MetaModulation: Learning Variational Feature Hierarchies for Few-Shot Learning with Fewer Tasks

Wenfang Sun [*1 2] **Yingjun Du** [*3] **Xiantong Zhen** [4] **Fan Wang** [1] **Ling Wang** [1] **Cees G.M. Snoek** [3]

Abstract

Meta-learning algorithms are able to learn a new task using previously learned knowledge, but they often require a large number of meta-training tasks which may not be readily available. To address this issue, we propose a method for few-shot learning with fewer tasks, which we call MetaModulation. The key idea is to use a neural network to increase the density of the meta-training tasks by modulating batch normalization parameters during meta-training. Additionally, we modify parameters at various network levels, rather than just a single layer, to increase task diversity. To account for the uncertainty caused by the limited training tasks, we propose a variational MetaModulation where the modulation parameters are treated as latent variables. We also introduce learning variational feature hierarchies by the variational MetaModulation, which modulates features at all layers and can consider task uncertainty and generate more diverse tasks. The ablation studies illustrate the advantages of utilizing a learnable task modulation at different levels and demonstrate the benefit of incorporating probabilistic variants in few-task meta-learning. Our MetaModulation and its variational variants consistently outperform state-of-the-art alternatives on four few-task meta-learning benchmarks.

1. Introduction

Learning to learn or *meta-learning* (Schmidhuber, 1987; Thrun & Pratt, 1998), offers a powerful tool for few-shot learning (Andrychowicz et al., 2016; Ravi & Larochelle, 2017; Finn et al., 2017). The crux for few-shot meta-learning is to accrue prior meta-knowledge from a set of meta-training tasks, which enables fast adaptation to a new task with limited data. Despite remarkable achievements of existing meta-learning algorithms for few-shot learning (Finn et al., 2017; Snell et al., 2017; Liu et al., 2022; Hu et al., 2022; He et al., 2022) these works depend on a large number of meta-training tasks during training. However, an extensive collection of meta-training tasks is unlikely to be available for many real-world applications. For example, in medical image diagnosis, a shortage of data samples and tasks arises due to the need for specialist labeling by physicians and patient privacy concerns. Additionally, rare disease types (Wang et al., 2017) present challenges for few-shot learning. In this paper, we focus on few-task meta-learning, where the number of available tasks at training time is limited.

To tackle the few-task meta-learning problem, a variety of task augmentation (Ni et al., 2021; Yao et al., 2021a) and task interpolation (Lee et al., 2022; Yao et al., 2021b) methods have been proposed. The key idea of task augmentation (Ni et al., 2021; Yao et al., 2021a) is to increase the number of tasks from the support set and query set during meta-training. The weakness of these approaches is that they are only able to capture the global task distribution within the distribution of the provided tasks. Task interpolation (Lee et al., 2022; Yao et al., 2021b) generates a new task by interpolating the support and query sets of different tasks by Mixup (Verma et al., 2019) or a neural set function (Lee et al., 2019). Here, a key question is how to combine tasks and at what feature level. For example, the state-of-the-art MLTI by (Yao et al., 2021b) randomly selects the features of a single layer from two known tasks for a linear mixup but ignores all other feature layers for new task generation. It leads to a sub-optimal interpolated task diversity. To address this limitation, we propose a new task modulation strategy that captures the knowledge from one known task at different levels.

One key aspect of task modulation is the ability to leverage the representation of a single task at different levels of abstraction. This allows the model to modulate representa-

[*]Equal contribution [1] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China/P. R. China. [2]University of Science and Technology of China, Hefei 230026, China/P. R. China. [3]University of Amsterdam, Amsterdam, the Netherlands. [4]United Imaging Healthcare, Co., Ltd., China. Correspondence to: Wenfang Sun <swf@mail.ustc.edu.cn>, Yingjun Du <y.du@uva.nl>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

tions of other tasks at varying levels of detail, depending on the specific needs of the new task. Conditional batch normalization (De Vries et al., 2017; Dumoulin et al., 2016; Perez et al., 2018) has been successfully applied to visual question answering and other multi-modal applications. In conditional batch normalization, the normalization parameters (i.e., the scale and shift parameters) are learned from a set of additional input conditions, which can be represented as a set of auxiliary variables or as a separate input branch to the network. This allows the network to adapt to the specific task at hand and improve its performance. Inspired by these general-purpose conditional batch normalization methods, we make in this paper three contributions.

In this paper, we propose a method for few-shot learning with fewer tasks called MetaModulation. It contains three key contributions. First, a meta-training task is randomly selected as a base task, and additional task information is introduced as a condition. We predict the scale and shift of the batch normalization for the base task from the conditional task. This allows the model to modulate the statistics of the conditional task on the base task for a more effective task representation. It is also worth noting that our modulation operates on each layer of the neural network, while previous methods (Yao et al., 2021b; Lee et al., 2022) only select a single layer for modulation. Thus, the model can generate more diverse tasks during meta-training, as it utilizes the statistical information of each level of the conditional task. As a second contribution, we introduce variational task modulation, which treats the conditional scale and shifts as latent variables inferred from the conditional task. The optimization is formulated as a variational inference problem, and new evidence lower bound is derived under the meta-learning framework. In doing so, the model obtains probabilistic conditional scale and shift values that are more informative and better represent the distribution of real tasks. As a third contribution, we propose hierarchical variational task modulation, which obtains the probabilistic conditional scale and shifts at each layer of the network. We cast the optimization as a hierarchical variational inference problem in the Bayesian framework; the inference parameters of the conditional scale and shift are jointly optimized in conjunction with the modulated task training.

To verify our method, we conduct experiments on four few-task meta-learning benchmarks: miniImagenet-S, ISIC, DermNet-S, and Tabular Murris. We perform a series of ablation studies to investigate the benefits of using a learnable task modulation method at various levels of complexity. Our goal is to illustrate the advantages of increasing task diversity through such a method, as well as demonstrate the benefits of incorporating probabilistic variations in the few-task meta-learning framework. Our experiments show that MetaModulation consistently outperforms state-of-the-art few-task meta-learning methods on the four benchmarks.

2. Preliminaries

Problem statement. For the traditional few-shot meta-learning problem, we deal with tasks $\mathcal{T}_i$, as sampled from a task distribution $p(\mathcal{T})$. We sample N-way k-shot tasks from the meta-training tasks, where k is the number of labeled examples for each of the N classes. Each t-th task includes a support set $\mathcal{S}^t{=}\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N\times k}$ and query set $\mathcal{Q}^t{=}\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{m}$ ($\mathcal{S}^t, \mathcal{Q}^t \subseteq \mathcal{X}$). Given a learning model f_ϕ, where ϕ denotes the model parameters, few-shot learning algorithms attempt to learn ϕ to minimize the loss on the query set $\mathcal{Q}_i$ for each of the sampled tasks using the data-label pairs from the corresponding support set $\mathcal{S}_i$. After that, during the testing stage, the trained model f_ϕ and the support set $\mathcal{S}_j$ for new tasks $\mathcal{T}_j$ perform inference and evaluate performance on the corresponding query set $\mathcal{Q}_j$. In this paper, we focus on *few-task* meta-learning. In this setting, the main challenge is that the number of meta-training tasks $\mathcal{T}_i$ is limited, which causes the model to overfit easily.

