# Empirical Bayes Meta-Learning with Synthetic Gradients

**Shell Xu Hu**[1] **Pablo G. Moreno**[2] **Xi Shen**[1] **Yang Xiao**[1]
**Neil D. Lawrence**[3] **Guillaume Obozinski**[4] **Andreas Damianou**[2]

[1]**École des Ponts ParisTech**
Champs-sur-Marne, France
{xu.hu, xi.shen, yang.xiao}@enpc.fr

[3]**University of Cambridge**
Cambridge, United Kingdom
ndl21@cam.ac.uk

[2]**Amazon**
Cambridge, United Kingdom
{morepabl, lawrennd, damianou}@amazon.com

[4]**Swiss Data Science Center**
Lausanne, Switzerland
guillaume.obozinski@epfl.ch

## Abstract

We revisit the hierarchical Bayes and empirical Bayes formulations for multi-task learning, which can naturally be applied to meta-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 an amortized variational inference that couples all the variational posteriors into a meta-model, which consists of a synthetic gradient network and an initialization network. Our empirical results on the mini-ImageNet benchmark for episodic few-shot classification significantly outperform previous state-of-the-art methods.

## 1 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 $x_t$.

### 1.1 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 $\psi$ 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

[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.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

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 $\ell_t$ denotes the *task-specific loss*, e.g., the cross entropy loss. The first argument in $\ell_t$ 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 and Blei, 2014], which interprets $\psi$ 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 and Meir [2018], Grant et al. [2018], Ravi and Beatson [2018].

### 1.2 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 $\theta_t$:

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

Maximizing the ELBO in (4) with respect to $\theta_1, \ldots, \theta_N$ and $\psi$ 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 $\phi$ 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 $q_{\theta_t}$ by $q_{\phi(x_t)}$, (5) can be written as

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

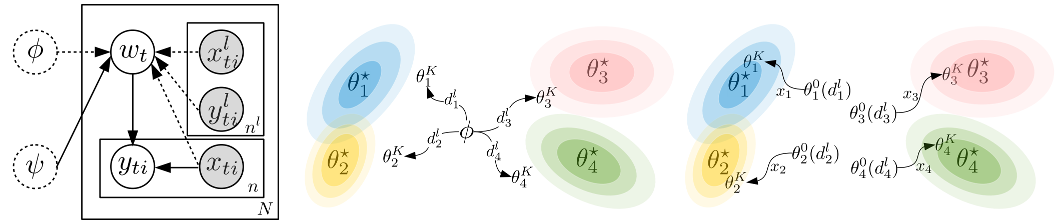
which is also known as *amortized* variational inference in the literature [Kingma and Welling, 2013, Rezende et al., 2014]. Note that this inference scheme is *transductive* since for testing each point in $x_t$ we will use the entire $x_t$ due to the variational posterior $q_{\phi(x_t)}$. Alternatively, 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 $\theta_t^K$, 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 $f, \psi$ from empirical Bayes and the amortization $\phi$ 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 $p_\psi$ are Gaussian distributions with diagonal covariance.

[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.



(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 $\theta_t^K$ by running $K$ synthetic gradient steps on $x_t$.

**Algorithm 1** Variational inference with synthetic gradients for empirical Bayes

```
Input: the dataset $\mathcal{D}$; the step size $\eta$; the number of inner iterations $K$; pretrained $f$.
Initialize the meta-models $\psi$, 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 $k = 1, \dots, K$ do
        Compute $\theta_t^k$ 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
```

## 2 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 $\theta_t$. More specifically, consider a stochastic gradient descent on $\theta_t$ for optimizing (5) with step size $\eta$:

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

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

By doing so, $\theta_t^K$ becomes a function of $\phi$, $\psi$ 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 $\lambda$ can be implemented by a permutation invariant network described in Gidaris and 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.

[3] $\theta_t^K$ 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 $\theta_t$ 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 $\theta_t^K$ 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 $\phi$ includes an initialization network $\lambda$ and a synthetic gradient network $\xi$. 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.

