Bayesian Meta-Learning

CS 330

Logistics

Homework 2 due **next Wednesday.**

Project proposal due in **two weeks**.

Poster presentation: **Tues 12/3** at **1:30 pm**.

Disclaimers

Bayesian meta-learning is an **active area of research**
(like most of the class content)

More **questions** than answers.

This lecture covers some of the most
advanced topics of the course.

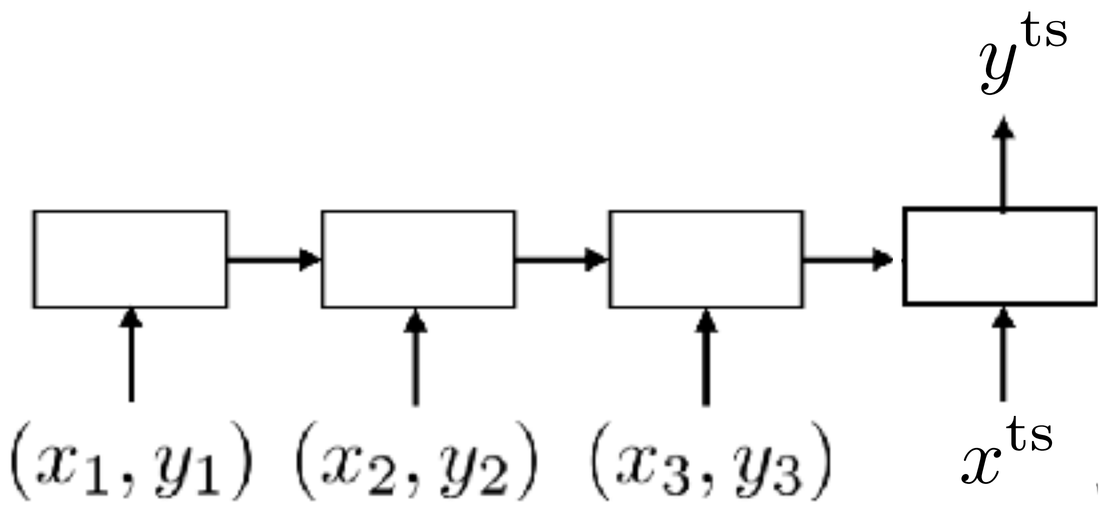
So ask **questions**!

Recap from last time.

***Computation graph* perspective**

Black-box

$$y^{\text{ts}} = f_\theta(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$



Optimization-based

$$\begin{aligned} y^{\text{ts}} &= f_{\text{MAML}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}}) \\ &= f_{\phi_i}(x^{\text{ts}}) \end{aligned}$$

$$\text{where } \phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$$

Non-parametric

$$\begin{aligned} y^{\text{ts}} &= f_{\text{PN}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}}) \\ &= \text{softmax}(-d\left(f_\theta(x^{\text{ts}}), \mathbf{c}_n\right)) \end{aligned}$$

$$\text{where } \mathbf{c}_n = \frac{1}{K} \sum_{(x,y) \in \mathcal{D}_i^{\text{tr}}} \mathbb{1}(y = n) f_\theta(x)$$

Recap from last time.

***Algorithmic properties* perspective**

Expressive power

the ability for f to represent a range of learning procedures

Why? scalability, applicability to a range of domains

Consistency

learned learning procedure will solve task with enough data

Why? reduce reliance on meta-training tasks, good OOD task performance

These properties are important for most applications!

Recap from last time.

***Algorithmic properties* perspective**

Expressive power

the ability for f to represent a range of learning procedures

Why? scalability, applicability to a range of domains

Consistency

learned learning procedure will solve task with enough data

Why? reduce reliance on meta-training tasks, good OOD task performance

Uncertainty awareness

ability to reason about ambiguity during learning

Why? active learning, calibrated uncertainty, RL principled Bayesian approaches

this lecture

Plan for Today

Why be Bayesian?

Bayesian meta-learning approaches

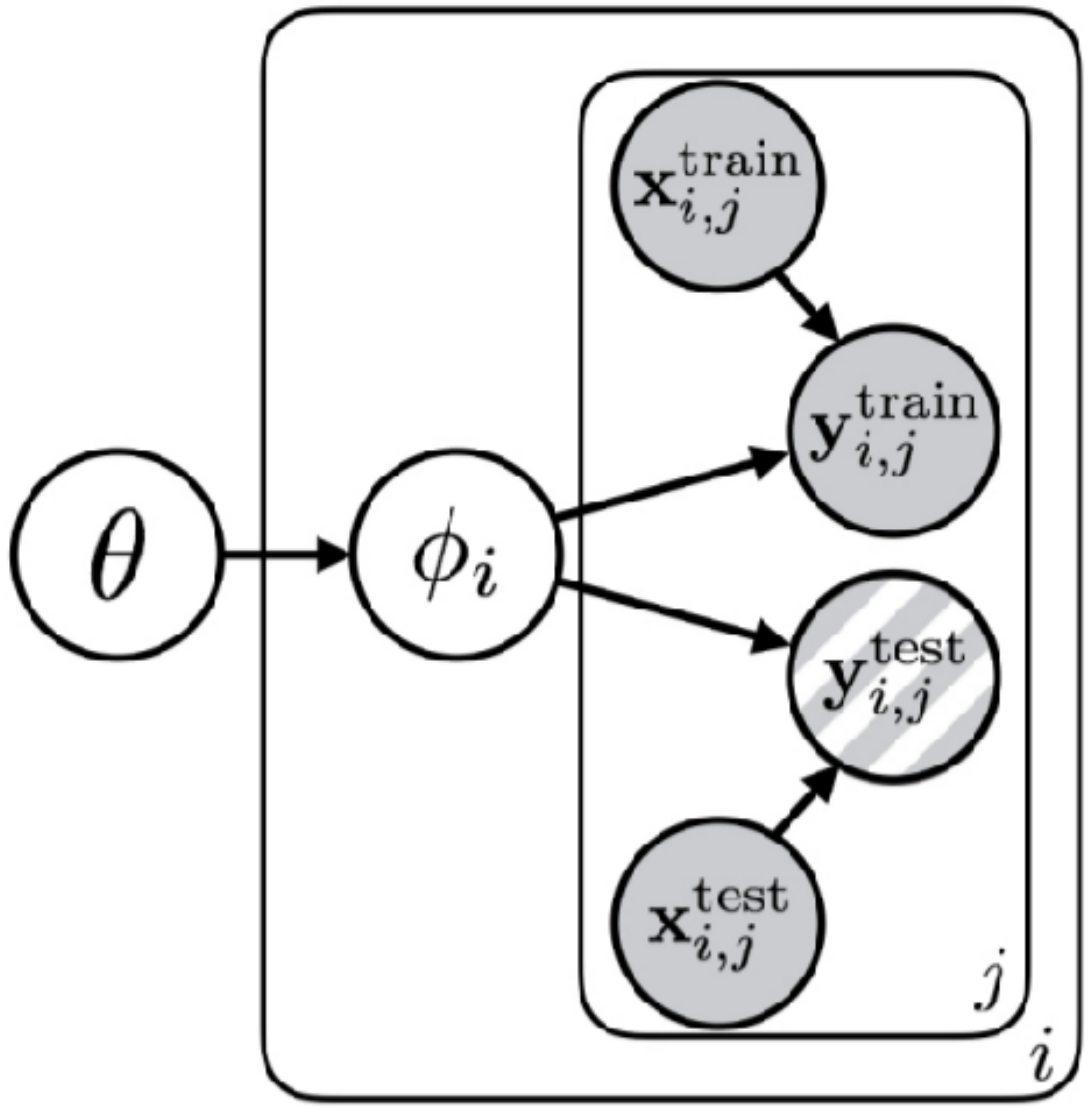
How to evaluate Bayesians.

Multi-Task & Meta-Learning Principles

Training and testing must match.

Tasks must share "structure."

What does "structure" mean? statistical dependence on shared latent information θ



If you condition on that information,

- task parameters become independent
  i.e. $\phi_{i_1} \perp\!\!\!\perp \phi_{i_2} \mid \theta$
  and are not otherwise independent $\phi_{i_1} \not\perp\!\!\!\perp \phi_{i_2}$
- hence, you have a lower entropy
  i.e. $\mathscr{H}(p(\phi_i \mid \theta)) < \mathscr{H}(p(\phi_i))$

Thought exercise #1: If you can identify θ (i.e. with meta-learning), when should learning ϕ_i be faster than learning from scratch?

