Meta Cyclical Annealing Schedule: A Simple Approach to Avoiding Meta-Amortization Error

Yusuke Hayashi\textsuperscript{1,}, Taiji Suzuki\textsuperscript{1,2,3}
\textsuperscript{1}Japan Digital Design, Japan
\textsuperscript{2}The University of Tokyo, Japan
\textsuperscript{3}Center for Advanced Intelligence Project, RIKEN, Japan
yusuke.hayashi@japan-d2.com, taiji@mist.i.u-tokyo.ac.jp

Abstract
The ability to learn new concepts with small amounts of data is a crucial aspect of intelligence that has proven challenging for deep learning methods. Meta-learning for few-shot learning offers a potential solution to this problem: by learning to learn across data from many previous tasks, few-shot learning algorithms can discover the structure among tasks to enable fast learning of new tasks. However, a critical challenge in few-shot learning is task ambiguity: even when a powerful prior can be meta-learned from a large number of prior tasks, a small dataset for a new task can simply be very ambiguous to acquire a single model for that task. The Bayesian meta-learning models can naturally resolve this problem by putting a sophisticated prior distribution and let the posterior be regularized through Bayesian decision theory. However, currently known Bayesian meta-learning procedures such as VERSA suffer from the so-called information preference problem, that is, the posterior distribution is degenerated to one point and is far from the exact one. To address this challenge, we design a novel meta-regularization objective using cyclical annealing schedule and maximum mean discrepancy (MMD) criterion. The cyclical annealing schedule is quite effective at avoiding such degenerate solutions. This procedure includes a difficult KL-divergence estimation, but we resolve the issue by employing MMD instead of KL-divergence. The experimental results show that our approach substantially outperforms standard meta-learning algorithms.

1 Introduction
The human visual system is efficient at grasping the main concepts of any new image from only a single or a few images. Over the last few years, few-shot learning techniques have been developed by many researchers to achieve human-level performance on image recognition tasks. Generally, it is expected that a “good” few-shot learning technique should satisfy properties such as the following: (i) it is able to learn new tasks with few-shot examples efficiently, thus learning the new categories fast; (ii) the performance can be improved even as increasing numbers of input samples are given on a new task; (iii) performance on the initial tasks at training time is not sacrificed (without forgetting).

Although many few-shot classification algorithms are proposed, it is a tough task to organize the best unified framework for few-shot learning. Metric learning methods [Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018] aim to learn a data-dependent metric to reduce intra-class distance and increase inter-class distances. Gradient-based meta-learning [Kim et al., 2018; Finn et al., 2017; Snell et al., 2017] attempts to learn the commonalities among various tasks. MAML [Finn et al., 2017] is an effective meta-learning method that directly optimizes the gradient descent procedure for task-specific learners. In the amortized Bayesian inference framework, [Qiao et al., 2018; Ravi and Beatson, 2019] proposed a method for predicting the weights of classes from activations of a pre-trained network to transfer from a high-shot classification task to a separate low-shot classification task. Recently, [Gordon et al., 2019] proposed a general meta-learning framework (ML-PIP) with approximate probabilistic inference and its implementation to few shot learning tasks (VERSA). ML-PIP unified a number of important approaches on meta-learning, including both gradient- and metric-based meta-learning [Kim et al., 2018; Finn et al., 2017; Snell et al., 2017; Qiao et al., 2018; Ravi and Beatson, 2019] with amortized inference frameworks (neural processes) [Garnelo et al., 2018a; Garnelo et al., 2018b]. It is a general framework because of the end-to-end training and supports full multi-task learning by sharing information between many tasks. In particular, VERSA replaces the optimization at test time with efficient posterior inference by generating a distribution over the task-specific parameters in a single forward pass. Therefore, this framework can amortize the cost of inference and relieve the need for second derivatives for few-shot training during test time. It is also worth noting that their inference framework is focused on the posterior predictive distribution, i.e., it aims to minimize the KL-divergence between the true and approximate predictive distributions rather than maximizing the ELBO, which is generally utilized in VAE-based methods [Kingma and Welling, 2014].