## 3 Few-shot classification on mini-ImageNet

We evaluate our method on the mini-ImageNet dataset, which is an episodic few-shot classification benchmark proposed by Vinyals et al. [2016]. An episode/task $i$ consists of a *query set* $d_i$ and a *support set* $d_i^{\text{supp}}$. When we say an episode $i$ is *$k$-way-$n$-shot* we mean that $d_i^{\text{supp}}$ is formed by first sampling $k$ categories from a pool of categories; then, for each sampled category, $n$ examples are drawn and a new label taken from $\{0, \ldots, k-1\}$ is assigned to these examples. The goal of this problem is to predict the labels of the query set, which are provided as ground truth during training.

The mini-ImageNet dataset contains 100 different categories with 600 images per category, each of size $84 \times 84$ pixels. We used the splits by Ravi and Larochelle [2016] that include $64$ categories to form $\mathcal{D}^{\text{train}}$, $16$ categories to form $\mathcal{D}^{\text{val}}$, and $20$ categories to form $\mathcal{D}^{\text{test}}$.

Following Gidaris and Komodakis [2018], we pretrain the feature network $f(\cdot)$ on $\mathcal{D}^{\text{train}}$ for standard 64-way classification. We also reuse their feature averaging network 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 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 and Komodakis [2018].

There are two types of evaluation: (a) the standard $k$-way few-shot classification proposed by Vinyals et al. [2016] and (b) the learning without forgetting (LwoF) few-shot classification proposed by Gidaris and Komodakis [2018]. We use the same evaluation code provided by Gidaris and Komodakis [2018]. For (b), we additionally evaluate the performance on the $64$ base categories as a $(64+5)$-way classification. In order to classify base categories, we implement $p_\psi$ as a mixture of Gaussians with $64$ components and equal mixing coefficients. The weight of the predictor for classifying base categories are sampled from $p_\psi$. Note that the KL terms can still be computed in closed form.

For training, we use ADAM with batch size 8 for 60 epochs, where the initial learning rate is $10^{-3}$ and dropped by a factor $0.1$ at epoch 10, 25, 50. We use the validation set $\mathcal{D}^{\text{val}}$ to select the best performing model and then use it to test on the test-set $\mathcal{D}^{\text{test}}$.

In Table 1 and Tabel 2 we show a comparison between the state-of-the-art approaches and several variants of our method (varying $T$ or $f(\cdot)$) on $\mathcal{D}^{\text{val}}$ and $\mathcal{D}^{\text{test}}$ respectively. We observe that our methods yield a clear performance boost on novel categories, especially when evaluated on the standard few-shot classification setting. Comparing the cases $T = 0$ and $T = 5$, there are clear $> 4\%$ and $> 10\%$ improvements with CNN feature networks, which becomes even more significant with WRN-28-10 features.

| Methods | 5-way-5-shot | | | 5-way-1-shot | | |
|---|---|---|---|---|---|---|
| | Novel | Base | Both | Novel | Base | Both |
| Vinyals et al. [2016] | 68.87 ± 0.38% | - | - | 55.53 ± 0.48% | - | - |
| Snell et al. [2017] | 72.67 ± 0.37% | 62.10% | 32.70% | 54.44 ± 0.48% | 52.35% | 26.68% |
| Gidaris and Komodakis [2018] | 74.92 ± 0.36% | **70.88%** | 60.50% | 58.55 ± 0.50% | **70.73%** | 50.50% |
| *Standard few-shot classification* | | | | | | |
| Ours $T = 0$ | 73.18 ± 0.34% | - | - | 55.42 ± 0.44% | - | - |
| Ours $T = 1$ | 76.09 ± 0.35% | - | - | 60.74 ± 0.50% | - | - |
| Ours $T = 3$ | 77.53 ± 0.35% | - | - | 65.14 ± 0.54% | - | - |
| Ours $T = 5$ | **77.74 ± 0.36%** | - | - | **66.04 ± 0.59%** | - | - |
| *LwoF few-shot classification* | | | | | | |
| Ours $T = 0$ | 73.13 ± 0.34% | 70.51% | 58.09% | 55.22 ± 0.45% | 70.01% | 47.56% |
| Ours $T = 1$ | 76.69 ± 0.34% | 70.40% | **62.10%** | 61.81 ± 0.50% | 70.09% | 53.53% |
| Ours $T = 3$ | 76.54 ± 0.35% | 69.30% | 60.91% | 63.92 ± 0.54% | 70.19% | **54.89%** |
| Ours $T = 5$ | 76.68 ± 0.35% | 70.28% | 61.93% | 64.39 ± 0.58% | 69.88% | 54.65% |