Prototype-based meta-learning. We develop our method based on the prototypical network (ProtoNet) by Snell et al. (2017). Specifically, ProtoNet leverages a non-parametric classifier that assigns a query point to the class having the nearest prototype in the learned embedding space. The prototype $\mathbf{c}_k$ of an object class c is obtained by: $\mathbf{c}_k{=}\frac{1}{K}\sum_k f_\phi(\mathbf{x}_{c,k})$, where $f_\phi(\mathbf{x}_{c,k})$ is the feature embedding of the sample $\mathbf{x}_{c,k}$, which is usually obtained by a convolutional neural network. For each query sample $\mathbf{x}^q$, the distribution over classes is calculated based on the softmax over distances to the prototypes of all classes in the embedding space:

$$p(\mathbf{y}_n^q = k|\mathbf{x}^q) = \frac{\exp(-d(f_\phi(\mathbf{x}^q), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\phi(\mathbf{x}^q), \mathbf{c}_{k'}))}, \tag{1}$$

where $\mathbf{y}^q$ denotes a random one-hot vector, with $\mathbf{y}_c^q$ indicating its n-th element, and $d(\cdot,\cdot)$ is some (Euclidean) distance function. Due to its non-parametric nature, the ProtoNet enjoys high flexibility and efficiency, achieving considerable success in few-shot learning.

Conditional batch normalization. The aim of Batch Normalization (Ioffe & Szegedy, 2015) is to accelerate the training of deep networks by reducing internal covariate shifts. For a layer with d-dimensional input $x{=}(x^{(1)}...x^{(d)})$ and activation $x^{(k)}$, batch normalization normalizes each scalar feature as follows:

$$\mathbf{y}^{(k)} = \gamma^{(k)} \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}] + \epsilon}} + \beta^{(k)}, \tag{2}$$

where ϵ is a constant added to the variance for numerical stability. $\gamma^{(k)}$ and $\beta^{(k)}$ are the scale and shift for batch normalization. Conditional batch normalization (CBN) (De Vries et al., 2017) is a class-conditional variant of conventional

batch normalization. The key idea of CBN is to predict the transformation parameters γ and β from a conditional embedding (*i.e.*, a language embedding). CBN enables the language embedding to manipulate the entire vision feature map by scaling them up or down, negating them, or shutting them off completely. Specifically, CBN uses two feed-forward multi-layer perceptrons (MLPs) to predict these changes instead of predicting the original transformations, which benefits training stability:

$$\Delta\beta = \texttt{MLP}(\mathbf{e}_q) \qquad \Delta\gamma = \texttt{MLP}(\mathbf{e}_q), \tag{3}$$

where e_q is an additional language embedding. So, given a feature map with C channels, these MLPs output a vector of size C. They then add these changes to the β and γ parameters:

$$\hat{\beta}_c = \beta_c + \Delta\beta_c \qquad \hat{\gamma}_c = \gamma_c + \Delta\gamma_c. \tag{4}$$

Finally, the updated $\hat{\beta}$ and $\hat{\gamma}$ are used as transformation parameters for the batch normalization (eq. (2)) of vision features. Rather than using a language embedding for the conditioning, we randomly select one additional task as a condition to predict the scale and shift of the batch normalization for another task.

Meta-learning task interpolation. Several methods (Yao et al., 2021b; Lee et al., 2022) have been suggested as ways to increase the diversity of the tasks used for meta-training. MLTI (Yao et al., 2021b) generates additional tasks by randomly sampling a pair of tasks and interpolating the corresponding features and labels using Manifold Mixup (Verma et al., 2019). Specifically, given examples from class n in task $\mathcal{T}_i$ and class n' in task $\mathcal{T}_j$, the interpolated features are defined as:

$$\hat{\mathbf{H}}_n^{s,l} = \lambda\mathbf{H}_{i;n}^{s,l} + (1-\lambda)\mathbf{H}_{j;n'}^{s,l}, \tag{5}$$

$$\hat{\mathbf{H}}_n^{q,l} = \lambda\mathbf{H}_{i;n}^{q,l} + (1-\lambda)\mathbf{H}_{j;n'}^{q,l}, \tag{6}$$

where l indicates the l-th layer ($0 \leq l \leq L$), and $\lambda \in [0,1]$ is sampled from a Beta distribution $\mathrm{Beta}(\alpha,\beta)$. The interpolated support samples $\hat{\mathbf{H}}_{cn;n}^{s,l}$ and query samples $\hat{\mathbf{H}}_{cn;n}^{q,l}$ can be regarded as the new classes in the interpolated task. However, MLTI (Yao et al., 2021b) randomly selects only the features of a single layer from two known tasks to be mixed and ignores all the other feature layers. It leads to the interpolated task's diversity being limited and therefore does not increase the generalizability of the model.

3. MetaModulation

In this paper, we propose MetaModulation for few-task meta-learning. We first introduce meta task modulation in section 3.1. To obtain more diverse meta-training tasks, we then propose variational task modulation in section 3.2,

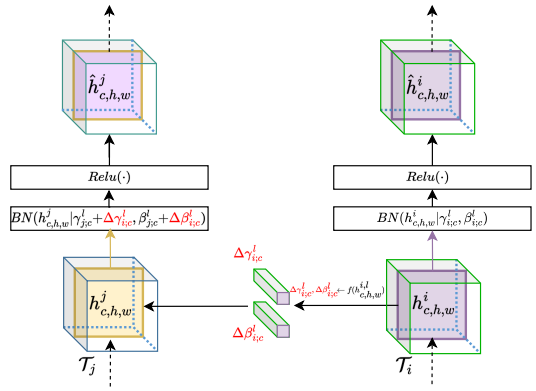

Figure 1. **Meta task modulation.** Various combinations of the transformation parameters γ and β from task $\mathcal{T}_i$ can modulate the individual activation of task $\mathcal{T}_j$ at different layers, which can make the newly generated task more diverse.

which introduces variational inference into the modulation. We also introduce hierarchical meta variational modulation in section 3.3, which adds variational modulation to each network layer to obtain a richer task distribution.

