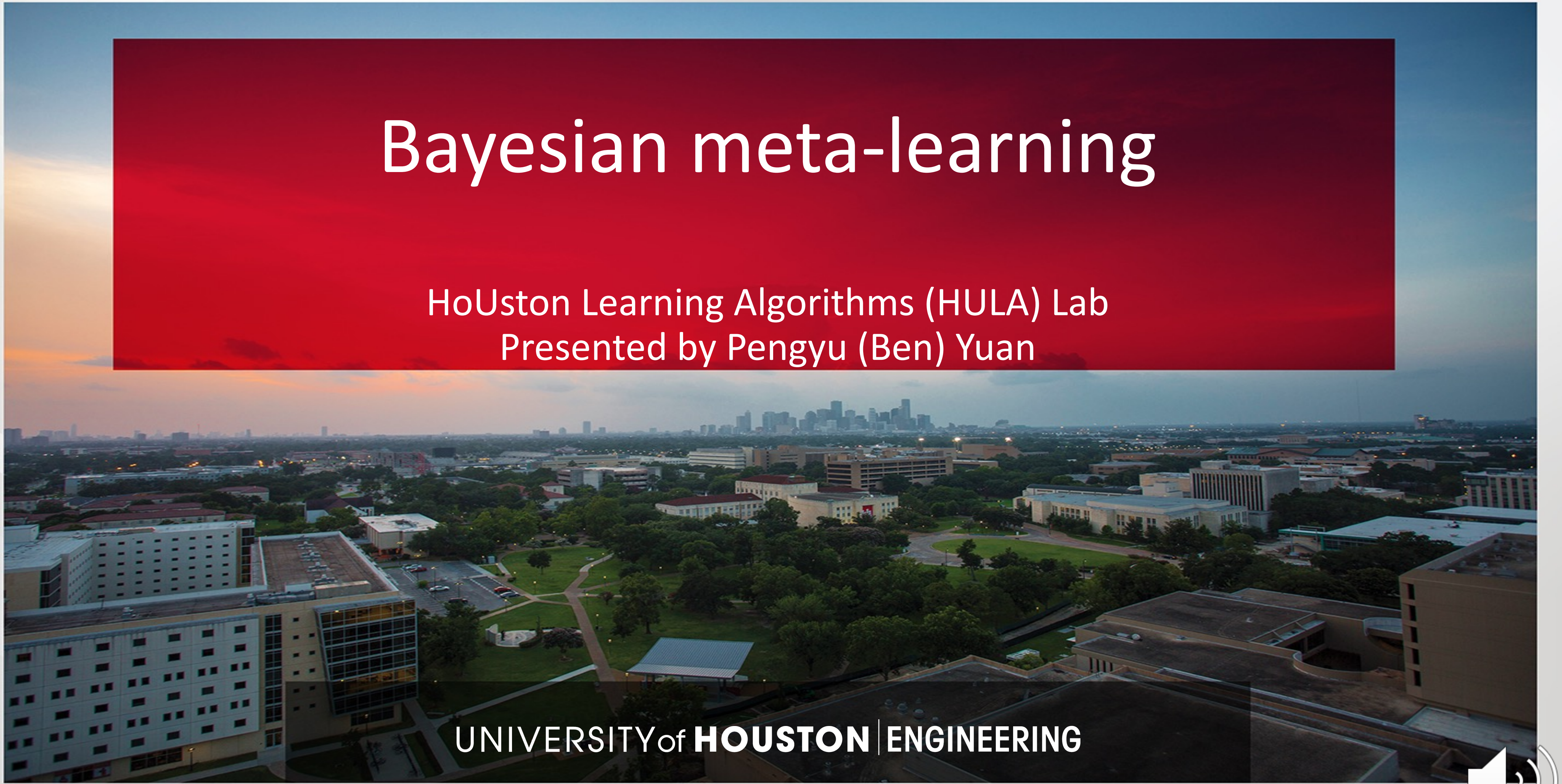
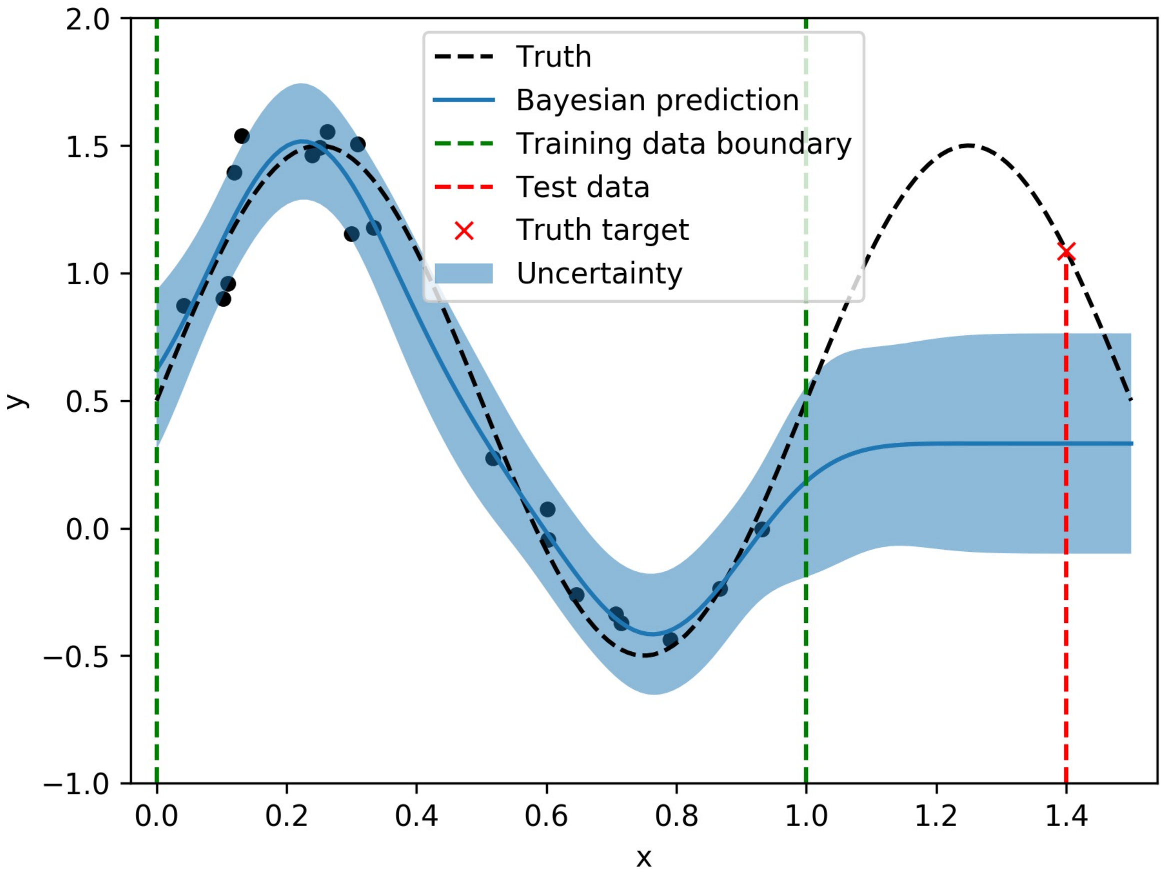
Bayesian meta-learning

HoUston Learning Algorithms (HULA) Lab
Presented by Pengyu (Ben) Yuan

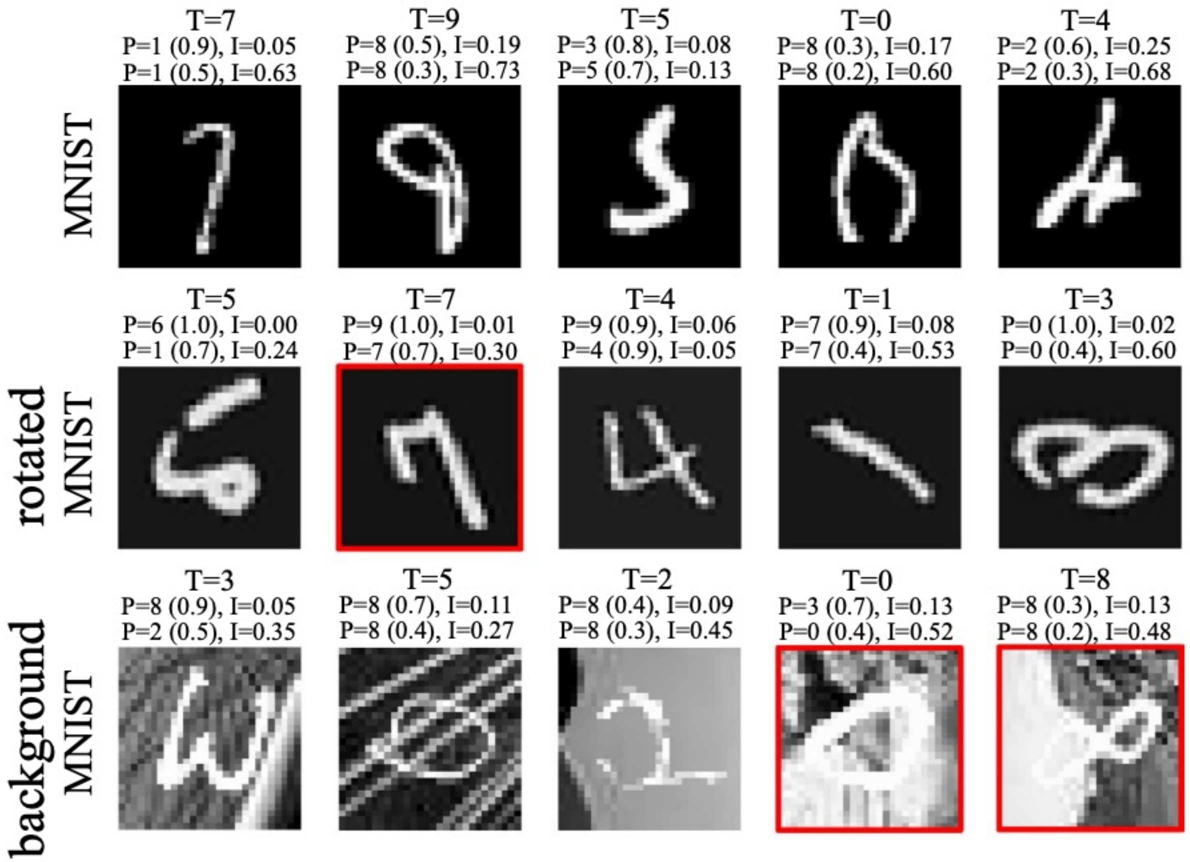


UNIVERSITY of HOUSTON | ENGINEERING

Uncertainty exists everywhere



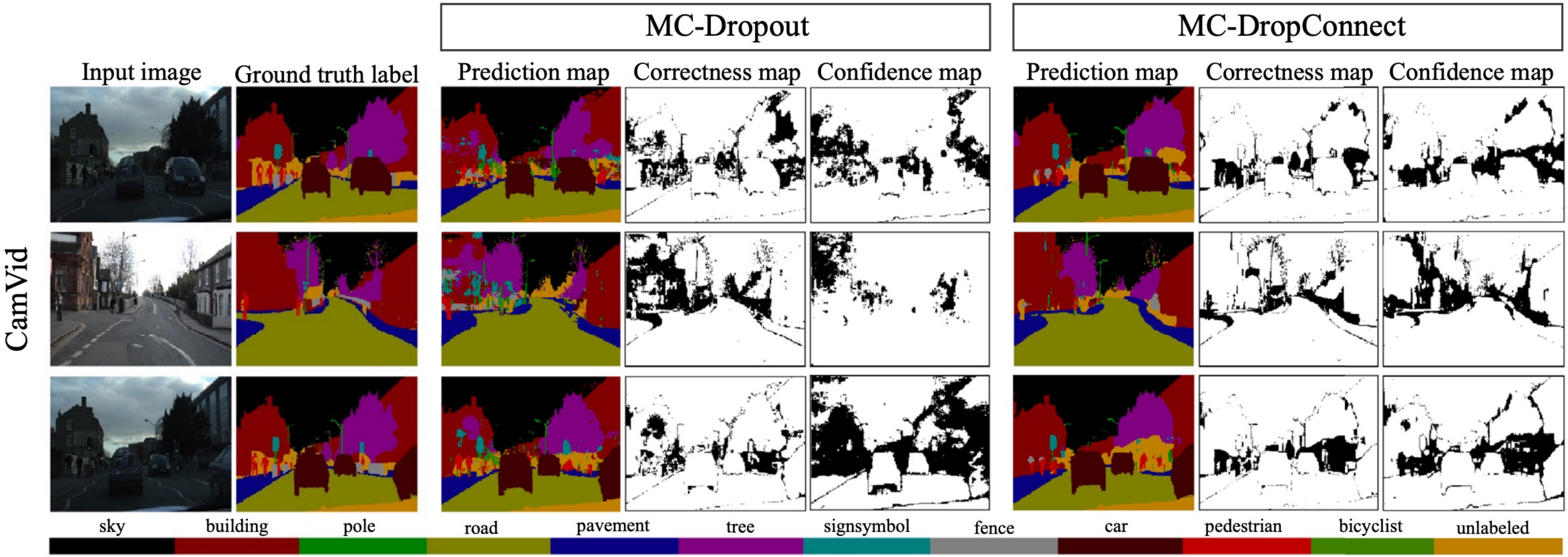
Regression problem: High uncertainty when outside of training distribution



Classification problem: Correcting wrong prediction when use uncertainty

https://www.nature.com/articles/s41598-0[illegible]34854-x

Uncertainty exists everywhere



Segmentation problem: High correlation between uncertainty and correctness

https://www.nature.com/articles/s41598-0[illegible]348[illegible]4-x

Deterministic meta-learning

- Learner's parameter (deterministic)

$$p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right) \approx \delta(\phi_i^*)$$

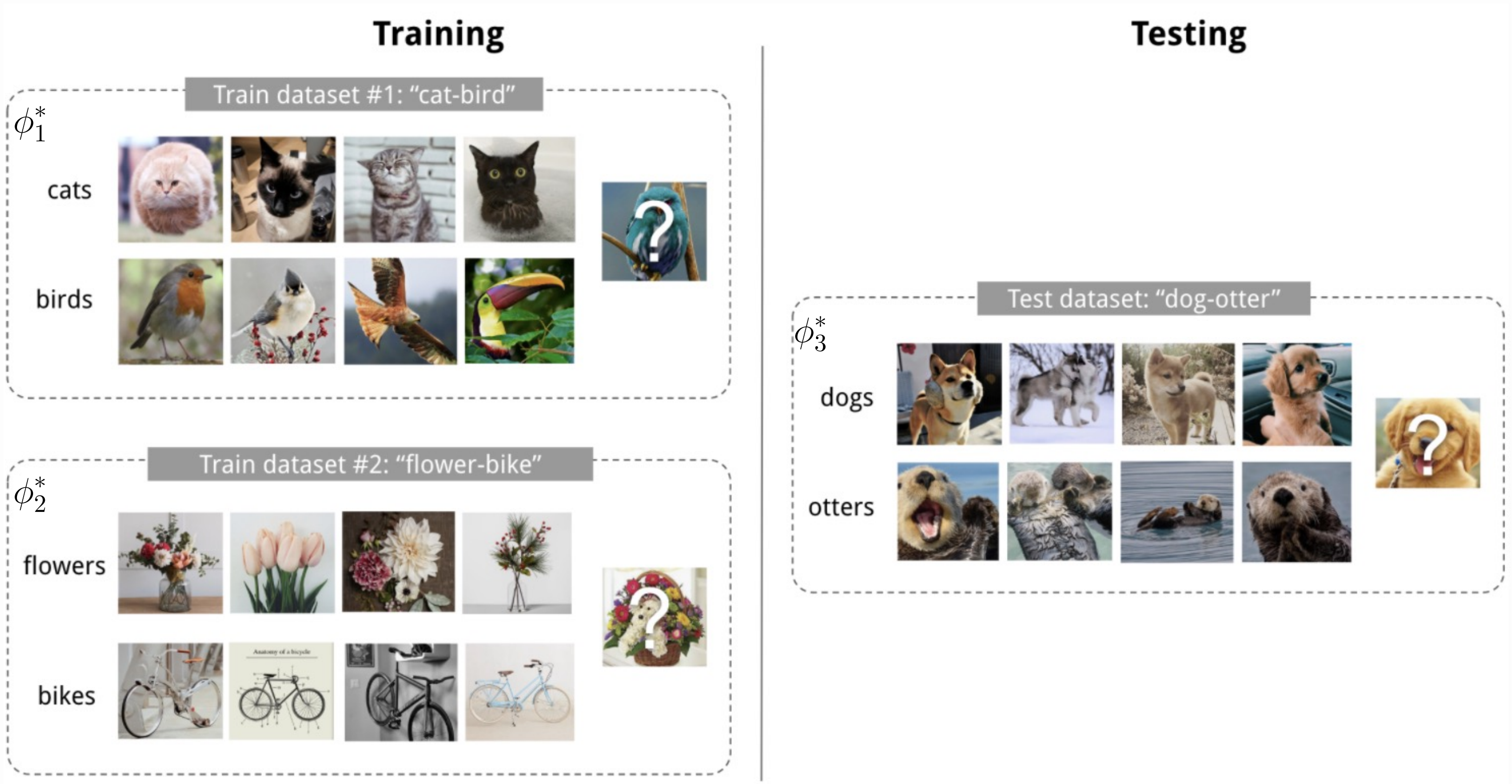
where

$$\phi_i^* = \arg\max_{\phi_i} \log p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$$

- Problems:
  - ϕ_i^* is a point estimate (MAP)
  - There is no uncertainty in the prediction:

    $$y^{\text{te}} = g_{\phi^*}(x^{\text{te}})$$

    where g is the learner's network
  - Few shot learning is ambiguous, easily **overfitting**



Training
Testing
Train dataset #1: "cat-bird"
ϕ_1^*
cats
birds
?
Train dataset #2: "flower-bike"
ϕ_2^*
flowers
bikes
?
Test dataset: "dog-otter"
ϕ_3^*
dogs
otters
?

ϕ_i^* is a set of classifier parameters for $\mathcal{D}_i$

Can we get a distribution of $p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$?

