Li et al, A Channel Coding Benchmark for Meta-Learning, NeurIPS'21]

Neural Channel Coding: Challenge ▶ Solution: Meta-learning

- Distribution shift between train and test ☹
  - => Performance drop!



- Meta-Learning: Few-shot adaptation to distribution shift.
  - ~Few-shot autoencoder adaptation
  - Meta-coding benchmark: [Li al, A Channel Coding Benchmark for Meta-Learning, NeurIPS'21 Benchmark Track]
  - ✓ Controllable task complexity. ✓ Controllable train-test distribution shift. ✓ Controllable task size.
  - Interesting results. EG: Meta Curvature is also very strong.

Video Quality Comparison

Standard Convolutional Viterbi Code

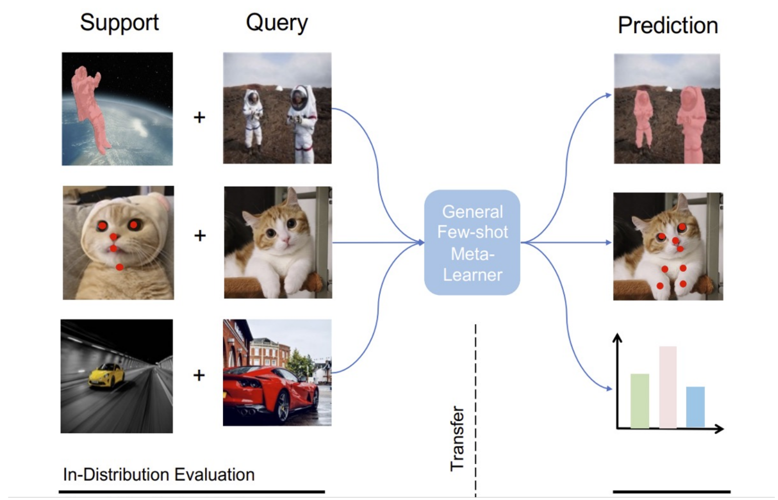
Adaptive Transformer Neural Code

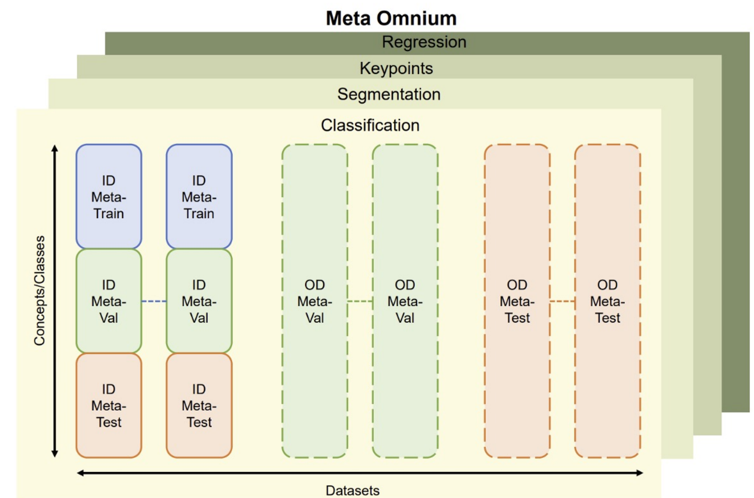
Meta-Omnium

- Mainstream meta-learning (meta-dataset, FS1K, etc):
  - ☹ Single task. Rewards over-engineered solutions to each task.
  - ☹ Single task. May not require feature adaptation.
  - ☹ Single task only.
  - ☹ Rewards standard pre-trained features.
  - ☹ Meta-dataset is too heavy.
  - ☹ Unclear HPO protocol. Rewards benchmark hacking.

- Meta-Omnium:
  - ☺ Multi-task. Rewards general purpose meta-learning.
  - ☺ Multi-task. Feature adaptation rewarded.
  - ☺ Provides multi-task vs single task comparison.
  - ☺ Rewards in-benchmark meta-learning.
  - ☺ Light enough for universities! (3GB, 3h-1080Ti)
  - ☺ Unclear HPO protocol. Rewards good research.

https://edi-meta-learning.github.io/meta-omnium/



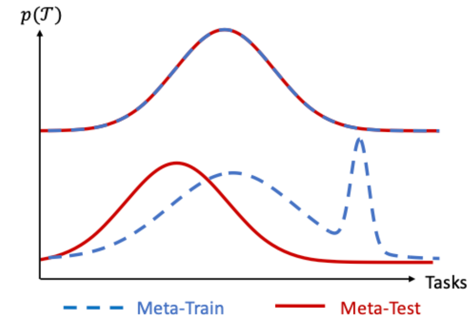


Bohdal et al, CVPR'23, Meta Omnium: A Benchmark for General-Purpose Learning-to-learn. [TUE-PM-341]

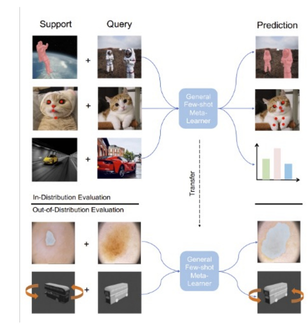
Outline

- Meta-Learning: Intro & Concepts
- Gradient-Based Meta-Learning
- Interlude: Some Theory
- Amortized Meta-Learning
- Meta-Learning vs Alternative FSL approaches
- Meta-Learning & “In-context learning”
- Applications
- Challenges & Outlook

Challenges & Outlook



- Multi-modal task distributions
- Meta-train > Meta-test distribution shift
- GBML vs Amortized (Efficiency vs Flexibility)
  - GBML: More novel choice of meta-parameters
- Better Benchmarks
- Integration with FMs
- Calibration
- Meta-Learning Beyond classification (later session)



Thank You! – Questions?