Optimization-Based Meta-Learning
and
(finishing from last time)

Non-Parametric Few-Shot Learning

CS 330

Logistics

Homework 1 due, Homework 2 out **this Wednesday**

Fill out **poster presentation preferences**! (Tues 12/3 or Weds 12/4)

Course project details & suggestions posted
Proposal due Monday 10/28

Plan for Today

Optimization-Based Meta-Learning
- Recap & discuss advanced topics

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

Recap from Last Time

pre-trained parameters

Fine-tuning [test-time]

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

training data for new task

MAML

$$\min_\theta \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

Optimizes for an effective initialization for fine-tuning.

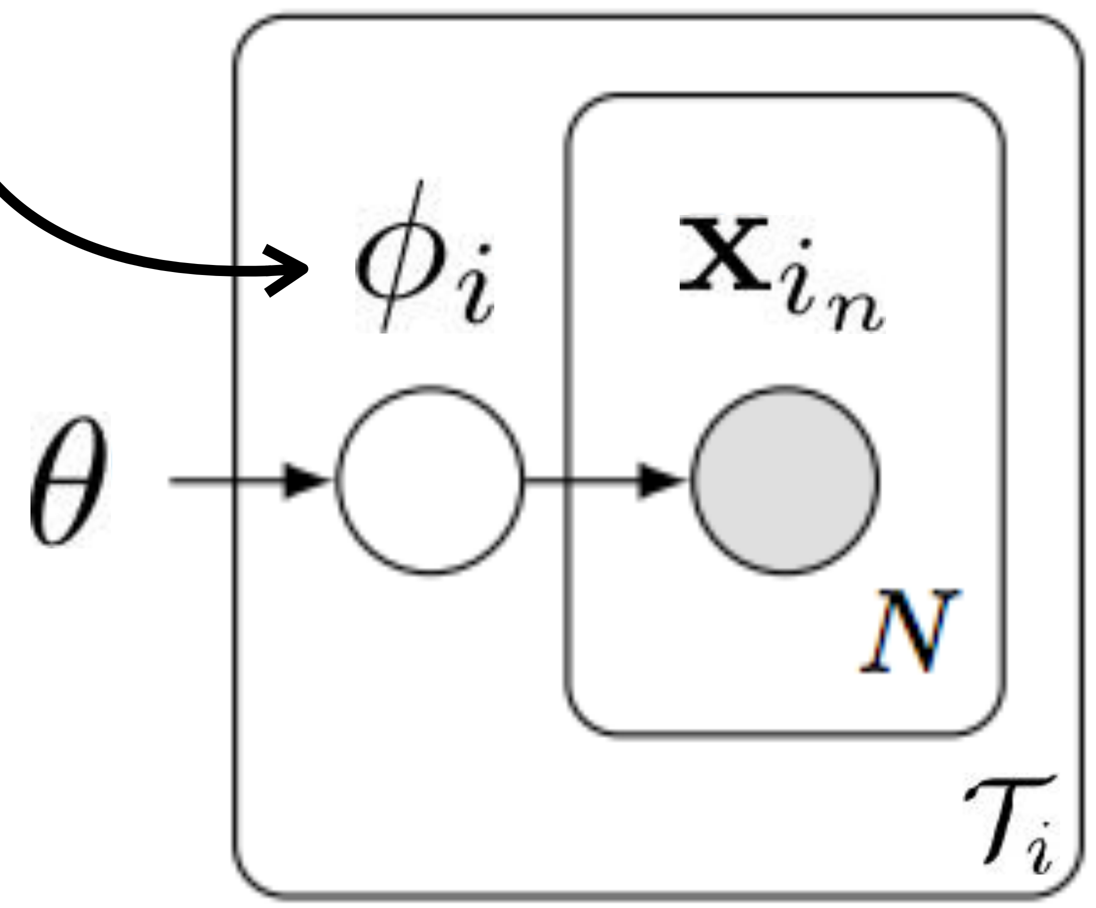
Discussed: performance on extrapolated tasks, expressive power

Probabilistic Interpretation of Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Meta-parameters θ serve as a prior. One form of prior knowledge: **initialization** for **fine-tuning**

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)

MAML approximates hierarchical Bayesian inference. Grant et al. ICLR '18

Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Meta-parameters θ serve as a prior. One form of prior knowledge: **initialization for fine-tuning**

Gradient-descent + early stopping (MAML): implicit Gaussian prior $\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$

Other forms of priors?

Gradient-descent with **explicit Gaussian prior** $\phi \leftarrow \min_{\phi'} \mathcal{L}(\phi', \mathcal{D}^{\text{tr}}) + \frac{\lambda}{2} ||\theta - \phi'||^2$

Rajeswaran et al. implicit MAML '19

Bayesian linear regression *on learned features* Harrison et al. ALPaCA '18

Closed-form *or* **convex optimization** *on learned features*

ridge regression, logistic regression
Bertinetto et al. R2-D2 '19

support vector machine
Lee et al. MetaOptNet '19

Current **SOTA** on few-shot image classification

Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Challenges

How to choose architecture that is effective for inner gradient-step?

Idea: Progressive neural architecture search + MAML

(Kim et al. Auto-Meta)

- finds highly non-standard architecture (deep & narrow)
- different from architectures that work well for standard supervised learning

MiniImagenet, 5-way 5-shot

MAML, basic architecture: **63.11%**

MAML + AutoMeta: **74.65%**

Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Challenges

Bi-level optimization can exhibit instabilities.

Idea: Automatically learn inner vector learning rate, tune outer learning rate
(Li et al. Meta-SGD, Behl et al. AlphaMAML)

Idea: Optimize only a subset of the parameters in the inner loop
(Zhou et al. DEML, Zintgraf et al. CAVIA)

Idea: Decouple inner learning rate, BN statistics per-step (Antoniou et al. MAML++)

Idea: Introduce context variables for increased expressive power.
(Finn et al. bias transformation, Zintgraf et al. CAVIA)

Takeaway: a range of simple tricks that can help optimization significantly

Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Challenges

Backpropagating through many inner gradient steps is compute- & memory-intensive.

Idea: [Crudely] approximate $\frac{d\phi_i}{d\theta}$ as identity

(Finn et al. first-order MAML ‘17, Nichol et al. Reptile ’18)

Takeaway: works for simple few-shot problems, but (anecdotally) not for more complex meta-learning problems.

Can we compute the meta-gradient *without differentiating through the optimization path*?

-> whiteboard

Idea: Derive meta-gradient using the implicit function theorem

(Rajeswaran, Finn, Kakade, Levine. Implicit MAML ’19)