In the state-of-the-art models on few-shot learning tasks, amortized inference distribution is practically utilized be-
cause it is efficient and scalable to large datasets, and it re-
quires only the specified parameters of the neural network.
However, to get proper amortized inference, we need to
tackle the amortization gap problem and information prefer-
ence problem as stated below. As analyzed in [Cremer et al.,
2018], the inference mismatch between the true and approx-
imate posterior which consists of two gaps (i) approximation
gap and (ii) amortization gap. Their conclusions are that in-
creasing the capacity of the encoder reduces the amortization
error and when efficient test time inference is required, en-
coder generalization is important and expressive approxima-
tions in decoder are likely advantageous. Another example of
the estimation difficulty of amortized inference is that cosine-
similarity-based non-amortization models [Chen et al., 2019]
achieved superior performance than those with amortization
inference on few-shot learning. This implies that effective es-
timation methodology for amortization inference has still not
been established.

Our contributions in this paper are as follows:

1. We show that one of the amortization gap problems
comes from the information preference problem of the
latent distribution.
2. We adapt both the annealing method and regularization
of parameter estimation in the amortized inference
network to avoid the information preference problem by ap-
plying cyclical annealing schedule and maximum mean
discrepancy.
3. Our proposal meets the “good” properties of few-shot
learning, get better performance on standard few-shot
classification tasks.

Despite its simplicity of our proposed method, it can signif-
icantly improve the performance. Through several experi-
mental analyses, we show that our methodology outperforms
other state-of-the-art few-shot learning algorithms.

2 Preliminaries

Along with the many few-shot learning methods, a num-ber of measures for assessing their actual performance has
also been proposed. The ML-PIP model unified a num-er of important approaches on meta-learning, including both
gradient and metric based meta-learning [Kim et al., 2018;
Finn et al., 2017; Snell et al., 2017; Qiao et al., 2018] with
amortized inference framework [Garnelo et al., 2018a; Gar-
nelo et al., 2018b]. Although their method is similar to these
models, it is more general, employing end-to-end training and
supporting full multi-task learning by sharing information be-
tween many tasks. In this section, we describe the multi-task
meta-learning problem that we deal with in this paper, and
we review the VERSA (implementation of ML-PIP for meta-
learning) and neural processes (NPs) [Gordon et al., 2019;
Garnelo et al., 2018b].

2.1 Meta-learning problem

In this paper, we mainly consider few-shot classification
problems, in which we are given few-shot (say, k-shot) ob-
servations consisting of input-output pairs \( \{(x_i^t, y_i^t)\}_{i=1}^k \) for
each of the \( C \)-classes (we call C-way), and we perform C-
class classification for an unseen test input data. We call
this problem C-way k-shot meta-learning problem. One typi-
cal approach to tackle this problem is to construct an “en-
coder” \( h(x) \in \mathbb{R}^d \) beforehand, estimate a weight vector
\( W_c (c = 1, \ldots, C) \) from the few-shot observations and
apply the softmax operation for the linear discriminator (we call
“decoder”) \( \{W_c^T h(x)\}_{c=1}^C \). The encoder \( h \) is usually trained
based on other training data (which typically does not contain
the C-class few-shot observations) so that \( h \) extracts informa-
tive features that can distinguish the unseen classes. For the
training phase, we are given training data for several tasks,
\( D_{tr}^{(t)} = D_{S, tr}^{(t)} \cup D_{Q, tr}^{(t)} \) \( (t \) is the task index: \( t = 1, \ldots, T \) where
\( D_{S, tr}^{(t)} = \{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^N \) and \( D_{Q, tr}^{(t)} = \{(\tilde{z}_j^{(t)}, \tilde{y}_j^{(t)})\}_{j=1}^M \)
the number of observations \( N = k \times C \) for each task is
supposed to be small. Based on the support dataset \( D_{S, tr}^{(t)} \),
we train the encoder \( h \) and the network which produces the
weight vector \( \{W_c\}_{c=1}^C \). This procedure can be seen as a
kind of learning a training procedure. In the test phase, we are
given test data \( D_{ts}^{(t)} = D_{S, ts}^{(t)} \cup D_{Q, ts}^{(t)} \) \( (t \) is the task in-
dex: \( t = 1, \ldots, T' \) where \( D_{S, ts}^{(t)} = \{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^N \) and
\( D_{Q, ts}^{(t)} = \{(\tilde{z}_j^{(t)}, \tilde{y}_j^{(t)})\}_{j=1}^M \) of new unseen tasks. Based
on the support dataset \( D_{S, ts}^{(t)} \), the encoder produces the new
weight vector \( \{W_c\}_{c=1}^C \). In the few-shot learning setting, \( \tilde{y}_j^{(t)} \)
is a class label among \( C \)-classes and the total number of data
is \( M \).