So

$$p(y^{\text{te}}|x^{\text{te}}, \mathcal{D}^{\text{tr}}, \theta^*) = \int_{\Phi} p(y^{\text{te}}|x^{\text{te}}, \phi)p(\phi|\mathcal{D}^{\text{tr}}, \theta^*)d\phi$$

https://lilianweng.github.io/lil-log/2018/11/30/meta-learning.html

Deterministic meta-learning

- Meta parameter (deterministic)

$$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right) \approx \delta(\theta^*)$$

where

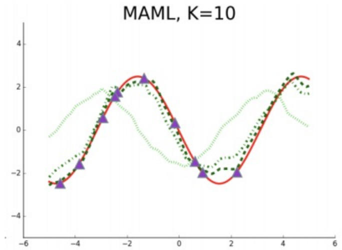
$$\theta^* = \arg\max_{\theta} \log p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$$

- Problems:
  - θ^* is also a point estimate (MAP)
  - When the number of tasks is small, there is high uncertainty in the meta parameters. Leads to **meta-overfitting**
  - Learner's parameters are affected by meta parameters:

$$p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta\right)$$

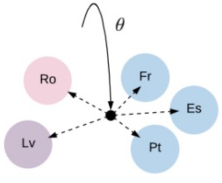
Thus, it can also affect the distribution of the prediction.

What information might $\boldsymbol{\theta}$ contain…



…in a toy sinusoid problem?

$\boldsymbol{\theta}$ corresponds to family of sinusoid functions (everything but phase and amplitude)



(c) Meta Learning

…in multi-language machine translation?

$\boldsymbol{\theta}$ corresponds to the family of all language pairs

θ **is the shared latent information from** $\mathcal{D}_{\text{meta-train}}$

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learn 2020.pdf

Why Bayesian in meta-learning?

Bayesian method can:

- give us a distribution over prediction
- prevent overfitting problem
- update model gradually by using online learning

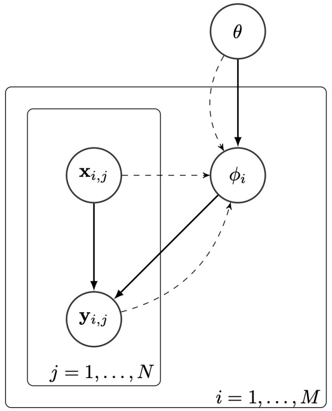
In meta-learning scenario, it can:

- Learn **safety-critical** few-shot model **(especially in medical imaging)**
- Learn to **actively annotate** new samples **(active learning)**
- Learn to **explore** in meta reinforcement learning

Bayesian tools

How ?

- Learn distribution over learner's parameters: $\delta(\phi_i^*) \longrightarrow p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta^*\right)$
- Learn distribution over meta parameters: $\delta(\theta^*) \longrightarrow p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$
- Can be either one or both



ϕ_i **as the latent variable**

Bayesian toolboxes:

- **Latent variable models + variational inference**

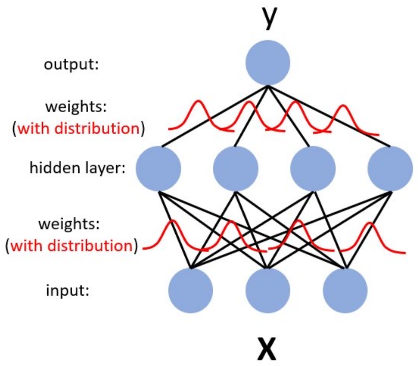
  approximate likelihood of latent variable model with variational lower bound
- **Bayesian ensembles**

  particle-based representation: train separate models on bootstraps of the data
- **Bayesian neural networks**

  explicit distribution over the space of network parameters
- ...

Single donkey

An ensemble of donkey



Bayesian neural network

https://openreview.net/pdf?id=rkgpy3C5tX
http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf
https://www.researchgate.net/publication/328757994_A_Batched_Scalable_Multi-Objective_Bayesian_Optimization_Algorithm

Outline

- Introduction
  - Why Bayesian meta-learning?
  - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
  - Amortized variational inference
    - Black-box
    - Optimization
  - Bayesian ensembles
  - Bayesian neural networks
- Bayesian meta-learning evaluation
  - Qualitative visualization
  - Quantitative evaluation
  - Active-learning evaluation

Outline

- Introduction
  - Why Bayesian meta-learning?
  - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
  - Amortized variational inference
    - Black-box
    - Optimization
  - Bayesian ensembles
  - Bayesian neural networks
- Bayesian meta-learning evaluation
  - Qualitative visualization
  - Quantitative evaluation
  - Active-learning evaluation

The evidence lower bound (ELBO)

- What is ELBO?

  It is an optimization function used in variational inference.

$$q(\phi) = \arg\max_q ELBO \qquad \text{where} \qquad ELBO = \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi$$

ϕ – parameters

$\mathcal{D}$ – observations

$q(\phi)$ – variational distribution

The evidence lower bound (ELBO)

- What is ELBO?

  It is an optimization function used in variational inference.

$$q(\phi) = \arg\max_q ELBO \quad \text{where} \quad ELBO = \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi$$

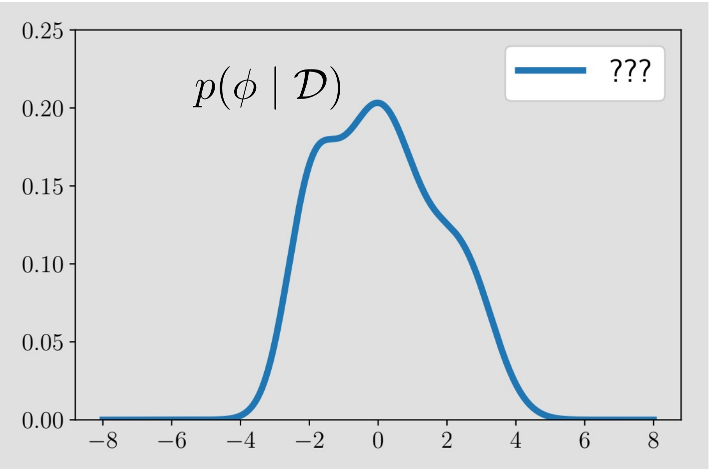
ϕ – parameters

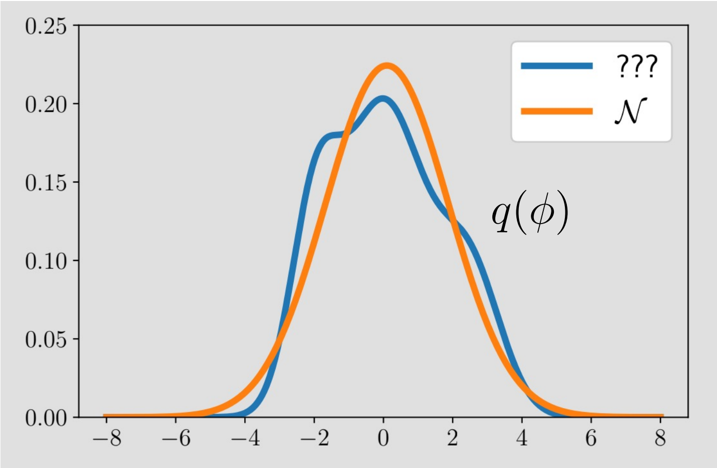
$\mathcal{D}$ – observations

$q(\phi)$ – variational distribution

- What is variational inference (VI)?

  Approximating a posterior distribution with some easy to manipulate distribution like the Gaussian





$$q(\phi) \rightarrow p(\phi|\mathcal{D})$$

The evidence lower bound (ELBO)

- What is ELBO?

It is an optimization function used in variational inference.

$$q(\phi) = \arg\max_{q} ELBO \qquad \text{where} \qquad ELBO = \int_{\Phi} q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi$$

ϕ – parameters

$\mathcal{D}$ – observations

$q(\phi)$ – variational distribution

- What is variational inference (VI)?

Approximating a posterior distribution with some easy to manipulate distribution like the Gaussian

$$q(\phi) \overset{\text{Approx.}}{\rightarrow} p(\phi|\mathcal{D})$$

- Why do we need VI?

It is intractable to calculate the true posterior distribution

$$p(\phi \mid \mathcal{D}) = \frac{p(\mathcal{D}, \phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi)p(\phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi)p(\phi)}{\underline{\int_{\Phi} p(\mathcal{D} \mid \phi)p(\phi)d\phi}}$$

because it is impossible to consider all configurations ϕ of the neural network.

$p(\phi \mid \mathcal{D})$ – true posterior distribution

$p(\mathcal{D} \mid \phi)$ – likelihood

$p(\phi)$ – prior distribution

The evidence lower bound (ELBO)

- What is ELBO?

  It is an optimization function used in variational inference.

$$q(\phi) = \arg\max_{q} ELBO \qquad \text{where} \qquad ELBO = \int_{\Phi} q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi$$

ϕ	– parameters
$\mathcal{D}$	– observations
$q(\phi)$	– variational distribution

- What is variational inference (VI)?

  Approximating a posterior distribution with some easy to manipulate distribution like the Gaussian

$$q(\phi) \overset{\text{Approx.}}{\rightarrow} p(\phi|\mathcal{D})$$

- Why do we need VI?

  It is intractable to calculate the true posterior distribution

$$p(\phi \mid \mathcal{D}) = \frac{p(\mathcal{D}, \phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi) p(\phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi) p(\phi)}{\underline{\int_{\Phi} p(\mathcal{D} \mid \phi) p(\phi) d\phi}}$$

- Why do we need posterior distribution?

  To get distribution (uncertainty) over predictions:

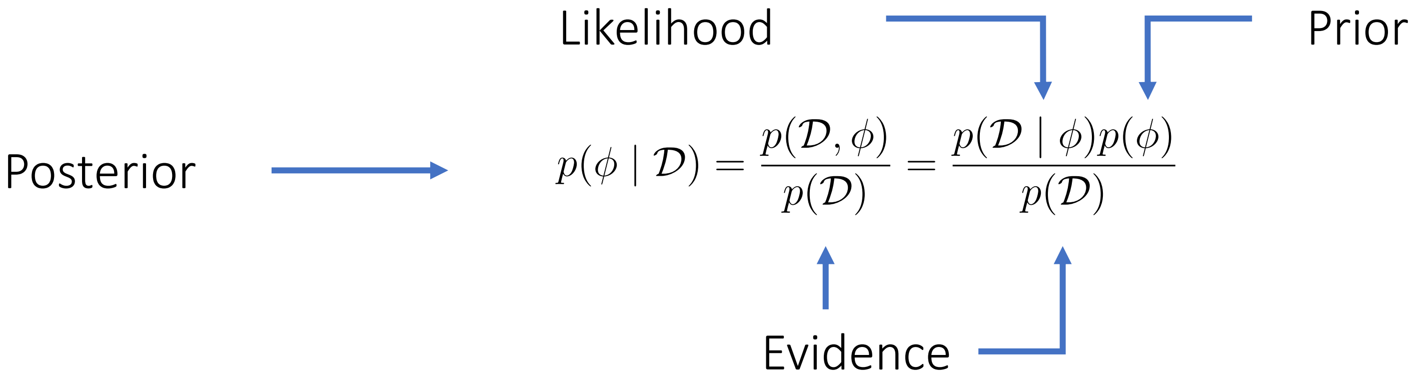
$$p(y^{\text{te}}|x^{\text{te}}, \mathcal{D}^{\text{tr}}) = \int_{\Phi} p(y^{\text{te}}|x^{\text{te}}, \phi) p(\phi|\mathcal{D}^{\text{tr}}) d\phi$$

$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution

The evidence lower bound (ELBO)

- Why optimizing the ELBO can help to approximate the true posterior distribution?

  Look at the Bayesian rule:



Likelihood

Prior

Posterior

$$p(\phi \mid \mathcal{D}) = \frac{p(\mathcal{D}, \phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi)p(\phi)}{p(\mathcal{D})}$$

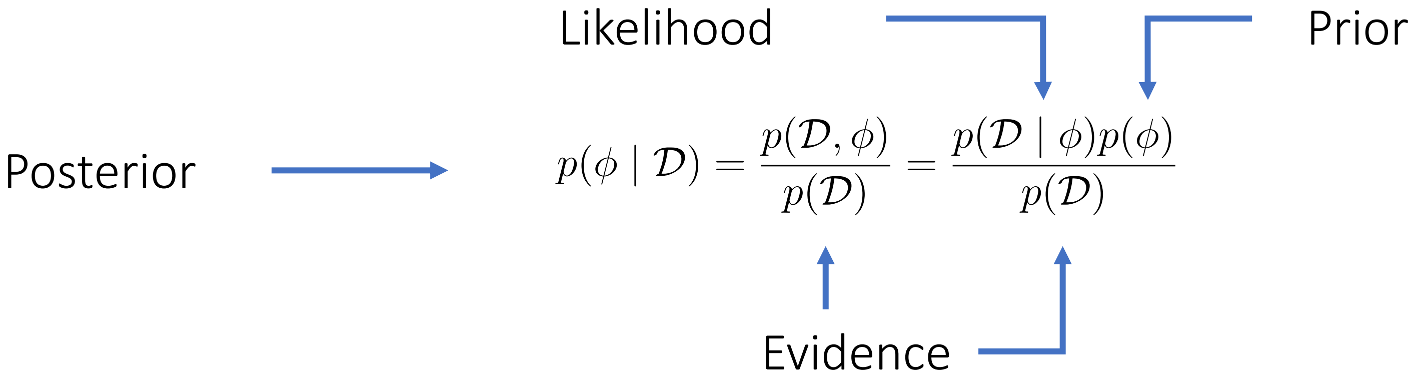
Evidence

$q(\phi)$	– variational distribution
$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution
$p(\mathcal{D})$	– evidence

The evidence lower bound (ELBO)

- Why optimizing the ELBO can help to approximate the true posterior distribution?

Look at the Bayesian rule:



$$p(\phi \mid \mathcal{D}) = \frac{p(\mathcal{D}, \phi)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \phi)p(\phi)}{p(\mathcal{D})}$$

Log evidence:

$$\ln p(\mathcal{D}) = ELBO + \underbrace{KL(q(\phi)||p(\phi|\mathcal{D}))}_{\text{Difference between the variational distribution and true posterior distribution}} \geqslant ELBO \qquad KL(\cdot||\cdot) \geqslant 0$$

Evidence is fixed by data, thus

$$q(\phi) = \arg\min_q KL(q(\phi)||p(\phi|\mathcal{D})) = \arg\max_q ELBO$$

$q(\phi)$	– variational distribution
$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution
$p(\mathcal{D})$	– evidence

The evidence lower bound (ELBO)

- Let’s get a closer look at ELBO

$$\max_q ELBO = \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi$$

$q(\phi)$	– variational distribution
$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution
$p(\mathcal{D})$	– evidence

The evidence lower bound (ELBO)

- Let's get a closer look at ELBO

$$\begin{aligned}\max_q ELBO &= \int_\Phi q(\phi)\ln\frac{p(\mathcal{D},\phi)}{q(\phi)}d\phi\\ &= \int_\Phi q(\phi)\ln\frac{p(\mathcal{D}|\phi)p(\phi)}{q(\phi)}d\phi\end{aligned}$$

$q(\phi)$	– variational distribution
$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution
$p(\mathcal{D})$	– evidence

The evidence lower bound (ELBO)

- Let's get a closer look at ELBO

$$
\begin{aligned}
\max_q ELBO &= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}|\phi) p(\phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln p(\mathcal{D} \mid \phi) - \int_\Phi q(\phi) \ln \frac{q(\phi)}{p(\phi)} d\phi
\end{aligned}
$$

$q(\phi)$ – variational distribution

$p(\phi \mid \mathcal{D})$ – true posterior distribution

$p(\mathcal{D} \mid \phi)$ – likelihood

$p(\phi)$ – prior distribution

$p(\mathcal{D})$ – evidence

The evidence lower bound (ELBO)

- Let’s get a closer look at ELBO

$$
\begin{aligned}
\max_q ELBO &= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}|\phi)p(\phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln p(\mathcal{D} \mid \phi) - \int_\Phi q(\phi) \ln \frac{q(\phi)}{p(\phi)} d\phi \\
&= \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))
\end{aligned}
$$

$q(\phi)$ – variational distribution

$p(\phi \mid \mathcal{D})$ – true posterior distribution

$p(\mathcal{D} \mid \phi)$ – likelihood

$p(\phi)$ – prior distribution

$p(\mathcal{D})$ – evidence

The evidence lower bound (ELBO)

- Let’s get a closer look at ELBO

$$
\begin{aligned}
\max_q ELBO &= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}, \phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln \frac{p(\mathcal{D}|\phi) p(\phi)}{q(\phi)} d\phi \\
&= \int_\Phi q(\phi) \ln p(\mathcal{D} \mid \phi) - \int_\Phi q(\phi) \ln \frac{q(\phi)}{p(\phi)} d\phi \\
&= \underbrace{\mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)]}_{\text{Samples from } q(\phi) \text{ to perform original tasks}} - \underbrace{KL(q(\phi) \| p(\phi))}_{\text{Regularization term}}
\end{aligned}
$$

$q(\phi)$	– variational distribution
$p(\phi \mid \mathcal{D})$	– true posterior distribution
$p(\mathcal{D} \mid \phi)$	– likelihood
$p(\phi)$	– prior distribution
$p(\mathcal{D})$	– evidence

The evidence lower bound (ELBO)

- More about ELBO

$$\ln \underbrace{p(\mathcal{D})}_{\text{evidence}} \geqslant ELBO = \mathbb{E}_{\underbrace{q(\phi)}_{\text{variational distribution}}}[\ln \underbrace{p(\mathcal{D} \mid \phi)}_{\text{likelihood}}] - KL(\underbrace{q(\phi)}_{\text{variational distribution}} \| \underbrace{p(\phi)}_{\text{prior distribution}})$$

The evidence lower bound (ELBO)

