Towards Efficient Learning of
Graphical Models and Neural Networks
with Variational Techniques

Shell Xu Hu

École des Ponts ParisTech – Université Paris-Est
1. Probabilistic machine learning

2. Information theoretical machine learning

3. Empirical Bayes transductive meta-learning with synthetic gradients

4. Variational information distillation for knowledge transfer
Related publications

- Shell X. Hu, Pablo G. Moreno, Andreas Damianou, Neil D. Lawrence. “β-BNN: A Rate-Distortion Perspective on Bayesian Neural Networks.” NeurIPS Workshop on Bayesian Deep Learning, 2018
Probabilistic machine learning
The probabilistic pipeline

Knowledge

Data

Probabilistic model assumptions

Inference & learning

Prediction with uncertainty quantification
Probabilistic graphical models (PGMs)

How do we model a random vector $x = (x_1, \ldots, x_n)$ when $n$ is large?

- Semantic segmentation: $x \in \{1, \ldots, K\}^n$.
- Human pose estimation: $x \in \mathbb{R}^n$.

PGMs are special distributions where **conditional independence (CI)** assumptions are made to enable a factorization according to a graph $G$:

$$p(x) \propto \prod_{a \in \mathcal{A}} \psi_a(x_a),$$

where $\mathcal{A}$ is a set of cliques in $G$. 
Latent variable models (LVMs)

What if we have no idea how to make CI assumptions among $x$?

- A LVM introduces a latent variable $z$ with joint distribution $p(x, z)$, which is the underpinning of *deep generative models* and *Bayesian neural networks*.

- **Inference** about $z$ based on the data is through *posterior*

$$p(z|x) = \frac{p(x, z)}{p(x)} = \frac{p(x|z)p(z)}{p(z)}.$$ 

- Since $p(x) = \int p(x, z)dz$ is intractable, we appeal to *approximate posterior inference*.
Variational inference for LVMs

- VI casts **inference as optimization**.
- Posit a **variational family** of distributions of the form $q(z; \nu)$.

- Fit the **variational posterior** $q(z; \nu)$ to be close to the true posterior $p(z|x)$ in terms of some divergence measure (e.g. Kullback-Leibler).
To compute $\log p(x) = \log \int p(x, z) dz$, the key idea to find an evidence lower bound (ELBO) by Jensen’s inequality:

$$\log \int p(x, z) dz = \log \int q(z) \frac{p(x, z)}{q(z)} dz \geq \int q(z) \log \frac{p(x, z)}{q(z)} dz =: \text{ELBO}$$

$$= -D_{KL}(q(z) \| p(z|x)) + \log p(x).$$

Thus, we have the classical equivalence

$$\max_q \text{ELBO} \iff \min_q D_{KL}(q(z) \| p(z|x)).$$
Case study: Bayesian models

- **Frequentist’s parametric model:** $p(y_{\text{test}}|x_{\text{test}}; w_{\text{train}})$.
- **Bayesian’s non-parametric model:**
  \[
  p(y_{\text{test}}|x_{\text{test}}, D_{\text{train}}) = \int_{\mathcal{W}} p(y_{\text{test}}|x_{\text{test}}, w)p(w|D_{\text{train}})dw
  \]
- **Compute the posterior via Bayes rule?**
  \[
  p(w|D_{\text{train}}) = \frac{p(D_{\text{train}}|w)p(w)}{p(D_{\text{train}})}
  \]
  given the likelihood $p(D_{\text{train}}|w)$ and the prior $p(w)$. 

Case study: Bayesian models

- In general, unless a conjugate prior is considered for the likelihood, the posterior cannot be computed in closed form.
- Alternatively, we do variational inference:

\[
q_{\text{train}} = \arg \min_{q \in Q} D_{KL}(q(w) \parallel p(w | D_{\text{train}}))
\]

and make prediction through

\[
q(y_{\text{test}} | x_{\text{test}}, D_{\text{train}}) = \int_{\mathcal{W}} p(y_{\text{test}} | x_{\text{test}}, w) \; q_{\text{train}}(w) \, dw
\]
Information theoretical machine learning
Mutual information is used to measure statistical dependence

\[ I(X; Y) := \mathbb{E}_{x,y \sim p(x,y)} \log \frac{p(x, y)}{p(x)p(y)} \]

\[ = H(X, Y) - H(X|Y) - H(Y|X) \]

\[ = H(X) - H(X|Y) \]

\[ H(X) = I(X; X) = \text{expected amount of information in } X \]
Mutual information: another variational tool

