Two Sides of Meta-Learning Evaluation: In vs. Out of Distribution

Anonymous Author(s)
Affiliation
Address
email

Abstract

We categorize meta-learning evaluation into two settings: *in-distribution* [ID], in which the train and test tasks are sampled *iid* from the same underlying task distribution, and *out-of-distribution* [OOD], in which they are not. While most meta-learning theory and some FSL applications follow the ID setting, we identify that most existing few-shot classification benchmarks instead reflect OOD evaluation, as they use disjoint sets of train (base) and test (novel) classes for task generation. This discrepancy is problematic because—as we show on numerous benchmarks—meta-learning methods that perform better on existing OOD datasets may perform significantly worse in the ID setting. In addition, in the OOD setting, even though current FSL benchmarks seem befitting, our study highlights concerns in 1) reliably performing model selection for a given meta-learning method, and 2) consistently comparing the performance of different methods. To address these concerns, we provide suggestions on how to construct FSL benchmarks to allow for ID evaluation as well as more reliable OOD evaluation. Our work aims to inform the meta-learning community about the importance and distinction of ID vs. OOD evaluation, as well as the subtleties of OOD evaluation with current benchmarks.

1 Introduction

Meta-learning considers learning algorithms that can perform well over a distribution of tasks [17, 35]. To do so, a meta-learning method first learns from a set of tasks sampled from a training task distribution (*meta-training*), and then evaluates the quality of the learned algorithm using tasks from a test task distribution (*meta-testing*). The test task distribution can be the same as the training task distribution (a scenario we term *in-distribution* generalization evaluation or ID evaluation) or a different task distribution (*out-of-distribution* generalization evaluation or OOD evaluation).

In this work, we argue that there is a need to carefully consider current meta-learning practices in light of this ID vs. OOD categorization. In particular, meta-learning is commonly evaluated on few-shot learning (FSL) benchmarks, which aim to evaluate meta-learning methods' ability to learn sample-efficient algorithms. Current benchmarks primarily focus on image classification and provide training tasks constructed from a set of train (base) classes that are completely disjoint and sometimes extremely different from the test (novel) classes used for test tasks. As we discuss in Section 3, this design choice imposes a natural shift in the train and test task distribution that makes current benchmarks reflective of OOD generalization. However, there are a number of reasons to also consider the distinct setting of ID evaluation. First, whether in terms of methodology or theory, many works motivate and analyze meta-learning under the assumption that train and test tasks are sampled *iid* from the same distribution (see Section 2). Second, we identify a growing number of applications, such as federated learning, where there is in fact a need for sample-efficient algorithms that can perform ID generalization. Crucially, we show across numerous benchmarks that methods that perform well OOD may perform significantly worse in ID settings. Our results highlight that it is

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

critical to clearly define which setting a researcher is targeting when developing new meta-learning methods, and we provide tools for modifying existing benchmarks to reflect both scenarios.

Beyond this, we also re-examine current OOD FSL benchmarks and analyze how the shift in the train and test task distributions may impact the reliability of OOD evaluations. We point out two concerns which we believe are not widely considered by the meta-learning community. First, unlike areas such as domain generalization where model selection challenges are more widely discussed [16, 21], we conduct to the best of our knowledge the first rigorous study demonstrating the difficulty of model selection due to the shift in the validation and test task distributions in FSL benchmarks. Second, because the OOD scenario in meta-learning does not assume a specific test task distribution, there is room for different test distributions to be used for evaluation. We show that comparing which meta-learning method performs better can be unreliable not only over different OOD FSL benchmarks, but also within a single benchmark depending on the number of novel classes.

Our main contributions are: **i)** We clearly outline both ID and OOD FSL evaluation scenarios and explain why most popular FSL benchmarks target OOD evaluation (Section 3). **ii)** We provide realistic examples of the ID scenario and show that the performance of popular meta-learning methods can drastically differ in ID vs. OOD scenarios (Section 4). **iii)** For existing OOD FSL benchmarks, we highlight concerns with a) current model selection strategies for meta-learning methods, and b) the reliability of meta-learning method comparisons (Section 5). **iv)** To remedy these concerns, we suggest suitable modifications to the current FSL benchmarks to allow for ID evaluation, and explain how to construct FSL benchmarks to provide more reliable OOD evaluation. Our hope in highlighting these evaluation concerns is for future researchers to take these issues into consideration when evaluating newly proposed meta-learning methods or designing new FSL benchmarks.

2 Related Work

Current FSL benchmarks. A plethora of few-shot image classification benchmarks (*e.g., mini*-ImageNet [39], CIFAR-FS [3]) have been developed for FSL evaluation. These benchmarks typically provide three disjoint sets of classes (base, validation, novel) taken from standard classification datasets, *e.g.,* ImageNet or CIFAR-100 [34, 30, 39]. Training, val, and test tasks are then constructed from the classes, which, as we discuss in Section 3, can induce a shift in their corresponding task distributions. Distribution mismatch can be particularly large with non-random splits created at the super class level, *e.g.,* FC-100 [28], or dataset level, *e.g.,* Meta-Dataset [38]. Recently, Ren et al. [33] proposed new benchmarks in a setting which they term flexible few-shot learning (FFSL), where the aim is to classify examples into a context instead of an object class. During testing, they perform OOD evaluation on unseen contexts. Inspired by their approach, we also provide an FSL benchmark where tasks are specified by contexts (Section 4), though we differ by exploring ID evaluation.