2.2 Meta-Learning via amortized Bayesian
inference

VERS A is a Bayesian meta-learning framework and is also
used in the few-shot classification task. Its graphical model
is shown in Figure [1](a). VERSA consists of two parts: an
encoder and an amortization network. The encoder \( h(x) =
h_\theta(x) \) maps an input to a feature vector, and its parameter
is denoted by \( \theta \) (we call global latent variable). We use the
same \( \theta \) across all the tasks. As we have described in the previous
section, this encoder is trained through the support dataset
\( D_{S, tr}^{(t)} \). When a new task appears at test time, the same en-
coder as the one estimated at the training time is used for test
as well; that is, for a newly observed task \( t \), shared statisti-
cal encoder \( h \) is fed \( \tilde{z}^{(t)} \) as input and it outputs \( h_\theta(\tilde{x}^{(t)}) \) as a
representation of the input \( \tilde{x} \). The amortization network
outputs the predictive distribution from the representation of
the input \( h_\theta(\tilde{x}^{(t)}) \). It is characterized by the task specific parame-
ter \( \lambda^{(t)} \) which represents a network that maps the encoded
input \( h_\theta(\tilde{x}) \) to the parameters of the approximated posterior
distribution of the parameters of the output label \( \hat{y}^{(t)} \). In
VERS A model, \( \lambda^{(t)} \) have to be trained with few-shot sam-
ple at training time using the training data \( D_{tr}^{(t)} \). In practice,
as the amortized function, essentially a neural network, is es-
timated to take a representation variable as input, and outputs
the mean and variance parameter for predictive distribution of
each task.
For the few-shot classification task, VERSA encodes the class \( c \in \{1, \ldots, C\} \) by the average of the encoded-input \( \bar{h}_c = \sum_{i=1}^{k} h_{\theta}(x^i_c) \). This acts like the weight vector \( W_c \) for the classification. Basically, the predictive distribution for \( \hat{y} \) is given by the softmax value of \( \{\bar{h}_c, h_{\theta}(x)\}_{c=1}^C \). To obtain the approximated posterior predictive distribution, VERSA generates \( \phi_{c,l} \) as a stochastic version of \( \bar{h}_c \) from the Gaussian distribution with mean \( \phi_{\mu} \) and variance \( \phi_{\sigma}^2 \) specified by the output of \( \lambda(t) \) \( (\phi_{c,l} \sim \mathcal{N}(\phi_{\mu}(\bar{h}_c)), \phi_{\sigma}(\bar{h}_c)) \), and sample the predictive distribution corresponding to \( \frac{1}{2} \sum_{l=1}^{L} \text{softmax} \left( \{ \phi_{c,l}^T h_{\theta}(x) \}_{c=1}^C \right) \).

This framework approximates the posterior predictive distribution by an amortized distribution as follows. Here, the predictive distribution of the test output \( \hat{y} \) given the input \( \hat{x} \) and the few-shot sample \( h_{\theta}(x) \) is given as

\[
p(\hat{y} | \hat{x}, h_{\theta}(x), \theta) = \int p(\hat{y} | \hat{x}, \phi, h_{\theta}(x), \theta) d\phi = \int p(\hat{y} | \hat{x}, \phi, \theta) p(\phi | \hat{x}, h_{\theta}(x), \theta) d\phi. \tag{1}
\]

