My Journey in Computer Vision, Machine Learning and Applied Math

Shell Xu Hu

Postdoc Researcher at KU Leuven

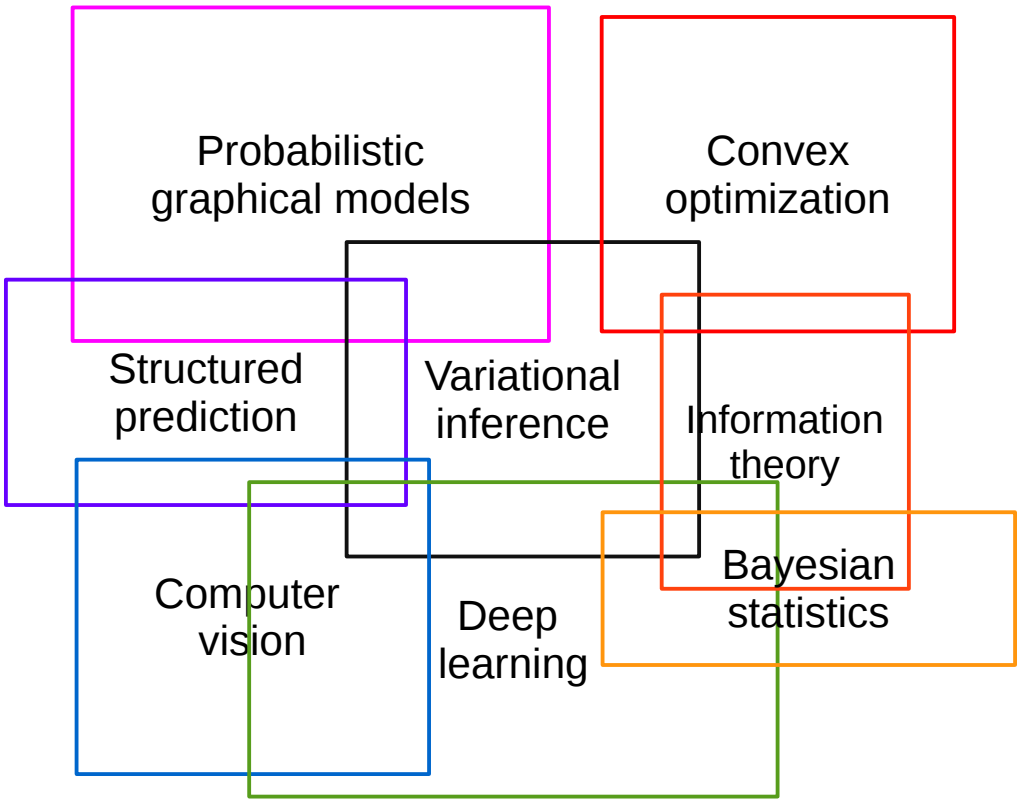
A brief introduction of myself

Research experience

- 2019.7–now, **Postdoc researcher**,
  Katholieke Universiteit Leuven,
  Supervisors: Johan Suykens and Panos Patrinos.
- 2015.3–2019.2, **PhD in Machine Learning**,
  École des Ponts ParisTech,
  Supervisors: Guillaume Obozinski and Nikos Komodakis.
- 2012.9–2015.3, **Master in Computer Science**,
  Oregon State University,
  Supervisors: Sinisa Todorovic and Tom Dietterich.
- 2011.9–2012.9, **Master in Computer Vision**,
  Universitat Autonòma de Barcelona,
  Supervisors: Marco Pedersoli and Jordi Gonzalez.
- 2006.9–2010.7, **Bachelor in Software Engineering**,
  Hangzhou Dianzi University.

My research topics

My website: `http://hushell.github.io/`



Publications during PhD

- Shell X. Hu, Pablo G. Moreno, Xi Shen, Yang Xiao, Guillaume Obozinski, Neil D. Lawrence, Andreas Damianou. **Empirical Bayes Transductive Meta-Learning with Synthetic Gradients**. NeurIPS 2019 Workshop on Meta-Learning. Long version accepted by ICLR 2020.
- Sungsoo Ahn, Shell X. Hu, Andreas Damianou, Neil D. Lawrence, Zhenwen Dai. **Variational Information Distillation for Knowledge Transfer**. CVPR 2019.
- Shell X. Hu, Sergey Zagoruyko, Nikos Komodakis. **Exploring Weight Symmetry in Deep Neural Networks**. Computer Vision and Image Understanding (CVIU), 187, p.102786, 2019.
- Shell X. Hu, Pablo G. Moreno, Andreas Damianou, Neil D. Lawrence. β-**BNN: A Rate-Distortion Perspective on Bayesian Neural Networks**. NeurIPS 2018 Workshop on Bayesian Deep Learning
- Shell X. Hu and Guillaume Obozinski. **SDCA-Powered Inexact Dual Augmented Lagrangian Method for Fast CRF Learning**. AISTATS 2018.

