Variational Task Encoders for Model-Agnostic Meta-Learning

Luuk Schagen[1], **Joaquin Vanschoren**[2]
[1] Jheronimus Academy of Data Science, the Netherlands
[2] Eindhoven University of Technology, the Netherlands

Abstract

Meta-learning allows an intelligent agent to leverage prior learning episodes as a basis for quickly improving performance on novel tasks. A critical challenge lies in the inherent uncertainty about whether new tasks can be considered similar to those observed before, and robust meta-learning methods would ideally reason about this to produce corresponding uncertainty estimates. We extend model-agnostic meta-learning with variational inference: we model the identity of new tasks as a latent random variable, which modulates the fine-tuning of meta-learned neural networks. Our approach requires little additional computation and doesn't make strong assumptions about the distribution of the neural network weights, and allows the algorithm to generalize to more divergent task distributions, resulting in better-calibrated uncertainty measures while maintaining accurate predictions.

1 Introduction

Meta-learning algorithms aim to leverage prior experience from multiple related tasks (Lake et al., 2017; Schmidhuber, 1987). However, it is not theoretically well understood what collection of tasks is sufficiently related so that effective meta-learning across these tasks can take place (Vuorio et al., 2019; Zintgraf et al., 2018). Because of this, it is important that meta-learning algorithms can reason with uncertainty about the tasks at hand. A robust meta-learner should be distribution agnostic, in the sense that it does not depend heavily on a uniform distribution over tasks to meta-learn from. While it is necessary to assume task similarity, applying a meta-learning algorithm to tasks lying outside of its originally trained task distribution should ideally not lead to nonsensical predictions, but to an honest conclusion by the algorithm that it is uncertainty about the task at hand.

To realize this, we place the work of Finn et al. (2017) and Vuorio et al. (2019) on Model Agnostic Meta-Learning (MAML) methods in a Bayesian framework (Blundell et al., 2015; Kingma & Welling, 2014). We continue the work of Ravi & Beatson (2018), Grant et al. (2018) and Finn et al. (2018), but then in a distribution agnostic context to allow for the application of meta-learning over more divergent tasks. In particular, we model the identity of new tasks as a dense variational vector which modulates the fine-tuning of meta-learned neural networks.

Sections 2 and 3 of this paper will discuss the merits of the aforementioned works. Section 4 describes our variational inference extension, evaluated empirically in sections 5 and 6. Section 7 concludes.

2 Meta-Learning: a probabilistic perspective

Probabilistic machine learning can be expressed as learning the posterior distribution $p(\theta|\mathrm{D})$, with model parameters θ and data D, by solving the following MAP estimation: $\theta^* =_\theta p(\mathrm{D}|\theta)p(\theta)$.

As shown graphically in Figure 1, we can translate this to a meta-learning setting by adding an intermediate step to this inference process, casting the problem as hierarchical Bayesian inference (Grant et al., 2018). In this case, θ represents the meta-parameters, and D represents a meta-dataset, or a dataset of datasets (tasks). The learned meta-parameters, in conjunction with a new task $\mathrm{D_i}$, allow us to recover the task-specific parameters ϕ_i:

$$p(\phi_i|\mathrm{D_i}, \theta), \tag{1}$$

5th Workshop on Meta-Learning at NeurIPS 2021, held virtually.

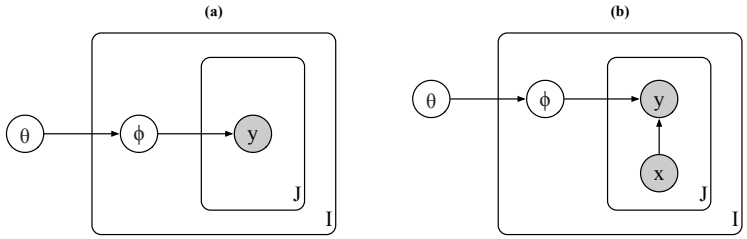


Figure 1: A high-level probabilistic graphical representation of meta-learning, in the unsupervised (a) and supervised (b) case. Index I represents the distinct tasks, while J indexes the individual datapoints within task I. θ represents the meta-parameters, ϕ represents the task specific parameters, and y are the labels to be predicted, informed by the input data x in the supervised case.

Applying principles derived from Bayes' rule, we arrive at the full hierarchical meta-learning objective:

$$p(\theta|\mathrm{D}) \propto \prod_i \left[\int_{\phi_i} p(\mathrm{D_i}|\phi_i) p(\phi_i|\theta) d\phi_i \right] p(\theta). \tag{2}$$

We can choose to replace this full marginalization by considering only the most likely value of ϕ_i, as obtained by maximizing (1) (Koller & Friedman, 2009). As this inference problem cannot be straightforwardly solved for the complex problems tackled in machine learning, two distinctive meta-learning approaches appeared that each approximate the inference procedure in their own way.

2.1 Model-based meta-learning

Model-based meta-learning approximates the inference procedure by introducing a meta-model f_θ, such as a neural network with weights θ, that maps the new task $\mathrm{D_i}$ to the task-specific parameters ϕ_i. This models the posterior over task-specific parameters ϕ_i explicitly as the output of a neural network, often using a point estimate of the maximum likely value of ϕ_i, which we refer to as ϕ_i^* (Schmidhuber et al., 1996). The predicted parameters ϕ_i^* parameterize a task-specific network $g_{\phi_i^*}$.

2.2 Model-agnostic meta-learning

In contrast, Model-Agnostic Meta-Learning (MAML) algorithms (Finn et al., 2017, 2018; Grant et al., 2018) meta-learn an initialization θ of the task-specific model parameters ϕ_i, to be finetuned with gradient descent to the new task to obtain ϕ_i^*. Grant et al. (2018) show that this corresponds to approximating (1) as the product of a likelihood $p(\mathrm{D_i}|\phi_i)$ and prior $p(\phi_i|\theta)$, as in the integral in (2), and is mathematically equivalent to placing a prior over the fine-tuned parameters ϕ_i^*, centered around θ. This means that the task-specific estimation of ϕ_i^* becomes a simple gradient descent fine-tuning procedure with a loss function L:

$$\phi_i^* \leftarrow \theta - \alpha \nabla_\theta \mathrm{L}(f(\mathrm{D_i}; \theta)), \tag{3}$$

performed for one or more gradient steps with learning rate α. The meta-learning objective becomes:

$$\min_\theta \mathrm{L}(f(\mathrm{D_i^{test}}; \phi_i^*)) \tag{4}$$

(Finn et al., 2017). This effectively results in an optimization problem where values for θ are found, such that it becomes easier to finetune them to optimal values for task specific parameters ϕ_i^* with a few steps of gradient descent.