where \( p(\hat{y} | \hat{x}, \phi, \theta) \) corresponds to softmax function. However, the posterior distribution of \( \phi \) is difficult to calculate. Therefore, VERSA approximates the predictive distribution by the amortized distribution \( p(\phi | \hat{x}, h_{\theta}(x), \theta) \) by utilizing the approximated posterior distribution \( q_{\lambda}(\phi | h_{\theta}(x)) \). VERSA employs a Gaussian distribution as the approximated posterior \( q_{\lambda}(\phi | h_{\theta}(x)) \) which is characterized by the network output: \( \phi \sim \mathcal{N}(\phi_{\mu}(h_{\theta}(x)), \phi_{\sigma}(h_{\theta}(x))) \). Then, the amortized predictive distribution is given as

\[
q_{\lambda}(\hat{y} | \hat{x}, h_{\theta}(x), \theta) = \int q_{\lambda}(\hat{y}, \phi | \hat{x}, h_{\theta}(x), \theta) d\phi = \int p(\hat{y} | \hat{x}, \phi, \theta) q_{\lambda}(\phi | h_{\theta}(x), \theta) d\phi. \tag{2}
\]

Since VERSA wants to approximate the predictive distribution as accurate as possible, the end-to-end training objective to be minimized for \( \theta \) and \( \lambda = \{\lambda(t)\}_{t=1}^T \) is given as follows:

\[
\mathcal{L}(\theta, \lambda) = -\mathbb{E}_{p(\hat{y} | \hat{x}, h_{\theta}(x), \theta)} [\log q_{\lambda}(\hat{y} | \hat{x}, h_{\theta}(x), \theta)] = -\mathbb{E}_{p(\hat{y} | \hat{x}, h_{\theta}(x), \theta)} \left[ \log \int p(\hat{y} | \hat{x}, \phi, \theta) q_{\lambda}(\phi | h_{\theta}(x), \theta) d\phi \right] = -\mathbb{E}_{p(\hat{y} | \hat{x}, h_{\theta}(x), \theta)} \left[ \log \int \frac{1}{T} \sum_{t}^{T} q_{l(t)}(\hat{y} | \hat{x}, \phi, \theta) d\phi \right]. \tag{3}
\]

However, in general, learning “good” latent code is difficult because even when a powerful prior can be meta-learned from a large number of prior tasks, a small dataset for a new task can simply be too ambiguous to acquire a single accurate model. Here, we consider a more general objective which includes the regularization term:

\[
\mathbb{D}_{\text{KL}} [p(\hat{y}, \phi | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)] = \mathbb{D}_{\text{KL}} [p(\hat{y} | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)] + \mathbb{D}_{\text{KL}} [p(\phi | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)]. \tag{4}
\]

In the objective, the KL-divergence \( \mathbb{D}_{\text{KL}} [p(\phi | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)] \) between the posterior distributions work as regularization. Unfortunately, the conditional prior \( p(\phi | \hat{x}, h_{\theta}(x), \theta) \) in the above expression is intractable. To resolve this issue, we instead use an approximated posterior \( q_{\lambda}(\phi | \hat{x}, h_{\theta}(x), \theta) \), which gives:

\[
\mathbb{D}_{\text{KL}} [p(\hat{y}, \phi | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)] \approx \mathbb{D}_{\text{KL}} [p(\hat{y} | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)] + \mathbb{D}_{\text{KL}} [p(\phi | \hat{x}, h_{\theta}(x), \theta) || q_{\lambda}(\phi | h_{\theta}(x), \theta)]. \tag{5}
\]

It is interesting to note that replacing the feature vector \( h_{\theta}(x) \) with \( r(h_{\theta}(x), y) \), we got the NPs objective from the above objective:

\[
\mathbb{D}_{\text{KL}} [p(\hat{y}, \phi | \hat{x}, r(h_{\theta}(x), y), \theta) || q_{\lambda}(\phi | r(h_{\theta}(x), y), \theta)] \approx \mathbb{D}_{\text{KL}} [p(\hat{y} | \hat{x}, r(h_{\theta}(x), y), \theta) || q_{\lambda}(\phi | r(h_{\theta}(x), y), \theta)] + \mathbb{D}_{\text{KL}} [p(\phi | \hat{x}, r(h_{\theta}(x), y), \theta) || q_{\lambda}(\phi | r(h_{\theta}(x), y), \theta)]. \tag{6}
\]
where the function $r$ is the neural network. NPs combines the strengths of neural networks and Gaussian processes to achieve both flexible learning and fast prediction in stochastic processes. While VERSA uses linear discriminator as a decoder, NPs uses neural network as it. Both models are represented in Figure 1(b).