Table 1: Average classification accuracies on the **validation set** of mini-ImageNet. The "Novel" columns report the average 5-way and 1-shot or 5-shot classification accuracies of novel classes (with $95\%$ confidence intervals), the "Base" and "Both" columns report the classification accuracies of base classes and of both type of classes respectively. In order to report those results we sampled 2000 tasks each with $15 \times k$ test examples of novel classes and $15 \times k$ test examples of base classes.

| Methods | 5-way-5-shot | | | 5-way-1-shot | | |
|---|---|---|---|---|---|---|
| | Novel | Base | Both | Novel | Base | Both |
| Vinyals et al. [2016] | 55.30% | - | - | 43.60% | - | - |
| Ravi and Larochelle [2016] | 60.20 ± 0.71% | - | - | 43.40 ± 0.77% | - | - |
| Finn et al. [2017] | 63.10 ± 0.92% | - | - | 48.70 ± 1.84% | - | - |
| Snell et al. [2017] | 68.20 ± 0.66% | - | - | 49.42 ± 0.78% | - | - |
| Mishra et al. [2017] | 68.88 ± 0.92% | - | - | 55.71 ± 0.99% | - | - |
| Gidaris and Komodakis [2018] | 73.00 ± 0.64% | **70.90%** | 59.35% | 55.95 ± 0.84% | **70.72%** | 49.08% |
| *Standard few-shot classification* | | | | | | |
| Ours $T = 0$ | 71.48 ± 0.64% | - | - | 53.62 ± 0.79% | - | - |
| Ours $T = 1$ | 74.12 ± 0.63% | - | - | 58.74 ± 0.89% | - | - |
| Ours $T = 3$ | 75.43 ± 0.67% | - | - | 62.59 ± 1.02% | - | - |
| Ours $T = 5$ | **75.73 ± 0.71%** | - | - | **63.26 ± 1.07%** | - | - |
| Ours $T = 3$ and $f =$ WRN-28-10 | **78.92 ± 0.37%** | - | - | **67.92 ± 0.55%** | - | - |
| *LwoF few-shot classification* | | | | | | |
| Ours $T = 0$ | 70.93 ± 0.63% | 69.46% | 56.79% | 54.43 ± 0.76% | 69.30% | 47.85% |
| Ours $T = 1$ | 74.42 ± 0.66% | 69.28% | **60.20%** | 60.35 ± 0.88% | 69.10% | 52.52% |
| Ours $T = 3$ | 73.86 ± 0.66% | 68.27% | 58.71% | 62.02 ± 0.93% | 69.45% | **53.52%** |
| Ours $T = 5$ | 74.10 ± 0.67% | 69.06% | 59.74% | 61.82 ± 1.00% | 68.80% | 52.95% |

Table 2: Average classification accuracies on the **test set** of mini-ImageNet. In order to report those results we sampled 600 tasks in a similar fashion as for the validation set of mini-ImageNet (see Table 1).

## References

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

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.

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.

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*, pages 4367–4375, 2018.

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.

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.

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 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.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 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*, pages 41–47. Springer, 1985.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 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*, pages 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*, pages 3630–3638. Curran Associates, Inc., 2016.