EMPIRICAL BAYES TRANSDUCTIVE META-LEARNING WITH SYNTHETIC GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a meta-learning approach that learns from multiple tasks in a transductive setting, by leveraging unlabeled information in the query set to learn a more powerful meta-model. To develop our framework we revisit the empirical Bayes formulation for multi-task learning. The evidence lower bound of the marginal log-likelihood of empirical Bayes decomposes as a sum of local KL divergences between the variational posterior and the true posterior of each task. We derive a novel amortized variational inference that couples all the variational posteriors into a meta-model, which consists of a synthetic gradient network and an initialization network. The combination of local KL divergences and synthetic gradient network allows for backpropagating information from unlabeled data, thereby enabling transduction. Our results on the Mini-ImageNet and CIFAR-FS benchmarks for episodic few-shot classification significantly outperform previous state-of-the-art methods.

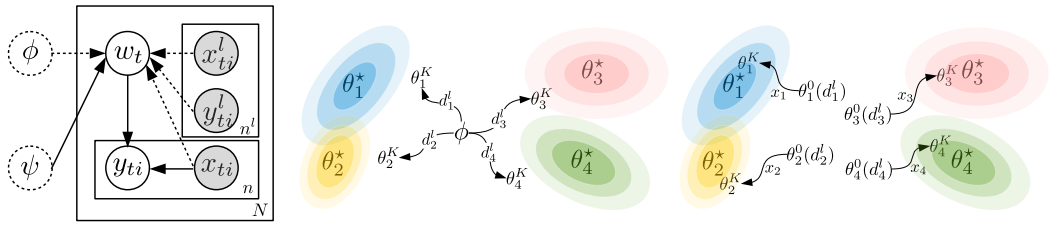
1 INTRODUCTION

While supervised learning of deep neural networks can achieve or even surpass human-level performance (He et al., 2015; Devlin et al., 2018), they can hardly extrapolate the learned knowledge beyond the domain where the supervision is provided. The problem of solving rapidly a new task after learning several other similar tasks is called *meta-learning* (Schmidhuber, 1987; Bengio et al., 1991; Thrun & Pratt, 1998); typically, the data is presented in a two-level hierarchy such that each data point at the higher level is itself a dataset associated with a task, and the goal is to learn a *meta-model* that generalizes across tasks. In this paper, we focus on *few-shot learning* (Vinyals et al., 2016), an instance of meta-learning problems, where a task t, in *meta-testing*, consists of an *unlabeled set* $x_t := \{x_{t,i}\}_{i=1}^n$ and a *labeled set* (aka *support set*) $d_t^l := \{(x_{t,i}^l, y_{t,i}^l)\}_{i=1}^{n^l}$, and the goal is to predict the corresponding labels, namely $y_t = \{y_{t,i}\}_{i=1}^n$, for the unlabeled set. In *meta-training*, y_t is provided as ground truth. The set $d_t := (x_t, y_t)$ is sometimes referred to as *query set*.

A particular important distinction to make is whether each task is solved in a *transductive* or *inductive* manner. The inductive setting is what was originally proposed by Vinyals et al. (2016): we use d_t^l to train a model and test it on x_t (one example at a time). Transduction, however, has the advantage of being able to see all points in x_t before making predictions. We argue that the transductive setting is more relevant to the problem since, as in semi-supervised learning, an inductive learner can always be built from a transductive one (Chapelle et al., 2006). In fact, Nichol et al. (2018) notice that most of the existing meta-learning methods follow the transductive setting unintentionally since they use x_t implicitly via the *batch normalization* (Ioffe & Szegedy, 2015).

Due to the hierarchical structure of the data, it is natural to formulate meta-learning as an instance of *hierarchical Bayes* (HB) (Good, 1980; Berger, 1985), or alternatively, empirical Bayes (EB) (Robbins, 1985; Kucukelbir & Blei, 2014). The difference is that the latter restricts the learning of meta-parameters to point estimates. In this paper, we focus on the EB model, since it largely simplifies the training and testing without losing the strength of the HB formulation.

The idea of using HB or EB for meta-learning is not new: Amit & Meir (2018) derive an objective similar to that of HB using PAC-Bayesian analysis; Grant et al. (2018) show that MAML (Finn et al., 2017) can be understood as a EB method; Ravi & Beatson (2018) consider a HB extension to MAML and compute posteriors via amortized variational inference. However, unlike our proposal, these



(a) Graphical model of EB (b) MAML (c) Our method (SIB)

Figure 1: **(a)** The generative and inference processes of the empirical Bayes model are depicted in solid and dashed arrows respectively, where the meta-parameters are denoted by dashed circles due to the point estimates. A comparison between MAML (7) and our method (SIB) (9) is shown in **(b)** and **(c)**. MAML is an inductive method since, for a task t, it first constructs a variational posterior $q_{\theta_t^K}$ as a function of the labeled set d_t^l, and then test on the unlabeled set x_t; while SIB constructs a better variational posterior as a function of both d_t^l and x_t: it starts from an initialization $\theta_t^0(d_t^l)$, and then yields θ_t^K by running K synthetic gradient steps on x_t.

methods do not take advantage of the unlabeled set. Roughly speaking, they construct the variational posterior as a function of the labeled set d_t^l without taking advantage of the unlabeled set x_t. The situation is similar in gradient based meta-learning methods (Finn et al., 2017; Ravi & Larochelle, 2016; Li et al., 2017; Nichol et al., 2018; Flennerhag et al., 2019) and many other meta-learning methods (Vinyals et al., 2016; Snell et al., 2017; Gidaris & Komodakis, 2018), where the mechanisms used to generate the task-specific parameters rely on groundtruth labels, thus, there is no place for the unlabeled set to contribute. We argue that this is a suboptimal choice, which may lead to overfitting when the labeled set is small and hinder the possibility of zero-shot learning (when the labeled set is not provided). An exception is Liu et al. (2018). They reuse the label propagation algorithm (Zhu et al., 2003) for transductive inference within each task and show that transduction is useful for boosting the performance.

