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....

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

--- | --- | ---
Classifier | Learned | Learned | Learned
Feature | Hand-Crafted | Learned | Learned
Learning Algorithm EG: Architecture, Hyper-params, Optimiser, etc | Hand-crafted | Hand-crafted | Learned

Defining Learning-to-Learn

- Machine Learning Definition (Mitchell, 1993):
  - Given: Task $T$, experience $E \sim T$, 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 $T \sim D$, experience of each task $E \sim T$, 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

Learned inductive bias in this example: Choice of regression kernel

A regression problem to solve: How would you regress this line?

Task 1

Task 2

Task N

Task N+1

Looks linear around here!

Looks quadratic around here!

\(\omega: \text{Linear}\)

\(\omega: \text{Quadratic}\)
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 $\omega^* = \text{argmax}_\omega \log p(\omega | \mathcal{D}_{mtr})$

\[\begin{array}{c}
\theta_1 \\
\theta_2 \\
\theta_N \\
\theta_{N+1}
\end{array}\]

$\omega$ \quad $\theta^\tau$ \quad $x^\tau_i$ \quad $y^\tau_i$
Probabilistic View

- Meta-Train: $\omega^* = \arg\max_\omega \log p(\omega|D_{mtr}) = \arg\max_\omega \sum_\tau \log p(\omega|D_\tau)$

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

Important #1:

Learn $\omega$ so that we generalize from $D^{tr}_\tau$ to $D^{va}_\tau$

$\omega^* = \arg\max_\omega \sum_\tau \log p(\theta_\tau|D^{va}_\tau)$

s.t. $\theta_\tau = A_\omega(D^{tr}_\tau)$

Implies this graphical model:
Compare:

(Meta) optimise for overfitting

\[ \omega^* = \arg\max_{\omega} \sum_\tau \log p(\theta_\tau | D_{\tau}^{tr} ) \]

s.t. \( \theta_\tau = A_\omega(D_{\tau}^{tr} ) \)

(Meta) optimise for generalisation

\[ \omega^* = \arg\max_{\omega} \sum_\tau \log p(\theta_\tau | D_{\tau}^{va} ) \]

s.t. \( \theta_\tau = A_\omega(D_{\tau}^{tr} ) \)

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 $D_\tau$

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

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

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

Meta-Testing

New Task $D_{new}$

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

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

Encapsulate training algorithm $A$

$\omega^*$: How to learn?
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: Summary**

$$\min_\omega \sum_{(D_{val}^{\tau}, D_{tr}^{\tau}) \in D} L_{meta}(D_{val}^{\tau}; A(D_{tr}^{\tau}, \omega))$$

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

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

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

Suggests amortised learner

Suggests iterative gradient descent–based learner

Val set

Aka: “query”

Few-shot train set

Aka: “Support”
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?

Finn, ICML’17, Model Agnostic Meta-Learning
Model Agnostic Meta-Learning

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

Meta-Train

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

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

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

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

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).
- 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
Growing space of meta-parameters $\omega$ to learn:

- **MAML**: $\theta \leftarrow \theta_0^\omega - \beta \nabla_{\theta} L(\theta)$ \hspace{1cm} $|\omega| = |\theta|$

- **MetaSGD**: $\theta \leftarrow \theta_0^\omega - \beta \text{diag}(\omega) \nabla_{\theta} L(\theta)$ \hspace{1cm} Elementwise learning rate: $|\omega| = 2|\theta|$

- **Sparse MAML**: $\theta \leftarrow \theta_0^\omega - \beta I_{\omega > 0} \nabla_{\theta} L(\theta)$ \hspace{1cm} Elementwise sparse updates: $|\omega| = 2|\theta|$

- **MetaCurve/MetaMD**: $\theta \leftarrow \theta_0^\omega - \beta P(\omega) \nabla_{\theta} L(\theta)$ \hspace{1cm} Preconditioning matrix, $|\omega| = |\theta| + |\theta|^2$

- **LEO/MMAML**: $\theta \leftarrow g_\omega(D_{\text{trn}}) - \beta \nabla_{\theta} L(\theta)$ \hspace{1cm} Initialization network, $|\omega| < |\theta|$

- **Neural Optimizers**: $\theta \leftarrow NN_\omega(\nabla_{\theta} L(\theta), \theta)$ \hspace{1cm} Neural Optimizer $|\omega| \ll |\theta|$ or $|\theta| \ll |\omega|$. 

---

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
MAML inner loop:
\[
\theta_1 \leftarrow \theta_0 - \beta \nabla_{\theta} L(D_{f_{sl}}^{tr}) \\
\vdots \\
\theta_K \leftarrow \theta_{k-1} - \beta \nabla_{\theta} L(D_{f_{sl}}^{tr})
\]
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 \( K \).