Highlights of my career

Undergraduate project: a cross-platform GUI toolkit for embedded system

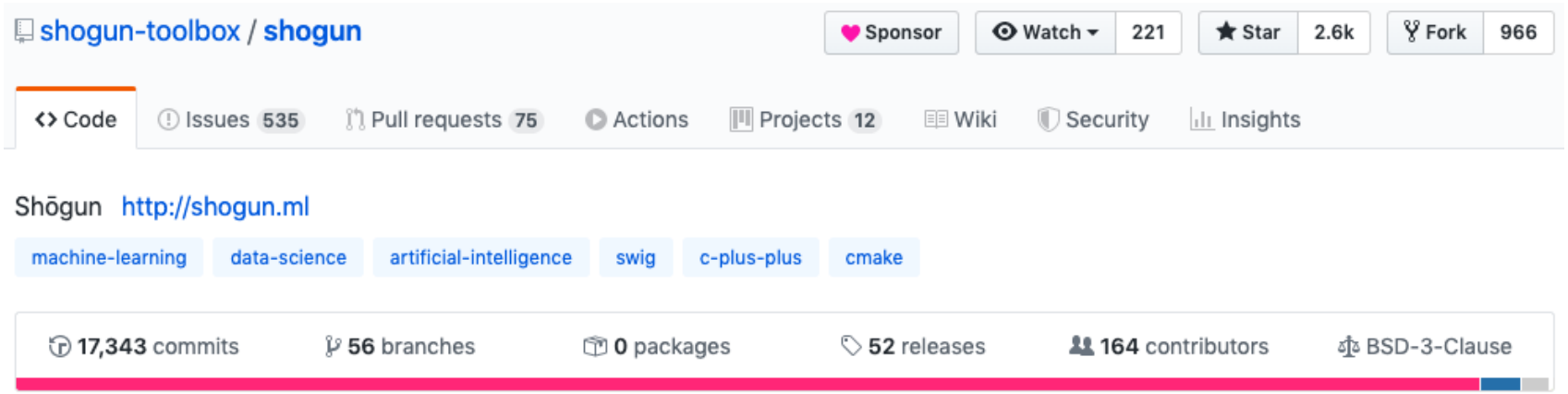
An Event Based GUI Programming Toolkit for Embedded System 18 2010
X Hu, C Jiang, W Zhang, J Zhang, R Yu, C Lv
2010 IEEE Asia-Pacific Services Computing Conference, 625-631

- A prototype of graphical interface toolkit (20,000+ lines of C++ code) for mobile devices with no operating system.
- Acquired by a startup called NewMsg in 2010.

Open source projects of Google summer of code

Contributed to Shogun Machine Learning Toolbox:

- GSOC 2013: General structured output learning framework.
- GSOC 2014: Structured Output Learning with Approximate Inference.



Mathematical project: a graphical model algorithm with convergence proofs

- Learned the theory of convex optimization from scratch.
- Proposed a new CRF learning algorithm with linear convergence proof; published at AISTATS 2018 (with Guillaume Obozinski):

$$
\begin{aligned}
D_\rho(\mu, \xi) &= -\sum_{c \in C} f_c^*(\mu_c) - r(\mu) \qquad \text{with} \qquad (7)\\
f_c^*(\mu_c) &:= -\gamma h_c(\mu_c) + \iota_{\triangle_c}(\mu_c)\\
r(\mu) &:= -\langle A^\intercal \xi + \ell, \mu \rangle + \frac{1}{2\lambda}\|\Psi\mu\|^2 + \frac{1}{2\rho}\|A\mu\|^2,
\end{aligned}
$$

Lemma 1 (Linear convergence of the outer iteration). *Let $\mathcal{A}$ be an algorithm that approximately solves $\max_\mu D_\rho(\mu, \xi^t)$ in the sense that*

$$\exists \beta \in (0,1), \qquad \mathbb{E}[\hat{\Delta}_t] \leq \beta\, \mathbb{E}[\Delta_t^0].$$

Then, $\exists \kappa \in (0,1)$ characterizing $d(\xi)$ and $C > 0$, such that, if $\lambda_{\max}(\beta)$ is the largest eigenvalue of the matrix

$$M(\beta) = \begin{bmatrix} 6\beta & 3\beta \\ 1 & 1-\kappa \end{bmatrix},$$

then after T_{ex} iterations of Algorithm 1 we have

$$\left\| \begin{matrix} \mathbb{E}[\hat{\Delta}_{T_{\text{ex}}}] \\ \mathbb{E}[\Gamma_{T_{\text{ex}}}] \end{matrix} \right\| \leq C\, \lambda_{\max}(\beta)^{T_{\text{ex}}} \left\| \begin{matrix} \mathbb{E}[\hat{\Delta}_0] \\ \mathbb{E}[\Gamma_0] \end{matrix} \right\|.$$

Corollary 1. *If $\mathcal{A}$ is a linearly convergent algorithm with rate π and if it is run for T_{in} iterations, such that, for some β: $\lambda_{\max}(\beta) < 1$, we have $(1-\pi)^{T_{\text{in}}} \leq \beta$, then $\mathbb{E}[\hat{\Delta}_t]$ and $\mathbb{E}[\Gamma_t]$ converge linearly to 0.*

Corollary 2. *Let $D_\infty(\mu) := \langle \ell, \mu \rangle - \gamma F_{\mathcal{I}}^*(\mu) - \frac{1}{2\lambda}\|\Psi\mu\|_2^2$, so that we have $D(\mu) = D_\infty(\mu) - \iota_{\{A\mu=0\}}$. If Δ_t and Γ_t converge linearly to 0, then $|D_\infty(\hat{\mu}^t) - D_\infty(\mu^\star)|$ and $\|A\hat{\mu}^t\|_2^2$ both converge to 0 linearly.*

Corollary 3. *With the notations of the previous corollary, for any $\beta \in (0,1)$ such that $\lambda_{\max}(\beta) < 1$, it is possible to obtain $\mathbb{E}[\hat{\Delta}_t] \leq \epsilon$ and $\mathbb{E}[\Gamma_t] \leq \epsilon$ with a total number of inner iterations $T_{\text{tot}} := T_{\text{in}} T_{\text{ex}}$ such that*

$$T_{\text{tot}} \geq \frac{\log(\beta)}{\log \lambda_{\max}(\beta) \log(1-\pi)} \log(\epsilon).$$

Corollary 4. *Let $\Delta^\star_{t\, T_{\text{in}}+s} := \Delta_t^s + \Gamma_t$. If $\kappa < \frac{1}{2}$ and $\alpha = \frac{1}{12}$, if $T_{\text{in}} \geq \frac{\log(\alpha\kappa)}{\log(1-\pi)}$, then, there exist a constant $C' > 0$ such that after a total of s inner updates, we have*

$$\mathbb{E}[\Delta_s^\star] \leq C' \left(1 - \frac{\kappa\pi}{2\log(12/\kappa)}\right)^s.$$