In this paper, we propose to use synthetic gradient (Jaderberg et al., 2017) to enable transductivity, such that the variational posterior is implemented as a function of the labeled set d_t^l and the unlabeled set x_t. The synthetic gradient is produced by a neural network and learned to be a surrogate of the true gradient. The optimization process is similar to the inner gradient descent in MAML, but it iterates on the unlabeled set x_t rather than on labeled set d_t^l, since it does not rely on y_t to compute the true gradient. The labeled set for an unseen task is now optional, which is only used to generate the initialization in our case. In summary, our main contributions are the following:

1. In section 2 and section 3, we develop a novel empirical Bayes formulation with transduction for meta-learning. To perform amortized variational inference, we propose a parameterization for the variational posterior based on synthetic gradient descent, which incoporates the contextual information from all the inputs of the query set.
2. In section 4, we show in theory that a transductive variational posterior yields better generalization performance. Besides, we show that the proposed empirical Bayes formulation is equivalent to the information bottleneck principle considered by Achille & Soatto (2017). We thus call our method *synthetic information bottleneck* (SIB).
3. In section 6, we verify our proposal empirically. Our experimental results demonstrate that our method significantly outperforms the state-of-the-art meta-learning methods on standard few-shot classification benchmarks.

2 META-LEARNING WITH TRANSDUCTIVE INFERENCE

The goal of meta-learning is to train a *meta-model* on a collection of tasks, such that it works well on another disjoint collection of tasks. Suppose that we are given a collection of N tasks for training. The associated data is denoted by $\mathcal{D} := \{d_t = (x_t, y_t)\}_{t=1}^N$. In the case of few-shot learning, we are given in addition a support set d_t^l for each task. In this section, we revisit the classical empirical Bayes model for meta-learning. Then, we propose to use a transductive scheme in the variational inference by constructing the variational posterior as a function of x_t.

2.1 EMPIRICAL BAYES MODEL

Due to the hierarchical structure among data, it is natural to consider a hierarchical Bayes model for the marginal likelihood

$$p_f(\mathcal{D}) = \int_\psi p(\mathcal{D}|\psi)p(\psi) = \int_\psi \Big[\prod_{t=1}^N \int_{w_t} p_f(d_t|w_t)p(w_t|\psi)\Big]p(\psi). \tag{1}$$

The generative process is illustrated in Figure 1 (left, in solid arrows): first, a meta-parameter ψ is sampled from the *hyper-prior* $p(\psi)$; then, for each task, a *task-specific parameter* w_t is sampled from the *prior* $p(w_t|\psi)$; finally, the dataset is drawn from the *likelihood* $p(d_t|w_t)$[1]. In particular, since different tasks may require different losses, we assume the log-likelihood takes a general form:

$$\log p_f(d_t|w_t) = -\frac{1}{n}\sum_{i=1}^n \ell_t\big(\hat{y}_{t,i}(f(x_{t,i}), w_t), y_{t,i}\big), \tag{2}$$

where ℓ_t denotes the *task-specific loss*, e.g., the cross entropy loss. The first argument in ℓ_t is the prediction, denoted by $\hat{y}_{t,i}$, for the i-th example, which takes as input the *feature representation* $f(x_{t,i})$ and the *task-specific weight* w_t.

Rather than following a fully Bayesian approach, we leave some random variables to be estimated by a frequentist approach, e.g., f is a part of the likelihood model for which we use a point estimate. As such, the posterior inference about these variables will be largely simplified. For the same reason, we derive the *empirical Bayes* (Robbins, 1985; Kucukelbir & Blei, 2014), which interprets ψ in a frequentist way:

$$p_{\psi,f}(\mathcal{D}) = \prod_{t=1}^N p_\psi(d_t) = \prod_{t=1}^N \int_{w_t} p_f(d_t|w_t)p_\psi(w_t). \tag{3}$$

The overall model formulation is the same as the ones considered by Amit & Meir (2018); Grant et al. (2018); Ravi & Beatson (2018).

2.2 AMORTIZED INFERENCE WITH TRANSDUCTION

Focusing on the empirical Bayes model (3), we derive an *evidence lower bound* (ELBO) on the log-likelihood by introducing a variational distribution $q_{\theta_t}(w_t)$ for each task with parameter θ_t:

$$\log p_{\psi,f}(\mathcal{D}) \geq \sum_{t=1}^N \Big[\mathbb{E}_{w_t\sim q_{\theta_t}}\big[\log p_f(d_t|w_t)\big] - D_{\mathrm{KL}}\big(q_{\theta_t}(w_t)\|p_\psi(w_t)\big)\Big]. \tag{4}$$

Maximizing the ELBO in (4) with respect to $\theta_1, \ldots, \theta_N$ and ψ is equivalent to

$$\min_\psi \min_{\theta_1,\ldots,\theta_N} \frac{1}{N}\sum_{t=1}^N D_{\mathrm{KL}}\Big(q_{\theta_t}(w_t) \,\|\, p_f(d_t|w_t)p_\psi(w_t)\Big), \tag{5}$$

However, the optimization in (5), as N increases, becomes more and more expensive in terms of the memory footprint and the computational cost. We therefore wish to bypass this heavy optimization and to take advantage of the fact that individual KL terms indeed share the same structure. To this end, instead of introducing N different variational distributions, we consider a commonly parameterized family of distributions, which is defined implicitly by a deep neural network ϕ taking as input x_t. Note that we do not include y_t as an input because it is not available during meta-testing.

Replacing each q_{θ_t} by $q_{\phi(x_t)}$, (5) can be written as

$$\min_\psi \min_\phi \frac{1}{N}\sum_{t=1}^N D_{\mathrm{KL}}\Big(q_{\phi(x_t)}(w_t) \,\|\, p_f(d_t|w_t)p_\psi(w_t)\Big), \tag{6}$$

which is also known as *amortized* variational inference in the literature (Kingma & Welling, 2013; Rezende et al., 2014). Note that this inference scheme is *transductive* since for testing each point

[1]Note that $\log p_f(d_t|w_t) = \sum_{i=1}^n \log p_f(y_{t,i}|x_{t,i}, w_t) +$ constant for a supervised setting.