Mismatch between meta-learning theory/methodology and evaluation. Despite theoretical works which analyze meta-learning OOD generalization [10, 12], there are many meta-learning theory papers [e.g., 1, 2, 7, 20] that first assume the train and test tasks are *iid* sampled from the same distribution despite validating their analyses on OOD FSL benchmarks. Additionally, several popular meta-learning methods [23, 14, 29] that are motivated in the ID scenario are largely evaluated on OOD benchmarks (see Appendix A). Prior work of Lee et al. [22] explores ID vs. OOD, however they treat the FSL setup as ID when the disjoint base and novel classes are from the same dataset and OOD only when they are not (*e.g.,* base from ImageNet, novel from CUB). We emphasize that even the disjointedness of base and novel from the same dataset can create a task distribution shift and hence unlike [22] we do not consider current FSL benchmarks (like *mini*) to be ID (Section 3).

FSL ID evaluation. We are unaware of any FSL classification benchmarks that are explicitly advertised for ID evaluation. As we discuss in Section 4, a growing number of works [6, 11, 18, 20, 25] use meta-learning for personalized federated learning, but do not clearly discuss the in-distribution nature of these benchmarks or how they differ from standard FSL benchmarks. In recent work, Chen et al. [9] extend current FSL benchmarks to evaluate their proposed method both ID and OOD, but only to further improve OOD performance on the original FSL benchmark's novel classes. Our work uses a similar setup for constructing ID evaluations with current OOD FSL benchmarks, but focuses on showing that certain meta-learning methods/design choices can improve OOD performance at the cost of ID performance. Prior work [15, 32] on incremental few-shot / low-shot learning also explores performance on both base and novel classes simultaneously. However, they differ in their methodology, as they use supervised learning (not meta-learning) to classify over the base classes.

OOD evaluation in other fields. Train and test distribution shifts are also found in domain generalization [4, 27] where the goal is to find a model that works well for a differing test environment. Due to this shift, Gulrajani and Lopez-Paz [16] specifically discuss difficulties in performing model selection in domain generalization benchmarks. They argue that it is the responsibility of the method designer (not benchmark designer) to determine model selection strategies for their method, and propose several model selection methods, mainly targeting hyperparameter selection. Unlike domain generalization, FSL benchmarks often have a pre-determined disjoint validation set, and so the need for developing selection methods may be less obvious. In Section 5, motivated by [16], we explore strategies for model selection for meta-learning. However, in contrast to [16], we focus on algorithm snapshot selection for meta-learning, which is required by hyperparameter selection as a subroutine.

3 Current FSL benchmarks: Background & Focus on OOD evaluation

Background and notation. In this work, we employ a general definition of a meta-learning FSL task: a task $\mathcal{T}$ is a distribution over the space of support and query dataset pairs (S, Q), where S, Q are two sets of examples from an example space $\mathcal{X} \times \mathcal{Y}$. The support set S is used by an algorithm to produce an adapted model which is then evaluated by the corresponding query set Q. Each time we interact with a task $\mathcal{T}$, an (S, Q) pair is sampled *iid* from $\mathcal{T}$, the mechanism for which depends on the specific application; we provide multiple examples below. As discussed in Section 1, meta-learning aims to learn an algorithm over a distribution of tasks $\mathbb{P}(\mathcal{T})$. During meta-training, we assume access to N $\{(S_i, Q_i)\}_{i\in[N]}$ pairs sampled from a *training task distribution*[1] $\mathbb{P}_{\text{tr}}(\mathcal{T})$ in the following way: First N tasks are sampled *iid* from the training task $\mathcal{T}_i \sim \mathbb{P}_{\text{tr}}(\mathcal{T})$ and then for each task $\mathcal{T}_i$, a support query pair $(S_i, Q_i) \sim \mathcal{T}_i$ is sampled *iid*. During meta-testing, we assume there is a *test task distribution* $\mathbb{P}_{\text{te}}(\mathcal{T})$ where fresh (S, Q) samples are similarly sampled based on $P_{\text{te}}(\mathcal{T})$. We define a learning scenario to be in-distribution (ID) if $\mathbb{P}_{\text{tr}} = \mathbb{P}_{\text{te}}$ and out-of-distribution (OOD) if $\mathbb{P}_{\text{tr}} \neq \mathbb{P}_{\text{te}}$. Note that whenever $\mathbb{P}_{\text{tr}} = \mathbb{P}_{\text{te}}$, the induced train and test marginal distributions of (S, Q) are also identical.

Construction of (S, Q) pairs in FSL. Most popular FSL benchmarks share a similar structure: they provide three disjoint sets of classes base classes $\mathcal{C}_B$, validation classes $\mathcal{C}_V$, and novel classes $\mathcal{C}_N$, where any class c in these sets specifies a distribution $\mathbb{P}_c$ over the example space $\mathcal{X} \times \mathcal{Y}$. An n-way k-shot q-query task $\mathcal{T}_{\boldsymbol{c}}$ in these benchmarks is specified by a length n non-repeated class tuple $\boldsymbol{c} = (c_1, \ldots, c_n)$ where $\boldsymbol{c} \in [\mathcal{C}^n] := \{(d_1, \ldots, d_n) \in \mathcal{C}^n : d_i \neq d_j, \forall i \neq j\}$. $\mathcal{T}_{\boldsymbol{c}}$ generates random (S, Q) pairs in the following way: For every class c_i, k support examples $S_i \sim (\mathbb{P}_{c_i})^k$ and q query examples $Q_i \sim (\mathbb{P}_{c_i})^q$ are sampled. The support and query set is formed by the union of such examples from all classes: $S = \cup S_i$, $Q = \cup Q_i$. By specifying the base and novel classes, the FSL benchmark has provided a collection of tasks for training $\{\mathcal{T}_{\boldsymbol{c}} : \boldsymbol{c} \in [\mathcal{C}_B^n]\}$ and test $\{\mathcal{T}_{\boldsymbol{c}} : \boldsymbol{c} \in [\mathcal{C}_N^n]\}$. These sets can be extremely large. For example, in *mini*, which has 64 base classes, the total number of 5-way training tasks is $\binom{64}{5} \approx 7.6 \times 10^6$. However, it is not explicitly specified what underlying task distribution $\mathbb{P}(\mathcal{T})$ generates the sets of tasks. We believe this may have lead prior work to discuss FSL benchmarks in the context of ID evaluation [e.g., 1, 2, 7], contrary to what we outline below.