3.1. Meta task modulation

To address the single layer limitation in MLTI (Yao et al., 2021b), we introduce meta task modulation for few-task meta-learning, which modulates the features of two different tasks at different layers. We modulate all layers of samples from a meta-training task $\mathcal{T}_j$ by predicting the γ and β of the batch normalization from base task $\mathcal{T}_i$. Following CBN (De Vries et al., 2017), we only predict the change $\Delta\beta_c$ and $\Delta\gamma_c$ on the original scalars from the task $\mathcal{T}_i$, which benefits training stability.

Specifically, to infer the conditional scale and shift $\Delta\beta_c$ and $\Delta\gamma_c$, we deploy two functions $f_\beta^l(\cdot)$ and $f_\gamma^l(\cdot)$ that take the activations $\mathbf{H}_{i;n}^l$ from task $\mathcal{T}_i$ as input, and the output are $\Delta\beta_{i;n;c}^l$ and $\Delta\gamma_{i;n;c}^l$. The functions $f_\beta^l(\cdot)$ and $f_\gamma^l(\cdot)$ are parameterized by two feed-forward multi-layer perceptrons:

$$\Delta\beta_{i;n;c}^{s,l} = \texttt{MLP}(\mathbf{H}_{i;n}^{s,l}) \qquad \Delta\gamma_{i;n;c}^{s,l} = \texttt{MLP}(\mathbf{H}_{i;n}^{s,l}) \tag{7}$$

where $\Delta\gamma_{i;n;c}^{s,l}$ and $\Delta\gamma_{i;n;c}^{s,l}$ are the changes of the support set. We obtain $\Delta\gamma_{i;n;c}^{q,l}$ and $\Delta\gamma_{i;n;c}^{q,l}$ of the query set by the same strategy. Note that the functions $f_\beta^l(\cdot)$ and $f_\gamma^l(\cdot)$ are shared by different channels in same layer and we learn L pairs of those functions if we have L convolutional layers.

Using the above functions, we generate the changes for the batch normalization scale and shift, then following eq. (4), we add these changes to the original $\beta_{j;n;c}^l$ and $\gamma_{j;n;c}^l$ from task $\mathcal{T}_j$:

$$\hat{\beta}_{j;n;c}^{s,l} = \beta_{j;n;c}^{s,l} + \Delta\beta_{i;n;c}^{s,l} \qquad \hat{\gamma}_{j;n;c}^{s,l} = \gamma_{j;n;c}^{s,l} + \Delta\gamma_{i;n;c}^{s,l} \tag{8}$$

Once we obtain the modulated scale $\hat{\gamma}_{i;n;c}^l$ and shift $\hat{\beta}_{i;n;c}^l$,

we compute the modulated features for the support and query set from task $\mathcal{T}_j$ based on eq. (2):

$$\hat{\mathbf{H}}_n^{s,l} = \hat{\gamma}_{i;n;c}^{s,l} \frac{\mathbf{H}_{j;n}^{s,l} - \mathbb{E}[\mathbf{H}_{j;n}^{s,l}]}{\sqrt{\mathrm{Var}[\mathbf{H}_{j;n}^{s,l}] + \epsilon}} + \hat{\beta}_{j;n;c}^{s,l}, \tag{9}$$

$$\hat{\mathbf{H}}_n^{q,l} = \hat{\gamma}_{i;n;c}^{q,l} \frac{\mathbf{H}_{j;n}^{q,l} - \mathbb{E}[\mathbf{H}_{j;n}^{q,l}]}{\sqrt{\mathrm{Var}[\mathbf{H}_{j;n}^{q,l}] + \epsilon}} + \hat{\beta}_{j;n;c}^{q,l}, \tag{10}$$

where $\mathbb{E}[\mathbf{H}_{i;n}^l]$ and $\mathrm{Var}[\mathbf{H}_{i;n}^l]$ are the mean and variance of samples features from $\mathcal{T}_j$. We illustrate the meta task modulation process in Figure 1.

However, the deterministic conditional scale and shift are not sufficiently representative of modulated tasks. Moreover, uncertainty is inevitable due to the scarcity of data and tasks, which should also be encoded into the conditional scale and shift. In the next section, we derive a probabilistic latent variable model by modeling conditional scale and shift as distributions, which we learn by variational inference.

3.2. Variational task modulation

In this section, we introduce variational task modulation using a latent variable model in which we treat the conditional scale $\Delta\beta_{i;n;c}^{s,l}$ and shift $\Delta\gamma_{i;n;c}^{s,l}$ as latent variables $\mathbf{z}$ inferred from one known task. We formulate the optimization of variational task modulation as a variational inference problem by deriving a new evidence lower bound (ELBO) under the meta-learning framework.

From a probabilistic perspective, the conditional latent scale and shift maximize the conditional predictive log-likelihood from two known tasks $\mathcal{T}_i, \mathcal{T}_j$.

$$\begin{aligned}
&\max_p \log p(\hat{\mathbf{y}}|\mathcal{T}_i, \mathcal{T}_j) \\
&= \max_p \log \int p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s) p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i, \mathcal{T}_j) \mathrm{d}\hat{\mathbf{x}}^q \mathrm{d}\hat{\mathbf{x}}^s \\
&= \max_p \log \int p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s) p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}, \mathcal{T}_j) p(\mathbf{z}|\mathcal{T}_i) \mathrm{d}\mathbf{z} \mathrm{d}\hat{\mathbf{x}}^q \mathrm{d}\hat{\mathbf{x}}^s
\end{aligned} \tag{11}$$

where $\hat{\mathbf{x}}^s, \hat{\mathbf{x}}^q$ are the support sample and query sample of the modulated task $\mathcal{T}$. Since $p(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i, \mathcal{T}_j) = p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}, \mathcal{T}_j) p(\mathbf{z}|\mathcal{T}_i)$ is generally intractable, we resort to a variational posterior $q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_j)$ for its approximation. We obtain the variational distribution by minimizing the Kullback-Leibler (KL) divergence:

$$D_{\mathrm{KL}}[q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_j)||p(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i, \mathcal{T}_j). \tag{12}$$

By applying the Baye's rule to the posterior $q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i)$, we derive the ELBO as:

$$\begin{aligned}
\log p(\hat{\mathbf{y}}|\mathcal{T}_i, \mathcal{T}_j) \geq & \mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} [\log p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)] \\
& - D_{\mathrm{KL}} [q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_j)||p(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i, \mathcal{T}_j)]
\end{aligned} \tag{13}$$

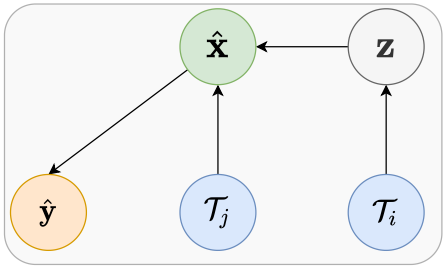

Figure 2. **Variational task modulation.** $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ denote the sample and label of newly generated task $\hat{\mathcal{T}}$ and $\mathbf{z}$ represents the latent modulation parameters.