If we know the distribution of $X$ and the joint distribution with decomposition $p(x, y) = p(x)q(y|x)$, then we can use mutual information to adjust $q(y)$ by either minimizing or maximizing

$$I(X; Y) \equiv I_q(X, Y) := \mathbb{E}_{x, y \sim p(x, y)} \log \frac{q(y|x)}{q(y)}$$

$$= \mathbb{E}_{p(x)} D_{\text{KL}}(q(y|x)\|q(y)) .$$

Note that the mutual information is a functional of $p$ and $q$. 
Variational characterization of mutual information

Computational issue: $I(X; Y)$ is intractable!

Solution: using variational techniques to derive bounds:

Lemma [Cover and Thomas, 2012, Theorem 10.8.1]

\[
I(X; Y) = \mathbb{E}_{x, y \sim p(x, y)} \log \frac{p(x|y)}{p(x)} = \max_{\phi(x|y)} \mathbb{E}_{x, y \sim p(x, y)} \log \frac{\phi(x|y)}{p(x)}
\]

\[
I(X; Y) = \mathbb{E}_{x, y \sim p(x, y)} \log \frac{q(y|x)}{q(y)} = \min_{m(y)} \mathbb{E}_{x, y \sim p(x, y)} \log \frac{q(y|x)}{m(y)}.
\]
Rate-distortion (RD) tradeoff and information bottleneck

For a **lossy compression** of $X \rightarrow \hat{X}$, when $p(x)$ is given:

\[
\begin{align*}
\text{min Rate} & \quad \min_{q(\hat{x}|x)} \quad I_q(X; \hat{X}) \\
\text{s.t. Distortion} \leq \text{const} & \quad \text{s.t.} \quad \sum_{x, \hat{x}} p(x) q(\hat{x}|x) \quad d(x, \hat{x}) \leq \text{const}.
\end{align*}
\]

The **information bottleneck (IB)** [Tishby et al., 2000] is an extension for supervised learning where the distortion is defined in terms of the relevance wrt the label $Y$:

\[
d(x, \hat{x}) = D_{KL}(p(y|x) \parallel p(y|\hat{x})).
\]

A more common form reads as (assuming $p(y|x)$ is fixed)

\[
\min_{q(\hat{x}|x)} I_q(X; \hat{X}) - \beta I_q(Y; \hat{X}).
\]
Rate-distortion based Bayesian inference

For a dataset $S$, consider a latent variable model defined by

Generative process: $P(S, w) = p(S \mid w)p(w)$.

The variational posterior $q(w \mid S)$ induces

Inference process: $q(S, w) = q(w \mid S)q^*(S)$.

The Bayesian version of the information bottleneck (BIB) [Achille and Soatto, 2017] can be derived from the RD tradeoff [Hu et al., 2018]:

$$\min_{q(w \mid S)} I_q(w; S) + \beta H_{q,p}(S \mid w)$$

where $H_{q,p}(S \mid w) := \mathbb{E}_{p^*}(S)\mathbb{E}_{q(w \mid S)}d(w, S)$

and $d(w, S) := -\log p(S \mid w)$.

This is an alternative objective for variational inference.
Empirical Bayes transductive meta-learning with synthetic gradients
Definition (meta-learning)

The problem is to solve rapidly a new task after learning several other similar tasks, where the dataset is a two-level hierarchy – dataset of datasets, one for each task. Meta-learning is sometimes called learning to learn [Schmidhuber, 1987, Thrun and Pratt, 1998].