2.3 Unifying the branches

Multimodal MAML (MMAML) (Vuorio et al., 2019) combines both branches. It trains a separate black-box meta-model, a recurrent network, to output a single vector z. Instead of directly using this vector as the parameters for the prediction task, it uses it to adapt an initialization θ. This is done by applying a number of simple single-layer decoders to the vector, the output of which each adapts a single layer of the network via a FiLM adaptation (Perez et al., 2017). Figure 2 shows a schematic representation, where the FiLM adaptation is denoted by $\circ$. From this initialization, a regular MAML procedure is performed on the task network $g_{\{\theta,z\}}$. This allows the combined models to learn an embedded representation of the task identity, used to nudge the initialization to a mode from which MAML can more easily fine-tune to the specific task. This allows the procedure to approximate a much wider range of optimal task-specific parameters ϕ_i^* with a neural network of limited depth.

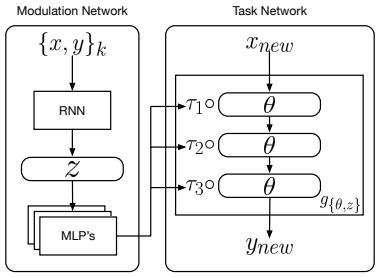


Figure 2: Schematic representation of Multimodal MAML. Image recreated from Vuorio et al. (2019).

2.4 Variational methods for meta-learning

While these methods show good empirical results, their approximations lose information about the uncertainty over predictions. Variational inference (Blundell et al., 2015) provides ways of maintaining these uncertainty considerations. It reframes the inference problem as an optimization problem by defining parameterized distributions and optimizing them to approximate intractable posteriors, by maximizing the Evidence Lower Bound (ELBO) (Blei et al., 2017):

$$\begin{aligned} \log p(\mathrm{D}) &= \mathrm{ELBO} + \mathrm{KL}(q_\psi(Z)||p(Z|\mathrm{D})), \\ \text{with ELBO} &= \mathbb{E}_q[\log p(\mathrm{D}, Z)] - \mathbb{E}_q \log q_\psi(Z) \\ &= \mathbb{E}_q[\log p(\mathrm{D}|Z)] - \mathrm{KL}(q_\psi(Z)||p(Z)). \end{aligned} \tag{5}$$

Here, ψ represents the variational parameters of the newly introduced distribution $q_\psi(Z)$ that are to be optimized, $\mathbb{E}_q$ represents the expectation over this distribution, and KL represents the Kullback-Leibler divergence (Kullback & Leibler, 1951). Maximizing this objective guarantees that the distribution $q(Z)$ approaches the intractable and unknown posterior $p(Z|D)$ (Blei et al., 2017).

3 Related work

A few previous methods have applied variational inference to meta-learning. First, Grant et al. (2018) propose an extension to MAML that uses a Laplace approximation of a Gaussian distribution centered around ϕ_i^*, calculated based on the curvature of the loss function at ϕ_i^*. This is a fairly crude approximation, and computationally expensive for larger models, but effective on smaller problems.

Amortized Bayesian Meta-Learning (ABML) (Ravi & Beatson, 2018) maximizes a lower bound to the model evidence from the hierarchical meta-learning formulation in (2). To optimize this lower bound, they introduce a variational distribution $q(\phi_i)$ over the posterior parameters, defined to be equal to the result of a fixed number of gradient descent steps from the prior parameters θ. By optimizing this with respect to θ, they obtain a method similar to the original MAML, but modeling each parameter as a mean and a variance. It therefore models the parameter space as independent Gaussian distributions, which is an oversimplification in theory, but tends to perform well in practice.

A third method, called PLATIPUS, does not treat the task specific parameters probabilistically, but uses a MAP estimate (Finn et al., 2018). As it learns a distribution via variational inference over the global parameters θ, repeated samples from this parameter space give rise to a distribution over ϕ_i as well, dependent on the initial conditions of the sample from θ. This results in a method akin to Hamiltonian Monte Carlo (Duane et al., 1987). In order to compensate for the fact that a MAP estimate is used for ϕ_i, the method additionally conditions the meta-parameters θ on the task specific data. It does so by applying MAML, and parameterizes the conditioned distribution through a single step of gradient descent so that samples from $p(\theta)$ are now samples from the task specific distribution $p(\theta_i) \sim \mathcal{N}(\mu_\theta - \nabla_\theta \mathrm{L}(\mathrm{D_i}), \sigma_\theta)$, where L is the loss function being optimized. This allows the parameters to be nudged in the right direction, before adding randomness through the sampling operation, and further fine-tuning towards samples of ϕ_i. Empirical results show good performance on ambiguous meta-learning tasks, indicating that the nudging of the prior allows the MAP estimation of ϕ_i to effectively cover the posterior distribution.

These prior methods are purely optimization based (Vuorio et al., 2019) and therefore don't generalize knowledge very well between tasks from multimodal task distributions. One exception is a continual meta-learner proposed by Jerfel et al. (2018), which deals with multimodal distributions by modelling the hyperparameters θ as a set of n different initializations, of which the final task specific parameter ϕ_i is considered to be a Dirichlet Mixture. This is conceptually identical to casting MAML as hierarchical Bayes (Grant et al., 2018), with a mixture distribution as a prior, instead of a Gaussian.

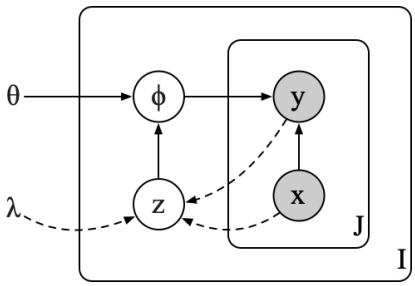


Figure 3: Probabilistic graphical model for the VTE-MAML algorithm. λ represents the variational parameters in the modulation network, z represent the variational encoding of the task identity. Solid arrows represent the probabilistic dependencies, and dashed arrows the variational approximations.

4 Method

We integrate variational inference into MMAML by introducing Variational Task Encoders for Model-Agnostic Meta-Learning (VTE-MAML). These Variational Task Encoders perform a nudging operation on θ, similar to PLATIPUS, except that the meta-parameters are modelled by an RNN, similar to MMAML, yielding improved coverage over multimodal task distributions (Vuorio et al., 2019). This yields the graphical model shown by the solid lines in Figure 3. Here, z_i represents a variational encoding of the task at hand. This is then decoded via a small decoder network for each layer in the network, the outputs of which are used to modulate θ. The MAML procedure is applied to this modulated initialization to find a MAP estimate ϕ_i^*. The structure of VTE-MAML is the same as in Figure 2, except that z is variational, requiring a novel objective that we derive next.

