Meta-Learning Frontiers: Universal, Uncertain, and Unsupervised

Sergey Levine
Chelsea Finn

UC Berkeley  Google Brain
Visual Distractors

real time

autonomous execution

Levine*, Finn*, Darrell, Abbeel, ‘16
Generalizable model-based RL via video prediction

Designated Pixel

Goal Pixel

Kalashnikov, Irpan, Pastor, Ibarz, Herzong, Jang, Quillen, Holly, Kalakrishnan, Vanhoucke, Levine. QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills
QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills
people can learn new skills extremely quickly
how?
we never learn from scratch!

what to transfer?
representations?
models?
can we just optimize for what we really want?
can we learn to learn?
Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference
Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference
The Meta-Learning Problem

**Supervised Learning:**
- Inputs: $x$
- Outputs: $y$
- Data: $\{(x, y)_i\}$
- $y = f(x; \theta)$

**Meta-Supervised Learning:**
- Inputs: $D_{train}$, $x_{test}$
- Outputs: $y_{test}$
- Data: $\{D_i\}$
- $D_i : \{(x, y)_j\}$
- $y_{test} = f(D_{train}, x_{test}; \theta)$

**Why is this view useful?**
Reduces the problem to the design & optimization of $f$.

Finn, Levine. *Meta-learning and Universality: Deep Representation...* ICLR 2018
Example: Few-Shot Classification

Given 1 example of 5 classes:

- Training data $D_{\text{train}}$

Classify new examples

- Test set $X_{\text{test}}$

Diagram adapted from Ravi & Larochelle ‘17

(meta-training $\mathcal{T}_1$, $\mathcal{T}_2$, $\ldots$)

(training classes $\ldots$)
Outline

- Meta-Learning Problem Statement

- Model-Agnostic Meta-Learning (MAML)

- Probabilistic Interpretation of MAML

- Meta-Learning with Automated Task Proposals

- Extensions to Robot Imitation & Intent Inference
Design of $f$?

**Recurrent network** (LSTM, NTM, Conv)

$y_{test} = f(D_{train}, x_{test}; \theta)$ Santoro et al. '16, Duan et al. '17, Wang et al. '17, Munkhdalai & Yu '17, Mishra et al. '17, ...

- complex model for complex task of learning
- impractical data requirements
Learning Few-Shot Adaptation

Key idea: Train over many tasks, to learn parameter vector $\theta$ that transfers.

Fine-tuning \([\text{test-time}]\)

Our method

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}_\text{test}^i (\theta - \alpha \nabla_{\theta} \mathcal{L}_\text{train}^i (\theta))$$

pretrained parameters

training data for new task

Finn, Abbeel, Levine ICML '17
Learning Few-Shot Adaptation

\[
\min_{\theta} \sum_{i} \mathcal{L}_{\text{test}}^i (\theta - \alpha \nabla_{\theta} \mathcal{L}_{\text{train}}^i (\theta))
\]

- \(\theta\): parameter vector being meta-learned
- \(\phi_i^*\): optimal parameter vector for task \(i\)

Model-Agnostic Meta-Learning

Finn, Abbeel, Levine ICML '17
...and the results keep getting better

**MiniImagenet few-shot benchmark: 5-shot 5-way**

Finn et al. ‘17: 63.11%
Li et al. ‘17: 64.03%
Kim et al. ‘18 (AutoMeta): 76.29%

---

**Program Synthesis**

Question: How many CRL teams are from York College?
SQL: `SELECT CRL Team FROM CRL`. Result: 8

Huang, Wang, Singh, Yih, He NAACL ’18

**Learning to Learn Distributions**

Reed, Chen, Paine, van den Oord, Eslami, Rezende, Vinyals, de Freitas ICLR ’18

**Federated Learning**

Chen, Dong, Li, He arXiv ’18

**Multi-Agent Competitions**

Al-Shedivat, Bansal, Burda, Sutskever Mordatch, Abbeel ICLR ’18

**Learning the learning rate**

Li, Zhou, Chen, Li arXiv ’17

**Masked Transformations**

Lee & Choi arXiv ’18

**Domain Generalization**

Li, Yang, Song, Hospedales AAAI ’18

**Semi-Supervised Few-Shot Learning**

Boney & Ilin ICLR workshop track ’18
Design of $f$?