Applications:

- Learning to do gradient descent [Andrychowicz et al., 2016].
- Learning to classify unseen categories [Vinyals et al., 2016].
- Learning to generalize across domains [Li et al., 2017].
An example: few-shot classification

Few-shot learning [Vinyals et al., 2016]:

Support set
\[ d^l_t := \{(x^l_t, y^l_t)\}_{i=1}^n \]

Query set
\[ x_t := \{x_t,i\}_{i=1}^n \quad y_t = \{y_t,i\}_{i=1}^n \]

<table>
<thead>
<tr>
<th>Labeled data</th>
<th>Unlabeled data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>✓</td>
</tr>
<tr>
<td>Testing</td>
<td>✓</td>
</tr>
</tbody>
</table>

\(N\)-way-\(K\)-shot setup:

Training task 1
Support set
\(K=2\)
\(N=3\)
Query set

Training task 2
Support set

Test task 1
Support set
Consider $N$ training tasks with associated data $\mathcal{D} := \{d_t := (x_t, y_t)\}_{t=1}^{N}$:

$$
\text{HB} \rightarrow \text{EB} : \quad p_f(\mathcal{D}) \rightarrow p_{\psi, f}(\mathcal{D}) = \int_{\psi} \left[ \prod_{t=1}^{N} \int_{w_t} p_f(d_t | w_t) p(w_t | \psi) \right] p(\psi),
$$

$$
\log p_f(d_t | w_t) \\
= \sum_{i=1}^{n} \log p_f(y_{t,i} | x_{t,i}, w_t) + \log p(x_{t,i} | w_t) \\
= - \sum_{i=1}^{n} \ell_t(\hat{y}_{t,i}(f(x_{t,i}), w_t), y_{t,i}) + \text{const}
$$
We derive an ELBO on the log-likelihood by introducing a variational distribution \( q_{\theta_t}(w_t) \) for each task with parameter \( \theta_t \):

\[
\log p_{\psi,f}(D) \geq \sum_{t=1}^{N} \left[ \mathbb{E}_{w_t \sim q_{\theta_t}} \left[ \log p_f(d_t|w_t) \right] - D_{KL}(q_{\theta_t}(w_t)\|p_{\psi}(w_t)) \right].
\]

Maximizing the ELBO with respect to \( \theta_1, \ldots, \theta_N \) and \( \psi \) is equivalent to

\[
\min_{\theta_1, \ldots, \theta_N} \sum_{t=1}^{N} D_{KL}(q_{\theta_t}(w_t) \| p_{\psi,f}(w_t|d_t))
\]
Amortized inference [Kingma and Welling, 2013] with transduction

Exact VI : \[
\min_{\theta_1, \ldots, \theta_N} \sum_{t=1}^{N} D_{KL}(q_{\theta_t}(w_t) \parallel p_{\psi,f}(w_t|d_t))
\]

For scalable inference, we introduce a neural network \( \phi \) to output \( \theta_t \). There are two choices to do the amortization:

Inductive AVI : \[
\min_{\phi} \sum_{t=1}^{N} D_{KL}(q_{\phi(d_t')}(w_t) \parallel p_{\psi,f}(w_t|d_t))
\]

Transductive AVI : \[
\min_{\phi} \sum_{t=1}^{N} D_{KL}(q_{\phi(d_t',x_t)}(w_t) \parallel p_{\psi,f}(w_t|d_t))
\]
Why transduction?

Motivation: to make use of the unlabeled data (i.e., $x_t$).

(b) MAML Finn et al. [2017]

(c) Our method

- MAML is an inductive method – only use the labeled data $d_t^l$ to construct a Dirac delta variational posterior;
- We construct a better variational posterior as a function of both labeled data $d_t^l$ and unlabeled data $x_t$. 
How do we implement the amortization network $\phi(x_t, d_t^l)$?

The best is through the exact inference (only doable in training)

$$
\phi(d_t^l, x_t) = \arg\min_{\theta_t} D_{KL}(q_{\theta_t}(w_t) \parallel p_{\psi,f}(w_t|d_t))
$$

However, we don’t have access to $y_t$ in testing tasks. Instead, we unroll

$$
\theta_{t+1}^k = \theta_t^k - \eta \nabla_{\theta_t} D_{KL}(q_{\theta_t^k}(w) \parallel p_{\psi,f}(w|d_t)).
$$

up to the $K$-th step by parameterizing

- the **initialization** $\theta_0^t$;
- the **gradient** $\nabla_{\theta_t} D_{KL}(q_{\theta_t^k}(w) \parallel p_{\psi,f}(w|d_t))$. 