Proposition 1. *If $\mathcal{A}$ is SDCA, let $|\mathcal{C}|$ be the total number of cliques, σ_c the strong convexity constant of f_c^*, and L_c the Lipschitz constant of $\mu_c \mapsto r(\mu)$, then $\mathcal{A}$ is linearly convergent with rate $\pi = \min_{c \in \mathcal{C}} \frac{\sigma_c}{|\mathcal{C}|(\sigma_c + L_c)}$.*

Proposition 2. *Let $\hat{w}^t = w(\hat{\mu}^t)$. If $\mathcal{A}$ is SDCA, then*

$$\mathbb{E}[P(\hat{w}^t) - P(w^\star)] \leq \frac{1}{\pi}\mathbb{E}[\hat{\Delta}_t] + \mathbb{E}[\Gamma_t].$$

Proposition 3. *Let $w^{t,s} = w(\mu^{t,s})$. If $\mathcal{A}$ is a linearly convergent algorithm and the function $\mu \mapsto -H_{\text{approx}} + \frac{1}{2\rho}\|A\mu\|_2^2$ is strongly convex, then $P(w^{t,s}) - P(w^\star)$ converges to 0 linearly.*

Deep learning projects: small-data problems, variational inference and information bottleneck

I started to work on deep learning in 2017, focused on small-data regime with information bottleneck principle:

- CVPR 2019: Mutual information based knowledge distillation for **transfer learning**.
- ICLR 2020: Empirical Bayes formulation for **meta-learning**.
- IJCAI 2020 (submitted): Unifying self-supervision and data augmentation for **domain adaptation**.

Other computer vision projects

- Deformable parts models:
  - Towards real-time pedestrian detection with CUDA-HOG.
  - Reconfigurable parts for fine-grained object recognition.
- Semantic segmentation on images and 3D point clouds.
- Probabilistic image segmentation.
- Image compression.
- Climate modeling with satellite image time series.

Information-theoretical deep learning: an overview

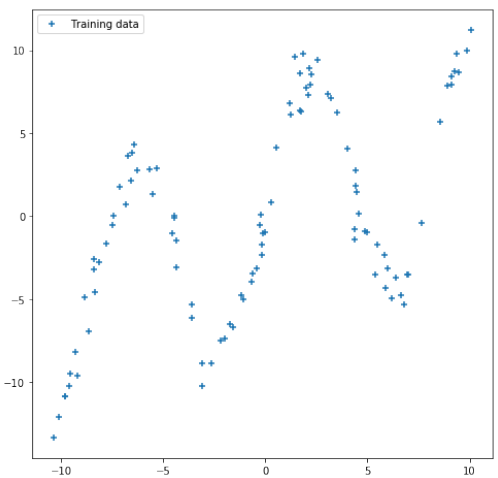
Machine learning paradigms

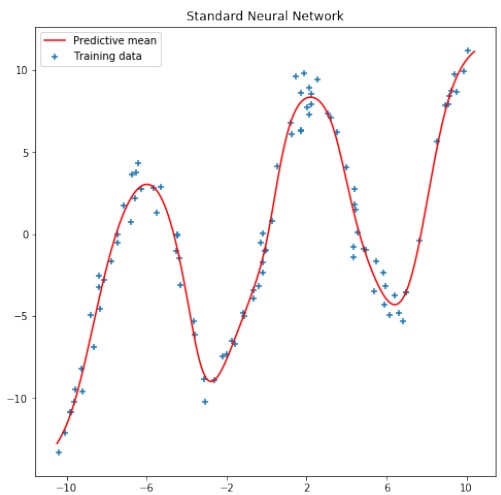
- Frequentist's paradigm: $p(y_*|x_*; w_{\text{train}})$, where w_{train} is the **maximum likelihood estimate** from $\mathcal{D}_{\text{train}}$.
- Bayesian's paradigm:

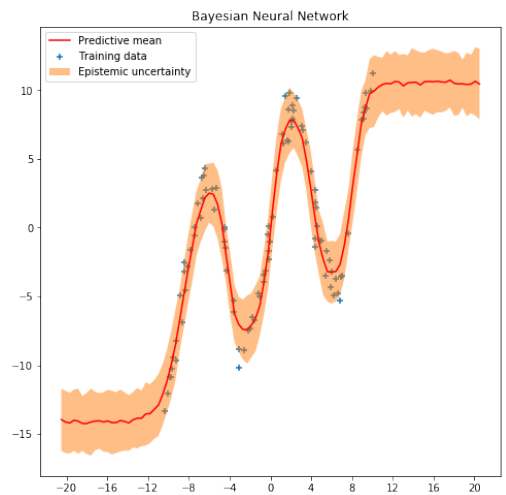
$$p(y_*|x_*, \mathcal{D}_{\text{train}}) = \int_{\mathcal{W}} p(y_*|x_*, w) p(w|\mathcal{D}_{\text{train}}) dw$$
$$\approx \int_{\mathcal{W}} p(y_*|x_*, w)\ q_{\text{train}}(w) dw =: q(y_*|x_*),$$

where q_{train} is obtained via **variational inference**:

$$q_{\text{train}} = \arg\min_{q \in \mathcal{Q}} D_{\text{KL}}\Big(q(w) \parallel p(w|\mathcal{D}_{\text{train}})\Big)$$







Uncertainty estimation granted by Bayesian paradigm

In practice, we use the Monte-Carlo estimate for the predictive distribution:

$$q(y_*|x_*) \approx \frac{1}{S}\sum_{i=1}^{S} p(y_*|x_*, w_s), \quad \text{where } w_s \sim q_{\text{train}}(w).$$