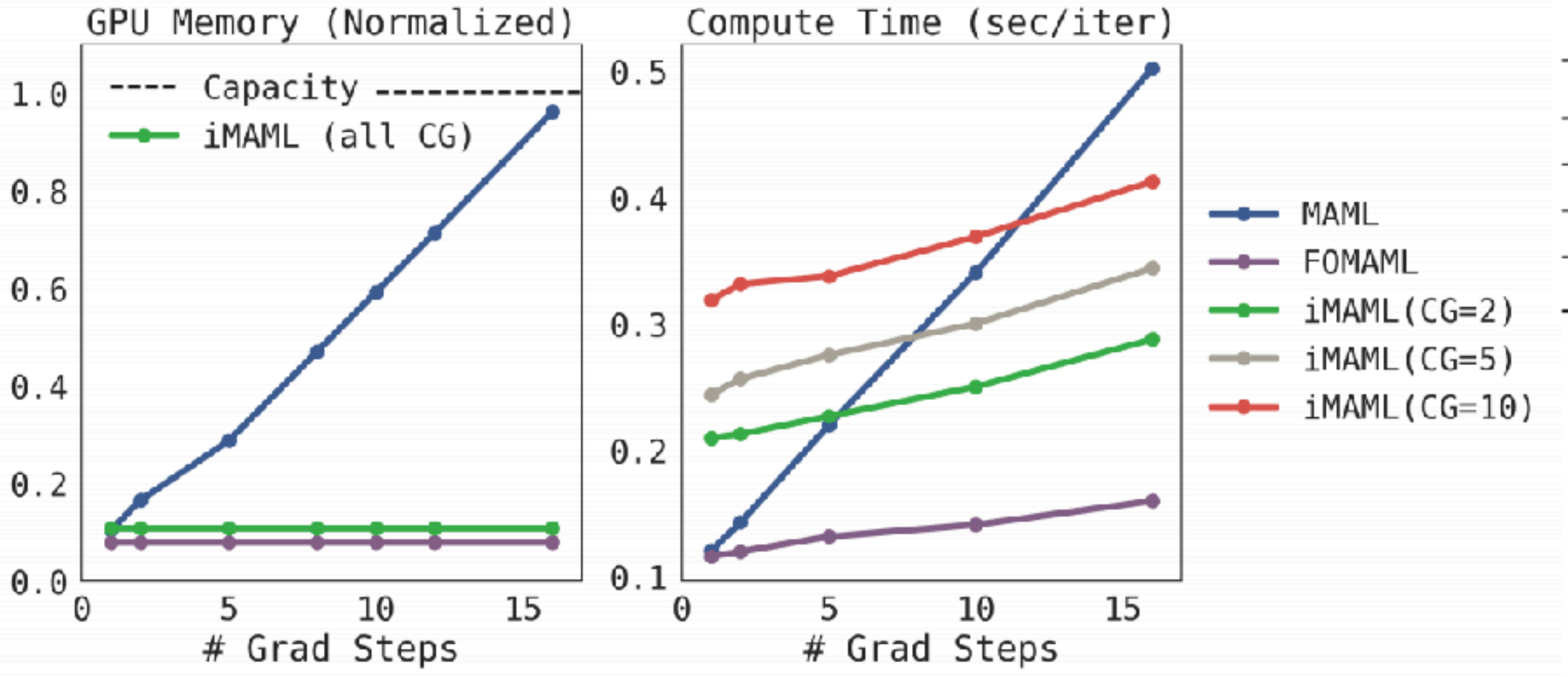
Optimization-Based Inference

Can we compute the meta-gradient *without differentiating through the optimization path*?

Idea: Derive meta-gradient using the implicit function theorem

(Rajeswaran, Finn, Kakade, Levine. Implicit MAML)

Memory and computation trade-offs



Allows for second-order optimizers in inner loop

Algorithm	5-way 1-shot	5-way 5-shot	20-way 1-shot	20-way 5-shot
MAML [15]	$98.7 \pm 0.4\%$	$\mathbf{99.9 \pm 0.1\%}$	$95.8 \pm 0.3\%$	$98.9 \pm 0.2\%$
first-order MAML [15]	$98.3 \pm 0.5\%$	$99.2 \pm 0.2\%$	$89.4 \pm 0.5\%$	$97.9 \pm 0.1\%$
Reptile [43]	$97.68 \pm 0.04\%$	$99.48 \pm 0.06\%$	$89.43 \pm 0.14\%$	$97.12 \pm 0.32\%$
iMAML, GD (ours)	$99.16 \pm 0.35\%$	$99.67 \pm 0.12\%$	$94.46 \pm 0.42\%$	$98.69 \pm 0.1\%$
iMAML, Hessian-Free (ours)	$\mathbf{99.50 \pm 0.26\%}$	$99.74 \pm 0.11\%$	$\mathbf{96.18 \pm 0.36\%}$	$\mathbf{99.14 \pm 0.1\%}$

A very recent development (NeurIPS '19)

(thus, all the typical caveats with recent work)

Optimization-Based Inference

Key idea: Acquire ϕ_i through optimization.

Takeaways: Construct *bi-level optimization* problem.

+ positive inductive bias at the start of meta-learning
+ consistent procedure, tends to extrapolate better
+ maximally expressive with sufficiently deep network
+ model-agnostic (easy to combine with your favorite architecture)

- typically requires second-order optimization
- usually compute and/or memory intensive

Can we embed a learning procedure *without* a second-order optimization?

Plan for Today

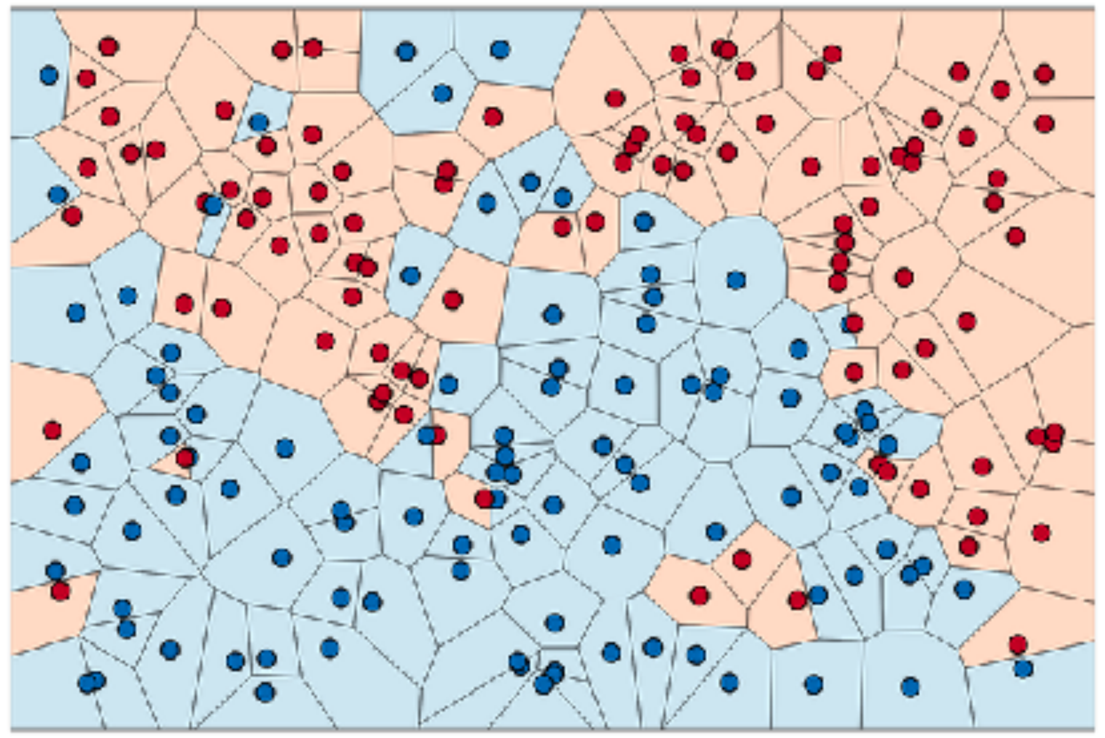
Optimization-Based Meta-Learning
- Recap & discuss advanced topics

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

So far: Learning parametric models.

In low data regimes, **non-parametric** methods are simple, work well.



During **meta-test time**: few-shot learning <-> low data regime

During **meta-training**: still want to be parametric

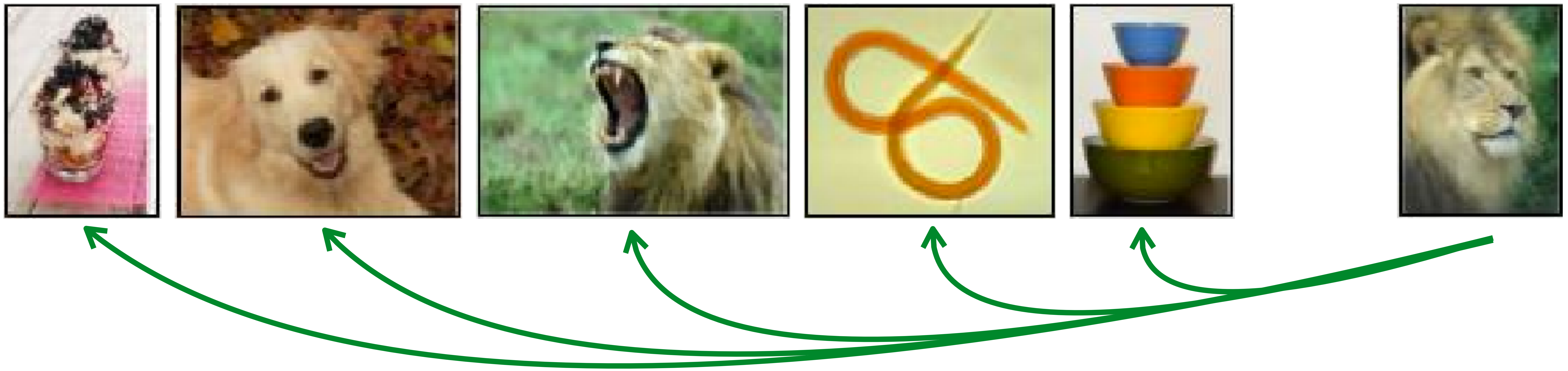
Can we use **parametric meta-learners** that produce effective **non-parametric learners**?