- More about ELBO

$$\ln \underbrace{p(\mathcal{D})}_{\text{evidence}} \geqslant ELBO = \mathbb{E}_{\underbrace{q(\phi)}_{\text{variational distribution}}}[\ln \underbrace{p(\mathcal{D} \mid \phi)}_{\text{likelihood}})] - KL(\underline{q(\phi)} \| \underbrace{p(\phi)}_{\text{prior distribution}})$$

variational distribution can be any form:

$$q(\phi) \Rightarrow q(\phi|\mathcal{D}),\ q(\phi|\theta)$$

For example $q(\phi|\theta) \xrightarrow{\text{Approx.}} p(\phi|\mathcal{D})$:

$$\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{\underline{q(\phi|\theta)}}[\ln p(\mathcal{D} \mid \phi)] - KL(\underline{q(\phi|\theta)} \| p(\phi))$$

The evidence lower bound (ELBO)

- More about ELBO

$$\underbrace{\ln p(\mathcal{D})}_{\text{evidence}} \geqslant ELBO = \mathbb{E}_{\underbrace{q(\phi)}_{\text{variational distribution}}}[\underbrace{\ln p(\mathcal{D} \mid \phi)}_{\text{likelihood}}] - KL(\underbrace{q(\phi)}_{\text{variational distribution}} \| \underbrace{p(\phi)}_{\text{prior distribution}})$$

variational distribution can be any form:

$$q(\phi) \Rightarrow q(\phi|\mathcal{D}),\ q(\phi|\theta)$$

For example $q(\phi|\theta) \overset{\text{Approx.}}{\rightarrow} p(\phi|\mathcal{D})$: $\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{\underline{q(\phi|\theta)}}[\ln p(\mathcal{D} \mid \phi)] - KL(\underline{q(\phi|\theta)} \| p(\phi))$

posterior can be conditioned on other variables:

$$p(\phi|\mathcal{D}) \Rightarrow p(\phi|\mathcal{D}, \theta)$$

For example $q(\phi) \overset{\text{Approx.}}{\rightarrow} p(\phi|\mathcal{D}, \theta)$: $\ln \underline{p(\mathcal{D}|\theta)} \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln \underline{p(\mathcal{D} \mid \phi, \theta)}] - KL(q(\phi) \| \underline{p(\phi|\theta)})$

The evidence lower bound (ELBO)

- Problem of ELBO:

$$\ln \underbrace{p(\mathcal{D})}_{\text{evidence}} \geqslant ELBO = \mathbb{E}_{\underbrace{q(\phi)}_{\text{variational distribution}}}[\ln \underbrace{p(\mathcal{D} \mid \phi)}_{\text{likelihood}}] - KL(\underbrace{q(\phi)}_{\text{variational distribution}} \| \underbrace{p(\phi)}_{\text{prior distribution}})$$

1. Cannot optimizing $q(\phi)$ directly, reparameterization trick is required.

- Variational distribution has variational parameters λ , i.e. $q(\phi) = q_\lambda(\phi)$ $\lambda = \{\mu, \sigma^2\}$
- It is in general difficult to calculate the derivative $\nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}$

https://gregorygundersen.com/blog/2018/04/29/reparameterization/

The evidence lower bound (ELBO)

- Problem of ELBO:

$$\ln \underbrace{p(\mathcal{D})}_{\text{evidence}} \geqslant ELBO = \mathbb{E}_{\underbrace{q(\phi)}_{\text{variational distribution}}}[\underbrace{\ln p(\mathcal{D} \mid \phi)}_{\text{likelihood}}] - KL(q(\phi) \| \underbrace{p(\phi)}_{\text{prior distribution}})$$

1. Cannot optimizing $q(\phi)$ directly, reparameterization trick is required.

- Variational distribution has variational parameters λ , i.e. $q(\phi) = q_\lambda(\phi)$ $\lambda = \{\mu, \sigma^2\}$
- It is in general difficult to calculate the derivative $\nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}$

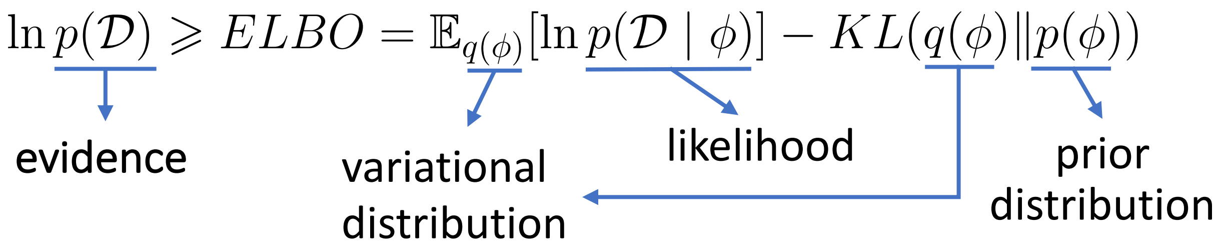
To see that, set $f_\lambda(\phi, \mathcal{D}) = \ln p(\mathcal{D} \mid \phi)$, then

$$\begin{aligned}
\nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}\left[f_\lambda(\phi, \mathcal{D})\right] &= \nabla_\lambda\left[\int_\Phi q_\lambda(\phi) f_\lambda(\phi, \mathcal{D}) d\phi\right] \\
&= \int_\Phi \nabla_\lambda\left[q_\lambda(\phi) f_\lambda(\phi, \mathcal{D})\right] d\phi \\
&= \int_\Phi f_\lambda(\phi, \mathcal{D}) \nabla_\lambda q_\lambda(\phi) d\phi + \int_\Phi q_\lambda(\phi) \nabla_\lambda f_\lambda(\phi, \mathcal{D}) d\phi \\
&= \underbrace{\int_\Phi f_\lambda(\phi, \mathcal{D}) \nabla_\lambda q_\lambda(\phi) d\phi}_{\text{What about this?}} + \mathbb{E}_{q_\lambda(\phi)}\left[\nabla_\lambda f_\lambda(\phi, \mathcal{D})\right]
\end{aligned}$$

https://gregorygundersen.com/blog/2018/04/29/reparameterization/

The evidence lower bound (ELBO)

- Problem of ELBO:



$$\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$$

1. Cannot optimizing $q(\phi)$ directly, reparameterization trick is required.

- Variational distribution has variational parameters λ , i.e. $q(\phi) = q_\lambda(\phi)$ $\lambda = \{\mu, \sigma^2\}$
- It is in general difficult to calculate the derivative $\nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}$

To see that, set $f_\lambda(\phi, \mathcal{D}) = \ln p(\mathcal{D} \mid \phi)$, then

$$\phi \sim N(\mu, \sigma^2) \xrightarrow{\text{reparameterization}} \begin{gathered} \phi = \mu + \sigma\epsilon, \text{ where } \epsilon \sim N(0, \mathrm{I}) \\ f_\lambda(\phi, \mathcal{D}) = f(g_\lambda(\epsilon, \mathcal{D})) \end{gathered}$$

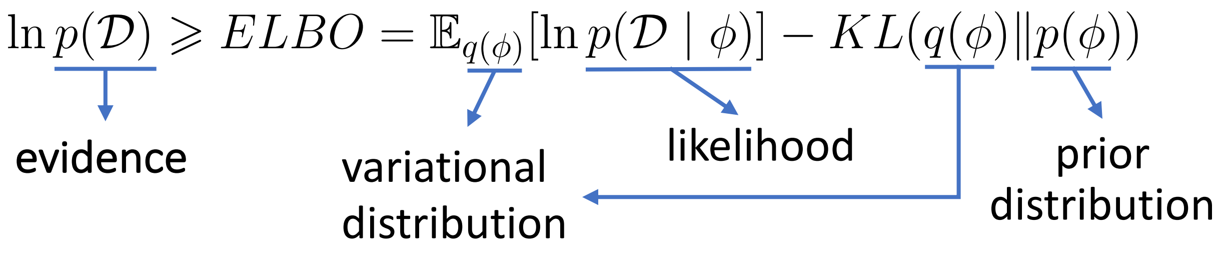
$$\nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}\left[f_\lambda(\phi, \mathcal{D})\right] = \underbrace{\int_\Phi f_\lambda(\phi, \mathcal{D}) \nabla_\lambda q_\lambda(\phi) d\phi}_{\text{What about this?}} + \mathbb{E}_{q_\lambda(\phi)}\left[\nabla_\lambda f_\lambda(\phi, \mathcal{D})\right]$$

$$\begin{aligned} \nabla_\lambda \mathbb{E}_{q_\lambda(\phi)}\left[f_\lambda\left(\phi, \mathcal{D}\right)\right] &= \nabla_\lambda \mathbb{E}_{p(\epsilon)}[f(g_\lambda(\epsilon, \mathcal{D}))] \\ &= \mathbb{E}_{p(\epsilon)}[\nabla_\lambda f(g_\lambda(\epsilon, \mathcal{D}))] \end{aligned}$$

https://gregorygundersen.com/blog/2018/04/29/reparameterization/

The evidence lower bound (ELBO)

- Problem of ELBO:



$$\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$$

evidence

variational distribution

likelihood

prior distribution

2. Can only model Gaussian variational distribution $q(\phi)$

- Variational distribution needs to be simple
- Reparameterization trick gives us Gaussian distribution
- KL divergence has analytic solution when both distributions are Gaussian

The evidence lower bound (ELBO)

Basic ideas: maximize ELBO to use $q(\phi)$ to approximate $p(\phi \mid \mathcal{D})$

- During training:
  1. Sample model parameters from $q(\phi)$
  2. Maximize the likelihood of the observation $p(\mathcal{D} \mid \phi)$ while minimize the gap between the $q(\phi)$ and $p(\phi)$

$$\max \ \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$$

- During test:
  1. Sample model parameters from $q(\phi)$
  2. Use it as the true posterior distribution for prediction

$$\begin{aligned} p(y^{\mathrm{te}}|x^{\mathrm{te}}, \mathcal{D}^{\mathrm{tr}}) &= \int_{\Phi} p(y^{\mathrm{te}}|x^{\mathrm{te}}, \phi) p(\phi|\mathcal{D}^{\mathrm{tr}}) d\phi \\ &\approx \int_{\Phi} p(y^{\mathrm{te}}|x^{\mathrm{te}}, \phi) q(\phi) d\phi \end{aligned}$$

Outline

- Introduction
  - Why Bayesian meta-learning?
  - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
  - Amortized variational inference
    - Black-box
    - Optimization
  - Bayesian ensembles
  - Bayesian neural networks
- Bayesian meta-learning evaluation
  - Qualitative visualization
  - Quantitative evaluation
  - Active-learning evaluation

Amortized variational inference

For dataset $\mathcal{D}_i$, the posterior distribution we need is: $p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right)$

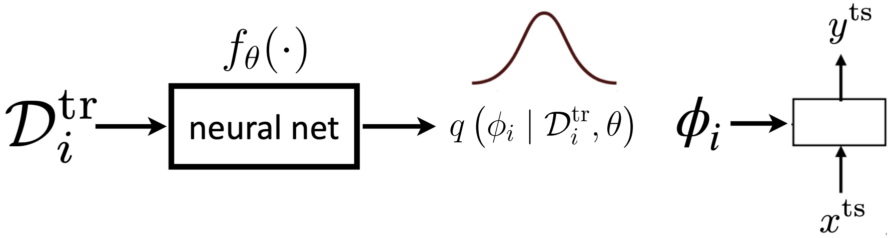
The variational distribution we used: $q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right) \sim N(\mu_i, \sigma_i^2)$

Amortized Variational Inference

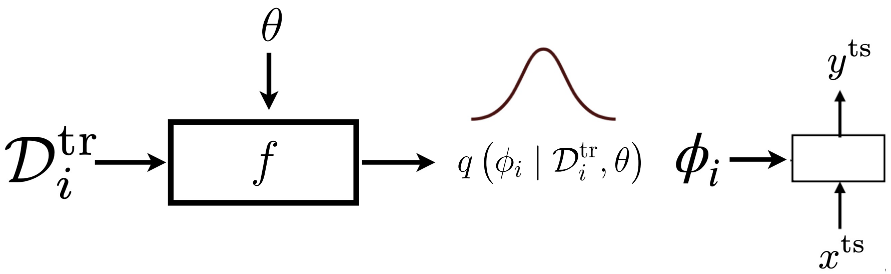
1. Introduce a parameterized model or function that outputs the variational parameters of the approximate posteriors.

$$\lambda = \{\mu_i, \sigma_i^2\} = f_\theta(\mathcal{D}_i^{\mathrm{tr}})$$

$$\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\mathrm{tr}}, \theta)$$



$\mathcal{D}_i^{\mathrm{tr}}$
$f_\theta(\cdot)$
neural net
$q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right)$
ϕ_i
y^{ts}
x^{ts}



θ
$\mathcal{D}_i^{\mathrm{tr}}$
f
$q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right)$
ϕ_i
y^{ts}
x^{ts}

2. Variational parameter λ is determined by meta parameter θ, thus optimizing variational parameter is the same as optimizing the meta parameter. This optimization is done by doing gradient descent on the **loss function for the variational inference**.

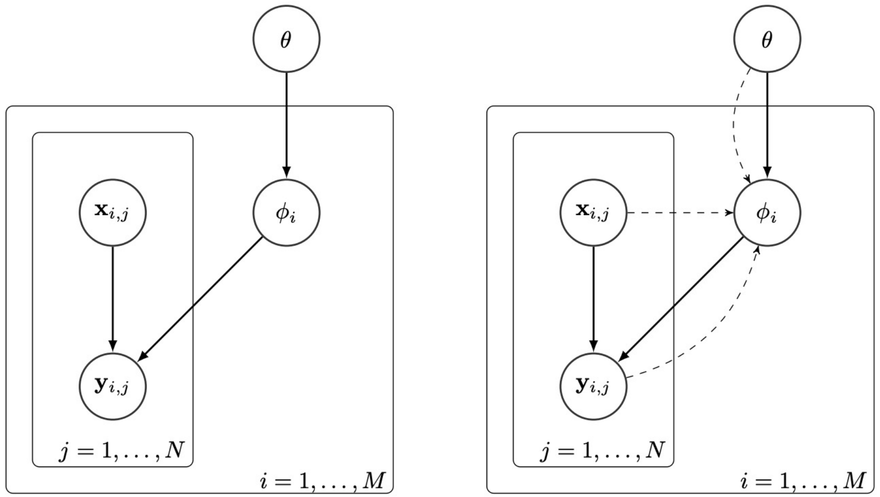
http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Amortized variational inference

- What is the loss function for the variational inference in meta-learning?

Variational inference: $\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$

Because of the meta-learning, we have additional meta-parameter θ



By replacing:

$$q(\phi) \Rightarrow q(\phi|\mathcal{D}^{\text{tr}}, \theta) \qquad\qquad p(\phi|\mathcal{D}) \Rightarrow p(\phi|\mathcal{D}, \theta)$$

We have:

$$\max_{\theta} ELBO = \max_{\theta} \ \mathbb{E}_{\underline{q(\phi|\mathcal{D}^{\text{tr}},\theta)}}[\ln p(\mathcal{D}|\phi, \underline{\theta})] - KL(\underline{q(\phi|\mathcal{D}^{\text{tr}}, \theta)} \| p(\phi|\underline{\theta}))$$

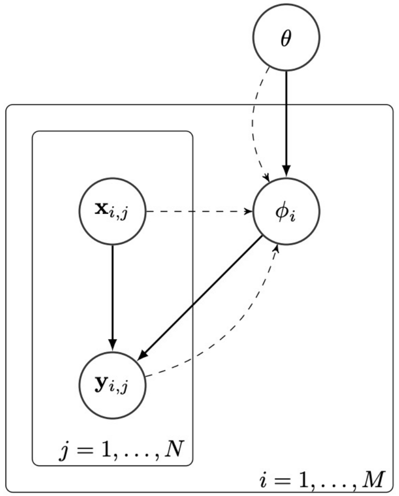
https://openreview.net/pdf?id-r py3C5tX

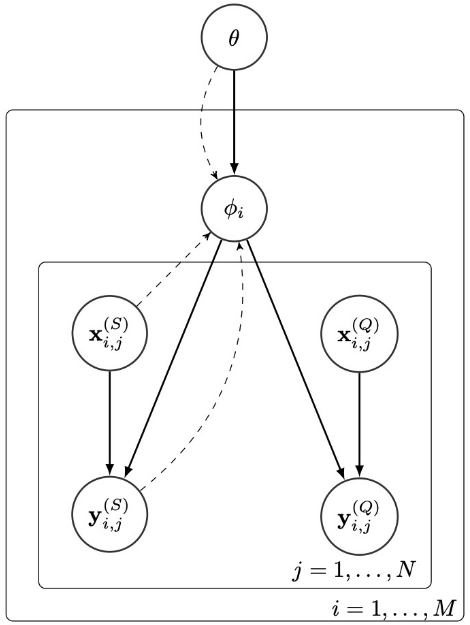
Amortized variational inference

- What is the loss function for the variational inference in meta-learning?

Variational inference: $\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$

Because of the meta-learning, we have additional meta-parameter θ





By replacing: $q(\phi) \Rightarrow q(\phi | \mathcal{D}^{\mathrm{tr}}, \theta)$ $\quad p(\phi | \mathcal{D}) \Rightarrow p(\phi | \mathcal{D}, \theta)$

We have: $\max_{\theta} ELBO = \max_{\theta} \mathbb{E}_{q(\phi | \mathcal{D}^{\mathrm{tr}}, \theta)}[\ln p(\mathcal{D} | \phi, \theta)] - KL(q(\phi | \mathcal{D}^{\mathrm{tr}}, \theta) \| p(\phi | \theta))$

Approximating the posterior of test data: $p(\phi | \mathcal{D}, \theta) \Rightarrow p(\phi | \mathcal{D}^{\mathrm{te}}, \theta)$

We have: $\max_{\theta} ELBO = \max_{\theta} \mathbb{E}_{q(\phi | \mathcal{D}^{\mathrm{tr}}, \theta)}[\ln p(\underline{\mathcal{D}^{\mathrm{te}}} | \phi, \theta)] - KL(q(\phi | \mathcal{D}^{\mathrm{tr}}, \theta) \| p(\phi | \theta))$

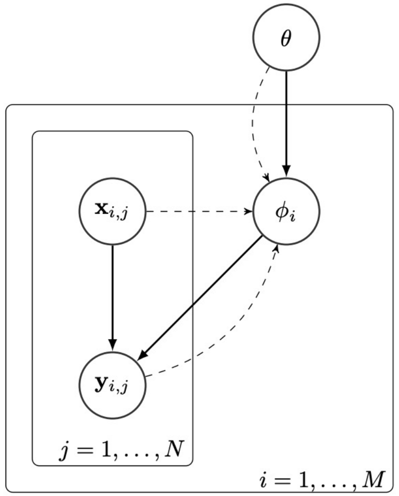
https://openreview.net/pdf?id=r[illegible]py3C5tX

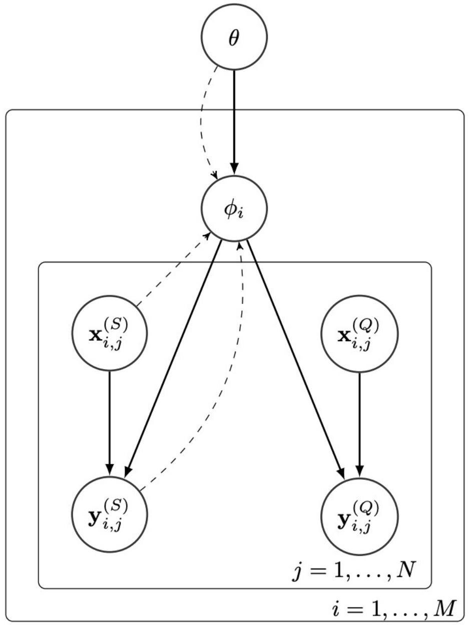
Amortized variational inference

- What is the loss function for the variational inference in meta-learning?

