Does MAML Only Work via Feature Re-use?

Anonymous Author(s)
Affiliation
Address
email

Abstract

It has been recently observed that a good embedding is all we need to solve many few-shot learning benchmarks. In addition, other work has strongly suggested that Model Agnostic Meta-Learning (MAML) mostly works via this same method – by learning a good embedding. This highlights our lack of understanding of what meta-learning algorithms are doing and when they work. In this work we provide empirical results that shed some light towards understanding meta-learning algorithms better. In particular we identify three interesting properties: 1) In contrast to previous work, we show that it is possible to define a family of synthetic benchmarks that result in a low degree of feature re-use – suggesting that current few-shot learning benchmarks *might not have the properties* needed for the success of meta-learning algorithms; 2) meta-overfitting occurs when the number of classes (or concepts) are finite, and this issue disappears once the task has an *unbounded* number of concepts (e.g. online learning); 3) more adaptation at meta-test time with MAML does not necessarily result in a significant representation change or even an improvement in meta-test performance – even when training on our proposed synthetic benchmarks. Finally, we suggest that, to understand meta-learning algorithms better, it is imperative that we go beyond tracking only absolute performance and in addition formally quantify the degree of meta-learning and track both metrics together. Reporting results in future work this way will help us identify the sources of meta-overfitting more accurately, and hopefully design more flexible meta-learning algorithms that learn beyond fixed feature re-use.

1 Introduction

Few-shot learning is a research challenge that assesses an artificial intelligence (AI) model's capacity to quickly adapt to new tasks or new environments. This has been the leading area where AI researchers apply meta-learning algorithms - where a strategy that learns to learn quickly is likely to be the most promising. However, it was recently shown by Tian et al. [20] that a model that only has a good embedding is able to match and beat many modern sophisticated meta-learning algorithms. In addition, there seems to be growing evidence that this is a real phenomena [1, 3, 7, 13]. Furthermore, analysis of the representations learned by Model Agnostic Meta-Learning (MAML) [9] (on few-shot learning tasks) revealed that MAML mainly works by learning a feature that is re-usable for many tasks [19] – what we are calling a good embedding in this paper.

These discoveries reveal a lack of understanding on when and why meta-learning algorithms work and are the main motivations for this work. In particular our contributions are:

1. It is possible to define a synthetic task that results in lower degree of feature re-use, thus suggesting that current few-shot learning benchmarks might not have the properties needed for the success of meta-learning algorithms;

Submitted to the 5th Workshop on Meta-Learning at NeurIPS 2021, held virtually. Do not distribute.

2. Meta-overfitting occurs when the number of classes (or concepts) are finite, and the issue disappears once the tasks have an unbounded number of concepts;
3. More adaptation for MAML does not necessarily result in representations that change significantly or even perform better at meta-test time.

2 Unified Framework for Studying Meta-Learning and Absolute Performance

We propose that future work on meta-learning should not only report absolute performance, but also quantify and report the degree of meta-learning. In addition, for us to be able to understand and trust such a system, we need metrics that can diagnose basic issues, e.g. if the system is meta-overfitting – defined as when the system has a high degree of meta-learning *coupled* with a high gap between meta-train and meta-test errors.

In this work, we make an important first step, inspired by [19], that defines the degree of meta-learning by measuring the normalized degree of change in the representation of a neural network nn_θ after using meta-adaptation A:

$$ML(nn_\theta) = \text{Diff}(nn_\theta, A(nn_\theta)). \tag{1}$$

In this work we set $ML(nn_\theta)$ to be distance based Canonical Correlation Analysis (dCCA) [17]. Note that dCCA is simply 1 minus CCA to switch the similarity based metric to a difference based metric is between 0 and 1.

3 Benchmarks that Require Meta-Learning

3.1 Background

Model-Agnostic Meta-Learning (MAML). The MAML algorithm [9] attempts to meta-learn an initialization of parameters for a neural network that is primed for quick gradient descent adaptation. It consists of two main optimization loops: 1) an outer loop used to prime the parameters for fast adaptation, and 2) an inner loop that does the fast adaptation. During meta-testing, only the inner loop is used to adapt the representation learned by the outer loop.

Feature re-use. In the context of MAML, this term usually means that the inner loop provides little adaptation during meta-testing, when solving an unseen task. In particular, Raghu et al. [19] showed that MAML has little representation change as measured with CCA and CKA after adaptation, during meta-testing on the mini-ImageNet few-shot learning benchmark.

3.2 Motivation for Our Work

The analysis by Raghu et al. [19] showing that MAML works mainly by feature re-use is the main motivation for our work. However, we argue that their conclusion is highly dependent on the benchmark used. This motivates us to construct a different benchmark and show that by *only* constructing a different benchmark, we can exhibit lower degrees of feature re-use in a statistically significant way. Therefore, our goal will be to show a lower degree of feature re-use than them. In particular, their work [19] showed that the representation layer of a neural network trained with MAML had a dCCA of 0.1 ± 0.02 [19]. *Therefore, our concrete goal will be to show that the dCCA on our task is greater than* 0.12. If this is achieved, it is good evidence that this new benchmark benefits from meta-learning and can be detected at a higher degree than previous work [19] in a statistically significant way. This is our main result of this section and is discussed in detail in Section 3.3.3.

3.3 Synthetic Task that Requires Meta-learning

3.3.1 Overview and Goal

The main idea is to sample functions to be approximated, such that the final layer needs little or no adaptation but the feature layers require a large amount of adaptation. This type of task would forcibly require that the meta-learner at least learns a representation that needs the feature layers to change to

achieve good meta-test performance (i.e. it cannot rely solely on feature re-use). Therefore, to perform well, not only would it be good to adapt the representation layers, but additionally performance is likely to be obtained from a (meta-learned) initialization that is primed to changed flexibly. In other words, tasks must not all have the same shared representation to be solved for meta-learning to be most useful and detectable.

3.3.2 Definition

In this section we describe a family of benchmarks that exhibits detectable meta-learning and requires more than a re-usable representation layer to be solved. We propose a set of regression functions specified as a fully connected neural network (FCNN), such that the magnitude of parameters of the representation are larger than the head. In particular we sample the parameters of the representation layer from a Gaussian with a larger standard deviation, compared to the parameter sampling of the head. We define the representation layer to be the first $L-1$ layers, and the head to be the final layer.

Next we describe the process to sample one function (regression task) from a Gaussian distribution. We have two pairs of benchmark parameters $[(\mu^{(1)}, \sigma^{(1)}), (\mu^{(2)}, \sigma^{(2)})]$: $(\mu^{(1)}, \sigma^{(1)})$ to sample the parameters for the representation layer, and $(\mu^{(2)}, \sigma^{(2)})$ to sample the parameters for the final layer. Then each regression task $f^{(t)}$ (with index t) is sampled as follows:

- Sample the representation parameters $w^{(l)} \sim N(\mu^{(1)}, \sigma^{(1)})$ for each layer $l \in [L-1]$ in the representation layers
- Sample the final layer parameters $w^{(L)} \sim N(\mu^{(2)}, \sigma^{(2)})$

The idea is that for some $c \in \mathbb{R}$ we have $\sigma^{(1)} > c \cdot \sigma^{(2)}$ such that the variance in tasks is due to the representation layers, and therefore adapting the representation layers is necessary. For all our experiments $\sigma^{(2)} = 1.0$. An example task can be seen in Figure 1. During meta-training, points are uniformly sampled from $[-1, 1]$, and the standard support set and query set are constructed by computing $f_w^{(t)}(x)$.

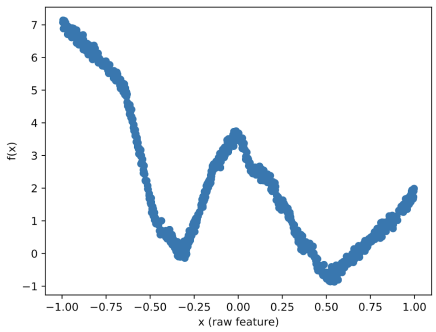

Figure 1: An example regression task constructed as described in Section 3.3.2. Addressing such tasks requires high degree of meta-learning.

3.3.3 Results on Benchmarks that Require Meta-Learning

In this section we show a higher degree of meta-learning and a lower degree of feature re-use from an initialization trained with MAML on the benchmarks described in Section 3.3.2. In particular we show this in Figure 2 because the dCCA value exhibited is much larger than 0.12 of previous work [19]. *Most importantly, the results are statistically significant, because the error bars do not intersect with the red dotted line with (worst case) dCCA value of* 0.12. The red dotted line is the top error band of previous work - i.e. the mean plus the standard deviation.

Note that a dCCA higher than 0.12 was observed across all of our experiments in over sixteen different benchmarks. In particular this happened even in models that had meta-overfitted e.g. see Figure 3. This is strongly suggestive that the benchmarks we defined in Section 3.3.2 require meta-learning, since they do not solely rely on feature re-use to be solved.

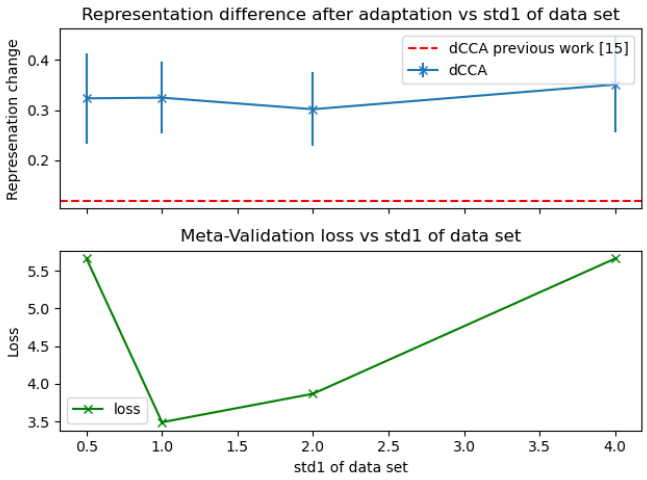

Figure 2: Shows the of lack of feature re-use and a higher degree of meta-learning, as the standard deviation of the representation layer $\sigma^{(1)}$ for generating regression. The x-axis is the standard deviation (std) of the parameter $\sigma^{(1)}$ for generating the tasks for the data sets. The models used for each point in the plot are models selected from early stopping (using the meta-validation MSE loss) when meta-trained with MAML. The models are the same architecture as the target function (4 layers fully connected neural network) with ReLU activation function. We also show the meta-validation loss vs the standard deviation of the task. The dCCA was computed by from the average and standard deviation over the representation layers, in this case the first three layers. The average is across different runs using the same meta-learned initialization. The red dotted line shows the value of 0.12 that our models have to be statistically significant. The only difference of this figure with respect to figure 3 is that we selected a model with the best validation here and in the figure 3 we selected the model in last step.

4 Meta-Overfitting

In this section, we show how being armed with the additional metric discussed in Section 2, we are able to identify an increasing gap between the meta-test and meta-train losses/accuracy – a term we refer to as *meta-overfitting*. In particular, this phenomena is observed when we meta-train models with MAML, and beceomes more pronounced as the number of iterations increases. We attribute this to the adaptation, because this increase in the meta-generalization gap is observed in conjunction to the low degree of feature re-use (as discussed in Section 3.3.3), and is most noticeable in our synthetic benchmarks compared to in mini-ImageNet [19]. Note that the dCCA of the models was much larger in our synthetic benchmarks than in mini-ImageNet. In addition, we show that if the number of regression tasks (in this case functions) is not fixed, then the meta-overfitting issue is no longer observed.

4.1 Finite Number of Tasks

When the number of regression tasks (functions) is finite (200 in our experiments), we consistently observe meta-overfitting. We show this in Figure 4 by increasing the meta-generalization gap (i.e. an increase in the difference between the meta-train and the meta-validation losses). This is consistently observed in over 30 experiments with a finite number of regression tasks.

Furthermore, meta-overfitting is also observed in a few-shot image recognition benchmark. This is shown in Figure 5 with mini-ImageNet. With a Pytorch ResNet-18 model, one can observe a meta-generalization gap of about 30%. With a state-of-the-art ResNet-12 [20], the meta-generalization gap is instead about 20%.

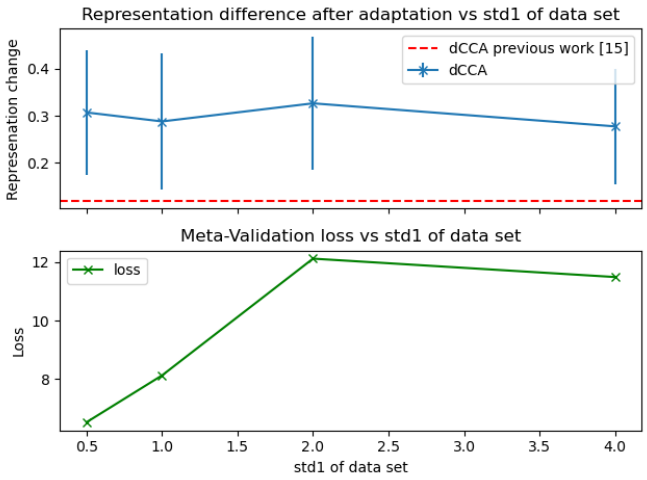

Figure 3: This figure supports the main result of the paper because a higher degree of meta-learning and a lack of feature re-use are present – even in models that are meta-overfitted. A meta-overfitted model can be easily obtained in our experiments by selecting a model at the final iteration. The x-axis is the standard deviation (std) of the parameter $\sigma^{(1)}$ for generating the tasks for the data sets. The red dotted line shows the value of 0.12 that our models have to be above for statistically significant results that support our claims. The only difference of this figure with respect to figure 2 is that we selected a model in last step (after trough and it had meta-overfitted) while in in figure 2 we select the model with lowest meta-validation loss.

4.2 Infinite Number of Tasks

We believe it is important to highlight that meta-overfitting was not observed when the number of regression tasks is unbounded, as shown in Figure 6. This suggests that, when the number of tasks is unbounded but sampled from a related set of tasks, meta-learning algorithms can leverage their power to adapt without meta-overfitting.

To measure the amount of meta-learning and the lack of feature re-use, we compute the dCCA value of the model as in Section 3.3.3 and observe a value of 0.31 ± 0.11. This also implies that the degree of meta-learning is higher when the number of tasks is unbounded.

5 Effects of More Meta-Adaptation

In this section, we show that increasing the number of inner steps for MAML during adaptation does not necessarily change the representation further as measured with dCCA (as in Equation 1). In addition, the meta-validation performance also does not change.

To show this, we obtain a single neural network meta-trained with MAML using a dataset as described in Section 3.3.2. Then we plot how the representation changes and how the meta-validation error changes as a function of the inner steps. We show this in Figures 7 and 8. We observe that the MAML neural networks are robust to meta-overfitting with respect to the inner steps of its inner adaptation rule.

Note that this is different from what was observed in Section 4.1, because that section shows it as a function of the meta iterations (what is sometimes called outer iterations). In addition, it is important to emphasize that the representation change in the plots is above the 0.12 compared to previous work [19], supporting the main results of section 3.3.3.

6 Related Work

Oh et al. [18] shows that one can encourage models to use less feature re-use purely algorithmically by setting the inner learning rate to zero for the final layer. They showed BOIL outperforms MAML

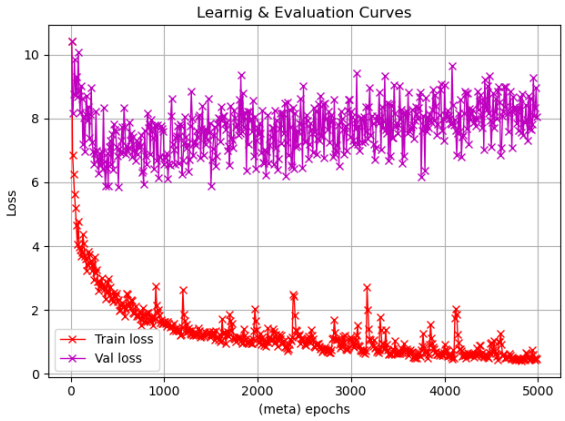

Figure 4: Shows meta-overfitting when the number of tasks (functions) is finite at 200 regression tasks because the meta-validation loss increases as the meta-train loss decreases. In particular the dCCA for this models was 0.36 ± 0.12 corresponding to $\sigma^{(1)} = 1.0$. The plot is the learning curve for a 4-layered fully connected neural network trained with MAML [9] using episodic meta-learning. Note that we use a (large) meta-batch size of 75 to decrease the noise during training in the figure. The main difference of this figure with figure 6 is that in this one has a finite set of tasks using our synthetic benchmark while the other has an infinite set of tasks using the sinusoid tasks suggested in [9].

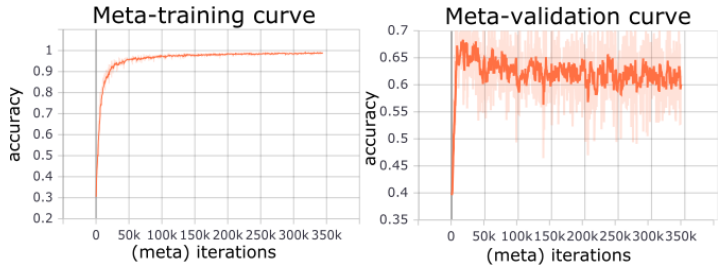

Figure 5: Shows that meta-overfitting is a real phenomenon in mini-Imagenet. We interpret this due to the peak in the meta-validation accuracy followed by a decline as the number of iterations increases. Importantly, the meta-train accuracy continues to increase as it converges. The model trained is an out-of-the-box PyTorch ResNet-18. Note that the higher noise of the meta-validation accuracy is due to having a meta-batch size of 2 to speed up experiments. We smoothed the meta-validation curve with a tensorboard smoothing weight of 0.8. We consistently saw that increases in meta-batch size lead to decreases in noise in the learning curves but we didn't re-run these experiments since it can take up to a week to reproduce a episodic meta-learning run - even on a Quadro RTX 6000.

in both traditional few-shot learning (e.g. meta-trained on mini-ImageNet then meta-tested on mini-ImageNet) and cross-domain few-shot learning (meta-trained on mini-ImageNet then meta-tested on tiered ImageNet). In particular, their cross-domain few-shot learning is similar in spirit to the synthetic task we propose in section 3.3.2. However, note that we show that even MAML (and algorithm that has been shown to work by feature re-use [19, 18]) can exhibit large representation changes if it is trained solely on a task that requires large feature changes. Concisely, we encourage rapid learning by only changing the task while Oh et al [18] do encourage it by changing the algorithm itself.

Guo el al.'s [11] work is similar to ours in that they focus on defining a benchmark more appropriate for meta-learning and transfer learning. They propose that meta-learning should be done in a fashion where the distribution of tasks sampled changes considerably when moving from meta-training to meta-evaluation. Our work is different in that we emphasize more that the meta-training tasks themselves need to have diversity to be able to encourage meta-learning. Although Guo et al.'s [11] meta-evaluation procedure is excellent, we believe - based on our results - that their benchmark won't

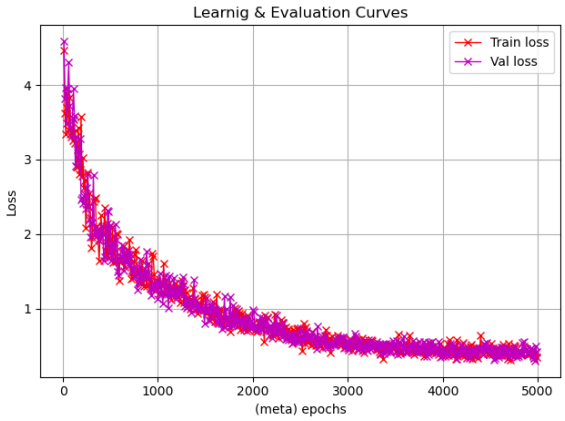

Figure 6: Shows that meta-overfitting does not occur and perfect meta-generalization occurs when the number of tasks (functions) is unbounded when training with MAML. In other words the meta-train and meta-validation error are indistinguishable and decrease together as the meta-iterations increases. The main difference of this figure with figure 4 is that in this one has a finite set of tasks using our synthetic benchmark while the other has an infinite set of tasks using the sinusoid tasks suggested in [9].

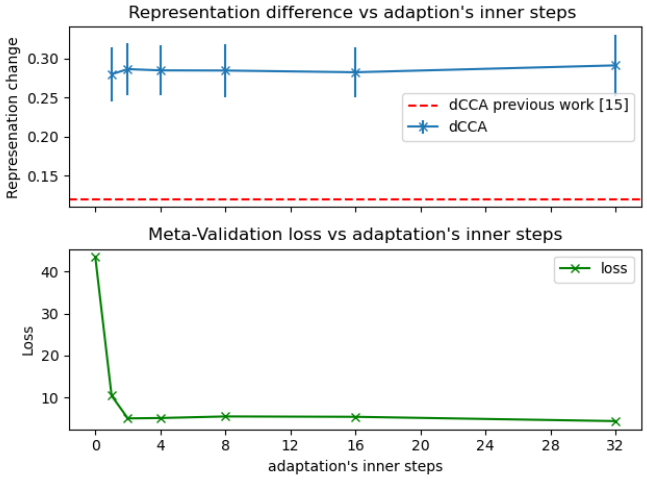

Figure 7: Shows 1) the lack of representation change and b) meta-validation change as the number of inner steps increases. 1 is shown by the relative flatness of the blue and orange lines in the upper plot. Similarly 2 is shown by the flatness of the green line in the lower plot. In particular notice that we exponentially increase the inner steps from 1 to 2 to 32. The models used are 4 layered FCNN tained with MAML with 1 inner step and 0.1 inner learning rate, selected using early stopping using the meta-validation set with the Sigmoid activation function. The only difference of this figure with figure 8 is that this figure uses a sigmoid activation and the other one uses a ReLU. Note that this is the model used for figure 4. Note the dCCA value remains above 0.12 suggesting lower degree of feature re-use.

have enough diversity to encourage large representation changes during meta-traning. However, we believe that combining our ideas and their's to design one benchmark is a promising step for creating a better benchmark for meta-learning.

Similar work by Triantafillou et al. [21] attempt to improve benchmarks by merging more data sets but we believe their data sets are not diverse enough to achieve this. In terms of methods our work is most similar to Raghu et al. [19], but they lack an analysis of the role of the tasks in explaining their observations. There is also other work [1, 3, 7, 13] that shows that a good representation is

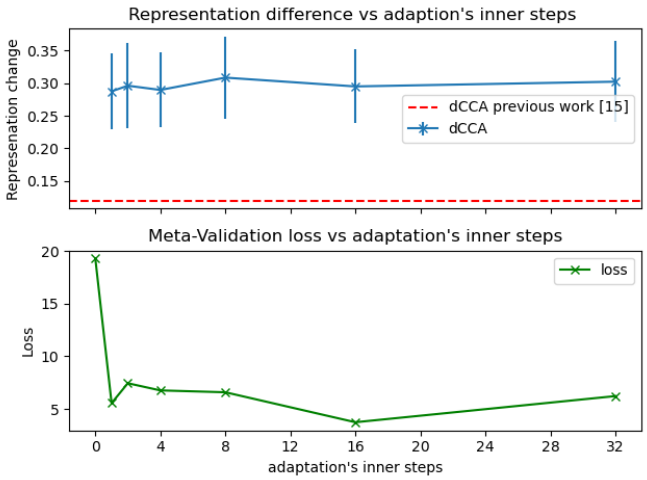

Figure 8: Shows 1) the lack of representation change and b) meta-validation change as the number of inner steps increases. 1 is shown by the relative flatness of the blue and orange lines in the upper plot. Similarly, 2 is shown by the flatness of the green line in lower figure. We want to emphasize that we exponentially increase the inner steps from 1 to 2 to 32. The models used are 4 layered FCNN tained with MAML with 1 inner step and 0.1 inner learning rate, selected using early stopping using the meta-validation set with the ReLU activation function. The only difference of this figure with figure 7 is that this figure uses a ReLU activation and the other one uses a sigmoid. Note the dCCA value remains above 0.12 suggesting lower degree of feature re-use.

sufficient to achieve a high meta-accuracy on modern few-shot learning tasks e.g. mini-Imagenet, tiered-Imagenet, Cifar FS, FC100, Omniglot, [20], which we hope to analyze in the future. We also believe in is imperative that a definition of meta-learning is developed and conned to the general intelligence. Chollet [4] takes this direction but to our understanding the proposed definition is mostly focused for program synthesis. We also hope that in the future a metric for AI safety is ubiquitously reported as suggested in Miranda et al. [15].

7 Discussion

We believe it's exciting that by only changing the few shot learning benchmark one can consistently showed higher degrees of representation changes as measured by two different metrics. We believe this is the case because the meta-learning system has to be trained explicitly with a task that demands it to learn to adapt.

An important discussion point is the lack of an authoritative definition for measuring meta-learning in our work and in the general literature. In particular in our work we decided to not report any results with CKA. We decided this because Ding et al [8] showed that it's possible remove up to 97% of the principal components of the weights of a layer until CKA starts to detect it. Thus, we used dCCA which doesn't have the problem. It instead has a higher variance but it's easier to address this with experiment repetition sand error bars (which we did). However, we believe it would be interesting to use and extend Orthogonal Procrustes as suggest by [8] in future work.

The most obvious gap in our work is a thorough analysis with a real world vision data set. We hope to repeat our work with the hinted extension in section 6 benchmarks as suggested in [11, 21].

In addition, Figures 7, 8 shows that as the number of inner steps increases, the dCCA does not increase. This is somewhat surprising given the meta-overfitting results observed in section 4 and further experiments would be valuable.

8 Broader Impact

8.1 Quantifying general intelligence through meta-learning

There is valuable efforts that try to make benchmarks which require higher level cognition e.g. [22]. An example of work that tries to quantify AGI and proposes a benchmark is [4]. We believe the second approach is likely to have more impact in the long run because it also deliberately quantifies general intelligence. We believe that suggesting benchmarks without clearly specifying the long term goal or measuring the metric we are trying to optimize is a suboptimal approach. However, we do believe grounding benchmarks on tasks that humans are able to perform is a good idea but suggest to augment these proposals with metrics and explicit discussions of general intelligence.

Another approach we believe has high potential is program synthesis [2] and theorem proving [16, 5] because humans create higher abstractions that are composed and re-sue thus suggesting to meta-learning might be taking place. We believe that higher level cognition tasks are a challenging to assess meta-learning algorithms.

8.2 Quantifying AI safety

We also believe quantifying and tracking metrics for AI safety as early as possible is crucial. Few-shot learning is likely one of simplest - and arguably the atomic buildings blocks for general intelligence. We believe AI safety could be enriched if research community deliberately tracks, discusses and report it in all it's research - especially in meta-learning research. For a brief discussion see [15].

8.3 Summary of Broader Impact

We hope that this discussion inspires the AI community - but especially the meta-learning research community - to always report their progress using, what Miranda et al. [15] call the "the big three": 1) the score for absolute performance (to ensure usefulness) 2) the score for general skill acquisition (to ensure flexibility and general intelligence) and 3) the AI safety score (to ensure positive outcome).

References

[1] Wei-Yu Chen et al. "A Closer Look at Few-shot Classification". In: *7th International Conference on Learning Representations, ICLR 2019* (2019). URL: http://arxiv.org/abs/1904.04232.

[2] Xinyun Chen et al. "Compositional generalization via neural-symbolic stack machines". In: NeurIPS (2020).

[3] Yinbo Chen et al. *A New Meta-Baseline for Few-Shot Learning*. Tech. rep. URL: https://github.com/.

[4] François Chollet. "On the Measure of Intelligence". In: (2019). eprint: 1911.01547. URL: http://arxiv.org/abs/1911.01547.

[5] Maxwell Crouse et al. *A Deep Reinforcement Learning Approach to First-Order Logic Theorem Proving*. Tech. rep. URL: https://arxiv.org/abs/1911.02065.

[6] Tristan Deleu et al. *Torchmeta: A Meta-Learning library for PyTorch*. Available at: https://github.com/tristandeleu/pytorch-meta. 2019. URL: https://arxiv.org/abs/1909.06576.

[7] Guneet S. Dhillon et al. "A Baseline for Few-Shot Image Classification". In: (2019). URL: http://arxiv.org/abs/1909.02729.

[8] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. "Grounding Representation Similarity with Statistical Testing". In: (). arXiv: 2108.01661v1. URL: https://github.com/js-d/sim_metric..

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: (2017). URL: http://arxiv.org/abs/1703.03400.

[10] Edward Grefenstette et al. "Generalized Inner Loop Meta-Learning". In: *arXiv preprint arXiv:1910.01727* (2019).

[11] Yunhui Guo et al. "A Broader Study of Cross-Domain Few-Shot Learning". In: (Dec. 2019). arXiv: 1912.07200. URL: http://arxiv.org/abs/1912.07200.

[12] Ryuichiro Hataya. *anatome, a PyTorch library to analyze internal representation of neural networks*. 2020. URL: https://github.com/moskomule/anatome.

[13] Shaoli Huang and Dacheng Tao. "All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning". In: (2019). URL: http://arxiv.org/abs/1911.12476.

[14] Brando Miranda. "DiMO : Differentiable Model Optimization and metaDiMO". In: *Illinois Digital Environment for Access to Learning and Scholarship (IDEALS)* (2019).

[15] Brando Miranda. "Establishing the foundations of Meta-learning - a Proposal". In: *Illinois Digital Environment for Access to Learning and Scholarship (IDEALS)* (2020). URL: https://doi.org/10.1007/s11633-017-1054-2.

[16] Brando Miranda. "Sketching: a Cognitively inspired Compositional Theorem Prover that Learns to Prove - a Proposal". In: *Illinois Digital Environment for Access to Learning and Scholarship (IDEALS)* (2020). URL: https://www.ideals.illinois.edu/handle/2142/109134.

[17] Ari S Morcos et al. *Insights on representational similarity in neural networks with canonical correlation*. Tech. rep.

[18] Jaehoon Oh et al. "Does MAML really want feature reuse only?" In: (Aug. 2020). URL: http://arxiv.org/abs/2008.08882.

[19] Aniruddh Raghu et al. *Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML*. Tech. rep. URL: https://arxiv.org/abs/1909.09157.

[20] Yonglong Tian et al. *Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?* 2020.

[21] Eleni Triantafillou et al. "Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples". In: (2019). URL: http://arxiv.org/abs/1903.03096.

[22] Ramakrishna Vedantam et al. *CURI: A Benchmark for Productive Concept Learning Under Uncertainty*. Tech. rep. URL: https://arxiv.org/abs/2010.02855v1.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [No]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We include hyperparms in the supp section. But are happy better instructions for reproducibility and the our weight and biases (wandb) repo with recordings of our runs etc and we are happy to make them available upon acceptance

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No] Not every single detail but every details has been recorded and we are happy to make them available upon acceptance

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] More details in supp

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Supplementary Material

A.1 Experimental and hyper-parameter details

A.1.1 Details for experiments on our benchmark that requires more than feature re-use

All models were trained on a CPU cluster with intel CPUs. All models were 4 layered FCNN. All models had batch-normalization and collected running statistics during meta-training but used batch statistics during training. MAML models for figure 2 and 3 had inner learning rate of 0.1 and 1 inner step. One inner super was chosen to further emphasizes the feature re-use since it is the lowest inner step we can choose (nothing lower exists except 0 which doesn't exhibit adaptation). Adam outer optimizer was used with learning rate 0.001 and default parameters. No learning schedulers were used but would be interesting to experiment with. Since there were 200 regression tasks, we trained the models with meta-epochs instead of meta-iterations. This means that we reported errors, losses etc. after all tasks were seen. Note that the input and target values were guaranteed to be novel because during meta-train and meta-testing we sample a function and generate data on the fly - similar to online learning. Note that this is very similar to how classification tasks for few-shot work (e.g. mini-Imagenet) because those tasks have a very limited number of image classes and thus results in highly correlated tasks. In addition, we showed how both exhibited similar meta-overfitting. For CCA value computation we used a query set size of 100 due to numerical issues with the Anatome library [12]. We did not use first order MAML. All models were trained until convergence with about $200,000$ meta-epochs.

For models with 8 and 7, we meta-trained with MAML but used 2 inner steps. The remaining hyperparameters remain the same. We did not use first order MAML.

A.1.2 Details for experiments on ResNet-18 meta-overfitting on mini-Imagenet

The ResNet-18 is a standard out of the box ResNet-18 downloaded from PyTorch. We trained the models for $5,000,000$ meta-iterations. We used 1 inner step with 0.1 inner learning rate. We used meta-batch size of 4 and 2 for meta-training and meta-testing respectively. We used an outer learning rate of 0.001 and Adam with default parameters. We did not use first order MAML.

A.2 Role of Backbone on meta-accuracy

In this section we describe the relation of the depth of a Pytorch ResNet model with the meta-test accuracy. The motivation for these experiments is that if we can close the gap on mini-Imagenet to over 90% by only increasing the back bone depth then this would provide strong evidence that such benchmarks really only need a good embedding. However, we discovered that for the ResNets used in [20] it seems that accuracy saturates at 80% (results not shown in paper) but when using the Pytorch models we see meta-overfitting and decreasing meta-test error 9. This suggests that even this simple scenario of few-shot learning still has still space for meta-learning to be a solution.

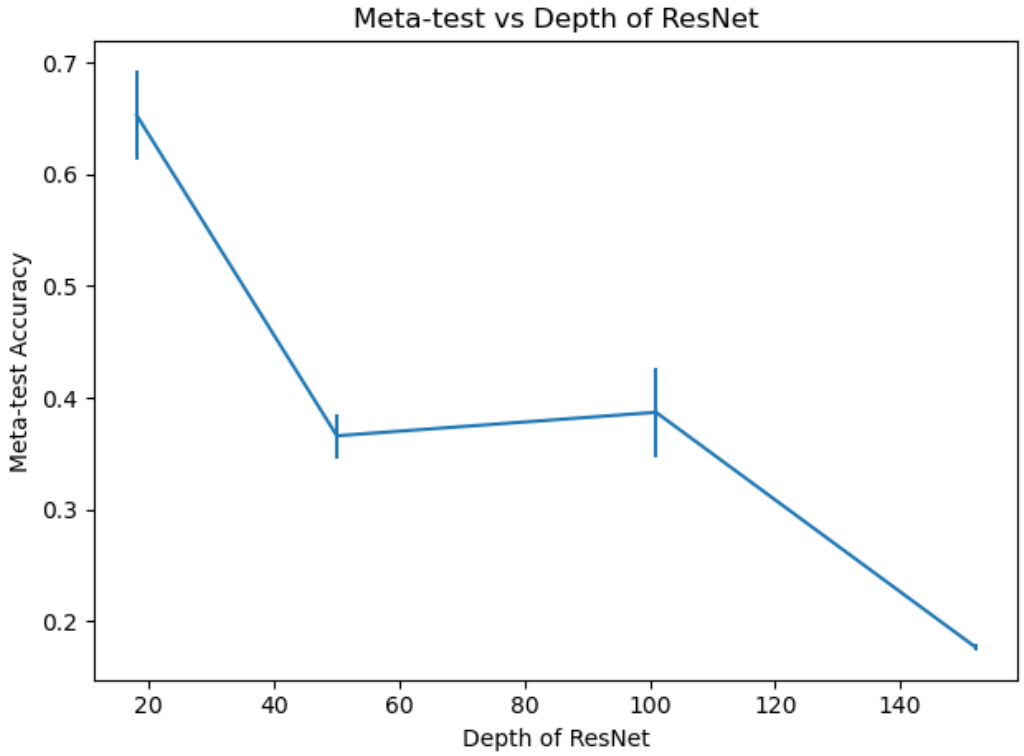

Figure 9: Shows that as the backbone of the Pytorch ResNets increases to 152 the meta-accuracy on mini-Imagenet decreases. These models were trained with supervised union training is in [20]. The meta-adaption algorithm used logistic regression and was adapted to convergence on the final layer as in [20]. When using the Pytorch ResNet models instead of the special ResNets designed for mini-imagenet [20] we observe see that the meta-accuracy decreases 9.

A.3 Analysis of meta-learned initialization

In this section we an experiment where fix the ResNet18 meta-learned initialization and use the adaption that only adapts the final later as in [20]. The results in figure 10 are mixed but it is interesting to note MAML with no inner steps performs worse than a random neural network. This result is interesting because this is very similar to supervised pre-raining in that no meta-learner is present during training but instead of seeing all 64 images it sees 5 randomly (but uses no meta-learner).

We would have expected that the initialization obtained would have been equivalent to one with supervised pre-training. Since they are not it shows a MAML is at the very least capable of learning a representation that is invariant to concept permutation.

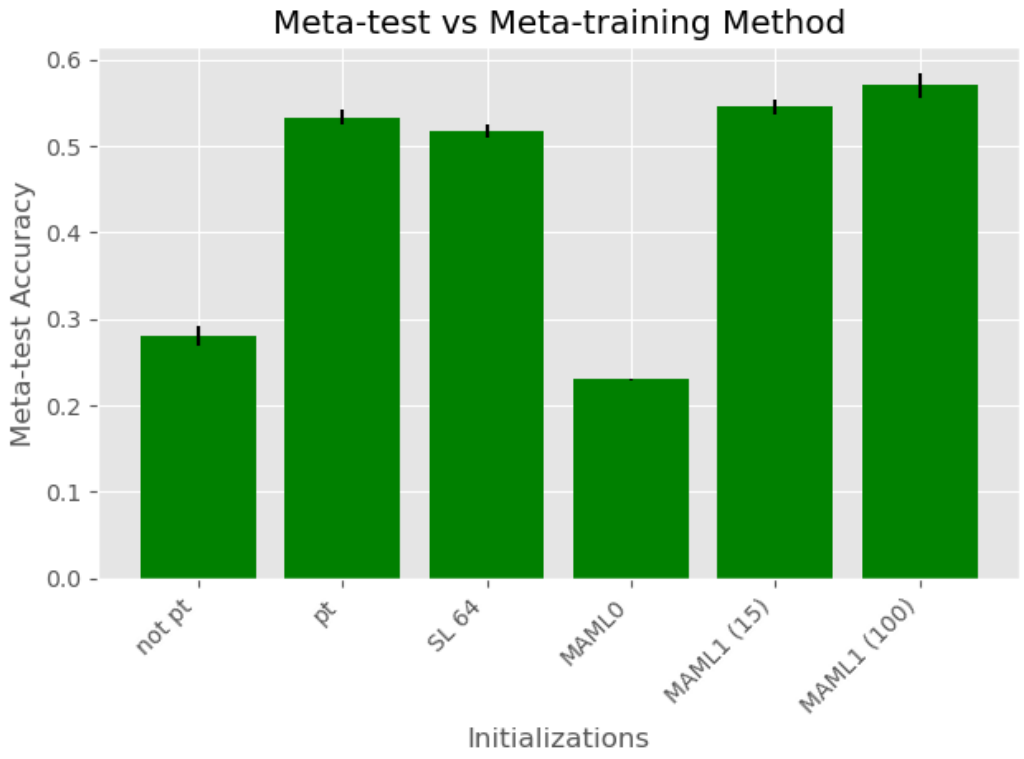

Figure 10: Shows relation of meta-test accuracy with models with a different meta-learned initialization. PT stands for Pre-trained on Imagenet. Not Pt stands for a random model. All models are ResNet18s from PyTorch. SL 64 stands for supervised union pre-training on mini-imagenet using all 64 labels during meta-training. MAML0 stands for only using episodic meta-training (i.e. MAML with zero inner steps). MAML1 (15) and MAML1 (100) stand for training using MAML with a query set of size 15 to 100. The meta-adaptation is the same as in [20] (training logistic regression in the final layer to convergence).

A.4 Training with zero number of inner steps

We believe it is an interesting observation that MAML with 0 inner steps (MAML0) (i.e. only using episodic meta-training) resulted in very different meta-learned initialization compared to MAML with 1 inner step (MAML1) on mini-Imagenet. Previous work observed that supervised pre-training [20] with all 64 images during meta-training results in a strong baseline. With this in mind it is natural to ask: what is the difference between seeing all 64 images during supervised pre-training or seeing only 5 using episodic training? With this in mind we trained MAML0 and obtained a model that performs at chance. Figure 11 compares MAML0 with MAML1 to show that MAML0 obtains a model that has a very high meta-training loss. Additionally, figure 10 shows such an initialization performed even worse than random. This is surprising but it seems that meta-learned initialization with MAML1 learn at least a model that is invariant to permutation of the order of the classes. Unfortunately, this result seems to only be reproducible in classification since training MAML0 in a synthetic regression task did converge to have model with low meta-train loss 12. This suggests future studies would be interesting to disentangle the casual factors.

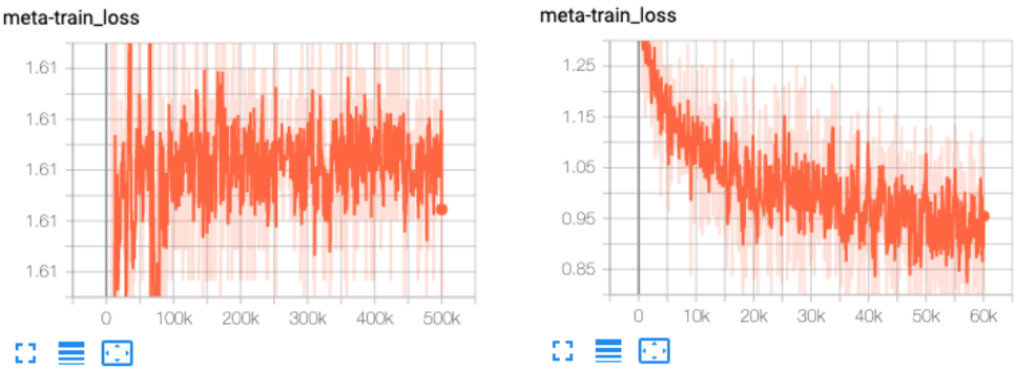

Figure 11: Compares MAML0 (only episodic training) vs MAML1 (MAML with 1 inner step). MAML0 remains close to chance with a high loss while MAML1 converges. This suggests MAML0 is not equivalent to supervised pre-training and that MAML1 does learns a representation that is invariant to class order permutation.

A.5 Tips and tricks for episodic meta-training

From our experiments we suggest the following (when episodic meta-training [9]):

1. Use a large number of query examples e.g. greater than popular 15 (since they often speeds up convergence of the meta-learning algorithm).
2. A large meta-batch size (since it's important to be able to have a low level of noise when tracking the meta-validation error/loss for doing early stopping). We found empirically for $75 - 100$ tasks to be a good meta-batch size.
3. Episodic training as suggested in [9] is expensive and takes at least a week to train on mini-Imagenet on a Quadro RTX 6000 using torchmeta and higher [6, 10], so these suggestions are important.
4. Experiments with synthetic data were run with CPU only.

A.6 Future work

A.6.1 Summary

1. Defining a synthetic benchmark that is a classification problem that also requires meta-learning (or rapid learning with MAML).
2. We also hope to construct a (real) benchmark from images that requires meta-learning. Formally, we propose a good start would be a benchmark were the probability of two task having the same class be small, otherwise we are more likely to see overfitting. Alternatively, a benchmark that requires the tasks to be different by at least requiring a different representation. We believe compositionality is an ideal benchmark since this would allow sophisticated re-use of lower level representations and simultaneously have an unbounded number of tasks. Humans are able to richly and flexibly cope with both. Additionally, it would be interesting to be able to quantify the distance between two different N-way, K-shot tasks to make these ideas more rigorous.
3. An interesting benchmark with a large number of classes with real images is taking the union of many vision classification tasks and re-scaling all images to be of the size of mini-Imagenet.
4. Plotting the meta-generalization gap (with a synthetic classification task) and demonstrate it decreases as the number of tasks increases would be interesting (note however we already have the limiting case when the number of tasks is unbounded and the meta-generalization gap is zero).

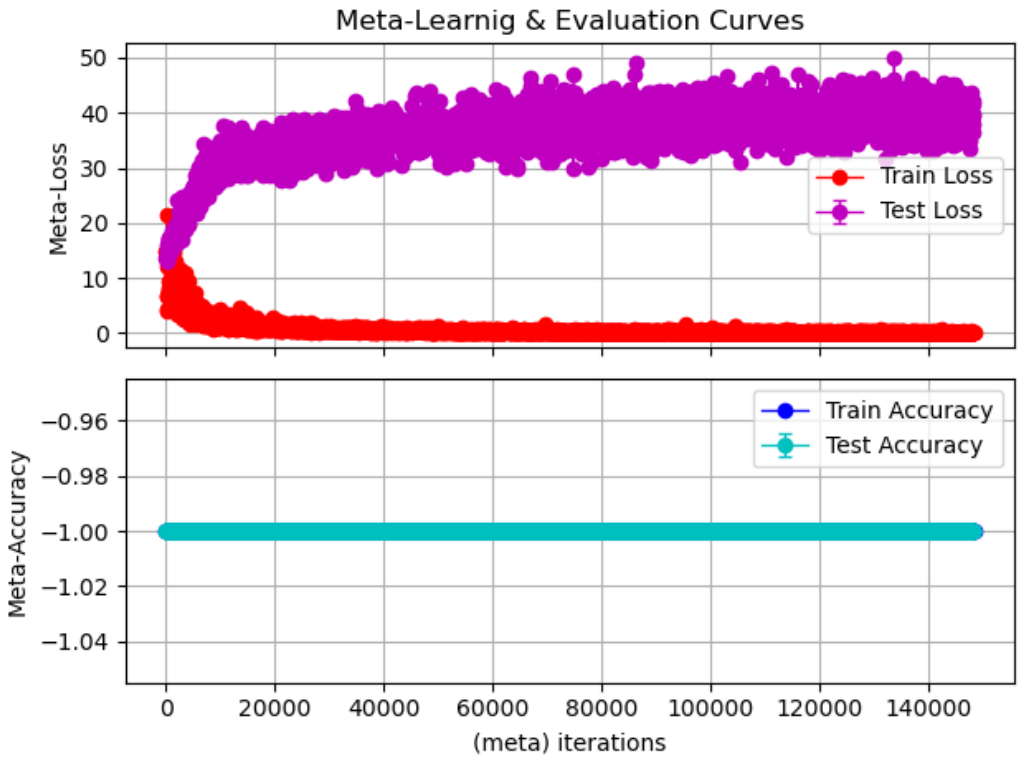

Figure 12: Shows MAML0 (only episodic training) getting zero meta-train loss (red curve) for a synthetic regression task. This suggests that meta-learning in regression and classification might not be enitrely equivalent. Note that meta-overfitting is still observed (purple curve). This is a regression task so the blue curves can be ignored.

5. An interesting experiment would be to train a deep neural network with the episodic training (but without the MAML inner loop) but have an unbounded number of tasks and see if the test error keeps increases (or stays at chance as observed when this is done with mini-Imagent 11).
6. An interesting hypothesis to investigate is if meta-learning algorithms get representation that are optimal for their respective meta-learner (or adaptation rule). If this is true it means methods like [20] can be improved by making the entire pipeline differential and learning it end-to-end [14].
7. Test meta-learning algorithms in domains were higher level cognition is required and thus compositionality is essential e.g. program synthesis [2] and theorem proving [16, 5].

A.6.2 Proposal on Synthetic classification task that possibly require meta-learning

Synthetic tasks that use classification instead of regression are not hard to define. Two possible alternatives are: 1) a mixture of Gaussians but the standard deviation controls the radius of limit of how far the classes can be from each other 2) another option is the similar as with a mixture of Gaussians but have the (vector) samples be weights of a Neural Networks (so that the goal is to identify from which Neural Network data is coming from)