Current FSL benchmarks target OOD evaluation. We now informally discuss our proof-by-contradiction reasoning for viewing current FSL benchmarks as reflecting OOD evaluation (formal result in Appendix B). If the training and test tasks in FSL benchmarks are indeed *iid* sampled from the same underlying task distribution, then this underlying distribution must be induced by a distribution over class tuples of an even larger class set $\mathcal{C}_L$ ($\mathcal{C}_L \supseteq (\mathcal{C}_B \cup \mathcal{C}_V \cup \mathcal{C}_N)$). Since sufficient number of task samples are necessary for learning, we assume that the number of training tasks is large enough and at least scales with $N = \Omega(\sqrt{|\mathcal{C}_L|})$. Under this assumption, we consider a dichotomy: **i)** when $|\mathcal{C}_L| \leq cN$, then with high probability both the training and test tasks would each cover a significant portion of all the classes in $\mathcal{C}_L$, making it extremely unlikely to have an empty intersection as in the current OOD benchmark. **ii)** when $c_1 N \leq |\mathcal{C}_L| \leq c_2 N^2$, then the number of classes covered by the training tasks alone would scale like $\sqrt{N}$. Because of the large number of training tasks N used in meta-training ($> 10^5$), it is highly impossible that all the training tasks would only cover a very low number of classes (64 in the case of *mini*).

Randomized class partitions do not imply randomized task partitions. Another issue that may cause some to view FSL benchmarks as ID that in certain benchmarks the base, val, novel classes are random partitions of *iid* drawn classes from a class level distribution (specifically *mini*ImageNet,

[1] We note that 'training task distribution' here refers to the true underlying distribution of training tasks, not the empirical distribution supported over the finite set of sampled training tasks.

CIFAR-FS, and not FC-100, *tiered*ImageNet). The logic here is that in standard machine learning practice, randomly partitioning *iid* sampled data points into train and test guarantees that the train and test samples are drawn *iid* from the same underlying distribution. However, it is important to notice that the first class citizen in common FSL benchmarks is not a class, but a task (represented by a class tuple). So, only a randomized partition of *iid* sampled class tuples would guarantee in-distribution.

How can we view $\mathbb{P}_{\text{tr}}, \mathbb{P}_{\text{te}}$ in common FSL benchmarks? Based on the discussion above, we view train and test tasks in current FSL benchmarks as coming from different task distributions, *i.e.,* $\mathbb{P}_{\text{tr}} \neq \mathbb{P}_{\text{te}}$. In order to ensure that both sets of tasks are still sampled *iid* from their respective distributions, it is convenient to view the train/test tasks as being *iid* sampled from a uniform distribution over all possible train/test class tuples induced by $\mathcal{C}_B/\mathcal{C}_N$ *i.e.,* $\mathbb{P}_{\text{tr}} = \mathbb{P}_{\mathcal{C}_B} := \text{Unif}(\{\mathcal{T}_{\boldsymbol{c}} : \boldsymbol{c} \in [\mathcal{C}_B^n]\})$ and test $\mathbb{P}_{\text{te}} = \mathbb{P}_{\mathcal{C}_N} := \text{Unif}(\{\mathcal{T}_{\boldsymbol{c}} : \boldsymbol{c} \in [\mathcal{C}_N^n]\})$—a view we will adopt in the rest of the paper.

4 Evaluating In-Distribution Performance

Although (as discussed in Section 3) current FSL benchmarks target OOD evaluation, we now explore example applications where ID generalization is instead required, and provide easily constructible benchmarks mirroring these scenarios. As we will show, this distinction is important because meta-learning methods may perform markedly different in ID vs OOD scenarios.

Example 1 **(Federated Learning):** Multiple works [6, 11, 18, 20, 25] have considered applying meta-learning methods in federated learning, in which the goal is to learn across a distributed network of devices [26, 24]. Meta-learning can produce personalized models for unseen devices/users, improving over a single globally-learned model's performance. In this setting, a popular benchmark is the FEMNIST [5] handwriting recognition dataset. For FEMNIST, we assume there exists a distribution of writers $\mathbb{P}(\texttt{id})$ in a federated network where each writer (with a unique `id`) is associated with a few-shot classification problem to recognize over the different character classes the writer has written for. We associate each writer `id` with a task $\mathcal{T}_{\texttt{id}}$ which randomly generates a support set with one random example per class and a query set with varying number of random examples per class.

ID evaluation on FEMNIST. We are given a total of ~ 3500 writers sampled *iid* from $\mathbb{P}(\texttt{id})$ and we randomly partition them into a $2509/538/538$ split for training, validation, and test tasks. Note that this random split is performed at the task/`id` level. By doing this, we can treat the training and test tasks as sampled *iid* from the same task distribution, unlike current FSL benchmarks.