Variational inference: $\ln p(\mathcal{D}) \geqslant ELBO = \mathbb{E}_{q(\phi)}[\ln p(\mathcal{D} \mid \phi)] - KL(q(\phi) \| p(\phi))$

Because of the meta-learning, we have additional meta-parameter θ





By replacing: $q(\phi) \Rightarrow q(\phi | \mathcal{D}^{\text{tr}}, \theta)$ $\quad p(\phi | \mathcal{D}) \Rightarrow p(\phi | \mathcal{D}, \theta)$

We have: $\max_{\theta} ELBO = \max_{\theta} \; \mathbb{E}_{q(\phi | \mathcal{D}^{\text{tr}}, \theta)}[\ln p(\mathcal{D} | \phi, \theta)] - KL(q(\phi | \mathcal{D}^{\text{tr}}, \theta) \| p(\phi | \theta))$

Approximating the posterior of test data: $p(\phi | \mathcal{D}, \theta) \Rightarrow p(\phi | \mathcal{D}^{\text{te}}, \theta)$

We have: $\max_{\theta} ELBO = \max_{\theta} \; \mathbb{E}_{q(\phi | \mathcal{D}^{\text{tr}}, \theta)}[\ln p(\underline{\mathcal{D}^{\text{te}}} | \phi, \theta)] - KL(q(\phi | \mathcal{D}^{\text{tr}}, \theta) \| p(\phi | \theta))$

For all tasks, the final objective is:

$$\boxed{\max_{\theta} \; \underline{\mathbb{E}_{p(\mathcal{D}_i)}} \; \underbrace{\left[\mathbb{E}_{q(\phi_i | \mathcal{D}_i^{\text{tr}}, \theta)}[\ln p(\mathcal{D}_i^{\text{te}} | \phi_i, \theta)] - KL(q(\phi_i | \mathcal{D}_i^{\text{tr}}, \theta) \| p(\phi_i | \theta))\right]}_{ELBO \text{ for } \mathcal{D}_i}}$$

https://openreview.net/pdf?id=r[illegible]py3C5tX

Amortized variational inference

Different way to model $q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right) \sim N(\mu_i, \sigma_i^2)$

$$\boxed{\max_{\theta}\ \mathbb{E}_{p(\mathcal{D}_i)}\left[\mathbb{E}_{q(\phi_i|\mathcal{D}_i^{\mathrm{tr}},\theta)}[\ln p(\mathcal{D}_i^{\mathrm{te}}|\phi_i,\theta)] - KL(q(\phi_i|\mathcal{D}_i^{\mathrm{tr}},\theta)\|p(\phi_i|\theta))\right]}$$

Two parametric meta-learning approaches:

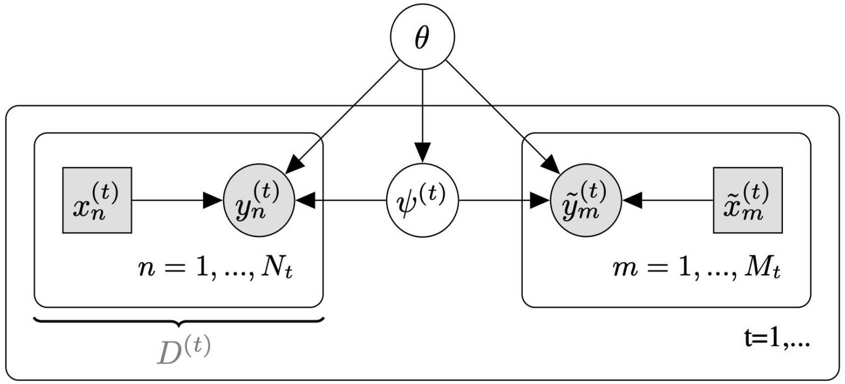
	Deterministic version	Bayesian version
- Black-box based. **Key idea:** Train a neural network to represent $q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right)$	$\phi_i = f_\theta(\mathcal{D}_i^{\mathrm{tr}})$	$\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\mathrm{tr}}, \theta)$
- Optimization based (normally using gradient descent). **Key idea:** Acquire ϕ_i through optimization on $\mathcal{D}_i^{\mathrm{tr}}$, meta parameter θ serves as a prior	$\phi_i = \arg\max_{\phi_i} \log p\left(\mathcal{D}_i^{\mathrm{tr}} \mid \phi_i\right) + \log p\left(\phi_i \mid \theta\right)$	$\lambda_i = \arg\max_{\lambda_i} \log p\left(\mathcal{D}_i^{\mathrm{tr}} \mid \phi_i\right) + \log p\left(\phi_i \mid \theta\right)$

Amortized variational inference

- Black-box based approach (VERSA)

$$\boxed{\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\text{tr}}, \theta)}$$

- Meta learner has two components:
  - Feature extraction network θ
  - Amortization network ϕ
- Meta parameter $\{\theta, \phi\}$
- Learner's parameter $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$



Training data $D^{(t)} = \{(x_n^{(t)}, y_n^{(t)})\}_{n=1}^{N_t}$, and test data $\{(\tilde{x}_m^{(t)}, \tilde{y}_m^{(t)})\}_{m=1}^{M_t}$

task specific parameters $\{\psi^{(t)}\}_{t=1}^{T}$ $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$

https://arxiv.org/pdf/1805.09921.pdf

https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-fd[illegible]09b2eff3

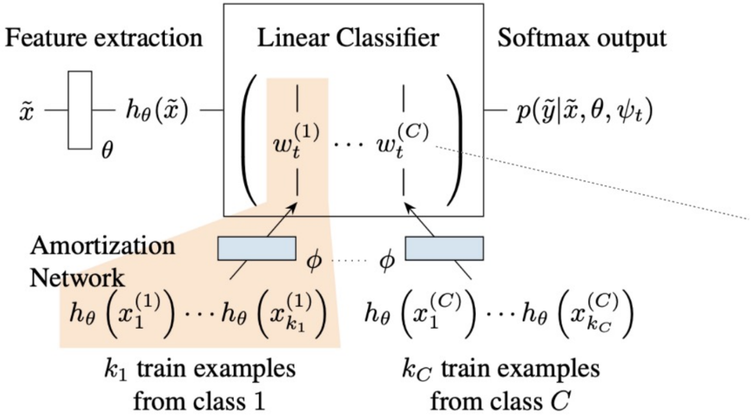
Amortized variational inference

- Black-box based approach (VERSA)

$$\boxed{\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\mathrm{tr}}, \theta)}$$

- Meta learner has two components:
  - Feature extraction network θ
  - Amortization network ϕ
- Meta parameter $\{\theta, \phi\}$
- Learner’s parameter $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$
- Approximate the posterior with

$$\psi_l^{(t)} \sim q_\phi(\psi | D^{(t)}, \theta)$$

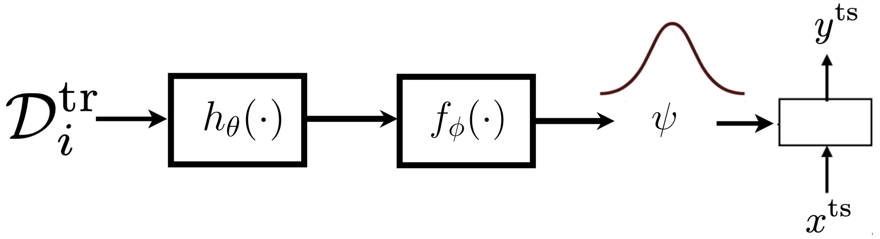


Feature extraction
Linear Classifier
Softmax output
$\tilde{x}$
θ
$h_\theta(\tilde{x})$
$w_t^{(1)} \cdots w_t^{(C)}$
$p(\tilde{y}|\tilde{x}, \theta, \psi_t)$
Amortization Network
ϕ
ϕ
$h_\theta\left(x_1^{(1)}\right) \cdots h_\theta\left(x_{k_1}^{(1)}\right)$
$h_\theta\left(x_1^{(C)}\right) \cdots h_\theta\left(x_{k_C}^{(C)}\right)$
k_1 train examples from class 1
k_C train examples from class C

$$\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$$

$$\psi_l^{(t)} \sim q_\phi(\psi | D^{(t)}, \theta)$$

Collect L samples of ψ



https://arxiv.org/pdf/1805.09921.pdf

https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

- Black-box based approach (VERSA)

$$\boxed{\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\mathrm{tr}}, \theta)}$$

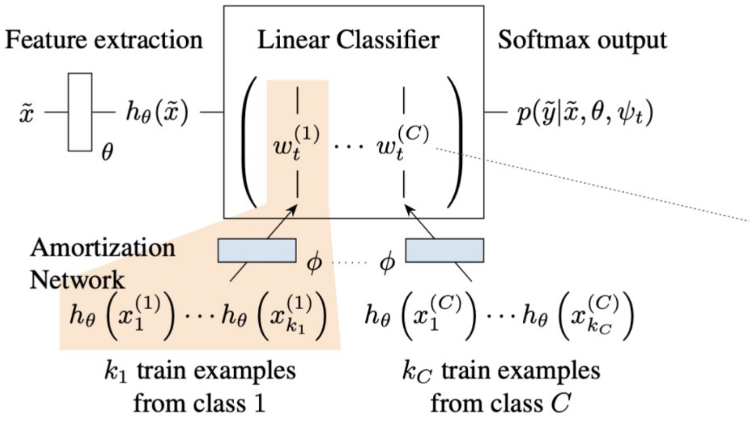
- Meta learner has two components:
  - Feature extraction network θ
  - Amortization network ϕ
- Meta parameter $\{\theta, \phi\}$
- Learner's parameter $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$
- Approximate the posterior with

$$\psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

- Objective function

$$\hat{\mathcal{L}}(\theta, \phi) = \frac{1}{MT}\sum_{M,T}\log\frac{1}{L}\sum_{l=1}^{L} p\left(\tilde{y}_m^{(t)}|\tilde{x}_m^{(t)}, \psi_l^{(t)}, \theta\right), \quad \text{with } \psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

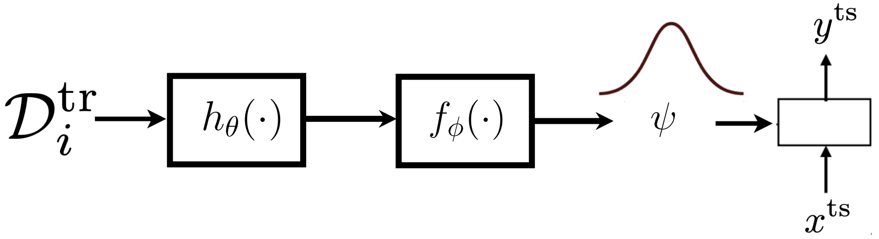
$$\boxed{\max_\theta \underline{\mathbb{E}_{p(\mathcal{D}_i)}[\mathbb{E}_{q(\phi_i|\mathcal{D}_i^{\mathrm{tr}},\theta)}[\ln p(\mathcal{D}_i^{\mathrm{te}}|\phi_i, \theta)]} - KL(q(\phi_i|\mathcal{D}_i^{\mathrm{tr}}, \theta)\|p(\phi_i|\theta))]}$$


Feature extraction
Linear Classifier
Softmax output
$\tilde{x}$
θ
$h_\theta(\tilde{x})$
$w_t^{(1)} \cdots w_t^{(C)}$
$p(\tilde{y}|\tilde{x}, \theta, \psi_t)$
Amortization Network
ϕ
ϕ
$h_\theta\left(x_1^{(1)}\right) \cdots h_\theta\left(x_{k_1}^{(1)}\right)$
$h_\theta\left(x_1^{(C)}\right) \cdots h_\theta\left(x_{k_C}^{(C)}\right)$
k_1 train examples from class 1
k_C train examples from class C

$$\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$$

$$\psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

Collect *L* samples of ψ



$$\delta(\phi_i^*) \qquad p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta^*\right)$$

$$\delta(\theta^*) \qquad p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$$

https://arxiv.org/pdf/1805[illegible]9921.pdf

https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

- Black-box based approach (VERSA)

$$\boxed{\lambda = \{\mu_i, \sigma_i^2\} = f(\mathcal{D}_i^{\text{tr}}, \theta)}$$

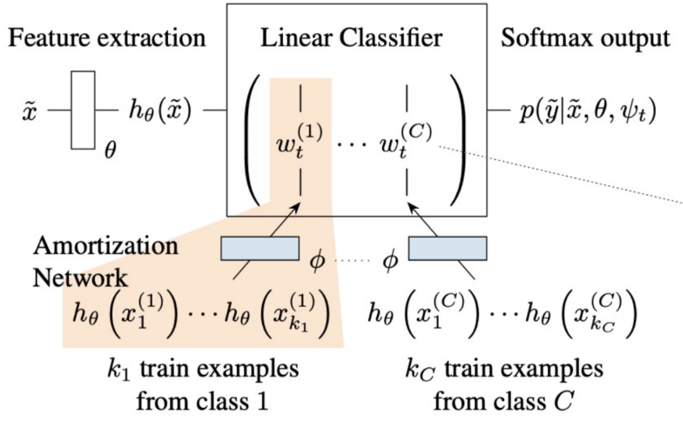
- Meta learner has two components:
  - Feature extraction network θ
  - Amortization network ϕ
- Meta parameter $\{\theta, \phi\}$
- Learner's parameter $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$
- Approximate the posterior with

$$\psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

- Objective function

$$\hat{\mathcal{L}}(\theta, \phi) = \frac{1}{MT} \sum_{M,T} \log \frac{1}{L} \sum_{l=1}^{L} p\left(\tilde{y}_m^{(t)} | \tilde{x}_m^{(t)}, \psi_l^{(t)}, \theta\right), \quad \text{with } \psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

$$\boxed{\max_\theta \underline{\mathbb{E}_{p(\mathcal{D}_i)}\left[\mathbb{E}_{q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)}[\ln p(\mathcal{D}_i^{\text{te}}|\phi_i, \theta)]\right.} - KL(q(\phi_i|\mathcal{D}_i^{\text{tr}}, \theta) \| p(\phi_i|\theta))\Big]}$$

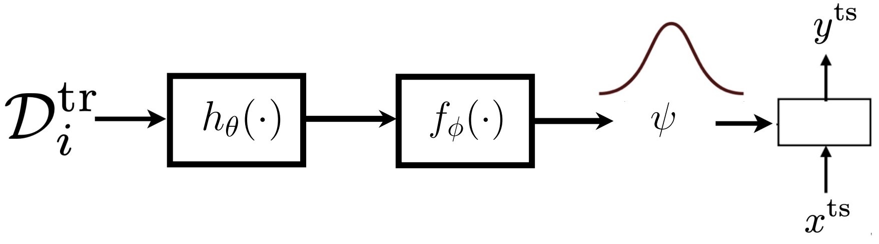


Feature extraction
Linear Classifier
Softmax output
$\tilde{x}$
$h_\theta(\tilde{x})$
θ
$w_t^{(1)} \cdots w_t^{(C)}$
$p(\tilde{y}|\tilde{x}, \theta, \psi_t)$
Amortization Network
ϕ
ϕ
$h_\theta\left(x_1^{(1)}\right) \cdots h_\theta\left(x_{k_1}^{(1)}\right)$
$h_\theta\left(x_1^{(C)}\right) \cdots h_\theta\left(x_{k_C}^{(C)}\right)$
k_1 train examples from class 1
k_C train examples from class C

$$\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$$

$$\psi_l^{(t)} \sim q_\phi(\psi|D^{(t)}, \theta)$$

Collect *L* samples of ψ



$\delta(\phi_i^*)$ ✗ $\quad p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$ ✓

$\delta(\theta^*)$ ✓ $\quad p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$ ✗

https://arxiv.org/pdf/1805.09921.pdf

https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

- Optimization based approach (Amortized Bayesian Meta-Learning)

- Recall objective function:

$$\boxed{\max_{\theta}\ \mathbb{E}_{p(\mathcal{D}_i)}\left[\mathbb{E}_{\underline{q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)}}[\ln p(\mathcal{D}_i^{\text{te}}|\phi_i,\theta)] - KL(\underline{q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)}\|p(\phi_i|\theta))\right]}$$

- $q\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta\right)$ is achieved by SGD on the mean and variance using $\mathcal{D}_i^{\text{tr}}$

$$\boxed{\lambda_i = \arg\max_{\lambda_i} \log p\left(\mathcal{D}_i^{\text{tr}} \mid \phi_i\right) + \log p\left(\phi_i \mid \theta\right)}$$

$$q_\theta\left(\phi_i \middle| \mathcal{D}_i^{(S)}\right) = \mathcal{N}\left(\phi_i; \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(\boldsymbol{K})}, \boldsymbol{\sigma}^{\boldsymbol{2}(\boldsymbol{K})}_{\boldsymbol{\lambda}}\right)$$