Algorithm 1 Variational inference with synthetic gradients for empirical Bayes

Input: the dataset $\mathcal{D}$; the step size η; the number of inner iterations K; pretrained f.
Initialize the meta-models ψ, and $\phi = (\lambda, \xi)$.
while not converged **do**
    Sample a task t and the associated dataset d_t (plus optionally the support set d_t^l).
    Compute the initialization $\theta_t^0 = \lambda$ or $\theta_t^0 = \lambda(d_t^l)$.
    for $k = 1, \ldots, K$ **do**
        Compute θ_t^k via (9).
    end for
    Compute $w_t = w_t(\theta_t^K, \epsilon)$ with $\epsilon \sim p(\epsilon)$.
    Update $\psi \leftarrow \psi - \eta \nabla_\psi D_{\mathrm{KL}}(q_{\theta_t^K(\psi)} \| p_\psi)$.
    Update $\phi \leftarrow \phi - \eta \nabla_\phi D_{\mathrm{KL}}(q_{\phi(x_t)} \| p_f \cdot p_\psi)$.
    Optionally, update $f \leftarrow f + \eta \nabla_f \log p_f(d_t | w_t)$.
end while

in x_t we will use the entire x_t due to the variational posterior $q_{\phi(x_t)}$. Alternatively, we can derive an *inductive* inference scheme by using the support set d_t^l to construct a variational posterior $q_{\phi(d_t^l)}$, since d_t^l and x_t are disjoint. As an example, MAML (Finn et al., 2017) is an inductive method, where $\phi(d_t^l)$ is realized as θ_t^K, the K-th iterate of the stochastic gradient descent

$$\theta_t^{k+1} = \theta_t^k + \eta \nabla_\theta \mathbb{E}_{w_t \sim q_{\theta_t^k}} \Big[\log p(d_t^l | w_t) \Big] \text{ with } \theta_t^0 = \phi. \tag{7}$$

In fact, nothing prevents us to come up with an even better variational posterior $q_{\phi(x_t, d_t^l)}$, shown in dashed arrows in Figure 1 (a), which is again transductive by definition.

In a nutshell, the meta-model includes f, ψ from empirical Bayes and the amortization ϕ for inference. To obtain a closed-form KL term in (6), we restrict ourselves to Gaussian models[2], such that both $q_{\phi(x_t)}$ and p_ψ are Gaussian distributions with diagonal covariance.

3 Variational inference with synthetic gradients

It is however non-trivial to design a network architecture to implement the amortization $\phi(x_t)$ directly since x_t is itself a dataset. The strategy adopted by *neural processes* (Garnelo et al., 2018) is to aggregate the information from all individual examples via a permutation invariant function. However, as pointed out by Kim et al. (2019), such a strategy tends to underfit x_t because the aggregation does not necessarily attain the most relevant information for identifying the task-specific parameter. We instead design a neural network $\phi(x_t)$ to parameterize the optimization process of θ_t. More specifically, consider a stochastic gradient descent on θ_t for optimizing (5) with step size η:

$$\theta_t^{k+1} = \theta_t^k - \eta \nabla_{\theta_t} D_{\mathrm{KL}} \Big(q_{\theta_t^k}(w) \,\|\, p_f(d_t | w) \cdot p_\psi(w) \Big). \tag{8}$$

We would like to parameterize this optimization dynamics up to the K-th step via $\phi(x_t)$, such that $q_{\theta_t^K}$ is a good approximation of the optimum $q_{\theta_t^\star}$. It consists of parameterizing

(a) the **initialization** θ_t^0 and (b) the **gradient** $\nabla_{\theta_t} D_{\mathrm{KL}}(q_{\theta_t} \,\|\, p_f \cdot p_\psi)$.

By doing so, θ_t^K becomes a function of ϕ, ψ and x_t[3], we therefore realize $q_{\phi(x_t)}$ as $q_{\theta_t^K}$.

For (a), we opt to either let $\theta_t^0 = \lambda$ to be a global data-independent initialization as in MAML (Finn et al., 2017) or let $\theta_t^0 = \lambda(d_t^l)$ with a few supervisions from the support set, where λ can be implemented by a permutation invariant network described in Gidaris & Komodakis (2018). In the second case, the features of the support set will be first averaged in terms of their labels and then scaled by a learned vector of the same size.

[2] It is possible to consider more powerful parameterization. For example, implementing the prior $p_\psi(w_t)$ by PixelCNN (Van den Oord et al., 2016) with lossy compression similar to that of VQ-VAE2 (Razavi et al., 2019). We leave that for future work.

[3] θ_t^K is also dependent of f. We deliberately remove this dependency to simplify the update of f.

For (b), the fundamental reason that we parameterize the gradient is because we do not have access to y_t during the meta-testing phase. Note that we are able to follow (8) in meta-training to obtain $q_{\theta_t^\star}(w_t) \propto p_f(d_t|w_t)p_\psi(w_t)$. To make a consistent parameterization in both meta-training and meta-testing, we thus discard y_t when constructing the variational posterior. Regarding the true gradient, a key observation is that, under a reparameterization $w_t = w_t(\theta_t, \epsilon)$ with $\epsilon \sim p(\epsilon)$,

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

where all the terms can be computed without y_t except for $\frac{\partial \ell_t}{\partial \hat{y}_{t,i}}$, thus, we introduce a deep neural network $\xi(\hat{y}_{t,i})$ to synthesize it. The idea of synthetic gradients was originally proposed by Jaderberg et al. (2017) to parallelize the back-propagation. Here, the purpose of $\xi(\hat{y}_{t,i})$ is to update θ_t regardless of the groundtruth labels, which is slightly different from its original purpose. Besides, we do not introduce an additional loss to force $\xi(\hat{y}_{t,i})$ to approximate $\frac{\partial \ell_t}{\partial \hat{y}_{t,i}}$ since $\xi(\hat{y}_{t,i})$ will be learned to yield a reasonable θ_t^K even without mimicking the true gradient.

To sum up, we have derived a particular implementation of $\phi(x_t)$ by parameterizing the ideal mean-field update, namely (8), on the query set d_t, such that the meta-model ϕ includes an initialization network λ and a synthetic gradient network ξ. Specifically, we have $\phi(x_t) = \theta_t^K$, the K-th iterate of the following update:

$$\theta_t^{k+1} = \theta_t^k - \eta\left[\mathbb{E}_\epsilon\Big[\frac{1}{n}\sum_{i=1}^{n}\xi(\hat{y}_{t,i})\frac{\partial \hat{y}_{t,i}}{\partial w_t}\frac{\partial w_t(\theta_t^k,\epsilon)}{\partial \theta_t}\Big] + \nabla_{\theta_t} D_{\mathrm{KL}}\Big(q_{\theta_t^k}\|p_\psi\Big)\right]. \tag{9}$$

The overall algorithm is depicted in Algorithm 1. A comparison with MAML is shown in Figure 1. Rather than viewing (9) as an optimization process, it may be more precise to think of it as a part of the computation graph created in the forward-propagation. As an extension, if we were deciding to estimate the feature network f in a Bayesian manner, we would have to compute the gradient of gradient wrt f in the case of MAML. This is super costly from a computational point of view and needs technical simplifications (Nichol et al., 2018). By introducing a series of synthetic gradient networks in a way similar to Jaderberg et al. (2017), the computation will be decoupled into computations within each layer, and thus becomes more feasible.