The second term in the ELBO can also be simplified. Since

$$\begin{aligned}
& D_{\mathrm{KL}} [q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)|\mathcal{T}_i||p(\mathbf{z}, \hat{\mathbf{z}}|\mathcal{T}_i, \mathcal{T}_j)] \\
& = \mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} \log \frac{q(\mathbf{z}, \hat{\mathbf{x}}|\mathcal{T}_i)}{p(\mathbf{z}, \hat{\mathbf{x}}|\mathcal{T}_i, \mathcal{T}_j)},
\end{aligned} \tag{14}$$

and

$$q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_j) = p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}, \mathcal{T}_j) q(\mathbf{z}), \tag{15}$$

we then combine eq. (14), eq. (15) and eq. (11), to obtain:

$$\begin{aligned}
& \mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} \log \frac{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_j)}{p(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i, \mathcal{T}_j)} \\
& = \mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} \log \frac{p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}, \mathcal{T}_i) q(\mathbf{z})}{p(\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}, \mathcal{T}_i) p(\mathbf{z}|\mathcal{T}_i)} \\
& = \mathbb{E}_{q(\mathbf{z})} \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathcal{T}_i)} \\
& = D_{\mathrm{KL}} [q(\mathbf{z})||p(\mathbf{z}|\mathcal{T}_i)] .
\end{aligned} \tag{16}$$

This provides the final ELBO for the variational task modulation:

$$\begin{aligned}
q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i) \geq & \mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} [\log p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)] \\
& - D_{\mathrm{KL}} [q(\mathbf{z})||p(\mathbf{z}|\mathcal{T}_i)]
\end{aligned} \tag{17}$$

The overall computation graph of variational task modulation is shown in Figure 2.

Directly optimizing the above objective does not take into account the task information of all model layers, since it only focuses on the conditional latent scale and shift at a specific layer. Thus, we introduce hierarchical variational inference into the variational task modulation by conditioning the posterior on both the known tasks and the conditional latent scale and shift from the previous layers.

3.3. Hierarchical variational task modulation

We replace variational distribution in eq. (12) with a new conditional distribution $q(\mathbf{z}^l, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}^{l-1}, \mathcal{T}_j)$ that makes latent scale and shift of current l-th layer also dependent on the latent scale and shift from the upper $l-1$-th layers.

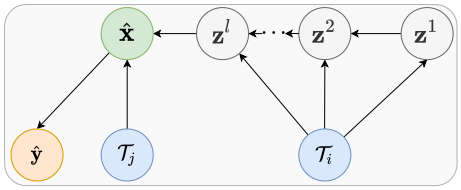

Figure 3. **Hierarchical variational task modulation.** $\mathbf{z}^l$ indicates the latent modulation parameters at the layer l. The latent transformation parameter $\mathbf{z}^l$ is depend on the task $\mathcal{T}_i$ and the upper $\mathbf{z}^{l-1}$.

The hierarchical variational inference gives rise to a new ELBO, as follows:

$$q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathcal{T}_i) \geq \mathbb{E}_{q(\mathbf{z}^l, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}^{l-1})} \left[\log p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)\right] - D_{\mathrm{KL}} \left[q(\mathbf{z}^l|\mathbf{z}^{l-1})||p(\mathbf{z}^l|\mathbf{z}^{l-1}, \mathcal{T}_i)\right] \tag{18}$$

The graphical model of hierarchical variational task modulation is shown in Figure 3.

In practice, the prior $p(\mathbf{z}^l|\mathbf{z}^{l-1}, \mathcal{T}_i)$ is implemented by an amortization network (Kingma & Welling, 2013) that takes the concatenation of the average feature representations of samples in the support set from $\mathcal{T}_i$ and the upper layer latent scale and shift $\mathbf{z}^{l-1}$ and returns the mean and variance of the current layer latent scale and shift $\mathbf{z}^l$. To enable back-propagation with the sampling operation during training, we adopt the reparametrization trick (Rezende et al., 2014; Kingma & Welling, 2013) as $\mathbf{z}{=}\mathbf{z}_\mu + \mathbf{z}_\sigma \odot \boldsymbol{\epsilon}$, where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$. The hierarchical probabilistic scale and shift provide a more informative task representation than the deterministic meta task modulation and have the ability to capture different representation levels, thus modulating more diverse tasks for few-task meta-learning.

In the meta-training stage, we use the known meta-training tasks $\mathcal{T}_i$ with our meta task modulation and its variational variants to generate the new tasks $\hat{\mathcal{T}}$ for the meta-training. To ensure that the original tasks are also trained together, we train the generated tasks together with the original tasks. Thus the loss function of our meta task modulation $\mathcal{L}_{\mathrm{MTM}}$ is as follows:

$$\mathcal{L}_{\mathrm{MTM}} = \frac{1}{T}\sum_{i}^{T}\Big(\sum_{(\hat{\mathcal{S}}_i, \hat{\mathcal{Q}}_i)\sim\hat{\mathcal{T}}_i} \mathcal{L}_{\mathrm{CE}} + \lambda \sum_{(\mathcal{S}_i, \mathcal{Q}_i)\sim\mathcal{T}_i} \mathcal{L}_{\mathrm{CE}}\Big). \tag{19}$$

The loss of variational task modulation $\mathcal{L}_{\mathrm{VTM}}$ is

$$\mathcal{L}_{\mathrm{VTM}} = \frac{1}{T}\sum_{i,j}^{T}\Big(\sum_{(\hat{\mathbf{x}}^q, \hat{\mathbf{y}})\in\hat{\mathcal{Q}}} -\mathbb{E}_{q(\mathbf{z}, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)} \left[\log p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)\right] + \beta D_{\mathrm{KL}} \left[q(\mathbf{z})||p(\mathbf{z}|\mathcal{T}_i)\right]\Big) + \lambda\frac{1}{T}\sum_{i}^{T}\sum_{(\mathcal{S}_i, \mathcal{Q}_i)\sim\mathcal{T}_i} \mathcal{L}_{\mathrm{CE}}. \tag{20}$$