The point is that, in VERSA and NPs, the central (stochastic) function being learnt has a form $\tilde{y} = f(\hat{x}, D_S, \theta)$, of an output $\tilde{y}$ given an input $\hat{x}$, a support dataset $D_S$ and the encoder’s parameter (global latent variable) $\theta$.

### 2.3 Information preference property

As in VERSA and NPs, we consider the following generative process for $\tilde{y}$,

$$\phi \sim p(\phi | \hat{x}, D_S, \theta), \quad \tilde{y} \sim p(\tilde{y} | \hat{x}, \phi, \theta), \quad (7)$$

where $p(\phi | \hat{x}, D_S, \theta)$ is the prior and $p(\tilde{y} | \hat{x}, \phi, \theta)$ is given by a generative model with parameter $\theta$. Under ideal conditions, optimizing the objective using sufficiently flexible model families for $p(\tilde{y} | \hat{x}, \phi, \theta)$ and $q_\lambda(\phi | D_S, \theta)$ over $\theta, \lambda$ will achieve both goals of correctly capturing $p(\tilde{y} | \hat{x}, D_S, \theta)$ and performing correct amortized inference. However, this approach suffers from the following problem: the decoder tends to neglect the latent variables $\phi$ altogether, that is, the mutual information between $\phi$ and $\tilde{y}$ conditioned on $(\hat{x}, \theta)$ becomes negligibly small. For example,

$$\frac{p(\phi | \hat{x}, D_S, \theta)}{q_\lambda(\phi | D_S, \theta)} = \frac{p(\phi | D_S, \theta)}{p(\phi | D_S, \theta) p(\hat{x} | D_S, \theta) q_\lambda(\phi | D_S, \theta)} \cdot \frac{p(\phi | D_S, \theta)}{q_\lambda(\phi | D_S, \theta)}.$$  

Therefore,

$$\mathbb{E}_{p(\phi | D_S, \theta)} [D_{KL} [p(\phi | \hat{x}, D_S, \theta) || q_\lambda(\phi | D_S, \theta)]]  \quad (8)$$

$$= \int p(\phi, \hat{x} | D_S, \theta) \log \frac{p(\phi | \hat{x}, D_S, \theta)}{q_\lambda(\phi | D_S, \theta)} d\phi d\hat{x}  \quad (9)$$

$$= \mathbb{I}_p [\phi, \hat{x} | D_S, \theta] + D_{KL} [p(\phi | D_S, \theta) || q_\lambda(\phi | D_S, \theta)].$$

The above objective tells us that the more the learning procedure proceeds (that is, the left-hand side reduces), the more the mutual information between $\phi$ and $\hat{x}$ decreases. It follows that the mutual information between $\phi$ and $\tilde{y}$ decrease. Intuitively, the reason is because the distribution of $\phi$ tends to shrink to a single point and $\phi$ is almost uniquely identified by a given $\hat{x}$ and $\theta$. This is undesirable because such a shrunken posterior of $\phi$ is far from the true posterior and severely lose variation of the posterior sampling for $\phi$. This effect, which we shall refer to as the information preference problem, was studied in the VAE framework with a coding efficiency argument [Kingma and Welling, 2014]. In the VAE framework, the issue causes two undesirable outcomes: (1) the learned features are almost identical to the uninformative Gaussian prior for all observed tasks; and (2) the decoder completely ignores the latent code, and the learned model reduces to a simpler model [Fu et al., 2019].

### 3 Proposed Method

As seen in the previous section, a unified Bayesian inference framework ML-PIP and its implementation VERSA and NPs utilize the amortized inference distribution because it is efficient and scalable to large datasets, and it requires only the specified parameters of the neural network. However, as we have seen in the previous section, the regularization in the objective causes the information preference problem (which is a well-known issue in the VAE framework) and inaccurate estimation of amortized inference distributions.