- SGD:

```
1. $\lambda^{(0)} = \lambda^{(init)}$
2. for $k = 0, \ldots, K-1$, set
   $\lambda^{(k+1)} = \lambda^{(k)} - \alpha \nabla_{\lambda^{(k)}} \underline{\mathcal{L}_{\mathcal{D}}(\lambda^{(k)}, \theta)}$
```

- $\lambda^{(init)}$ is θ like MAML, SGD is the inner loop optimization

https://openreview.net/pdf?id=rkgpy3C5tX

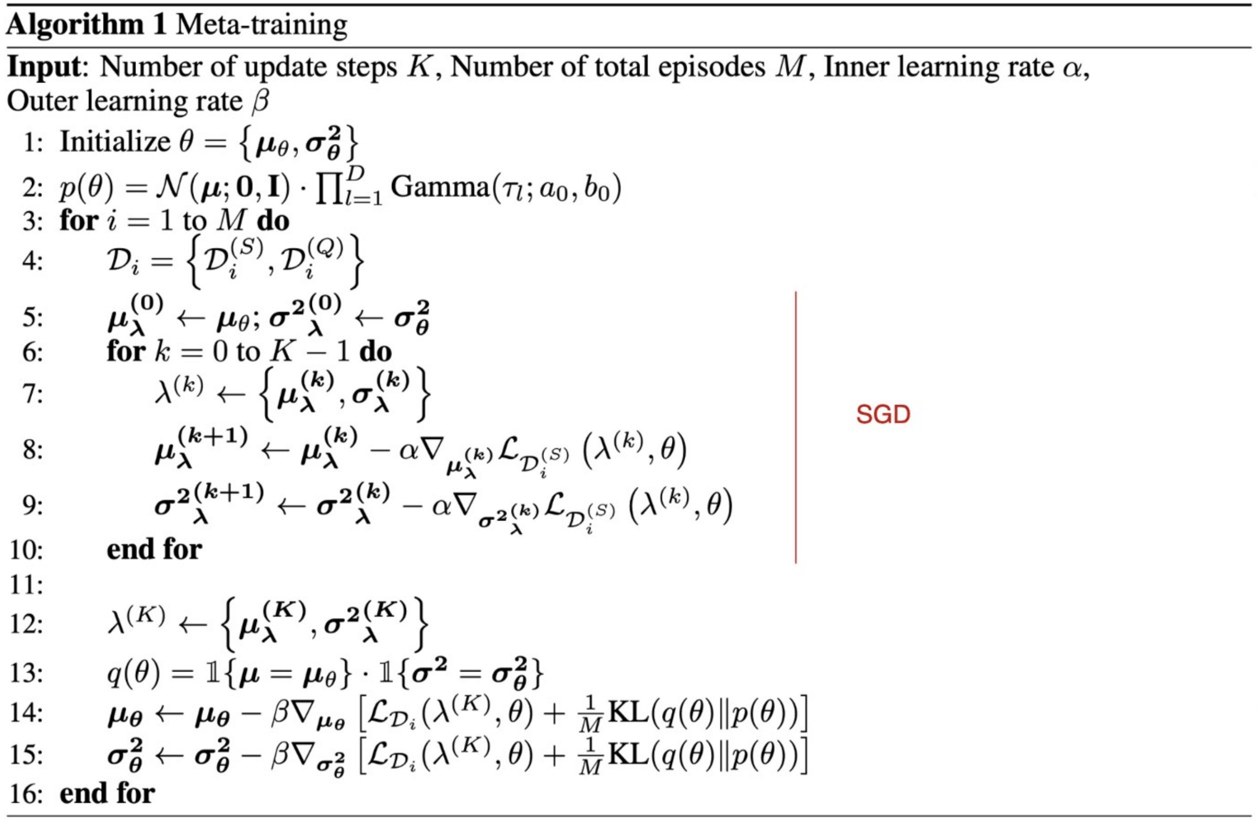
https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

- Optimization based approach (Amortized Bayesian Meta-Learning)

$$\delta(\phi_i^*) \qquad p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$$

$$\delta(\theta^*) \qquad p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$$

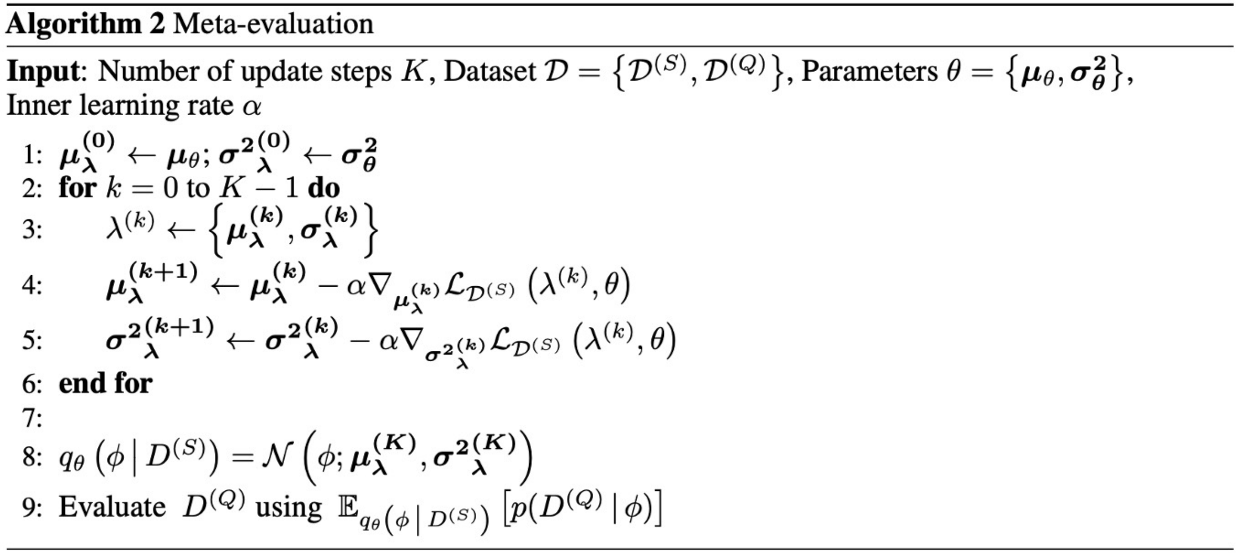


Algorithm 1 Meta-training

```
Input: Number of update steps $K$, Number of total episodes $M$, Inner learning rate $\alpha$, Outer learning rate $\beta$
Initialize $\theta = \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2\}$
$p(\theta) = \mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, \mathbf{I}) \cdot \prod_{l=1}^{D} \text{Gamma}(\tau_l; a_0, b_0)$
for $i = 1$ to $M$ do
    $\mathcal{D}_i = \left\{\mathcal{D}_i^{(S)}, \mathcal{D}_i^{(Q)}\right\}$
    $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\mu}_\theta; \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\sigma}_\theta^2$
    for $k = 0$ to $K-1$ do
        $\lambda^{(k)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}, \boldsymbol{\sigma}_{\boldsymbol{\lambda}}^{(k)}\right\}$
        $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}_i^{(S)}}\left(\lambda^{(k)}, \theta\right)$
        $\boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}_i^{(S)}}\left(\lambda^{(k)}, \theta\right)$
    end for

    $\lambda^{(K)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(K)}, \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(K)}\right\}$
    $q(\theta) = \mathbb{1}\{\boldsymbol{\mu} = \boldsymbol{\mu}_\theta\} \cdot \mathbb{1}\{\boldsymbol{\sigma}^2 = \boldsymbol{\sigma}_\theta^2\}$
    $\boldsymbol{\mu}_\theta \leftarrow \boldsymbol{\mu}_\theta - \beta \nabla_{\boldsymbol{\mu}_\theta}\left[\mathcal{L}_{\mathcal{D}_i}(\lambda^{(K)}, \theta) + \frac{1}{M}\text{KL}(q(\theta) \| p(\theta))\right]$
    $\boldsymbol{\sigma}_\theta^2 \leftarrow \boldsymbol{\sigma}_\theta^2 - \beta \nabla_{\boldsymbol{\sigma}_\theta^2}\left[\mathcal{L}_{\mathcal{D}_i}(\lambda^{(K)}, \theta) + \frac{1}{M}\text{KL}(q(\theta) \| p(\theta))\right]$
end for
```

SGD



Algorithm 2 Meta-evaluation

```
Input: Number of update steps $K$, Dataset $\mathcal{D} = \left\{\mathcal{D}^{(S)}, \mathcal{D}^{(Q)}\right\}$, Parameters $\theta = \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2\}$, Inner learning rate $\alpha$
$\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\mu}_\theta; \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\sigma}_\theta^2$
for $k = 0$ to $K-1$ do
    $\lambda^{(k)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}, \boldsymbol{\sigma}_{\boldsymbol{\lambda}}^{(k)}\right\}$
    $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}^{(S)}}\left(\lambda^{(k)}, \theta\right)$
    $\boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}^{(S)}}\left(\lambda^{(k)}, \theta\right)$
end for

$q_\theta\left(\phi \,|\, D^{(S)}\right) = \mathcal{N}\left(\phi; \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(K)}, \boldsymbol{\sigma}^2{}_{\boldsymbol{\lambda}}^{(K)}\right)$
Evaluate $D^{(Q)}$ using $\mathbb{E}_{q_\theta(\phi \,|\, D^{(S)})}\left[p(D^{(Q)} \,|\, \phi)\right]$
```

https://openreview.net/pdf?id=rkgpy3C5tX

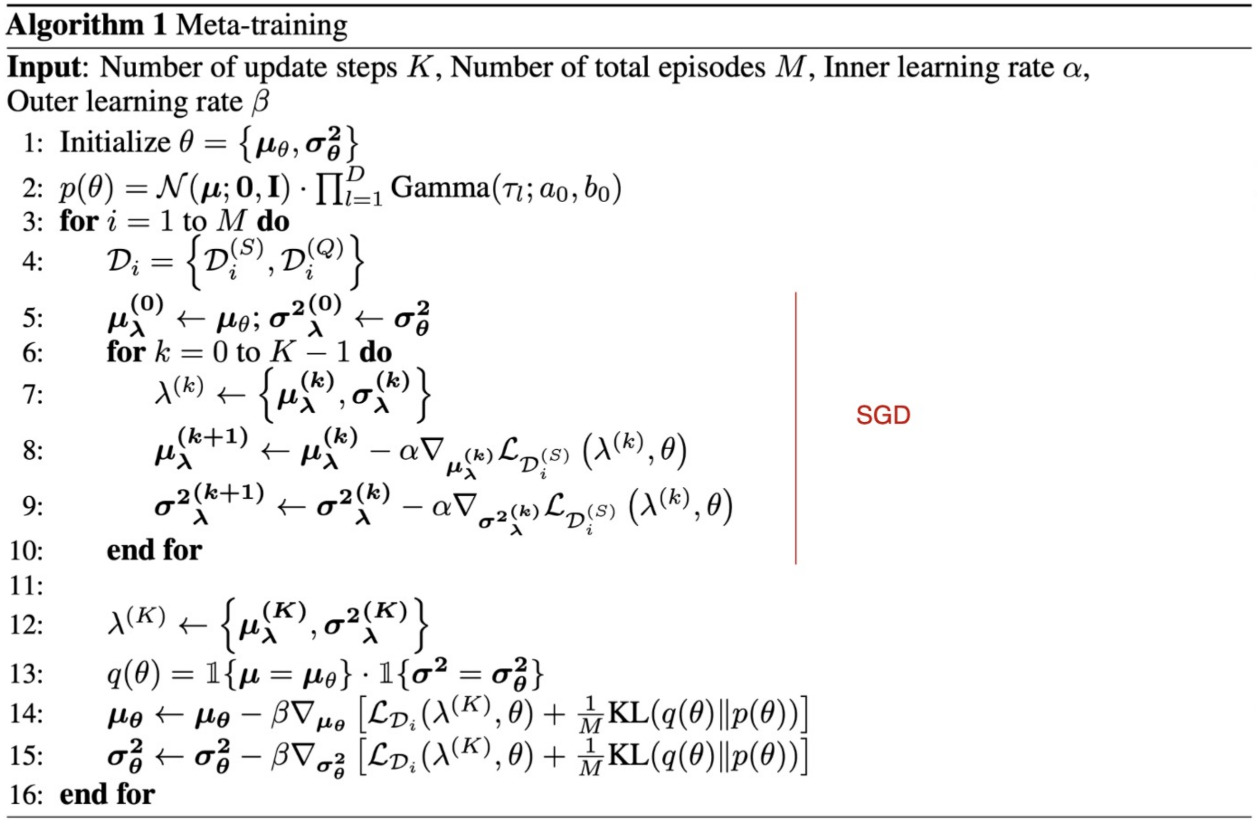
https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-r 09b2eff3

Amortized variational inference

- Optimization based approach (Amortized Bayesian Meta-Learning)

$\delta(\phi_i^*)$ ✗ $p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$ ✓

$\delta(\theta^*)$ ✗ $p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$ ✓



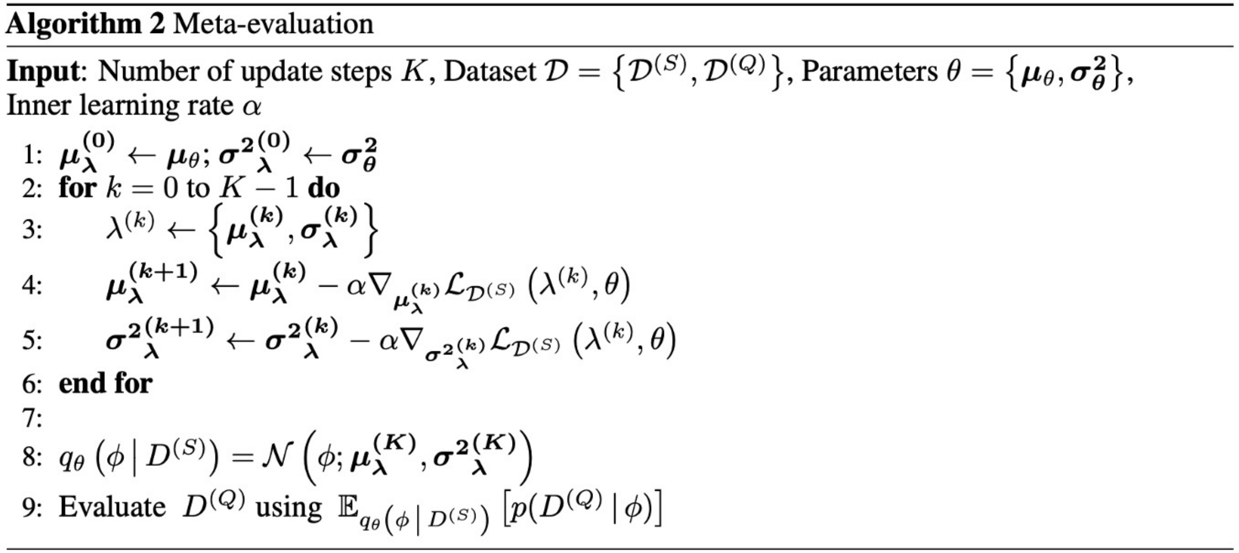
Algorithm 1 Meta-training

Input: Number of update steps K, Number of total episodes M, Inner learning rate α, Outer learning rate β

```
Initialize $\theta = \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2\}$
$p(\theta) = \mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, \mathbf{I}) \cdot \prod_{l=1}^{D} \text{Gamma}(\tau_l; a_0, b_0)$
for $i = 1$ to $M$ do
    $\mathcal{D}_i = \left\{\mathcal{D}_i^{(S)}, \mathcal{D}_i^{(Q)}\right\}$
    $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\mu}_\theta;\ \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\sigma}_\theta^2$
    for $k = 0$ to $K - 1$ do
        $\lambda^{(k)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}, \boldsymbol{\sigma}_{\boldsymbol{\lambda}}^{(k)}\right\}$
        $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}_i^{(S)}}\left(\lambda^{(k)}, \theta\right)$
        $\boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}_i^{(S)}}\left(\lambda^{(k)}, \theta\right)$
    end for

    $\lambda^{(K)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(K)}, \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(K)}\right\}$
    $q(\theta) = \mathbb{1}\{\boldsymbol{\mu} = \boldsymbol{\mu}_\theta\} \cdot \mathbb{1}\{\boldsymbol{\sigma}^2 = \boldsymbol{\sigma}_\theta^2\}$
    $\boldsymbol{\mu}_\theta \leftarrow \boldsymbol{\mu}_\theta - \beta \nabla_{\boldsymbol{\mu}_\theta}\left[\mathcal{L}_{\mathcal{D}_i}(\lambda^{(K)}, \theta) + \frac{1}{M}\text{KL}(q(\theta) \| p(\theta))\right]$
    $\boldsymbol{\sigma}_\theta^2 \leftarrow \boldsymbol{\sigma}_\theta^2 - \beta \nabla_{\boldsymbol{\sigma}_\theta^2}\left[\mathcal{L}_{\mathcal{D}_i}(\lambda^{(K)}, \theta) + \frac{1}{M}\text{KL}(q(\theta) \| p(\theta))\right]$
end for
```