And the loss of hierarchical variational task modulation can be written as

$$\mathcal{L}_{\mathrm{HVTM}} = \frac{1}{T}\sum_{i,j}^{T}\Big(\sum_{(\hat{\mathbf{x}}^q, \hat{\mathbf{y}})\in\hat{\mathcal{Q}}} -\mathbb{E}_{q(\mathbf{z}^l, \hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s|\mathbf{z}^{l-1})} \left[\log p(\hat{\mathbf{y}}|\hat{\mathbf{x}}^q, \hat{\mathbf{x}}^s)\right] - \beta D_{\mathrm{KL}} \left[q(\mathbf{z}^l|\mathbf{z}^{l-1})||p(\mathbf{z}^l|\mathbf{z}^{l-1}, \mathcal{T}_i)\right]\Big) + \lambda\frac{1}{T}\sum_{i}^{T}\sum_{(\mathcal{S}_i, \mathcal{Q}_i)\sim\mathcal{T}_i} \mathcal{L}_{\mathrm{CE}}, \tag{21}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss,

$$\mathcal{L}_{\mathrm{CE}} = \frac{1}{N_C N_{\mathcal{Q}}}\left[d(f_\phi(x^q), c_k) + \log\sum_{k'}\exp(-d(f_\phi(x^q), c_k))\right], \tag{22}$$

N_C and $N_{\mathcal{Q}}$ are the number of prototypes and query samples in each task, and $\lambda > 0$ and $\beta > 0$ are the regularization hyper-parameters.

In the meta-test stage, we directly input the support set $\mathcal{S}$ using the meta-trained feature extractor $f_\phi(\cdot)$ to obtain the prototype c_k from the test task. Then we obtain the prediction of the query set $\mathbf{x}^q$ for performance evaluation based on eq. (1).

4. Experiments

4.1. Experimental setup

Datasets. We conduct experiments on four few-task meta-learning challenges, *i.e.*, miniImagenet, ISIC, DermNet and Tabular Murris (Cao et al., 2020). miniImagenet (Vinyals et al., 2016) is constructed from ImageNet (Deng et al., 2009) and comprises a total of 100 different classes (each with 600 instances). All images are downsampled to 84 $\times$ 84. We follow (Yao et al., 2021b) and reduce the number of tasks by limiting the number of meta-training classes to obtain miniImagenet-S, with 12 meta-training classes and 20 meta-test classes. ISIC (Milton, 2019) aims to classify dermoscopic images among nine different diagnostic categories. 10,015 images are available for training across 8 different categories. We select 4 categories as the meta-training classes. DermNet is one of the largest open resources of images of skin diseases, with more than 23,000 images. Following (Yao et al., 2021b), we construct Dermnet-S, which selects 30 diseases as the meta-training classes. Tabular Murris considers cell type classification across organs and contains nearly 100,000 cells from 20 organs and tissues. Following (Yao et al., 2021b), we choose 57 base classes as the meta-training classes. For our ablation studies we report on miniImagenet-S, ISIC and Dermnet-S, for our comparison with the state-of-the-art, we also consider Tabular Murris. Sample images from all datasets are provided in the appendix.

Implementation details. For miniImagenet-S, ISIC, DermNet-S and Tabular Murris, we follow (Yao et al.,

	miniImagenet-S		ISIC		Dermnet-S	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Vanilla	36.26	50.72	58.56	66.25	44.21	60.33
MTM	**42.44**	**56.25**	**63.13**	**74.23**	**49.46**	**66.12**

Table 1. **Benefit of meta task modulation** in (%) on three few-task meta-learning challenges. Our meta task modulation (MTM) achieves better performance compared to a vanilla ProtoNet.

	Network layer					
	1st	2nd	3rd	4th	random	**All (HVTM)**
5-way 1-shot						
MTM	41.30	41.32	41.31	39.47	39.98	42.44
VTM	41.25	42.05	41.63	39.97	40.91	**43.21**
5-way 5-shot						
MTM	54.21	54.30	54.13	52.62	53.32	56.25
VTM	54.47	55.82	54.36	52.80	54.43	**57.26**

Table 2. **Benefit of variational task modulation** for varying layers on miniImageNet-S. Variational task modulation (VTM) improves over any of the selected individual layers using MTM.

2021b) using a network containing four convolutional blocks and a classifier layer. Each block comprises a 32-filter 3×3 convolution, a batch normalization layer, a ReLU nonlinearity, and a 2×2 max pooling layer. We train a ProtoNet (Snell et al., 2017) using Euclidean distance in the 1-shot and 5-shot scenarios with training episodes. Each image is re-scaled to the size of $84 \times 84 \times 3$. For all experiments, we use an initial learning rate of 10^{-3} and an SGD optimizer with Adam (Kingma & Ba, 2014). The variational neural network is parameterized by three feed-forward multiple-layer perception networks and a ReLU activation layer. The number of Monte Carlo samples is 20. The batch and query sizes of all datasets are set as 4 and 15. The total training iterations are 50,000. The average few-task meta-learning classification accuracy (%, top-1) is reported across all test images and tasks. Code available at: `https://github.com/lmsdss/MetaModulation`.

4.2. Results

Benefit of meta task modulation. To show the benefit of meta task modulation, we first compare our method with a vanilla Prototypical network (Snell et al., 2017) on all tasks, without using task interpolation, in Table 1. Our model performs better under various shot configurations on all few-task meta-learning benchmarks. We then compare our model with the state-of-the-art MLTI (Yao et al., 2021b) in Table 5, which interpolates the task distribution by Mixup (Verma et al., 2019). Our meta task modulation also compares favorably to MLTI under various shot configurations. On ISIC, for example, we surpass MLTI by 2.71% on the 5-way 5-shot setting. This is because our model can learn how to modulate the base task features to better capture the task distribution instead of using linear interpolation as described in the (Yao et al., 2021b).

Benefit of variational task modulation. We investigate the benefit of variational task modulation by comparing it

	miniImagenet-S		ISIC		DermNet-S	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
VTM	42.05	55.82	64.04	72.59	49.19	64.62
HVTM	**43.21**	**57.26**	**65.16**	**76.40**	**50.45**	**67.05**

Table 3. **Hierarchical vs. flat variational modulation.** Hierarchical variational task modulation (HVTM) is more effective than flat variational task modulation (VTM) for few-task meta-learning.

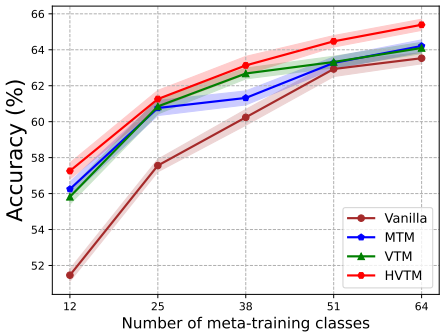

Figure 4. **Influence of the number of meta-training tasks** for 5-way 5-shot on miniImageNet. All MetaModulation implementations improve over a vanilla prototype network, especially when fewer tasks are available for meta-learning. Where a vanilla network requires 64 tasks to reach 63.7% accuracy, we need 40.

with deterministic meta task modulation. The results are reported on miniImageNet-S under various shots in Table 2. $\{1^{st}, 2^{nd}, 3^{rd}, 4^{th}\}$, random and, all are the selected determined layer, the randomly chosen one layer and all the layers to be modulated, respectively. The variational task modulation consistently outperforms the deterministic meta task modulation on any selected layers, demonstrating the benefit of probabilistic modeling. By using probabilistic task modulation, the base task can be modulated in a more informative way, allowing it to encompass a larger range of task distributions and ultimately improve performance on the meta-test task.