\( \theta_0 \) is dealt with elegantly by meta-learning, but \( K \) is still a heuristic.
Regularising MAML to improve few-shot reliability

- MAML: $\theta \leftarrow \theta_0^{\omega} - \beta \nabla_{\theta} L(\theta)$ \hspace{1cm} Regularize by limiting steps to $K=1,2,3$. But can’t meta-learn $K \smiley$

- iMAML: $\theta \leftarrow \theta_0^{\omega} - \beta \nabla_{\theta} (L(\theta) + \lambda \|\theta - \theta_0^{\omega}\|^2)$ \hspace{1cm} Regularize by limiting steps and weight decay But can’t (efficiently) meta-learn $\lambda \frown$

- CAMEL: $\theta \leftarrow \text{project}_{|\theta - \theta_0^{\omega}| < \rho^{\omega}}(\theta_0^{\omega} - \beta \nabla_{\theta} L(\theta))$ \hspace{1cm} Regularize by constraining net update size Can efficiently meta-learn $\rho^{\omega} \smile$

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 $\omega$ 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?

**Standard Deep Learning Theory**

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

**Deep Meta-Learning Theory**

\[ \mathcal{L}(\mathcal{H}_\theta(0), \rho) \leq \hat{\mathcal{L}}(\mathcal{H}_\theta(0), \rho) + \frac{\Omega_1(\rho, \tau)}{\sqrt{m}} + \frac{\Omega_2(\rho, \tau)}{\sqrt{n}} \]

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

\[
\min_{\omega} \sum_{(D_{va}^{tr}, D_{tr}) \in \mathcal{D}} \mathcal{L}_{\text{meta}}(D_{va}^{tr}; A(D_{tr}^{tr}, \omega))
\]

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

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

\[
y_{\tau}' = f_{\theta_{\text{new}}^*}(x_{\tau}')
\]

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.

Meta-train

Meta-test

Val set
Aka: “query”

Few-shot train set
Aka: “Support”

Suggests amortised learner

Suggests iterative gradient descent –based learner
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: \( A(D, \omega) : \theta_k = \frac{1}{|D_k|} \sum_{(x_i,y_i) \in D_k} f_{\omega}(x_i) \)

• Classify with: \( p(y = k | x) \propto \exp(-\| f_{\omega}(x_i) - \theta_k \|^2) \)

• Meta-Learn by: \( \min_{\omega} \sum_{D_t^v, D_t^t = D_t} \mathcal{L}_{\text{meta}} (D_t^v; A(D_t^t, \omega)) \)

\( \omega \): How shall we represent the inputs Before measuring Euclidean distance?

Snell et al, NIPS’17, Prototypical Networks for Few Shot Learning
AML Trends: Metrics / Dyn. Feats. / Joint Inference

**Improved Distance Metrics**

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

**ProtoNet:** Deep Embedding + Euclidean Distance

$p(y = k|x) \propto \exp(-\|f_{\omega}(x_i) - \theta_k\|^2)$

**DeepEMD:** Deep Embedding + Earth Movers Distance

**RelationNet:** Deep Embedding + Neural Distance

**SubspaceNet:** Deep Embedding + Point-to-Plane Distance

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

FEAT

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

Garnelo, ICML’18, Conditional Neural Processes.
Ye, CVPR’20, Few-shot learning … with set-to-set functions.
Kim, ICLR’19, Attentive neural processes
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....

Train a model on large-scale source datasets

Transfer the learned representation

Target datasets

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 $D = \{f(x), y\}$. Meta-learn shallow classifier $g_\theta(\cdot)$, so that $g_\theta(f(x))$ performs well even with few training examples for $g_\theta$.

How?
Learn a Bayesian prior on $\theta$.

Prior over classifier parameters

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

Learning by
Bayesian Inference

Support Set

Episodic training of the parameter prior $\omega$

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

Minimize the expected meta loss

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

Quadratic Discriminant Analysis $\rightarrow$ Bayesian QDA

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

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

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

[ Zhang, ICCV’21, MetaQDA ]
Bayesian QDA: Recognize query set by Bayesian inference on Gaussians. Integrate out their unknown means & covariances:

\[
p(D_q|D_s, \omega) = \int p(x, y|\theta)p(\theta|D_s, \omega)d\theta
\]

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

✓ 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 \(\omega\) 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

Recent SotA on Meta-Dataset:
- FLUTE (ICML’21)
- PMF (CVPR’22)
- FiT (ICLR’23)
- TSA (CVPR’22)

Key Idea:
Careful adaptation of pre-trained features

Freeze & Insert adapters

... but where?

Selective Fine-tuning

Evolutionary search

$$\min_{\omega} \sum_{(D^v, D^t) \in \mathcal{D}} \mathcal{L}(D^v; A(D^t, \omega))$$

$$\theta^* = A(D^t, \omega) = \arg\min_{\theta} \mathcal{L}(D^t; \theta, \omega)$$

[ Eustratiadis, arXiv’23, Neural Fine Tuning Search ]
Neural Fine-Tuning Search: Results

Evolutionary search

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

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

Meta-Dataset: Multi Domain

Fitness (Accuracy) of each fine-tuning mask

Final Masks

[ 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...

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

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...

Setup:
1. Align vision encoder & language decoder by training a captioning objective.
2. Exploit language model’s emergent ability to perform amortised in-context learning.
Leveraging (meta) ICL for few-shot vision...

**Setup:** 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
MetaAudio

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!

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

[ 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 FM$s$
• Calibration
• Meta-Learning Beyond classification (later session)
Thank You! – Questions?