4 THEORETICAL ANALYSIS

In this section, we study the generalization ability of the empirical Bayes model.

Empirical Bayes induces a simpler graphical model compared to hierarchical Bayes due to the conditional independence: $w_t \perp\!\!\!\perp \mathcal{D} \setminus d_t \mid d_t$ (see Appendix A for a full discussion). In this case, rather than viewing $w_1, \ldots, w_N$ as N different random variables, we may view $\{(w_t, d_t)\}_{t=1}^N$ as iid samples drawn from $q(w, d) = q(d)q(w|d)$, where $q(d)$ is the underlying data distribution and $q(w|d) = q_{\phi(d)}(w)$, the ideal variational posterior. The data distribution $q(d)$ is in fact a marginal distribution computed by $q(d) = \int_t q(t)q(d|t)$, where t is the random variable representing the task. To simplify the analysis, we do not distinguish samples (i.e. d) drawn from different tasks. We may call $q(w|d)$ *pseudo posterior* since as the family induced by ϕ becomes larger it would have to approximate the true posterior. Correspondingly, we have $q(d|w) = \frac{q(w|d)q(d)}{q(w)}$, the *pseudo likelihood*, where $q(w) = \mathbb{E}_{q(d)}q(w|d)$ is sometimes referred to as *aggregated posterior* (Makhzani et al., 2015; Tomczak & Welling, 2017).

Before going into details, we first introduce a few notations: the *entropy* is defined as $H_p(\mathrm{x}) := \mathbb{E}_{p(\mathrm{x})}[-\log p(\mathrm{x})]$; the *mutual information* is given by $I_p(\mathrm{x};\mathrm{y}) := D_{\mathrm{KL}}\big(p(\mathrm{x},\mathrm{y})\|p(\mathrm{x})p(\mathrm{y})\big)$; the *cross entropy* is defined as $H_{p,q}(\mathrm{x}) := \mathbb{E}_{p(\mathrm{x})}[-\log q(\mathrm{x})]$.

To quantify the generalization ability of the meta-model is equivalent to quantify the generalization of the resulting pseudo posterior. To this end, we first identify the empirical risk for a single task as $L(w, d) := \frac{1}{n}\sum_{i=1}^{n}\ell_t(\hat{y}_i(f(x_i), w), y_i)$. Then, the average empirical risk is the expectation over all possible samples and weights, which depends on the pseudo posterior and the data distribution:

$$\hat{R}(q(w|d), q(d)) := \mathbb{E}_{d\sim q(d)}\mathbb{E}_{w\sim q(w|d)}L(w, d) \tag{10}$$

The true risk should be independent of any particular pseudo posterior, where we sample the task-specific weight from the aggregated posterior:

$$\begin{aligned} R(q(w), q(d)) &:= \mathbb{E}_{w\sim q(w)}\mathbb{E}_{t\sim q(t)}\mathbb{E}_{(x_{t,i},y_{t,i})}\ell_t(\hat{y}_{t,i}(f(x_{t,i}), w), y_{t,i}) \\ &= \mathbb{E}_{w\sim q(w)}\mathbb{E}_{t\sim q(t)}\mathbb{E}_{d\sim q(d|t)}\frac{1}{n}\sum_{t=1}^{n}\ell_t(\hat{y}_i(f(x_i), w), y_i) \\ &= \mathbb{E}_{w\sim q(w)}\mathbb{E}_{d\sim q(d)}L(w, d). \end{aligned} \tag{11}$$

Finally, the generalization error is defined as

$$\text{gen}(q(w|d), q(d)) := R(q(w), q(d)) - \hat{R}(q(w|d), q(d)). \tag{12}$$

Intuitively, the generalization error measures how much the empirical risk concentrated on the true risk is. That is, we would like to bound the tail of the distribution over all possible values of the empirical risk. By taking an expectation of the objective in (6) with respect to the choice of $\mathcal{D}$, we have $\mathbb{E}_{\mathcal{D}}[\frac{1}{N}\sum_{t=1}^{N} D_{\text{KL}}(q(w_t|d_t)\|p(d_t|w_t)p(w_t))] = \mathbb{E}_{d\sim q(d)}D_{\text{KL}}(q(w|d)\|p(d|w)p(w))$. We show in Proposition 1 that the quantity on the right hand side directly affects the generalization performance.

Theorem 1. *Given distributions $q(w|d)$, $q(d)$, $p(w)$ and $p(d|w)$, if $\ell_t(\hat{y}(w), y)$ is σ-subgaussian for all w, the following inequalities hold:*

$$\mathbb{E}_{d\sim q(d)}D_{KL}\Big(q(w|d)\|p(d|w)p(w)\Big) \geq I_q(w; d) + H_{q,p}(d|w) \tag{13}$$

$$\geq \frac{n}{2\sigma^2}gen(q(w|d), q(d))^2 + \hat{R}(q(w|d), q(d));$$

$$\mathbb{E}_{d\sim q(d)}D_{KL}\Big(q(w|d)\|p(w|d)\Big) \leq D_{KL}\Big(q(w, d)\|p(w, d)\Big). \tag{14}$$

The equality in (13) holds if $q(w) = p(w)$. The equality in (14) holds if $q(d) = p(d)$.

The proof of Theorem 1 can be found in Appendix B. The inequality (13) basically says that (6) can be seen a regularized empirical risk minimization in which the regularization term is an upper bound of the mutual information between the weight and the sample. In general, there is a tradeoff between the generalization error and the empirical risk controlled by the coefficient $\frac{n}{2\sigma^2}$. If n is small, then we are in the overfitting regime. This is the case of many existing gradient-based meta-learning methods, where they basically implement the pseudo posterior $q(w|d)$ by $q_{\phi(d^l)}(w)$. Recall that d^l is the support set which is fairly small by the definition of few-shot learning. Consequently, we expect these methods to have large generalization errors in light of the above analysis. On the other hand, if we were following the transductive setting, the sample size n would be larger since we implement the pseudo posterior as $q_{\phi(x)}(w)$ or $q_{\phi(x,d^l)}(w)$.