Hierarchical vs. flat variational task modulation. We compare hierarchical modulation with flat variational modulation, which only selects one layer to modulate. As shown in Table 3, the hierarchical variational modulation improves the overall performance under both the 1-shot and 5-shot settings on all three benchmarks. The hierarchical structure is well-suited for increasing the density of the task distribution across different levels of features, which leads to better performance compared to flat variational modulation. This makes sense because the hierarchical structure allows for more informative transformations of the base task, enabling it to encompass a broader range of task distributions. Note that, we use hierarchical variational task modulation to compare the state-of-the-art methods in the subsequent experiments.

Influence of the number of meta-training tasks. In Figure 4, we analyze the effect of the number of available meta-

	mini → Dermnet		Dermnet → mini	
	1-shot	5-shot	1-shot	5-shot
Vanilla	33.12	50.13	28.11	40.35
MLTI	35.46	51.79	30.06	42.23
ATA	35.83±0.58	51.65±0.6	-	-
This paper	**37.15**±0.75	**53.92** ± 1.01	**31.56** ± 0.68	**44.13** ± 0.92

Table 4. **Cross-domain adaptation ability.** MetaModulation achieves better performance even in a challenging cross-domain adaptation setting compared to a vanilla prototype network and MLTI by Yao et al. (2021b).

training tasks on the performance of our model under a 5-shot setting on miniImageNet-S. Naturally, our model's performance improves, as the number of meta-training classes increases. The number of meta-training tasks is important for making the model more generalizable through meta-learning. More interesting, our model's performance is considerably improved by using a learnable modulation that incorporates information from different levels of the task. Compared to the best result of a vanilla prototype network, 63.7% for 64 meta-training classes, we can reduce the number of classes to 40 for the same accuracy.

Cross-domain adaptation ability. To further evaluate the effectiveness of our proposed method, we conducted additional tests to assess the performance of MetaModulation in cross-domain adaptation scenarios. We trained MetaModulation on one source domain and then evaluated it on a different target domain. Specifically, we chose the miniImagenet-S and Dermnet-S domains. The results, as shown in Table 4, indicate MetaModulation generalizes better even in this more challenging scenario.

Analysis of modulated tasks. To understand how our MetaModulation is able to improve performance, we plotted the similarity between the vanilla, interpolated and modulated tasks and the meta-test tasks in Figure 5. Red numbers indicate the accuracy per model on each task. Specifically, we select 4 meta-test tasks and 300 meta-train tasks per model from the 1-shot miniImagenet-S setting to compute the task representation of each model. We then used instance pooling to obtain the representation of each task. Instance pooling involves combining a task's support and query sets and averaging the feature vectors of all instances to obtain a fixed-size prototype representation. This approach allows us to represent each task by a single vector that captures the essence of the task. We calculated the similarity between meta-train and meta-test tasks using Euclidean distance. When using the vanilla prototype model (Snell et al., 2017) directly, the similarity between meta-train and meta-test tasks is extremely low, indicating a significant difference in task distribution between meta-train and meta-test. This results in poor performance as seen in Figure 5 red numbers due to the distribution shift. However, the tasks modulated by our MetaModulation have a higher similarity with the meta-test tasks compared to the vanilla (Snell et al., 2017) and MLTI (Yao et al., 2021b), resulting in high accuracy.

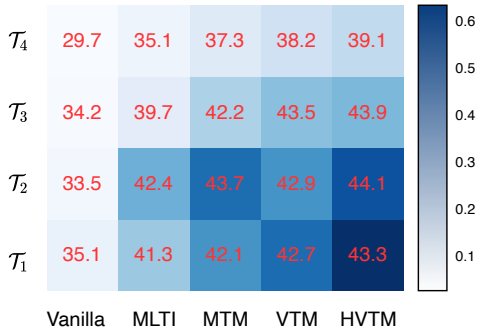

Figure 5. **Analysis of modulated tasks.** Similarity of meta-training tasks to meta-test tasks for different methods, and the corresponding accuracy (red numbers) for the meta-test tasks. The tasks modulated by MetaModulatation have high similarity with the meta-test tasks, resulting in high accuracy.

But, the similarity between the modulated tasks by our MetaModulation and $\mathcal{T}_4$ is also relatively low and performance is also poor. This may be because the task distribution of $\mathcal{T}_4$ is an outlier in the entire task distribution, making it hard to mimic this task during meta-training. Future work could investigate ways to mimic these outlier tasks in the meta-training tasks.

Comparison with state-of-the-art. We evaluate MetaModulation on the four different datasets under 5-way 1-shot and 5-way 5-shot in Table 5. Our model achieves state-of-the-art performance on all four few-task meta-learning benchmarks under each setting. On miniImagenet-S, our model achieves 43.21% under 1-shot, surpassing the second-best MLTI (Yao et al., 2021b), by a margin of 1.85%. On ISIC (Milton, 2019), our method delivers 76.40% for 5-shot, outperforming MLTI (Yao et al., 2021b) with 4.88%. Even on the most challenging DermNet-S, which forms the largest dermatology dataset, our model delivers 50.45% on the 5-way 1-shot setting. The consistent improvements on all benchmarks under various configurations confirm that our approach is effective for few-task meta-learning.

5. Related work

Few-task meta-learning. In few-task meta-learning, the goal is to develop meta-learning algorithms that learn quickly and efficiently from a small number of examples with limited tasks in order to adapt to new tasks with minimal additional training. A common strategy for few-task meta-learning is task augmentation (Yao et al., 2021a; Vu et al., 2021; Murty et al., 2021; Zhou et al., 2021; Wang & Deng, 2021; Wu et al., 2022; Wang et al., 2023), which adds additional tasks to the training data. One such approach is to generate additional tasks by perturbing the original tasks in some way (Yao et al., 2021a; Murty et al., 2021; Zhou et al., 2021; Wu et al., 2022; Wang et al., 2023). For example, MetaMix (Yao et al., 2021a) mixes support and query sets with Manifold Mixup (Verma et al., 2019) to construct a