Note: some of these methods precede parametric approaches

Non-parametric methods

Key Idea: Use non-parametric learner.

training data $\mathcal{D}_i^{\mathrm{tr}}$ test datapoint x^{ts}



Compare test image with training images

In what space do you compare? With what distance metric?

pixel space, l_2 distance?

In what space do you compare? With what distance metric?

pixel space, l_2 distance?

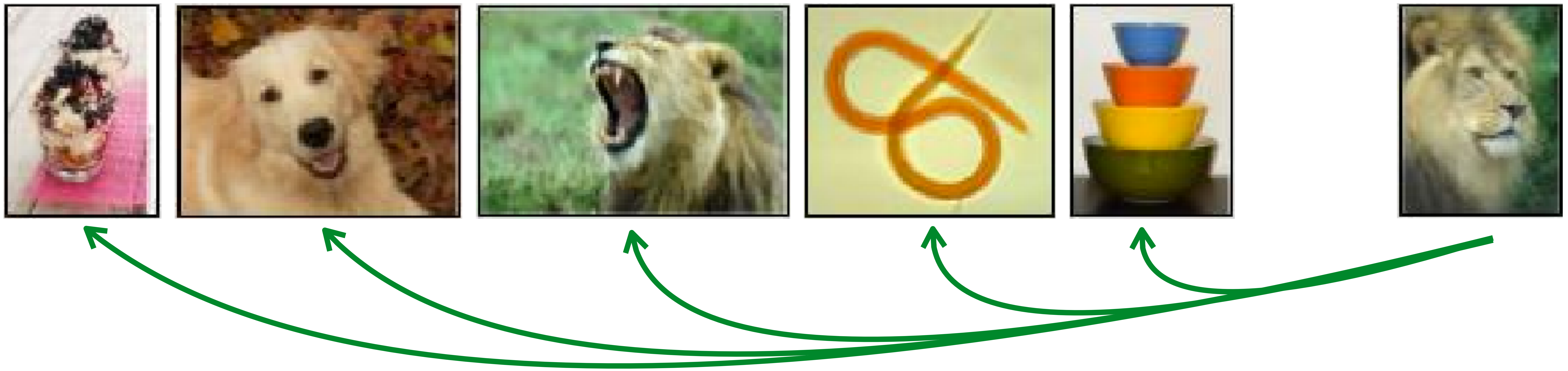
Zhang et al. (arXiv 1801.03924)

Non-parametric methods

Key Idea: Use non-parametric learner.

training data $\mathcal{D}_i^{\mathrm{tr}}$

test datapoint x^{ts}



Compare test image with training images

In what space do you compare? With what distance metric?

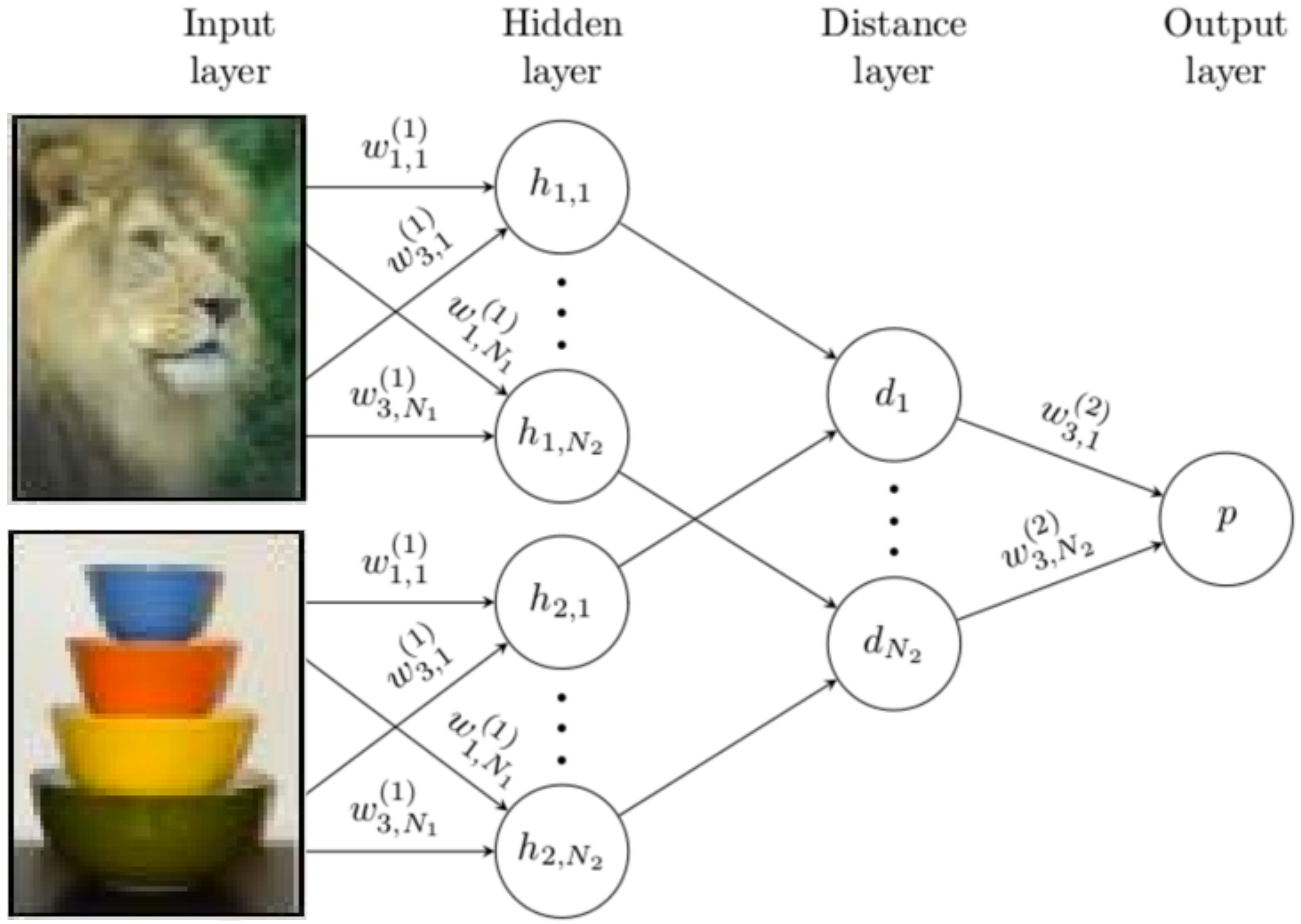
~~pixel space, l_2 distance?~~

Learn to compare using meta-training data!