Example 2 **(Online Recommendation):** Ren et al. [33] propose the use of *Zappos* [40] dataset as a meta-learning benchmark where each task is a binary classification of shoe images into an attribute context. This mimics an online shopping recommendation problem, where each user has different shoe preferences based on specific shoe attributes (hence a single global predictive model would not perform well), and the recommendation system must quickly learn a user's likes/dislikes through a few interactions. In this simplified setup, we fix a universal set of shoe attributes $\mathcal{A}$, and each user's preference is represented by a specific pair of unique attributes $\boldsymbol{a} = (a_1, a_2)$, $a_1 \neq a_2$. A task $\mathcal{T}_{\boldsymbol{a}}$ representing a user with attribute preference $\boldsymbol{a}$ generates 2-way k-shot q-query (S, Q) pair by *iid* sampling $k + q$ examples from both the set of images that carry both attributes in $\boldsymbol{a}$ (positive examples) and the set that does not (negative examples). Our task distribution is a uniform distribution over tasks of all attribute pairs $\mathbb{P}_{\mathcal{A}}(\mathcal{T}) = \text{Unif}(\{\mathcal{T}_{\boldsymbol{a}} : \boldsymbol{a} \in [\mathcal{A}^2]\})$.

ID evaluation on Zappos. Unlike in [33], where Zappos is used to measure OOD performance by having disjoint train and test attributes $\mathcal{A}_{\text{tr}} \cap \mathcal{A}_{\text{te}} = \phi$, in this work we use Zappos for ID evaluations by *iid* sampling both meta-train and meta-test tasks from the same distribution $\mathbb{P}_{\mathcal{A}}(\mathcal{T})$. Through this modification of Ren et al.'s setup, we sample 1000 attribute pairs from an attribute set $|\mathcal{A}| = 36$ to construct 1000 training tasks (each with a randomly sampled (S, Q)) and evaluate ID performance on another 25000 test tasks sampled in the same way. We believe that our evaluation setup captures a realistic setting where the goal is to have a learned algorithm to generalize to the entire online shopper population despite being trained only with data from a randomly chosen subset of shoppers.

Remark. Even with ID evaluation it is possible to encounter unseen classes/attributes in meta-test tasks, specifically when the number of meta-train tasks is smaller than the number of underlying classes/attributes (Section 3). However, a suitable number of *iid* sampled meta-train tasks is needed to ensure good performance, which would naturally encompass a larger set of meta-train classes than those considered by OOD FSL benchmarks. For example, in Zappos-ID, there are 16 attribute pairs in the meta-test tasks that have never been seen during meta-training, but it is still in-distribution since the sampling distributions of train and test attribute pairs (and thus of tasks) are identical.

ID benchmark results. We evaluate the ID performance of four popular meta-learning methods: Protoypical Networks (PN) [37], MetaOptNet-SVM (SVM) [23], MetaOptNet-Ridge Regression (RR) [23, 3] and FOMAML [13] on our identified ID FSL benchmarks (Table 1). Since FEMNIST's tasks have varying number of ways, FOMAML cannot be directly used due to the logit layer shape mismatch. We note that the performance order of the four methods are consistent on all three ID benchmarks yet surprisingly **almost completely opposite** to the performance order observed on the OOD benchmark *mini* (except for FOMAML). Thus without modifications these common OOD FSL benchmarks (like *mini*) cannot be used to analyze ID performance, giving further evidence to the danger of such practices (see Section 2). To further understand this phenomenon, we propose a way to also enable ID performance evaluation over these common OOD FSL benchmarks and see if there still exists a difference in ID, OOD performance orders.

Table 1: Ranking in () of meta-algorithms' test performance on **i)** ID benchmarks FEMNIST, Zappos-ID; and **ii)** OOD FSL benchmark *mini*ImageNet (*mini*).

Dataset /	FEMNIST	Zappos-ID		*mini*
Method	*w1s	2w10s	2w5s	5w5s
PN	$^{(1)}94.72 \pm 0.41\%$	$^{(1)}88.40 \pm 0.13\%$	$^{(1)}86.58 \pm 0.15\%$	$^{(3)}76.22 \pm 0.14\%$
Ridge	$^{(1)}94.71 \pm 0.42\%$	$^{(2)}88.01 \pm 0.14\%$	$^{(2)}85.56 \pm 0.16\%$	$^{(2)}77.20 \pm 0.15\%$
SVM	$^{(3)}94.22 \pm 0.45\%$	$^{(3)}87.75 \pm 0.14\%$	$^{(3)}85.12 \pm 0.16\%$	$^{(1)}77.72 \pm 0.15\%$
FO-MAML	N/A	$^{(4)}81.90 \pm 0.14\%$	$^{(4)}80.14 \pm 0.15\%$	$^{(4)}75.96 \pm 0.17\%$

Modifying OOD FSL benchmarks for ID evaluation. From our previous discussion, we have shown that we can think of the FSL training tasks as sampled *iid* from the task distribution $\mathbb{P}_{\mathcal{C}_B} := \text{Unif}(\{\mathcal{T}_{\boldsymbol{c}} : \boldsymbol{c} \in [\mathcal{C}_B^n]\})$. To conduct an in-distribution evaluation, we need to sample fresh test tasks *iid* from $\mathbb{P}_{\mathcal{C}_B}$. For a freshly sampled $\mathcal{T}_{\boldsymbol{c}} \sim \mathbb{P}_{\mathcal{C}_B}$, we also need to *iid* sample a fresh $(S, Q) \sim \mathcal{T}_{\boldsymbol{c}}$. To ensure independent sampling from the already seen meta-training (S, Q) pairs, we need to introduce new examples from $\mathbb{P}_c$ for each class $c \in \mathcal{C}_B$. Thus we construct slightly modified versions of four common FSL benchmarks **i)** *mini*ImageNet-Mod (*mini*-M) [39], in which we find (≈ 700) unused examples (from ImageNet) for each base class and use them to evaluate the performance over $\mathbb{P}_{\mathcal{C}_B}$. Here the original 600 examples of each base class are still only used for meta-training. **ii)** CIFAR-FS-Mod (*cifar*-M) [28], FC-100-Mod (FC-M) [3], and *tiered*ImageNet-Mod (*tiered*-M) [31]: As we don't have additional samples for base classes, we randomly partition each base class's current examples into an approximate $80/20$ split where the training tasks are constructed using the former and the latter is reserved for ID evaluation.