The variance of the predictions under $q(y_*|x_*)$ is given by

$$\begin{aligned}
\mathbb{VAR}_{q(y_*|x_*)}[y_*] &= \mathbb{E}_{q(y_*|x_*)}[y_* y_*^\top] - \mathbb{E}_{q(y_*|x_*)}[y_*]\,\mathbb{E}_{q(y_*|x_*)}[y_*]^\top \\
&= \underbrace{\int \left[\text{diag}(\mathbb{E}_{p(y_*|x_*,w)}[y_*]) - \mathbb{E}_{p(y_*|x_*,w)}[y_*]\,\mathbb{E}_{p(y_*|x_*,w)}[y_*]^\top\right] q_{\text{train}}(w)dw}_{\text{aleatoric (data) uncertainty}} \\
&+ \underbrace{\int \Big(\mathbb{E}_{p(y_*|x_*,w)}[y_*] - \mathbb{E}_{q(y_*|x_*)}[y_*]\Big)\Big(\mathbb{E}_p[y_*] - \mathbb{E}_q[y_*]\Big)^\top q_{\text{train}}(w)dw}_{\text{epistemic (model) uncertainty}}
\end{aligned}$$

Information bottleneck for frequentist paradigm

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

$$\text{min } \textbf{Rate} \qquad \min_{q(\hat{x}|x)} \; I_q(X;\hat{X})$$

$$\text{s.t. } \textbf{Distortion} \leq \text{const} \qquad \text{s.t.} \underbrace{\sum_{x,\hat{x}} p(x)q(\hat{x}|x)\; d(x,\hat{x})}_{D_q(X,\hat{X})} \leq \text{const.}$$

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

$$d(x,\hat{x}) = D_{\mathrm{KL}}\big(p(y|x)\|p(y|\hat{x})\big).$$

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

$$\min_{q(\hat{x}|x)} I_q(X;\hat{X}) - \beta\; I_q(Y;\hat{X}).$$

Information bottleneck for Bayesian paradigm

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

$$\text{Generative process}: \quad P(S, w) = p(S \mid w)p(w).$$

The variational posterior $q(w|S)$ induces

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

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

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

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

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

This is an alternative objective for variational inference.

Empirical Bayes transductive meta-learning with synthetic gradients

Meta-learning: a framework for small-data problems

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:

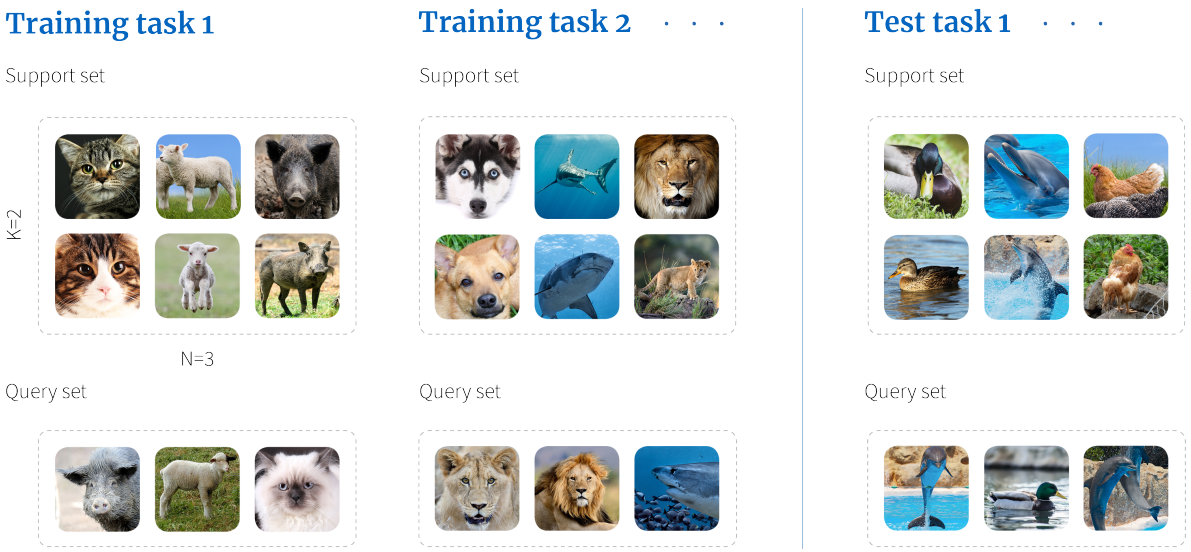
- Learning to do gradient descent [Andrychowicz et al., 2016].
- Learning to classify unseen categories [Vinyals et al., 2016].
- Learning to generalize across domains [Li et al., 2017].

An example: few-shot classification

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

	Support set $d_t^l := \{(x_{t,i}^l, y_{t,i}^l)\}_{i=1}^{n^l}$	**Query set** $x_t := \{x_{t,i}\}_{i=1}^{n}$	$y_t = \{y_{t,i}\}_{i=1}^{n}$
	Labeled data	Unlabeled data	
Training	✓	✓	✓
Testing	✓	✓	✗

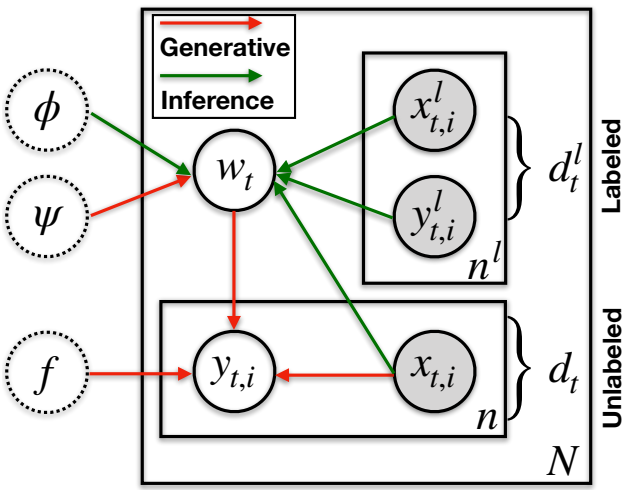
N-way-K-shot setup:



From hierarchical Bayes (HB) to empirical Bayes (EB)

