# FEW-SHOT LEARNING WITH WEAK SUPERVISION

**Ali Ghadirzadeh**[*]**, Huaxiu Yao & Chelsea Finn**
Stanford University {ghadiri,huaxiuyao,cbfinn}@stanford.edu

**Petra Poklukar**[*]**, Xi Chen, Hossein Azizpour, Mårten Björkman & Danica Kragic**
KTH Royal Institute of Technology {poklukar,xi8,azizpour,celle,dani}@kth.se

## ABSTRACT

Few-shot meta-learning methods aim to learn the common structure shared across a set of tasks to facilitate learning new tasks with small amounts of data. However, provided only a few training examples, many tasks are ambiguous. Such ambiguity can be mitigated with side information in terms of weak labels which is often readily available. In this paper, we propose a Bayesian gradient-based meta-learning algorithm that can incorporate weak labels to reduce task ambiguity and improve performance. Our approach is cast in the framework of amortized variational inference and trained by optimizing a variational lower bound. The proposed method is competitive to state-of-the-art methods and achieves significant performance gains in settings where weak labels are available.

## 1 INTRODUCTION AND RELATED WORK

A critical issue in few-shot meta-learning problems is the lack of sufficient information contained in the training examples to uniquely determine the task. Probabilistic meta-learning algorithms address this task ambiguity problem by learning generalizable priors from a set of related tasks, and leveraging Bayesian inference to generate several potential neural network solutions to a given ambiguous task (Finn et al., 2018; Gordon et al., 2019; Yoon et al., 2018; Rusu et al., 2019; Grant et al., 2018; Ravi & Beatson, 2019). However, existing methods often fail to cover all possible solutions because of the complexity and multi-modality of their distribution. Moreover, there is no mechanism for a practitioner to provide extra information to only produce certain types of solutions. One way to mitigate these issues is by providing side information in terms of weak labels (Denevi et al., 2020). These labels are often either readily available or can be easily collected with little efforts.

We introduce a probabilistic meta-learning framework that enables efficient integration of weak labels. Our method, called variational model-agnostic meta-learning (VMAML), exploits the weak labels to structure a low-dimensional task latent space of a task embedding trained using amortized variational inference (AVI) (Ravi & Beatson, 2019). Given a few-shot task, the embedding assigns a Gaussian distribution over the latent space endowed with a rejection sampling mechanism that rejects samples that are inconsistent with the weak labels. In this way, we mitigate the multi-modality of the distribution over the neural network solutions. Furthermore, this mechanism provides a way to flexibly choose whether or not to exploit the weak labels at meta-test time. As shown in Figure 1, VMAML extends MAML (Finn et al., 2017; Rajeswaran et al., 2019) into a *conditional* and *probabilistic* meta-learning algorithm. It obtains initial parameters for the gradient-descent update per input task instead of a globally-shared initialization (Wang et al., 2020), and finds such initializations stochastically instead of deterministically.

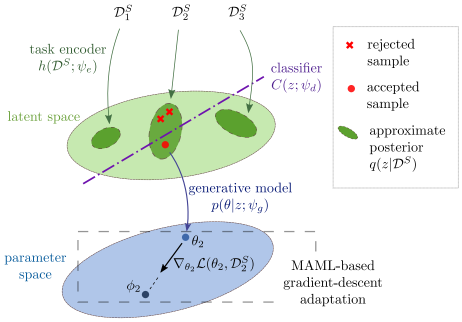

Figure 1: An overview of the proposed VMAML framework discussed in Section 3.

[*] Petra Poklukar and Ali Ghadirzadeh have contributed equally to this work

The main contribution of this work is a framework for providing a principled and practical solution for handling few-shot task ambiguities by exploiting task embedding and weak supervision. We experimentally evaluate VMAML performance on an ambiguous few-shot regression and classification tasks, and demonstrate that it achieves competitive performance with state-of-the-art methods without using weak labels and superior performance when additionally provided with weak labels.