Unrolling exact inference with synthetic gradient

Key observation: $y_t$ only appears in $\partial \ell_t$ term.

$$\nabla_{\theta_t} D_{KL}(q_{\theta_t} \parallel p_{\psi,f}) = \mathbb{E}_{\epsilon} \left[ \sum_{i=1}^{n} \frac{\partial \ell_t(\hat{y}_{t,i}, y_{t,i})}{\partial \hat{y}_{t,i}} \frac{\partial \hat{y}_{t,i}}{\partial w_t} \frac{\partial w_t(\theta_t, \epsilon)}{\partial \theta_t} \right]$$

$$+ \nabla_{\theta_t} D_{KL}(q_{\theta_t} \parallel p_{\psi}) .$$

By replacing $\frac{\partial \ell_t(\hat{y}_{t,i}, y_{t,i})}{\partial \hat{y}_{t,i}} \approx \xi(\hat{y}_{t,i})$, we can perform **synthetic gradient descent** without using $y_t$:

$$\theta_{t}^{k+1} = \theta_{t}^{k} - \eta \left[ \mathbb{E}_{\epsilon} \left[ \sum_{i=1}^{n} \xi(\hat{y}_{t,i}) \frac{\partial \hat{y}_{t,i}}{\partial w_t} \frac{\partial w_t(\theta_{t}^{k}, \epsilon)}{\partial \theta_t} \right] + \nabla_{\theta_t} D_{KL}(q_{\theta_t}^{k} \parallel p_{\psi}) \right].$$

The idea of synthetic gradient was originally proposed by Jaderberg et al. [2017] for asynchronous forward and backward passes.
Computation graph of our method

- **Classifier forward**: $\theta_t^K \equiv \phi(d_t^l, x_t)$
  - $\nabla_\theta D_{KL}$
  - $\hat{y}_t$

- **Classifier backward**: $\xi(\hat{y}_t) \approx \frac{\partial \ell_t}{\partial \hat{y}_t}$

- **SGD**: $\theta_0(d_t^l)$, $\theta_1^t$

- **Synthetic gradient module**: $f(x_t). detach()$

- **Init**: $\lambda$

- **f**: $x_t \rightarrow f(x_t)$
  - $y_t^l$

- **SGD**: $\theta_0(d_t^l) \rightarrow \theta_1^t \rightarrow \ldots \rightarrow \theta_t^l$

- **KL**: $D_{KL}(q_{\theta_t^K} \parallel p_\psi)$

- **Loss**: $\ell_t$
Variational EM algorithm

1: while not converged do
2: Sample a task $t$ and its data: $d_t, d'_t$.
3: Compute the initialization $\theta^0_t = \lambda(d'_t)$.
4: %======== E-step =======
5: for $k = 1, \ldots, K$ do

$$\theta_{t}^{k+1} = \theta_{t}^{k} - \eta \text{ synthetic gradient.}$$

6: %======== M-step =======
7: Update $\phi \leftarrow \phi - \eta \nabla_{\phi} D_{KL}(q_{\phi(x_t,d'_t)} \| p_f \cdot p_\psi)$.
8: Update $\psi \leftarrow \psi - \eta \nabla_{\psi} D_{KL}(q_{\theta^K_t(\psi)} \| p_\psi)$.
9: Optionally, update $f \leftarrow f + \eta \nabla_{f} \log p_f(d_t|w_t)$.
Abstract form of empirical Bayes

Note that

$$\log p_{\psi, f}(w_1, \ldots, w_N, D) = \sum_{t=1}^{N} \log p_f(d_t|w_t) + \log p_{\psi}(w_t)$$

is equal to the log-density of $N$ iid samples drawn from

$$p(w, d, t) \equiv p_{\psi, f}(w, d, t) = p_f(d|w, t)p_{\psi}(w)q(t)$$

if $q(t)$ is uniform. Correspondingly, there is another decomposition

$$q(w, d, t) \equiv q_{\phi}(w, d, t) = q_{\phi}(w|d, t)q(d|t)q(t)$$

induced by the abstract variational posterior $q_{\phi}(w|d, t)$. When $N \to \infty$, we have an abstract form of EB:

$$\mathbb{E}_{q(t)}\mathbb{E}_{q(d|t)}\left[\mathbb{E}_{q(w|d, t)}\left[ -\log p(d|w, t) \right] + D_{KL}(q(w|d, t)||p(w)) \right].$$
Empirical Bayes is related to information bottleneck