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

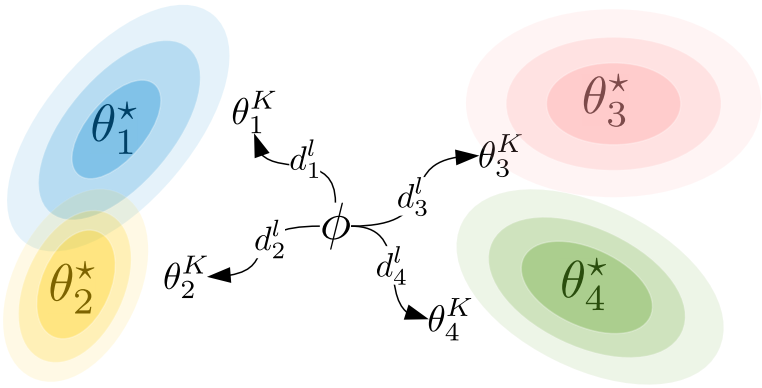
$$\text{HB} \to \text{EB}: \quad p_f(\mathcal{D}) \to p_{\psi,f}(\mathcal{D}) = \int_\psi \Big[\prod_{t=1}^N \int_{w_t} p_f(d_t|w_t) p(w_t|\psi)\Big] p(\psi),$$



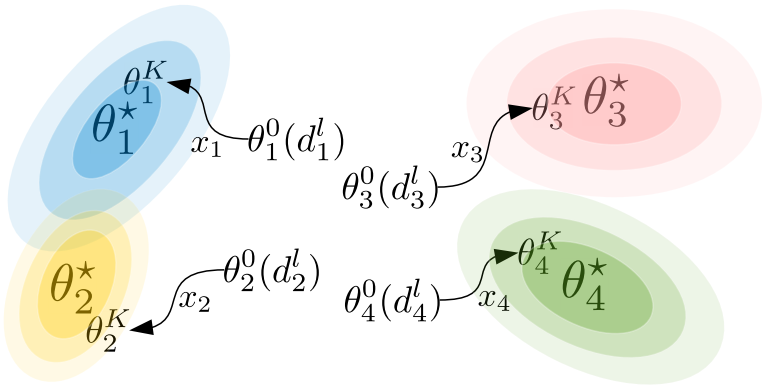
$$\log p_f(d_t|w_t)$$
$$= \sum_{i=1}^n \log p_f(y_{t,i}|x_{t,i}, w_t) + \log p(x_{t,i}|w_t)$$
$$= -\sum_{i=1}^n \ell_t\big(\hat{y}_{t,i}(f(x_{t,i}), w_t), y_{t,i}\big) + \text{const}$$

What is missing in MAML?

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.

Variational inference for empirical Bayes

We derive an ELBO on the log-likelihood by introducing a variational distribution $q_{\theta_t}(w_t)$ for each task with parameter θ_t:

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

Maximizing the ELBO with respect to $\theta_1, \ldots, \theta_N$ and ψ is equivalent to

$$\min_{\theta_1,\ldots,\theta_N} \sum_{t=1}^{N} D_{\mathrm{KL}}\Big(q_{\theta_t}(\mathrm{w}_t) \;\Big\|\; p_{\psi,f}(\mathrm{w}_t|\mathrm{d}_t) \Big)$$

Amortized inference [Kingma and Welling, 2013] with transduction

$$\text{Exact VI :} \qquad \min_{\theta_1,\ldots,\theta_N} \sum_{t=1}^{N} D_{\mathrm{KL}}\left(q_{\theta_t}(\mathrm{w}_t) \;\middle\|\; p_{\psi,f}(\mathrm{w}_t|\mathrm{d}_t) \right)$$

For scalable inference, we introduce a neural network ϕ to output θ_t.
There are **two choices to do the amortization**:

$$\text{Inductive AVI :} \qquad \min_{\phi} \sum_{t=1}^{N} D_{\mathrm{KL}}\left(q_{\phi(d_t^I)}(\mathrm{w}_t) \;\middle\|\; p_{\psi,f}(\mathrm{w}_t|\mathrm{d}_t) \right)$$

$$\text{Transductive AVI :} \qquad \min_{\phi} \sum_{t=1}^{N} D_{\mathrm{KL}}\left(q_{\phi(d_t^I,x_t)}(\mathrm{w}_t) \;\middle\|\; p_{\psi,f}(\mathrm{w}_t|\mathrm{d}_t) \right)$$

Unrolling exact inference with synthetic gradient

How do we implement the amortization network $\phi(x_t, d_t^l)$?

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

$$\phi(d_t^l, x_t) = \arg\min_{\theta_t} D_{\mathrm{KL}}\Big(q_{\theta_t}(\mathrm{w}_t) \;\Big\|\; p_{\psi,f}(\mathrm{w}_t|\mathrm{d}_t)\Big)$$

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

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

up to the K-th step by parameterizing

- the **initialization** θ_t^0;
- the **gradient** $\nabla_{\theta_t} D_{\mathrm{KL}}\Big(q_{\theta_t^k}(w) \,\|\, p_{\psi,f}(w|d_t)\Big)$.