**Recurrent network**

$y_{\text{test}} = f(D_{\text{train}}, x_{\text{test}}; \theta)$

network implements the "learned learning procedure"

Does it converge?
- Sort of?

What does it converge to?
- Who knows…

What to do if not good enough?
- Nothing

---

**MAML**

$y_{\text{test}} = f(x_{\text{test}}; \theta - \alpha \nabla_\theta \mathcal{L}(D_{\text{train}}))$

Does it converge?
- Yes (it’s gradient descent…)

What does it converge to?
- A local optimum (it’s gradient descent…)

What to do if not good enough?
- Keep taking gradient steps (it’s gradient descent..)

---

Does this structure come at a cost?
Recurrent network

\[ y_{test} = f(D_{train}, x_{test}; \theta) \]

\[ y_{test} = f(x_{test}; \theta - \alpha \nabla_\theta L(D_{train})) \]

Does this structure come at a cost?

For a sufficiently deep \( f \),
MAML function can approximate any function of \( D_{train}, x_{test} \)

**Assumptions:**
- nonzero \( \alpha \)
- loss function gradient does not lose information about the label
- datapoints in \( D_{train} \) are unique

Finn & Levine, ICLR 2018

Why is this interesting?
MAML has benefit of inductive bias without losing expressive power.

Is this structure useful?

Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018
How well can methods generalize to similar, but extrapolated tasks?

The world is non-stationary.

Omniglot image classification

MAML  TCML, MetaNetworks

Finn, Levine. *Meta-learning and Universality: Deep Representation...* ICLR 2018
How well can methods generalize to similar, but extrapolated tasks?

The world is non-stationary.

Takeaway: Strategies learned with MAML consistently generalize better to out-of-distribution tasks

Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018
Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference
A probabilistic interpretation

that's nice but is it... **Bayesian**?

kind of... but it can be more Bayesian

A useful property:

start from $\phi = \theta$ and follow gradient of $\log p(Y|X, \phi)$ for $K$ steps

dthis is equivalent to MAP on $p(\phi|X, Y)$

for a prior $p(\phi|\theta) = \mathcal{N}(\theta, \Sigma)$

and for a linear model $E[Y] = X^T \phi$

(Santos, 1996)
A probabilistic interpretation

start from $\phi = \theta$ and follow gradient of $\log p(Y|X, \phi)$ for $K$ steps
this is equivalent to MAP on $p(\phi|X, Y)$

for a prior $p(\phi|\theta) = \mathcal{N}(\theta, \Sigma)$
and for a linear model $E[Y] = X^T \phi$

MAP inference in this model

$\phi_i = \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, D_{\text{train}})$

can we do better than MAP?
can use Laplace estimate:

\[-\log p(X|\theta) \approx \sum_i \left[ -\log p(X_j | \hat{\phi}_j) - \log p(\hat{\phi}_j | \theta) + \frac{1}{2} \log \det(H_j) \right] \]

Grant, Finn, Levine, Darrell, Griffiths. “Recasting gradient-based meta-learning as hierarchical Bayes.”
Modeling ambiguity

+ -

=?

Can we learn to generate hypotheses about the underlying function?

Important for:

- learning to actively learn
- safety-critical few-shot learning (e.g. medical imaging)
- learning to explore in meta-RL
Modeling ambiguity

Can we *sample* classifiers?

**Intuition:** we want to learn a prior where a random *kick* can put us in different modes

\[
\phi \leftarrow \theta + \epsilon \\
\phi \leftarrow \phi + \alpha \nabla_{\phi} \mathcal{L}(\phi, \mathcal{D}_{\text{train}})
\]
Meta-learning with ambiguity

\[ \theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta) \]

\[ \phi_i \sim p(\phi_i | \theta) \]