EB can be understood as matching the following processes:

Inference process:
\[ q(w, d, t) = q(t)q(d|t)q(w|d, t) \]

Generative process:
\[ p(w, d, t) = p(d|w, t)p(w)q(t) \]

Theorem

\[
\mathbb{E}_{q(t)}\mathbb{E}_{q(d|t)} \left[ \mathbb{E}_{q(w|d, t)} \left[ - \log p(d|w, t) \right] + D_{KL}(q(w|d, t)\| p(w)) \right] \\
\geq I_q(w; d|t) + H_{q,p}(d|w, t).
\]

In light of this connection, we call our method **synthetic information bottleneck (SIB)**.
Few-shot classification experiments

- **MiniImageNet** [Vinyals et al., 2016] contains 100 classes, split into 64 training classes, 16 validation classes and 20 testing classes, where each class consists of 600 image-label pairs and each image is of size $84 \times 84$.

- **CIFAR-FS** [Bertinetto et al., 2018] is created by dividing the original CIFAR-100 into 64 training classes, 16 validation classes and 20 testing classes; each image is of size $32 \times 32$. 
### Few-shot classification experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>MinImageNet, 5-way</th>
<th>CIFAR-FS, 5-way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>Matching Net [Vinyals et al., 2016]</td>
<td>Conv-4-64</td>
<td>44.2%</td>
<td>57%</td>
</tr>
<tr>
<td>MAML [Finn et al., 2017]</td>
<td>Conv-4-64</td>
<td>48.7±1.8%</td>
<td>63.1±0.9%</td>
</tr>
<tr>
<td>Prototypical Net [Snell et al., 2017]</td>
<td>Conv-4-64</td>
<td>49.4±0.8%</td>
<td>68.2±0.7%</td>
</tr>
<tr>
<td>Relation Net [Sung et al., 2018]</td>
<td>Conv-4-64</td>
<td>50.4±0.8%</td>
<td>65.3±0.7%</td>
</tr>
<tr>
<td>GNN [Satorras and Bruna, 2017]</td>
<td>Conv-4-64</td>
<td>50.3%</td>
<td>66.4%</td>
</tr>
<tr>
<td>R2-D2 [Bertinetto et al., 2018]</td>
<td>Conv-4-64</td>
<td>49.5±0.2%</td>
<td>65.4±0.2%</td>
</tr>
<tr>
<td>TPN [Liu et al., 2018]</td>
<td>Conv-4-64</td>
<td>55.5%</td>
<td>69.9%</td>
</tr>
<tr>
<td>Gidaris et al. [2019]</td>
<td>Conv-4-64</td>
<td>54.8±0.4%</td>
<td>71.9±0.3%</td>
</tr>
<tr>
<td><strong>SIB K=0 (Pre-trained feature)</strong></td>
<td>Conv-4-64</td>
<td>50.0±0.4%</td>
<td>67.0±0.4%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=3</strong></td>
<td>Conv-4-64</td>
<td><strong>58.0±0.6%</strong></td>
<td>70.7±0.4%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=0</strong></td>
<td>Conv-4-128</td>
<td>53.62±0.79%</td>
<td>71.48±0.64%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=1</strong></td>
<td>Conv-4-128</td>
<td>58.74±0.89%</td>
<td>74.12±0.63%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=3</strong></td>
<td>Conv-4-128</td>
<td>62.59±1.02%</td>
<td>75.43±0.67%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=5</strong></td>
<td>Conv-4-128</td>
<td><strong>63.26±1.07%</strong></td>
<td><strong>75.73±0.71%</strong></td>
</tr>
<tr>
<td>TADAM [Oreshkin et al., 2018]</td>
<td>ResNet-12</td>
<td>58.5±0.3%</td>
<td>76.7±0.3%</td>
</tr>
<tr>
<td>SNAIL [Santoro et al., 2017]</td>
<td>ResNet-12</td>
<td>55.7±1.0%</td>
<td>68.9±0.9%</td>
</tr>
<tr>
<td>MetaOptNet-RR [Lee et al., 2019]</td>
<td>ResNet-12</td>
<td>61.4±0.6%</td>
<td>77.9±0.5%</td>
</tr>
<tr>
<td>MetaOptNet-SVM [Lee et al., 2019]</td>
<td>ResNet-12</td>
<td>62.6±0.6%</td>
<td>78.6±0.5%</td>
</tr>
<tr>
<td>CTM [Li et al., 2019]</td>
<td>ResNet-18</td>
<td>64.1±0.8%</td>
<td><strong>80.5±0.1%</strong></td>
</tr>
<tr>
<td>Qiao et al. [2018]</td>
<td>WRN-28-10</td>
<td>59.6±0.4%</td>
<td>73.7±0.2%</td>
</tr>
<tr>
<td>LEO [Rusu et al., 2019]</td>
<td>WRN-28-10</td>
<td>61.8±0.1%</td>
<td>77.6±0.1%</td>
</tr>
<tr>
<td>Gidaris et al. [2019]</td>
<td>WRN-28-10</td>
<td>62.9±0.5%</td>
<td>79.9±0.3%</td>
</tr>
<tr>
<td><strong>SIB K=0 (Pre-trained feature)</strong></td>
<td>WRN-28-10</td>
<td>60.6±0.4%</td>
<td>77.5±0.3%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=1</strong></td>
<td>WRN-28-10</td>
<td>67.3±0.5%</td>
<td>78.2±0.3%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=3</strong></td>
<td>WRN-28-10</td>
<td>69.6±0.6%</td>
<td>78.9±0.4%</td>
</tr>
<tr>
<td><strong>SIB η=1e-3, K=5</strong></td>
<td>WRN-28-10</td>
<td><strong>70.0±0.6%</strong></td>
<td>78.9±0.4%</td>
</tr>
</tbody>
</table>
Variational information distillation for knowledge transfer
Deep learning is data-hungry