Unrolling exact inference with synthetic gradient

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

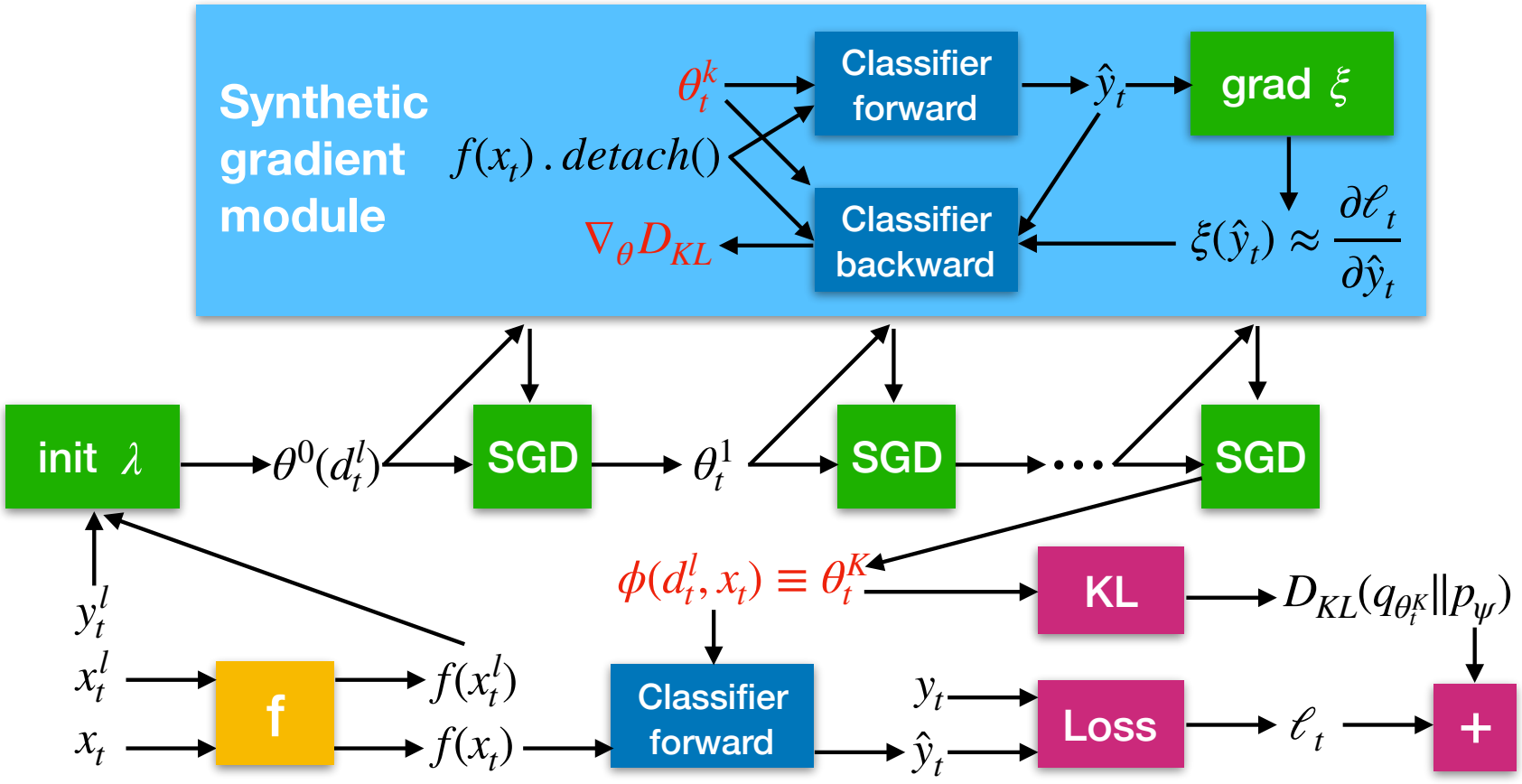
$$\nabla_{\theta_t} D_{\mathrm{KL}}\Big(q_{\theta_t} \| p_{\psi,f}\Big) = \mathbb{E}_\epsilon\Big[\sum_{i=1}^{n} \frac{\partial \ell_t(\hat{y}_{t,i}, y_{t,i})}{\partial \hat{y}_{t,i}} \frac{\partial \hat{y}_{t,i}}{\partial w_t} \frac{\partial w_t(\theta_t, \epsilon)}{\partial \theta_t}\Big] + \nabla_{\theta_t}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

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:

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

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

Computation graph of our method



Variational EM algorithm

while not converged **do**

  Sample a task t and its data: d_t, d_t^l.

  Compute the initialization $\theta_t^0 = \lambda(d_t^l)$.

  %========== E-step ==========

  for $k = 1, \ldots, K$ **do**

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

  %========== M-step ==========

  Update $\phi \leftarrow \phi - \eta \nabla_\phi D_{\mathrm{KL}}(q_{\phi(x_t, d_t^l)} \| p_f \cdot p_\psi)$.

  Update $\psi \leftarrow \psi - \eta \nabla_\psi D_{\mathrm{KL}}(q_{\theta_t^K(\psi)} \| p_\psi)$.

  Optionally, update $f \leftarrow f + \eta \nabla_f \log p_f(d_t | w_t)$.

Empirical Bayes is related to information bottleneck

EB can be understood as matching the following processes:

$$\text{Inference process :} \qquad q(w, d, t) = q(t)q(d|t)q(w \mid d, t)$$

$$\text{Generative process :} \qquad p(w, d, t) = p(d \mid w, t)p(w)q(t)$$

Theorem

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

In light of this connection, we call our method **synthetic information bottleneck** (SIB).

Few-shot classification experiments

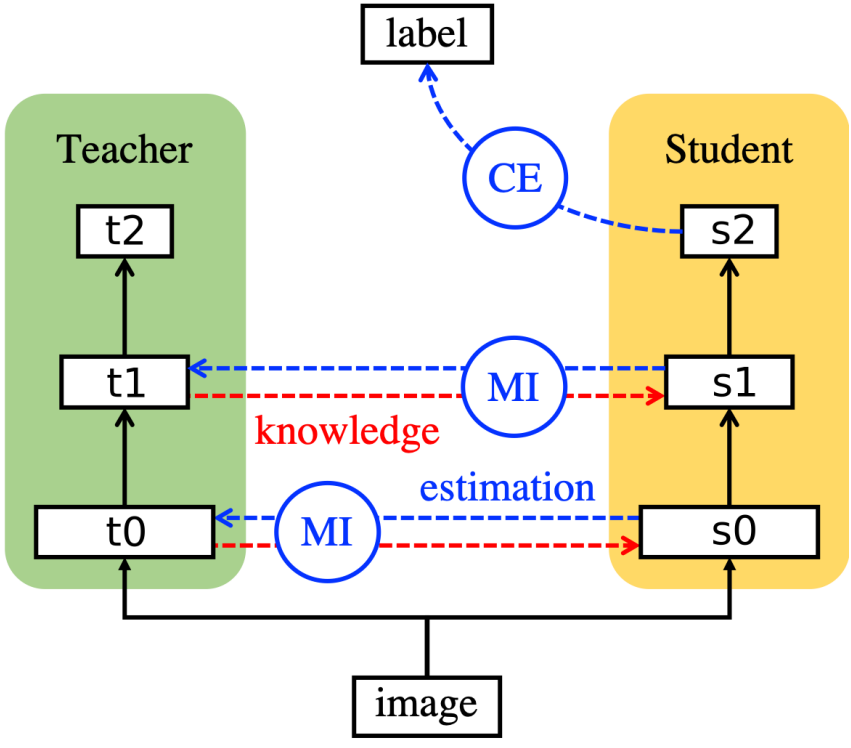
- **MiniImageNet** [Vinyals et al., 2016] contains 100 classes, split into 64 training classes, 16 validation classes and 20 testing classes, where each class consists of 600 image-label pairs and each image is of size 84×84.
- **CIFAR-FS** [Bertinetto et al., 2018] is created by dividing the original CIFAR-100 into 64 training classes, 16 validation classes and 20 testing classes; each image is of size 32×32.