Goal: sample \( \phi_i \sim p(\phi_i | x_i^{\text{train}}, y_i^{\text{train}}, x_i^{\text{test}}) \)

\[ \log p(y_i^{\text{train}} | x_i^{\text{train}}, \phi_i) \]

\[ \log p(y_i^{\text{test}} | x_i^{\text{test}}, \phi_i) \]

\[ \mathcal{L}(\phi, D_{\text{train}}) \]
Sampling parameter vectors

\[ \theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta) \]

\[ \log p(y_{i\text{train}} | x_{i\text{train}}, \phi_i) \]

\[ \phi_i \sim p(\phi_i | \theta) \]

\[ \log p(y_{i\text{test}} | x_{i\text{test}}, \phi_i) \]

Goal: sample \( \phi_i \sim p(\phi_i | x_{i\text{train}}, y_{i\text{train}}) \)

\[ p(\phi_i | x_{i\text{train}}, y_{i\text{train}}) \propto \int p(\theta)p(\phi_i | \theta)p(y_{i\text{train}} | x_{i\text{train}}, \phi_i) d\theta \]

\[ \Rightarrow \text{this is completely intractable!} \]

what if we knew \( p(\phi_i | \theta, x_{i\text{train}}, y_{i\text{train}}) \)?

\[ \Rightarrow \text{now sampling is easy! just use ancestral sampling!} \]

**key idea:** \( p(\phi_i | \theta, x_{i\text{train}}, y_{i\text{train}}) \approx \delta(\hat{\phi}_i) \)

this is extremely crude

but **extremely** convenient!

\[ \hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_{i\text{train}} | x_{i\text{train}}, \theta) \]

Sampling parameter vectors

$$\theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$$

**key idea:**

$$p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \delta(\hat{\phi}_i) \quad \hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$$

What does ancestral sampling look like?

1. $$\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$$
2. $$\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$$

Training the model – not so simple!

\[ \theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta) \]

**key idea:** \( p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \delta(\hat{\phi}_i) \quad \hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta) \)

Can’t do ancestral sampling anymore!

need to figure out \( p(\theta|x_i^{\text{test}}, y_i^{\text{test}}) \) (intractable inference problem)

approximate using \( q(\theta|x_i^{\text{test}}, y_i^{\text{test}}) \), train via variational bound

\[
\log p(x_i^{\text{train}}, y_i^{\text{train}}, x_i^{\text{test}}, y_i^{\text{test}} | \mu_\theta, \Sigma_\theta) \\
\leq E_{\theta \sim q} \left[ \log \int p(\theta)p(\phi_i|\theta)p(y_i^{\text{train}}|x_i^{\text{train}}, \phi_i)p(y_i^{\text{test}}|x_i^{\text{test}}, \phi_i) d\phi_i - \log q(\theta|x_i^{\text{test}}, y_i^{\text{test}}) \right] \\
\approx E_{\theta \sim q} \left[ \log p(\theta) + \log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta)) - \log q(\theta|x_i^{\text{test}}, y_i^{\text{test}}) \right] \\
= E_{\theta \sim q} \left[ \log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta)) \right] - D_{\text{KL}}(q(\theta|x_i^{\text{test}}, y_i^{\text{test}}) \| p(\theta))
\]
Training the model with amortized inference

need to figure out \( p(\theta|x_{i}^{\text{test}}, y_{i}^{\text{test}}) \) (intractable inference problem)

approximate using \( q(\theta|x_{i}^{\text{test}}, y_{i}^{\text{test}}) \), train via variational bound

\[
E_{\theta \sim q} \left[ \log p(y_{i}^{\text{test}}|x_{i}^{\text{test}}, \hat{\phi}_{i}(\theta)) \right] - D_{KL}(q(\theta|x_{i}^{\text{test}}, y_{i}^{\text{test}})||p(\theta))
\]

how do we represent \( q \)?

big neural network that outputs parameters? too intractable...

idea: \( q(\theta|x_{i}^{\text{test}}, y_{i}^{\text{test}}) = \mathcal{N}(\mu_{\theta} + \alpha \nabla_{\mu_{\theta}} \log p(y_{i}^{\text{test}}|x_{i}^{\text{test}}, \mu_{\theta}), \Sigma_{q}) \)

intuitive interpretation: cheat during meta-training by peeking at test set

...but minimize divergence against the prior

1. \( \theta \leftarrow \mu_{\theta} + \alpha \nabla_{\mu_{\theta}} \log p(y_{i}^{\text{test}}|x_{i}^{\text{test}}, \mu_{\theta}) + \epsilon \)