Non-parametric methods

Key Idea: Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class



label

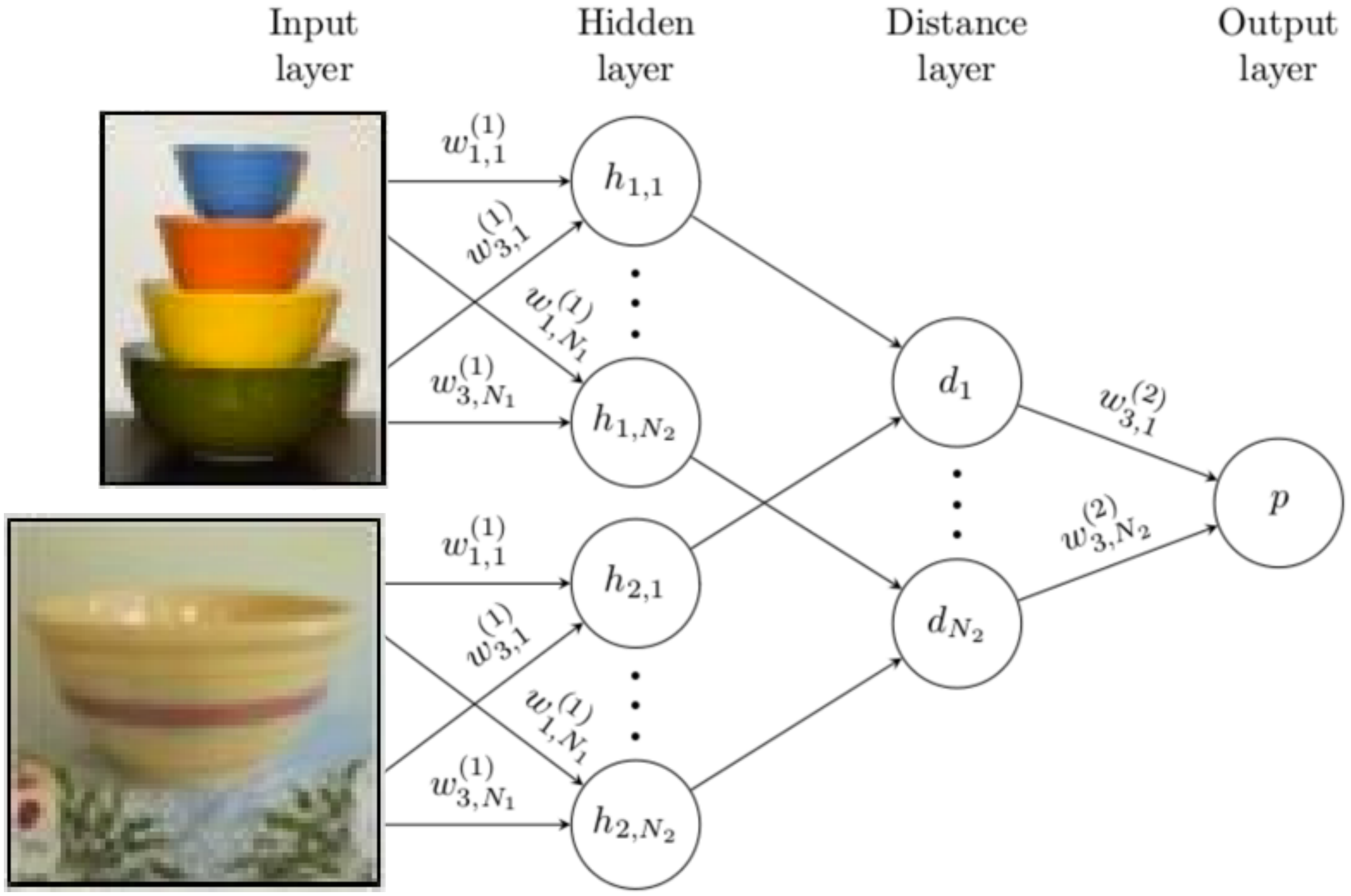
0

Koch et al., ICML '15

Non-parametric methods

Key Idea: Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class



label

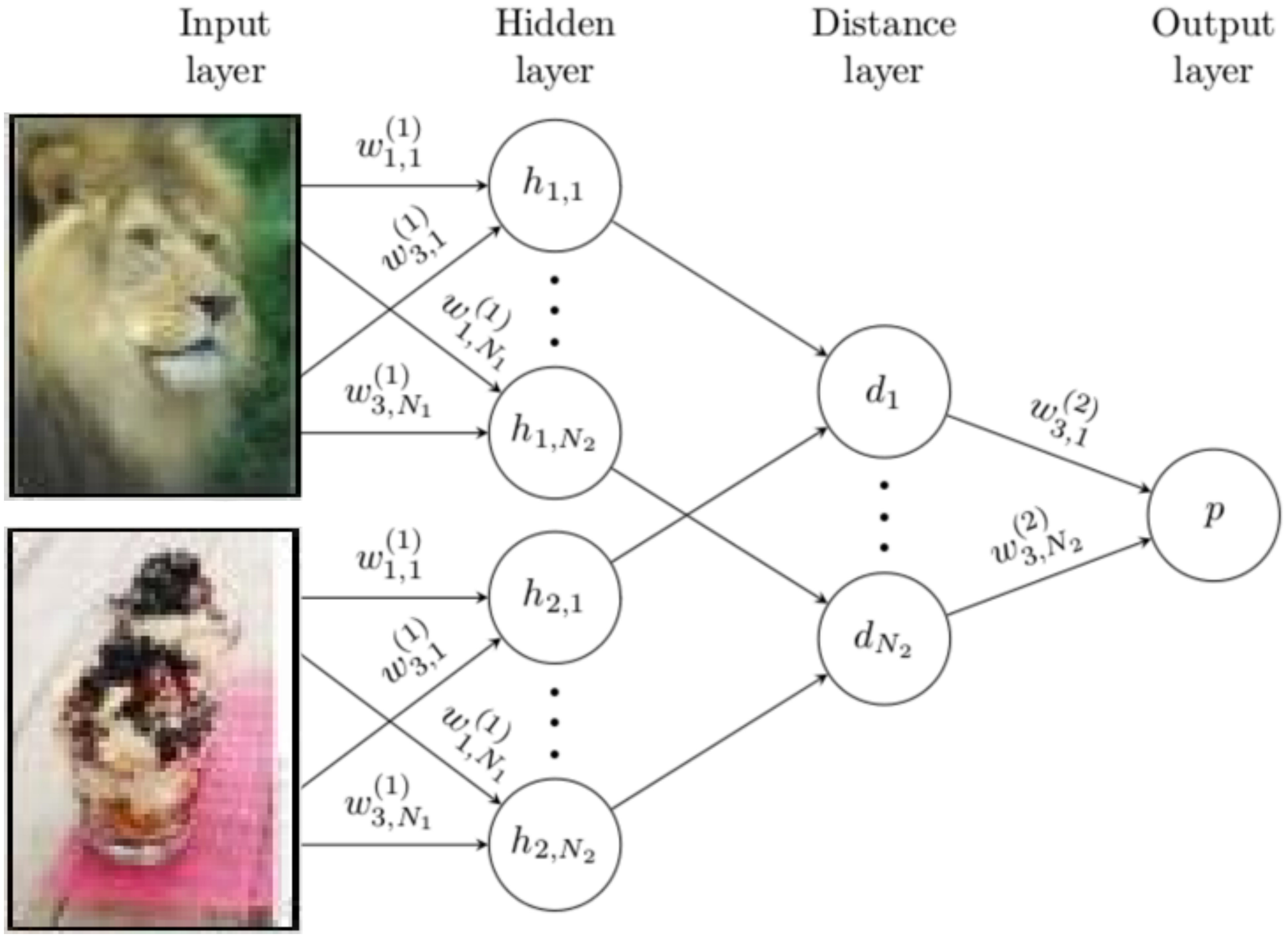
1

Koch et al., ICML '15

Non-parametric methods

Key Idea: Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class



label

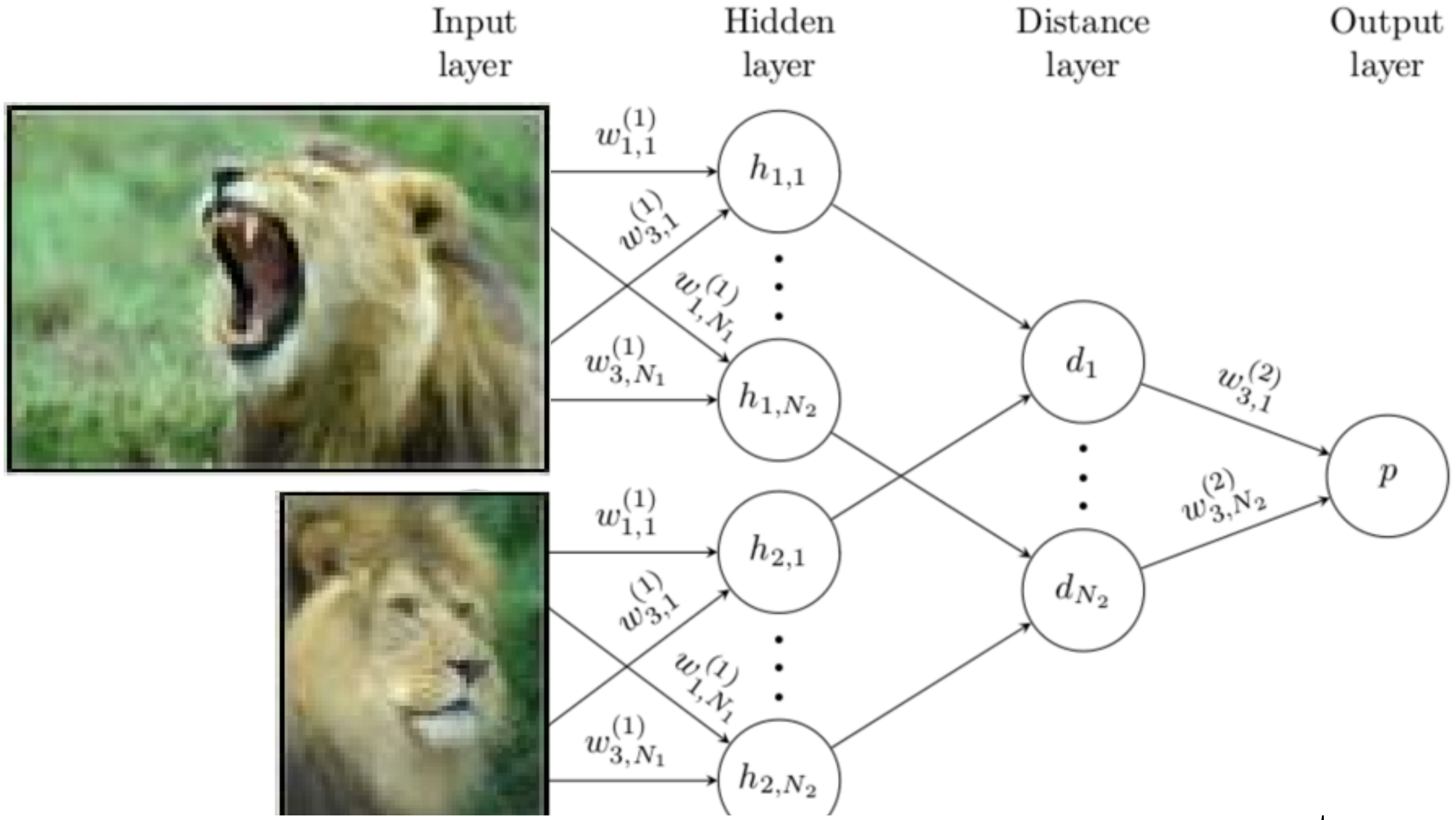
0

Koch et al., ICML '15

Non-parametric methods