An obvious message from the inequality (14) is that, if we choose appropriate likelihood, prior and pseudo posterior such that $p(w, d)$ is aligned with $q(w, d)$, the empirical Bayes model can actually be pretty good, that is, both the generalization error and the empirical risk are close to zero. If we look at the message more carefully, it also implies that a good likelihood is the key to represent the data well; the inference model is less important as long as the aggregated posterior coincides with the prior.

Besides, (13) reveals a connection between empirical Bayes and information bottleneck (Tishby et al., 2000; Achille & Soatto, 2017). The right hand side of (13) is exactly the IB objective considered by Achille & Soatto (2017) with a coefficient equal to 1. We find this connection interesting, thus, we call our method *synthetic information bottleneck* (SIB).

5 Experiments: zero-shot regression on spinning lines

To verify the proposed algorithm, we first look at a toy multi-task problem, where we have the full control of the data generation. It is called *zero-shot* since there is no labeled set or support set attached for each task during meta-testing.

Denote by $D_{\text{train}} := \{d_t\}_{t=1}^{N}$ the train set, which consists of datasets of size n: $d = \{(x_i, y_i)\}_{i=1}^{n}$. We construct a dataset d by firstly sampling iid Gaussian random variables as inputs: $x_i \sim \mathcal{N}(\mu, \sigma^2)$. Then, we generate the weight for each dataset by calculating the mean of the inputs and shifting

with a Gaussian random variable ϵ_w: $w = \frac{1}{n}\sum_i x_i + \epsilon_w$, $\epsilon_w \sim \mathcal{N}(\mu_w, \sigma_w^2)$. The output for x_i is $y_i = w \cdot x_i$. We decide ahead of time the hyperparameters $\mu, \sigma, \mu_w, \sigma_w$ for generating x_i and y_i.

Recall that a weighted sum of iid Gaussian random variables is still a Gaussian random variable. Specifically, if $w = \sum_i c_i x_i$ and $x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, then $w \sim \mathcal{N}(\sum_i c_i \mu_i, \sum_i c_i^2 \sigma_i^2)$. Therefore, we have $p(w) = \mathcal{N}(\mu + \mu_w, \frac{1}{n}\sigma^2 + \sigma_w^2)$. On the other hand, if we are given a dataset d of size n, the only uncertainty about w comes from ϵ_w, that is, we should consider x_i as a constant given d. Therefore, the posterior $p(w|d) = \mathcal{N}(\frac{1}{n}\sum_{i=1}^n x_i + \mu_w, \sigma_w^2)$.

Although the data generation is relatively simple, the problem is difficult from a regression point of view. Note that, for each x, there are potentially multiple corresponding y's. In other words, we are solving a one-to-many mapping, which is not a function by definition. We use a simple implementation of SIB to solve this problem. For a task t, the loss ℓ_t is the mean squared error, and the variational posterior is realized by

$$q_{\theta_t^K}(w) = \mathcal{N}(w; \theta_t^K, \sigma_w), \;\; \theta_t^{k+1} = \theta_t^k - 10^{-3}\sum_{i=1}^n x_i \xi(\theta_t^k x_i), \;\text{ and } \theta_t^0 = \lambda. \tag{15}$$

The synthetic gradient network $\xi(y)$ is implemented by a three-layer MLP with hidden size 8. In addition, we let $p_\psi(w)$ to be a Gaussian. Thus, we have three sets of parameters in total: (λ, ξ, ψ). As a comparsion, we implement a Bayesian neural network (BNN) (Blundell et al., 2015) with diagonal Gaussian variational posterior as the baseline, which has exactly twice the number of weights as ξ (since it has the mean and the standard deviation).

In the experiment, we sample 240 tasks respectively for both D_{train} and D_{test}. We learn SIB and BNN on D_{train} for 150 epochs using the ADAM optimizer (Kingma & Ba, 2014), with learning rate set to 10^{-3} and batch size to 8. Other hyperparameters are set as follows: $n = 32, K = 3, \mu = 0, \sigma = 1, \mu_w = 1, \sigma_w = 0.1$. The results are shown in Figure 2. It is clear that both $D_{\text{KL}}(q_{\theta_t^K}(w)\|p(w|d))$ and $D_{\text{KL}}(p_\psi(w)\|p(w))$ are close to zero indicating the success of the learning. In contrast to SIB, the MLP baseline tends to predict the mean value for each x. The learned $\lambda = 0.9771$ captures the mean value as well (since initially $y = \lambda x$), but the dynamics in equation (15) are able to adaptively refine the weights, overall learned in an end-to-end fashion. It is interesting to see how θ_t^k evolves gradually towards the ground truth by varying K (Figure 2 middle).

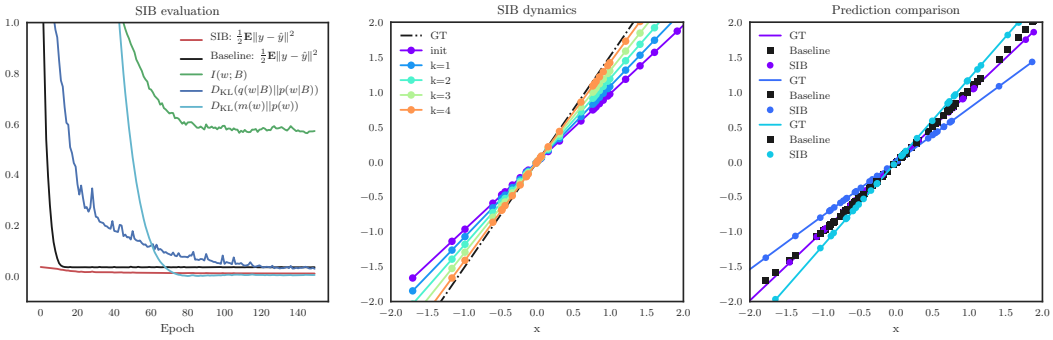


Figure 2: **Left**: the mean-square errors on D_{test}, $D_{\text{KL}}(q_{\theta_t^K}(w)\|p(w|d))$, $D_{\text{KL}}(p_\psi(w)\|p(w))$ and the estimate of $I(w;d) \approx \mathbb{E}_d D_{\text{KL}}(q_{\theta_t^K}(w)\|p_\psi(w))$. **Middle**: the predicted y's by $y = \theta_t^K x$ for $K = 0, \ldots, 4$, which shows how SIB refines θ_t^0 to be closer to the Ground Truth (GT). **Right**: the comparsion between the predicted y's made by SIB and by the baseline respectively.