## 2 The few-shot learning setting

We consider a few-shot learning task $\mathcal{T}$ to be sampled from an unknown task distribution $p(\mathcal{T})$, and characterized by a distinct dataset $\mathcal{D} = \{(x_i, y_i, l) : x_i \in X, y_i \in Y, l \in L\}_{i=1}^n$ with $n$ independent and identically distributed input-output pairs $(x_i, y_i)$ and a weak label $l$ represented by a multi-hot vector. We denote by $\phi$ the parameters of the task-specific neural network $f(x; \phi)$ that outputs a prediction $\tilde{y}$ for a given input $x$. We focus on conditional and optimization-based meta-learning algorithms that, given on a few-shot set $\mathcal{D}$, first output *initial* parameters $\theta$ of the neural network $f(x; \theta)$. These are then further optimized using a task-specific loss function $\mathcal{L}(\theta, \mathcal{D})$ to obtain the final parameters $\phi = \arg\min_\theta \mathcal{L}(\theta, \mathcal{D})$ used to solve the task $\mathcal{T}$.

The objective of a meta-learning algorithm, or a meta-learner, is to efficiently solve *unseen* tasks, sampled from $p(\mathcal{T})$, provided with a set of few data points. The meta-learner is trained during *meta-train* phase where a nested optimization is performed at two levels: *task-level* (inner loop) and *meta-level* (outer loop). At each training iteration, we first sample a batch of tasks $\{\mathcal{T}_t\}$ represented by above defined datasets $\{\mathcal{D}_t\}$, each consisting of a *support set* $\mathcal{D}_t^S$, also known as the few-shot set, and a *query set* $\mathcal{D}_t^Q$ such that $\mathcal{D}_t = \mathcal{D}_t^S \sqcup \mathcal{D}_t^Q$. For each task $\mathcal{T}_t$ in the batch, the inner loop optimizes the meta-learner's output $\theta_t$ (initial parameters) using the support set $\mathcal{D}_t^S$ to yield the final task-specific parameters $\phi_t$. At the meta-level (the outer loop), the meta-model's parameters themselves are updated by optimizing the sum of the task-specific losses on the query samples $\mathcal{D}_t^Q$. Intuitively, the meta-learner first adapts each task using the support set $\mathcal{D}_t^S$ and then optimizes for few-shot generalization based on how well the adapted model generalizes to new data points from the query set $\mathcal{D}_t^Q$. During the meta-test phase, a trained meta-learner is evaluated on a set of held-out novel tasks $\mathcal{T}_u \sim p(\mathcal{T})$ that were not used during the meta-train phase. The model is given a few-shot support set $\mathcal{D}_u^S$ corresponding to the new task $\mathcal{T}_u$ and outputs the initial task-parameters $\theta_u$ that are again further optimized to yield the parameters $\phi_u$ of the task-specific model $f(x; \phi_u)$. The obtained model $f(x; \phi_u)$ is then used to predict labels for a set of *unlabeled* inputs $\{x_i \in X\}$ for the task $\mathcal{T}_u$. The performance of the obtained solution $\phi_u$ is evaluated on an unseen but *labeled* query set $\mathcal{D}_u^Q$ using the task-specific loss function $\mathcal{L}(\phi_u, \mathcal{D}_u^Q)$.

## 3 Variational Model-Agnostic Meta-Learning (VMAML)

We extend MAML to a *conditional* and *probabilistic* meta-learning algorithm that can incorporate weak labels to mitigate the task ambiguity problem inherent in few-shot learning settings. VMAML, visualized in Figure 1, consists of three neural networks: (1) a *task encoder* $h(\mathcal{D}^S; \psi_e)$, (2) a *generative model* $p(\theta|z; \psi_g)$ and (3) a *classifier* $C(z; \psi_d)$. Their individual network parameters $(\psi_e, \psi_g, \psi_d)$ make up the parameters of the meta-model that is trained in a nested optimization scheme containing inner and outer loops similar to Section 2.

**VMAML Inner Loop Adaptation** The goal of inner loop adaptation procedure is to obtain task-specific parameters $\phi_t$ provided the values of the meta-learner's parameters $(\psi_e, \psi_g, \psi_d)$ and a task support set $\mathcal{D}_t^S$. First, the task encoder $h(\mathcal{D}_t^S; \psi_e)$ takes the support set $\mathcal{D}_t^S$ as input and outputs parameters of a low-dimensional task latent distribution $q(z|\mathcal{D}_t^S)$ over a task latent variable $z$. We refer to it as the approximate posterior distribution and model it as a multivariate diagonal Gaussian $q(z|\mathcal{D}^S) = \mathcal{N}(z; \mu_z, \mathrm{diag}(\sigma_z^2))$ represented by mean vector $\mu_z$ and covariance vector $\sigma_z^2$.

We incorporate weak labels $l$ into the learning procedure and learn similarities among the encodings $z$ based on the attributes provided by $l$. We achieve this by endowing the approximate posterior $q$ with a rejection mechanism based on a shallow classifier $C(z; \psi_d)$. The classifier $C$ takes a latent sample $z$ as an input and outputs softmax probabilities over the weak labels. If the classifier's prediction matches the weak label $l$, the sample $z$ is accepted and further processed by the rest of the network. In the opposite case, a new sample is drawn from $q$ and the process is repeated. In this way, ensure that the latent sample that is further processed by the meta-learner is consistent with the provided weak label. The proposed mechanism both improves the expressiveness of $q$ and mitigates the multi-modality of the true distribution over possible solutions.

The accepted latent sample $z$ is then passed to the generative model $p(\theta|z;\psi_g)$ to generate a task-specific neural network initialization $\theta_t$. Following the inner loop of MAML, the generated parameters $\theta_t$ are further optimized by one gradient-descent step on the same support set $\mathcal{D}_t^S$ which results in the final task-specific parameters $\phi_t$. More precisely, we obtain the final task parameters $\phi_t = \theta_t - \alpha\nabla_\theta\mathcal{L}(\theta_t, \mathcal{D}_t^S)$ where $\alpha$ denotes the learning rate. The final solution of the inner loop adaptation for a learning task $\mathcal{T}_t$ is the network $f(x;\phi_t)$ which can be used to make queries on unlabeled inputs. In contrast to prior work (Rusu et al., 2019; Gordon et al., 2019), the distribution over $\phi_t$ is modelled implicitly, and therefore not constrained to be Gaussian.

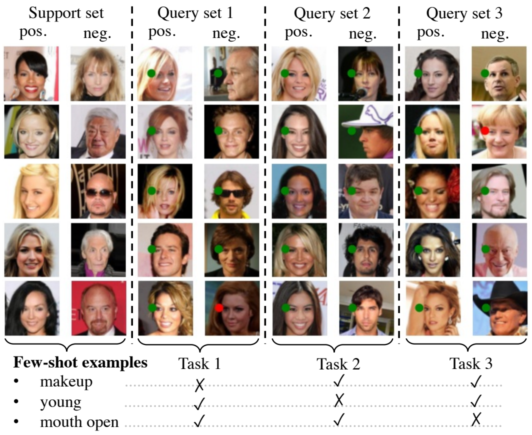

Figure 2: Visualization of classification of three different classifiers sampled from a trained VMAML-*info*. Left two columns show the support set $\mathcal{D}_u^S$, while other columns show images for each of the three ambiguous modes of the tasks. Each image is marked with either green dot if the predicted classification was correct or red otherwise.

**VMAML Outer Loop Optimization** The goal of the outer loop optimization is to update the parameters of the meta-learner $(\psi_e, \psi_g, \psi_d)$ such that the adaptation procedure produces task parameters $\phi_t$ that generalize to the corresponding query set $\mathcal{D}_t^Q$. The meta-learner is trained based on two components: (i) a variational lower bound to find the approximate posterior distribution $q$ of the task latent variable $z$, and (ii) a cross-entropy term to update the classifier $C$, by optimizing the objective

$$\min_{\psi_e,\psi_g,\psi_d} \sum_t \mathbb{E}_{\substack{z_t\sim q(z|\mathcal{D}_t^S)\\ \theta_t\sim p(\theta|z_t;\psi_g)}} \left[\mathcal{L}(\phi_t, \mathcal{D}_t^Q)\right] - \beta D_{\mathrm{KL}}(q(z|\mathcal{D}_t^S) \,\|\, p(z)) - l_t * \log C(z_t;\psi_d), \quad (1)$$

where $\phi_t = \theta_t - \alpha\nabla_\theta\mathcal{L}(\theta_t, \mathcal{D}_t^S)$, $D_{\mathrm{KL}}$ denotes the Kullback–Leibler (KL-) divergence, $p(z)$ is the prior distribution modelled as a zero-mean and unit variance Gaussian distribution, $\beta$ is a parameter to balance the KL-loss against the task loss, and $*$ represents the scalar product of two vectors. The first two terms in equation 1 represent the variational lower bound (i), while the last term represents the cross-entropy update (ii) for the classifier $C$. The MAML-based gradient-descent step that adapts the task-specific parameters $\theta_t$ is absorbed into the generative model $p$ in the first term of equation 1.

## 4 Experiments

We evaluated VMAML on the ambiguous version of CelebA few-shot attribute classification, and an extended version of the ambiguous few-shot multi-modal 1D regression problem, both introduced by Finn et al. (2018). The details of the tasks can be found in the supplementary materials. We investigated the importance of weak labels to mitigate the task ambiguity and the effect of the rejection sampling mechanism by training VMAML under three conditions: (i) without accessing the weak labels (VMAML-*wo*), (ii) providing the weak labels as an input to the meta-learner (VMAML-*info**), and (iii) using the rejection sampling mechanism in addition to (ii) (VMAML-*info*). We compared VMAML's performance to the state-of-the-art methods MAML (Finn et al., 2017), PLATIPUS (Finn et al., 2018), VERSA (Gordon et al., 2019), and LEO (Rusu et al., 2019). In Figure 3, we show qualitative performance of VMAML trained both without and with access to the weak labels $l$. In the left part of the image, we visualize the ground truth by green shaded area. We can see that VMAML-*wo* (top row) outputs a variety of functions that are possible solutions for the given set of support points. When provided with the weak labels $l$, VMAML-*info* (bottom row) outputs significantly improved predictions where the sampled functions match the provided weak labels. In the bottom right part of Figure 3 weak labels are used as the user's input to produce specific types of solutions, i.e., *sinusoidal*, *polynomial* and *linear*, while keeping both the parameters of VMAML-*info* and training examples fixed, and changing only samples from the posterior distribution $q(z|\mathcal{D}^S)$

| Method | Precision Error | Recall Error |
|---|---|---|
| MAML | 0.627 ± 0.008 | 0.956 ± 0.009 |
| PLATIPUS | 0.751 ± 0.009 | 0.244 ± 0.001 |
| LEO | 0.171 ± 0.013 | 0.230 ± 0.010 |
| VMAML-*wo* | 0.184 ± 0.002 | 0.249 ± 0.003 |
| VMAML-*info*$^*$ | 0.170 ± 0.002 | 0.247 ± 0.002 |
| VMAML-*info* | **0.095 ± 0.001** | **0.228 ± 0.003** |

Table 1: Results of the 5-shot regression task.

| Method | Accuracy | Coverage ($\leq$ 3) |
|---|---|---|
| MAML | $\mathbf{89.7 \pm 2.1\%}$ | $1.00 \pm 0.00$ |
| PLATIPUS | $87.8 \pm 0.3\%$ | $1.75 \pm 0.02$ |
| LEO | $83.4 \pm 0.3\%$ | $1.52 \pm 0.02$ |
| VERSA | $87.4 \pm 0.4\%$ | $1.04 \pm 0.01$ |
| VMAML-*wo* | $85.4 \pm 1.8\%$ | $1.25 \pm 0.06$ |
| VMAML-*info*$^*$ | $86.2 \pm 0.9\%$ | $1.37 \pm 0.08$ |
| VMAML-*info* | $84.2 \pm 1.3\%$ | $\mathbf{2.34 \pm 0.04}$ |

Table 2: Results of the CelebA task.

that pass the rejection sampling mechanism. In this way, the behavior of VMAML models can be controlled by the value of the weak label $l$.

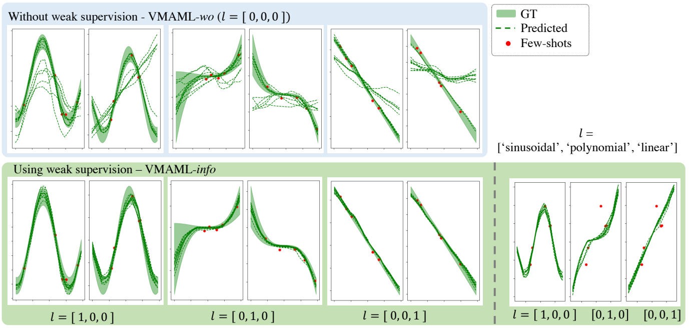

Figure 3: Predictions made by a VMAML-*wo* (top left) and VMAML-*info* (bottom left) on the same support samples (red dots). The green shaded area represents the ground truth. In the bottom right, weak labels are used to generate specific solutions.

We quantitatively evaluated the models based on two introduced measures: *precision error* and *recall error*. For a given test few-shot task $\mathcal{T}_u$, we generate 100 solutions for each method. We calculate the mean absolute error (MAE) of each of the generated solutions and all ground truth solutions. For each task, we choose the minimum MAE. The precision error is then defined as the average over all minimum MAEs across the test tasks. Similarly, the recall error is found by choosing 100 distinct solutions from the ground truth set and calculating the average over minimum MAEs determined between the sampled ground truth solutions and all 100 generated solutions. Table 1 shows precision and recall errors for VMAML models as well as the benchmark methods. VMAML-*info* attains the best precision and recall among all evaluated methods which demonstrates its ability to exploit weak labels while reasoning about possible solutions. Since the performance of VMAML-*info*$^*$ is only slightly improved compared to that of VMAML-*wo* but significantly lower than that of VMAML-*info*, we conclude that the proposed rejection sampling mechanism in the latent space beneficially exploits the provided weak labels $l$. We observe that LEO performs comparable to VMAML, while PLATIPUS achieves worse precision potentially because it generates a wide range of possible solutions that are not included in the ground truth evaluation set. Finally, MAML obtained a deteriorated recall because of its deterministic nature, while its high precision error is due to the fact that it optimizes only one globally shared set of parameters.

Figure 2 illustrates classifications made by three different sampled classifiers from a trained VMAML-*info* on an ambiguous few-shot dataset (see supplementary material for details on the adjusted VMAML architecture). The support set $\mathcal{D}_u^S$ given as input to the model is shown on the left, while the rest of the columns show query images that correspond to the three possible modes that are classified by the generated classifiers. We can see from the figure that VMAML-*info* can successfully produce solutions for all three ambiguous query sets. This is supported by our quantitative analysis reported in Table 2, where VMAML-*info* achieves coverage of $2.34$ covering more than two ambiguous task modes, while maintaining high accuracy of $84.2\%$ To cover all modes of the task, it is not sufficient to produce classifiers that perform well on all instances of the ambiguous tasks. Instead, each sampled classifier must give the highest log-likelihood on at least one of the given modes of the task provided the same support images $\mathcal{D}_u^S$. We hypothesise that this could be a reason why even stochastic approaches such as VERSA fail to cover more than one mode of the task. Among the considered benchmarks, PLATIPUS achieves the highest coverage of $1.75$ which is slightly lower than the value reported by Finn et al. (2018) potentially due to the fact that we used a different test split of attributes. Consistent with the results reported by Finn et al. (2018), MAML achieves good accuracy but can cover only one mode because of its deterministic nature. Similar to the regression problem, VMAML yields comparable performance to the evaluated state-of-the-art methods in terms of coverage and accuracy. Moreover, we observe that the rejection sampling mechanism yields superior coverage compared to the alternative way of receiving the weak labels as an extra input to the meta-learner (*info*$^*$), as shown by the performance of VMAML-*info* in Table 2.

# REFERENCES

Giulia Denevi, Massimiliano Pontil, and Carlo Ciliberto. The advantage of conditional meta-learning for biased regularization and fine tuning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

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

Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1704–1713, 2018.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2018a.

Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018b.

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

Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 3015–3022. ijcai.org, 2019.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Uncertainty in model-agnostic meta-learning using variational inference. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 3090–3100, 2020.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in neural information processing systems*, 2019.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pp. 4077–4087, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*, 2018.

Ruohan Wang, Yiannis Demiris, and Carlo Ciliberto. A structured prediction approach for conditional meta-learning. *arXiv preprint arXiv:2002.08799*, 2020.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7343–7353, 2018.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pp. 7115–7123. PMLR, 2019.

Baoquan Zhang, Ka-Cheong Leung, Yunming Ye, and Xutao Li. Metaconcept: Learn to abstract via concept graph for weakly-supervised few-shot learning. *arXiv preprint arXiv:2007.02379*, 2020.

# A Appendix

## A.1 Related work

The aim of meta-learning is to learn a meta-learner for facilitating the learning process of newly unseen tasks Vanschoren (2018). There are two major categories of meta-learning: metric-based and gradient-based meta-learning. For metric-based meta-learning, the goal is to learn a well-generalized embedding space Snell et al. (2017); Vinyals et al. (2016); Sung et al. (2018); Yoon et al. (2019). However, the applications of metric-based methods are limited to classification. In contrast, we focus on the gradient-based methods, which are independent of the problem types and regard meta-knowledge as model parameter initializations Finn et al. (2017; 2018); Grant et al. (2018); Li et al. (2017); Rajeswaran et al. (2019); Flennerhag et al. (2020); Rusu et al. (2019)). In the next parts, we will detail two subcategories of gradient-based meta-learning methods related to this paper.

**Meta-learning with Task Ambiguity.** A common approach to tackle task uncertainty in meta-supervised learning is by formulating the learning problem as a hierarchical Bayesian framework (Grant et al., 2018; Kim et al., 2018b; Finn et al., 2018; Gordon et al., 2019; Ravi & Beatson, 2019), which benefits from the inherent regularization of the Bayesian approaches. In such formulations, the parameters of each task $t$ are typically modelled as a random variable $\phi_t$ distinct from the random variable assigned to the meta-level parameters $\theta$. The task-specific random variables $\{\phi_t\}$ are mutually dependent where the dependence is realized through individual links to the meta-parameter variable $\theta$.

In these Bayesian meta-learning algorithms, variational inference has demonstrated promising results in many recent works Finn et al. (2018); Kim et al. (2018b); Ravi & Beatson (2019); Rusu et al. (2019); Nguyen et al. (2020). Finn et al. (2018) extended MAML to learn a distribution over the meta-parameters by optimizing a variational lower bound. In a similar work, Ravi & Beatson (2019) proposed to leverage MAML to update the amortized variational parameters using gradient descent at the initialization given by the meta model. Our approach resembles the work of Finn et al. (2018); Ravi & Beatson (2019) in that we also leverage MAML in an amortized VI framework to adapt the meta-parameters. However, the key difference is that we amortize a distribution over a low-dimensional task embedding instead of the high-dimensional parameter space. Our framework also incorporates a mechanism to reduce ambiguity via weak task labels.

**Meta-learning with Task Representations.** Besides Bayesian methods, constructing compact task representations has been shown to be a promising direction to address the task heterogeneity and uncertainty problems in few-shot learning (Yao et al., 2019; Hausman et al., 2018; Garnelo et al., 2018; Kim et al., 2018a; Rusu et al., 2019; Gordon et al., 2019; Vuorio et al., 2018). Unlike Rusu et al. (2019); Gordon et al. (2019), which formulate the task uncertainty by a diagonal Gaussian distribution in the high-dimensional parameter space, VMAML models the task uncertainty by an amortized distribution defined over the low-dimensional task embedding. Moreover, Rusu et al. (2019) integrates SGD optimization in the learning of the embedding space, while we perform SGD directly on the generated network parameters which scales well to high-dimensional parameter spaces.

The task uncertainty can be further mitigated by providing side information about the tasks in the form of weak labels, where labels are used to build a graph to express the relations across samples Liu et al. (2019); Zhang et al. (2020); Denevi et al. (2020). In contrast to these methods, VMAML exploits the labels to learn a shallow classifier defined in the low-dimensional task latent space, which is jointly trained with the meta model. Using the well-trained classifier, VMAML can generate solutions specific to a given label, and thereby improving the coverage of possible solutions.

## A.2 Details of the 1D regression task

Each task dataset was constructed by uniformly choosing one of the three base functions, and randomly sampling the corresponding parameters: phase in $[0, \pi]$ and amplitude in $[0.1, 5]$ for sinusoidal, slope and intercept in $[-3, 3]$ for linear, and intercept in $[-1, 1]$ and higher degree coefficients in $[-0.1, 0.1]$ for polynomials. Each support task dataset consisted of $n = 5$ pairs $(x_i, y_i)$ with $x_i$ randomly sampled from the interval $[-4.0, 4.0]$ and $y_i$ corrupted by Gaussian noise with standard deviation 0.3. The query dataset consisted of 50 such points. Weak labels $l$ were given as one-hot vectors: $l = [1, 0, 0]$ for sinusoidal, $l = [0, 1, 0]$ for polynomial, $l = [0, 0, 1]$ for linear functions, while $l = [0, 0, 0]$ was used for VMAML-*wo*. We predetermine a set of ground truth

Table 3: The split of Train/Test/Validation of attributes. The superscript denotes the index of the weak label: $^1$ for the attributes in the eyes and eye- brows areas, $^2$ for the forehead and hair, $^3$ for the nose and mouth areas, $^4$ for the ears, cheeks, jawline, chin and neck areas, and $^5$ for more general attributes.

| **Split** | **Attributes** |
|---|---|
| **Meta-training** | Arched Eyebrows$^1$, Attractive$^5$, Bags Under Eyes$^1$, Bald$^2$, Bangs$^2$, Big Lips$^3$, Wearing Earrings$^4$, Black Hair$^2$, Blond Hair$^2$, Blurry$^5$, Brown Hair$^2$, Sideburns$^4$, Bushy Eyebrows$^1$, Chubby$^5$, Goatee$^4$, High Cheekbones$^4$, Wearing Lipstick$^3$, No Beard$^4$, Oval Face$^5$, Pointy Nose$^3$, Receding Hairline$^2$, Male$^5$, Rosy Cheeks$^4$, Straight Hair$^2$, Wavy Hair$^2$. |
| **Meta-validation** | Wearing Necklace$^4$, Smiling$^3$, Pale Skin$^5$, Wearing Necktie$^4$, Big Nose$^3$. |
| **Meta-testing** | Eyeglasses$^1$, Gray Hair$^2$, Narrow Eyes$^1$, Wearing Hat$^2$, Mouth Slightly Open$^3$, Mustache$^3$, 5 o Clock Shadow$^4$, Double Chin$^4$, Young$^5$, Heavy Makeup$^5$. |

solutions for 200 tasks, each containing 100 possible function solutions. For each test task, the ground truth solutions were found by extensively searching for function parameters that yielded average error below a given threshold on the support data points. Note that by construction there might exists many solutions for a given few-shot task due to the noisy labels and multi-modality of the tasks arising from the three different base functions.

### A.3 Details of CelebA classification task

We evaluated VMAML on the ambiguous version of the CelebA few-shot attribute classification problem introduced by Finn et al. (2018). We consider $N = 2$ way and $K = 5$ shot classification and use the VMAML architecture presented in Section A.4. Each few-shot support dataset $\mathcal{D}^S$ contains a positive class of $K = 5$ images that have three attributes in common and a negative class containing the same number of images with neither of the attributes. The positive class of the query set $\mathcal{D}^Q$ contains images that satisfy only two of the three attributes, hence producing three possibilities (modes) to interpret the task determined by $\mathcal{D}^S$. In this experiment, weak labels are given by a multi-hot vector representing the categories of the positive attributes which are grouped into 5 categories. We denote by $l = [1, 0, 0, 0, 0]$ the attributes in the eyes and eyebrows areas, such as ”*Brushy Eyebrows*” and ”*Eye Glasses*”, $l = [0, 1, 0, 0, 0]$ for the forehead and hair, $l = [0, 0, 1, 0, 0]$ for the nose and mouth areas, $l = [0, 0, 0, 1, 0]$ for the ears, cheeks, jawline, chin and neck areas, and $l = [0, 0, 0, 0, 1]$ for general attributes of the face such as ”*Young*”. A complete description of the attributes as well as the construction of the weak labels and the train, test and validation splits is given in the supplementary material.

Similar to Finn et al. (2018), we construct few-shot CelebA attribute classification task using the split of meta-train/val/test set containing 162770/19867/19962 images with the attribute split presented in Table 3. Furthermore, we divided the attributes into 5 categories and assigned a one-hot vector as the weak label $l$ to each category. The index of the categories is provided by superscripts.

We chose the splits such that there are exactly two attributes in each of the five categories of weak labels to minimize task ambiguity at test time. Note that the tasks are still ambiguous in cases where two of the positive attributes have the same weak label.

We used the same coverage and accuracy evaluation protocol introduced by Finn et al. (2018). For every test few-shot dataset $\mathcal{D}_u$, we produce several classifier models using the support set $\mathcal{D}_u^S$, and assign each of them to one of the three possible classification tasks that yields the highest log-likelihood. The coverage is determined by measuring the average number of tasks that receive at least one generated classification model per dataset $\mathcal{D}_u^S$. Using the assigned classifiers, accuracy is defined by averaging their accuracy on these tasks.

### A.4 Extending VMAML to N-way K-shot tasks

In classification problems, we can avoid generating very high-dimensional parameters by generating only the top layer weights and biases of the final classifier Rusu et al. (2019); Gordon et al. (2019). In this case, a feature extractor model can be shared across different tasks to internally process the inputs $x$ from the support set $\mathcal{D}^S$.

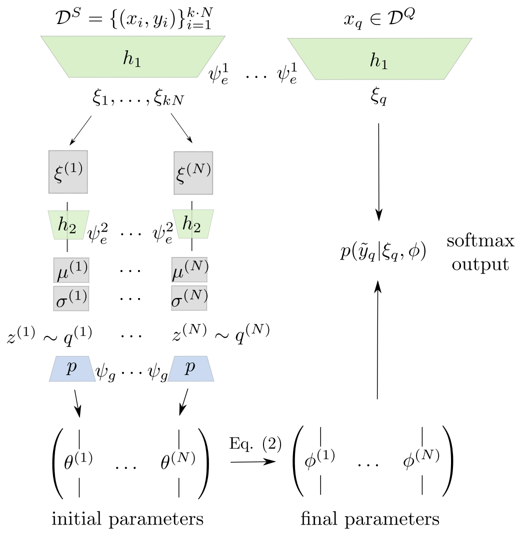
$\mathcal{D}^S = \{(x_i, y_i)\}_{i=1}^{k \cdot N}$
$x_q \in \mathcal{D}^Q$
$h_1$
$\psi_e^1 \dots \psi_e^1$
$h_1$
$\xi_1, \dots, \xi_{kN}$
$\xi_q$
$\xi^{(1)}$
$\xi^{(N)}$
$h_2$ $\psi_e^2 \dots \psi_e^2$ $h_2$
$\mu^{(1)}$ $\dots$ $\mu^{(N)}$
$\sigma^{(1)}$ $\dots$ $\sigma^{(N)}$
$z^{(1)} \sim q^{(1)}$ $\dots$ $z^{(N)} \sim q^{(N)}$
$p$ $\psi_g \cdots \psi_g$ $p$
$p(\tilde{y}_q | \xi_q, \phi)$
softmax output
$\theta^{(1)} \dots \theta^{(N)}$
Eq. (2)
$\phi^{(1)} \dots \phi^{(N)}$
initial parameters
final parameters

Figure 4: The computational flow of VMAML for an $N$-way $K$-shot classification problem, where $N$ denotes the number of classes and $K$ number of samples per class. Support data points that belong to the same class $c$ are processed separately by $h_2$ which outputs the parameters $\mu^{(c)}, \sigma^{(c)}$ of the approximated posterior $q^{(c)}$. A latent sample $z^{(c)}$ from $q^{(c)}$ is mapped to $\theta^{(c)}$ representing one column of the weights and biases of the initial linear classifier using the generative model $p$. The output parameters are further optimized based on MAML gradient descent update to yield the task specific parameters $\phi^{(c)}$.

Therefore, the architecture of the task encoder $h(\mathcal{D}^S; \psi_e)$ is divided into two smaller neural networks $h_1$ and $h_2$ parameterized by $\psi_e^1$ and $\psi_e^2$, respectively, such that $\psi_e = \psi_e^1 \sqcup \psi_e^2$. The computational flow of the adjusted task encoder is visualized in Figure 4. In the inner loop, the entire support dataset $\mathcal{D}^S$ is processed by $h_1(\mathcal{D}^S; \psi_e^1)$ which outputs features $\xi_i$ for each input $x_i$ (where weak labels $l$ are omitted for simplicity). The obtained features belonging to the same class $c \in \{1, \dots, N\}$ are then concatenated into one feature vector $\xi^{(c)}$ and processed by $h_2(\xi^{(c)}; \psi_e^2)$, which outputs parameters $\mu^{(c)}, \sigma^{(c)}$ of the approximate posterior $q^{(c)}$. A sample $z^{(c)} \sim q^{(c)}$ accepted by the rejection sampling mechanism is then given to the generative model $p$ which outputs the initial parameters $\theta^{(c)}$ representing top layer weights and biases of the classifier. As before, these are further optimized by one gradient descent step to yield the final parameters $\phi^{(c)}$. In the outer loop, a query point $x_q \in \mathcal{D}^Q$ is first processed by $h_1$ to extract the features $\xi_q$. These are then fed to the linear classifier with weights and biases given by $\phi^{(1)}, \dots, \phi^{(N)}$ that outputs softmax probability $p(\tilde{y}_q | \xi_q, \phi)$ for each of the $N$ classes. In this case, the KL divergence term in the outer loop optimization in equation 1 is calculated as the sum over individual KL terms for each class $c$ using $q^{(c)}$.