**Issue**: over-parameterized neural networks are often trained with huge data, which is infeasible for certain applications, such as

- Medical applications is constrained by the number of patients of a particular disease.
- Semantic segmentation requires pixel-level annotation.

A potential **solution**: transfer learning.

- **Finetuning**: initialize with the weights of the source network.
- **Teacher-student knowledge transfer** by Ba and Caruana [2014], Hinton et al. [2015].
Prior work on teacher-student knowledge transfer

It works well empirically but there is no commonly agreed theory behind this framework.

\[
\min \sum_i \| t^{(i)} - W^{(i)} s^{(i)} \|
\]

**Figure 1:** FitNet by Romero et al. [2014].

\[
\min \sum_i \| f(t^{(i)}) - f(s^{(i)}) \|^2_2
\]

**Figure 2:** Attention transfer by Zagoruyko and Komodakis [2016].
Denote by $t$ and $s$ the activations of the teacher and the student respectively. Intuitively, $I(t; s)$ is maximized when $t = s$. The idea is to add a term to the information bottleneck principle [Tishby et al., 2000]:

$$\min I(x; s) + \beta H(y|s) - \lambda I(t; s).$$
Variational information distillation (VID)

Knowledge transfer as a regularizer with SGD:

\[ \mathcal{L} = \text{Implicit regularization} + \text{Cross-entropy} - \sum_{k=1}^{K} \lambda_k I(t^{(k)}, s^{(k)}), \]

Recall the variational characterization:

\[ I(t; s) = H(t) - H(t|s) \]
\[ = H(t) + \mathbb{E}_{t,s}[\log p(t|s)] \]
\[ = H(t) + \mathbb{E}_{t,s}[\log q(t|s)] + \mathbb{E}_s[D_{KL}(p(t|s)||q(t|s))] \]
\[ \geq H(t) + \mathbb{E}_{t,s}[\log q(t|s)], \]

Instead of searching for all valid \( q \), we focus on diagonal Gaussians:

\[ -\log q(t|s) = \sum_{n=1}^{N} \log \sigma_n + \frac{(t_n - \mu_n(s))^2}{2\sigma_n^2} + \text{constant}, \]
**Experiments: transfer from ImageNet to bird data**

Dataset: Caltech-UCSD Birds 200.

Networks: teacher (ResNet-34), student (ResNet-18).