6 Experiments: few-shot classification

In this section, we present our experimental results on MiniImageNet (Vinyals et al., 2016) and CIFAR-FS (Bertinetto et al., 2018)

6.1 SETUP

Datasets We conduct experiments for few-shot classification on two datasets. Each dataset is composed of disjoint training, validation and testing categories. **MiniImageNet** is proposed by Vinyals et al. (2016), which contains 100 categories, splitted into 64 training classes, 16 validation classes and 20 testing classes, where each category consists of 600 image-label pairs and each image is of size 84×84. **CIFAR-FS** is proposed by Bertinetto et al. (2018), which 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.

Network architectures Following Gidaris & Komodakis (2018); Qiao et al. (2018); Gidaris et al. (2019), we implement f by a 4-layer convolutional network (Conv-4-64 or Conv-4-128[4]) or a WideResNet (WRN-28-10) (Zagoruyko & Komodakis, 2016). We pretrain the feature network $f(\cdot)$ on the 64 training categories for a stardard 64-way classification. We reuse the feature averaging network proposed by Gidaris & Komodakis (2018) as our initialization network $\lambda(\cdot)$, which basically averages the feature vectors of all data points from the same category and then scale each feature dimension differently by a learned coefficient. For the synthetic gradient network $\xi(\cdot)$, we implement a three-layer MLP with hidden-layer size $8k$. Finally, for the predictor $\hat{y}_{ij}(\cdot, w_i)$, we adopt the cosine-similarity based classifier advocated by Chen et al. (2019) and Gidaris & Komodakis (2018).

Evaluation metrics In few-shot classification, a task (aka episode) t consists of a *query set* d_t and a *support set* d_t^l. When we say the task t is k-*way*-n^l-*shot* we mean that d_t^l is formed by first sampling k categories from a pool of categories; then, for each sampled category, n^l examples are drawn and a new label taken from $\{0, \ldots, k-1\}$ is assigned to these examples. By default, each query set contains $15k$ examples. The goal of this problem is to predict the labels of the query set, which are provided as ground truth during training. The evaluation is the average classification accuracy on the tasks created with testing categories. We sample 2000 and 5000 testing tasks respectively for MiniImageNet and CIFAR-FS following Gidaris et al. (2019).

Training details We use ADAM with batch size 8 for 60 epochs, where the learning rate is 10^{-3} and we may optionally dropp the learning rate by a factor 0.1 at epoch 3, 10, 25, 50. During training, we freeze the feature network. To select the best hyper-parameters, for each dataset, we sample 1000 tasks from the validation categories and reuse them at each training epoch. We use these validation tasks to select the best meta-model and then use it for the final evaluation on the tasks sampled from testing categories.

6.2 COMPARISON TO STATE-OF-THE-ART META-LEARNING METHODS

In Table 1, we show a comparison between the state-of-the-art approaches and several variants of our method (varying K or $f(\cdot)$). First of all, comparing SIB ($K = 3$) to SIB ($K = 0$), we observe a clear improvement, which suggests that, by taking a few synthetic gradient steps, we do obtain a better variational posterior as promised.

For 1-shot learning, SIB (when $K = 3$ or $K = 5$) significantly outperforms previous methods on both MiniImageNet and CIFAR-FS. For 5-shot learning, the results are are also comparable. It should be noted that the performance boost is consistently observed with different feature networks, which suggests that SIB is a general method for few-shot learning.

However, we also observe a potential limitation of SIB: when the support set is relatively large, e.g., 5-shot, with a good feature network like WRN-28-10, the initialization θ_t^0 can already be close to a local minima, then if we have relatively large learning rate for the synthetic gradient descent, the result of θ_t^K may yield no improvement over that of θ_t^0.

[4]Conv-4-64 consists of 4 convolutional blocks each implemented with a 3×3 convolutional layer followed by BatchNorm + ReLU + 2×2 max-pooling units. All blocks of Conv-4-64 have 64 feature channels. Conv-4-128 has 64 feature channels in the first two blocks and 128 feature channels in the last two blocks.

Method	Backbone	MiniImageNet, 5-way 1-shot	MiniImageNet, 5-way 5-shot	CIFAR-FS, 5-way 1-shot	CIFAR-FS, 5-way 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 & 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%	78.8±0.4%	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%**	79.2±0.4%	**80.0±0.6%**	85.3±0.4%

Table 1: Average classification accuracies (with 95% confidence intervals) on the test set of MiniImageNet and CIFAR-FS. For evaluation, we sample 2000 and 5000 episodes respectively for MiniImageNet and CIFAR-FS and test three different architectures as the feature extractor: Conv-4-64, Conv-4-128 and WRN-28-10. We report the results for learning rate $\eta = 1e-3$ and different choices of K as well as the performance only using the pre-trained feature (*i.e*, K=0 in the table).

7 Conclusion

We have presented an empirical Bayesian framework for meta-learning. To enable an efficient variational inference, we followed the amortized inference paradigm, and proposed to use a transductive scheme for constructing the variational posterior. To implement the transductive inference, we make use of two neural networks: a synthetic gradient network and an initialization network, which together enables a synthetic gradient descent on the unlabeled data to generate the parameters of the amortized variational posterior dynamically. We have studied the theoretical properties of the proposed framework and shown that it yields significant performance boost on MiniImageNet and CIFAR-FS for few-shot classification.

References

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

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. In *International Conference on Machine Learning*, pp. 205–214, 2018.

Y Bengio, S Bengio, and J Cloutier. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pp. 969–vol. IEEE, 1991.

James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag New York, 1985.

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

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

O Chapelle, B Schölkopf, and A Zien. A discussion of semi-supervised learning and transduction. *Semi-Supervised Learning*, pp. 457–462, 2006.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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*, pp. 1126–1135. JMLR. org, 2017.

Sebastian Flennerhag, Pablo G Moreno, Neil D Lawrence, and Andreas Damianou. Transferring knowledge across learning processes. *International Conference on Learning Representations (ICLR)*, 2019.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

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.

Irving John Good. Some history of the hierarchical bayesian methodology. *Trabajos de estadística y de investigación operativa*, 31(1):489, 1980.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

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*, pp. 1627–1635. JMLR, 2017.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Alp Kucukelbir and David M Blei. Population empirical bayes. *arXiv preprint arXiv:1411.0292*, 2014.