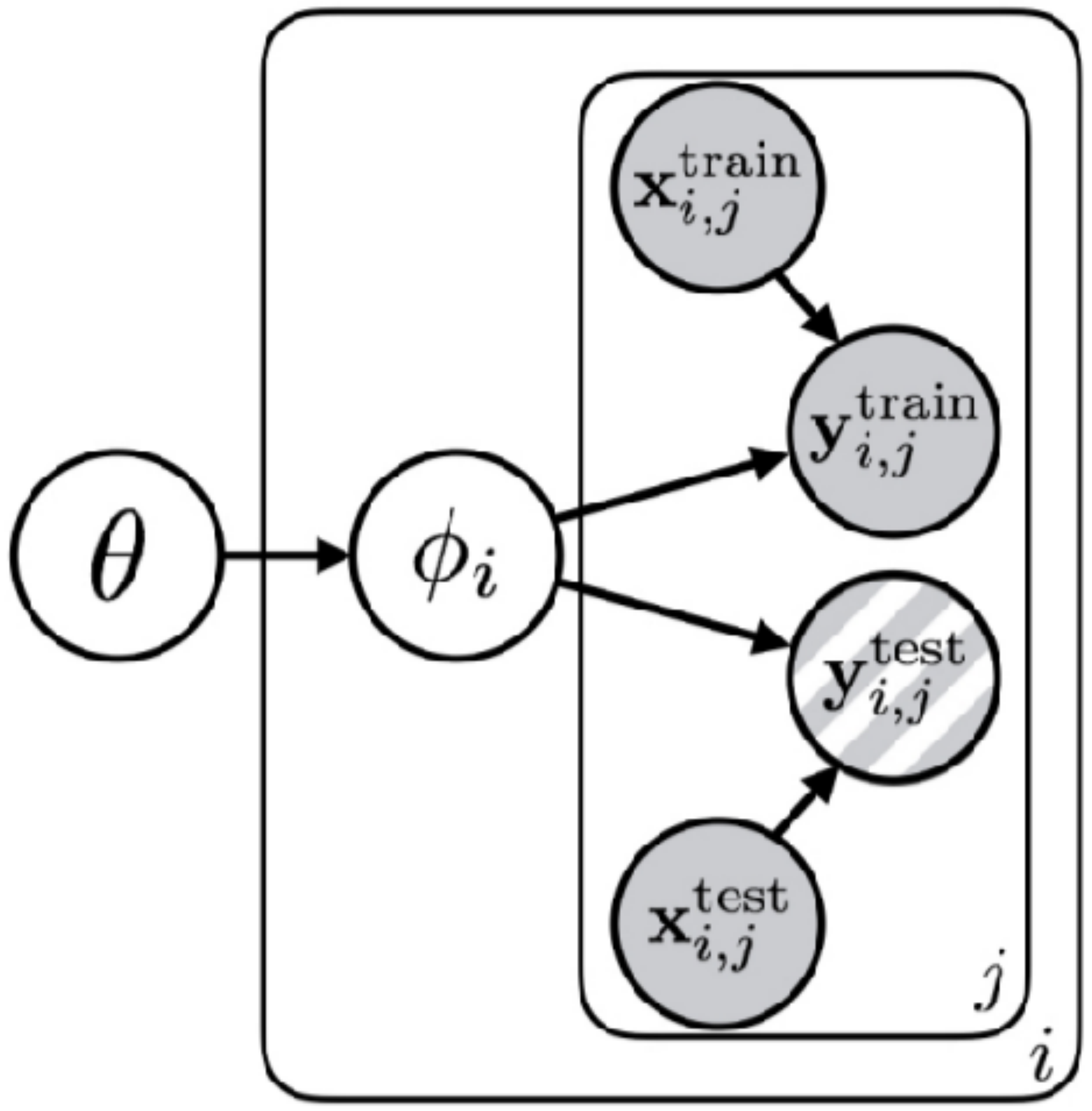
Thought exercise #2: what if $\mathscr{H}(p(\phi_i \mid \theta)) = 0$?

Multi-Task & Meta-Learning Principles

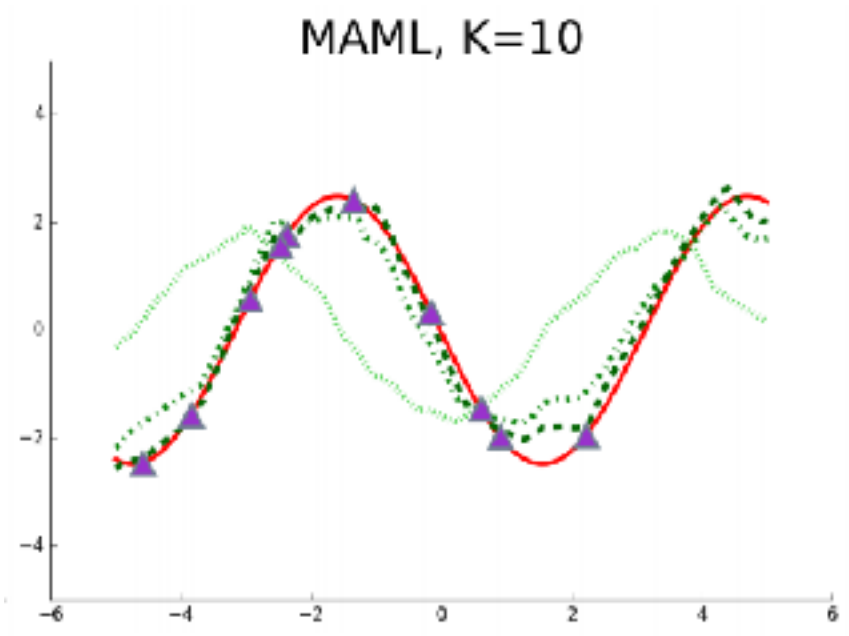
Training and testing must match.

Tasks must share "structure."

What does "structure" mean? statistical dependence on shared latent information θ

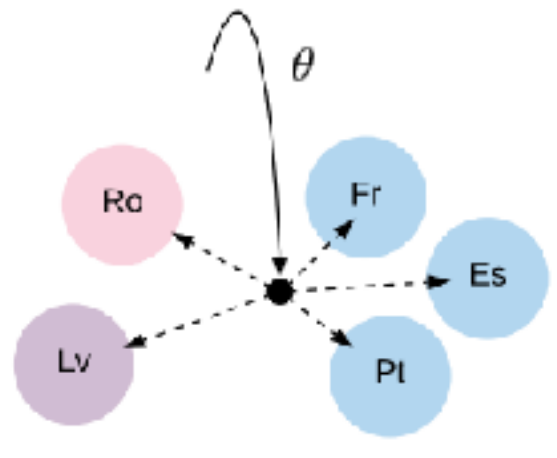


What information might θ contain...



...in the toy sinusoid problem?

θ corresponds to family of sinusoid functions (everything but phase and amplitude)



(c) Meta Learning

...in the machine translation example?

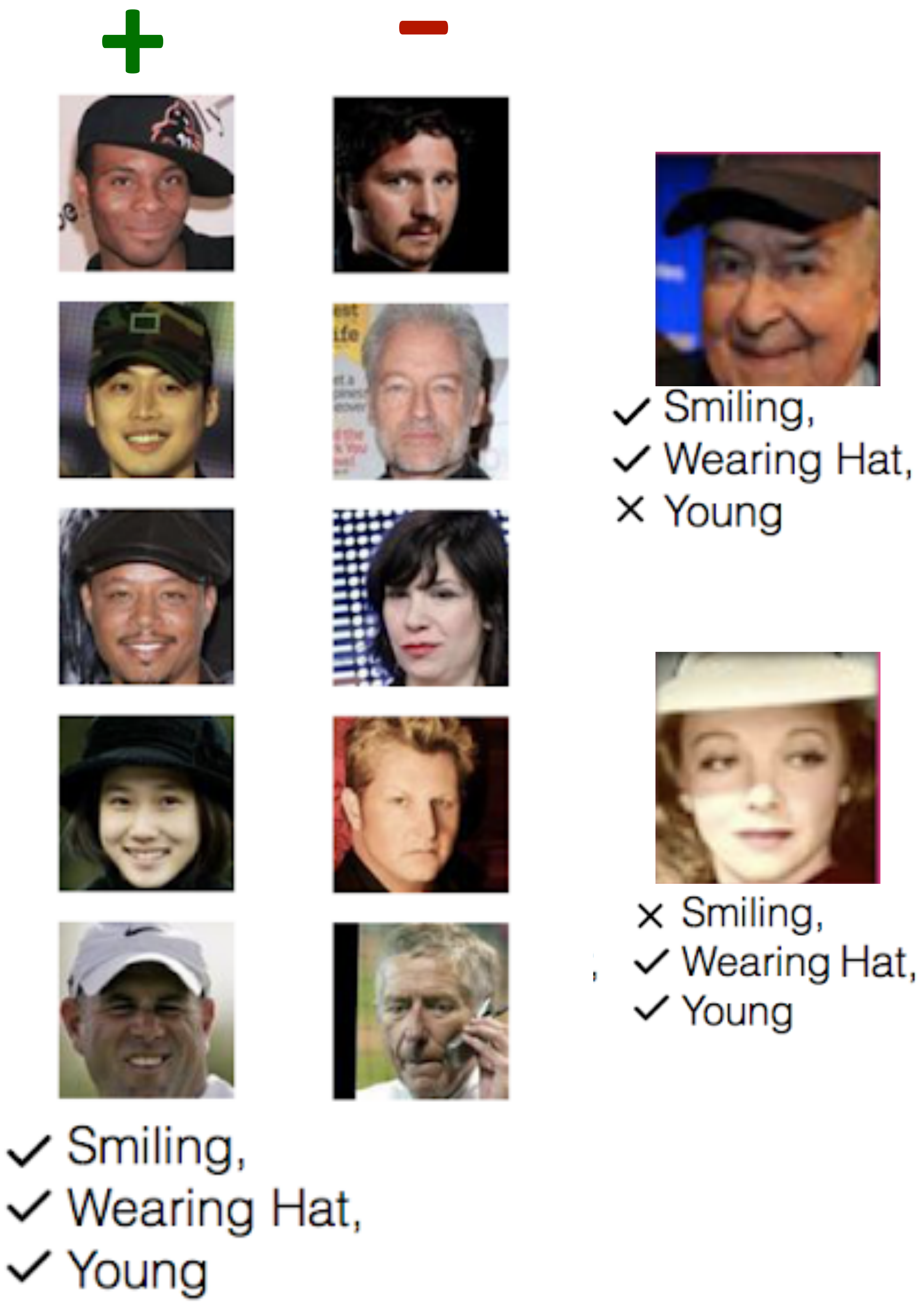
θ corresponds to the family of all language pairs

Note that θ is narrower than the space of all possible functions.

Thought exercise #3: What if you meta-learn without a lot of tasks?

"meta-overfitting"

Recall parametric approaches: Use **deterministic** $p(\phi_i|\mathcal{D}_i^{\mathrm{tr}}, \theta)$ (i.e. a point estimate)



Why/when is this a problem?

Few-shot learning problems may be ***ambiguous***.
(even with prior)

Can we learn to *generate hypotheses* about the underlying function?
i.e. sample from $p(\phi_i|\mathcal{D}_i^{\mathrm{tr}}, \theta)$

Important for:

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

Active learning w/ meta-learning: Woodward & Finn ’16, Konyushkova et al. ’17, Bachman et al. ’17

Plan for Today

Why be Bayesian?