One approach to remedy this issue is to introduce a hyperparameter $\beta$ to control the strength of regularization [Higgins et al., 2017]. Furthermore, [Fu et al., 2019] found that scheduling $\beta$ during the model training highly improve the performance. In addition, [Zhao et al., 2018] reported an alternative approach: replacing the KL-divergence of the latent distributions in the objective with the alternative divergence.

However, these previous studies applied their methodologies to only single task learning framework. In contrast, this paper considers the cyclical annealing schedule for $\beta$ during multi-task learning (meta-training) and replacing the divergence with the maximum mean discrepancy criterion. This procedure leads high mutual information between $\hat{x}$ and $\phi$, but
which encourages the model to use the latent code and avoids the information preference problem.

3.1 Cyclic annealing schedule

Several attempts have been made to ameliorate the information preference problem in the VAE framework. Among them, the simplest solution is monotonic KL annealing, where the weight of the KL penalty term $\beta$ is scheduled to gradually increase during training (monotonic schedule) [Bowman et al., 2016] (see Figure 3 (a)). In the VAE framework, latent code $\phi$ learned earlier can be viewed as the initialization; such latent variables are much more informative than random and are thus ready for the decoder to use. Therefore, to mitigate the information preference problem, it is key to have meaningful $\phi$ at the beginning of training the decoder, so that $\phi$ can be utilized. Furthermore, [Fu et al., 2019] found that simply repeating the monotonic schedule multiple times (cyclical annealing schedule) enables high improvement in the above method (see Figure 3 (b)).

We apply this type of regularization to the multi-task learning context. We basically consider the following objective that anneals the second term of the right-hand side in Eq. (6) by the factor $\beta$ with $0 \leq \beta \leq 1$:

$$\mathcal{L}_{MCA} = \mathcal{L}(\theta, \lambda) + \beta \mathbb{D}_{KL}[p(\phi|x, D_S, \theta) || q_\lambda(\phi|D_S, \theta)] .$$

We can see that by annealing the regularization term (second term), the effect of information preference property discussed in Section 2.3 is mitigated. However, it is experimentally observed that just utilizing a fixed $\beta$ does not produce a good result. Thus, we gradually change the penalty term $\beta$ during the training. For that purpose, we decompose the objective into each task and control $\beta$ depending on each task as

$$\mathcal{L}_{MCA} = \sum_{t=1}^{T} \mathcal{L}_t (\theta, \lambda(t)) + \beta(t) \mathcal{R}(t),$$

where $\mathcal{L}_t$ is the loss corresponding to a task $t$ and $\mathcal{R}(t)$ is the regularization term for each training task. During the training, we randomly sample task $t$ one after another and update the parameters where we apply different $\beta(t)$ at each update. There could be several possibility of scheduling $\beta(t)$ (Figure 3), but we employ the cyclical annealing schedule during the training phase. This realizes that each task can be trained with several different values of $\beta$ throughout the training, which yields avoiding the information preference problem. Our experimental results show that this approach helps each task-specific learner to avoid falling into local minima. We call this approach as the meta cyclical annealing (MCA).

3.2 Maximum mean discrepancy

Unfortunately, the KL-term in the right-hand side in Eq. 3 is difficult to compute. Therefore, we employ the maximum mean discrepancy (MMD) [Gretton et al., 2007] as a discrepancy measure between the distributions, which enables us to compute the corresponding term. MMD is a framework to quantify the distance between two distributions by comparing all of their moments via a kernel technique. Letting $k(\cdot, \cdot)$ be any positive definite kernel, such as Gaussian kernel $k(\phi, \phi') = \exp(-\frac{||\phi-\phi'||^2}{2\sigma^2})$. MMD between $p$ and $q$ is defined as: $\mathbb{D}_{MMD}[p(\phi) || q(\phi')] = \mathbb{E}_{p(\phi),p(\phi')}[k(\phi, \phi')] + \mathbb{E}_{q(\phi),q(\phi')}[k(\phi, \phi')].$ It is known that if the kernel $k$ is characteristic, $\mathbb{D}_{MMD} = 0$ if and only if $p(\phi) = q(\phi)$ [Muandet et al., 2017]. A rough intuition of MMD is that difference of the moments of each distributions $p(\phi)$ and $q(\phi)$ are measured through the (characteristic) kernel to know how different those distributions are. MMD can accomplish this efficiently via the kernel embedding trick.