	miniImagenet-S		ISIC		Dermnet-S		Tabular Murris	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet (Snell et al., 2017)	36.26	50.72	58.56	66.25	44.21	60.33	80.03	89.20
MAML (Finn et al., 2017)	38.27	52.14	57.59	65.24	43.47	60.56	79.08	88.55
Meta-Dropout (Lee et al., 2020)	38.32	52.53	58.40	67.32	44.30	60.86	78.18	89.25
TAML (Jamal & Qi, 2019)	38.70	52.75	58.39	66.09	45.73	61.14	79.82	89.11
MetaMix (Yao et al., 2021a)	39.67	53.10	60.58	70.12	47.71	62.68	81.06	89.75
Meta-Maxup (Yao et al., 2021a)	39.80	53.35	59.66	68.97	46.06	62.97	79.56	88.88
Meta Interpolation (Lee et al., 2022)	40.28	53.06	-	-	-	-	-	-
ATA (Wang et al., 2023)	40.62	54.59	-	-	-	-	-	-
MLTI (Yao et al., 2021b)	41.36	55.34	62.82	71.52	49.38	65.19	81.89	90.12
ATU (Wu et al., 2022)	42.60	56.78	62.84	74.50	48.33	65.16	82.03	91.42
This paper: ***MetaModulation***	**43.21**±0.73	**57.26**±0.72	**65.61**±1.09	**76.40**±0.89	**50.45**±0.84	**67.05**±0.74	**83.13**±0.89	**91.23**±0.57

Table 5. **Comparison with state-of-the-art.** All results, except for the MetaInterpolation (Lee et al., 2022), are sourced from MLTI (Yao et al., 2021b). MetaModulation is a consistent top performer for all settings and datasets.

new query set. Another approach is to rely on unsupervised or self-supervised learning to generate additional tasks from the training data (Vu et al., 2021; Wang & Deng, 2021). An alternative few-task meta-learning strategy is task interpolation (Yao et al., 2021b; Lee et al., 2022), which trains a model to learn from a set of interpolated tasks. For example, MLTI (Yao et al., 2021b) performs Manifold Mixup on support and query sets from two tasks for task augmentation. Set-based meta-interpolation (Lee et al., 2022) leverages expressive neural set functions (Lee et al., 2019) to interpolate a given set of tasks and trains the interpolating function using bilevel optimization so that the meta-learner trained with the augmented tasks generalizes to meta-validation tasks. Both task augmentation and interpolation methods often randomly mix the features of two known tasks in a linear way without considering the features of other layers. This limits the diversity of the interpolated task and its potential benefit for increasing model generalizability. In contrast, we propose a learnable task modulation method that enables the model to learn a more diverse set of tasks by considering the features of each layer and allowing for a non-linear modulation between tasks.

Conditional batch normalization. Batch normalization (Ioffe & Szegedy, 2015) is a crucial milestone in the development of deep neural networks. Conditional batch normalization (CBN) (De Vries et al., 2017) allows a neural network to learn different normalization parameters per class of input data. Note the contrast to traditional batch normalization, which uses the same normalization parameters for all inputs to a network layer. By conditioning the normalization on additional information, such as the class labels of the training examples, CBN allows the network to adapt its normalization parameters to the specific class characteristics. Similarly, Perez *et al.* (Perez et al., 2018) propose the feature-wise linear modulation layer for deep neural networks. In this paper, we take inspiration from conditional batch normalization and propose meta task modulation for few-task meta-learning, where the condition stems from the samples of a meta-training task. We use the conditional task as the condition, instead of data from another modality as in (De Vries et al., 2017), to predict the scale and shift parameters of the batch normalization for the base task.

6. Conclusion

In this paper, we addressed the issue of meta-learning algorithms requiring a large number of meta-training tasks which may not be readily available in real-world situations. We propose MetaModulation, which is to use a neural network to increase the density of the meta-training tasks by modulating batch normalization parameters during meta-training. Our MetaModulation consists of three different implementations. First is the meta task modulation, which modified parameters at various levels of the neural network to increase task diversity. Furthermore, we proposed a variational meta task modulation where the modulation parameters are treated as latent variables. We also introduced learning variational feature hierarchies by the variational meta task modulation. Our ablation studies showed the advantages of utilizing a learnable task modulation at different levels and the benefit of incorporating probabilistic variants in few-task meta-learning. Our MetaModulation and its variational variants consistently outperformed state-of-the-art few-task meta-learning methods on four few-task meta-learning benchmarks.

Acknowledgment

This work is financially supported by the Inception Institute of Artificial Intelligence, the University of Amsterdam and the allowance Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy, the National Key R&D Program of China (2022YFC2302704), the Special Foundation of President of the Hefei Institutes of Physical Science (YZJJ2023QN06), and the Postdoctoral Researchers' Scientific Research Activities Funding of Anhui Province (2022B653).

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. 1

Cao, K., Brbic, M., and Leskovec, J. Concept learners for few-shot learning. *ICLR*, 2020. 5

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. Modulating early visual processing by language. In *NeurIPS*, 2017. 2, 3, 8

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 2

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICLR*, pp. 1126–1135, 2017. 1, 8

He, Y., Liang, W., Zhao, D., Zhou, H.-Y., Ge, W., Yu, Y., and Zhang, W. Attribute surrogates learning and spectral tokens pooling in transformers for few-shot learning. In *CVPR*, 2022. 1

Hu, S. X., Li, D., Stühmer, J., Kim, M., and Hospedales, T. M. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, pp. 9068–9077, June 2022. 1

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456. PMLR, 2015. 2, 8

Jamal, M. A. and Qi, G.-J. Task agnostic meta-learning for few-shot learning. In *CVPR*, pp. 11719–11727, 2019. 8

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv: Learning*, 2014. 6

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

Lee, H. B., Nam, T., Yang, E., and Hwang, S. J. Meta dropout: Learning to perturb latent features for generalization. In *ICLR*, 2020. 8

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 1, 8

Lee, S., Andreis, B., Kawaguchi, K., Lee, J., and Hwang, S. J. Set-based meta-interpolation for few-task meta-learning. In *NeurIPS*, 2022. 1, 2, 3, 8

Liu, Y., Zhang, W., Xiang, C., Zheng, T., Cai, D., and He, X. Learning to affiliate: Mutual centralized learning for few-shot classification. In *CVPR*, pp. 14411–14420, June 2022. 1

Milton, M. A. A. Automated skin lesion classification using ensemble of deep neural networks in isic 2018: Skin lesion analysis towards melanoma detection challenge. *arXiv preprint arXiv:1901.10802*, 2019. 5, 7

Murty, S., Hashimoto, T. B., and Manning, C. D. Dreca: A general task augmentation strategy for few-shot natural language inference. In *ACL*, pp. 1113–1125, 2021. 7

Ni, R., Goldblum, M., Sharaf, A., Kong, K., and Goldstein, T. Data augmentation for meta-learning. In *ICML*, pp. 8152–8161. PMLR, 2021. 1