Key Idea: Use non-parametric learner.

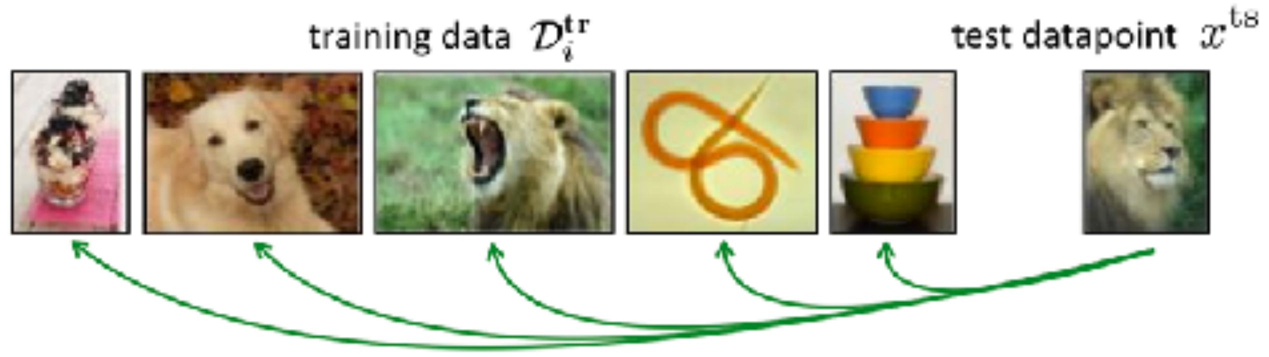
train Siamese network to predict whether or not two images are the same class



label

1

Meta-test time: compare image $\mathbf{x}_{\text{test}}$ to each image in $\mathcal{D}_j^{\text{tr}}$



Meta-training: Binary classification
Meta-test: N-way classification

Can we **match** meta-train & meta-test?

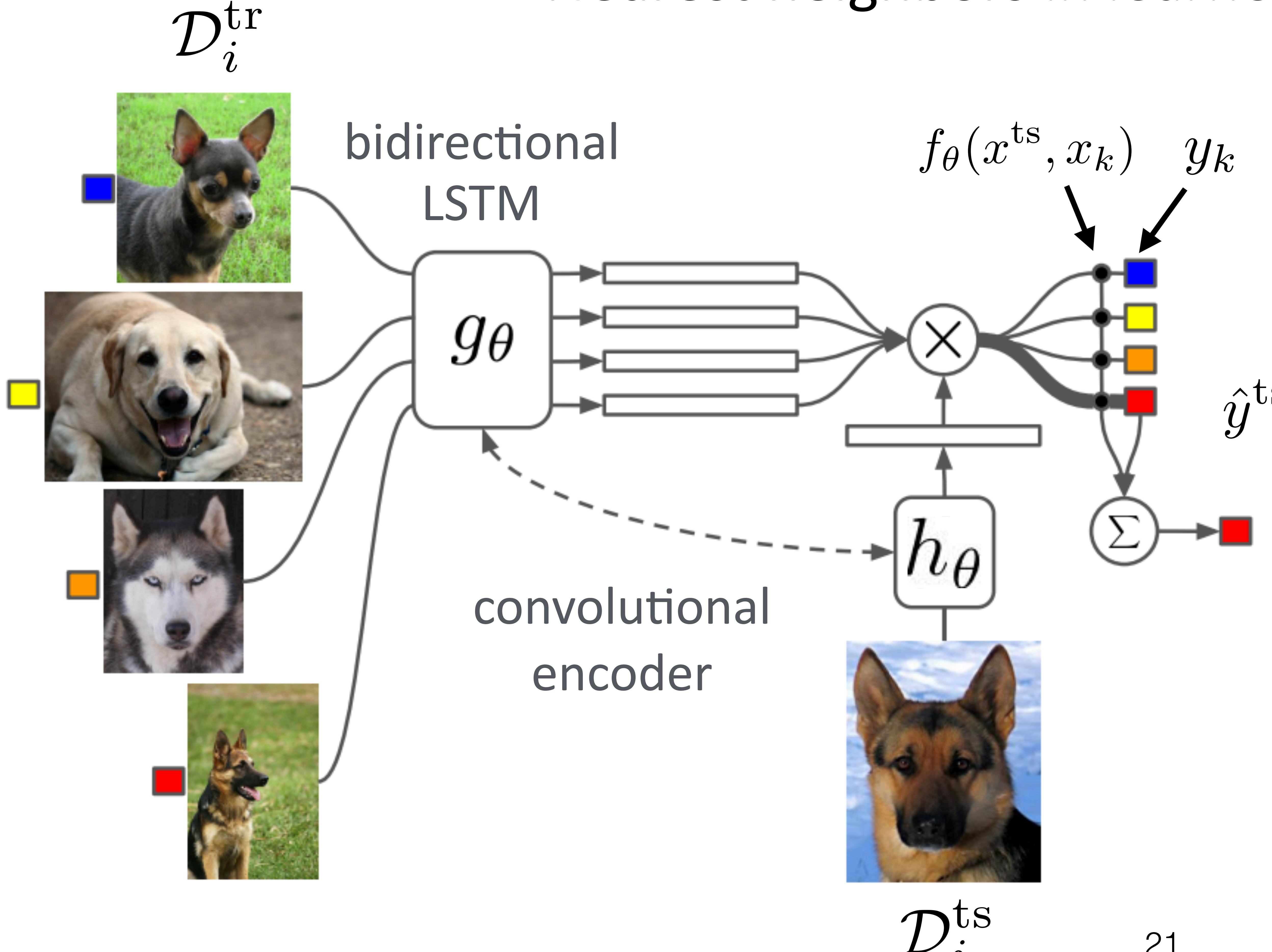
Koch et al., ICML '15

Non-parametric methods

Key Idea: Use non-parametric learner.

Can we **match** meta-train & meta-test?