We propose to employ MMD as the alternative divergence because MMD is easy to calculate and is stable against the support mismatch between the two distributions. To do so, instead of optimizing the objective introduced in Eq. 8, we minimize the following objective:

$$\mathcal{L}_{MMD} = \mathcal{L}(\theta, \lambda) + \beta \mathbb{D}_{MMD}[p(\phi|x, D_S, \theta) || q_\lambda(\phi|D_S, \theta)].$$

![Figure 3: Annealing $\beta$ with (a) monotonic schedule, (b) cyclical annealing schedule during multi-task learning (meta-training).](image-url)
complex and difficult enough to evaluate few-shot classification tasks. Training proceeds in the same episodic manner as with Omniglot.

4.3 Effect of regularization

Here, we checked if our model estimates the latent code \( \phi \) as expected via MCA and MMD. MCA and MMD regularizes the latent distribution close to standard Gaussian distribution to avoid information preference problem. Figure 4 shows the distributions of \( \phi \). We can see that the distributions of MCA+NPs and MCA+MMD+NPs are well regulated and close to Gaussian distributions (see Figure 4(c) and (d)). This is because of our MCA and MMD regularization. On the other hand, the distribution of NPs is far from Gaussians and the distribution of each class tends to degenerate to one-point (like a delta-distribution) which loses variation of \( \phi \), resulting in worse posterior approximation (see Figure 4(a)). This supports our expectation that the MMD-regularization effectively avoids the information preference problem.

4.4 Few-shot classification

To compare with existing methods, we focus our method on standard few-shot classification tasks, 20-way classification for Omniglot and 5-way classification for mini-ImageNet. We do not evaluate 5-way classification for Omniglot because it is already set to more than 99% with the existing methods, which is too high for comparing accuracy.

The results of Omniglot are shown in Table 1. Our proposal, MCA+NPs and MCA+MMD+NPs set good results. For 20-way 1-shot classification of Omniglot, our model achieves a new state-of-the-art result \(99.81\pm0.14\) which is significantly improved comparing with existing methods.

The results ofmini-ImageNet is shown in Table 2. We see that, for mini-ImageNet, MCA+NPs achieves \(77.33\pm1.67\)% for 5-way 1-shot classification, MCA+MMD+NPs achieves \(91.78\pm0.89\)% for 5-way 5-shot classification. Both results are also new state-of-the-art. Furthermore, our experimental results demonstrate that our models surpass VERSA in terms of performance, which suggests that mitigating amortization error provides improvement.

5 Conclusions

In this paper, we proposed the MCA+NPs and MCA+MMD+NPs models to improve amortized inference distribution with regularization techniques based on the latest few-shot learning framework, VERSA [Gordon et al., 2019] and NPs [Garnelo et al., 2018]. Through comparing methods on a common ground, our results show that the MCA+MMD+NPs model is comparable to state-of-the-art models under standard conditions, and the MCA+NPs model achieves comparable performance to recent state-of-the-art meta-learning algorithms on both the Omniglot and mini-ImageNet benchmark datasets. Additionally, our proposal seems to avoid the information preference problem by analysis.

| METHODS | Omniglot 20-way ACCURACY(%) |
|---------|-----------------------------|
|         | 1-shot | 5-shot |
| Matching Nets [Vinyals et al., 2016] | 93.8 | 98.5 |
| MAML [Finn et al., 2017] | 98.5±0.3 | 98.9±0.2 |
| Prototypical Nets [Snell et al., 2017] | 95.4 | 98.7 |
| Meta-SGD [Li et al., 2017] | 95.93±0.38 | 98.97±0.19 |
| Relation Net [Sung et al., 2018] | 97.6±0.1 | 99.1±0.1 |
| Reptile [Nichol et al., 2018] | 89.43±0.14 | 97.12±0.32 |
| CNPs [Garnelo et al., 2018a] | 89.9 | 96.8 |
| VERSA [Gordon et al., 2019] | 97.66±0.29 | 98.77±0.18 |
| MCA+NPs | 99.73±0.12 | 99.9±0.21 |
| MCA+MMD+NPs | 99.81±0.14 | 99.9±0.17 |