ID vs. OOD conflicts still exist. In addition to evaluating the four aforementioned meta-learning methods, we also consider two supervised pretraining methods: the supervised learning baseline (SB) [8] and Meta-Baseline (MB) [9]. Both of these methods have been shown to outperform many meta-learning methods on on the current OOD FSL benchmarks. For succintness, we call the performance over $\mathbb{P}_{\mathcal{C}_B}$ *BaseGen* and performance over $\mathbb{P}_{\mathcal{C}_N}$ *NovelGen* (we will refer to performance on validation tasks from $\mathbb{P}_{\mathcal{C}_V}$ *ValGen* in a later section). We plot the BaseGen and NovelGen performances of the aforementioned methods on *mini*ImageNet-Mod and *tiered*ImageNet-Mod in Figure 1(a)(b) (other datasets see Appendix E), with a dotted line connecting BaseGen and NovelGen value of same learned algorithm snapshot of a meta-learning method. **We see that the ID/OOD performance order flips still happen within current FSL benchmarks themselves, showing that the issue of improving OOD at the expense of ID is a realistic phenomenon for multiple benchmarks.** In addition, despite outperforming all meta-learning methods on NovelGen, the non-meta-learning methods SB and MB cannot beat the best meta-learning methods on BaseGen, which demonstrates their restrictive advantage. More broadly speaking, these OOD FSL datasets are constructed due to a belief that there exists a commonality between the training and test task distributions so that an algorithm capturing this commonality using the training distribution alone would also generalize to the test tasks. Thus the phenomenon of improving OOD while sacrificing ID means the algorithm is in some sense fails to learn the essence/commonality of the given training tasks. Additionally, we see that the BaseGen performance ranking of the four meta-learning methods are exactly the same as the performance order of the three ID benchmarks in Table 1. Thus researchers who want to also test on their proposed methods' ID generalization performance can perform BaseGen evaluation method in our proposed way as it is a simple addition to their existing NovelGen evaluation setup.

Certain OOD training choices might harm ID generalization. For a given meta-learning method we now ask whether specific meta-training choices found to be most effective on NovelGen would still be optimal for BaseGen. **[Meta-training hyperparameters]** In Figure 1(c) we see that a higher learning rate when training FOMAML finds algorithms that have higher BaseGen, whereas lower learning rates are better for NovelGen. Additionally, we see that the proposed technique of training

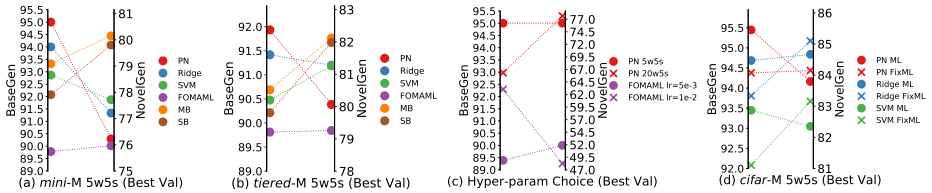


Figure 1: We show the BaseGen and NovelGen performance tradeoff (for best validation snapshots): over the choice of a set of four meta-learning and two supervised pre-training methods on *mini*-M (a) and *tiered*-M (b); over the number of ways to train PN on *mini*-M and different learning rates to train FOMAML on FC-M; over the use of FIX-ML (S, Q) generation strategy or not (ML) with SVM, RR and PN on *cifar*-M in (d).

with more ways for PN in [37] can lead to better NovelGen performance but the BaseGen is worse than 5 way PN whose training and test task configuration match. **[Meta-training (S, Q) generation alternatives]** It was found in [36] that always using the same support examples for every base class (FIX-ML) when constructing S can improve NovelGen performance for several meta-learning methods over multiple OOD FSL benchmarks. However, restricting the training (S, Q)'s diversity sampled from $\mathbb{P}_{\mathcal{C}_B}$ seems counter-intuitive and we wouldn't expect to improve the in-distribution generalization BaseGen. In Figure 1(d), we indeed see that FIX-ML only improves NovelGen performance at the expense of BaseGen by training over a much less diverse set of tasks. **These findings caution us that some training techniques used to boost test accuracy on FSL benchmarks might only work for the OOD scenario but not the ID scenario.**

5 Challenges With Out-of-Distribution Evaluation

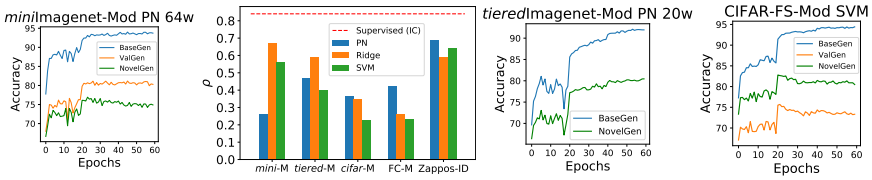
After identifying some example benchmarks for ID evaluation, we come back to the current OOD FSL benchmarks to further examine some subtle challenges in OOD evaluation. In particular, we highlight two concerns: **1)** Despite providing a validation set of classes for meta-validation, it is not clear whether this is a reliable way to perform model selection. **2)** As there is no requirement on how similar the training and test task distributions need to be in the OOD setting, there can be inconsistencies in method comparisons when the evaluation setup is modified.