SGD



Algorithm 2 Meta-evaluation

Input: Number of update steps K, Dataset $\mathcal{D} = \left\{\mathcal{D}^{(S)}, \mathcal{D}^{(Q)}\right\}$, Parameters $\theta = \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2\}$, Inner learning rate α

```
$\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\mu}_\theta;\ \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(0)} \leftarrow \boldsymbol{\sigma}_\theta^2$
for $k = 0$ to $K - 1$ do
    $\lambda^{(k)} \leftarrow \left\{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}, \boldsymbol{\sigma}_{\boldsymbol{\lambda}}^{(k)}\right\}$
    $\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}^{(S)}}\left(\lambda^{(k)}, \theta\right)$
    $\boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k+1)} \leftarrow \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k)} - \alpha \nabla_{\boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(k)}} \mathcal{L}_{\mathcal{D}^{(S)}}\left(\lambda^{(k)}, \theta\right)$
end for

$q_\theta\left(\phi \mid D^{(S)}\right) = \mathcal{N}\left(\phi; \boldsymbol{\mu}_{\boldsymbol{\lambda}}^{(K)}, \boldsymbol{\sigma}^{2}{}_{\boldsymbol{\lambda}}^{(K)}\right)$
Evaluate $D^{(Q)}$ using $\mathbb{E}_{q_\theta(\phi \mid D^{(S)})}\left[p(D^{(Q)} \mid \phi)\right]$
```

https://openreview.net/pdf?id=r[illegible]gpy3C5tX

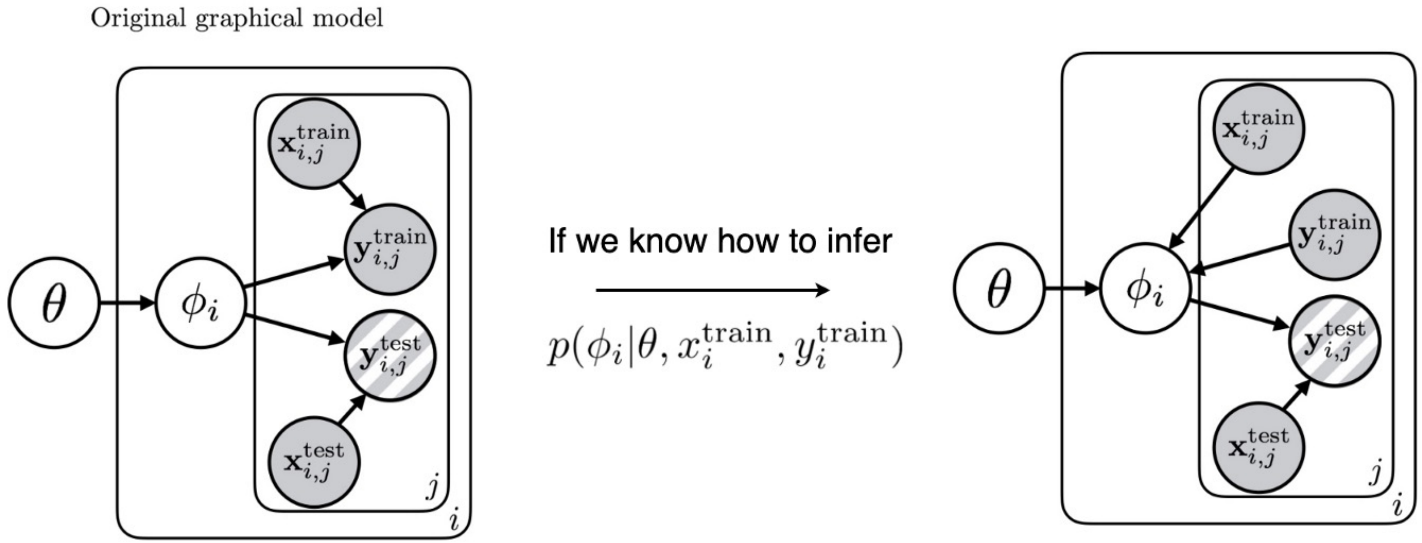
https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

- Optimization based approach (Probabilistic MAML)

How about approximating posterior distribution directly on θ ?

$\delta(\phi_i^*)$	✓	$p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$	✗
$\delta(\theta^*)$	✗	$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$	✓



Original graphical model
θ
ϕ_i
$\mathbf{x}_{i,j}^{\text{train}}$
$\mathbf{y}_{i,j}^{\text{train}}$
$\mathbf{y}_{i,j}^{\text{test}}$
$\mathbf{x}_{i,j}^{\text{test}}$
j
i

If we know how to infer

$$\xrightarrow{\quad\quad}$$

$$p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}})$$

For example, in MAML,

$$p(\phi_i|\mathbf{x}_i^{\text{tr}}, \mathbf{y}_i^{\text{tr}}, \theta) \approx \delta(\phi_i = \phi_i^\star) \text{ where } \phi_i^\star \text{ is obtained via gradient descent starting from } \theta.$$

https://arxiv.org/pdf/1806.02817.pdf

https://jonathan-hui.medium.com/meta-learning-bayesian-meta-learning-weak-supervision-[illegible]09b2eff3

Amortized variational inference

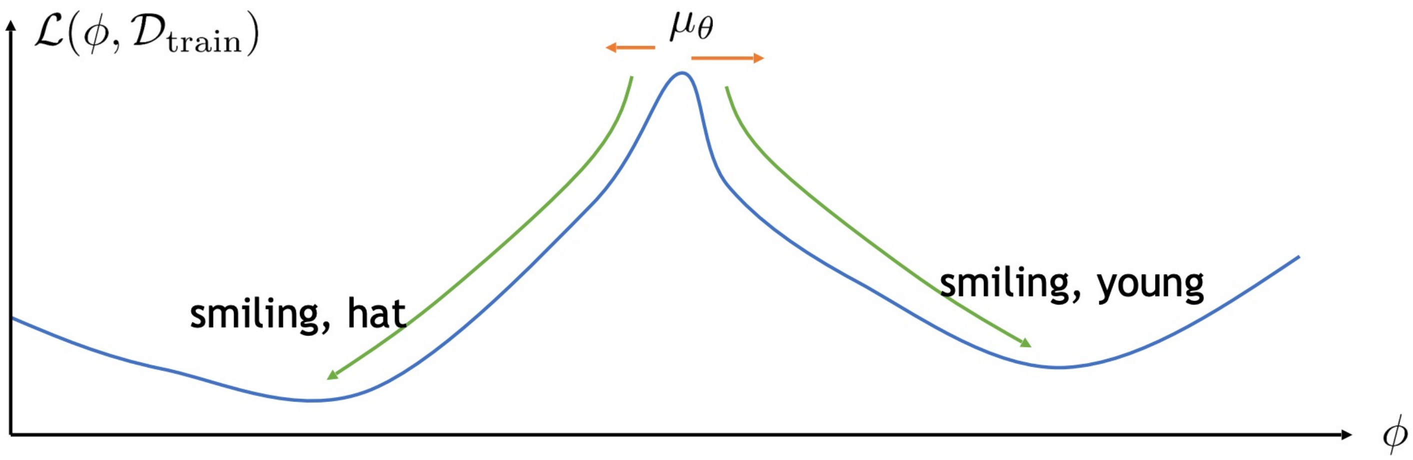
- Optimization based approach (Probabilistic MAML)

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

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

What does ancestral sampling look like?

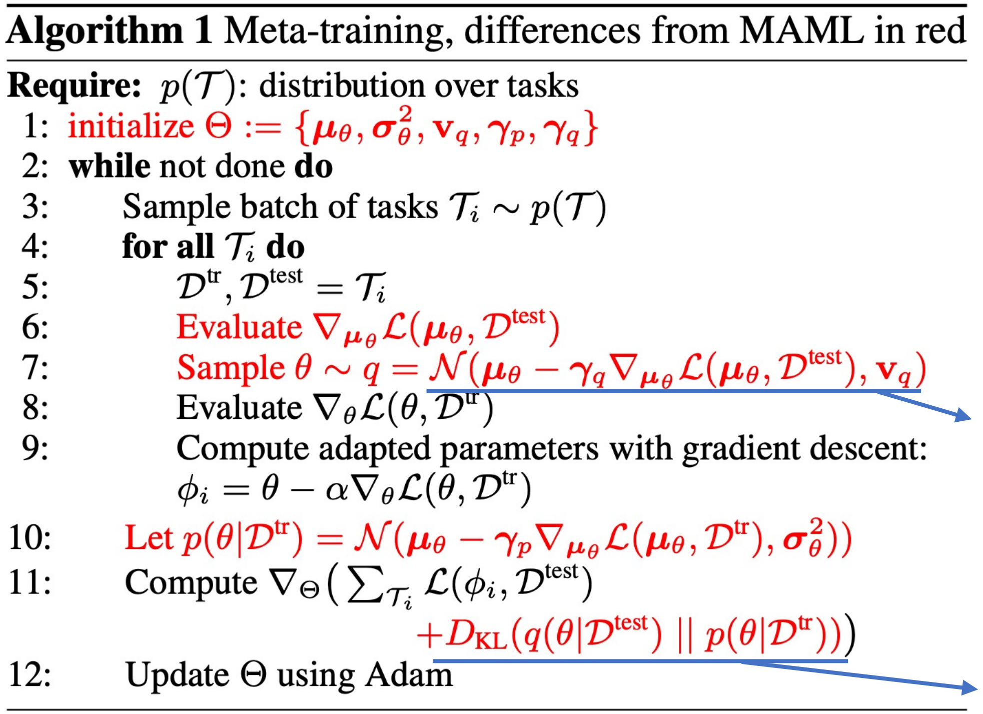
1. $\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$



https://arxiv.org/pdf/18[illegible]2817.pdf

Amortized variational inference

- Optimization based approach (Probabilistic MAML)

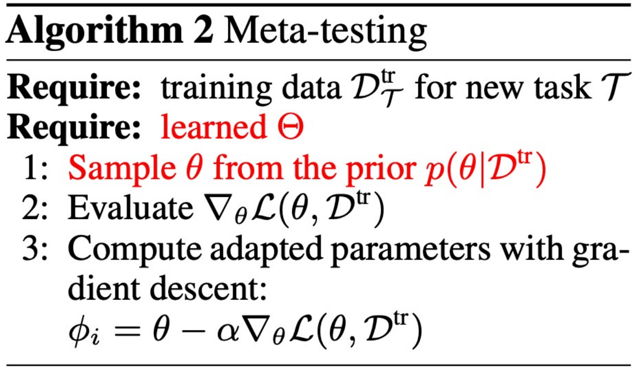


Algorithm 1 Meta-training, differences from MAML in red

```
Require: $p(\mathcal{T})$: distribution over tasks
initialize $\Theta := \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2, \mathbf{v}_q, \gamma_p, \gamma_q\}$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ do
        $\mathcal{D}^{\text{tr}}, \mathcal{D}^{\text{test}} = \mathcal{T}_i$
        Evaluate $\nabla_{\boldsymbol{\mu}_\theta} \mathcal{L}(\boldsymbol{\mu}_\theta, \mathcal{D}^{\text{test}})$
        Sample $\theta \sim q = \mathcal{N}(\boldsymbol{\mu}_\theta - \gamma_q \nabla_{\boldsymbol{\mu}_\theta} \mathcal{L}(\boldsymbol{\mu}_\theta, \mathcal{D}^{\text{test}}), \mathbf{v}_q)$
        Evaluate $\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
        Compute adapted parameters with gradient descent:
        $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
    Let $p(\theta|\mathcal{D}^{\text{tr}}) = \mathcal{N}(\boldsymbol{\mu}_\theta - \gamma_p \nabla_{\boldsymbol{\mu}_\theta} \mathcal{L}(\boldsymbol{\mu}_\theta, \mathcal{D}^{\text{tr}}), \boldsymbol{\sigma}_\theta^2))$
    Compute $\nabla_\Theta \big( \sum_{\mathcal{T}_i} \mathcal{L}(\phi_i, \mathcal{D}^{\text{test}})$
            $+ D_{\text{KL}}(q(\theta|\mathcal{D}^{\text{test}}) \,||\, p(\theta|\mathcal{D}^{\text{tr}})) \big)$
    Update $\Theta$ using Adam
```

Posterior distribution on test data $q(\theta|\mathcal{D}^{\text{test}})$

Minimize the gap between two posterior distribution



Algorithm 2 Meta-testing

```
Require: training data $\mathcal{D}_{\mathcal{T}}^{\text{tr}}$ for new task $\mathcal{T}$
Require: learned $\Theta$
Sample $\theta$ from the prior $p(\theta|\mathcal{D}^{\text{tr}})$
Evaluate $\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
Compute adapted parameters with gradient descent:
$\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
```

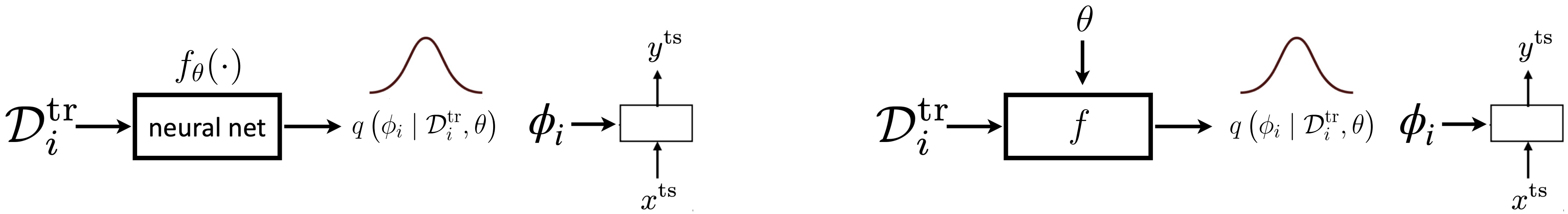
https://arxiv.org/pdf/18[illegible]2817.pdf

Amortized variational inference

- **Summary**

- **Latent variable models + variational inference (approximating $p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$)**

  approximate likelihood of latent variable model with variational lower bound


$f_\theta(\cdot)$
$\mathcal{D}_i^{\text{tr}}$
neural net
$q\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta\right)$
ϕ_i
y^{ts}
x^{ts}
θ
$\mathcal{D}_i^{\text{tr}}$
f
$q\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta\right)$
ϕ_i
y^{ts}
x^{ts}

$$\boxed{\max_\theta \ \mathbb{E}_{p(\mathcal{D}_i)}\left[\mathbb{E}_{\underline{q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)}}[\ln p(\mathcal{D}_i^{\text{te}}|\phi_i,\theta)] - KL(\underline{q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)}\|p(\phi_i|\theta))\right]}$$

Pros:

+ can represent non-Gaussian distributions over y^{ts}
+ produces distribution over functions

Cons:

- Can only represent Gaussian distributions $p(\phi_i|\theta)$

Not always restricting: e.g. if $p(y_i^{\text{ts}}|x_i^{\text{ts}}, \phi_i, \theta)$ is also conditioned on θ.

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learni…2020.pdf

Amortized variational inference

- **Summary**

- **Latent variable models + variational inference (approximating $p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$)**

approximate likelihood of latent variable model with variational lower bound

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

$$
\min_{\lambda_\theta} \left(\sum_{\mathcal{T}_i} \mathcal{L}\left(\phi_i, \mathcal{D}^{\text{te}}\right) + KL\left(q\left(\theta \mid \mathcal{D}^{\text{te}}\right) \| p\left(\theta \mid \mathcal{D}^{\text{tr}}\right)\right) \right)
$$

Pros: Non-Gaussian posterior, simple at test time, only one model instance.

Con: More complex training procedure.

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Outline

- Introduction
    - Why Bayesian meta-learning?
    - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
    - Amortized variational inference
        - Black-box
        - Optimization
    - Bayesian ensembles
    - Bayesian neural networks
- Bayesian meta-learning evaluation
    - Qualitative visualization
    - Quantitative evaluation
    - Active-learning evaluation

Bayesian ensembles

- Key idea: train separate models on bootstraps of the data

- Ensemble of MAML (EMAML)

  Train M independent MAML models then take the average

  Does not work as the ensemble members are too similar

- Bayesian Meta-Learning with Chaser Loss (BMAML)

  Use stein variational gradient descent (SVGD) to push the particles away from one another

  Use chaser loss to improve the generalization ability

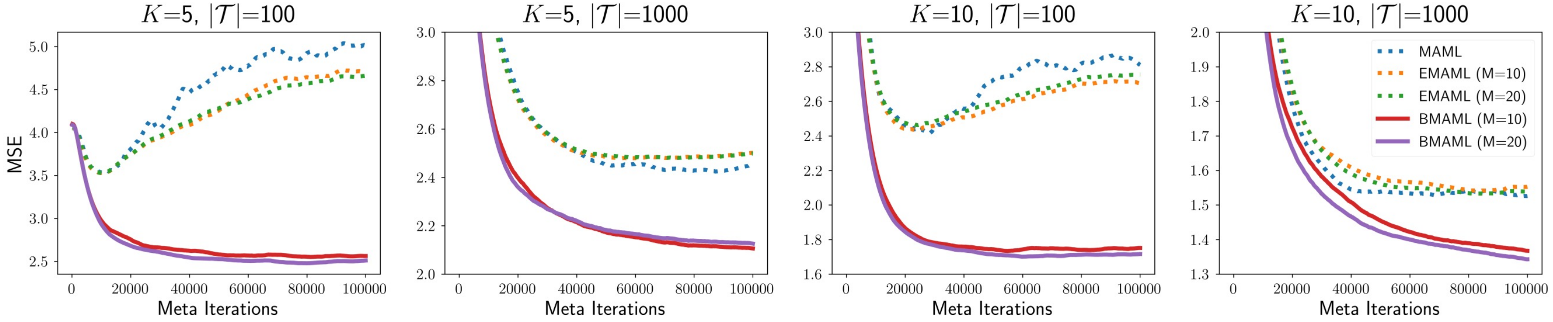


Figure 1: Sinusoidal regression experimental results (meta-testing performance) by varying the number of examples (K-shot) given for each task and the number of tasks $|\mathcal{T}|$ used for meta-training.