2. \( \phi_{i} \sim p(\phi_{i}|\theta, x_{i}^{\text{train}}, y_{i}^{\text{train}}) \approx \hat{\phi}_{i} = \theta + \alpha \nabla_{\theta} \log p(y_{i}^{\text{train}}|x_{i}^{\text{train}}, \theta) \)
need to figure out $p(\theta|x_i^{\text{test}}, y_i^{\text{test}})$ (intractable inference problem)
approximate using $q(\theta|x_i^{\text{test}}, y_i^{\text{test}})$, train via variational bound

$$E_{\theta \sim q} \left[ \log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta)) \right] - D_{\text{KL}}(q(\theta|x_i^{\text{test}}, y_i^{\text{test}})||p(\theta))$$

During meta-training:
1. $\theta \leftarrow \mu_\theta + \alpha \nabla_{\mu_\theta} \log p(y_i^{\text{test}}|x_i^{\text{test}}, \mu_\theta) + \epsilon$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha \nabla_{\theta} \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$

take gradient step on variational bound w.r.t. $\theta$, $\Sigma$, and $\Sigma_q$

During meta-testing:
1. $\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha \nabla_{\theta} \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$
PLATIPUS
Probabilistic LATent model for Incorporating Priors and Uncertainty in few-Shot learning

Ambiguous regression:

Ambiguous classification:

PLATIPUS
Probabilistic Latent model for Incorporating Priors and Uncertainty in few-Shot learning

<table>
<thead>
<tr>
<th>Ambiguous celebA (5-shot)</th>
<th>Accuracy</th>
<th>Coverage (max=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>69.26 ± 2.18%</td>
<td>1.00 ± 0.0</td>
</tr>
<tr>
<td>MAML + noise</td>
<td>54.73 ± 0.8 %</td>
<td>2.60 ± 0.12</td>
</tr>
<tr>
<td>PLATIPUS (ours)</td>
<td>69.97 ± 1.32 %</td>
<td>2.62 ± 0.11</td>
</tr>
</tbody>
</table>

Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference
Let’s Talk about Meta-Overfitting

• Meta learning requires task distributions
• When there are too few meta-training tasks, we can meta-overfit
• Specifying task distributions is hard, especially for meta-RL!
• Can we propose tasks automatically?

after MAML training

after 1 gradient step
A General Recipe for Unsupervised RL

Random Task Proposals

- Use randomly initialize discriminators for reward functions

\[ R(s, z) = \log p_D(z|s) \]

\( D \rightarrow \) randomly initialized network

- Important: Random functions over state space, **not** random policies

Random policy – exponential
Random reward – polynomial
Diversity-Driven Proposals

Task Reward for UML: \[ R(s, z) = \log p_D(z | s) \]

- Policy \( \rightarrow \) visit states which are discriminable
- Discriminator \( \rightarrow \) predict skill from state

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.
Examples of Acquired Tasks

Cheetah

Ant

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.
Does it work?

Meta-test performance with rewards

What about varying transition models?

Task distribution is more-or-less free!

online adaptation = few-shot learning

tasks are temporal
Meta-learning using either:
- Dynamics RNN (RBAC)
- MAML (GBAC)

**Online Adaptation**
- Store recent history
  \[ \{s_{t-M:t}, a_{t-M:t}\} \]
- Take action \( a_t \) via MPC

**Model Adaptation**
\[ \theta' = \psi(s_{t-M:t}, a_{t-M:t}, \theta_*) \]
Online Adaptation via Meta-Learning

with MAML

Clavera*, Nagabandi*, Abbeel, Levine, Finn, arxiv 2018
Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference
One-Shot Visual Imitation Learning

**Goal:** Given one visual demonstration of a new task, learn a policy.

Visual imitation is expensive.