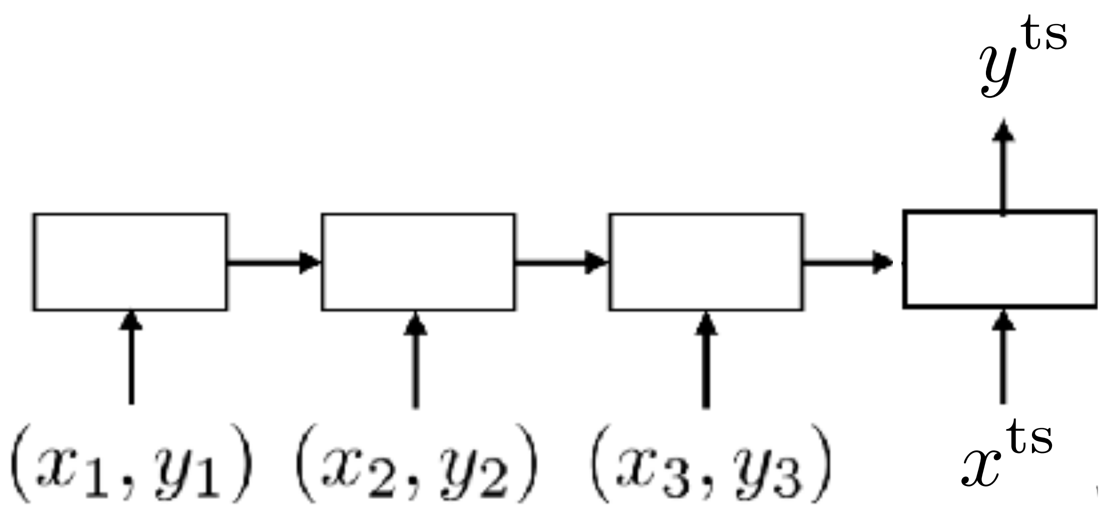
Bayesian meta-learning approaches

How to evaluate Bayesians.

Computation graph perspective

Black-box

$$y^{\text{ts}} = f_\theta(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$



Optimization-based

$$y^{\text{ts}} = f_{\text{MAML}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$
$$= f_{\phi_i}(x^{\text{ts}})$$
$$\text{where } \phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$$

Non-parametric

$$y^{\text{ts}} = f_{\text{PN}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$
$$= \text{softmax}(-d\left(f_\theta(x^{\text{ts}}), \mathbf{c}_n\right))$$
$$\text{where } \mathbf{c}_n = \frac{1}{K} \sum_{(x,y)\in\mathcal{D}_i^{\text{tr}}} \mathbb{1}(y=n) f_\theta(x)$$

Version 0: Let f output the parameters of a distribution over y^{ts}.

For example:

- probability values of discrete **categorical distribution**
- mean and variance of a **Gaussian**
- means, variances, and mixture weights of a **mixture of Gaussians**
- for multi-dimensional y^{ts}: parameters of a **sequence of distributions** (i.e. autoregressive model)

Then, optimize with maximum likelihood.

Version 0: Let f output the parameters of a distribution over y^{ts}.

For example:

- probability values of discrete **categorical distribution**
- mean and variance of a **Gaussian**
- means, variances, and mixture weights of a **mixture of Gaussians**
- for multi-dimensional y^{ts}: parameters of a **sequence of distributions** (i.e. autoregressive model)

Then, optimize with **maximum likelihood**.

Pros:

\+ simple

\+ can combine with variety of methods

Cons:

- can't reason about uncertainty over the underlying function [to determine how uncertainty across datapoints relate]
- limited class of distributions over y^{ts} can be expressed
- tends to produce poorly-calibrated uncertainty estimates

Thought exercise #4: Can you do the same maximum likelihood training for ϕ?

The Bayesian Deep Learning Toolbox

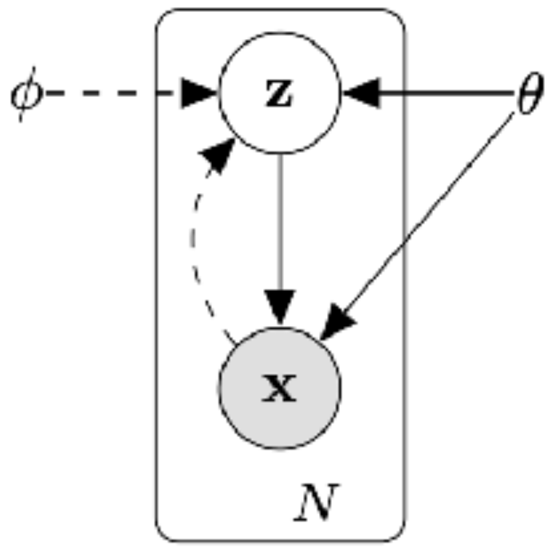
a broad one-slide overview

(CS 236 provides a thorough treatment)

Goal: represent distributions with neural networks

Latent variable models + variational inference (Kingma & Welling ‘13, Rezende et al. ‘14):

- approximate likelihood of latent variable model with variational lower bound

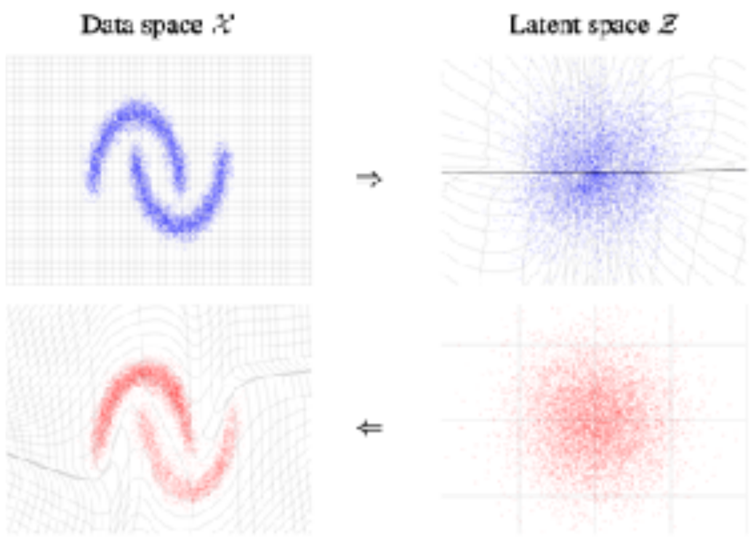


Bayesian ensembles (Lakshminarayanan et al. ‘17):

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

Bayesian neural networks (Blundell et al. ‘15):

- explicit distribution over the space of network parameters

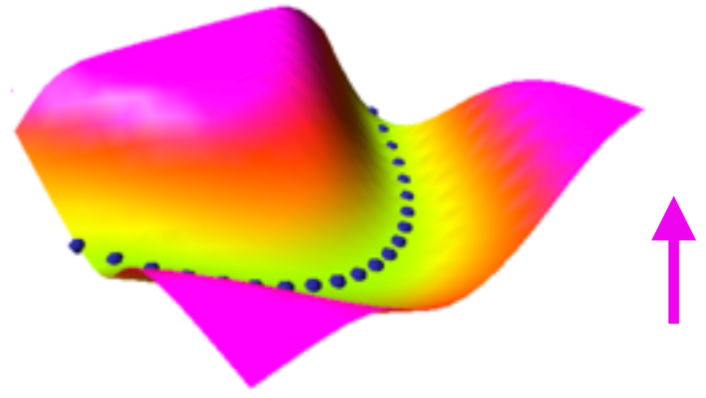


Normalizing Flows (Dinh et al. ‘16):

- invertible function from latent distribution to data distribution

Energy-based models & GANs (LeCun et al. ’06, Goodfellow et al. ‘14):

- estimate unnormalized density



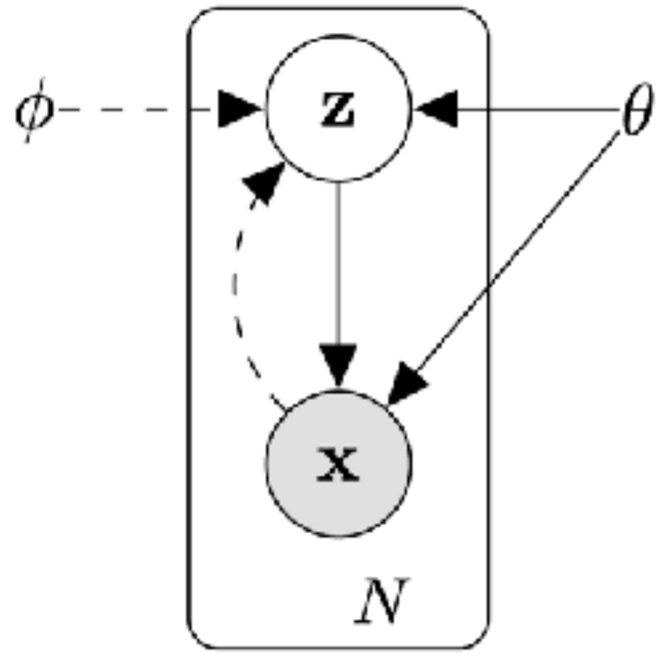
data

everything else

We’ll see how we can leverage the first two.

The others could be useful in developing new methods.

Background: The Variational Lower Bound



Observed variable x, latent variable z

ELBO: $\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log p(x, z)\right] + \mathcal{H}(q(z|x))$

Can also be written as: $= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - D_{KL}\left(q(z|x) \| p(z)\right)$

p: model
- $p(x|z)$ represented w/ neural net,
- $p(z)$ represented as $\mathcal{N}(\mathbf{0}, \mathbf{I})$

model parameters θ,
variational parameters ϕ

$q(z|x)$: inference network, variational distribution

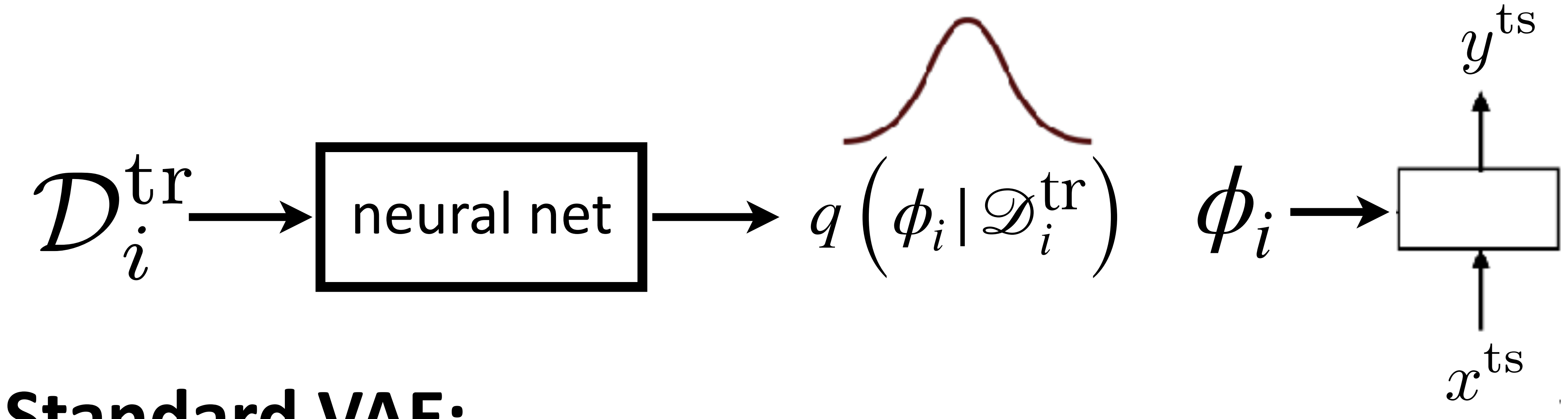
Problem: need to backprop through sampling
i.e. compute derivative of $\mathbb{E}_q$ w.r.t. q

Reparametrization trick For Gaussian $q(z|x)$:
$q(z|x) = \mu_q + \sigma_q \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Can we use amortized variational inference for meta-learning?