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

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.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

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.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 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*, pp. 7229–7238, 2018.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representation*, 2016.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Herbert Robbins. An empirical bayes approach to statistics. In *Herbert Robbins Selected Papers*, pp. 41–47. Springer, 1985.

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`.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 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. 1998.

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

Jakub M Tomczak and Max Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

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*, pp. 3630–3638. Curran Associates, Inc., 2016.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 2524–2533, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

APPENDIX

A CONDITIONAL INDEPENDENCE IMPLIED BY EB

In the case of the hierarchical Bayes formulation, note that the posterior over the parameter w_t for a given task, namely, $p(w_t|d_t)$, is different from its global counterpart, namely, $p(w_t|\mathcal{D})$. Indeed, $p(w_t|\mathcal{D})$ is coupled by

$$p(w_1, \ldots, w_N|\mathcal{D}) = \frac{\int_\theta \prod_{t=1}^N p(d_t|w_t)p(w_t|\theta)p(\theta)}{p(\mathcal{D})}. \tag{16}$$

Hence, the *global posterior* takes a form of

$$p(w_t|\mathcal{D}) = \frac{p(d_t|w_t) \int_\theta \prod_{k\neq i} p(w_k|\theta)p(d_k|\theta)p(\theta)}{p(\mathcal{D})}. \tag{17}$$

On the other hand, the *local posterior* is given by

$$p(w_t|d_t) = \frac{\int_\theta p(d_t|w_t)p(w_t|\theta)}{\int_\theta \int_{w_t} p(d_t|w_t)p(w_t|\theta)}, \tag{18}$$

which is not equal to $p(w_t|\mathcal{D})$ apparently.

On the other hand, in the empirical Bayes approach, learning is only coupled via the estimation of ψ by the type-II likelihood, because for a fixed choice of ψ, we have

$$p_\psi(w_1, \ldots, w_N|\mathcal{D}) = \frac{\prod_{t=1}^N p(d_t|w_t)p_\psi(w_t)}{\int_{w_1,\ldots,w_N} \prod_{t=1}^N p(d_t|w_t)p_\psi(w_t)} = \frac{\prod_{t=1}^N p(d_t|w_t)p_\psi(w_t)}{\prod_{t=1}^N \int_{w_t} p(d_t|w_t)p_\psi(w_t)} = \prod_{t=1}^N p_\psi(w_t|d_t), \tag{19}$$

which makes the global and local posteriors coincide with each other:

$$p_\psi(w_t|\mathcal{D}) = p_\psi(w_t|d_t) = \frac{p(d_t|w_t)p_\psi(w_t)}{\int_{w_t} p(d_t|w_t)p_\psi(w_t)}. \tag{20}$$

B PROOF OF THEOREM 1

Before going into details, we first introduce a few notations, definitions, and a technical lemma.

- The *entropy* is defined as $H_p(\mathrm{x}) := \mathbb{E}_{p(\mathrm{x})}[-\log p(\mathrm{x})]$;
- The *mutual information* is given by $I_p(\mathrm{x};\mathrm{y}) := D_{\mathrm{KL}}\big(p(\mathrm{x},\mathrm{y})\|p(\mathrm{x})p(\mathrm{y})\big)$.

- Very often, we decompose the mutual information as $I_p(\mathrm{x};\mathrm{y}) = H_p(\mathrm{x}) - H_p(\mathrm{x}|\mathrm{y})$ with $H_p(\mathrm{x}|\mathrm{y}) := \mathbb{E}_{p(\mathrm{x},\mathrm{y})}[-\log p(\mathrm{x}|\mathrm{y})]$ the *conditional entropy*.
- The *cross entropy* is defined as $H_{p,q}(\mathrm{x}) := \mathbb{E}_{p(\mathrm{x})}[-\log q(\mathrm{x})]$. Similarly, we have the *cross conditional entropy* $H_{p,q}(\mathrm{x}|\mathrm{y}) := \mathbb{E}_{p(\mathrm{x},\mathrm{y})}[-\log q(\mathrm{x}|\mathrm{y})]$ and the *cross mutual information* $I_{p,q}(\mathrm{x};\mathrm{y}) = H_{p,q}(\mathrm{x}) - H_{p,q}(\mathrm{x}|\mathrm{y})$.

Definition 1. *A random variable* x *is subgaussian if there exists a positive number* σ *such that* $\mathbb{E}[\exp(\lambda(\mathrm{x} - \mathbb{E}\mathrm{x}))] \le \exp(\sigma^2\lambda^2/2)$ *for all* $\lambda \in \mathbb{R}$.

Lemma 1 (Xu & Raginsky (2017))**.** *If g(*x, y*) is* σ*-subgaussian, then*

$$\left|\mathbb{E}_{p(\mathrm{x},\mathrm{y})}g(\mathrm{x},\mathrm{y}) - \mathbb{E}_{p(\mathrm{x})p(\mathrm{y})}g(\mathrm{x},\mathrm{y})\right| \le \sqrt{2\sigma^2 I_p(\mathrm{x};\mathrm{y})}.$$

Now, we are ready to prove Theorem 1.

Proof. The expected KL divergence can be rewritten as follows:

$$\begin{aligned}
&\mathbb{E}_{q(d)}D_{\mathrm{KL}}\Big(q(w|d)\|p(w|d)\Big) = \mathbb{E}_{q(d)}\mathbb{E}_{q(w|d)}\Big[\log\frac{q(w|d)p(d)q(w)}{p(d|w)p(w)q(w)}\Big] \\
&= \mathbb{E}_{q(d)}\mathbb{E}_{q(w|d)}\Big[\log\frac{q(w|d)}{q(w)}\Big] + \mathbb{E}_{q(d)}\mathbb{E}_{q(w|d)}\Big[-\log p(d|w)\Big] \\
&\quad + \mathbb{E}_{q(w)}\Big[\log\frac{q(w)}{p(w)}\Big] + \mathbb{E}_{q(d)}\mathbb{E}_{q(w|d)}\Big[\log p(d)\Big] \\
&= I_q(w;d) + H_{q,p}(d|w) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) - H_{q,p}(d) \qquad (21)\\
&= I_q(w;d) - I_{q,p}(d;w) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) \\
&\ge I_q(w;d) - I_{q,p}(d;w) = I_q(w;d) - I_{q,p}(w;d). \qquad (22)
\end{aligned}$$

Note that $I_{q,p}(d;w) = I_{q,p}(w;d)$ and $D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) = 0$ iff $q(w) = \mathbb{E}_{q(d)}q(w|d) = p(w)$. Similarly, we have

$$\mathbb{E}_{q(d)}D_{\mathrm{KL}}\Big(q(w|d)\|p(d|w)p(w)\Big) \ge I_q(w;d) + H_{q,p}(d|w). \qquad (23)$$

by removing the term $H_{q,p}(d)$.