<table>
<thead>
<tr>
<th>data per class</th>
<th>≈29.95</th>
<th>20</th>
<th>10</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>37.22</td>
<td>24.33</td>
<td>12.00</td>
<td>7.09</td>
</tr>
<tr>
<td>Finetuned</td>
<td>76.69</td>
<td>71.00</td>
<td>59.25</td>
<td>44.07</td>
</tr>
<tr>
<td>LwF</td>
<td>55.18</td>
<td>42.13</td>
<td>26.23</td>
<td>14.27</td>
</tr>
<tr>
<td>FitNet</td>
<td>66.63</td>
<td>56.63</td>
<td>46.68</td>
<td>31.04</td>
</tr>
<tr>
<td>AT</td>
<td>54.62</td>
<td>41.44</td>
<td>28.90</td>
<td>16.55</td>
</tr>
<tr>
<td>NST</td>
<td>55.01</td>
<td>41.87</td>
<td>23.76</td>
<td>15.63</td>
</tr>
<tr>
<td>VID</td>
<td><strong>73.25</strong></td>
<td><strong>67.20</strong></td>
<td><strong>56.86</strong></td>
<td><strong>46.21</strong></td>
</tr>
</tbody>
</table>
Experiments: transfer from ImageNet to indoor-scene data


Networks: teacher (ResNet-34), student (VGG-9).

<table>
<thead>
<tr>
<th></th>
<th>data per class</th>
<th>≈80</th>
<th>50</th>
<th>25</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>53.58</td>
<td>43.96</td>
<td>29.70</td>
<td>15.97</td>
<td></td>
</tr>
<tr>
<td>Finetuned</td>
<td>65.97</td>
<td>58.51</td>
<td>51.72</td>
<td>39.63</td>
<td></td>
</tr>
<tr>
<td>LwF</td>
<td>60.90</td>
<td>52.01</td>
<td>41.57</td>
<td>27.76</td>
<td></td>
</tr>
<tr>
<td>FitNet</td>
<td>70.90</td>
<td>64.70</td>
<td>54.48</td>
<td>40.82</td>
<td></td>
</tr>
<tr>
<td>AT</td>
<td>60.90</td>
<td>52.16</td>
<td>42.76</td>
<td>25.60</td>
<td></td>
</tr>
<tr>
<td>NST</td>
<td>55.60</td>
<td>46.04</td>
<td>35.22</td>
<td>21.64</td>
<td></td>
</tr>
<tr>
<td>VID</td>
<td><strong>72.01</strong></td>
<td><strong>67.01</strong></td>
<td><strong>59.33</strong></td>
<td><strong>45.90</strong></td>
<td></td>
</tr>
</tbody>
</table>
Experiments: transfer from CNNs to MLPs

Dataset: CIFAR-10.

Networks: teacher (WRN-40-2), student (MLP).

<table>
<thead>
<tr>
<th>Network</th>
<th>MLP-4096</th>
<th>MLP-2048</th>
<th>MLP-1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>70.60</td>
<td>70.78</td>
<td>70.90</td>
</tr>
<tr>
<td>KD</td>
<td>70.42</td>
<td>70.53</td>
<td>70.79</td>
</tr>
<tr>
<td>FitNet</td>
<td>76.02</td>
<td>74.08</td>
<td>72.91</td>
</tr>
<tr>
<td>VID</td>
<td><strong>85.18</strong></td>
<td><strong>83.47</strong></td>
<td><strong>78.57</strong></td>
</tr>
<tr>
<td>Urban et al. [2017]</td>
<td></td>
<td></td>
<td>74.32</td>
</tr>
<tr>
<td>Lin et al. [2015]</td>
<td></td>
<td></td>
<td>78.62</td>
</tr>
</tbody>
</table>
Conclusion

- **Meta-learning**
  - Formulated transductive meta-learning with empirical Bayes model.
  - Implemented transductive amortized inference using synthetic gradient descent.
  - Achieved state-of-the-art results on few-shot learning benchmarks.
  - Derived the connection to Bayesian information bottleneck.

- **Transfer learning**
  - Proposed a teacher-student knowledge transfer framework inspired by information bottleneck.
  - Achieved state-of-the-art results on transfer learning benchmarks.
  - Empirically verified knowledge transfer between CNN and MLP.
Thank you for your attention!