Few-shot classification experiments

Method	Backbone	MiniImageNet, 5-way		CIFAR-FS, 5-way	
		1-shot	5-shot	1-shot	5-shot
Matching Net [Vinyals et al., 2016]	Conv-4-64	44.2%	57%	–	–
MAML [Finn et al., 2017]	Conv-4-64	48.7±1.8%	63.1±0.9%	58.9±1.9%	71.5±1.0%
Prototypical Net [Snell et al., 2017]	Conv-4-64	49.4±0.8%	68.2±0.7%	55.5±0.7%	72.0±0.6%
Relation Net [Sung et al., 2018]	Conv-4-64	50.4±0.8%	65.3±0.7%	55.0±1.0%	69.3±0.8%
GNN [Satorras and Bruna, 2017]	Conv-4-64	50.3%	66.4%	61.9%	75.3%
R2-D2 [Bertinetto et al., 2018]	Conv-4-64	49.5±0.2%	65.4±0.2%	62.3±0.2%	77.4±0.2%
TPN [Liu et al., 2018]	Conv-4-64	55.5%	69.9%	–	–
Gidaris et al. [2019]	Conv-4-64	54.8±0.4%	**71.9±0.3%**	63.5±0.3%	**79.8±0.2%**
SIB K=0 (*Pre-trained feature*)	Conv-4-64	50.0±0.4%	67.0±0.4%	59.2±0.5%	75.4±0.4%
SIB η=1e-3, K=3	Conv-4-64	**58.0±0.6%**	70.7±0.4%	**68.7±0.6%**	77.1±0.4%
SIB η=1e-3, K=0	Conv-4-128	53.62 ± 0.79%	71.48 ± 0.64%	–	–
SIB η=1e-3, K=1	Conv-4-128	58.74 ± 0.89%	74.12 ± 0.63%	–	–
SIB η=1e-3, K=3	Conv-4-128	62.59 ± 1.02%	75.43 ± 0.67%	–	–
SIB η=1e-3, K=5	Conv-4-128	**63.26 ± 1.07%**	**75.73 ± 0.71%**	–	–
TADAM [Oreshkin et al., 2018]	ResNet-12	58.5±0.3%	76.7±0.3%	–	–
SNAIL [Santoro et al., 2017]	ResNet-12	55.7±1.0%	68.9±0.9%	–	–
MetaOptNet-RR [Lee et al., 2019]	ResNet-12	61.4±0.6%	77.9±0.5%	72.6±0.7%	84.3±0.5%
MetaOptNet-SVM [Lee et al., 2019]	ResNet-12	62.6±0.6%	78.6±0.5%	72.0±0.7%	84.2±0.5%
CTM [Li et al., 2019]	ResNet-18	64.1±0.8%	**80.5±0.1%**	–	–
Qiao et al. [2018]	WRN-28-10	59.6±0.4%	73.7±0.2%	–	–
LEO [Rusu et al., 2019]	WRN-28-10	61.8±0.1%	77.6±0.1%	–	–
Gidaris et al. [2019]	WRN-28-10	62.9±0.5%	79.9±0.3%	73.6±0.3%	**86.1±0.2%**
SIB K=0 (*Pre-trained feature*)	WRN-28-10	60.6±0.4%	77.5±0.3%	70.0±0.5%	83.5±0.4%
SIB η=1e-3, K=1	WRN-28-10	67.3±0.5%	78.2±0.3%	76.8±0.5%	84.9±0.4%
SIB η=1e-3, K=3	WRN-28-10	69.6±0.6 %	78.9±0.4%	78.4±0.6%	85.3±0.4%
SIB η=1e-3, K=5	WRN-28-10	**70.0±0.6%**	78.9±0.4%	**80.0±0.6%**	85.3±0.4%

Variational information distillation for knowledge transfer

Mutual information for knowledge transfer



Denote by $\boldsymbol{t}$ and $\boldsymbol{s}$ the activations of the teacher and the student respectively. Intuitively, $I(\boldsymbol{t}; \boldsymbol{s})$ is maximized when $\boldsymbol{t} = \boldsymbol{s}$. The idea is to add a term to the *information bottleneck* principle [Tishby et al., 2000]:

$$\text{minimize } I(\boldsymbol{x}; \boldsymbol{s}) - \beta\ I(\boldsymbol{y}; \boldsymbol{s}) - \lambda\ I(\boldsymbol{t}; \boldsymbol{s})$$

Variational information distillation (VID)

Knowledge transfer as a regularizer with SGD:

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

Recall the variational characterization:

$$\begin{aligned} I(\boldsymbol{t}; \boldsymbol{s}) &= H(\boldsymbol{t}) - H(\boldsymbol{t}|\boldsymbol{s}) \\ &= H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log p(\boldsymbol{t}|\boldsymbol{s})] \\ &= H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})] + \mathbb{E}_{\boldsymbol{s}}[D_{\mathrm{KL}}(p(\boldsymbol{t}|\boldsymbol{s})||q(\boldsymbol{t}|\boldsymbol{s}))] \\ &\geq H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})], \end{aligned}$$

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

$$-\log q(\boldsymbol{t}|\boldsymbol{s}) = \sum_{n=1}^{N} \log \sigma_n^2 + \frac{(t_n - \mu_n(\boldsymbol{s}))^2}{2\sigma_n^2} + \text{constant},$$

Experiments: transfer from ImageNet to indoor-scene data

Dataset: MIT-67.