Let us rewrite (21) as

$$\begin{aligned}
(21) &= I_q(w;d) + \mathbb{E}_{q(w)}D_{\mathrm{KL}}\big(q(d|w)\|p(d|w)\big) + H_q(d|w) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) - H_{q,p}(d) \\
&= H_q(d) + D_{\mathrm{KL}}\mathbb{E}_{q(w)}\big(q(d|w)\|p(d|w)\big) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) - H_{q,p}(d) \\
&= -D_{\mathrm{KL}}\big(q(d)\|p(d)\big) + \mathbb{E}_{q(w)}D_{\mathrm{KL}}\big(q(d|w)\|p(d|w)\big) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) \\
&\le \mathbb{E}_{q(w)}D_{\mathrm{KL}}\big(q(d|w)\|p(d|w)\big) + D_{\mathrm{KL}}\big(q(w)\big\|p(w)\big) = D_{\mathrm{KL}}\Big(q(w,d)\|p(w,d)\Big), \qquad (24)
\end{aligned}$$

which attains the right hand side of the second inequality.

Recall that $L(w,d) := \frac{1}{n}\sum_{i=1}^n \ell_t(\hat{y}_i(f(x_i),w),y_i)$. If $\ell_t(\hat{y}_i(f(x_i),w),y_i)$ is σ-subgaussian for all w, then $L(w,d)$ is $\sigma/\sqrt{n}$-subgaussian due to the iid assumption on d. Thus, by Lemma 1, we have

$$\big|\mathrm{gen}(q(w|d),q(d))\big| \le \sqrt{\frac{2\sigma^2}{n}I_q(w;d)}.$$

On the other hand, $H_{q,p}(d|w) = \hat{R}(q(w|d),q(d))$. Combining both, we have

$$\frac{n}{2\sigma^2}\mathrm{gen}(q(w|d),q(d))^2 + \hat{R}(q(w|d),q(d)) \le I_q(w;d) + H_{q,p}(d|w)$$

as desired. □

Recall that information bottleneck (IB) involves an optimization

$$\min_{q(w|d)} I_q(w;d) - \beta\, I_{q,p}(w;d) \text{ with } \beta > 0. \qquad (25)$$

Thus, we have established a connection between local empirical Bayes and information bottleneck. The idea of IB is to view w as a compressed representation of d from a *rate-distortion perspective*:

- $I_q(w;d)$ is known as *rate*, which is a regularization term discouraging memorization;
- $-I_{q,p}(w;d) = H_{q,p}(d|w) - H_{q,p}(d)$ is naturally a measure of *distortion* since $H_{q,p}(d|w)$ is equal to the expected negative log-likelihood and $H_{q,p}(d)$ is a constant wrt $q(w|d)$.

Thus, solving (6) amounts to minimizing an upper bound of the IB objective if we generalize the posterior as follows:

$$p_\psi(w|d) = \frac{p(d|w)^\beta p_\psi(w)}{\int_w p(d|w)^\beta p_\psi(w)} \text{ with } \beta > 0. \tag{26}$$

C IMPORTANCE OF SYNTHETIC GRADIENTS

To further verify the effectiveness of the synthetic gradient descent, we implement an inductive version of SIB inspired by MAML, where the initialization θ_t^0 is generated exactly the same way as SIB using $\lambda(d_t^l)$, but it then follows the iterations in (7) as in MAML rather than follows the iterations in (9) as in standard SIB.

We conduct an experiment on CIFAR-FS using Conv-4-64 feature network. The results are shown in Table 2. It can be seen that there is no improvement over SIB ($K = 0$) suggesting that the inductive approach is insufficient.

		inductive SIB		**SIB**			
		Training on 1-shot		**Training on 1-shot**		**Training on 5-shot**	
		Testing on		**Testing on**		**Testing on**	
K	η	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
0	-	59.7±0.5%	75.5±0.4%	59.2±0.5%	75.4±0.4%	59.2±0.5%	75.4±0.4%
1	1e-1	59.8±0.5%	71.2±0.4%	65.3±0.6%	75.8±0.4%	64.5±0.6%	77.3±0.4%
3	1e-1	59.6±0.5%	75.9±0.4%	65.0±0.6%	75.0±0.4%	64.0±0.6%	77.0±0.4%
5	1e-1	59.9±0.5%	74.9±0.4%	66.0±0.6%	76.3±0.4%	64.0±0.5%	76.8±0.4%
1	1e-2	59.7±0.5%	75.5±0.4%	67.8±0.6%	74.3±0.4%	63.6±0.6%	77.3±0.4%
3	1e-2	59.5±0.5%	75.8±0.4%	68.6±0.6%	77.4±0.4%	67.8±0.6%	78.5±0.4%
5	1e-2	59.8±0.5%	75.7±0.4%	67.4±0.6%	72.6±0.6%	67.7±0.7%	77.7±0.4%
1	1e-3	59.5±0.5%	75.6±0.4%	66.2±0.6%	75.7±0.4%	64.6±0.6%	78.1±0.4%
3	1e-3	59.9±0.5%	75.9±0.4%	68.7±0.6%	77.1±0.4%	66.8±0.6%	78.4±0.4%
5	1e-3	59.4±0.5%	75.7±0.4%	69.1±0.6%	76.7±0.4%	66.7±0.6%	78.5±0.4%
1	1e-4	58.8±0.5%	75.5±0.4%	59.0±0.5%	75.7±0.4%	59.3±0.5%	75.7±0.4%
3	1e-4	59.4±0.5%	75.9±0.4%	58.9±0.5%	75.6±0.4%	59.3±0.5%	75.9±0.4%
5	1e-4	59.3±0.5%	75.3±0.4%	60.1±0.5%	76.0±0.4%	60.5±0.5%	76.4±0.4%

Table 2: Average 5-way classification accuracies (with 95% confidence intervals) with Conv-4-64 on the test set of CIFAR-FS. For each test, we sample 2000 episodes containing 5 categories (5-way) and 15 queries in each category. We report the results with using different learning rate η as well as different number of updates K. Note that $K = 0$ is the performance only using the pre-trained feature.