5.1 Model Selection

To compare a set of meta-learning methods on the OOD test task distribution, one needs to find a single representative algorithm for each method. To do so, one should first decide what are the set of hyperparameter configurations to choose from for a given meta-learning method (we define the entire set of all hyperparameters for a training run as a *hyperparameter config*). For each considered config, after its training is completed, we need to choose one of the algorithm snapshots saved during training to represent it (which we call *snapshot selection*). Then the set of hyperparameter configs are compared based on their respectively selected snapshots, and a single config is then chosen among them (which we term *hyperparameter selection*). This config's selected snapshot will represent the given meta-learning method to be finally evaluated on the test tasks. We refer to the combined problem of hyperparameter and snapshot selection as model selection.

Snapshot vs Hyperparameter selection. If the snapshot selection strategy cannot reliably identify a snapshot with good test performance, it is possible that some hyperparameter configs will be unfairly represented by a mischosen snapshot, leading to erroneous hyperparameter selection. Thus we believe snapshot selection is a more fundamental problem and we focus our analyses on it (more on hyperparameter selection in Appendix F). In OOD FSL benchmarks, a reserved set of validation classes (disjoint from the set of novel classes) is provided, which as we have argued, provides tasks not *iid* from the test task distribution. As a result, we ask: *is validating on the given validation tasks reliable and if not are there other options?* In contrast, there is no need for such concerns in standard ID supervised learning, where the validation set is sampled *iid* from the test distribution.

***Option 1*: Snapshot selection tradition using ValGen.** By providing a set of validation classes, it has become the default practice for meta-learning methods to use ValGen performance for snapshot selection. However, because $\mathbb{P}_{\mathcal{C}_V}$ and $\mathbb{P}_{\mathcal{C}_N}$ are different tasks distributions, it is not clear whether a higher ValGen performance is *strongly correlated with* a higher NovelGen performance. In Figure 2(a), we plot the progression of ValGen and NovelGen of a 64-way trained PN on *mini*ImageNet-Mod 5w5s tasks. We notice that ValGen is consistently higher than NovelGen, indicating that meta-val performance is not an accurate estimator of NovelGen. More importantly, we see trendwise that while



(a) (b) (c) (d)

Figure 2: We plot the (Base, Val, Novel)Gen progression of 64(max)-way trained PN on *mini*-M in (a) and of SVM trained on *cifar*-M in (d). In (b) we compute the Kendall rank correlation coefficient (ρ) between the validation and test rankings of model snapshots for IC (trained on *cifar*) and algorithm snapshots for PN, SVM, RR on OOD datasets *mini*-M (last 40 epochs), *cifar*-M (last 40), FC-M (last 10), *tiered*-M (last 20) and ID dataset Zappos-ID (last 30); in (c) we show the BaseGen tracking NovelGen for 20-way trained PN on *tiered*-M.

ValGen is generally non-decreasing, NovelGen starts to decrease after epoch 30. Thus the snapshot selected according to ValGen is not the snapshot with the best possible meta-test performance.

Ranking similarity analysis. In light of the above observations, we ask a more general quantitative question: *How **similar** is the ranking of the training snapshots using meta-val performance (ValGen) **to** the ranking using the meta-test performance (NovelGen)?* To answer this, we compute the Kendall rank correlation coefficient [2] (ρ) [19] between the ValGen and NovelGen rankings of algorithm snapshots trained on four OOD FSL benchmarks (Figure 2(b)) and our ID benchmark Zappos-ID whose validation and test tasks come from the same distribution $\mathbb{P}_{\mathcal{A}}$. More concretely, for each meta-learning method and dataset combination, we save the algorithm snapshots (one from each epoch) throughout meta-training and rank these algorithm snapshots according to their ValGen and NovelGen value respectively. Then ρ is computed between the two rankings for this specific meta-learning method/dataset combination. For snapshot selection through ValGen to work reliably, we need ρ to be close to 1. For context, we also compute ρ for a standard supervised image classification problem (IC), where train, val and test examples are sampled from the same example-level distribution.

Unreliabity of ValGen snapshot selection. From Figure 2(b), we see that when using validation samples generated *iid* from the test distribution (Zappos-ID and supervised learning IC), the value of ρ is consistently higher than the OOD benchmarks, indicating the validation performance can more reliably track the trend of test performance in the ID setting than on the OOD FSL benchmarks. In particular, for the *cifar*-M and FC-M datasets, the ValGen ranking of algorithm snapshots seems to be only weakly correlated with the true meta-test ranking for all the meta-learning methods. Besides, the meta-val and meta-test rankings of the most useful snapshots can sometimes even be negatively correlated $\rho \approx -0.12 < 0$ (over all snapshots after epoch 30) in the training scenario shown in Figure 2(a). These results show that **on the OOD FSL benchmarks, snapshot selection using the pre-assigned validation tasks can sometimes be unreliable/unable to identify a snapshot candidate with the top-tier meta-test performance among all snapshots.**