Nearest neighbors in learned embedding space



$$\hat{y}^{\text{ts}} = \sum_{x_k, y_k \in \mathcal{D}^{\text{tr}}} f_\theta(x^{\text{ts}}, x_k) y_k$$

Trained **end-to-end.**

Meta-train & **meta-test** time **match.**

Vinyals et al. Matching Networks, NeurIPS '16

Non-parametric methods

Key Idea: Use non-parametric learner.

General Algorithm:

~~Amortized approach~~ Non-parametric approach (matching networks)

1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. $\cancel{\text{Compute } \phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})}$ Compute $\hat{y}^{\text{ts}} = \sum_{x_k, y_k \in \mathcal{D}^{\text{tr}}} f_\theta(x^{\text{ts}}, x_k) y_k$
4. $\cancel{\text{Update } \theta \text{ using } \nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})}$ Update θ using $\nabla_\theta \mathcal{L}(\hat{y}^{\text{ts}}, y^{\text{ts}})$

(Parameters ϕ integrated out, hence **non-parametric**)

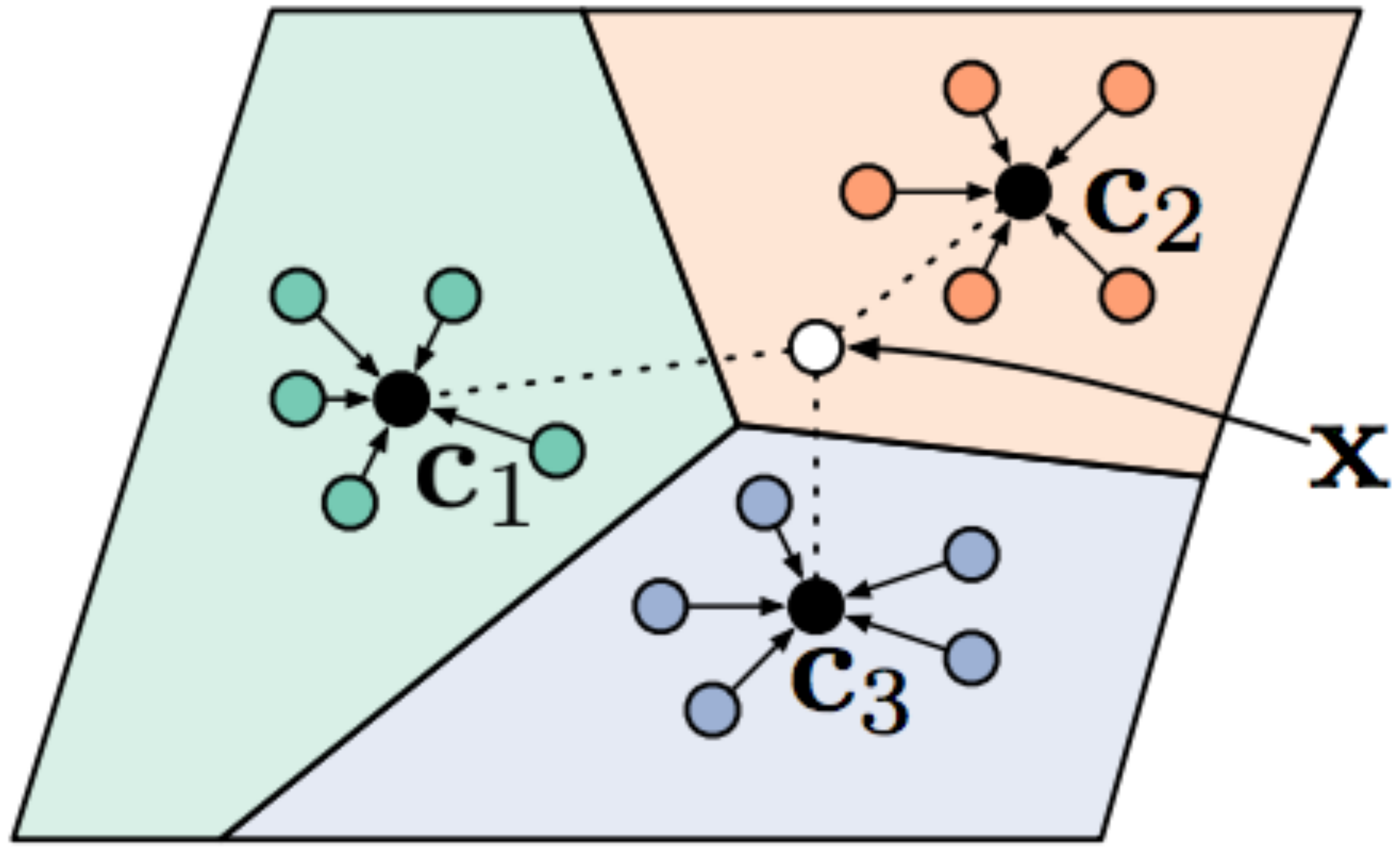
What if **>1 shot**?

Matching networks will perform comparisons **independently**

Can we **aggregate class information** to create a **prototypical embedding**?

Non-parametric methods

Key Idea: Use non-parametric learner.



(a) Few-shot

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

$$p_\theta(y = n | x) = \frac{\exp(-d\left(f_\theta(x), \mathbf{c}_n\right))}{\sum_{n'} \exp(d(f_\theta(x), \mathbf{c}_{n'}))}$$

d: Euclidean, or cosine distance

Snell et al. Prototypical Networks, NeurIPS ‘17

Non-parametric methods

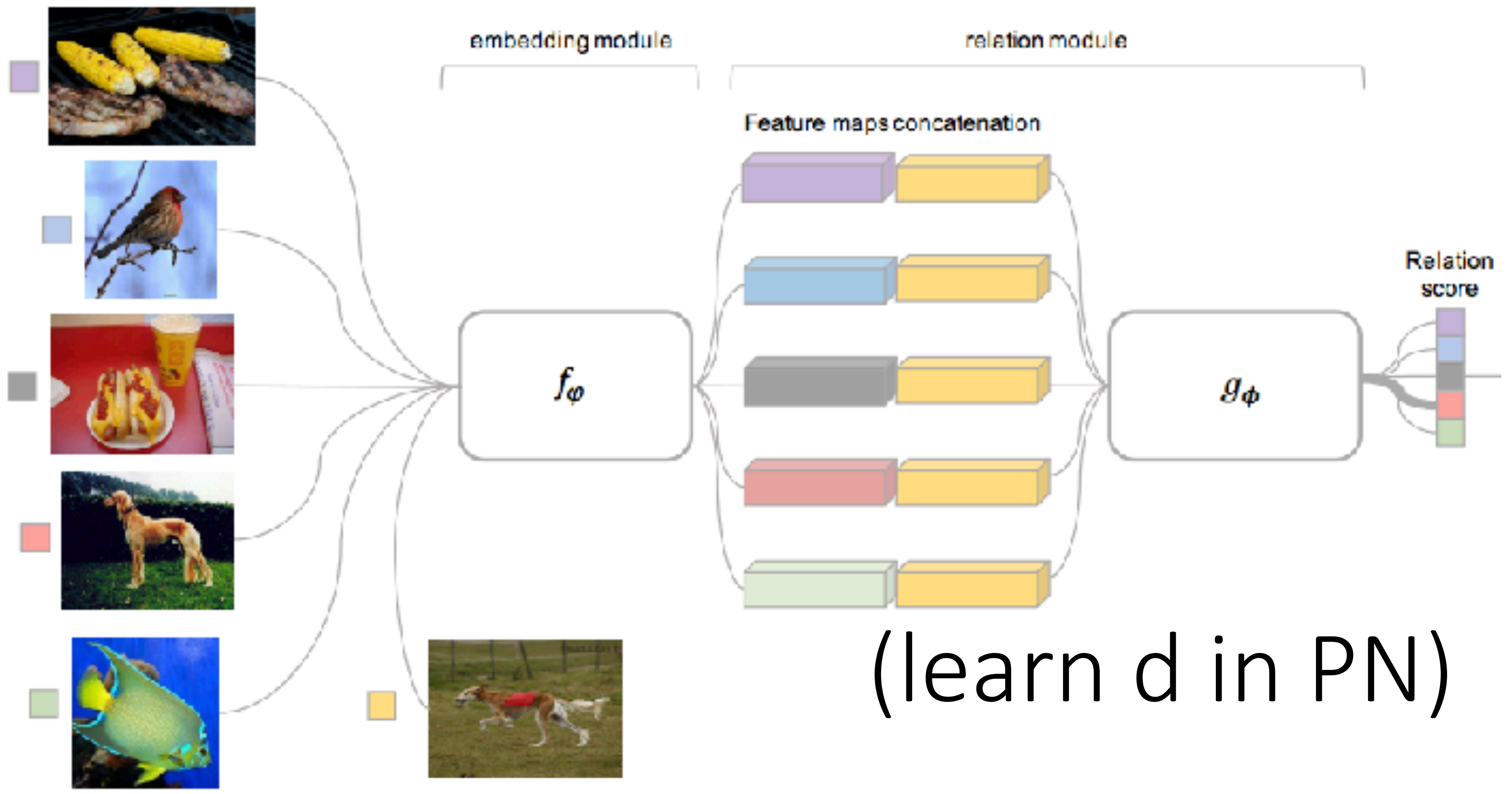
So far: Siamese networks, matching networks, prototypical networks

Embed, then nearest neighbors.

