Building a Robust Question Answering System with Meta-Learning

Stanford CS224N Default Project
Track: RobustQA
Mentor: Kendrick Shen

Dylan Cunningham
Department of Computer Science
Stanford University
dcunnin2@stanford.edu

Doug Klink
Department of Computer Science
Stanford University
dklink@stanford.edu

Abstract

Question Answering (QA) is a benchmark task in NLP because it effectively evaluates a model's ability to understand a passage of text. There is a great diversity of problem domains within QA, evidenced by the many QA datasets which exist today. Due to these numerous domains, it is useful for a model trained on data-rich domains to be able to generalize to new, data-poor domains. Here, we pursue meta-learning as an approach to this generalization task. We consider three data-rich, "in-domain" QA datasets, and three data-poor, "out-of-domain" QA datasets, with the goal of performing well on the out-of-domain datasets using only a small number of training examples. In our approach, we consider our three data-rich QA datasets as individual meta-learning tasks, and fine-tune a pretrained DistilBERT using the MAML algorithm. After further fine-tuning on the out-of-domain training data, this approach performs about as well as our baseline model trained with a standard training regime. However, it does not surpass the performance of simply fine-tuning our baseline model on the three data-poor datasets. In response, we investigate the benefits and drawbacks of the meta-learning approach, and discuss its potential usefulness for this particular problem.

1 Introduction

Question Answering is one of the central tasks in NLP research today. It is particularly important because numerous NLP problems can be reframed as a question-answering problem, and a good system requires many of the traits we associate with "true intelligence" (ability to take in information, reason about it, and produce meaningful insights). QA also receives a lot of research attention because, unsurprisingly, it is extremely challenging. Even the simplified span-retrieval formulation of the problem can be challenging for humans, and though it's straightforward to formulate trivially easy questions (e.g. "What year was George Washington born?", given the first sentence of his Wikipedia article), it's easy to imagine questions which are almost arbitrarily difficult. Yet, large pretrained transformers such as BERT and its variations have had great success, surpassing human performance on benchmark datasets such as SQuAD [1].

Our project combines QA with a second challenge: domain adaptation. There exist a great number of QA datasets, and each is formulated and constructed differently. Yet, we would like a model which is trained on one to perform well on the other. This ability to generalize (a model's "robustness") is a major research focus across many fields of deep learning. There are many variants of the domain-adaptation problem; in ours, we have access to three large "in-domain" QA datasets (50k examples each), and also have access to a small amount of data for three "out-of-domain" datasets (127 examples each). Thus, the challenge is to use the data-rich datasets to build a good question answering system, but also find a way to leverage the scarce out-of-domain training data to ultimately achieve good performance on the held out test set of out-of-domain contexts and questions.

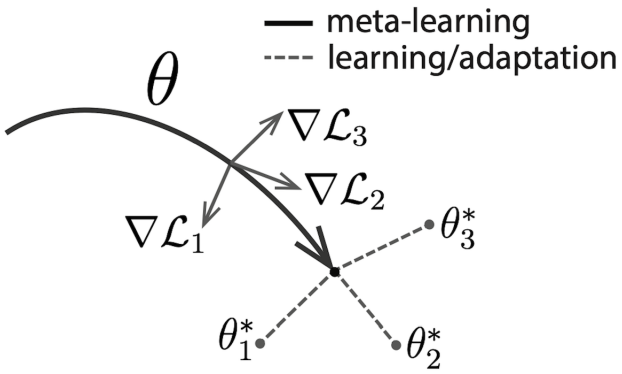

Figure 1: The MAML training framework learning overall θ based on multiple tasks that each have their own ideal θ^* [2]

To accomplish this challenge, we chose to pursue a meta-learning strategy. In meta-learning, the objective is no longer simply to train a model which performs well on some dataset, but to train a model which can quickly adapt to a new dataset, given a small number of examples. This is accomplished by repeatedly training the model on a small sample of data from one of many disjoint "tasks," then evaluating its performance on this task after this short burst of training. The resulting loss from each task is then used to update the original model parameters. The objective of meta-learning aligns nicely with our problem, with each of our datasets corresponding to a "task." Figure 1 demonstrates how meta-learning can create model parameters that are able to quickly learn and adapt to new tasks.

Though our intuition was that meta-learning was well-suited to this problem, we found that our models trained using meta-learning did not outperform models trained using the standard learning paradigm. However, we also found that our meta-trained models were able to leverage the out-of-domain data more effectively, which provides some hope that meta-learning could be useful with further optimizations.

2 Related Work

There are several approaches to meta-learning, and we chose to follow the model-agnostic meta-learning (MAML) algorithm [2]. This approach, invented by Stanford professor Chelsea Finn, has been highly influential, with the paper receiving over five thousand citations at the time of this writing. One of the key advantages to this approach is that it adds no new parameters, and is model-agnostic, which is very helpful when working with large pretrained transformers. Numerous modifications to the MAML algorithm have been proposed which address some of its shortcomings, but we chose to stick with the standard approach for simplicity of implementation.

Although previous work has applied meta-learning to visual question answering about images[3] and querying from knowledge bases [4][5], we were not able to find many papers which focus on applying meta-learning to question answering as it is framed here. As such, we had little intuition into whether our approach would be successful or not. We suspect this lack of publications is due to our problem being artificially constructed. Though we are given only 100 training examples from each out-of-domain dataset, in reality these datasets each have over a hundred thousand examples available. As such, researchers are probably more interested in maximizing performance using all available resources, rather than artificially simulating data-scarcity. However, this lack of research also shows that our project could lead to the discovery of a new application where meta-learning shows promise.

3 Approach

The main challenge of our project was correctly implementing the MAML training regime. Though the algorithm is not unduly complex, it was tricky to conceptually understand the motivation for its

Algorithm 1 MAML Training Loop

Require: $p(\mathcal{T})$: distribution over tasks
1: Use pre-trained DistilBERT as initial θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K examples from $\mathcal{T}_i$
6: **for** n adaptation steps **do**
7: Evaluate $\mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
8: Compute adapted parameters: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
9: **end for**
10: Compute task loss: $\mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
11: **end for**
12: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
13: **end while**

various components, such as where gradients were flowing during the meta-update (Algorithm 1, line 12), and to keep track of training versus meta-training (versus training-during-evaluation, etc.). We also enjoyed debugging memory overflows, and learning how pytorch accumulated memory across the adaptation steps when building its computational graph (Algorithm 1, line 6). Our implementation involved creating a MetaTrainer class based on the default Trainer, and re-working the "train" and "evaluate" methods from scratch. We utilized a library called LEARN2LEARN which includes a lightweight wrapper for a MAML 'learner' which cleanly handles the fast-adaptation SGD, and allowed us to avoid calculating second derivatives in the meta-update step (per advice from Chelsea Finn in CS 330 materials). We then also built some machinery to model our datasets as tasks and to randomly sample examples from them. Throughout implementing this new training regime, we wound up reworking and refactoring much of the starter code. In the end, we were pleased that the time we invested in thoroughly understanding the MAML algorithm paid off in the form of a working implementation.

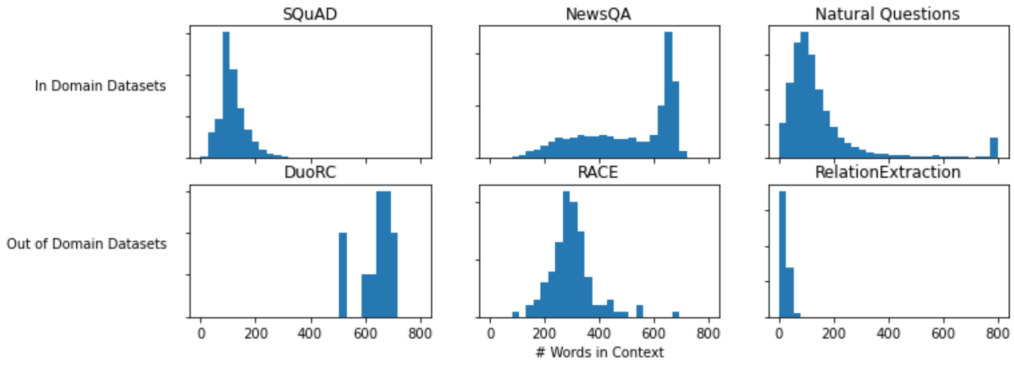

Figure 2: Histograms of Context Length in Training Datasets

When researching MAML, we realized that example use-cases of meta-learning typically used many more than three tasks. This led us to think about ways in which we could subdivide our three in-domain datasets into salient subtasks. Our intuition was that to properly prepare our model to react to new sorts of question-answer pairs, we should provide it with as many tasks as possible, such that those tasks differed in notable ways from each other. This led us to investigate the distribution of our training data, where we found notable variations, particularly in context length (Fig. 2). For example, NewsQA clearly has a bi-modal distribution with respect to context length, and could be split into two smaller tasks accordingly. Further, we noticed that the context-length distribution of each out-of-domain dataset differed strongly from the others and from the in-domain datasets. This led us to believe that simulating short, medium, and long-context tasks from each in-domain dataset could help prepare the model to adapt to the novel distributions found in the out-of-domain datasets.

To artifically create extra tasks out of our training data, we wrote a script that took a training dataset as input and split that dataset into two or three smaller subdatasets based on context length. We then were able to carry out two experiments with these new sets of tasks.

4 Experiments

4.1 Data

Our training data consists of labeled context, question, and answer examples. There are three large in-domain datasets (SQuAD [6], NewsQA [7], and Natural Questions [8]) with 50,000 examples each (150,000 total), and three small out-of-domain datasets (DuoRC [9], RACE [10], and RelationExtraction [11]) with 127 examples each (381 total). Each training example consists of a context, question, and answer triple. Some questions share the same context paragraph. We evaluate our models on a combined validation set containing 382 total examples from DuoRC (126), RACE (128), and RelationExtraction (128). We test our best model on a combined test set containing 4,360 total examples from DuoRC (1,248), RACE (419), and RelationExtraction (2,693).

4.2 Evaluation method

We use the provided metrics of exact match (EM) and F1 scores to evaluate our model performance on span detection for question answering. EM and F1 scores provide a reliable and simple way to compare the performance of different models. Each of our models were evaluated on the combined out-of-domain validation dataset and the two best performing models were then evaluated on the combined out-of-domain test dataset and submitted to the class leaderboard.

4.3 Experimental details

We used a consistent learning rate of 3^{-5} for all of our experiments. We used a batch size of 16 whenever we used the regular training regime either for our baseline models or for finetuning with out-of-domain data. Whenever we ran the regular training regime to train our baseline models or finetune with out-of-domain data, we used a batch size of 16 and carried out 10 epochs of training (unless otherwise specified below).

Baseline: Our baseline model is a pretrained DistilBERT from the Hugging Face library [12] trained on the three in-domain training datasets for three epochs with the standard training regime.

FS-base: Initialized DistilBERT with Baseline parameters and then finetuned those parameters on our three out-of-domain training datasets with the standard training regime.

Meta-base: Used pretrained DistilBERT as the initial parameters and then used the meta-training regime to train on a distribution of three tasks: one task for each of in-domain training datasets. Ran 3000 iterations of the outer while loop in Algorithm 1. Each iteration, we sampled 3 tasks and 16 examples from each task with 2 adaptation steps and a learning rate of $\alpha = 3^{-5}$. We chose to use 16 sampled examples for each meta-learning step, because that tends to be near the optimal value found by Cioba et al. when they tested how to distribute data across tasks for meta-learning [13].

Meta-FS-base: Initialized DistilBERT with Meta-base parameters and then finetuned those parameters on our three out-of-domain training datasets using the standard training regime.

Meta-FS-AVG: For each out-of-domain datasets, we initialized DistilBERT with Meta-base parameters and then finetuned those parameters with just that train dataset using the standard training regime and then evaluated on the corresponding val dataset. We then averaged these scores by hand and reported the average in our results table below.

Meta-NewsQA-Split: We split NewsQA train dataset into two tasks based on context length and then trained a model with the meta-learning regime on the two NewsQA tasks and the other two train datasets. The 4 tasks were sampled from at random, so the resulting model was trained on NewsQA more often than SQuAD or Natural Questions. NewsQA seemed to be the dataset most similar to the RACE out-of-domain dataset, and FS-base was struggling the most with RACE, so we hoped that overrepresenting the NewsQA dataset during training would help solve this problem. We used a pretrained DistilBERT and the same hyperparameters as we did for Meta-base in this step. We then

finetuned those resulting parameters on the out-of-domain training data using the standard training regime.

Meta-Even-Split: We split all the train datasets into evenly sized small, medium, and large subdatasets based on context length. We wanted to see if having a larger amount of tasks and a smaller amount of data per task would lead to better performance of our model. Each of those subdatasets served as a task and we trained a model with the meta-learning regime on those nine tasks using a pretrained DistilBERT and the same hyperparameters as we did for Meta-base. We then finetuned those resulting parameters on the out-of-domain training data using the standard training regime.

4.4 Results

Table 1: Model Performance on Out-of-Domain Test Set

Model	EM	F1
FS-base	**40.46**	**58.33**
Meta-FS-base	37.50	54.81

Table 2: Model Performance on Out-of-Domain Validation Set

Model	EM	F1
Baseline	30.63	47.72
FS-base	**33.51**	**49.83**
Meta-base	27.75	43.65
Meta-FS-base	**33.51**	47.37
Meta-FS-AVG	32.71	47.86
Meta-NewsQA-Split	31.15	46.33
Meta-Even-Split	30.89	46.48

As meta-learning was a new topic for both of us, we initially did not have any specific expectations about how well a meta-learning training regime would perform in this context. It intuitively seemed like meta-learning was built for problems similar to the one defined here, but we were unclear about how much better meta-learning would perform compared to the baseline. It's interesting to see that all of our meta-learning models performed about the same as the baseline model and none of them were able to beat our FS-base model.

We were glad to see that fine-tuning the baseline model with a small amount of out-of-domain examples led to increased performance on the out-of-domain validation set. We initially expected that the model might overfit to the small amount of training examples, but it looks like this is not the case.

Changing Hyperparameters: We think that a few factors led to the mediocre performance of our meta-training based models. First, we did not spend enough time tuning our hyperparameters. The learning rate for the inner meta-training loop does critical work in our training regime and we did not experiment beyond 3^{-5} learning rate that was used in the baseline model. The loss curve in tensorboard looked reasonable, but further optimization of this hyperparameter could help our model improve on the QA problem. Similarly, we also didn't widely experiment with number of adaptation steps or number of sampled examples, and instead chose to lock in those variables based on prior work we had seen from examples in the MAML paper and in the learn2learn library. We expect that if we spent more time optimizing these hyperparameters, our Meta-base model could have reached a similar level of performance to the baseline model.

Our most promising result is the jump from Meta-base to Meta-FS-base compared to the jump from Baseline to FS-base. Both Meta-base and Baseline were finetuned on the out-of-domain train datasets in the same way with the same hyperparameters. However, the increase in EM and F1 scores from Meta-base to Meta-FS-base (EM:+5.76 and F1:+3.72) was far larger compared to the increase from Baseline to FS-base (EM:+2.88 and F1:+2.11). This indicates that our Meta-base model was more effective at quickly learning on the out-of-domain datasets than the baseline model. The whole goal of meta-learning is to produce a model that can learn quickly, so we were happy to see this promising result. Assuming that we could improve our Meta-base model through hyperparameter tuning to

reach a level of performance on par with Baseline, we would expect Meta-FS-base to become our best-performing model.

Meta-Learning Performance Evaluation: After seeing our results and experimenting with artificial task augmentation, we began to question if our choice to use meta-learning for this project was actually a good choice. Is this specific problem of being given three in-domain training datasets with 50,000 examples each and then three out-of-domain datasets with 127 training examples each actually a good fit for meta learning? As we discussed earlier, meta-learning requires a distribution of well-formed tasks which the model can learn from. We initially thought that making each dataset its own meta learning task made intuitive sense as each dataset is unique. However, we are now questioning how distinct these tasks actually are and if those differences are appropriate for meta-learning. In the original MAML paper [2] and other papers that work with the MAML training regime [14], tasks are clearly differentiated and unique. In our case, although the three datasets are in different domains, some of the types of questions asked are quite similar. This may indicate that the tasks are not well formed when we define each dataset as a unique task. Furthermore, it's not clear that our out-of-domain datasets are made up of novel tasks. Instead, the sorts of questions found in the out-of-domain questions may be similar in character to questions found in the in-domain datasets, though they are typically more challenging.

Task Augmentation: Finally, our results indicate that task augmentation through simple dataset subdivision does not lead to increased performance. Overrepresenting NewsQA in the training regime, which we thought was most similar to RACE, the dataset our model was struggling on the most, did not lead to better overall performance. Also, splitting up the datasets evenly based on context length to artificially have 9 training tasks to sample from instead of 3 did not help. In both cases, the performance of Meta-FS-Base was higher. For the even split experiment, the result intuitively makes a lot of sense as the distribution of the data didn't change between Meta-Even-Split and Meta-FS-base. Likely the cause in both cases was that the tasks we artificially added were not well formed and meaningfully different than the other tasks we already had. Therefore, our model was not able to improve at quickly learning to adapt to new tasks.

5 Analysis

It was interesting to look at actual examples of context, question, answer triplets and compare how our best meta-learning model, Meta-FS-base, compared to our best non-meta-learning model, FS-base. They each performed surprisingly well in many situations, and we were impressed to see the answers they were able to correctly extract. Both models showed a somewhat deep level of understanding for certain passages. However, they were far from perfect. Both models often selected a long span of text that contained the correct answer but also contained many surrounding words. For example, when asked *"What instrument is Yellow River Piano Concerto scored for? ANSWER: piano"*, FS-base responded "piano concerto", and Meta-FS-Base responded "piano concerto arranged by a collaboration between musicians including Yin Chengzong and Chu Wanghua". This issue seemed to suggest regularization of the span detection based on its length could be useful, encouraging the model to generate the shortest good answer it could. We rarely saw cases where the models did not predict a long enough span. Also, both models struggled with difficult questions phrased in passive voice, and cases where the context surrounding an answer contained a double negative.

From looking at numerous examples, the FS-base model seemed to us to be a fundamentally better question answering model than Meta-FS-base. This can be seen through both correct answers as well as their incorrect guesses. In one example, when asked *"Whats the name of the English woman? ANSWER: Elizabeth Hadley"*, FS-base responded correctly, while Meta-FS-Base responded "Nathan Muir." Not only is that the incorrect answer, but it is a terrible guess, as that is clearly a man's name, and the question was asking for a woman. These sorts of mistakes were made often by Meta-FS-base compared to FS-base, implying that FS-base had learned better generalist question answering skills. In addition, when they were both incorrect, Meta-FS-base tended to give worse guesses than FS-base. For example, when asked *"Who knocks Pinky down in a freak car accident? ANSWER: Nandu"*, FS-base guesses "Lafangey Parindey" while Meta-FS-base guesses "one-shot." Clearly, FS-base has a better idea of what a good answer might look like.

However, there were some situations where Meta-FS-base seemed better than FS-base at answering complicated questions. For example, consider the following question:

Context: *Whaam! was first exhibited at the Leo Castelli Gallery in New York City in 1963, and purchased by the Tate Gallery, London, in 1966.*
Question: *What is the name of the place where Whaam! can be found?*
Answer: *Tate*
FS-Base: *Leo Castelli Gallery*
Meta-FS-Base: *Tate Gallery*

Here, FS-Base gets tripped up, and selects the gallery where Whaam! was first exhibited, while Meta-FS-Base is able to reason that "can be found" is present-tense, and so should select the gallery where the art was most recently located. Our interpretation is that Meta-FS-base is able to learn more from the out-of-domain datasets when training on them compared to FS-base, and thus learn how to handle these trickier questions. This quality likely buoyed the EM and F1 scores of Meta-FS-base so they were somewhat similar to FS-base, despite it missing a lot of low-hanging-fruit in the area of choosing good, well formed answers.

6 Conclusion

In this project, we were able to implement a MAML-style meta-learning approach to finetune a pretrained DistilBERT on the dual tasks of question answering and domain adaptation. We reasoned that by training a model to learn quickly from few examples, it could perform better on a novel dataset with a small amount of training data. We found evidence that this hypothesis was correct, but at the same time, our meta-trained model lost some fundamental competency compared to our baseline model. These gains and losses more or less cancelled out, resulting in a model which matched the performance of our fine-tuned baseline. We believe that with further work, we could optimize our meta-training regime to match the competency of the baseline, nullifying the losses and resulting in a substantive improvement over our fine-tuned baseline. Finally, we experimented with augmenting our datasets for meta-learning by subdividing them into salient sub-tasks, and found that this style of data augmentation is not useful. In all, we are proud to have successfully implemented a complex training regime for a deep language model, and to have shown it has the potential to help question answering systems adapt to novel, data-poor domains.

References

[1] Zhuosheng Zhang, Junjie Yang, and Hai Zhao. Retrospective reader for machine reading comprehension. *CoRR*, abs/2001.09694, 2020.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[3] Damien Teney and Anton van den Hengel. Visual question answering as a meta learning task. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[4] Wenbo Zheng, Lan Yan, Fei-Yue Wang, and Chao Gou. Learning from the guidance: Knowledge embedded meta-learning for medical visual question answering. In Haiqin Yang, Kitsuchart Pasupa, Andrew Chi-Sing Leung, James T. Kwok, Jonathan H. Chan, and Irwin King, editors, *Neural Information Processing*, pages 194–202, Cham, 2020. Springer International Publishing.

[5] Yuncheng Hua, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, and Wei Wu. Retrieve, program, repeat: Complex knowledge base question answering via alternate meta-learning. *CoRR*, abs/2010.15875, 2020.

[6] Konstantin Lopyrev Pranav Rajpurkar, Jian Zhang and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *CoRR*, 2016.

[7] Xingdi Yuan Justin Harris Alessandro Sordoni Philip Bach-man Adam Trischler, Tong Wang and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Association for Computational Linguistics*, 2017.

[8] Olivia Redfield Michael Collins Ankur Parikh Chris Alberti-Danielle Epstein Illia Polosukhin Matthew Kelcey Jacob Devlin Kenton Lee Kristina N. Toutanova Llion Jones Ming-Wei Chang Andrew Dai Jakob Uszkoreit Quoc Le Tom Kwiatkowski, Jennimaria Palomaki and Slav Petrov. Natural questions: a benchmark for question answering research. In *Association for Computational Linguistics*, 2019.

[9] Mitesh M. Khapra Amrita Saha, Rahul Aralikatte and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *Association for Computational Linguistics*, 2018.

[10] Hanxiao Liu Yiming Yang Guokun Lai, Qizhe Xie and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[11] Eunsol Choi Omer Levy, Minjoon Seo and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *arXiv*, 2017.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[13] Alexandru Cioba, Michael Bromberg, Qian Wang, Ritwik Niyogi, Georgios Batzolis, Da-Shan Shiu, and Alberto Bernacchia. How to distribute data across tasks for meta-learning? volume abs/2103.08463, 2021.

[14] Han-Jia Ye and Wei-Lun Chao. How to train your MAML to excel in few-shot classification. *CoRR*, abs/2106.16245, 2021.