4.1 The VTE-MAML objective

To perform inference on this graphical model, we derive a variational optimization objective for the posterior $p(z_i|D_i)$. First, we express the full joint probability of the model as

$$p(D, z, \phi, \theta) = p(\theta) \prod_i [p(z_i)p(\phi_i|z_i, \theta)p(D_i|\phi_i)]. \tag{6}$$

where $p(\theta)$ is the prior over the meta-parameters θ, i indexes the tasks in our meta-dataset, $p(z_i)$ represent the priors over the variational task embeddings, $p(\phi_i|z_i, \theta)$ represents the task-specific parameters conditional on the task identity and the meta-parameters, and $\mathrm{D_i}$ is the dataset for task i. We derive the model evidence $p(D)$ by marginalizing over all other variables in the joint probability:

$$p(D) = \textstyle\int_\theta p(\theta) \prod_i \left[\int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right] d\theta. \tag{7}$$

Following the procedure for deriving the ELBO for hierarchical models from Ranganath et al. (2016), we introduce a variational distribution over the global parameters $q_\lambda(\theta)$:

$$\begin{aligned}
\log p(D) &= \log \int_\theta p(\theta) \prod_i \left[\int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right] \frac{q(\theta)}{q(\theta)} d\theta \\
&= \log \mathbb{E}_{q(\theta)} \left[\frac{p(\theta) \prod_i \left[\int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right]}{q(\theta)} \right] \\
&\geq \mathbb{E}_{q(\theta)} \left[\log p(\theta) + \sum_i \left[\log \int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right] - \log q(\theta) \right] \\
&= \mathbb{E}_{q(\theta)} \left[\sum_i \log \int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right] - \mathbb{E}_{q(\theta)} \left[\log \frac{q(\theta)}{p(\theta)} \right] \\
&= \mathbb{E}_{q(\theta)} \left[\sum_i \log \int_{z_i} p(z_i) \int_{\phi_i} p(\phi_i|z_i, \theta)p(D_i|\phi_i)d\phi_i dz_i \right] - \mathrm{KL}(q(\theta)||p(\theta)).
\end{aligned} \tag{8}$$

Algorithm 1 VTE-MAML Meta-Training

Input: Meta-dataset: D
Hyperparameters: Priors: $p(\theta) = \{\mu_\theta, \sigma_\theta\}$; $p(z) = \{\mu_{z_i}, \sigma_{z_i}\}_i$; Inner learning rate: α; Outer learning rate: β
Initialize θ, λ
repeat
  Sample batch of tasks $\mathcal{T} : D_i$
  for all D_i in $\mathcal{T}$ **do**
    $D_i = D_i^{train}, D_i^{test}$
    Sample $\epsilon \sim \mathcal{N}(0, 1)$
    Infer $(\mu_{z_i}, \sigma_{z_i}) = q_\lambda(z_i | D_i^{train})$
    Evaluate $\mathrm{KL}((\mu_{z_i}, \sigma_{z_i}) || p(z_i))$
    Reparameterize $z_i = \mu_{z_i} + \sigma_{z_i} * \epsilon$
    Modulate $\theta_i \leftarrow \mathrm{FiLM}(z_i, \theta)$
    Fine-tune $\phi_i \leftarrow \mathrm{SGD_k}(\theta_i; \alpha, D_i^{train})$
    Evaluate $-\log p(D_i^{test} | \phi_i)$
  end for
  $\theta \leftarrow \theta - \beta \nabla_\theta \sum_\mathcal{T} [-\log p(D_i^{test} | \phi_i)] - \log p(\theta)$
  $\lambda \leftarrow \lambda - \beta \nabla_\lambda \sum_\mathcal{T} [-\log p(D_i^{test} | \phi_i) + \mathrm{KL}((\mu_{z_i}, \sigma_{z_i}) || p(z_i))]$
until converged
return: θ, λ

By introducing the variational distribution, we can express the integral over θ as an expectation over our new distribution $q_\lambda(\theta)$. By applying Jensen's inequality in the third step, we recover the ELBO as originally derived (Blei et al., 2017). The exact procedure above can be repeated for the integral over z_i (Ranganath et al., 2016). This results in:

$$\begin{aligned} \log p(D) \geq & \mathbb{E}_{q(\theta)} \Big[\sum_i \mathbb{E}_{q(z_i)} [\log \int_{\phi_i} p(\phi_i | z_i, \theta) p(D_i | \phi_i) d\phi_i] \\ & - \mathrm{KL}(q(z_i) || p(z_i)) \Big] - \mathrm{KL}(q(\theta) || p(\theta)). \end{aligned} \tag{9}$$

$$\begin{aligned} & \log p(D) \geq \max_\lambda \mathbb{E}_{q(\theta)} \Big[\sum_i \mathbb{E}_{q(z_i)} [\log p(D_i | \phi_i^*)] \quad - \mathrm{KL}(q(z_i) || p(z_i)) \Big] - \mathrm{KL}(q(\theta) || p(\theta)) \\ & \text{with } \phi_i^* = \mathrm{SGD_k}(\theta; \alpha, \mathrm{D_i}) \approx_{\phi_i} p(\phi_i | z_i, \theta). \end{aligned} \tag{10}$$

$\mathrm{SGD_k}$ represents k steps of stochastic gradient descent with learning rate α, on task-specific data $\mathrm{D_i}$. This allows us to replace the intractable integral over ϕ_i with a simple evaluation at the MAP estimate in the objective. Similar to Finn et al. (2018), we assume that the probabilistic conditioning of the initialization on the task will provide enough variance over the initialization so that the MAP estimate will not significantly affect the uncertainty estimates, as in Hamiltonian Monte Carlo (Duane et al., 1987).

Finally, we assume that during meta-training we have a large number of tasks, with only very few data points per task, i.e. $I \gg J$. In this case, the posterior distribution over θ, approximated by variational distribution $q(\theta)$, will approach a Dirac delta function at $\theta^* =_\theta p(\theta | D)$ (Amit & Meir, 2018; Ravi & Beatson, 2018). This allows us to replace the expectation over $q(\theta)$ with a simple evaluation at θ^* and replaces the final KL-divergence term with $-\log p(\theta)$, equal to an L2-regularization norm in the case of a Gaussian prior. This leads to our final objective, here inverted to serve as a loss function to be minimized:

$$\mathcal{L}(\theta, \lambda; D) = \sum_i \Big[\mathbb{E}_{q(z_i)} [-\log p(D_i | \phi_i^*)] + \mathrm{KL}(q(z_i) || p(z_i)) \Big] - \log p(\theta). \tag{11}$$

4.2 The VTE-MAML algorithm

The full meta-training strategy is shown in Algorithm 1. The meta-testing strategy is analogous to MMAML and detailed in the supplementary materials. In this version of the algorithm, the KL is calculated analytically and the expectation of the likelihood over q, $\mathbb{E}_{q(z_i)}[-\log p(D_i | \phi_i^*)]$, is approximated based on a single sample, since previous work has shown no significant performance increase when using more samples (Blundell et al., 2015; Kingma & Welling, 2014).

This method has a number of advantages over existing MAML-based meta-learning methods. It reasons with uncertainty about the underlying tasks and task distribution, without making strong

assumptions about the distribution over the network weights ϕ_i. While VTE-MAML still uses a Gaussian variational distribution in the task embedding, the decoder networks can transform this distribution to an arbitrary shape before modulating the network parameters, which allows for correlations to exist between the distributions over model parameters. Secondly, the method is more distribution agnostic than existing probabilistic methods, as it is able to cover a flexible distribution over tasks, and reason appropriately about tasks that are outside the task distribution it was trained on. As Vuorio et al. (2019) have shown, modulating the network with a separate model-based modulation greatly improves the range of distributions covered by the meta-learner. Turning this modulation into a variational task encoding can be intuitively understood as modelling the identity of the task, that informs the model about some critical characteristics of the task at hand. Finally, while improving on these aspects, the meta-learning procedure is still model agnostic, and requires little additional computation. Further details are given in the supplementary materials.

5 Experimental setup

We evaluate VTE-MAML on a range of synthetic problems specifically designed to exhibit uncertainty over diverse task distributions, as well as standard few-shot learning benchmarks. To evaluate VTE-MAML's generalizability over multimodal task distributions, we compare against Multimodal MAML (MMAML) (Vuorio et al., 2019). To evaluate probabilistic properties, such as uncertainty calibration, we compare against PLATIPUS (Finn et al., 2018). We also compare against MAML (Finn et al., 2017) as a baseline.

5.1 Datasets

We reconstruct the synthetic multi-modal regression tasks used in Vuorio et al. (2019). They consists of five families of functions, generated with a range of possible parameters and added noise. In addition, we generate multi-modal classification tasks consisting of three modes, each with a range of permutations and added noise. The models will be trained on 6 data points, constituting a 2-way 3-shot classification task. We also evaluate all methods on the MiniImagenet benchmark (Vinyals et al., 2016), using the same canonical splits of 64 training, 12 validation, and 24 test classes to generate the n-way, k-shot tasks used to evaluate MAML, MMAML, and PLATIPUS in prior work.

5.2 Model details

Each algorithm was trained with identical underlying model architectures. Where possible, hyperparameters from previous literature were applied, along with a grid search around these parameters. For the synthetic tasks, the base model had four dense layers with 100 units and ReLU non-linearities. For the MiniImagenet task, the base model had 4 convolutional blocks with kernels of size 3 with 32 filters, followed by batch normalization, a ReLU non-linearity, and 2 x 2 max-pooling. The output is flattened and passed through a single dense layer to get the 5-way output required. Multimodal MAML and VTE-MAML also use a modulation network. For the synthetic tasks, we used two-layer bidirectional LSTMs with 40 hidden units to mirror the implementation by Vuorio et al. (2019). The networks used to decode the task encoding consist of three dense layers of 80 units each. For the MiniImagenet task, the encoders are convolutional neural networks with four layers with channel sizes 32, 64, 128, and 256, respectively, again mirroring Vuorio et al. (2019). The activations are fed through a final dense layer which transforms this vector into the final task encoding of size 128. These are decoded by 3-layer neural networks with 128 hidden units, beforer FiLM adaptation.

5.3 Performance metrics

To evaluate the accuracy of the uncertainty estimates, we evaluate the models based on Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) (Guo et al., 2017). These statistics estimate how much our confidence in the predictions may differ from the actual results of prediction. To evaluate whether our method is distribution agnostic, we train the meta-learners on a limited number of task modes, and evaluate them on the other modes. This allows us to observe how well the algorithms deal with tasks that are not identically distributed with the tasks it was trained on. All experiments are repeated ten times, and we report the mean along with the 95% confidence interval.

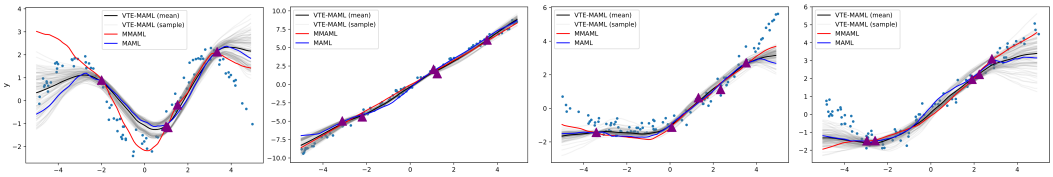


Figure 4: Qualitative results from the Multimodal Regression experiments with 5 modes. The red triangles represent the training examples, the blue dots the true distribution of the unseen test data. Many additional examples are provided in the additional materials.

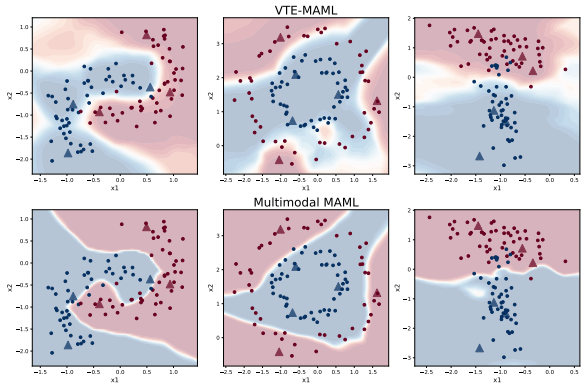


Figure 5: Illustration of classification tasks. Triangles show training points, dots show the true data distribution. Background colors represent model predictions.

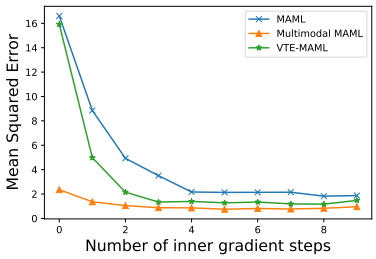


Figure 6: Convergence of MAML, Multimodal MAML and VTE-MAML after varying number of inner update steps. All three algorithms were trained with 5 inner update steps.

6 Results

6.1 Multimodal supervised learning performance

First, Figure 4 illustrates the behavior of the meta-learners on a few 5-mode regression tasks. Additional examples, as well as full details for reproducing all results can be found in the supplementary materials. For VTE-MAML, the sampled grey curves show the associated uncertainty, which seems to correspond well to the available training data. Full evaluation results are shown in Table 1 (left). In all tables, we indicate the best result and those within the confidence interval of the best result in bold. We find that VTE-MAML performs particularly well on unseen modes (function families not seen in the training tasks), while performing on par with with MMAML on 5-mode tasks. If we further reduce the multi-modality of the data to 3 modes (by using only 3 function families), VTE-MAML slightly under-performs the state of the art Multimodal MAML (MMAML) model. PLATIPUS has no explicitly modelled task identity, making it less able to adapt to multimodal task distributions.

Qualitative results on the multi-model classification tasks are shown in Figure 5. Lighter background colors represent higher uncertainty in the predictions. We observe that both Multimodal MAML and VTE-MAML give reasonable predictions. However, MMAML is overconfident in areas without training data, while VTE-MAML clearly shows increased uncertainty in areas where no training points exist. Full results are shown in Table 1. Once again, with fewer modes, MMAML is more accurate, but the difference disappears on more multi-modal tasks, and VTE-MAML outperforms all other methods on unseen modes. PLATIPUS has good performance in general, although it also degrades when the number of modes increases.

Table 1: Multimodal classification (left) and regression (right) results

Model	Mean Squared Error					
	3 modes		5 modes		Unseen modes	
MAML	1.35	±0.09	2.12	±0.22	4.33	±0.27
MMAML	**0.42**	±**0.04**	**0.90**	±**0.15**	4.52	±0.34
PLATIPUS	0.96	±0.12	1.80	±0.19	4.02	±0.31
VTE-MAML	0.76	±0.05	**1.10**	±**0.09**	**3.35**	±**0.29**

Model	Accuracy (%)					
	2 modes		3 modes		Unseen mode	
MAML	85.55	±0.13	82.40	±0.25	76.09	±0.42
MMAML	**86.58**	±**0.09**	**84.05**	±**0.30**	75.37	±0.49
PLATIPUS	86.02	±0.12	82.90	±0.33	76.03	±0.52
VTE-MAML	85.93	±0.20	**83.75**	±**0.38**	**77.90**	±**0.46**

Table 2: 1-shot 5-way MiniImagenet results.

Model	Accuracy (%)		ECE		MCE	
MAML	47.58	±0.13	5.75	±0.07	25.48	±0.52
MMAML	**49.90**	**±0.08**	7.10	±0.16	31.76	±0.92
PLATIPUS	**49.80**	**±0.45**	**4.10**	**±0.13**	21.15	±0.62
VTE-MAML	48.96	±0.14	**3.96**	**±0.12**	**17.42**	**±0.45**

Table 3: Calibration metrics on Multimodal classification task.

Model	2 modes		3 modes		Unseen mode	
	ECE	MCE	ECE	MCE	ECE	MCE
MAML	**2.34**	**7.93**	4.94	**13.69**	15.04	31.51
MMAML	7.06	24.30	11.07	44.87	14.65	34.54
PLATIPUS	3.21	**8.05**	**4.10**	**13.76**	14.90	29.31
VTE-MAML	4.16	13.27	**4.08**	**13.86**	**12.28**	**26.32**

6.2 MiniImagenet

The results of the MiniImagenet benchmark are shown in Table 2. VTE-MAML shows slightly degraded performance compared to both Multimodal MAML and PLATIPUS. This is somewhat expected since this task does not contain any explicit multimodal aspects, reducing the advantages of VTE-MAML over PLATIPUS. However, VTE-MAML does produce slightly better uncertainty estimates. We will now study this in more detail.

6.3 Uncertainty calibration

Table 3 shows the calibration metrics ECE and MCE on the classification tasks. This shows that VTE-MAML has substantially better calibration of uncertainty compared to MMAML. PLATIPUS provides better uncertainty estimates on tasks with very few modes, but since it is not explicitly designed with multimodal distributions in mind, this benefit disappears when additional modes are added to the problem. On tasks unseen during training, VTE-MAML estimates uncertainty better than the other models, and as shown in Table 1, it is more accurate on such tasks as well. Note that the original MAML algorithm also produces good calibration of uncertainty, although mainly on tasks with very few modes. Looking at the results on the real-world but less multi-modal MiniImagenet task (Table 2), VTE-MAML also shows improved calibration compared to MAML and MMAML. Compared to PLATIPUS, it has similar ECE scores and slightly better MCE scores.

6.4 Behavior of the variational task encoding

Figure 6 displays the Mean Squared Error scores on the 5-mode regression tasks for MAML, Multimodal MAML and VTE-MAML, in relation to the number of inner gradient steps performed. Without gradient updates, VTE-MAML performs similar to MAML, which has no modulation network. This indicates that the modulation network in VTE-MAML places the modulated parameters in a local maximum with high curvature, so that fewer gradient updates result in improved performance. This does raise the question of whether the modulation network is doing anything significant at all. There are a few pieces of evidence that suggest otherwise. First, the performance on most multimodal tasks improves with the use of the variational modulation network. Second, the functions resulting from VTE-MAML show clearly increased variance in areas where fewer data points are provided. If the algorithm learned to ignore the modulation network, this would not occur. Third, we see in Figure 6 that the modulation network helps VTE-MAML to converge faster than MAML.

7 Conclusion

The approach presented in this paper improves on model-agnostic meta-learning (MAML) methods by making them reason with uncertainty about the target tasks, and making them more robust to dealing with more disparate and multimodal distributions. It does this by modelling the identity of tasks as a dense variational embedding vector. By adapting the initialization of MAML with decoded samples from the variational embedding, we approximate hierarchical Bayesian inference. As such, this method combines the benefits of two state-of-the-art techniques: MultiModal MAML and PLATIPUS. We empirically observe that the procedure covers more divergent task distributions, while maintaining more reasonable estimates of uncertainty over the tasks. It is more honest about its generalization to tasks that it has not seen in training, and makes more reasonable predictions on these tasks. This can be achieved with relatively few additional computations, and without making strong assumptions about the distributions of the neural network weights, remaining model-agnostic.

References

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

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 9516–9527, 2018.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

Jerfel, G., Grant, E., Griffiths, T. L., and Heller, K. Reconciling meta-learning and continual learning with online mixtures of tasks. *arXiv preprint arXiv:1812.06080*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2014.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Kullback, S. and Leibler, R. A. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL `https://doi.org/10.1214/aoms/1177729694`.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International Conference on Machine Learning*, pp. 324–333, 2016.

Ravi, S. and Beatson, A. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

Schmidhuber, J. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL `http://www.idsia.ch/~juergen/diploma.html`.

Schmidhuber, J., Zhao, J., and Wiering, M. Simple principles of metalearning. *Technical report IDSIA*, 69:1–23, 1996.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pp. 1–12, 2019.

Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. CAML: fast context adaptation via meta-learning. *CoRR*, abs/1810.03642, 2018. URL `http://arxiv.org/abs/1810.03642`.

Variational Task Encoders for Model-Agnostic Meta-Learning: Supplementary Material

1. Meta-Testing Algorithm pseudo code

Algorithm 1 shows the meta-testing, or fine-tuning, procedure for VTE-MAML on new tasks. This is straightforwardly derived from the training algorithm presented in the Methods section. In this algorithm D_i represents the new dataset, θ represents the learned meta-parameters of the model, λ represents the learned variational parameters of the modulation network to model the variational distribution $q_\lambda(z_i|D_i)$.

Algorithm 1 VTE-MAML Meta-Testing

Input: New task dataset: D_i, Learned: θ, λ
Hyperparameters: Inner learning rate: α
Infer $(\mu_{z_i}, \sigma_{z_i}) = q_\lambda(z_i|D_i)$
for each sample of ϕ_i **do**
  Sample $\epsilon \sim \mathcal{N}(0, 1)$
  Reparameterize $z_i = \mu_{z_i} + \sigma_{z_i} * \epsilon$
  Modulate $\theta_i \leftarrow \text{FiLM}(z_i, \theta)$
  Fine-tune $\phi_i \leftarrow \text{SGD}_\text{k}(\theta_i; \alpha, D_i)$
end for
return: all samples of ϕ_i

2. Experiment Details

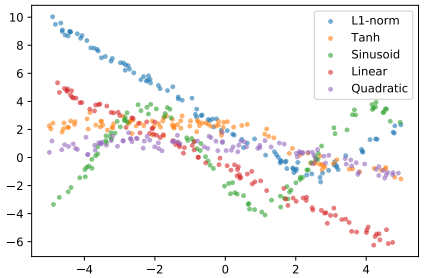
This section contains details on how we constructed the synthetic multimodal tasks used in our experiments. The regression tasks are reconstructed from Vuorio et al. (2019) to allow a direct comparison.

2.1. Multimodal Regression

The training tasks for the Multimodal Regression meta-dataset are generated on the fly during training and testing and correspond exactly to the functions used in Vuorio et al. (2019)[1]. These tasks are sampled randomly from one of the function families, with equal likelihood for each function family. The function families for the 3-modal regression tasks are Sinusoidal, Linear and Quadratic functions. For the 5-modal regression tasks additionally the L1-norm and Tanh functions are added. For the experiments performed on unseen modes, the model is trained on the first three modes, and evaluated on the last two. For each of the regression tasks, the input values x are sampled uniformly from the interval $[-5, 5]$. Each of the functions also has added noise ϵ, which is sampled for individual data points from a Gaussian distribution $\mathcal{N}(0, 0.3)$. Figure 1 shows an illustration of these function families.

[1]This does not exactly correspond to the description in the paper by Vuorio et al. (2019), as their supplementary materials contain minor errors. Instead, these values correspond to the ones used in their experiments, as can be inspected in their source code.

Figure 1. Illustration of the multimodal regression task, with 100 data points generated per function for illustrative purposes. In reality, the algorithm will be trained on a 5-shot regression task.



Sinusoidal functions are generated according to the following formula:

$$y = A * \sin(x + b) + \epsilon,$$

where $A \in [0.1, 5]$, and $b \in [0, 2\pi]$. Linear functions are generated according to the formula:

$$y = A * x + b,$$

where $A \in [-3, 3]$, and $b \in [-3, 3]$. Quadratic functions are generated according to the formula:

$$y = A * (x - c)^2 + b,$$

where $A \in [-0.15, -0.02] \cup [0.02, 0.15]$, $b \in [-3, 3]$, and $c \in [-3, 3]$. L1-norm functions are generated according to the formula:

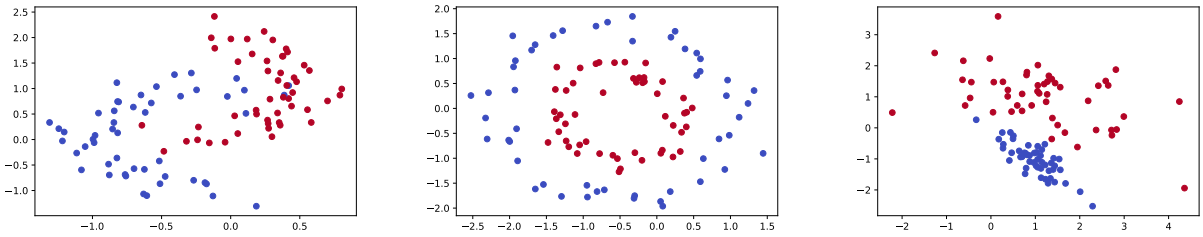
$$y = A * |x - c| + b,$$

where $A \in [-3, 3]$, $b \in [-3, 3]$, and $c \in [-3, 3]$. The Tanh functions are generated according to the formula:

$$y = A * \tanh(x - c) + b,$$

where $A \in [-3, 3]$, $b \in [-3, 3]$, and $c \in [-3, 3]$.

Figure 2. Illustration of the three modes of the classification tasks, showing the intertwined half moons, concentric circles and linear separation respectively. Showing 100 data points per task for illustrative purposes. In reality, the algorithm is trained on 6 data points



2.2. Multimodal Classification

The multimodal classification tasks are generated during training and testing as well, and belong to three distinct modes. The 2-modal classification task consists of intertwined half moons and concentric circles. The 3-modal classification task additionally has a linearly separable classification task added. Each of these datasets can be straightforwardly generated by using the Scikit-learn python package (Pedregosa et al., 2011), which implements these as functions in the dataset module. These functions are $\mathrm{make\_moons}, \mathrm{make\_circles}$ and $\mathrm{make\_classification}$ respectively. With the following parameters. For $\mathrm{make\_moons}$, $\mathrm{noise} = 0.2$. For $\mathrm{make\_circles}$, $\mathrm{distance} = 0.5$ and $\mathrm{noise} = 0.1$. For $\mathrm{make\_classification}$, $\mathrm{n\_features} = 2, \mathrm{n\_informative} = 2$ and $\mathrm{n\_redundant} = 0$.

Since $\mathrm{make\_moons}$ and $\mathrm{make\_circles}$ generate identical datasets each time, apart from the noise, additional translations are applied to these functions to make sure that each task is distinct. The intertwined half moons are rotated by sampling an angle $\alpha \in [0, 2\pi]$ and multiplying the input values $\{x_1, x_2\}$ with the rotation matrix R corresponding to that angle. The concentric circle tasks are made unique by scaling and translating the input values $\{x_1, x_2\}$ according to the following formula:

$$x_1^* = A_1 * x_1 + b_1$$
$$x_2^* = A_2 * x_2 + b_2,$$

where where $A_1, A_2 \in [0.5, 2.5]$, and $b_1, b_2 \in [-2.5, 2.5]$

3. Model Implementation and Training

For the regression tasks each algorithm uses a meta-batch size of 125 tasks for the 5-modal regression, and 75 tasks for the 3-modal regression, ensuring that each task batch has 25 tasks from each mode. Each algorithm performs five inner update steps during training and ten inner update steps during validation and evaluation. VTE-MAML requires priors to be defined for the distributions $p(\theta)$ and $p(z_i)$, as discussed in section 7.1, this choice does not seem to significantly impact the model performance, so for each experiment simple unit Gaussians were used. For the classification tasks, the hyperparameters used were identical to those used for regression, with the exception that for each classification task the meta-batch size was kept at 125 tasks, irrespective of the number of modes.

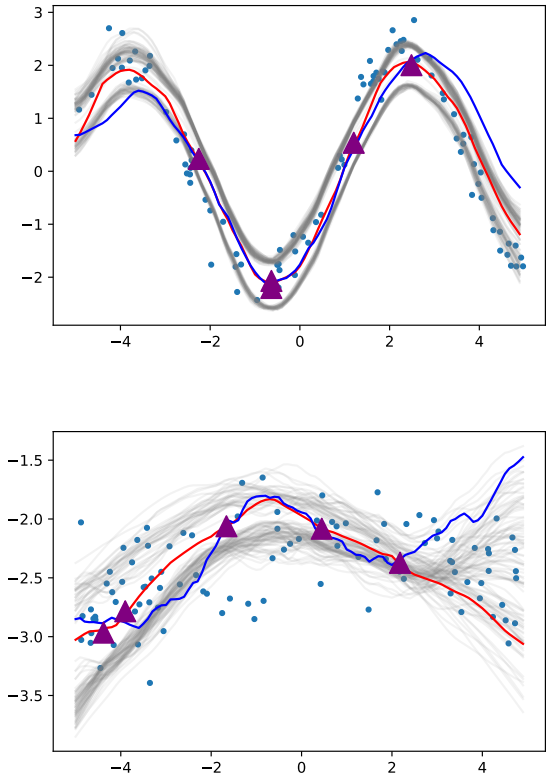
Table 1. Learning rates (LR) used in experiments.

Task	**Model**	Inner LR	Outer LR
Regression	MAML	0.01	0.001
	MMAML	0.01	0.001
	PLATIPUS	0.001	0.001
	VTE-MAML	0.001	0.001
Classification	MAML	0.01	0.001
	MMAML	0.01	0.001
	PLATIPUS	0.01	0.001
	VTE-MAML	0.001	0.001
MiniImagenet	MAML	0.01	0.005
	MMAML	0.01	0.005
	PLATIPUS	0.001	0.001
	VTE-MAML	0.001	0.0005

For MiniImagenet, the models were trained on meta-batches consisting of 4 tasks, as larger meta-batch sizes ran into memory limitations. Once again, the models were updated for 5 inner steps during training, and 10 inner steps during evaluation and validation.

The most significant hyperparameters that were iterated on during training were the inner and outer learning rates. The inner learning rate corresponds to the learning rate used for the fine-tuning of the global parameters θ via SGD to result in the task-specific parameters ϕ_i. The outer learning rate corresponds to the learning rate that is used during meta-training to optimize the global parameters θ, and in the case of VTE-MAML, also λ. The learning rates used for each of the models on each of the experiments were established using a grid search, and the final used values are shown in Table 1.

Figure 3. Illustration of the "split decision" artefact that can be observed occasionally in regression tasks



4. Additional results and analysis

This section provides additional results, and dive deeper into some of the observations reported in the paper.

4.1. Qualitative Samples

Figure 4 shows an extensive range of sampled multi-model regression tasks and the results for VTE-MAML, MAML, and MMAML. These are the results for the 5-mode experiments, and every column shows examples of the result for one specific mode: sinusoid, linear, quadratic, L1, and tanh, respectively. Purple triangles represent the training points, blue dots the true data distribution. The blue line and red line represent the MAML and Multimodal-MAML predictions, respectively.

Grey lines represent samples from VTE-MAML (ours), showing that our method indeed provides accurate uncertainly estimates on multi-modal tasks. As explained in the paper, occasionally some overconfident predictions are made, especially when the few provided training points suggest that a simpler model can be used.

4.2. Bimodal posterior examples

As discussed in the paper, we occasionally observed that the non-probabilistic fine-tuning step occasionally causes a collapse to a bimodal posterior, or a 'split decision' between two alternative functions which are close to the mean. Figure 3 shows a few extreme cases where this was observed.

4.3. Model calibration details

Figure 5 shows model calibration details for VTE-MAML, MAML, and MMAML. Ideally, the histograms should correspond to the diagonal. Red bars below diagonal indicate overconfidence in positive label. Grey bars in overlay indicate the fraction of predictions that fall within the corresponding bucket. The top row shows results for the 3-mode classification experiments, the second row (unseen mode) shows models trained on 2 modes and tested on the held out mode, while the last row shows the results on MiniImageNet.

Overall, we can conclude that VTE-MAML produces much more accurate uncertainty estimates, although some overconfidence can still be observed.

5. Computational Complexity

In terms of computational complexity during training, VTE-MAML is broadly comparable to Multimodal MAML. The majority of the computational complexity arises from backpropagation through the inner update steps of the MAML procedure. Multimodal MAML and VTE-MAML add an additional modulation network, which leads to a roughly 25% increase in training time in our experiments, compared to non-multimodal alternatives.

Note that, since all uniquely probabilistic properties of VTE-MAML are relegated to this modulation network, the MAML procedure could be replaced with a more computationally efficient alternative such as REPTILE (Nichol et al., 2018) or iMAML (Rajeswaran et al., 2019) if a performance increase is desired. This does however constitute further approximations of the probabilistic objective, which may affect the quality of the uncertainty estimated. Such a faster implementation and the resulting speed-accuracy trade-off was not yet evaluated in this work.

During training, VTE-MAML does not markedly differ in complexity from Multimodal MAML, as only a single sample from the Variational Task Encoder is required for optimization, and the KL-divergence can be straightforwardly calculated analytically. However, when fine-tuning, our algorithm does require separate fine-tuning for each desired sample from the model. In practice, this is rarely problematic, as each task only requires a very limited number of gradient updates in the fine-tuning procedure.

Figure 4. Randomly sampled multi-model regression tasks and the results for VTE-MAML, MAML, and MMAML.

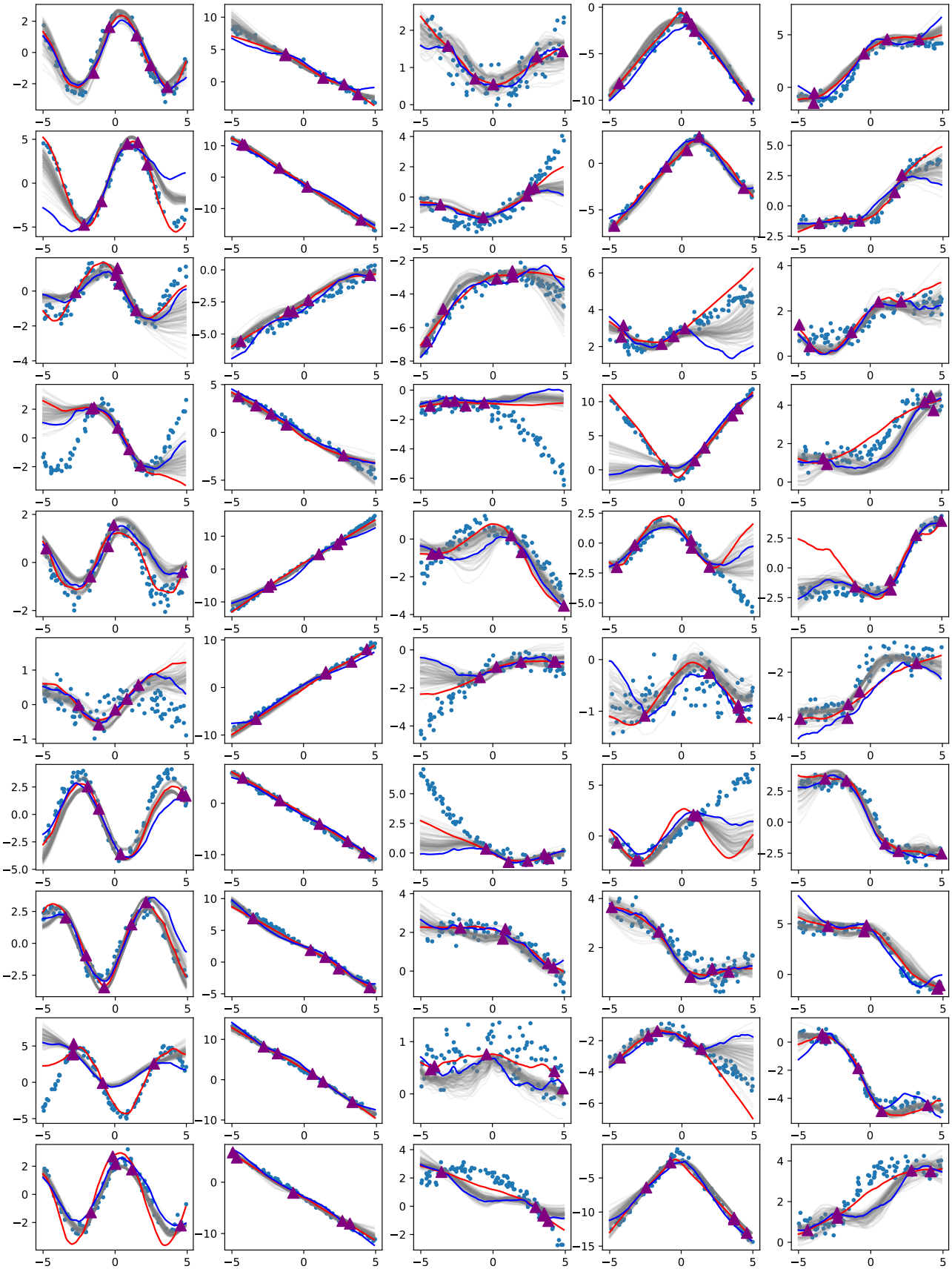
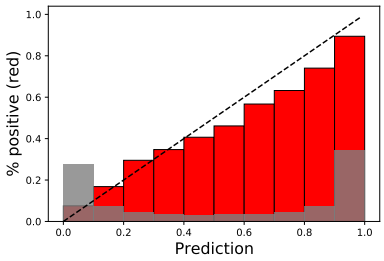
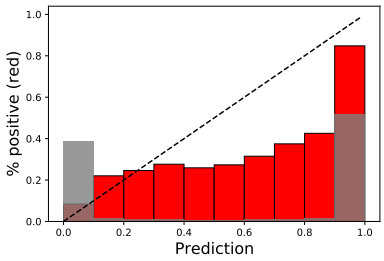


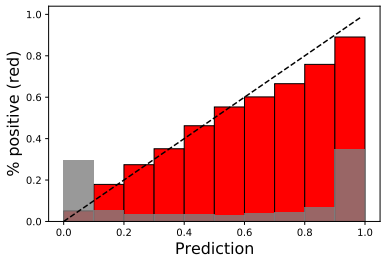
Figure 5. Model calibration on Multimodal classification and MiniImagenet tasks. Each bucket contains at least 500 predictions. Top row shows calibrations for multimodal classification with three modes. Middle row shows calibration on unseen mode of classification task. Bottom row shows calibration on MiniImagenet.



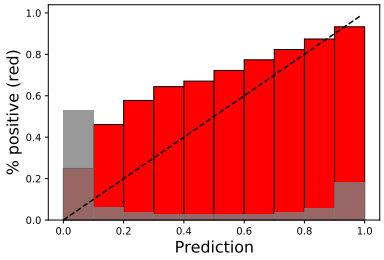
(a) MAML classification.



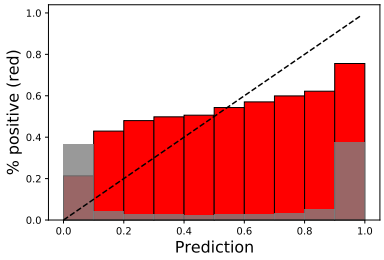
(b) Multimodal MAML classification



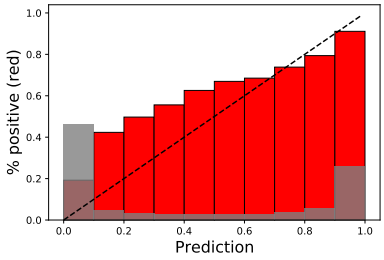
(c) VTE-MAML classification



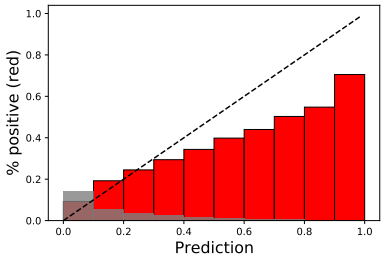
(d) MAML unseen mode



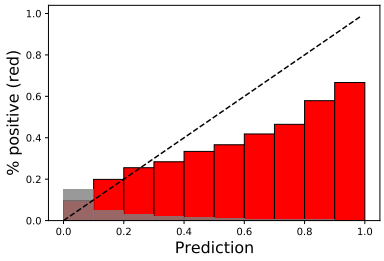
(e) Multimodal MAML unseen mode



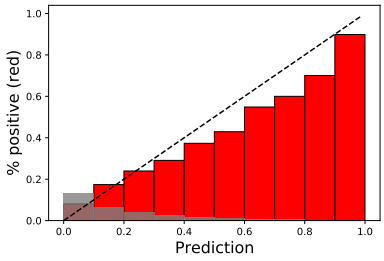
(f) VTE-MAML unseen mode



(g) MAML MiniImagenet



(h) Multimodal MAML MiniImagenet



(i) VTE-MAML MiniImagenet

The MiniImagenet experiments in this work were trained on a single NVIDIA RTX 2070 Super GPU for approximately 12 hours. The experiments on the synthetic classification and regression datasets did not experience any significant speedup from a GPU, due to the moderate input and network size. These were therefore trained on a four-core CPU, again for approximately 12 hours.

6. Reproducibility

All results can be reproduced using the provided code, which will be available on GitHub after the reviewing phase.

The README provides instructions for downloading or generating the used datasets, as well as the commands to re-train all the models. Already trained models are also provided. The 'Experiments' folder also includes a set of Jupyter notebooks that can be run to reproduce all of the figures shown in the paper and the additional materials.

References

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rajeswaran, A., Finn, C., Kakade, S., and Levine, S. Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*, 2019.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pp. 1–12, 2019.