Challenge

What if you need to reason about more complex relationships between datapoints?

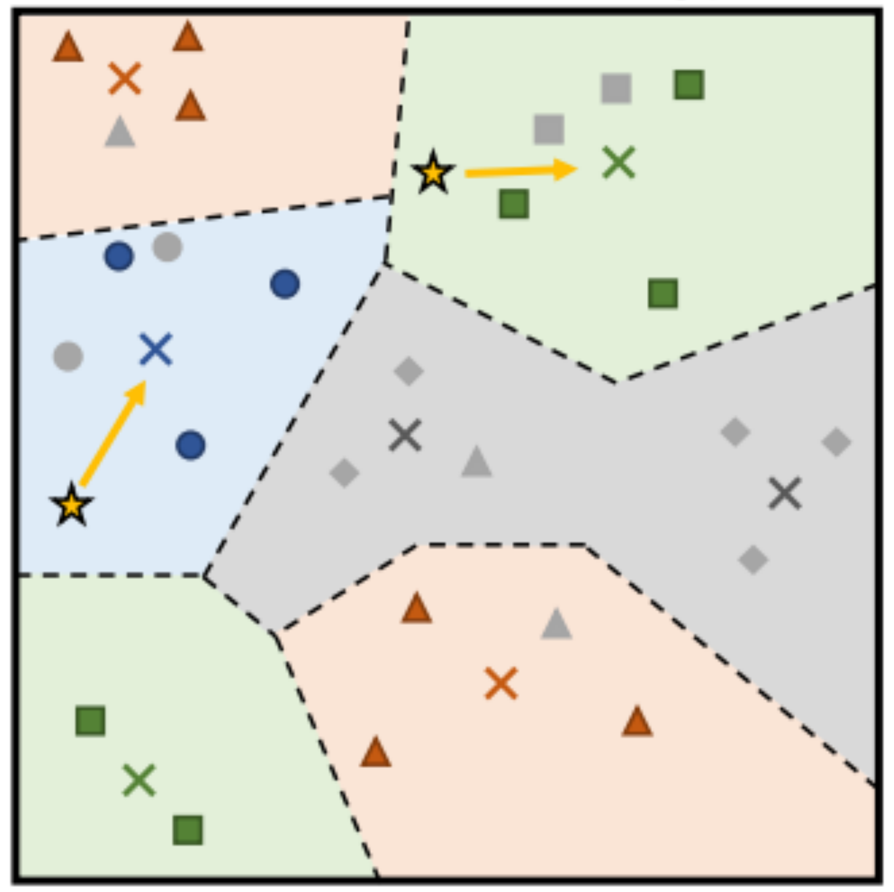
Idea: Learn non-linear relation module on embeddings



(learn d in PN)

Sung et al. Relation Net

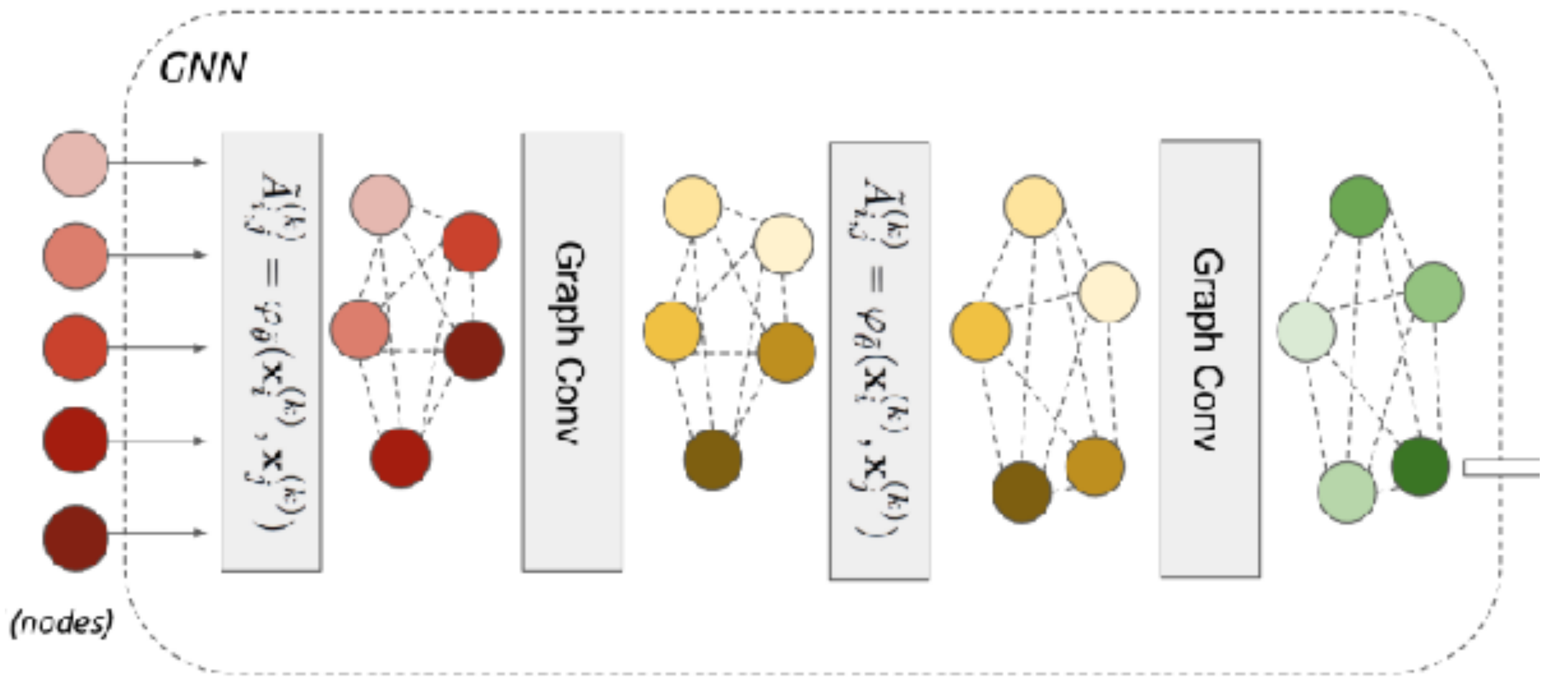
Idea: Learn infinite mixture of prototypes.



