A Supplementary Material

A.1 Related Work (Cont.)

The ARC benchmark (Chollet, 2019) was designed with AGI in mind – potentially the ultimate meta-learner. However, ARC's focus is primarily on visual reasoning utilizing program synthesis techniques. We believe it's a promising path, but our work inspires extensions that transcend program synthesis approaches.

The study by Chen et al. (2021) offers, to our knowledge, the first non-vacuous generalization bounds for the (supervised) meta-learning setting. However, their results do not aim to differentiate classes of meta-learning, as our work attempts to do empirically.

The work by Wang et al. (2021) proposes the concept of global labels, equivalent to what we call USL in our paper. Their theoretical analysis, however, is dependent on a fixed feature extractor, and fails to accommodate different feature extractors that might be trained, such as comparing USL vs MAML directly in an end-to-end fashion. This was partially addressed theoretically and empirically in Miranda et al. (2022).

The study by Denevi et al. (2020) presents a theoretical treatment of meta-learning using meta-learners with closed-form equations derived from ridged regularization using fixed features. However, their results are highly theoretical, whereas ours focus on empirical results, and they do not explore their findings in the context of modern few-shot learning benchmarks like MiniImagenet, Cifar-fs, FC100, TieredImagenet, Meta-Dataset, etc. like we do.

The work by Goldblum et al. (2020) provides strong evidence that adaptation at test time is best done when the meta-trained model matches the adaptation it was meta-trained with. However, their results cannot beat Tian et al. (2020) and thus do not help separate the role of meta-training and pre-training (PT) with a union of all the data.

The study by Gao & Sener (2020) provides theoretical bounds of when the expected risk of MAML and DRS (Domain Randomized Search) by bounding the gradient norm. However, they do not provide in depth empirical analysis with respect to any real few-shot learning benchmarks like MiniImagenet or Cifar-fs.

The work by Rosenfeld et al. (2021) provides a theoretical analysis on the difference between interpolation and extrapolation in transfer learning. We believe this type of theory may be helpful as an inspiration to explore why in the high diversity regime there seems to be a difference between the performance of meta-learning and transfer learning or pre-trained methods.

Finally, the work by Miranda (2020b;a) first demonstrated that there exist synthetic data sets capable of exhibiting higher degrees of adaptation compared to the original work by Raghu et al. (2020). Their main focus was on comparing adapted MAML models vs. unadapted MAML models, a difference from our approach in this paper.

Previous work demonstrated that in datasets with low diversity, the difference MAML and pre-training is small (Miranda et al., 2022). While we substantiate these results to a degree, we introduce a nuance that, when evaluated through the statistical lens of effect size, pre-training can outperform MAML. This subtle detail underscores the critical role of the selected statistical measure in the comparative analysis of these algorithms. Finally, we provide the final piece of evidence to complete their story (Miranda et al., 2022), for high diversity datasets. Crucially, we include the large scale meta-dataset (mds) and demonstrate that merging/unioning datasets is an effective mechanism for increasing the formal diversity of a dataset.

BiT (Kolesnikov et al., 2020) is a study that demonstrates good performance on a wide set of datasets (20) using transfer learning by pre-training on a large (JFT-300M) scale vision data set. They fine-tune the entire network (with SGD and momentum) during adaptation and provide heuristics for choosing the hyperparameters with the HyperRule heuristic. The main contrast with our work is that they do not do a direct fair comparison with meta-learning (like MAML) as we did. Contrary to this previous work that leaves the comparative merits of pre-training and meta-learning algorithms indeterminate, our work directly addresses this comparison as its primary focus. We think the authors Kolesnikov et al. (2020) should have compared their test time adaptation method that fine-tunes the entire paper to the one proposed by Tian et al. (2020) that uses optimal convergence at the final layer with gradient-based BFGS fine-tuning. We note they use a large dataset for pre-training, and it's important to use such a dataset for the training of MAML to be able to do a fair comparison between MAML and pre-training. Our experiments on meta-dataset suggest that on large-scale formally diverse dataset MAML might be marginally better than pre-training.

Memory efficient meta-learning with large images (Bronskill et al., 2021) demonstrates that if one subsamples the support set (using their method called LIME) to meta-train many meta-learning algorithms, then one can match the performance of a pre-training network that has been fine-tuned with 50 steps. The main contrast between their work and ours is: 1. They use confidence intervals to separate pre-training methods vs meta-learning methods, while we use effect size 2. We add another level of structure to the analysis by separating the results in datasets that have a formal low diversity vs a formal high diversity. This analysis shows that MAML in fact can outperform pre-training, although being small when using the effect size as the measuring metric. We posit that our work, in conjunction with Bronskill et al. (2021), provides a complete perspective on meta-learning – where we conjecture that meta-learning methods in general marginally outperform pre-training methods. Their work (Bronskill et al., 2021) supports our counter-narrative that pre-training methods are always better.

The Vendi Score is a recently proposed formal diversity score different from the diversity coefficient proposed in Miranda et al. (2022). The Vendi score is mainly a more sophisticated aggregation method than an expectation given pair-wise comparisons. Their aggregation score is interesting, but it is unclear what the advantages of it are compared to a simpler expectation. For a sample of n already embedded tasks (or data points), the Vendi score takes $O(n^3)$ (due to the use of eigenvalue computation), while ours uses the faster to compute expectation, which takes $O(n^2 - n/2) = O(n^2)$. We hope to explore the Vendi score in future work and compare it with the expectation aggregation score. However, the main weakness of the Vendi score that previous work address (Miranda et al., 2022) is the use of Task2Vec (Achille et al., 2019) to compute embeddings of tasks. The Vendi score assumes one already has such a comparison by assuming a Kernel/Grahm Matrix and unfortunately circumvents arguably the hardest part of the problem – computing effective embeddings a task. Their formulation also implies their analysis is mostly focused on individual data point diversity, while the diversity coefficient also works embedding tasks, batches, and even entire datasets.

The ranges of 0.2, 0.5, and 0.8 as small, medium and large effect sizes were proposed in Andrade (2020).

A.2 All Meta-test accuracy of a Pre-trained (PT) model vs. MAML

In this section, we report the raw meta-test accuracy of used (to compute the effect size) when comparing PT vs MAML models in the main body of the text section 4.

A.2.1 Meta-test accuracy of a Pre-trained (PT) model vs. a first-order (fo) MAML model on low diversity datasets

Meta-test accuracy (with 95% confidence intervals) of a Pre-trained (PT) model vs. a first-order (fo) MAML model on low diversity datasets are in table 9.

A.2.2 Meta-test accuracy of a Pre-trained (PT) model vs. a higher-order (ho) MAML model on low diversity datasets

Meta-test accuracy (with 95% confidence intervals) of a Pre-trained (PT) model vs. a higher-order (ho) MAML model on low diversity datasets are in table 10.

A.2.3 Meta-test accuracy of a Pre-trained (PT) model vs. a MAML model on high diversity datasets

Meta-test accuracy (with 95% confidence intervals) of a pre-trained (PT) model vs. MAML model on high diversity datasets are in table 11.

Table 9: **Meta-Test accuracy of a Pre-trained (PT) model vs. a first-order (fo) MAML model with 95% confidence intervals on low diversity few-shot learning vision datasets.** We used a meta-batch size of 300 few-shot learning tasks. The data sets' diversity is low, as shown in table 7. Resnet12 has 1,427,525 parameters.

Model (Dataset)	PT (test acc.)	MAML5 (test acc.)	MAML10 (test acc.)
Resnet12 (cifar-fs)	0.755 ± 0.0102	0.779 ± 0.00975	0.786 ± 0.00996
Resnet12 (fc100)	0.438 ± 0.00949	0.458 ± 0.00931	0.459 ± 0.00988
Resnet12 (mini-imagenet)	0.719 ± 0.00893	0.685 ± 0.00947	0.706 ± 0.0104
Resnet12 (tiered-imagenet)	0.788 ± 0.00945	0.769 ± 0.0107	0.786 ± 0.0107
Resnet12 (aircraft)	0.592 ± 0.010	0.659 ± 0.013	0.685 ± 0.011
Resnet12 (flower)	0.928 ± 0.005	0.856 ± 0.008	0.870 ± 0.007
Resnet12 (dtd)	0.610 ± 0.011	0.511 ± 0.012	0.528 ± 0.011
Resnet12 (delaunay)	0.735 ± 0.010	0.614 ± 0.012	0.632 ± 0.010
Resnet12 (cubirds)	0.787 ± 0.008	0.829 ± 0.008	0.821 ± 0.009
ResNet12 (vggair)	0.727 ± 0.027	0.745 ± 0.019	0.760 ± 0.019
ResNet12 (vggdtd)	0.737 ± 0.019	0.701 ± 0.022	0.701 ± 0.021

Table 10: **Meta-Test accuracy of a Pre-trained (PT) model vs. a higher-order (ho) MAML model with 95% confidence intervals on low diversity few-shot learning vision datasets.** We used a meta-batch size of 300 few-shot learning tasks. The data sets' diversity is low, as shown in table 7. Resnet12 has 1,427,525 parameters.

Model (Dataset)	PT (test acc.)	MAML5 (test acc.)	MAML10 (test acc.)
Resnet12 (cifar-fs)	0.753 ± 0.00941	0.804 ± 0.00982	0.809 ± 0.0107
Resnet12 (fc100)	0.432 ± 0.0102	0.503 ± 0.0100	0.489 ± 0.00988
Resnet12 (mini-imagenet)	0.721 ± 0.00889	0.704 ± 0.0100	0.732 ± 0.00952
Resnet12 (tiered-imagenet)	0.791 ± 0.00922	0.771 ± 0.0103	0.695 ± 0.0178
Resnet12 (aircraft)	0.576 ± 0.0116	0.647 ± 0.0127	0.667 ± 0.0112
Resnet12 (flower)	0.921 ± 0.00534	0.902 ± 0.00597	0.899 ± 0.00581
Resnet12 (dtd)	0.600 ± 0.0156	0.501 ± 0.0162	0.519 ± 0.0159
Resnet12 (delaunay)	0.734 ± 0.00984	0.655 ± 0.00981	0.665 ± 0.00986
Resnet12 (cubirds)	0.785 ± 0.00839	0.857 ± 0.00721	0.857 ± 0.00726

A.2.4 Meta-Test accuracy of a Pre-trained (PT) model vs. a MAML model on a varying size of 5CNN on the MICOD high diversity dataset

Meta-test accuracy (with 95% confidence intervals) of a Pre-trained (PT) model vs. MAML model on varying size of 5CNNs on the MICOD high diversity dataset are in table 11.

Table 11: **Meta-Test accuracy of a Pre-trained (PT) model vs. a MAML model with 95% confidence intervals on low diversity few-shot learning vision datasets.** Their diversity is high, as shown in table 8. Resnet12 has 1,427,525 parameters, while Resnet50 has 50,685,637 parameters.

Model (Seeds) (Dataset)	PT (test acc.)	MAML5 (test acc.)	MAML10 (test acc.)
Resnet12 (fo maml) (omniglot)	0.993 ± 0.00148	0.993 ± 0.00139	0.992 ± 0.00164
Resnet12 (ho maml) (omniglot)	0.994 ± 0.00110	0.985 ± 0.00219	0.988 ± 0.00180
ResNet12 (fo maml) (mio)	0.845 ± 0.0121	0.849 ± 0.0136	0.848 ± 0.0133
ResNet12 (micod)	0.778 ± 0.0124	0.781 ± 0.0124	0.786 ± 0.0119
ResNet12 (hdb6-afdo)	0.786 ± 0.0205	0.802 ± 0.0190	0.782 ± 0.0178
ResNet12 (hdb7-afto)	0.756 ± 0.0227	0.745 ± 0.0226	0.779 ± 0.0216
ResNet12 (hdb8-cado)	0.711 ± 0.0218	0.744 ± 0.0227	0.733 ± 0.0208
ResNet12 (hdb9-cavdo)	0.772 ± 0.0210	0.771 ± 0.0207	0.762 ± 0.0211
ResNet12 (hdb10-micova)	0.713 ± 0.0244	0.764 ± 0.0177	0.766 ± 0.0167
Resnet50 (seed1 vs seed1) (mds)	0.775 ± 0.0133	0.762 ± 0.0133	0.768 ± 0.0144
Resnet50 (seed1 vs seed2) (mds)	0.752 ± 0.0138	0.758 ± 0.0150	0.768 ± 0.0144
Resnet50 (seed2 vs seed1) (mds)	0.750 ± 0.0141	0.759 ± 0.0151	0.772 ± 0.0152
Resnet50 (seed2 vs seed1) (mds)	0.765 ± 0.0135	0.762 ± 0.0147	0.776 ± 0.0143

Table 12: **Meta-Test accuracy of a Pre-trained (PT) model vs. a MAML model with 95% confidence intervals on the high diversity MICOD few-shot learning vision dataset using varying size of 5CNNs.** We used a meta-batch size of 500 few-shot learning tasks. The Diversity Coefficient for the MICOD dataset is 0.174; details can be found in table 8.

Filter Size (Dataset)	PT (test acc.)	MAML5 (test acc.)	MAML10 (test acc.)
2 (micod)	0.481 ± 0.0205	0.493 ± 0.0197	0.467 ± 0.0184
6 (micod)	0.588 ± 0.0169	0.626 ± 0.0189	0.608 ± 0.0178
8 (micod)	0.606 ± 0.0161	0.591 ± 0.0178	0.607 ± 0.0184
16 (micod)	0.655 ± 0.0149	0.678 ± 0.0154	0.681 ± 0.0157
32 (micod)	0.689 ± 0.0151	0.682 ± 0.0150	0.701 ± 0.0154
64 (micod)	0.694 ± 0.0135	0.704 ± 0.0155	0.718 ± 0.0152
256 (micod)	0.711 ± 0.0139	0.702 ± 0.0163	0.695 ± 0.0156
512 (micod)	0.653 ± 0.0175	0.718 ± 0.0158	0.724 ± 0.0154

A.3 Dataset composition details

Here we detail how we made our high diversity datasets. The method we used was taking the union as in (Tian et al., 2020) of different datasets. Fe used global labels (Wang et al., 2021) during training.

Here we outline what the acronyms mean in tables 8 and 7: Here we outline what the acronyms mean in tables 8 and 7:

- HDBi stands for High-Diversity Benchmark number i.
- MIO stands for combining MiniImagenet (Vinyals et al., 2017) and Omniglot (Lake et al., 2015).
- MICOD stands for combining MiniImagenet (Vinyals et al., 2017), Cifar-fs (Bertinetto et al., 2019), Omniglot (Lake et al., 2015), and Delaunay (Gontier et al., 2022).
- AFDO stands for combining fgvcAircraft (Maji et al., 2013), vggFlower (Nilsback & Zisserman, 2006), Delaunay (Gontier et al., 2022), and Omniglot (Lake et al., 2015).
- AFTO stands for combining fgvcAircraft (Maji et al., 2013), vggFlower (Nilsback & Zisserman, 2006), describableTextures (Cimpoi et al., 2013), and Omniglot (Lake et al., 2015).
- CADO stands for combining Cifar-fs (Bertinetto et al., 2019), FGVCAircraft (Maji et al., 2013), Delaunay (Gontier et al., 2022), and Omniglot (Lake et al., 2015).
- CAVDO stands for combining Cifar-fs (Bertinetto et al., 2019), FGVCAircraft (Maji et al., 2013), VGGFlower (Nilsback & Zisserman, 2006), DescribableTextures (Cimpoi et al., 2013), Omniglot (Lake et al., 2015).

- MICOVA stands for combining MiniImagenet (Vinyals et al., 2017), Cifar-fs (Bertinetto et al., 2019), Omniglot (Lake et al., 2015), VGGFlower (Nilsback & Zisserman, 2006), FGVCAircraft (Maji et al., 2013).
- MDS stands for Meta-Dataset (Triantafillou et al., 2019).
- dtd stands for Describable Textures Dataset (Cimpoi et al., 2013).
- VGGFlower is the alternative name for fgvcFlower (Nilsback & Zisserman, 2006).
- vggair stands for combining VGGflower (Nilsback & Zisserman, 2006) and fgvcAircraft (Maji et al., 2013).
- vggdtd stands for combining VGGflower (Nilsback & Zisserman, 2006) and DTD (Cimpoi et al., 2013).

A.4 Further testing of the Diversity Coefficient

A.4.1 Validating The Task2Vec task embeddings used in the Diversity Coefficient

In this section, we further test if the Task2Vec task embeddings distances cluster in a semantically meaningful way in our dataset MIO. This test is important because if the Task2Vec embeddings used to compute the diversity coefficient have the structure we'd expect, then it makes the diversity coefficient itself more trustworthy. The MIO dataset was created by combining the MiniImagenet Omniglot. Therefore, if Task2Vec is a valid embedding for tasks, we would expect three modes for our histogram: 1. One mode for the distances between tasks generated from MiniImagenet and MiniImagnet 2. Another mode for distances between tasks generated from Omniglot and Omniglot 3. And the last mode for distances between tasks generated from MiniImagenet and Omniglot That is indeed what is seen as shown in figure 1

One interesting observation is that the average distance between Task2Vec embeddings (i.e. the diversity coefficient) is larger for smaller networks.

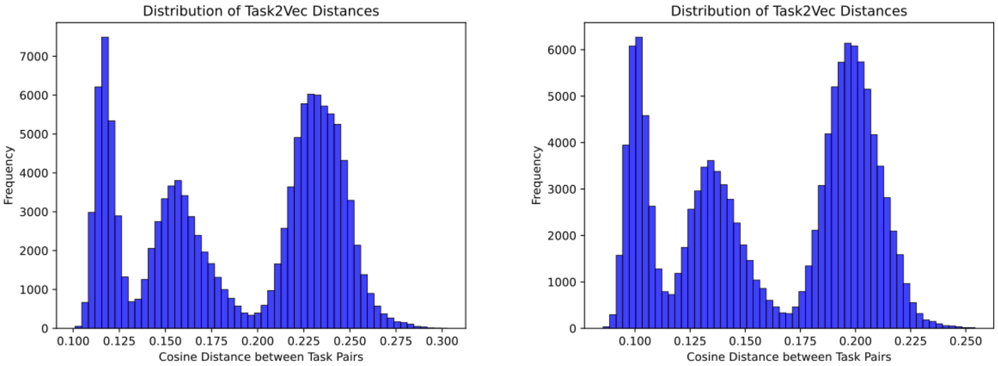

Figure 1: **The Task2Vec distances the histogram cluster in a way that reflect the semantic information of the union of the MiniImagenet and Omniglot (MIO) training datasets.** Left plot show the histogram of the cosine distance between Task2Vec embeddings made using a Resnet18 backbone pre-trained on Imagenet. Right show the same, but when using Resnet34. The meta-batch size was 500 meaning we used 500 tasks to compute these histograms. The diversity coefficients are 0.188 ± 0.00416, 0.161 ± 0.00351 for the LHS and RHS plots.

A.5 Discussion (Cont.)

We'd also like to remark the trade-offs between pre-training and meta-learning (MetaL) that Bronskill et al. (2021) articulates clearly – especially given the evidence we present countering the prevailing narrative that advocates for pre-training (PT) and transfer learning methods. The selection between PT vs MetaL strategies should be guided by: the available data, computational resources, and the application's specific requirements. For singular task types with ample data and no computational or temporal constraints, fine-tuning within transfer learning may suffice. Conversely, meta-learning would be more appropriate in scenarios requiring the acquisition of diverse tasks with sparse data on resource-limited devices, or in continual or online learning environments. In addition, we provide a novel perspective where we show formally diverse datasets are a scenario when meta-learning methods are marginally better than pre-training methods.

A potential drawback of our work could be our focus on mainly comparing pre-training (PT) against MAML, instead of considering a wider set of meta-learning algorithms. Our justification is as follows: The current narrative (Tian et al., 2020; Chen et al., 2019; 2020; Dhillon et al., 2019; Huang & Tao, 2019) implies that PT can beat *any* meta-learning algorithm. We'd like to emphasize the word *any*, because it implicates a "for all" quantifier. Therefore, to counter the current narrative, we only need to provide evidence against it (and thus show it is likely false) by considering a *single* meta-learning algorithm. Therefore, if *PT cannot even beat MAML* – the simplest of meta-learning algorithms – it's good evidence against the current narrative. Therefore, we only need a *single* meta-learning algorithm to support our conclusions. In addition, memory efficient meta-learning (Bronskill et al., 2021) demonstrated that other meta-learning algorithms can match pre-trained models. However, as we explained in the related work section, our contributions are novel, complementary and different from Bronskill et al. (2021) does because: 1. We contextualize our claims in a data-centric framework using formal diversities over an extensive set of formally diverse datasets, 2. Our analysis goes beyond using confidence intervals and reports the effect size, a method we justify in section 3, and, 3. Our novel analysis demonstrates that meta-learning (via MAML) and pre-training can be (marginally) separated in performance when considering the formal diversity of the dataset. In addition, it is reasonable to expect, given the memory efficient results (Bronskill et al., 2021), that when using their memory efficient methods that a similar trend with other meta-learning algorithms would be observed – especially given we already showed an initial separation between PT and MAML. Furthermore, our experiments were are extensive over a large set of formally diverse datasets.

Another potential drawback of our work could be the use of the arbitrary 1% thresholds for our decision rule in 3. In machine learning, it is not uncommon to accept papers due to 1% differences. We cite this ICCV 2021 paper (Li et al., 2021) which gives the performance variance of common models on meta-dataset in table 1, which commonly ranges from 0.5 to 1.0. We also cite this ECCV 2020 paper (Rodríguez et al., 2020) that provides the variance for MiniImageNet, where the standard deviations range around ≈ 0.8. However, we'd like to underscore that we **do not** rely solely on this 1% cutoff to interpret our experiments. We also report the raw effect size (and test accuracy) and use the classically accepted ranges for what is considered small effect size (Andrade, 2020). However, there is no silver bullet for statistical analysis. All of them have assumptions (e.g. CIs, effect size are best for normally distributed data), and some notion of arbitrary values (e.g. p-values, 95%-confidence intervals, effect size ranges, our 1% threshold, etc.) are always chosen to give meaning to the results. However, one can avoid confirmation bias by choosing the statistical method before the analysis of the experiments is done – which we do. In addition, our main rationale to chose effect size is that one can't manipulate (deliberately or accidentally) the sample size to have the decision rule match our preconceived assumptions – unlike the p-value in t-tests or confidence intervals where it has been an issue noted here (Lin et al., 2013) in the large sample size regime. Therefore, we attempted to protected our interpretations from confirmation bias.

Another criticism of our work could be the lack of theoretical analysis. One reason we choose not to do theoretical analysis is that it is often difficult to give non-vacuous bounds in theory. Though some progress has been done here Chen et al. (2021) but does not aim to separate pre-training methods vs meta-learning methods. However, our experiments have good theoretical motivation inspired from Wang et al. (2021) and align with conjectures we explain in detail in section A.6.2.

The challenges in vectorization of MAML and meta-learning algorithms in general stems from the arises because of the task are different across a meta-batch, so the support set has different arbitrary labels across tasks. Therefore, vectorization is not straightforward without custom CUDA implementations. However, instead of vectorizing, one could use the memory-efficient meta-learning strategy (Bronskill et al., 2021) to speed up MAML. This is an argument in favor of meta-learning given this new possible memory optimization. We leave this promising direction for future work but conjecture this will make MAML competitive against pre-training given our results and their results (Bronskill et al., 2021).

Most of our experiments are on small models but hypothesize they are all generalize. There is a debate about emergence in large language models, however, we hypothesize our results generalize to all size models. We hypothesize this because the observation of emergence is highly dependent on the metric and the sharp unpredictable jumps go away when using smooth metrics (Schaeffer et al., 2023).

An additional benefit of using the effect size is that it also protects the researchers from confirmation bias. For example, the researcher cannot deliberately choose a sample/batch size to fit pre-conceived assumptions.

A.5.1 Why and when does diversity matter?

We conjecture two main reasons why diversity matters and explain our rationale:

1. **Conjecture 1: Diversity matters because it enables learning-to-learn (proxy for General Intelligence).** This is the main conjecture we provide evidence in this paper. The main argument is that if there is high diversity, it means there are many tasks in the dataset. Therefore, for the model to do well, it has to do well on all tasks. One way to do it is by learning-to-learn and therefore transfer when challenged with solving a new task. An alternative would be memorizing all the tasks.
2. **Conjecture 2: Diversity matters because it increases changes that training set covers test set.** Diversity is a formalization of coverage – it aims to be the effective (average) number of tasks in a dataset. Therefore, the higher the diversity, the more tasks a dataset has. This (might) increase the probability that the training set covers the test set and improves performance. This exploration of this conjecture is left for future work.

A.6 Motivation

This work is inspired by three ideas/questions: 1. Does explicitly train to "learn to learn" (i.e., meta-learn) improve the performance of a machine learning algorithm? 2. What is a data-centric inductive bias that might explain when explicit meta-learning methods are needed? 3. Previous theoretical results hint that pre-training with all the data upper-bounds the meta-learning episodic loss, therefore, might this be the reason pre-training be slightly worse with imperfect settings? (e.g. imperfect optimization and limited data). We proceed to explain the latter two in more detail in this section.

A.6.1 What is the right inductive bias for the application of meta-learning?

Our work is motivated by the conjecture that the appropriate inductive bias for meta-learning is when the intrinsic diversity of a dataset is high. In other words, the distance between tasks sampled from a dataset is often high, i.e. a large variation of tasks is present. This is what the diversity coefficient is designed to measure (Miranda et al., 2022). The reasoning, behind this conjecture, is the following: 1. By definition of the problem – solving tasks from a high diversity dataset – we have tasks sampled from the dataset have large changes/distances 2. Therefore, a model that learns to adapt/learn/change might experience an advantage, because it has autonomously learns to change to these changing tasks. Consequently, if the tasks exhibit high variability/diversity, a meta-learning model may be the preferred choice.

The conjecture that meta-learning surpasses pre-training methods finds support in our empirical results 4, because the effect size is in favor of MAML on average across a wide variety of formally diverse datasets. However, intriguingly, pre-training methods outperform on low diversity datasets. This observation may clarify the prevailing narrative favoring pre-training methods. Pre-training methods might be better for lower diversity datasets. We hypothesize MAML may be ”meta-overfitting”

(Miranda et al., 2021) on such datasets, unlike pre-training methods with a fixed embedding that might have a lower (meta) variance.

Our problem/data-centric approach to meta-learning is inspired by applying Marr's level of analysis (Hamrick & Mohamed, 2020; Marr, 1982) to few-shot learning. Marr emphasized the importance of understanding the computational problem being solved and not only analyzing the algorithms or hardware that attempts to solve them. An example Marr gives is marveling at the rich structure of bird feathers without also understanding the problem they solve: flight. Similarly, there has been an analysis of MAML models and transfer learning without putting the problem such models should solve into perspective (Raghu et al., 2020; Tian et al., 2020). Therefore, in this work, we hope to clarify some of these results by partially placing the current state of affairs in meta-learning from a problem-centric view. We do this by computing the formal diversity of a dataset using the diversity coefficient.

A.6.2 Theoretical Motivation

Our work is also inspired by the theory from Wang et al. (2021), which theoretically shows the loss of pre-training on all the data upper bounds the episodic meta-learning loss.

More formally, for a fixed feature embedding model $\psi_\theta(x)$ with weights θ:

$$\mathbb{E}_{(X,Y)\in Q}\left[L_{ce}(W[Y],(\psi_\theta(X),Y))\right] \leq \mathbb{E}_{(x,y)\in D(Q)}\left[l_{ce}(W\psi_\theta(x),y)\right] \tag{4}$$

where $Q = \{(x_i, y_i)\}_{i=1}^{n_Q}$ is a standard few-shot learning query set, L_{ce} is the empirical risk of the learner over few-shot tasks using the cross-entropy loss (i.e. on the support or a query set), X is a dataset of raw input values x e.g. raw images, Y is a dataset of target labels e.g. labels for the images, $\psi_\theta(X) = \{(\psi_\theta(x) \mid x \in X\}$ the embedded few-shot task/dataset, and $D(Q)$ denotes the union of all query tasks from the source dataset e.g. union of all few-shot learning of MiniImagenet.

Equation 4 therefore implies that using a fixed embedding method, that a pre-trained model using all the tasks upper bounds the true meta-learning loss we wish to minimize for a model to "learn to learn". The main caveat is that this only applies for a fixed embedding model, so if the left-hand side uses a MAML model and the right-hand side uses a pre-trained model, then the above inequality doesn't apply. However, it provides good heuristics for our approach: 1. Get an extremely large dataset, 2. Train both models to convergence, ideally zero train-loss, 3. then compare them. The above suggests the difference should be small, which is, which is in line with our main contribution. In addition, the effect size is negative which suggests MAML is better, as one might conjecture using equation 4. In addition, as the meta-train set encompasses all possible tasks, we conjecture there is no difference between meta-leanring algorithms and pre-training trained on a union of all the data.

A.7 MAML experiments on GPT-2

Extending MAML for language modelling is a challenging task. Large language models (LLMs) including GPT-2 are typically trained in a supervised learning setting where the model is trained to predict the token coming after each token in a sentence(Radford et al., 2019). This does not naturally translate to a k-shot learning task which MAML was intended for. In particular, we note that the vocabulary size for each token is in the order of tens of thousands (50257 for GPT-2) which is a lot bigger than the few thousand (1623 for Omniglot(Lake et al., 2015)) classes MAML is typically used for. Additionally, each token does not have an equal number of instances in the language. Finally and most importantly, training a model to chose between a few classes given example occurrences of those classes, and comparing it against a model trained to predict one class out of 50257 is an apples to oranges comparison.

We however note the primary motivation of the paper to establish the performance of "learning a task" against "learning to learn the same task". Hence instead of using examples of specific target classes as the support set as is usually done for MAML, we use example sequences as the support set and example sequences as the query set. An initial idea is for each batch size of size b, we can train the model to learn from the first $b/2$ examples and predict the next $b/2$ correctly. However, since the batches are independent, we don't give the model a chance to learn from context and this implementation of MAML reduces to a harder to optimize version of the usual supervised learning setting.

We solve this issue by separating the support and query sets within examples in a batch instead of within batches. That is, if the token size of the model is t, we split each example of t tokens into support and query sets. We make the first $t/2$ tokens of each example as the support set and the next $t/2$ tokens as the query set. Hence given a token size t, we train the model to learn to predict the last $t/2$ tokens based on the first $t/2$ tokens. Formally, for each example in the batch, we perform an inner loop optimization on the cross-entropy loss of next-token predictions for the first $t/2$ tokens. We then perform the outer loop optimization on the cross-entropy loss of next-token predictions using the obtained model on the last $t/2$ tokens.

Evaluation comparison between the two training techniques is done by following a similar approach of inner loop optimization using the first half tokens and reporting accuracy values on the second half post inner loop optimization.

A.8 Fair comparison

Unlike previous, we ensure fair comparison between pre-training vs MAML by using a consistent neural network architecture, optimizer, and all models trained to convergence.

Architecture: We only compared pre-training vs MAML when they **both** had the same architecture. When we used a ResNet we used the one described in Tian et al. (2020).

Optimization: We only compared pre-training vs MAML when they **both** had the same optimization and scheduling rate. We used the Adam optimizer for all experiments except for GPT2 and Resnet50 on Meta-DataSet (MDS) where we used Adafactor with default hyperparameters. We did this because Adafactor has a setting in the Fairseq that requires no hyperparameter search and since Meta-DataSet is a large we scale dataset. It took us about 1 month to train on MDS with Resnet50. In addition, previous work demonstrated Adafactor can be fast 1 order of magnitude faster (speedup of 2 hours to 39 hours) than Adam with hyperparameter search when training transformer models (Miranda et al., 2023).

Training to Convergence: We show how our models were trained by providing some sample learning curves for pre-training and MAML in the following figures 6, 7, 4, 5, 2, 3 .

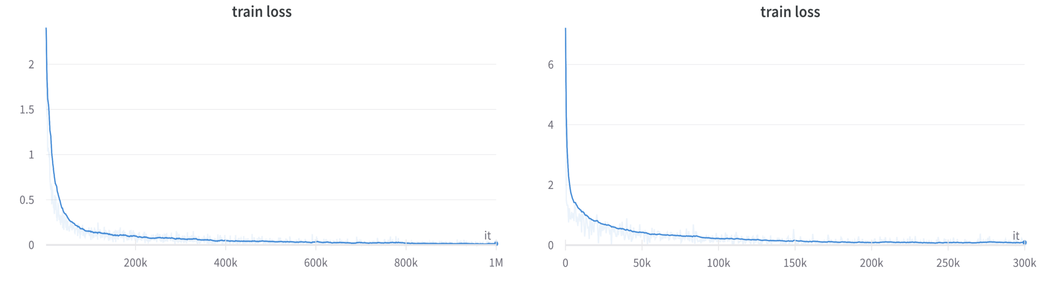

Figure 2: Plot showing convergence of Resnet12 on a high-diversity benchmark (MICOD). The left-most plot depicts the training loss curve for the pre-training algorithm, and the rightmost plot depicts the training loss curve for MAML.

A.9 Summary of CI decision results

When comparing PT (pre-training) and MAML using confidence intervals, our experiments indicate that MAML and PT tend to perform equivalently under high-diversity benchmarks, while MAML and PT perform differently (either MAML outperforming or underperforming PT) under lower-diversity benchmarks.

Figure 10 shows how average MAML(5,10) performs better than PT. This supports our main hypothesis because 1. MAML is better than PT in the high diversity regime but 2. The difference is marginal, as shown by the confidence intervals being close.

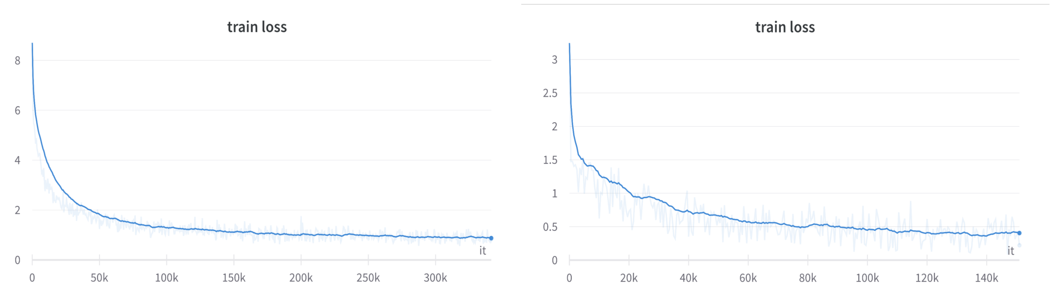

Figure 3: Plot showing convergence of Resnet50 on a high-diversity benchmark Meta-DataSet (MDS) (Triantafillou et al., 2019). The left-most plot depicts the training loss curve for the pre-training algorithm, and the rightmost plot depicts the training loss curve for MAML.

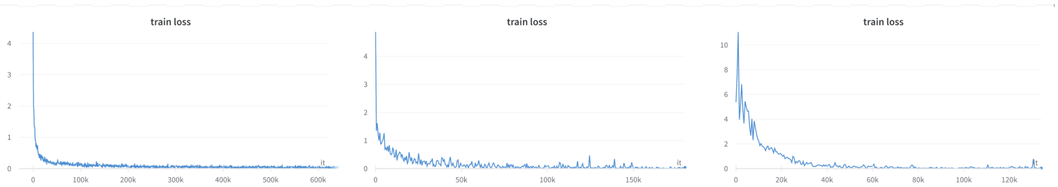

Figure 4: Plot showing convergence of Resnet12 on a low-diversity benchmark (fc100). The left-most plot depicts the training loss curve for the pre-training algorithm, the center plot depicts the training loss curve for first-order MAML, and the rightmost plot depicts the training loss curve for higher-order MAML.

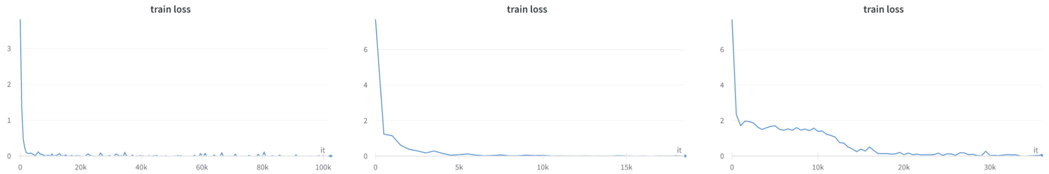

Figure 5: Plot showing convergence of Resnet12 on a low-diversity benchmark (aircraft). The left-most plot depicts the training loss curve for the pre-training algorithm, the center plot depicts the training loss curve for first-order MAML, and the rightmost plot depicts the training loss curve for higher-order MAML.

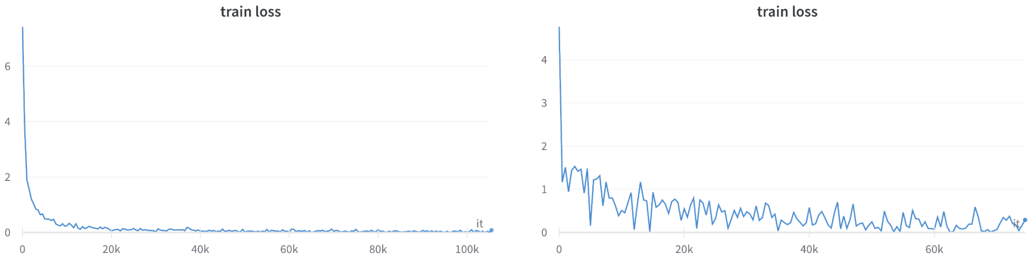

Figure 6: Plot showing convergence of Resnet12 on a high-diversity benchmark (hdb8-cado).The left plot depicts the training loss curve for the pre-training algorithm and the right plot depicts the training loss curve for MAML.

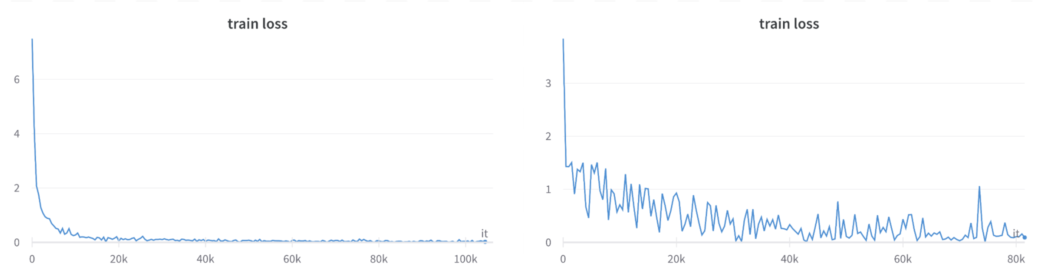

Figure 7: Plot showing convergence of Resnet12 on a high-diversity benchmark (hdb9-cavdo). The left plot depicts the training loss curve for the pre-training algorithm and the right plot depicts the training loss curve for MAML.

Table 13: **Results of performance comparison between pre-training and MAML using confidence intervals for high-diversity benchmarks.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 8 times and we rejected the null hypothesis in favor of the MAML alternative 2 times (once for MAML5 and once for MAML10). The diversity for them is high, as shown in table 8.

Dataset	pt vs maml5 CI decision	pt vs maml10 CI decision
hdb6-afdo	H0 no diff	H0 no diff
hdb7-afto	H0 no diff	H0 no diff
hdb8-cado	H0 no diff	H0 no diff
hdb9-cavdo	H0 no diff	H0 no diff
hdb10-micova	H1 maml5	H1 maml10

A.10 L2 MODEL NORMS AND VALIDATION LOSS CURVES SUGGEST THAT MAML HAS LESS META-OVERFITTING THAN PT

We demonstrate evidence that may suggest that MAML has less overfitting than PT, both via MAML and PT validation loss curves (see Figures 8 and 9), as well as the L2 model norms of trained MAML and PT models (see Tables 22, 23, 24).

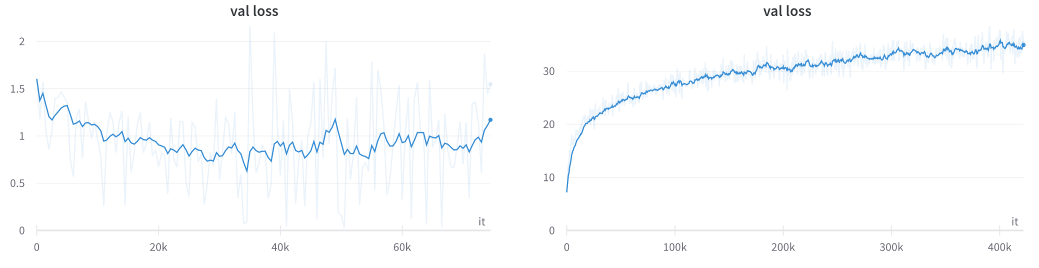

Figure 8: **On a high-diversity dataset, the validation loss of MAML stays relatively unchanged over time, while the validation loss of PT increases over time, suggesting that MAML has less meta-overfitting than PT.** The left plot depicts the validation loss curve for the MAML algorithm and the right plot depicts the validation loss curve for the PT algorithm, both on the high-diversity hdb8-cado dataset.

A.11 HYPERPARAMETERS

A.11.1 HYPERPARAMETER DETAILS

Few-shot learning details: All experiment had a 5-way, 20 shot setting – i.e, 5 train shot and 15 eval shot, for training MAML. For more details refer to section A.10.

Table 14: **Results of performance comparison between pre-training and MAML using confidence intervals for high-diversity benchmarks with a 1% overlap threshold.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 8 times and we rejected the null hypothesis in favor of the MAML alternative 2 times (once for MAML5 and once for MAML10). The diversity for them is high, as shown in table 8.

Dataset	pt vs maml5 CI decision (1% overlap)	pt vs maml10 CI decision (1% overlap)
hdb6-afdo	H0 no diff	H0 no diff
hdb7-afto	H0 no diff	H0 no diff
hdb8-cado	H0 no diff	H0 no diff
hdb9-cavdo	H0 no diff	H0 no diff
hdb10-micova	H1 maml5	H1 maml10

Table 15: **Results of performance comparison between pre-training and (fo) MAML using confidence intervals for low-diversity benchmarks.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 11 times, we rejected the null hypothesis in favor of the PT alternative 7 times, and rejected the null hypothesis in favor of the MAML alternative 6 times (MAML5 accounted for 3 of these rejections while MAML10 accounted for 3). The diversity for them is low, as shown in table 7.

Dataset	pt vs maml5 CI decision	pt vs maml10 CI decision
aircraft	H1 no diff	H1 no diff
flower	H1 pt	H1 pt
dtd	H1 pt	H1 pt
delaunay	H1 pt	H1 pt
cubirds	H1 maml5	H1 maml10
cifar-fs	H1 maml5	H1 maml10
fc100	H1 maml5	H1 maml10
mini-imagenet	H1 pt	H0 no diff
omniglot	H0 no diff	H0 no diff
tiered-imagenet	H0 no diff	H0 no diff
vggair	H0 no diff	H0 no diff
vggdtd	H0 no diff	H0 no diff

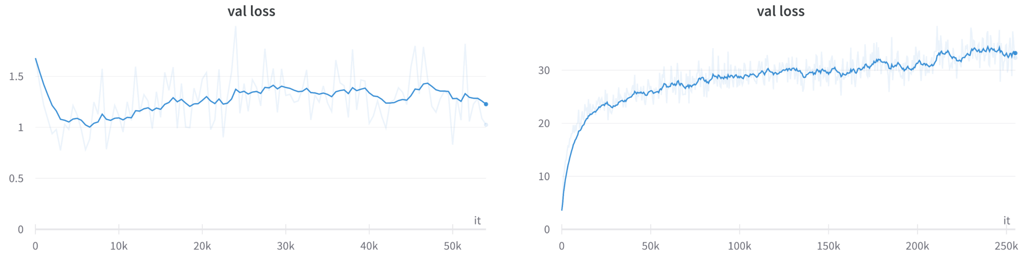

Figure 9: **On a low-diversity dataset, the validation loss of MAML stays relatively unchanged over time, while the validation loss of PT increases over time, suggesting that MAML has less meta-overfitting than PT.** The left plot depicts the validation loss curve for the MAML algorithm and the right plot depicts the validation loss curve for the PT algorithm, both on the low-diversity DTD dataset.

Hyperparameter Details for Resnet12 on low diversity data sets for Pre-training and fo/ho-MAML: We used the Resnet12 architecture provided by Tian et al. (2020). The Adam optimizer (Kingma & Ba, 2017) was utilized with a constant learning rate of 1e-3. No learning rate scheduler was used. Training was performed for 600,000 iterations for pre-training and 160,000 first-order MAML iterations, with a batch size of 256. The outer loop consisted of 130,000 MAML iterations.

Table 16: **Results of performance comparison between pre-training and (fo) MAML using confidence intervals for low-diversity benchmarks with a 1% overlap threshold.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 14 times, we rejected the null hypothesis in favor of the PT alternative 7 times, and rejected the null hypothesis in favor of the MAML alternative 3 times (MAML5 accounted for 1 of these rejections while MAML10 accounted for 2). The diversity for them is low, as shown in table 7.

Dataset	pt vs maml5 decision (1% overlap)	pt vs maml10 decision (1% overlap)
aircraft	H0 no diff	H0 no diff
flower	H1 pt	H1 pt
dtd	H1 pt	H1 pt
delaunay	H1 pt	H1 pt
cubirds	H1 maml5	H1 maml10
cifar-fs	H0 no diff	H1 maml10
fc100	H0 no diff	H0 no diff
mini-imagenet	H1 pt	H0 no diff
omniglot	H0 no diff	H0 no diff
tiered-imagenet	H0 no diff	H0 no diff
vggair	H0 no diff	H0 no diff
vggdtd	H0 no diff	H0 no diff

Table 17: **Results of performance comparison between pre-training and (ho) MAML using confidence intervals for low-diversity benchmarks.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 2 times, we rejected the null hypothesis in favor of the PT alternative 10 times, and rejected the null hypothesis in favor of the MAML alternative 8 times (MAML5 accounted for 4 of these rejections while MAML10 accounted for 4). The diversity for them is low, as shown in table 7.

Dataset	pt vs maml5 CI decision	pt vs maml10 CI decision
aircraft	H1 maml5	H1 maml10
flower	H1 pt	H1 pt
dtd	H1 pt	H1 pt
delaunay	H1 pt	H1 pt
cubirds	H1 maml5	H1 maml10
cifar-fs	H1 maml5	H1 maml10
fc100	H1 maml5	H1 maml10
mini-imagenet	H0 no diff	H0 no diff
omniglot	H1 pt	H1 pt
tiered-imagenet	H1 pt	H1 pt

We used an inner learning rate of 0.1 and 5 inner steps. No weight decay was applied. Training was performed on a single NVIDIA PU with at most 48GB memory select by a HPC automatically. All experiments were trained to convergence (less than 0.01 loss) and took on average at most 1 week. All implementations were done in PyTorch (Paszke et al., 2019).

Hyperparameter Details for 5CNN on High Diversity Datasets: We utilized the 5CNN architecture proposed in Tian et al. (2020) with varying filter sizes. The Adam optimizer Kingma & Ba (2017) was used with a learning rate of 1e-3 without any learning rate decay. A batch size of 256 was used for both pre-training and MAML training. No weight decay was applied. For pre-training, we trained for 200,000 iterations. For first-order MAML, we trained for 100,000 iterations with an inner loop of 5 steps and an inner learning rate of 0.1. We annealed the learning rate with a cosine scheduler with scheduler freq 2000 with minimum learning rate 1e-5 (similar to MAML++). All models were trained to convergence, which took approximately 1 week on a single NVIDIA GPU with at least 48GB of memory allocated by the HPC scheduler. All implementations were done in PyTorch Paszke et al. (2019).

Table 18: **Results of performance comparison between pre-training and (ho) MAML using confidence intervals for low-diversity benchmarks with a 1% overlap threshold.** These performance comparison experiments were conducted using a batch size of 300. The summary for the decision counts is as follows: we failed to reject the null hypothesis H0 (no difference) 6 times, we rejected the null hypothesis in favor of the PT alternative 6 times, and rejected the null hypothesis in favor of the MAML alternative 8 times (MAML5 accounted for 4 of these rejections while MAML10 accounted for 4). The diversity for them is low, as shown in table 7.

Dataset	pt vs maml5 CI decision (1% overlap)	pt vs maml10 CI decision (1% overlap)
aircraft	H1 maml5	H1 maml10
flower	H0 no diff	H1 pt
dtd	H1 pt	H1 pt
delaunay	H1 pt	H1 pt
cubirds	H1 maml5	H1 maml10
cifar-fs	H1 maml5	H1 maml10
fc100	H1 maml5	H1 maml10
mini-imagenet	H0 no diff	H0 no diff
omniglot	H0 no diff	H0 no diff
tiered-imagenet	H0 no diff	H1 pt

Table 19: 1% effect sizes for performance comparison between pre-training and (fo) MAML for low-diversity benchmarks.

Dataset	pt vs maml5 1% ES	pt vs maml10 1% ES
aircraft	0.100	0.109
flower	0.171	0.195
dtd	0.122	0.125
delaunay	0.106	0.122
cubirds	0.135	0.133
cifar-fs	0.114	0.113
fc100	0.121	0.117
mini-imagenet	0.123	0.117
omniglot	0.789	0.727
tiered-imagenet	0.112	0.113
vggair	0.059	0.059
vggdtd	0.056	0.056

Hyperparameter Details for ResNet12 on High Diversity Benchmarks: We utilized the ResNet12 architecture from Tian et al. (2020) for our experiments. The Adam optimizer (Kingma & Ba, 2017) was used with a learning rate of 1e-3 without any learning rate decay. For pre-training, we trained for 1 million iterations with a batch size of 256. For first-order MAML (Finn et al., 2017), we trained for 300,000 iterations also with a batch size of 256. The MAML outer loop consisted of 5 inner update steps with an inner learning rate of 0.1. No weight decay was applied. We annealed the learning rate with a cosine scheduler with scheduler freq 2000 with minimum learning rate 1e-5 (similar to MAML++). All models were trained to convergence on a single NVIDIA GPU with at least 48GB of memory allocated by the cluster scheduler. Training took approximately 1-2 week to converge for both pre-training and MAML. All implementations were done in PyTorch (Paszke et al., 2019).

Hyperparameter Details for ResNet50 on High Diversity Meta-Dataset Benchmarks: We utilized the ResNet50 architecture from Tian et al. (2020) in our experiments on the high diversity meta-dataset benchmarks. The Adafactor optimizer (Shazeer & Stern, 2018) was used with default settings and no learning rate decay. We used Adafactor from Seqfair because it had a setting with no hyperparameter choices, the memory benefits that we needed given our compute and the evidence of previous work showing the training was 2.5-fold faster (Miranda et al., 2023). For pre-training, we trained for 300,000 iterations with a batch size of 256. For first-order MAML (Finn et al., 2017), we trained for 140,000 iterations also with a batch size of 256. The MAML outer loop consisted of 5 inner update steps with Adafactor's default inner learning rate. We used Adafactor default annealing scheduler in Seqfair. Due to computational constraints, we limited the number of random seeds to 4 – especially

Table 20: 1% effect sizes for performance comparison between pre-training and (ho) MAML for low-diversity benchmarks.

Dataset	pt vs maml5 1% ES	pt vs maml10 1% ES
aircraft	0.094	0.100
flower	0.201	0.204
dtd	0.125	0.126
delaunay	0.116	0.115
cubirds	0.145	0.145
fc100	0.113	0.113
cifar-fs	0.118	0.113
mini-imagenet	0.120	0.123
omniglot	0.655	0.762
tiered-imagenet	0.116	0.080

Table 21: 1% effect sizes for performance comparison between pre-training and (fo) MAML for high-diversity benchmarks.

Dataset	pt vs maml5 1% ES	pt vs maml10 1% ES
hdb6-afdo	0.057	0.059
hdb7-afto	0.050	0.051
hdb8-cado	0.051	0.053
hdb9-cavdo	0.054	0.054
hdb10-micova	0.056	0.057

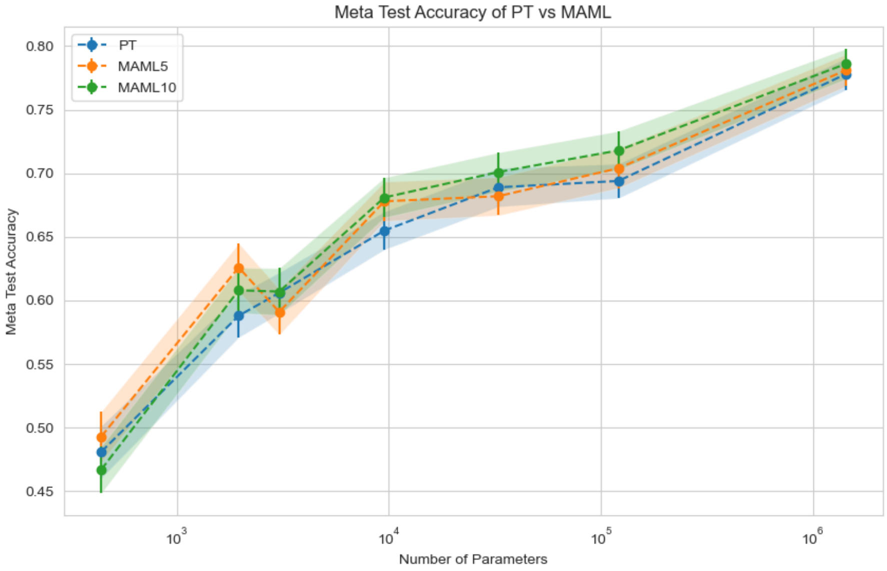

Figure 10: **Shows how meta-test accuracy between PT and MAML(5,10) intersects in the high diversity data set MICOD.** However, on average MAML(5,10) performs better than PT. This supports our main hypothesis because: 1. MAML is better than PT in the high diversity regime but 2. The difference is marginal, as shown by the confidence intervals being close.

given that MDS combined 10 large scale vision data sets that includes ImageNet. Pre-training and MAML training took approximately 1 month each to converge on NVIDIA GPUs with 48GB memory allocated automatically by the cluster scheduler. All implementations were done in PyTorch (Paszke et al., 2019).

Table 22: **The L2 norm of a trained first-order MAML model is less than the L2 norm of a trained PT model for each low-diversity benchmark, suggesting that MAML has less meta-overfitting than PT.**

Dataset	L2 model norm (MAML)	L2 model norm (PT)
cifar-fs	851.012	9813.269
fc100	919.964	8302.170
omniglot	663.715	4676.182
mini-imagenet	930.304	5776.625
tiered-imagenet	926.896	11097.097

Table 23: **The L2 norm of a trained higher-order MAML model is less than the L2 norm of a trained PT model for each low-diversity benchmark, suggesting that MAML has less meta-overfitting than PT.**

Dataset	L2 model norm (MAML)	L2 model norm (PT)
dtd	2949.312	5548.194
tiered-imagenet	731.893	11097.097
omniglot	594.381	4676.182
fc100	758.372	8302.170
delaunay	2810.343	4856.295
aircraft	1517.484	6144.078
cifar-fs	725.017	9813.269
mini-imagenet	752.201	5776.625
cubirds	3127.715	5252.014
flower	2333.556	7307.350

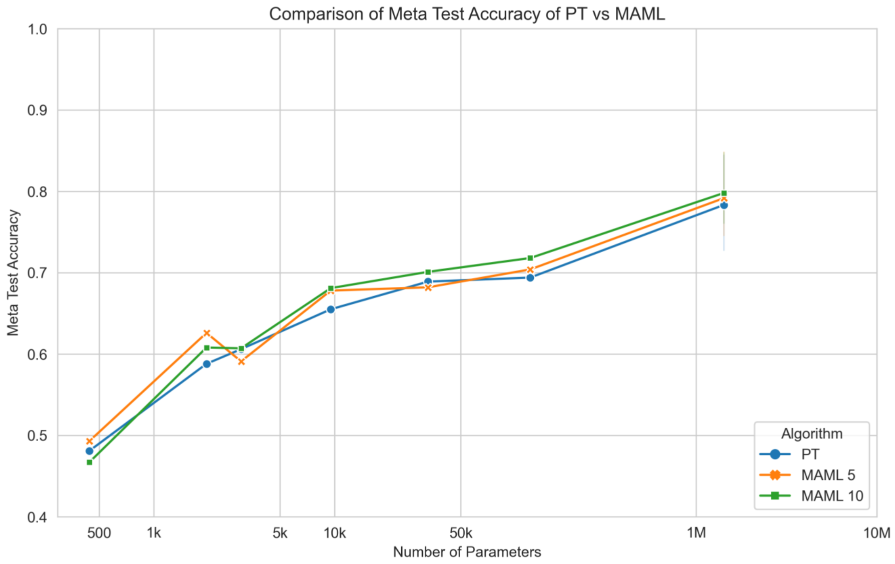

Figure 11: **MAML(5,10) outperforms PT in the high diversity data set MICOD.** Same as Figure 10 but without confidence intervals. This supports our main hypothesis as it demonstrates that MAML is marginally better than PT in the high diversity regime.

Hyperparameter Details for mini-Gpt2 on openwebtext: We derived our architecture from nanoGPT[1]. We set a block size of 32 and create a model with 4 layers and 4 heads per layer with an

[1] https://github.com/karpathy/nanoGPT

Table 24: **The L2 norm of a trained higher-order MAML model is less than the L2 norm of a trained PT model for each high-diversity benchmark, suggesting that MAML has less meta-overfitting than PT.**

Dataset	L2 model norm (MAML)	L2 model norm (PT)
hdb6-afdo	2426.827	7699.514
hdb7-afto	2598.023	3573.5355
hdb8-afdo	3441.040	8072.174
hdb9-cavdo	3997.130	7919.635
hdb10-micova	3810.151	7757.541

embedding size of 192 to have a smaller model that is easy for experimentation that we dub mini-gpt2. The Adafactor optimizer and scheduler (Shazeer & Stern, 2018) were used with default settings (that is, no hyperparameters were specified) and no learning rate decay. In addition due to the memory benefits that we needed given our compute and the evidence of previous work showing the training was 2.5-fold faster (Miranda et al., 2023). Training was performed till visual convergence of loss curves was reached for all experiments. For MAML training, we set inner loop learning rate as 1e-3 and performed 5 inner update steps. Training was performed in a distributed setting on 4 NVIDIA PU with at most 48GB memory select by a HPC automatically. All implementations were done in PyTorch (Paszke et al., 2019).