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 2, 8

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *ICLR*, 2017. 1

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pp. 1278–1286. PMLR, 2014. 5

Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987. 1

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 1, 2, 6, 7, 8

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 1998. 1

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pp. 6438–6447, 2019. 1, 3, 6, 7

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *NeurIPS*, 2016. 5

Vu, T., Luong, M.-T., Le, Q. V., Simon, G., and Iyyer, M. Strata: Self-training with task augmentation for better few-shot learning. *arXiv preprint arXiv:2109.06270*, 2021. 7, 8

Wang, H. and Deng, Z.-H. Cross-domain few-shot classification via adversarial task augmentation. *arXiv preprint arXiv:2104.14385*, 2021. 7, 8

Wang, H., Mai, H., Gong, Y., and Deng, Z.-H. Towards well-generalizing meta-learning via adversarial task augmentation. *Artificial Intelligence*, 317:103875, 2023. 7, 8

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, pp. 2097–2106, 2017. 1

Wu, Y., Huang, L.-K., and Wei, Y. Adversarial task upsampling for meta-learning. In *Advances in Neural Information Processing Systems*, 2022. 7, 8

Yao, H., Huang, L.-K., Zhang, L., Wei, Y., Tian, L., Zou, J., Huang, J., et al. Improving generalization in meta-learning via task augmentation. In *ICML*, pp. 11887–11897. PMLR, 2021a. 1, 7, 8

Yao, H., Zhang, L., and Finn, C. Meta-learning with fewer tasks through task interpolation. *arXiv preprint arXiv:2106.02695*, 2021b. 1, 2, 3, 5, 6, 7, 8

Zhou, J., Zheng, Y., Tang, J., Li, J., and Yang, Z. Flipda: Effective and robust data augmentation for few-shot learning. *arXiv preprint arXiv:2108.06332*, 2021. 7

A. Effect of the β.

We test the impact of β in (20) and (21). The value of β control how much information in the base task will be modulated during the meta-training stage. The experimental results on the three datasets under both 1-shot and 5-shot setting are shown in Figure 6 and 7. We can see that the performance achieves the best when the values of β are 0.01. This means that in each modulate we need to keep the majority of base task.

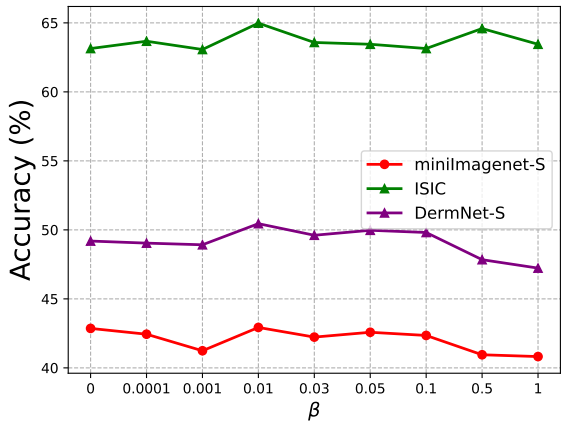

Figure 6. Performance comparison by using various β on the three few-task meta-learning dataset under 1-shot.

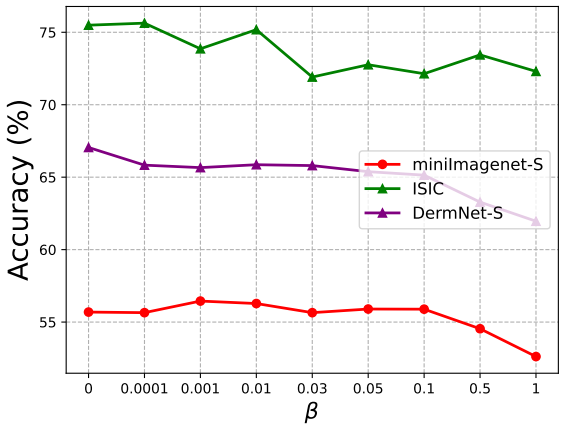

Figure 7. Performance comparison by using various β on the three few-task meta-learning dataset under 5-shot.

B. Effect of the λ.

We would like to emphasize that the hyper-parameters λ (Eq. 19, 20, 21) enable us to introduce constraints on new tasks, beyond just minimizing prediction loss. By adjusting the value of λ, we can control the trade-off between the prediction loss of the new tasks and the constraints imposed by the meta-training tasks. To clarify the impact of λ, we performed an ablation on the HVTM (Eq. 21). The results in Table 6 show that when the original tasks have higher weight, the performance is worse. Additionally, we have conducted experiments to investigate the distribution differences between the meta-training and generated tasks. Specifically, in Table 6, we analyze the task representations of meta-training and generated tasks and show that they are similar, indicating that the generated tasks have a similar distribution as the meta-training tasks.

	miniImagenet-S		**ISIC**	
	1-shot	5-shot	1-shot	5-shot
0.0001	41.97	55.23	65.25	76.23
0.001	42.65	56.18	65.61	76.40
0.01	**43.21**	**57.26**	**65.13**	**76.27**
0.05	43.14	57.09	65.07	76.13
0.1	42.86	56.16	63.05	74.72
0	42.25	55.97	62.95	74.15
1	41.46	55.12	62.15	72.73
10	40.26	53.17	60.03	70.95
100	38.01	51.25	59.12	68.23

Table 6. **Ablation on the λ.**

C. Dataset.

We apply our method to four few-task meta-learning image classification benchmarks. Sample images from each dataset are provided in Figure 8.

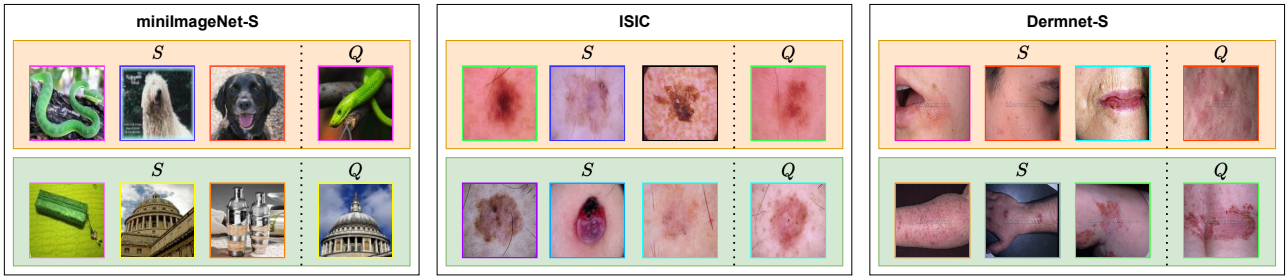

Figure 8. Examples from each dataset. Orange and green boxes indicate the meta-training and meta-test tasks for each dataset.