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

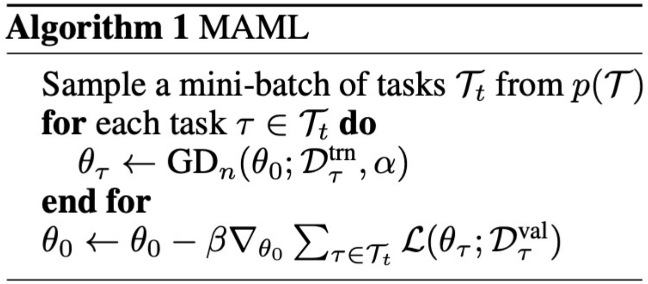
https://proceedings.neurips.cc/paper/2018/file/e1021d43911ca2c1845910d84f40ae...Paper.pdf

Bayesian ensembles

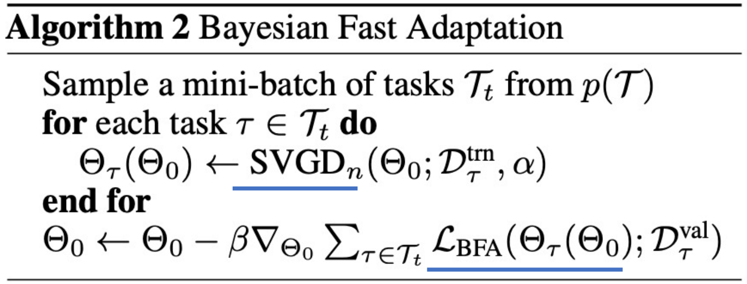
- Notations
  - Meta parameter $\theta \Rightarrow \theta_0$
  - Learner's parameter $\phi \Rightarrow \theta_\tau$
- Stein variational gradient descent (SVGD)

$$\theta_{t+1} \leftarrow \theta_t + \epsilon_t \phi(\theta_t) \;\text{ where }\; \phi(\theta_t) = \frac{1}{M} \underbrace{\sum_{j=1}^{M}}_{\text{keep M models}} \left[\underbrace{k(\theta_t^j, \theta_t)}_{\text{positive-definite kernel}} \nabla_{\theta_t^j} \log p(\theta_t^j) + \underbrace{\nabla_{\theta_t^j} k(\theta_t^j, \theta_t)}_{\text{push the models away from one another}} \right],$$

- MAML vs. MAML with SVGD



```
Algorithm 1 MAML
  Sample a mini-batch of tasks $\mathcal{T}_t$ from $p(\mathcal{T})$
  for each task $\tau \in \mathcal{T}_t$ do
    $\theta_\tau \leftarrow \mathrm{GD}_n(\theta_0; \mathcal{D}_\tau^{\mathrm{trn}}, \alpha)$
  end for
  $\theta_0 \leftarrow \theta_0 - \beta \nabla_{\theta_0} \sum_{\tau \in \mathcal{T}_t} \mathcal{L}(\theta_\tau; \mathcal{D}_\tau^{\mathrm{val}})$
```



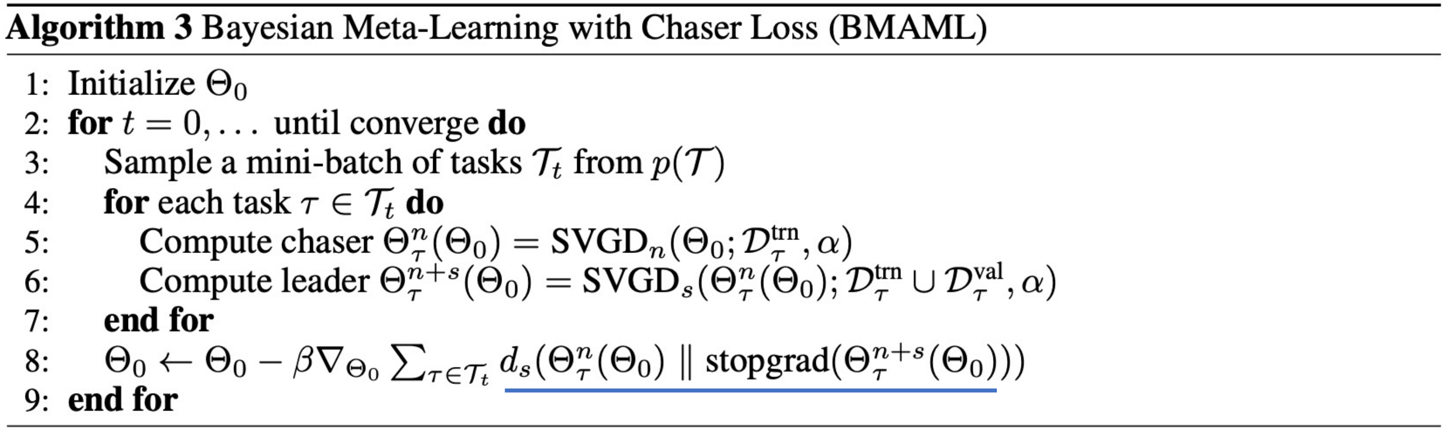
```
Algorithm 2 Bayesian Fast Adaptation
  Sample a mini-batch of tasks $\mathcal{T}_t$ from $p(\mathcal{T})$
  for each task $\tau \in \mathcal{T}_t$ do
    $\Theta_\tau(\Theta_0) \leftarrow \underline{\mathrm{SVGD}_n}(\Theta_0; \mathcal{D}_\tau^{\mathrm{trn}}, \alpha)$
  end for
  $\Theta_0 \leftarrow \Theta_0 - \beta \nabla_{\Theta_0} \sum_{\tau \in \mathcal{T}_t} \underline{\mathcal{L}_{\mathrm{BFA}}(\Theta_\tau(\Theta_0); \mathcal{D}_\tau^{\mathrm{val}})}$
```

$$\text{where } \mathcal{L}_{\mathrm{BFA}}(\Theta_\tau(\Theta_0); \mathcal{D}_\tau^{\mathrm{val}}) = \log \left[\frac{1}{M} \sum_{m=1}^{M} p(\mathcal{D}_\tau^{\mathrm{val}} | \theta_\tau^m) \right]$$

https://proceedings.neurips.cc/paper/2018/file/e1021d43911ca2c1845910d84f40ae[illegible]Paper.pdf

Bayesian ensembles

- Bayesian MAML with Chaser loss



Algorithm 3 Bayesian Meta-Learning with Chaser Loss (BMAML)

```
Initialize $\Theta_0$
for $t = 0, \dots$ until converge do
    Sample a mini-batch of tasks $\mathcal{T}_t$ from $p(\mathcal{T})$
    for each task $\tau \in \mathcal{T}_t$ do
        Compute chaser $\Theta_\tau^n(\Theta_0) = \mathrm{SVGD}_n(\Theta_0; \mathcal{D}_\tau^{\mathrm{trn}}, \alpha)$
        Compute leader $\Theta_\tau^{n+s}(\Theta_0) = \mathrm{SVGD}_s(\Theta_\tau^n(\Theta_0); \mathcal{D}_\tau^{\mathrm{trn}} \cup \mathcal{D}_\tau^{\mathrm{val}}, \alpha)$
    end for
    $\Theta_0 \leftarrow \Theta_0 - \beta \nabla_{\Theta_0} \sum_{\tau \in \mathcal{T}_t} \underline{d_s(\Theta_\tau^n(\Theta_0) \parallel \mathrm{stopgrad}(\Theta_\tau^{n+s}(\Theta_0))}))$
end for
```

- Chaser loss

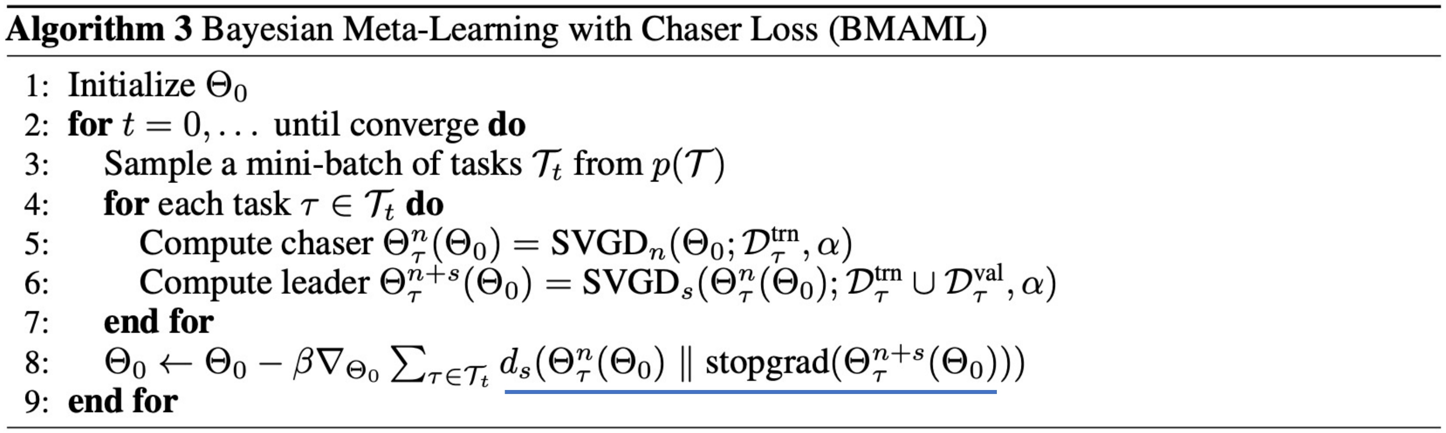
$$\mathcal{L}_{\mathrm{BMAML}}(\Theta_0) = \sum_{\tau \in \mathcal{T}_t} d_s(\Theta_\tau^n \parallel \Theta_\tau^{n+s}) = \sum_{\tau \in \mathcal{T}_t} \sum_{m=1}^{M} \|\theta_\tau^{n,m} - \theta_\tau^{n+s,m}\|_2^2$$

$\delta(\phi_i^*)$	$p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta^*\right)$
$\delta(\theta^*)$	$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning 2020.pdf
https://proceedings.neurips.cc/paper/2018/file/e1021d43911ca2c1845910d84f40ae Paper.pdf

Bayesian ensembles

- Bayesian MAML with Chaser loss



Algorithm 3 Bayesian Meta-Learning with Chaser Loss (BMAML)

```
Initialize $\Theta_0$
for $t = 0, \ldots$ until converge do
    Sample a mini-batch of tasks $\mathcal{T}_t$ from $p(\mathcal{T})$
    for each task $\tau \in \mathcal{T}_t$ do
        Compute chaser $\Theta^n_\tau(\Theta_0) = \text{SVGD}_n(\Theta_0; \mathcal{D}^{\text{trn}}_\tau, \alpha)$
        Compute leader $\Theta^{n+s}_\tau(\Theta_0) = \text{SVGD}_s(\Theta^n_\tau(\Theta_0); \mathcal{D}^{\text{trn}}_\tau \cup \mathcal{D}^{\text{val}}_\tau, \alpha)$
    end for
    $\Theta_0 \leftarrow \Theta_0 - \beta \nabla_{\Theta_0} \sum_{\tau \in \mathcal{T}_t} \underline{d_s(\Theta^n_\tau(\Theta_0) \parallel \text{stopgrad}(\Theta^{n+s}_\tau(\Theta_0}))))$
end for
```

- Chaser loss

$$\mathcal{L}_{\text{BMAML}}(\Theta_0) = \sum_{\tau \in \mathcal{T}_t} d_s(\Theta^n_\tau \parallel \Theta^{n+s}_\tau) = \sum_{\tau \in \mathcal{T}_t} \sum_{m=1}^{M} \|\theta^{n,m}_\tau - \theta^{n+s,m}_\tau\|^2_2$$

Pros: Simple, tends to work well, non-Gaussian distributions.

Con: Need to maintain M model instances. (or do gradient-based inference on **last layer only**)

$\delta(\phi_i^*)$	✗	$p\left(\phi_i \mid \mathcal{D}_i^{\text{tr}}, \theta^*\right)$	✓
$\delta(\theta^*)$	✗	$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$	✓

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf
https://proceedings.neurips.cc/paper/2018/file/e1021d43911ca2c1845910d84f40ae[illegible]Paper.pdf

Outline

- Introduction
  - Why Bayesian meta-learning?
  - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
  - Amortized variational inference
    - Black-box
    - Optimization
  - Bayesian ensembles
  - Bayesian neural networks
- Bayesian meta-learning evaluation
  - Qualitative visualization
  - Quantitative evaluation
  - Active-learning evaluation

Bayesian neural networks

- Key idea: explicit distribution over the space of network parameters

- Monte Carlo dropout in neural networks can be used to perform variational inference to make it Bayesian

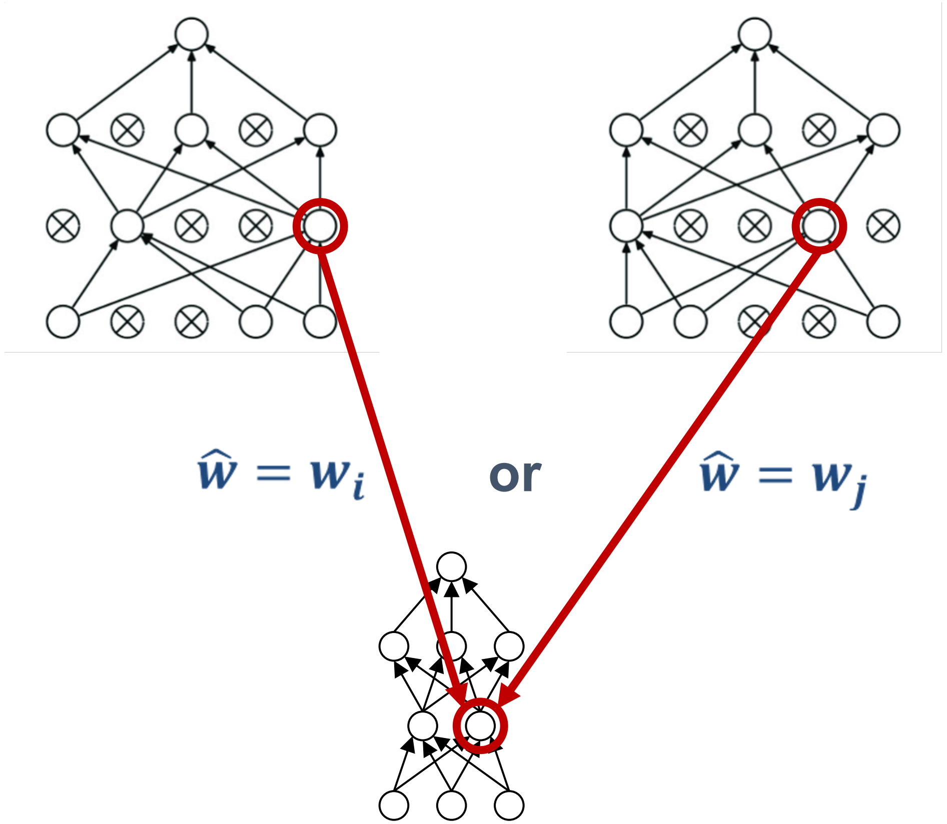
model parameter is sampled by dropping different neurons

$q(\boldsymbol{\omega})$:

$$\mathbf{W}_i = \mathbf{M}_i \cdot \text{diag}([\mathbf{z}_{i,j}]_{j=1}^{K_i})$$
$$\mathbf{z}_{i,j} \sim \text{Bernoulli}(p_i) \text{ for } i = 1, ..., L, \; j = 1, ..., K_{i-1}$$

Easiest way to get a distribution approximating the posterior distribution

$$q(\boldsymbol{\omega}) \xrightarrow{\text{Approx.}} p(\boldsymbol{\omega}|\mathcal{D}^{\text{tr}})$$



https://arxiv.org/pdf/1506.02142.pdf
https://www.hvnguyen.com/bayesiandeeplearning.

Bayesian neural networks

Combine the SGD with Bayesian neural networks (AGILE)

$$q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta) \to p(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)$$

1. Initial learner's parameter with θ
2. Calculate the inner loop loss on $\mathcal{D}_i^{\text{tr}}$ using model with dropout parametrized by ϕ_i
3. Update parameter ϕ_i using SGD
4. Go back to step 2 for more gradient descent steps

Loss function:

- Inner loop loss for optimizing ϕ_i : $\lambda_i = \arg\max_{\lambda_i} \log p\left(\mathcal{D}_i^{\text{tr}} \mid \phi_i\right) + \log p\left(\phi_i \mid \theta\right)$, $\lambda_i = \mathrm{M}_i$ is a set of complete parameter (no dropout)
- Outer loop loss for optimizing θ :

$$\begin{aligned}\theta^* &= \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{i=1}^{M} p(\mathcal{D}_i^{\text{te}}|\mathcal{D}_i^{\text{tr}},\theta) \\ &= \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{i=1}^{M} \left(\int p(\mathcal{D}_i^{\text{te}}|\phi_i)p(\phi_i|\mathcal{D}_i^{\text{tr}},\theta)d\phi_i\right) \\ &\approx \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{i=1}^{M} \left(\frac{1}{T}\sum_{t=1}^{t} p(y_i^{\text{te}}|x_i^{\text{te}},\phi_i^t)\right), \quad \text{where } \phi_i^t \sim q(\phi_i|\mathcal{D}_i^{\text{tr}},\theta).\end{aligned}$$

https://arxiv.org/pdf/200[illegible]5009.pdf.

Bayesian neural networks

Combine the SGD with Bayesian neural networks (AGILE)

- Dropout as well during test

$$p(\mathbf{y}^{\mathrm{te}}|\mathbf{x}^{\mathrm{te}}, \mathcal{D}^{\mathrm{tr}}, \theta) = \int p(\mathbf{y}^{\mathrm{te}}|\mathbf{x}^{\mathrm{te}}, \phi) q(\phi|\mathcal{D}^{\mathrm{tr}}, \theta) d\phi$$

$$\approx \frac{1}{T} \sum_{t=1}^{t} p(y^{\mathrm{te}}|x^{\mathrm{te}}, \underline{\phi^t}), \quad \text{where } \phi^t \sim q(\phi|\mathcal{D}^{\mathrm{tr}}, \theta)$$

Not the complete parameter set M

Pros: Simple, only one model instance

Cons: Can only model Gaussian distribution (Bayesian neural network), need to finetune the hyperparameter dropout rate

$\delta(\phi_i^*)$	✗	$p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta^*\right)$	✓
$\delta(\theta^*)$	✓	$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$	✗

https://arxiv.org/pdf/200[illegible]5009.pdf.

Bayesian meta-learning approaches summary

- **Latent variable models + variational inference**

approximate likelihood of latent variable model with variational lower bound

Approximating $p\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta^*\right)$

Pros:

+ can represent non-Gaussian distributions over y^{ts}
+ produces distribution over functions

Cons:

- Can only represent Gaussian distributions $p(\phi_i|\theta)$

$p\left(\theta \mid \mathcal{D}_{\text{meta-train}}\right)$

Pros: Non-Gaussian posterior, simple at test time, only one model instance.

Con: More complex training procedure.

- **Bayesian ensembles**

particle-based representation: train separate models on bootstraps of the data

Pros: Simple, tends to work well, non-Gaussian distributions.