***Option 2*: Snapshot selection alternative using BaseGen.** Beyond using OOD validation tasks for snapshot selection, inspired by the domain generalization community [21, 16], we consider using the ID performance over the training task distribution for snapshot selection. However, perhaps because of the lack of an ID evaluation setup in common FSL benchmarks, this possibility has not been widely considered. Enabled by our modifications of these benchmarks, we can now evaluate the ID generalization performance (BaseGen) throughout training in addition to ValGen. We plot how BaseGen and NovelGen progress for meta-learning methods trained on two different datasets in 2(c)(d). Here we see that on *tiered*-M, the BaseGen and NovelGen of PN both increase pretty consistently; thus picking the algorithm snapshot with the highest BaseGen performance (roughly the end-of-training snapshot) would also give close-to-best NovelGen. However, on *cifar*-M, after the learning rate drop at epoch 20, SVM's BaseGen keeps improving while NovelGen starts deteriorating. In this case, selecting snapshots according to the best BaseGen would pick a much worse snapshot than picking according to the best ValGen. This ambiguity of whether ID or OOD Validation based snapshot selection is better has also been documented in domain generalization, where Gulrajani and Lopez-Paz [16] find ID model selection can perform better in some settings while Koh et al. [21] find OOD validation model selection is better in others. Despite this ambiguity, **we believe that the commonly neglected in-distribution (BaseGen) snapshot selection approach should be considered by users of these OOD FSL benchmarks as a viable alternative to the default ValGen selection approach in appropriate settings**.

[2] $\rho \in [-1, 1]$, $\rho \approx 0$ means no correlation, while $\rho = 1/\rho = -1$ means exactly same/opposite rankings.

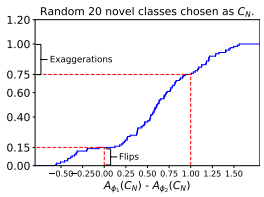
5.2 Inconsistencies in Meta-learning Method Performance Comparisons

After discussing concerns regarding OOD model selection for a given meta-learning method, we now analyze the reliability and consistency of conclusions drawn from comparing different meta-learning methods' OOD performance on these benchmarks. In particular, we focus on two cases:

***Inconsistency example 1*: Limited number of novel classes in a single benchmark.** Since in OOD FSL we specifically care about the learned algorithms' ability to quickly learn many unseen concepts, we should not be satisfied with an algorithm performing well only on tasks constructed from a small number of pre-selected novel classes. However, for many widely used FSL benchmarks (*mini*ImageNet, CIFAR-FS, FC-100), only 20 novel classes are used for meta-testing. Ideally, even if we don't measure it, we would hope that our learned algorithm would also generalize to tasks made by other set of classes different from the fixed small set of novel classes. To formalize this notion, we suppose the existence of a much larger collection of classes $\mathcal{C}_L$, where the novel classes $\mathcal{C}_N$ used for meta-testing is a small random subset with each element class sampled uniformly and non-repeatedly from $\mathcal{C}_L$ and fixed thereafter for NovelGen evaluation. Ideally, our goal is to evaluate an algorithm snapshot ϕ on the task distribution $\mathbb{P}_{\mathcal{C}_L}$ (denote this performance by $A_\phi(\mathcal{C}_L)$), yet during evaluation we only have access to the novel classes in $\mathcal{C}_N$ and thus we can only compute the performance $A_\phi(\mathcal{C}_N)$. It is easy to see that when we randomize over different choices of $\mathcal{C}_N$, the expected performance over the sampled novel classes would match the true performance: $\mathbb{E}_{\mathcal{C}_N} A_\phi(\mathcal{C}_N) = A_\phi(\mathcal{C}_L)$. However, when using a single randomly sampled novel set, the estimator $A_\phi(\mathcal{C}_N)$ can have high variance (see Appendix F). Instead of relying on $A_\phi(\mathcal{C}_N)$ to directly estimate $A_\phi(\mathcal{C}_L)$, we ask a more relaxed question: for a pair of algorithms A_{ϕ_1} and A_{ϕ_2} (given by two meta-learning methods), if the true performance $A_{\phi_1}(\mathcal{C}_L) - A_{\phi_2}(\mathcal{C}_L) = \epsilon > 0$, how frequently will we observe an opposite conclusion *i.e.,* $\mathbb{P}(A_{\phi_1}(\mathcal{C}_N) < A_{\phi_2}(\mathcal{C}_N))$ over a randomly sampled $\mathcal{C}_N$ (we call this event *conclusion flip*)? Additionally, it is also possible that the observed performance difference on $\mathcal{C}_N$ is greater than the true difference ϵ by some amount $\delta > 0$. In this case, the NovelGen observation would make the algorithm snapshot ϕ_1 look better than ϕ_2 more than it actually is on $\mathcal{C}_L$. Thus we also estimate the probability of $\mathbb{P}(A_{\phi_1}(\mathcal{C}_N) - A_{\phi_2}(\mathcal{C}_N) > \epsilon + \delta)$ and we call such events *improvement exaggeration*.

Checking the frequency of conclusion flips and exaggerations. To answer the above question empirically, we first suggest some larger class sets $\mathcal{C}_L$ for *mini* and *tiered*. For both we select unused classes from ImageNet disjoint from the base and validation classes. For *tiered*, we use all the remaining $1000 - 351$ (base) -97 (novel) $= 552$ classes as $\mathcal{C}_L$ while for *mini*, we randomly choose 120 or 552 (to match $|\mathcal{C}_L|$ in *tiered*) unused classes. We fix these $\mathcal{C}_L$ choices in the following analysis.