Bayesian black-box meta-learning

with standard, deep variational inference



Standard VAE:

Observed variable x, latent variable z

ELBO: $\mathbb{E}_{q(z|x)}\left[\log p(x \mid z)\right] - D_{KL}\left(q(z \mid x) \| p(z)\right)$

p: model, represented by a neural net

q: inference network, variational distribution

↓

Meta-learning:

Observed variable $\mathscr{D}$, latent variable ϕ

$$\max \mathbb{E}_{q(\phi)}\left[\log p(\mathscr{D} \mid \phi)\right] - D_{KL}\left(q(\phi) \| p(\phi)\right)$$

What should q condition on?

$$\max \mathbb{E}_{q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}\right)}\left[\log p(\mathscr{D} \mid \phi)\right] - D_{KL}\left(q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}\right) \| p(\phi)\right)$$

$$\max \mathbb{E}_{q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}\right)}\left[\log p\left(y^{\mathrm{ts}} \mid x^{\mathrm{ts}}, \phi\right)\right] - D_{KL}\left(q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}\right) \| p(\phi)\right)$$

What about the meta-parameters θ?

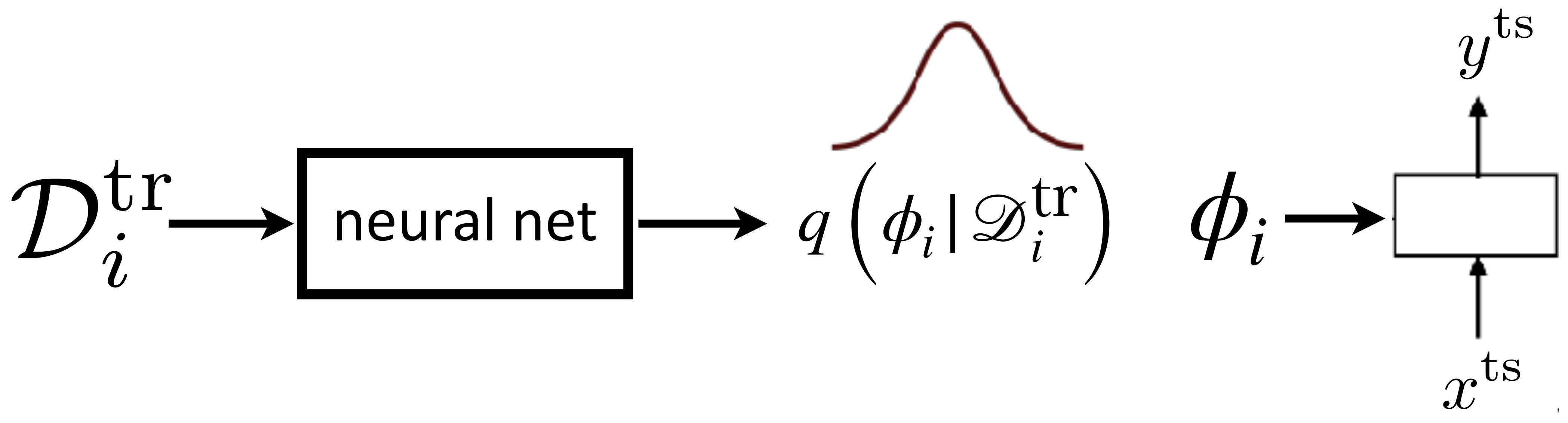
$$\max_{\theta} \mathbb{E}_{q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}, \theta\right)}\left[\log p\left(y^{\mathrm{ts}} \mid x^{\mathrm{ts}}, \phi\right)\right] - D_{KL}\left(q\left(\phi \mid \mathscr{D}^{\mathrm{tr}}, \theta\right) \| p(\phi \mid \theta)\right)$$

Can also condition on θ here

Final objective (for completeness): $\max_{\theta} \mathbb{E}_{\mathscr{T}_i}\left[\mathbb{E}_{q\left(\phi_i \mid \mathscr{D}_i^{\mathrm{tr}}, \theta\right)}\left[\log p\left(y_i^{\mathrm{ts}} \mid x_i^{\mathrm{ts}}, \phi_i\right)\right] - D_{KL}\left(q\left(\phi_i \mid \mathscr{D}_i^{\mathrm{tr}}, \theta\right) \| p(\phi_i \mid \theta)\right)\right]$

Bayesian black-box meta-learning

with standard, deep variational inference



$$\max_\theta \mathbb{E}_{\mathscr{T}_i}\left[\mathbb{E}_{q\left(\phi_i \mid \mathscr{D}_i^{\mathrm{tr}}, \theta\right)}\left[\log p\left(y_i^{\mathrm{ts}} \mid x_i^{\mathrm{ts}}, \phi_i\right)\right] - D_{KL}\left(q\left(\phi_i \mid \mathscr{D}_i^{\mathrm{tr}}, \theta\right) \| p(\phi_i \mid \theta)\right)\right]$$

Pros:

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

Cons:

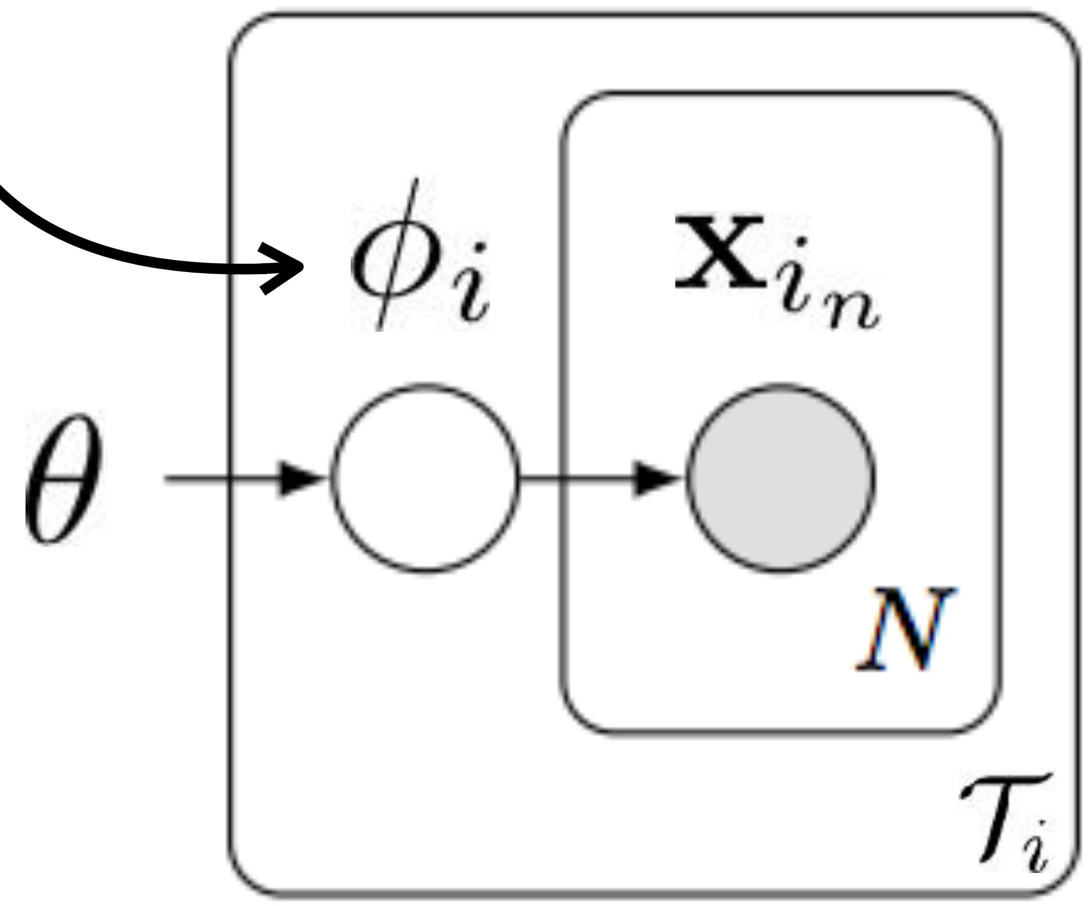
- Can only represent Gaussian distributions $p(\phi_i \mid \theta)$

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

What about Bayesian optimization-based meta-learning?

Recall: *Recasting Gradient-Based Meta-Learning as Hierarchical Bayes* (Grant et al. '18)

task-specific parameters



meta-parameters

$$\max_\theta \log \prod_i p(\mathcal{D}_i|\theta)$$

$$= \log \prod_i \int p(\mathcal{D}_i|\phi_i)p(\phi_i|\theta)d\phi_i \quad \text{(empirical Bayes)}$$

$$\approx \log \prod_i p(\mathcal{D}_i|\hat{\phi}_i)p(\hat{\phi}_i|\theta)$$

MAP estimate

How to compute MAP estimate?