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

data per class	≈80	50	25	10
Student only	53.58	43.96	29.70	15.97
Finetuned from ImageNet	65.97	58.51	51.72	39.63
LwF	60.90	52.01	41.57	27.76
FitNet	70.90	64.70	54.48	40.82
AT	60.90	52.16	42.76	25.60
NST	55.60	46.04	35.22	21.64
VID	**72.01**	**67.01**	**59.33**	**45.90**

Experiments: transfer from CNNs to MLPs

Dataset: CIFAR-10.

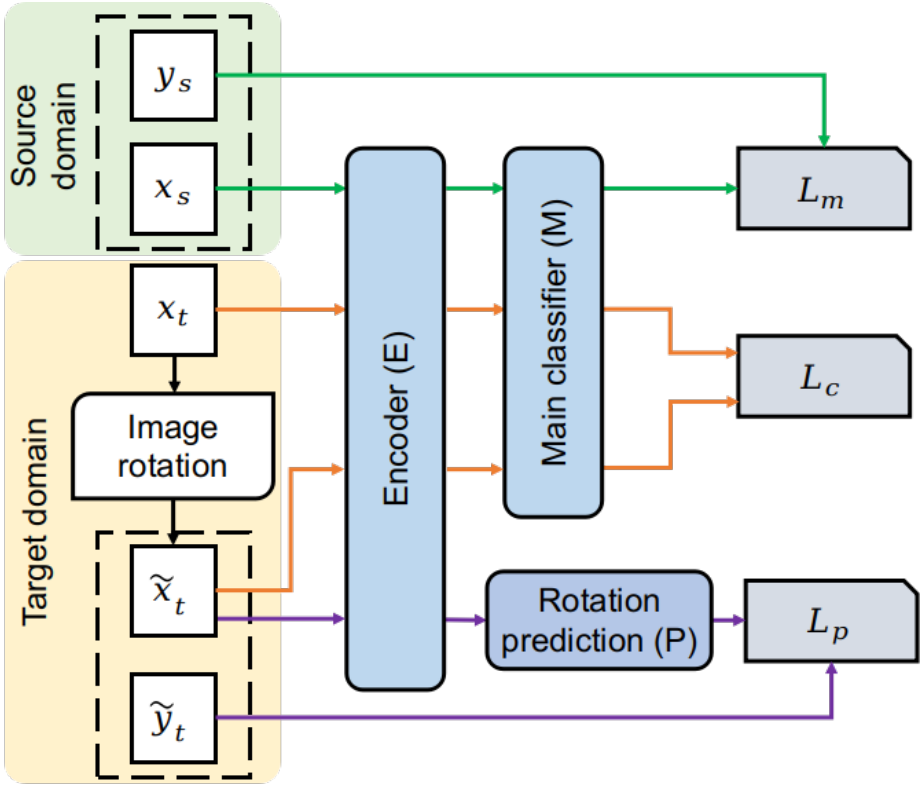
Networks: teacher (WRN-40-2), student (MLP).

Network	MLP-4096	MLP-2048	MLP-1024	Urban17	Lin15
Student only	70.60	70.78	70.90	74.32	78.62
KD	70.42	70.53	70.79		
FitNet	76.02	74.08	72.91		
VID	**85.18**	**83.47**	**78.57**		

Unifying self-supervision and data-augmentation for domain adaptation

The overall idea

Image rotation is used in defining both the **self-supervision loss** and the **data-augmentation consistency loss**.



Minimizing the consistency loss amounts to maximizing

$$-I(\tilde{\mathbf{x}}; y) = \mathbb{E}_{p(\mathbf{x},\tilde{\mathbf{x}})}\Big[D_{\mathrm{KL}}\big(p(y|\mathbf{x})\|p(y|\tilde{\mathbf{x}})\big) - D_{\mathrm{KL}}\big(p(y|\mathbf{x})\|p(y)\big)\Big],$$

Experiments: domain adaptation on Office-Home dataset

Office-Home dataset contains 4 domains, 65 categories, about 15,500 images. The 4 domains are: Art (**ar**), Clipart (**cl**), Product (**pr**) and Real-World (**rw**).

Method	*ar→cl*	*ar→pr*	*ar→rw*	*cl→ar*	*cl→pr*	*cl→rw*	*pr→ar*	*pr→cl*	*pr→rw*	*rw→ar*	*rw→cl*	*rw→pr*	Avg.
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E	50.7	**70.6**	**76.0**	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
Rot	50.4	67.8	74.6	58.7	66.7	67.4	55.7	52.4	77.5	71.0	59.6	81.2	65.3
Ours	**51.7**	69.0	75.4	**60.4**	**70.3**	**70.7**	**57.7**	**53.3**	**78.6**	**72.2**	**59.9**	**81.7**	**66.7**

Table 1: Accuracy (%) on Office-Home (ResNet-50).

Thank you for your attention!

References

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentangling in deep representations. *arXiv preprint arXiv:1706.01350*, 2017.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2018.

References ii

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. *arXiv preprint arXiv:1906.05186*, 2019.

Shell X. Hu, G. Pablo Moreno, Lawrence D. Neil, and Andreas Damianou. β-bnn: A rate-distortion perspective on bayesian neural networks. *NeurIPS-BDL*, 2018.

References iii

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1627–1635. JMLR, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.

References iv

Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding Task-Relevant Features for Few-Shot Learning by Category Traversal. In *CVPR*, 2019.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

References v

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BJgklhAcK7`.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

Victor Garcia Satorras and Joan Bruna. Few-shot learning with graph neural networks. *ArXiv*, abs/1711.04043, 2017.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Phd thesis, Technische Universitat Munchen, Germany, 1987. URL `http://www.idsia.ch/~juergen/diploma.html`.

References vi

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Kluwer Academic Publishers, 1998.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.