Table 1: Accuracy comparison of few-shot classification. The ± sign indicates the 95% confidence interval. Bold text indicates the highest scores that overlap in their confidence intervals.

| METHODS | mini-ImageNet 5-way ACCURACY(%) |
|---------|-----------------------------|
|         | 1-shot | 5-shot |
| Matching Nets [Vinyals et al., 2016] | 46.6 | 60 |
| MAML [Finn et al., 2017] | 48.7±1.84 | 63.11±0.92 |
| Prototypical Nets [Snell et al., 2017] | 46.61±0.78 | 65.77±0.70 |
| Meta-SGD [Li et al., 2017] | 50.47±1.87 | 64.03±0.94 |
| Relation Net [Sung et al., 2018] | 50.44±0.82 | 65.32±0.70 |
| Reptile [Nichol et al., 2018] | 49.97±0.32 | 65.99±0.58 |
| VERSA [Gordon et al., 2019] | 53.40±1.82 | 67.37±0.86 |
| LEO [Rusu et al., 2019] | 61.76±0.08 | 77.59±0.12 |
| MetaOptNet-SVM [Lee et al., 2019] | 64.09±0.62 | 80.00±0.45 |
| ACC+Amphibian [Sung et al., 2019] | 64.21±0.62 | 87.75±0.73 |
| MCA+NPs | 77.33±1.67 | 90.83±0.88 |
| MCA+MMD+NPs | 76.67±1.46 | 91.78±0.89 |

Table 2: Accuracy comparison of few-shot classification. The ± sign indicates the 95% confidence interval. Bold text indicates the highest scores that overlap in their confidence intervals.
Acknowledgments
We would like to thank Naonori Ogasahara and Iwato Amano for insightful discussions. Taiji Suzuki was partially supported by JSPS Kakenhi (26280009, 15H05707 and 18H03201), and JST-CREST.

References
[Bowman et al., 2016] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. Proceedings of Conference on Computational Natural Language Learning SIGNLL, 2016.

[Chen et al., 2019] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. The International Conference on Learning Representations (ICLR), 2019.

Cremer et al., 2018] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. The International Conference on Learning Representations (ICLR), 2018.

[Deng et al., 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[Finn et al., 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (ICML), 2017.

[Fu et al., 2019] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclic annealing schedule: A simple approach to mitigating KL vanishing. Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019.

[Garnelo et al., 2018a] Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional neural processes. In International Conference on Machine Learning (ICML), 2018.

[Garnelo et al., 2018b] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. In International Conference on Machine Learning (ICML), 2018.

[Gordon et al., 2019] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for Prediction. The International Conference on Learning Representations (ICLR), 2019.

[Gretton et al., 2007] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schlkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In Advances in Neural Information Processing Systems (NIPS), 2007.

[Higgins et al., 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. The International Conference on Learning Representations (ICLR), 2017.

[Kim et al., 2018] Yoon Kim, Sam Wiseman, Andrew C. Miller, David Sontag, and Alexander M. Rush. Semi-amortized variational autoencoders. The International Conference on Learning Representations (ICLR), 2018.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. The International Conference on Learning Representations (ICLR), 2014.

[Lake et al., 2015] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 2015.

[Lee et al., 2019] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. 2019.

[Li et al., 2017] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. 2017.

[Muandet et al., 2017] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Scholkopf, et al. Kernel mean embedding of distributions: A review and beyond. Foundations and Trends in Machine Learning, 10(1-2):1–141, 2017.

[Nichol et al., 2018] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. 2018.

[Qiao et al., 2018] Situan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[Ravi and Beatson, 2019] Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. The International Conference on Learning Representations (ICLR), 2019.

[Rusu et al., 2019] Andrei A. Rusu, Dushyant Rao, Jakub Synnorski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. 2019.

[Snell et al., 2017] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems (NIPS), 2017.

[Song et al., 2019] Liang Song, Jinlu Liu, and Yongqiang Qin. Fast and generalized adaptation for few-shot learning. 2019.

[Sung et al., 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[Vinyals et al., 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 (NIPS), 2016.
[Zhao et al., 2018] Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Information maximizing variational autoencoders. 2018.