adaptive number of clusters

Allen et al. IMP, ICML '19

Idea: Perform message passing on embeddings



Garcia & Bruna, GNN

Plan for Today

Optimization-Based Meta-Learning
- Recap & discuss advanced topics

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

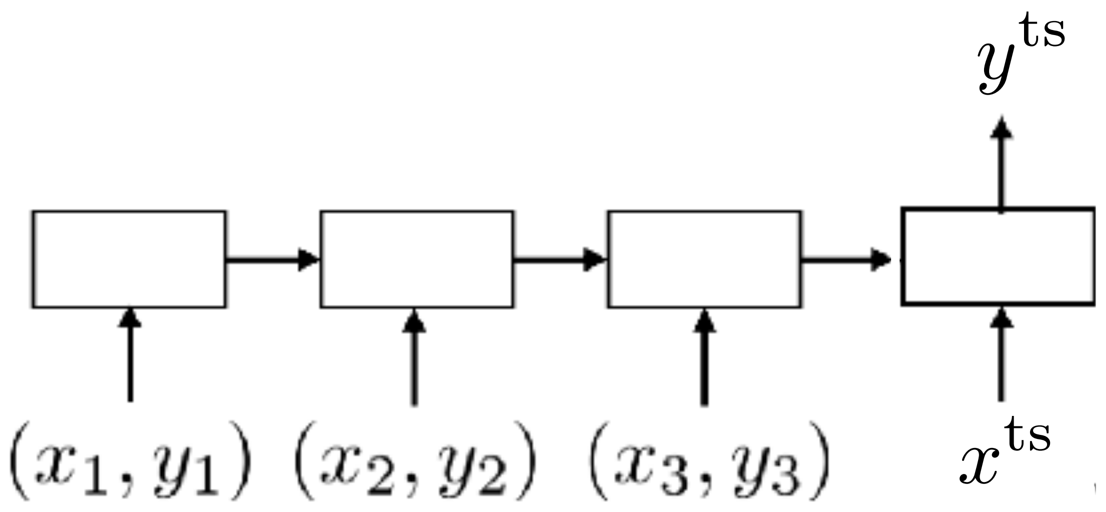
How can we think about how these methods compare?

Black-box vs. Optimization vs. Non-Parametric

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

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

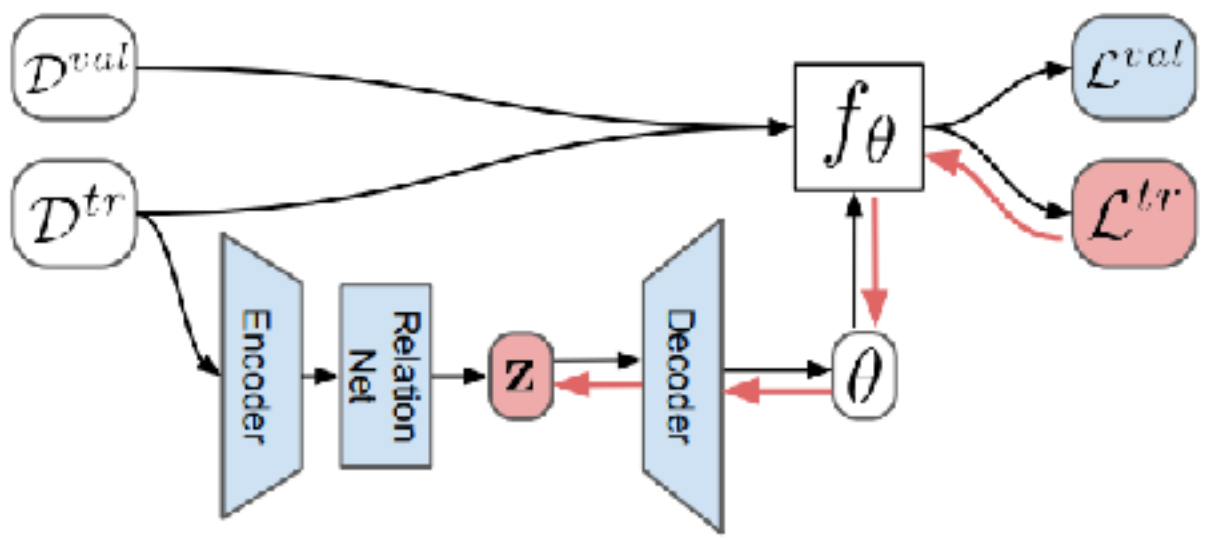
$$\text{where } \mathbf{c}_n = \frac{1}{K} \sum_{(x,y) \in \mathcal{D}_i^{\text{tr}}} \mathbb{1}(y = n) f_\theta(x)$$

Note: (again) Can mix & match components of computation graph

Both condition on data & run gradient descent.

Jiang et al. CAML '19

Gradient descent on relation net embedding.



Rusu et al. LEO '19

MAML, but initialize last layer as ProtoNet during meta-training

Triantafillou et al. Proto-MAML '19

Black-box vs. Optimization vs. Non-Parametric

***Algorithmic properties* perspective**

Expressive power

the ability for f to represent a range of learning procedures

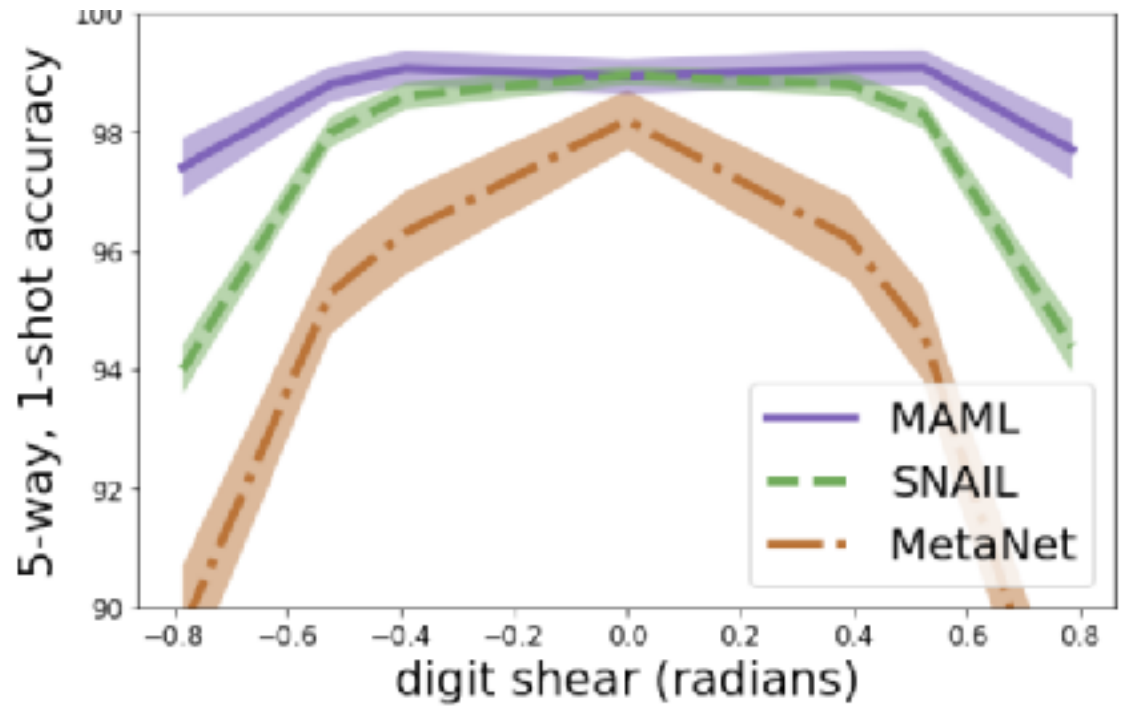
Why? scalability, applicability to a range of domains

Consistency

learned learning procedure will solve task with enough data

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

Recall:



These properties are important for most applications!

Black-box vs. Optimization vs. Non-Parametric

Black-box

+ **complete expressive power**

- **not consistent**

+ easy to combine with **variety of learning problems** (e.g. SL, RL)

- **challenging optimization** (no inductive bias at the initialization)
- often **data-inefficient**

Optimization-based

+ **consistent**, **reduces** to **GD**

~ **expressive** for **very deep models***

+ **positive inductive bias** at the start of meta-learning
+ handles **varying** & **large K** well
+ **model-agnostic**

- **second-order optimization**
- usually **compute** and **memory** intensive

Non-parametric

+ **expressive** for **most architectures**

~ **consistent** under **certain conditions**

+ entirely **feedforward**
+ **computationally fast** & **easy to optimize**

- **harder to generalize** to **varying K**
- hard to scale to **very large K**
- so far, **limited to classification**

Generally, well-tuned versions of each perform **comparably** on existing few-shot benchmarks!

(likely says more about the benchmarks than the methods)

Which method to use depends on your **use-case**.

*for supervised learning settings

Black-box vs. Optimization vs. Non-Parametric

***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

We'll discuss this next time!

Reminders

Homework 1 due, Homework 2 out **this Wednesday**

Fill out **poster presentation preferences**! (Tues 12/3 or Weds 12/4)

Course project details & suggestions posted
Proposal due Monday 10/28