Rahmanizadeh et al. ‘17  Zhang et al. ‘17
learns from raw pixels,
but requires many demonstrations

**Through meta-learning:** reuse data from other tasks/objects/environments

Finn*, Yu*, Zhang, Abbeel, Levine CoRL ‘17
One-Shot Visual Imitation Learning

**How?** Learn to learn the policy from one demonstration

- Meta-training tasks
  - training
  - validation
  - task 1
  - task 2
  - …

**Meta-test time:** Learn policy from one demo.

*Finn*, *Yu*, *Zhang*, *Abbeel*, *Levine* CoRL ’17
real-world placing \textit{from pixels}

input demo

resulting policy

Finn*, Yu*, Zhang, Abbeel, Levine CoRL ’17
Few-Shot Learning from Weak Supervision

Given one teleoperated demonstration:

Learn a policy.

Given a video of a human:

Classify new examples

Given 1 example of 5 classes:

Given 1 positive example:

Chelsea Finn, UC Berkeley

Grant, Finn, Peterson, Abbott, Levine, Darrell, Griffiths NIPS CIAI Workshop '17

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS '18
Learning to Learn from Weak Supervision

**meta-training**

\[ \min_{\theta} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta)) \]

**meta-test**

\[ \theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta) \]

What if the weakly supervised loss is unavailable?

**imitation loss**

\[ \mathcal{L} = \sum_t \| \pi_\theta(o_t) - a^*_t \|^2 \]

\[ \min_{\theta, \psi} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\psi}(\theta)) \]
placing *from pixels*

input demo

resulting policy

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS ’18
pick-and-place from pixels

input demo

resulting policy

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS ’18
pick-and-place *from pixels*

input demo

resulting policy

What should the robot do to improve?

Error in *inferring the goal* or *executing on it*?

Can we decouple *goal inference* & *policy learning*?
Meta-Learning Approach

How? Learn to learn the policy from one demonstration

Meta-training tasks

Meta-test time: Learn policy from one demo.
Meta-Learning Approach

**How?** Learn to **infer the objective** from one demonstration

<table>
<thead>
<tr>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>task 1</td>
<td></td>
</tr>
<tr>
<td>task 2</td>
<td></td>
</tr>
</tbody>
</table>

Meta-test time:

Learn **objective** from one demo.
Learn policy by optimizing reward.
Given a few observations of reaching the goal:

Infer an objective:

(task 1)

(task 2)

(then run RL or planning)

(task N)

meta-training tasks

Xie, Singh, Levine, Finn ’18 (under review)
Given a few observations of reaching the goal:

Infer an objective:

(then run RL or planning)

MAML with supervised loss:

\[
\min_{\theta} \sum_{\text{task } i} \mathcal{L}_{\text{test}}^i(\theta - \alpha \nabla_\theta \mathcal{L}_{\text{train}}^i(\theta))
\]

both positive & negatives

only positive examples

another form of weak supervision!

Test time: only need positive examples

\[
\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(\theta)
\]

Xie, Singh, Levine, Finn ’18 (under review)
Novel Object Positioning via Visual Planning

Given 5 examples of success

Visual MPC with learned objective

infer goal classifier

visual MPC w.r.t. goal classifier

Xie, Singh, Levine, Finn '18 (under review)
Novel Object Positioning via Visual Planning

Visual MPC with learned objective

Given 5 examples of success
Rope Manipulation via Reinforcement Learning

Given 5 examples of success

RL policy with learned objective

infer goal classifier

visual RL w.r.t. goal classifier

Xie, Singh, Levine, Finn ’18 (under review)
Rope Manipulation via Reinforcement Learning

Given 5 examples of success

RL policy with learned objective

Xie, Singh, Levine, Finn ’18 (under review)
Questions?

Meta-Learning Foundations
Finn & Levine, *Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm*. ICLR ‘18
Grant, Finn, Levine, Darrell, Griffths, *Recasting Gradient-Based Meta-Learning as Hierarchical Bayes*. ICLR ‘18

Meta-Learning for Control
Xu*, Ratner, Dragan, Levine, Finn, *Learning a Prior over Intent via Meta-Inverse Reinforcement Learning*. under review ‘18
Xie, Singh, Levine, Finn, *Few-Shot Goal Inference for Visuomotor Learning and Planning*. under review ‘18