Gradient descent with **early stopping** = **MAP inference** under **Gaussian prior** with mean at initial parameters [Santos '96]

(exact in linear case, approximate in nonlinear case)

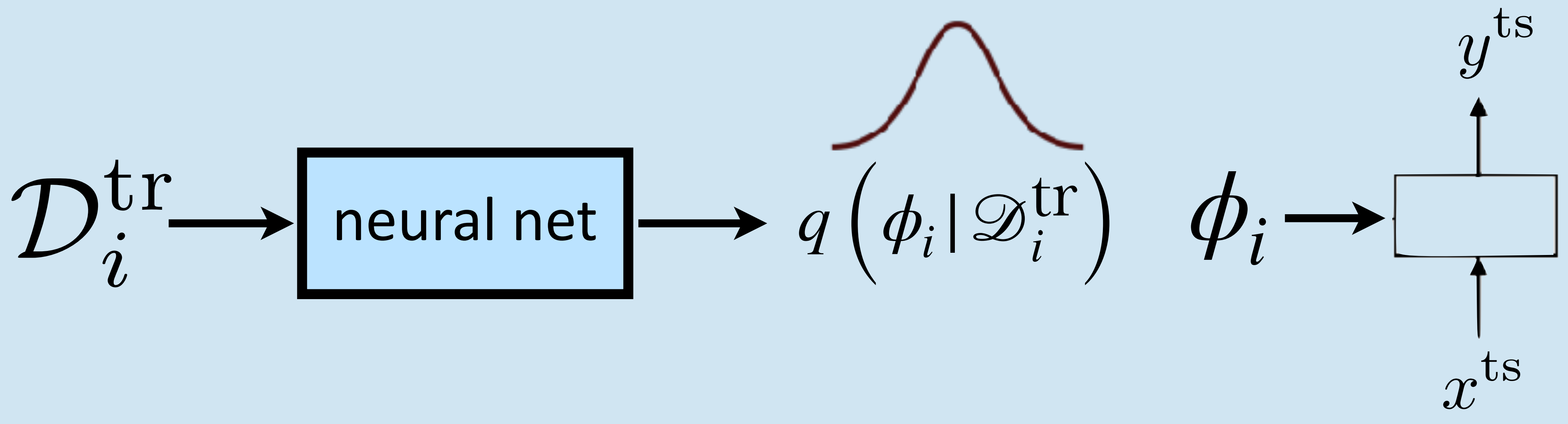
Provides a Bayesian interpretation of MAML.

But, we can't **sample** from $p\left(\phi_i \,|\, \theta, \mathscr{D}_i^{\text{tr}}\right)$!

What about Bayesian **optimization-based** meta-learning?

Recall: Bayesian black-box meta-learning

with standard, deep variational inference



$$\max_{\theta} \mathbb{E}_{\mathcal{T}_i}\left[\mathbb{E}_{q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right)}\left[\log p\left(y_i^{\mathrm{ts}} \mid x_i^{\mathrm{ts}}, \phi_i\right)\right] - D_{KL}\left(q\left(\phi_i \mid \mathcal{D}_i^{\mathrm{tr}}, \theta\right) \| p(\phi_i \mid \theta)\right)\right]$$

q: an arbitrary function

q can include a gradient operator!

Amortized Bayesian Meta-Learning
(Ravi & Beatson '19)

q corresponds to SGD on the mean & variance of neural network weights $(\mu_\phi, \sigma_\phi^2)$, w.r.t. $\mathcal{D}_i^{\mathrm{tr}}$

Pro: Running gradient descent at test time. **Con**: $p(\phi_i \mid \theta)$ modeled as a Gaussian.

Can we model **non-Gaussian** posterior?

What about Bayesian optimization-based meta-learning?

Can we use ensembles?

Kim et al. Bayesian MAML '18

An ensemble of mammals

Ensemble of MAMLs (EMAML)

Train M independent MAML models.

Won't work well if ensemble members are too similar.

Note: Can also use ensembles w/ black-box, non-parametric methods!

A more diverse ensemble of mammals

Stein Variational Gradient (BMAML)

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

$$\phi(\theta_t) = \frac{1}{M}\sum_{j=1}^{M}\left[k(\theta_t^j, \theta_t)\nabla_{\theta_t^j}\log p(\theta_t^j) + \underline{\nabla_{\theta_t^j} k(\theta_t^j, \theta_t)}\right]$$

Optimize for distribution of M particles to produce high likelihood.

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

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**)

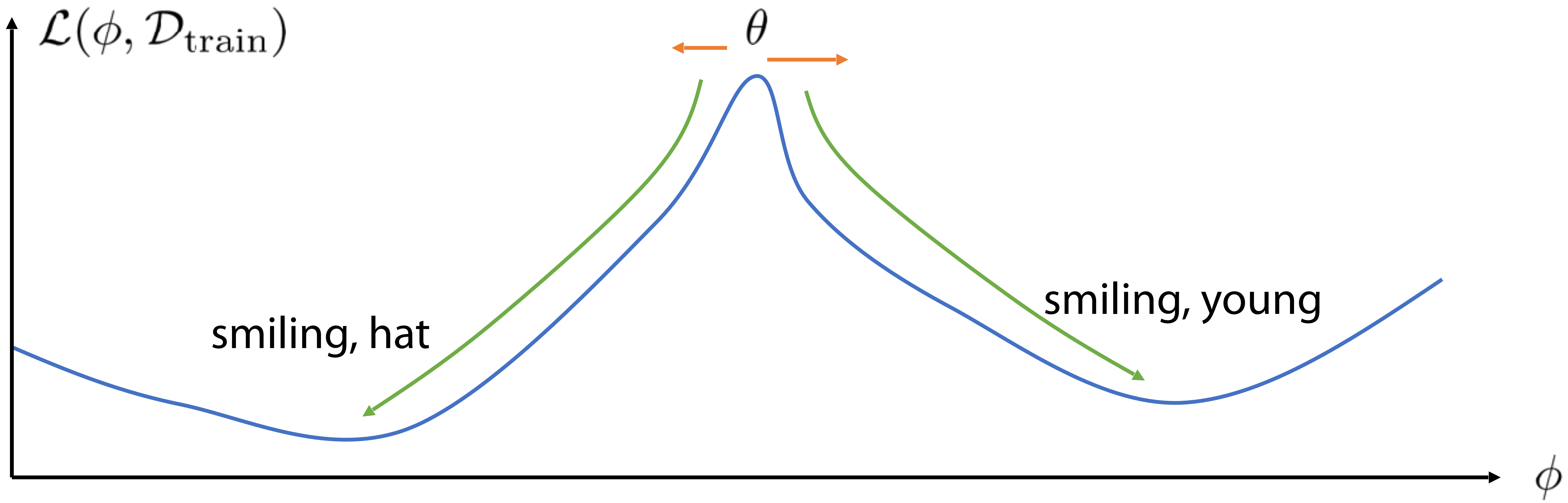
Can we model **non-Gaussian** posterior over **all parameters**?

What about Bayesian optimization-based meta-learning?

Sample parameter vectors with a procedure like Hamiltonian Monte Carlo?

Finn*, Xu*, Levine. Probabilistic MAML '18

Intuition: Learn a prior where a random kick can put us in different modes



$$\phi \leftarrow \theta + \epsilon$$

$$\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}(\phi, \mathcal{D}_{\text{train}})$$

What about Bayesian **optimization-based** meta-learning?

Sample parameter vectors with a procedure like **Hamiltonian Monte Carlo**?

Finn*, Xu*, Levine. Probabilistic MAML '18

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

(not single parameter vector anymore)

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

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

$\Rightarrow$ this is completely intractable!

what if we knew $p(\phi_i | \theta, x_i^{\text{train}}, y_i^{\text{train}})$?

$\Rightarrow$ now sampling is easy! just use ancestral sampling!

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

approximate with MAP

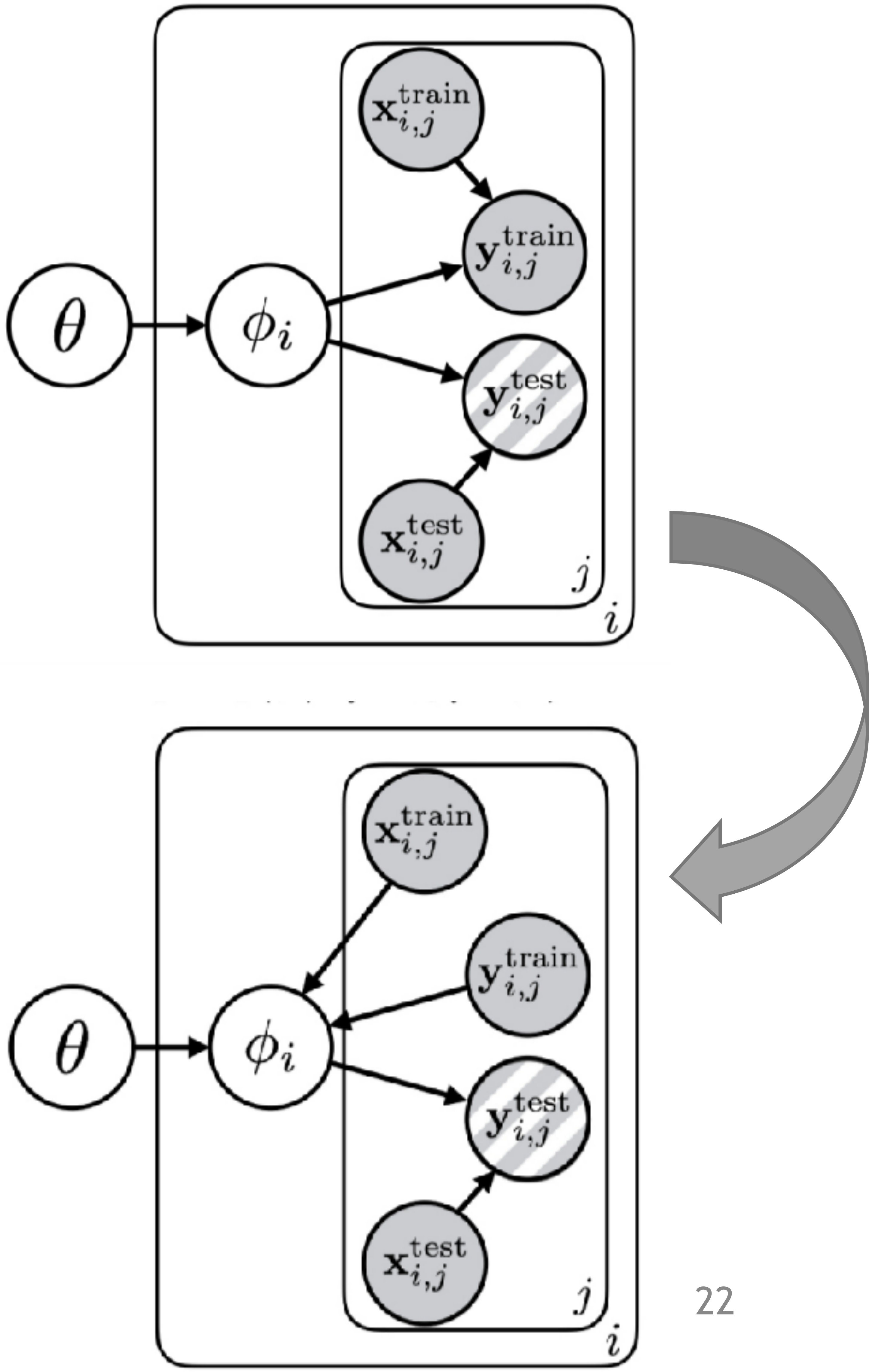
this is **extremely** crude

but **extremely** convenient!

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

(Santos '92, Grant et al. ICLR '18)

Training can be done with **amortized variational inference**.



What about Bayesian **optimization-based** meta-learning?

Sample parameter vectors with a procedure like **Hamiltonian Monte Carlo**?

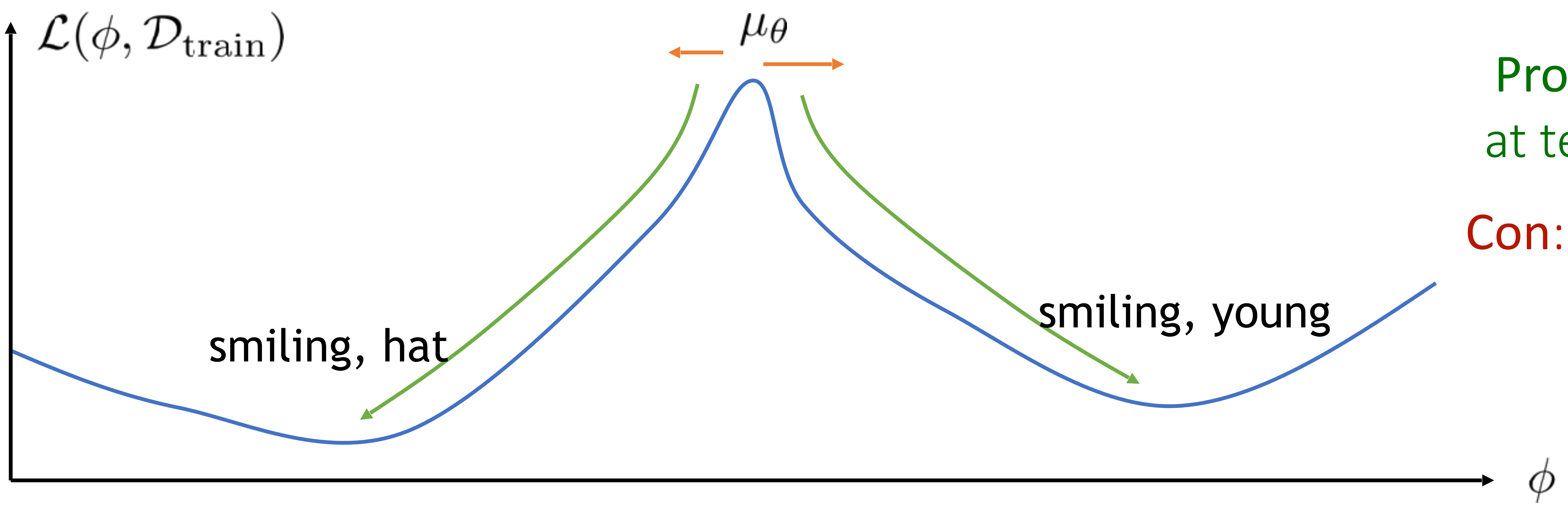
Finn*, Xu*, Levine. Probabilistic MAML '18

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

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

What does ancestral sampling look like?

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



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

Con: More complex training procedure.

Methods Summary

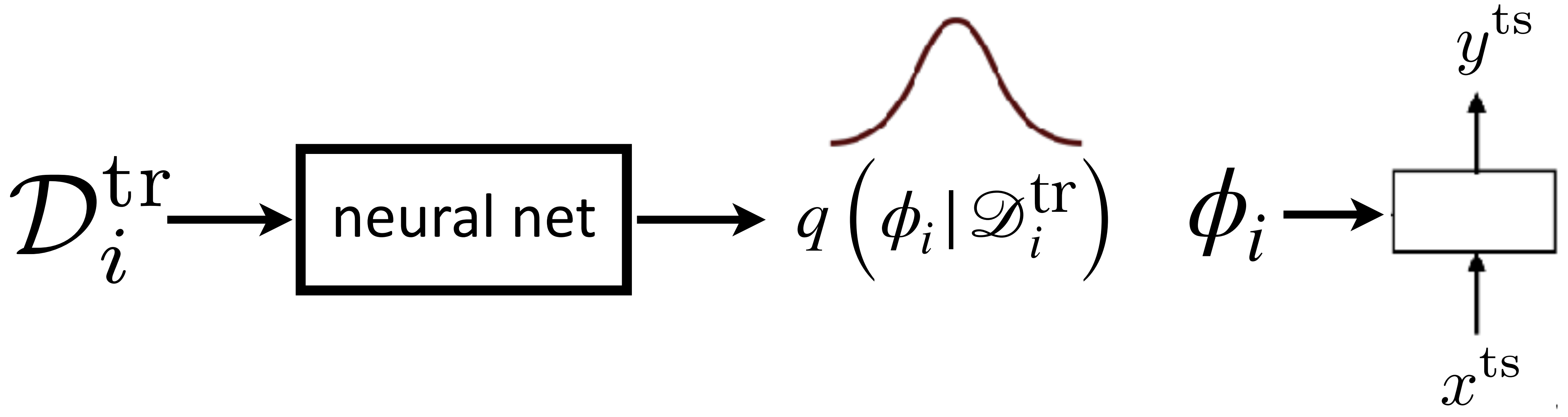
Version 0: f outputs a distribution over y^{ts}.

Pros: simple, can combine with variety of methods

Cons: can't reason about uncertainty over the underlying function,

limited class of distributions over y^{ts} can be expressed

Black box approaches: Use latent variable models + amortized variational inference



Pros: can represent non-Gaussian distributions over y^{ts}

Cons: Can only represent Gaussian distributions $p(\phi_i \mid \theta)$

(okay when ϕ_i is latent vector)

Optimization-based approaches:

Amortized inference

Pro: Simple.

Con: $p(\phi_i \mid \theta)$ modeled as a Gaussian.

Ensembles

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

Con: maintain M model instances.

(or do inference on **last layer only**)

Hybrid inference

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

Con: More complex training procedure.

Plan for Today

Why be Bayesian?

Bayesian meta-learning approaches

How to evaluate Bayesians.

How to evaluate a Bayesian meta-learner?

Use the standard benchmarks?
(i.e. MiniImagenet accuracy)

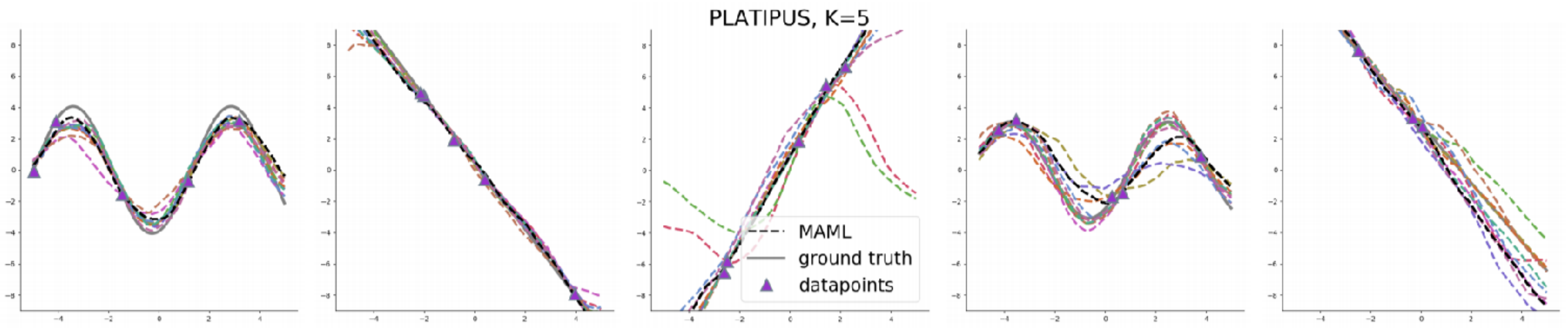
+ standardized
+ real images
+ good check that the approach didn’t break anything
- metrics like accuracy don't evaluate uncertainty
- tasks may not exhibit ambiguity
- uncertainty may not be useful on this dataset!

What are better problems & metrics?
It depends on the problem you care about!

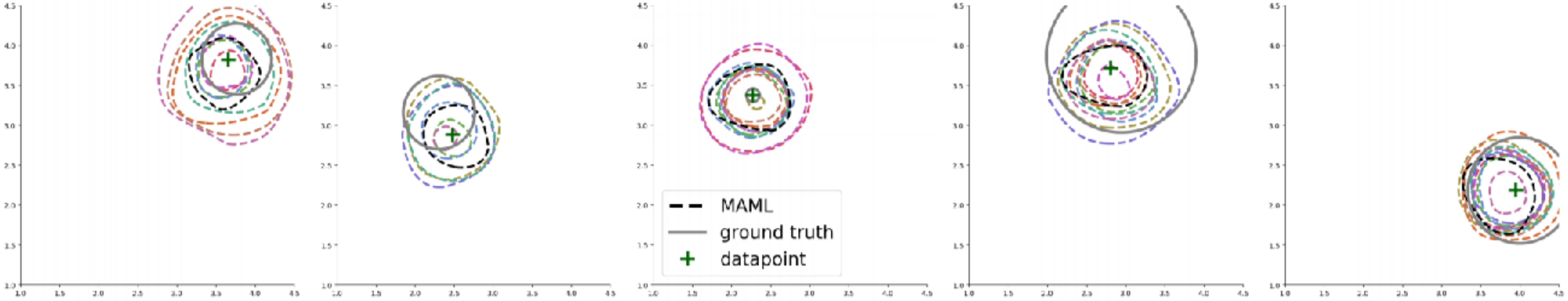
Qualitative Evaluation on Toy Problems with Ambiguity

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

Ambiguous regression:

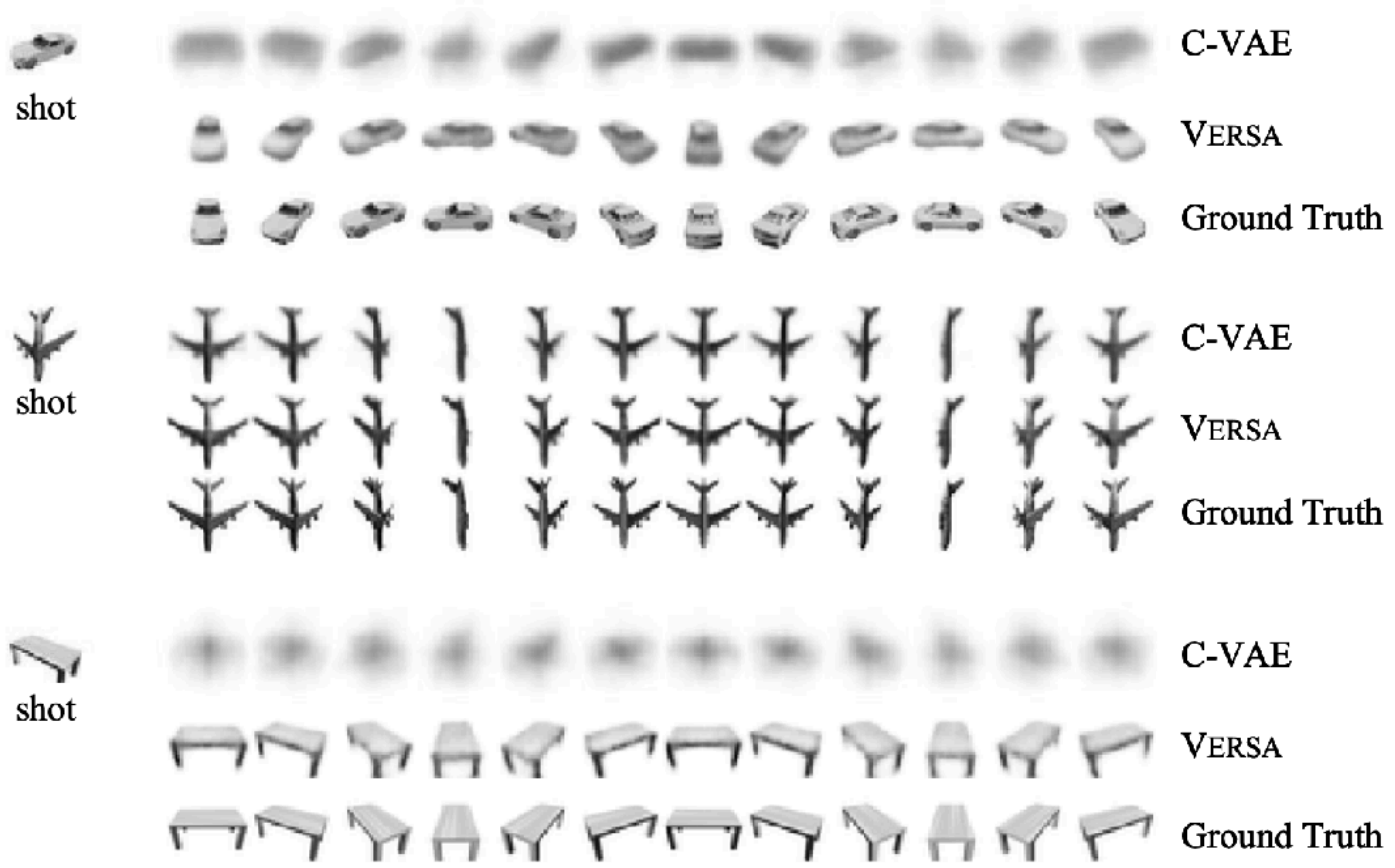


Ambiguous classification:



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.

Accuracy, Mode Coverage, & Likelihood on Ambiguous Tasks

(Finn*, Xu*, Levine, NeurIPS ’18)

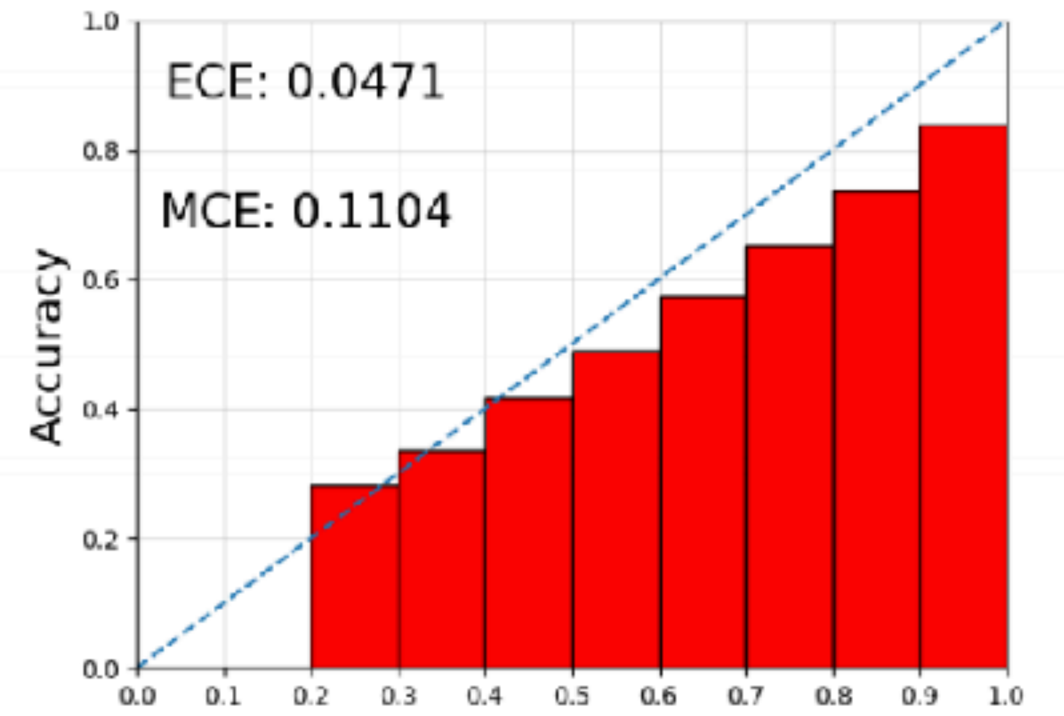
Ambiguous celebA (5-shot)			
	Accuracy	Coverage (max=3)	Average NLL
MAML	**89.00 ± 1.78%**	1.00 ± 0.0	0.73 ± 0.06
MAML + noise	84.3 ± 1.60 %	1.89 ± 0.04	0.68 ± 0.05
PLATIPUS (ours) (KL weight = 0.05)	**88.34 ± 1.06 %**	1.59 ± 0.03	0.67± 0.05
PLATIPUS (ours) (KL weight = 0.15)	**87.8 ± 1.03 %**	**1.94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

Reliability Diagrams & Accuracy

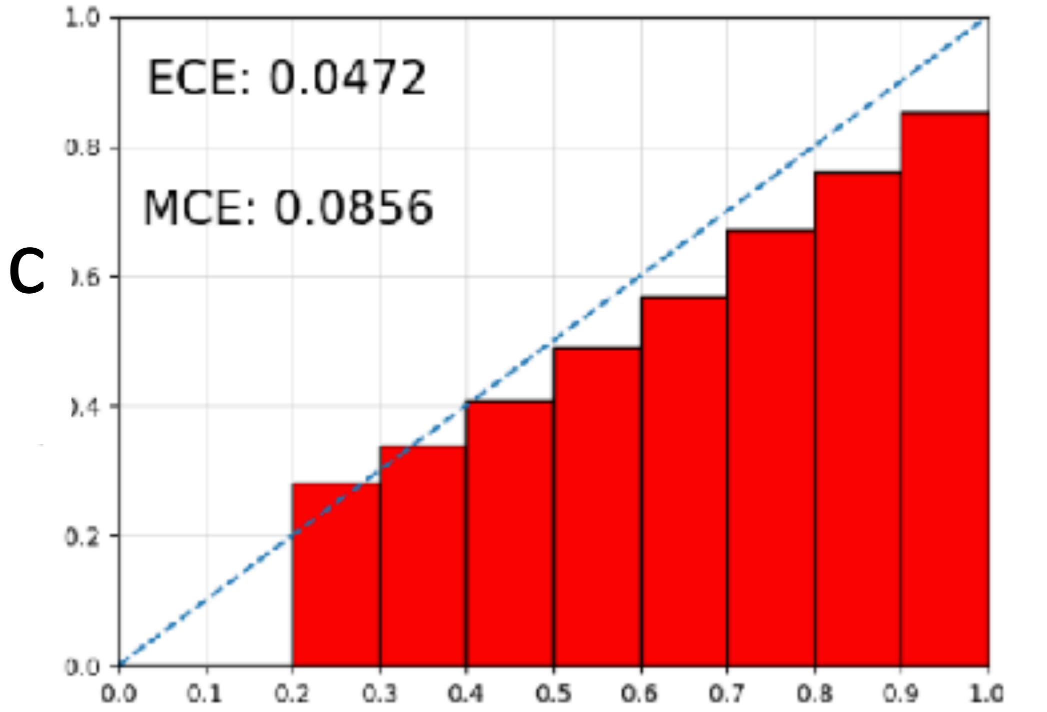
(Ravi & Beatson, ICLR ’19)

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

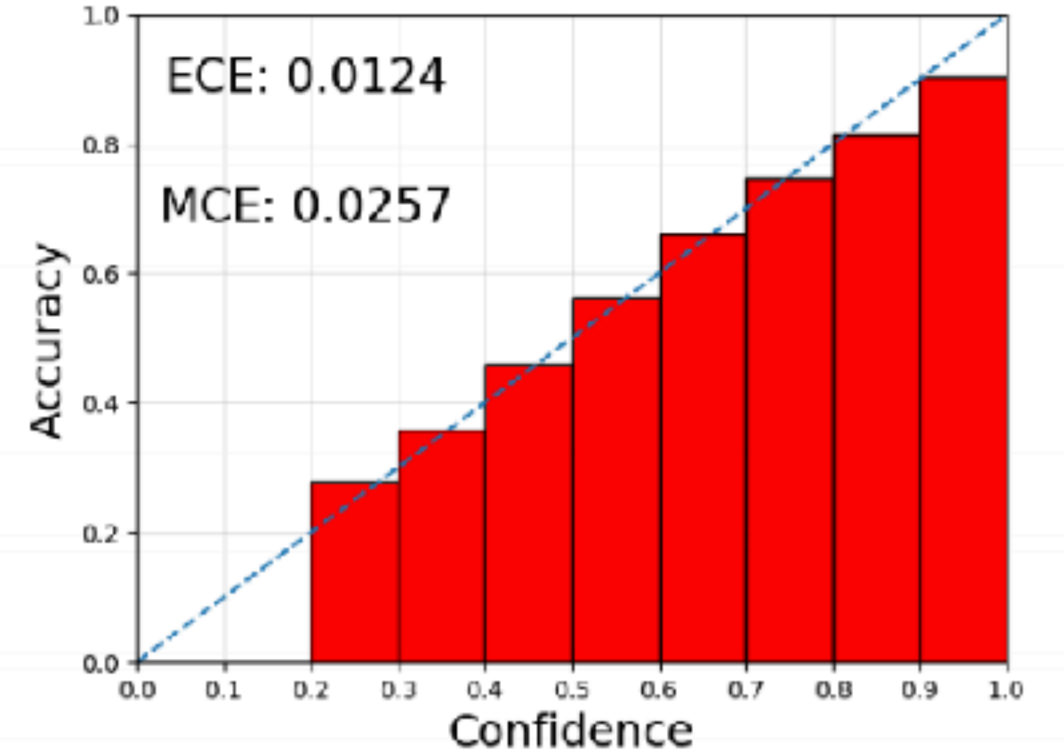
MAML



Probabilistic MAML



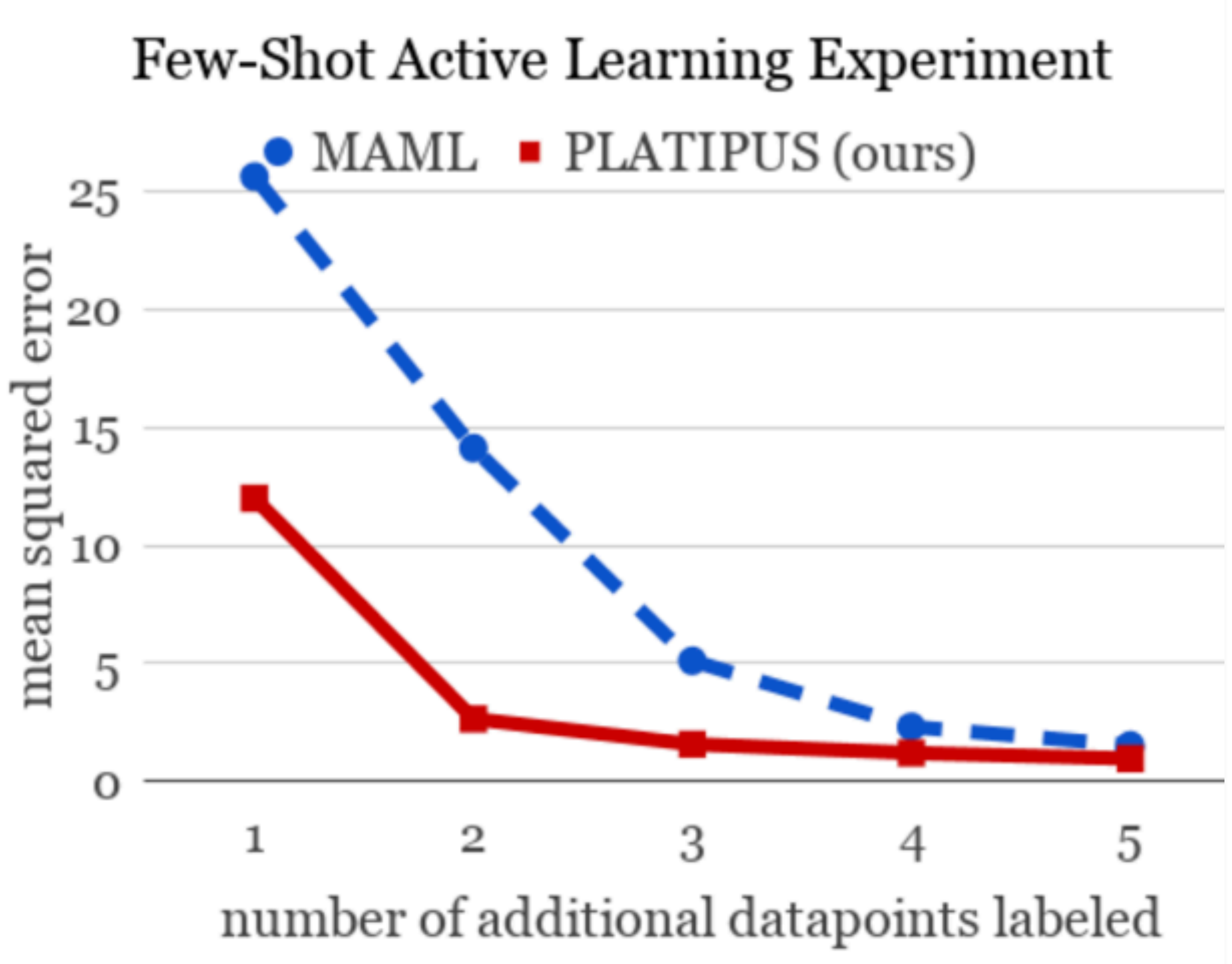
Ravi & Beatson



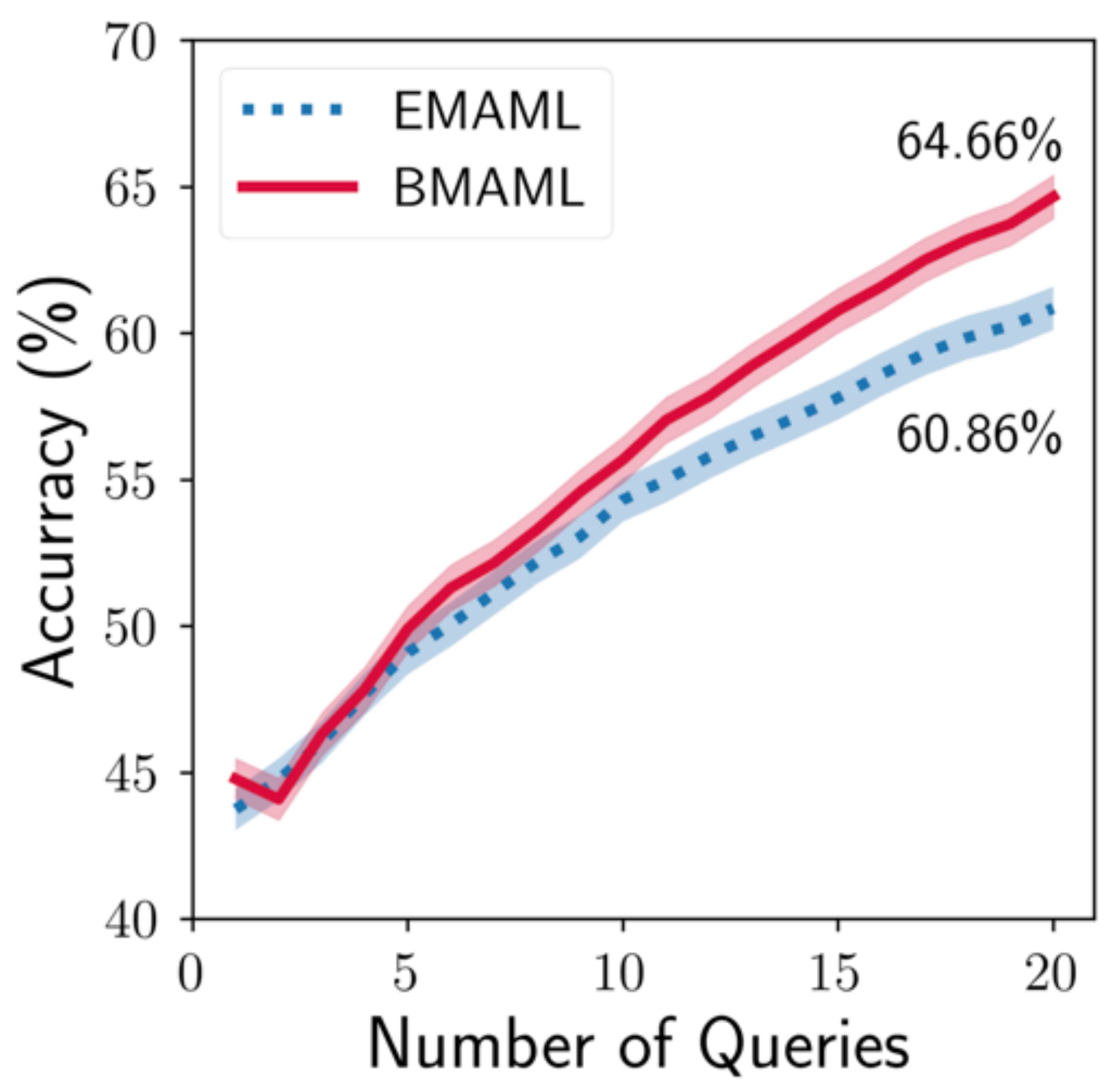
*mini*ImageNet	1-shot, 5-class
MAML (ours)	47.0 $\pm$ 0.59
Prob. MAML (ours)	47.8 $\pm$ 0.61
Our Model	45.0 $\pm$ 0.60

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)

***Algorithmic properties* perspective**

Expressive power

the ability for f to represent a range of learning procedures

Why? scalability, applicability to a range of domains

Consistency

learned learning procedure will solve task with enough data

Why? reduce reliance on meta-training tasks, good OOD task performance

Uncertainty awareness

ability to reason about ambiguity during learning

Why? active learning, calibrated uncertainty, RL principled Bayesian approaches

Next Time

Wednesday:
Meta-learning for unsupervised, semi-supervised, weakly-supervised, active learning

Next Monday:
Start of reinforcement learning!

Reminders

Homework 2 due **next Wednesday.**

Project proposal due in **two weeks**.

Poster presentation: **Tues 12/3** at **1:30 pm**.