Con: Need to maintain M model instances. (or do gradient-based inference on **last layer only**)

- **Bayesian neural networks**

explicit distribution over the space of network parameters

Pros: Simple, only one model instance

Cons: Can only model Gaussian distribution (Bayesian neural network), need to finetune the hyperparameter dropout rate

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learni 2020.pdf

Outline

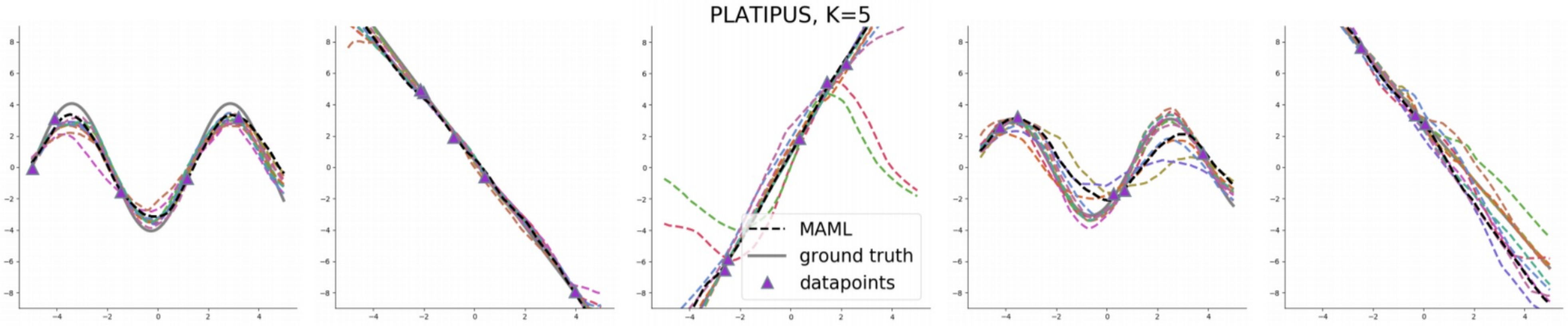
- Introduction
  - Why Bayesian meta-learning?
  - The evidence lower bound (ELBO)
- Bayesian meta-learning approaches based on
  - Amortized variational inference
    - Black-box
    - Optimization
  - Bayesian ensembles
  - Bayesian neural networks
- Bayesian meta-learning evaluation
  - Qualitative visualization
  - Quantitative evaluation
  - Active-learning evaluation

Qualitative visualization

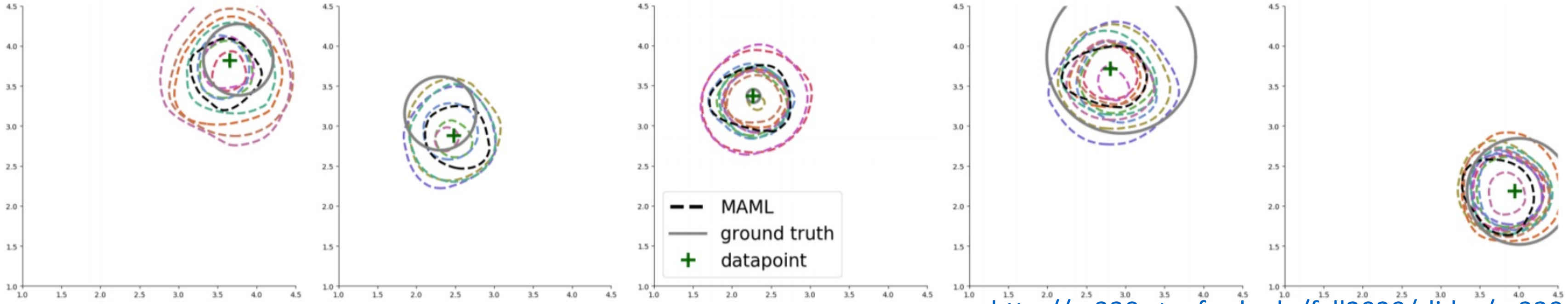
Qualitative Evaluation on Toy Problems with Ambiguity

(Finn*, Xu*, Levine, NeurIPS ’18)

Ambiguous regression:



Ambiguous classification:

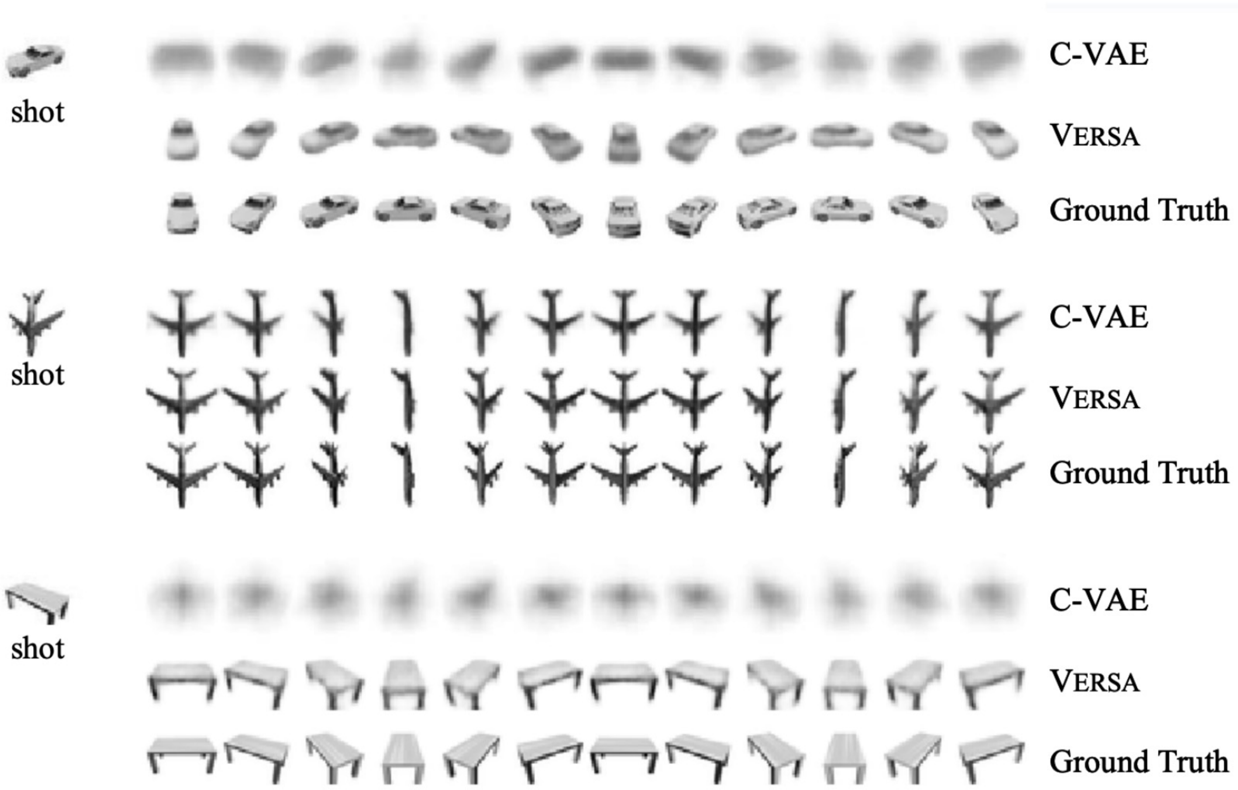


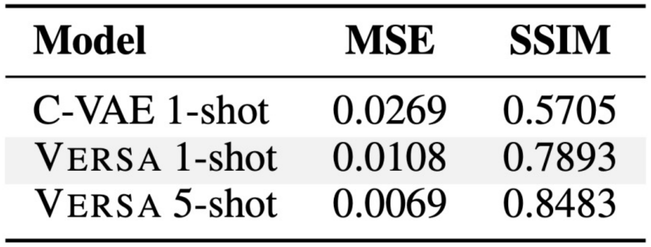
http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Quantitative evaluation

Evaluation on Ambiguous Generation Tasks

(Gordon et al., ICLR ’19)





Model	MSE	SSIM
C-VAE 1-shot	0.0269	0.5705
VERSA 1-shot	0.0108	0.7893
VERSA 5-shot	0.0069	0.8483

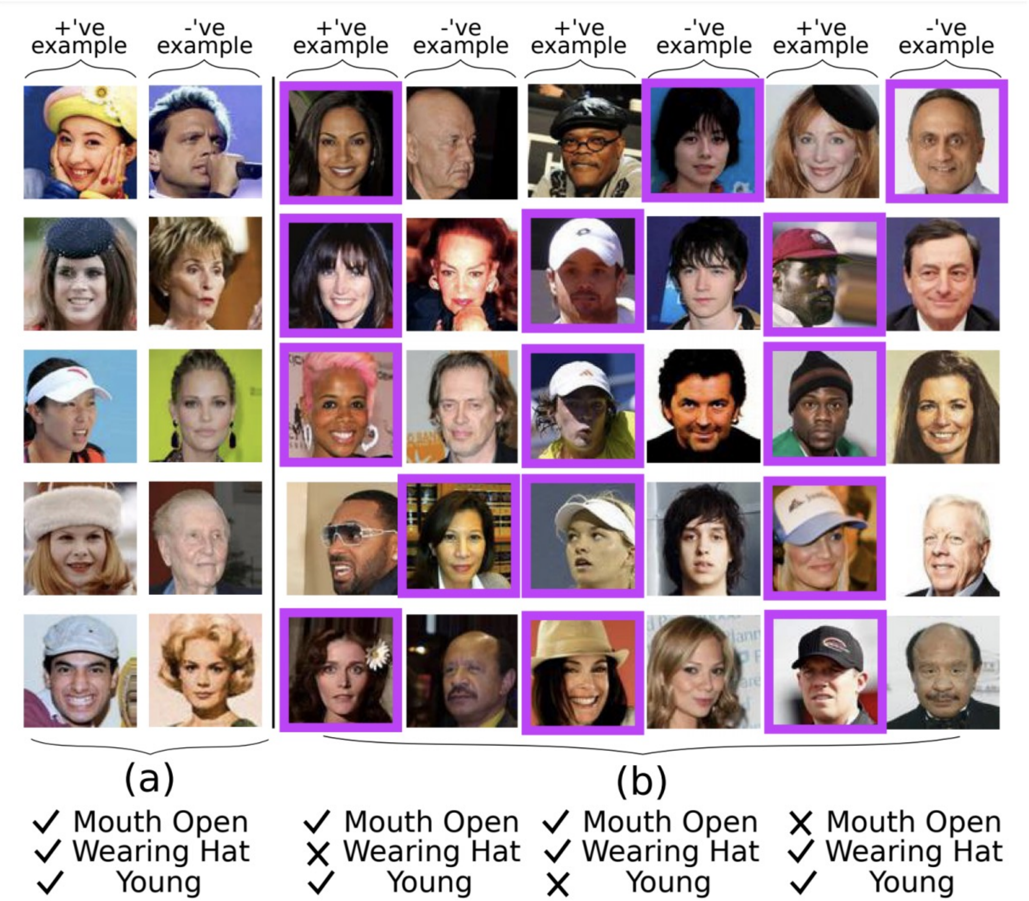
Table 2: View reconstruction test results.

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Quantitative evaluation

Accuracy, Mode Coverage, & Likelihood on Ambiguous Tasks

(Finn*, Xu*, Levine, NeurIPS '18)



Ambiguous celebA (5-shot)			
	Accuracy	Coverage (max=3)	Average NLL
MAML	**89.00 ± 1.78%**	1.00 ± 0.0	0.73 ± 0.06
MAML + noise	84.3 ± 1.60 %	1.89 ± 0.04	0.68 ± 0.05
PLATIPUS (ours) (KL weight = 0.05)	**88.34 ± 1.06** %	1.59 ± 0.03	0.67± 0.05
PLATIPUS (ours) (KL weight = 0.15)	**87.8 ± 1.03** %	**1.94 ± 0.04**	**0.56 ± 0.04**

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Quantitative evaluation

Reliability Diagrams & Accuracy

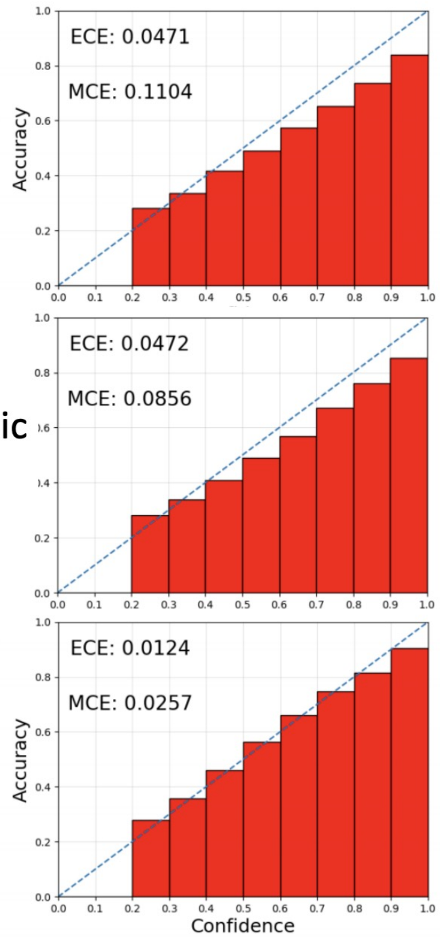
(Ravi & Beatson, ICLR ’19)

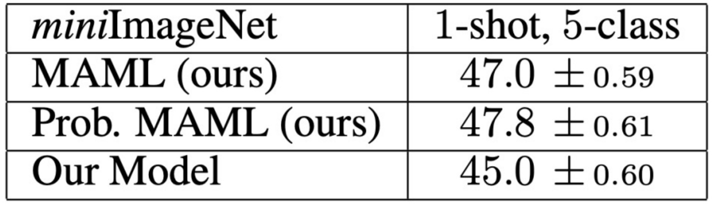
*mini*ImageNet: 1-shot, 5-class

MAML

Probabilistic MAML

Ravi & Beatson



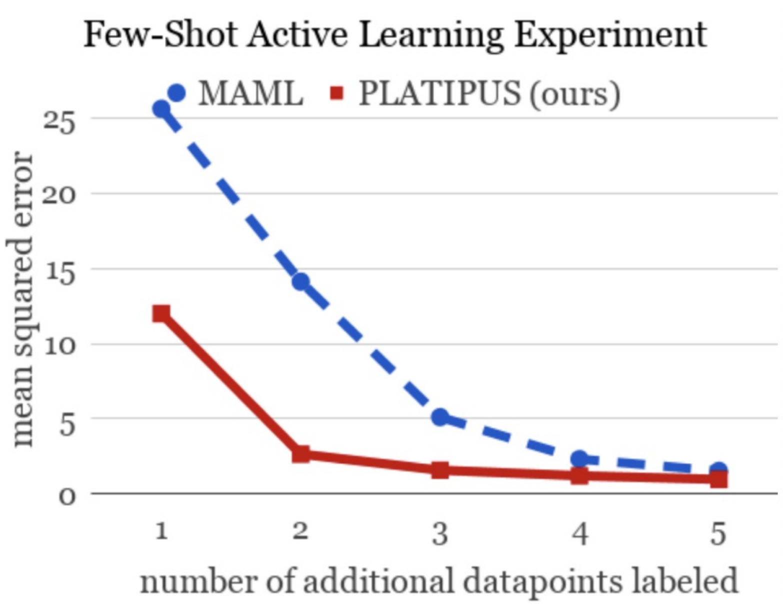


*mini*ImageNet	1-shot, 5-class
MAML (ours)	47.0 ± 0.59
Prob. MAML (ours)	47.8 ± 0.61
Our Model	45.0 ± 0.60

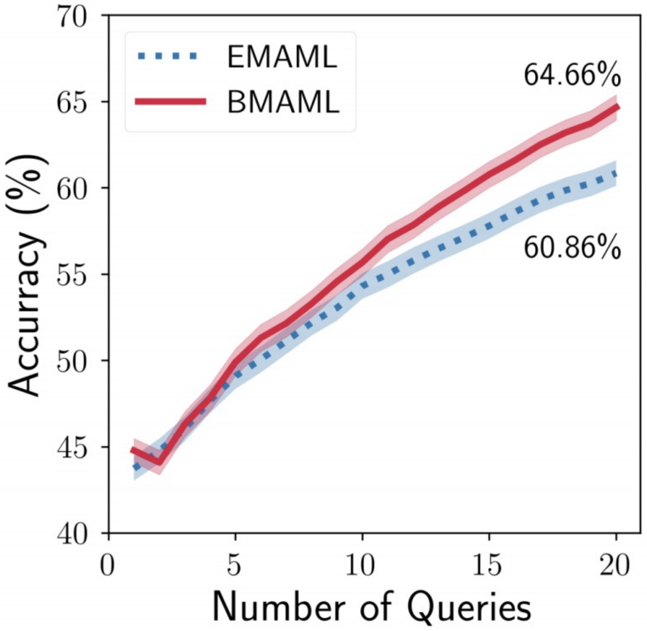
http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learning_2020.pdf

Active-learning evaluation

Finn*, Xu*, Levine, NeurIPS ’18
Sinusoid Regression



Kim et al. NeurIPS ’18
MiniImageNet



Both experiments:

- Sequentially choose datapoint with **maximum predictive entropy** to be labeled
- or choose datapoint at random (MAML)

http://cs330.stanford.edu/fall2020/slides/cs330_bayesian_meta_learni... 2020.pdf

Take away

- Uncertainty is important when study meta-learning as few annotated samples are provided for each tasks
- Uncertainty can exist in either meta parameter or learner's parameter or both
- There are several tools to make the meta learning algorithm Bayesian:
  - Latent variable models + variational inference
  - Bayesian ensembles
  - Bayesian neural networks
- Learn different ways to evaluate the Bayesian meta learning algorithm

рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
kiitos
dankie
nanni
nandri
dhanyavad
mauruuru
köszönöm
thank you
gracias
хвала
asante
manana
tenki
obrigada
bedankt
bayarlalaa
gràçie
hvala
ačiū
dankon
akun
djiere dieuf
tau
mochchakkeram
murakoze
enkosi
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
дякую
mamnun
chokrane
sobodi
dĕkuji
sagolun
sukriya
kop khun krap
taiku
go raibh maith agat
obrigado
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
arigatô
takk
dakujem
trugarez
kam sah hamnida
terima kasih
rahmet
rahmat
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci