Model-Agnostic Learning to Meta-Learn

Arnout Devos*, Yatin Dandi*
School of Computer and Communication Sciences
Swiss Federal Institute of Technology Lausanne (EPFL)

Abstract

In this paper, we propose a learning algorithm that enables a model to quickly exploit commonalities among related tasks from an unseen task distribution, before quickly adapting to specific tasks from that same distribution. We investigate how learning with different task distributions can first improve adaptability by meta-finetuning on related tasks before improving goal task generalization with finetuning. Preliminary experiments on toy data validate the intuition that learning to meta-learn can help to improve adaptability and consecutively generalization. The methodology, setup, and hypotheses in this proposal were positively evaluated by peer review before conclusive experiments were carried out.

1 Introduction

Recent years have seen encouraging developments in meta-learning based approaches for deep neural networks and their successful application to various domains [Finn et al., 2017a, Rajeswaran et al., 2019, Nichol et al., 2018]. These approaches typically assume a distribution over tasks and aim to exploit the shared properties across tasks to learn a model that can adapt to unknown tasks from this distribution using only a few training data points. Even though these approaches "learn to learn" on the given task distribution, their adaptive capabilities do not generalize well to unseen tasks from related but different task distributions.

A number of recent works have proposed addressing the presence of different sets of related tasks by explicitly factoring in the heterogeneous nature of the task distribution in the design of the architecture and update rule [Requeima et al., 2019, Yao et al., 2020, 2019, Vuorio et al., 2019a]. However, these approaches still assume a fixed task distribution, such as tasks sampled from a fixed set of families of functions or a multi-modal distribution arising out of a fixed set of task datasets. We argue that the ability to generalize across unknown datasets and task distributions is a fundamentally more difficult problem than fixed distribution meta-learning. With a new task distribution or dataset, it is unrealistic to expect the model to quickly adapt to any arbitrary task from such a distribution. Instead, we expect the model to quickly learn to adapt to any task from the new task distribution after being exposed to only a few tasks of it. This generalizes the notion of few-shot learning to *few-task* (few-shot) learning. Changes in task distributions might also arise due to natural or artificial transformations of the data. With different task distributions arising from a "distribution over task distributions", it is not only desirable to "quickly adapt" to unseen tasks but also "quickly learn to adapt" to unseen task distributions.

We propose a general framework for adapting to unseen task distributions by "learning to meta-learn" on different task distributions during training. Thus, the heterogeneity of tasks in our approach is not fixed but flexibly modeled through hierarchical sampling from a distribution over task distributions. Similar to MAML [Finn et al., 2017a], we propose a general framework to learn a suitable initialization for a single set of parameters. Unlike MAML, which only trains a model to quickly adapt the parameters on a new task using few task-specific gradient steps, our model is also trained to quickly adapt its initialization to a new task distribution using few meta gradient steps on this unseen task distribution. We hypothesize that our approach would allow models to transfer learning capabilities

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

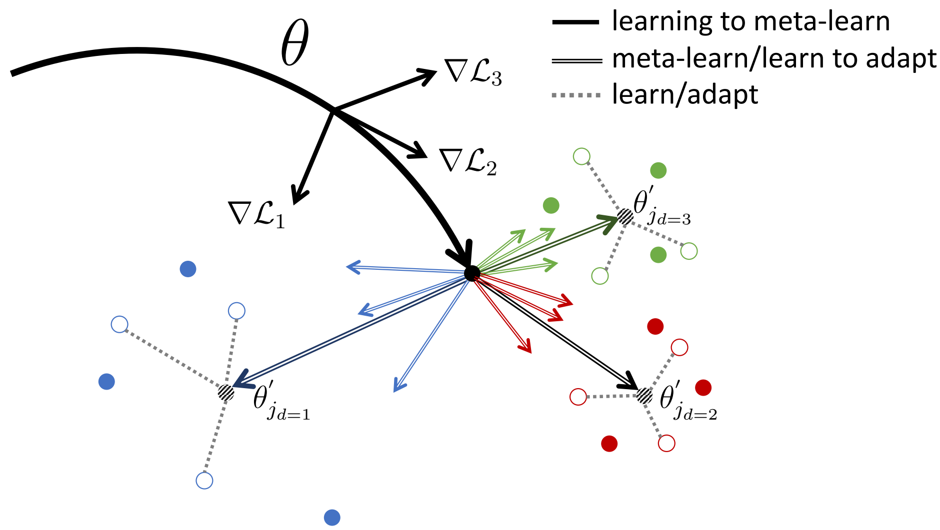


Figure 1: Diagram of our model-agnostic learning to meta-learn algorithm (MALTML), which optimizes for a representation θ that can quickly adapt to new task distributions and consecutively to their tasks. Illustrated here with single gradient steps for the meta-learning and learn/adapt phases.

across datasets in supervised and unsupervised learning, and new environments in reinforcement learning as well as quickly adapt to unseen augmentations and distortions at test time.

2 Related Work

Model-Agnostic Meta-Learning (MAML) by Finn et al. [2017b] is a seminal work in few-shot meta-learning which seeks a common model initialization that allows the model to perform well on any goal task from the training task distribution with few gradient steps (and samples). Multimodal MAML (MMAML) by Vuorio et al. [2018, 2019b] extends MAML with the capability to identify tasks sampled from a multimodal task distribution and adapt quickly through gradient updates. Yao et al. [2019] proposed the hierarchically structured meta-learning (HSML) algorithm that explicitly tailors the transferable knowledge to different clusters of tasks. Automated Relational Meta-Learning (ARML) by Yao et al. [2020] extracts the cross-task relations and constructs a meta-knowledge graph. When a new task arrives, it can quickly find the most relevant structure and tailor the learned structure knowledge to the meta-learner. Still, MMAML, HSML, and ARML only learn to learn from a *fixed* task distribution. Unlike our approach, they are not expected to generalize to new task distributions.

Research on improving meta-learning algorithms is vast, and we will highlight work most related to our approach. Following MAML, Li et al. [2017] and Antoniou et al. [2019] learn the inner learning rate in the outer loop to improve performance, while reducing the hyperparameter tuning requirement. Finn et al. [2017b] proposed a first-order approximation of MAML (fo-MAML) to scale to larger models, which was subsequently improved upon by Nichol et al. [2018] with a first-order method called Reptile. Reptile can naturally be extended to our proposed approach, making it scalable and efficient. Chen et al. [2020] found that with increasingly deep architectures, common pre-training and transfer learning can outperform meta-learning from scratch in the visual classification domain. Based on this, they proposed to combine regular pre-training with subsequent meta-learning, which empirically gives a further performance improvement. Raghu et al. [2020] found that feature reuse is a dominant factor in MAML, and proposed a variant called Almost no Inner Loop (ANIL) which learns to only fine-tune the last layer linear classifier. On the contrary, Oh et al. [2020] came to the opposite conclusion that fast learning is crucial, and proposed a Body Only update in Inner Loop (BOIL) algorithm. These works motivate several of our research questions for the experiments.

3 Methodology

We aim to train models that can quickly change their adaptability before rapid adaptation. We formalize this setting as few-task few-shot learning. In this section, we will clarify the problem setup and we will formalize this learning to meta-learn problem setting for supervised learning, but it can easily be generalized to unsupervised and reinforcement learning (RL).

3.1 Learning to Meta-Learn Problem Set-Up

In our learning to meta-learn scenario, we consider a distribution $p(j)$ over task distributions such as families of related functions or datasets with similarities. Our goal is to allow the model to adapt to unseen task distributions as well as specific tasks within such distributions. In the L-task K-shot setting, the model is trained to meta-learn a new task distribution j_d from only L tasks with only K examples each for task-learning and Q examples for meta-learning, before learning a single goal task $\mathcal{T}_i$ drawn from $p_{j_d}(\mathcal{T})$ from only K samples. The model f is then improved by considering how the test error on new (validation) data from $\mathcal{T}_i$ changes with respect to the original parameters. This test error on the final goal tasks serves as the training error of the learning to meta-learn process. At the end of training, new families are sampled from $p(j_d)$ and the model's learning to meta-learn performance is measured by the model's performance after meta-finetuning on L tasks with $K+Q$ examples and finetuning on one or multiple goal tasks from the same family with K examples.

3.2 A Model-Agnostic Learning to Meta-Learn Algorithm

We propose a method that can learn the parameters of any model via learning to meta-learn in such a way as to prepare that model to first quickly change its adaptation capability (initialization) and consecutively adapt quickly to a goal task. The intuition behind this approach is that some internal representations are more transferable to meta-learn with. For example, a neural network could learn features that are broadly applicable to all tasks in the distribution over task distributions $p(j)$ and can then specialize to an individual task distribution $p_{j_d}(\mathcal{T})$, rather than to a single task distribution or task. We assume no specific form of the model, other than that it is parametrized by some parameter set θ, and that the loss functions are sufficiently smooth in θ such that we can employ gradient-based learning techniques.

Formally, we consider a model represented by a parametrized function f_θ with parameters θ. When adjusting the adaptability to a new task *family* j_d, the model's parameters θ become θ'_{j_d}, and consecutively when adapting (finetuning) to a new task $\mathcal{T}_c \sim p_{j_d}(\mathcal{T})$ the model's parameters θ'_{j_d} become θ'_c. In our approach, the updated parameter vector θ'_{j_d} is obtained by few (inner) *meta*-learning steps on few tasks from dataset j_d, also called *meta-finetuning*. Each meta-finetuning step is taken across few task-finetuning steps. A single ($r=1$) *task* gradient step with step size α is:

$$\theta'_{\mathcal{T}_i} = U^{r=1,\alpha}_{\mathcal{T}_i}(\theta) = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta). \tag{1}$$

Then, the family-specific meta-finetuning update, with one ($r=1$) task-level parameter update as in Equation (1) and one ($m=1$) meta-level update across tasks from j_d with step size β, is:

$$\theta'_{j_d} = V^{m=1,\beta}_{j_d}(\theta) = \theta - \beta \nabla_\theta \left(\sum_{\mathcal{T}_i \sim p_{j_d}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{U^{r=1,\alpha}_{\mathcal{T}_i}(\theta)}) \right). \tag{2}$$

Consecutively, the updated task-specific parameter vector θ'_c is obtained by taking few gradient descent steps, with learning rate γ, on tasks $\mathcal{T}_c \sim p_{j_d}(\mathcal{T})$, starting from θ'_{j_d}. For example, using Equation (1), with 1 gradient step: $\theta'_c = U^{r=1,\gamma}_{\mathcal{T}_c}(\theta'_{j_d})$. Note that using the same r for meta-finetuning and goal task finetuning is a natural choice, but can be deviated from. The step-sizes α, β, and γ may be fixed as hyperparameters or learned on the outer learning loop (see below).

Finally, the global model parameters θ are optimized to perform well after this two-step (meta-learn, then learn) process. Specifically, the model parameters are trained by optimizing the performance of every $f_{\theta'_c}$ on its task $\mathcal{T}_c$. This is done across task distributions sampled from $p(j)$ and tasks sampled from them ($\mathcal{T}_c \sim p_{j_d}(\mathcal{T})$). Concretely, the *learning-to-meta-learn* objective is:

$$\min_\theta \sum_{j_d \sim p(j)} \sum_{\mathcal{T}_c \sim p_{j_d}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_c}(f_{\theta'_c}) = \min_\theta \sum_{j_d \sim p(j)} \sum_{\mathcal{T}_c \sim p_{j_d}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_c}\left(U^{r,\gamma}_{\mathcal{T}_c}(V^{m,\beta}_{j_d}(\theta)) \right)$$

Note that the learning to meta-learn optimization is performed over the model parameters θ, whereas the learning to meta-learn objective itself is computed using the updated model parameters θ'_c.

Algorithm 1 Model-Agnostic Learning to Meta-Learn

Require: $p(j)$: distribution over task distributions parameter j
Require: $\alpha, \beta, \gamma, \eta$: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of task distributions $j_d \sim p(j)$
  for all j_d **do**
    Sample L tasks $\mathcal{T}_i \sim p_{j_d}(\mathcal{T})$
    for all $\mathcal{T}_i$ **do**
      Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
      Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    end for
    Update $\theta'_{j_d} \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$ using Q examples per task
    Sample batch of tasks $\mathcal{T}_c \sim p_{j_d}(\mathcal{T})$
    for all $\mathcal{T}_c$ **do**
      Evaluate $\nabla_{\theta'_{j_d}} \mathcal{L}_{\mathcal{T}_c}(f_{\theta'_{j_d}})$ with respect to K examples
      Compute adapted parameters with gradient descent: $\theta'_c = \theta'_{j_d} - \gamma \nabla_{\theta'_{j_d}} \mathcal{L}_{\mathcal{T}_c}(f_{\theta'_{j_d}})$
    end for
  end for
  Update $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{j_d \sim p(j)} \sum_{\mathcal{T}_c \sim p_{j_d}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_c}(f_{\theta'_c})$
end while

The outer optimization across task distributions is performed via stochastic gradient descent (SGD), such that the model parameters θ are updated as follows:

$$\theta \leftarrow \theta - \eta \nabla_\theta \sum_{j_d \sim p(j)} \sum_{\mathcal{T}_c \sim p_{j_d}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_c}(f_{\theta'_c}) \tag{3}$$

where η is the outer step size. The full algorithm, in the general case, is outlined in Algorithm 1.

The MALTML outer gradient update yields a third-order gradient with respect to θ. To make MALTML computationally usable for high-dimensional models, we propose a first-order approximation. Concretely, following Nichol et al. [2018], for every family j_d we employ multiple *first-order* meta-learning updates $\theta'_{j_d} = \widetilde{V}^{m>1}_{j_d}(\theta)$, before updating the model parameters with $\theta \leftarrow \theta + \eta \sum_{j_d}(\theta'_{j_d} - \theta)$. Note that in this case, due to the nature of our two-level first-order approximation, goal task finetuning is not required anymore.

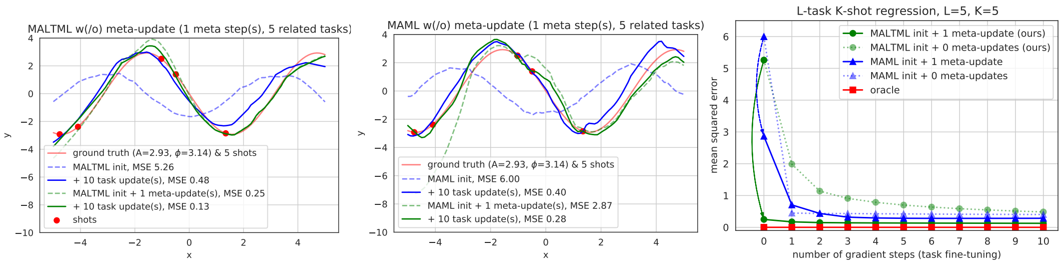
4 Experimental Protocol

The goal of our experimentals is to get conclusive results in different learning domains on whether MALTML can enable a quick and significant change of adaptability to goal tasks from new task distributions. Moreover, we wish to examine whether fast learning (to meta-learn) [Oh et al., 2020] or feature reuse [Raghu et al., 2020] is the dominant factor in the performance.

All the learning to meta-learn problems we consider require some form of change in adaptability and subsequent adaptation to new tasks at test time. When possible, we will compare our results to an oracle that receives the identities of the family and goal task as an additional input or a MAML oracle that is able to meta-finetune on a large number of tasks from the new task distribution, as upper bounds on the performance of the models. Regarding model architecture and optimization, we will follow Finn et al. [2017b]. We will use insights from Antoniou et al. [2019] to stabilize training where applicable and follow its hierarchical hyperparameter search methodology. We will carry out the experiments in PyTorch [Paszke et al., 2019], using the torchmeta package [Deleu et al., 2019]. The code will be available online.

4.1 Illustrative preliminary experiment: regression

We start with a toy regression problem which illustrates the experimental protocol of few-task few-shot learning and the basic principles of MALTML.



(a) (b) (c)

Figure 2: Effect of the meta-update and task fine-tuning on (a) MALTML (b) MAML (c) Test error vs task fine-tuning steps (the meta-update improvement is indicated by the arrows at 0 steps).

Each task distribution (family) consists of sinusoid regression tasks with a specific phase. Thus, $p(j)$ is continuous, where the phase j varies uniformly within $[0, \pi]$. Each task involves regressing from the input to the output of a sine wave, where the amplitude is varied between tasks. Thus, $p_{j_d}(\mathcal{T})$ is continuous, where the amplitude varies within $[0.1, 5.0]$ and the input and output both have a dimensionality of 1. During training and testing, datapoints x are sampled uniformly from $[-5, 5]$. The loss is the mean squared error between the prediction $f(x)$ and the true value. The model is a neural network with 2 hidden layers of size 40 with ReLU nonlinearities. When training with MALTML, we use single gradient updates with fixed step sizes of $\alpha = 0.001$, $\beta = 0.01$, $\gamma = 0.001$, and use Adam [Kingma and Ba, 2015] with an initial learning rate of $\eta = 0.001$. The baselines are also trained with Adam, and an inner learning rate of $\alpha = 0.001$ for MAML. For the 5-task 5-shot regression experiment, we train for 70,000 outer steps with a family batch size of 10, $Q = 5$, and 2 validation tasks per family.

For this preliminary experiment we only contrast with a MAML baseline, which disregards the family structure, and an oracle receiving the true amplitude and phase of the goal task as additional input. In general, we intend to compare to the other oracles described before and another baseline: pretraining on all tasks, which in this case involves training a model to regress random sinusoid functions.

The toy results in Figures 2a and 2c show that the learned MALTML model is able to quickly change its adaptability to the new family's phase on 5 related 5-shot tasks. Due to this, it reaches a better fit than MAML, which benefits less from the meta-adaptation of its initialization (Figures 2b and 2c).

4.2 Specifics of main experiments

Besides the toy experiment in section 4.1, we propose to test the effectiveness of MALTML for:

Classification. We propose to apply our method to modified versions of the Omniglot [Lake et al., 2015] and ImageNet [Russakovsky et al., 2015] datasets. For ImageNet, we will propose a few-task few-shot dataset, making use of its hierarchical structure to generate a sufficient amount of training families. Given the finite number of alphabets in Omniglot, which will serve as families, we will use data augmentations similar to the ones used in Khodadadeh et al. [2019] to generate a large number of training families. To arrive at a realistic setting where the (imposed) hierarchical structure from the supervised case is lacking, we will use a hierarchically augmented version of unsupervised meta-learning [Khodadadeh et al., 2019]. Specifically, a subset of data augmentation parameters will be sampled per family, before applying them randomly (with the remaining subset) on samples to generate tasks from each family. Note that in this case, on a (family) meta-level the method needs to be sensitive to augmentations, whereas on a task-level it should aim to be invariant to them.

2D Navigation. We propose to evaluate MALTML on a set of families of RL tasks where a point agent must move to different goal positions in 2D, while being given related tasks from the same family. Every family constitutes of a random crop of the unit square, and every task is randomly chosen from within that rectangle. The crops are bounded by 25% to 75% of the original unit length.

Continual Regression We propose to evaluate a continual learning extension of MALTML on incremental sine wave learning as described in Javed and White [2019]. Different families of continual learning prediction problems will correspond to different frequencies of the sine waves. The inner meta-learning objective for this setting will be replaced by the Online-aware Meta-Learning objective from Javed and White [2019].

Continual Reinforcement Learning Besides the sine waves continual regression problem, we will aim to evaluate a more challenging and real-world setting of continuous control inspired by Kaplanis et al. [2020].

5 Future Work

Based on the results of our main experiments, future work could involve a multi-task setting corresponding to meta-learning on different categories of tasks such as classification, segmentation, and depth estimation on a single dataset or a set of related datasets. This could also involve augmenting our objective for enforcing cross-task (distribution) consistency [Zamir et al., 2020].

Acknowledgements

We would like to express gratitude to the anonymous reviewers, as well as Matthias Grossglauser for insightful comments. Arnout Devos acknowledges funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 754354.

References

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJGven05Y7`.

Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A Meta-Learning library for PyTorch, 2019. URL `https://arxiv.org/abs/1909.06576`. Available at: https://github.com/tristandeleu/pytorch-meta.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017a.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017b.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019.

Christos Kaplanis, Claudia Clopath, and Murray Shanahan. Continual reinforcement learning with multi-timescale replay, 2020.

Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. In *Advances in Neural Information Processing Systems*, pages 10132–10142, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

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.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. Does maml really want feature reuse only?, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkgMkCEtPB`.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, 2019.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, pages 7957–7968, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

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

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2019a.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2019b.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2019.

Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task consistency, 2020.