Figure 3(a) shows the empirical CDF of the performance differences $A_{\phi_1}(\mathcal{C}_N) - A_{\phi_2}(\mathcal{C}_N)$ computed over 100 randomly sampled size-20 novel class sets $\mathcal{C}_N$ for a fixed pair of PN and RR algorithm snapshots whose true performance difference over the larger 120 classes $\mathcal{C}_L$ is $\epsilon = 0.5\%$. In 15% of the cases the performance order is flipped from the true order, while in 25% of them improvements are exaggerated by more than 0.5% (total difference greater than 1%). Moreover, for some of the performance order flips, the observed performance difference can be quite negative $< -0.5\%$ thus significantly opposite to the true performance order. (Here for each run we evaluate both methods on 20,000 tasks sampled from $\mathbb{P}_{\mathcal{C}_N}$ in order to significantly reduce the randomness in estimating the true $A_{\phi_1}(\mathcal{C}_N), A_{\phi_2}(\mathcal{C}_N)$.) For comparison, we check for the conclusion flip and improvement exaggeration frequency when only a random subset of the full test set is used to compare two supervised learning image classification models with the same full test set performance difference of $\epsilon = 0.5$ (row (IC) in Table 3). Here we see that **compared to supervised learning, the chances of getting an incorrect performance comparison is much higher for the meta-learning OOD FSL benchmarks when evaluating on only 20 chosen novel classes (as done in several FSL benchmarks)**. Even worse, if we were to care about an even larger set set of underlying classes ($|\mathcal{C}_L| = 552$) despite still using only 20 random novel classes for evaluation comparison, the conclusions are even less reliable (Table 3(b) (i) vs (ii)). On the other hand, we do see that the performance comparison becomes comparatively more consistent if the true performance difference ϵ is higher (1% in (iii) compared to 0.5% in (ii)), despite that there still exists a statistically significant chance (10%) of getting an opposite conclusion. In practice, because **1)** we never specify exactly what and how big the underlying set of classes that we care about is, and **2)** some of the recent meta-learning methods (SVM vs PN on *cifar* in Table 2 of [23], R2-D2 vs GNN on *mini* in Table 1 of [3], FIX-ML [36]) sometimes only improve over the prior works by $< 1\%$, we believe the **meta-learning researchers should be aware of the possibility of getting a performance conclusion that is inconsistent over a single randomly chosen and fixed set of 20 novel classes used by some of these benchmarks.**



(a)

		ϵ	$\lvert\mathcal{C}_B\rvert$	$\lvert\mathcal{C}_N\rvert$	$\lvert\mathcal{C}_L\rvert$	CF	IE,0.5%
(IC)	*mini*-M	0.5%	100 imgs/class			3%	1%
(i)	*mini*-M	0.5%	64	20	120	15%	25%
(ii)	*mini*-M	0.5%	64	20	552	20%	24%
(iii)	*mini*-M	1.0%	64	20	552	10%	12%
(iv)	*tiered*-M	0.5%	64	20	552	20%	23%
(v)	*tiered*-M	0.5%	64	160	552	0%	7%
(vi)	*tiered*-M	0.5%	351	20	552	9%	15%
(vii)	*tiered*-M	0.5%	351	160	552	0%	1%

(b)

Dataset	ZAPPOS-OOD		*mini*-OOD
Methods	2w10s	2w5s	5w5s
PN	$^{(1)}80.51 \pm 0.13\%$	$^{(1)}74.67 \pm 0.14\%$	$^{(3)}76.22 \pm 0.14\%$
Ridge	$^{(1)}80.51 \pm 0.13\%$	$^{(2)}73.50 \pm 0.14\%$	$^{(2)}77.20 \pm 0.15\%$
SVM	$^{(3)}79.70 \pm 0.13\%$	$^{(3)}72.84 \pm 0.14\%$	$^{(1)}77.72 \pm 0.15\%$
FO-MAML	$^{(4)}72.89 \pm 0.14\%$	$^{(4)}68.12 \pm 0.15\%$	$^{(4)}75.96 \pm 0.17\%$

(c)

Figure 3: In (a) we show the CDF plot of $A_{\phi_1}(\mathcal{C}_N) - A_{\phi_2}(\mathcal{C}_N)$ over 100 randomly chosen sets of $\mathcal{C}_N$ with 20 (out of total $|\mathcal{C}_L|$=120) novel classes each. In (b) for different values of ϵ, $|\mathcal{C}_L|$, and training ($|\mathcal{C}_B|$), evaluation ($|\mathcal{C}_N|$) class sizes we show the percentage of conclusion flips (CF) and improvement exaggerations (IE) with $\delta = 0.5\%$ computed over 100 evaluations. In (c), we demonstrate the inconsistencies in performance rankings for PN, SVM, RR, and FOMAML on two OOD benchmarks: Zappos-OOD and *mini*-OOD.

Actionable suggestions. Since the size of the unknown underlying set $\mathcal{C}_L$ and the true difference ϵ might not be something one can directly control while designing the OOD benchmark, we now discuss two actionable choices that can reduce the chances of conclusion flips: **i) Use more novel classes in the evaluation:** By comparing (iv) vs (v) and (vi) vs (vii) in Table 3, we see that the frequency of CF and IE are **much** lower when 160 novel classes are used as opposed to 20 when $|\mathcal{C}_L|$ is the same. **ii) Train on more base classes:** The *tiered* dataset has more base classes ($|\mathcal{C}_B| = 351$ compared to 64 for *mini*) to train on. When comparing PN and RR snapshots trained on a modified version of *tiered* with fewer (randomly sampled 64 out of 351 to match *mini*) base classes, we see that the CF frequency is twice as high compared to when 351 base classes are used (Table 3(b)(iv) vs (vi)). Based on these two trends, for more reliable comparisons of meta-learning methods' OOD performance we suggest using datasets like *tiered*ImageNet and MetaDataset (both with much larger set of base and novel classes) in addition to the smaller benchmarks like *mini*ImageNet, CIFAR-FS, and FC-100, which some recent works [28, 3] still solely rely upon.

***Inconsistency example 2*: Consistency across multiple OOD FSL benchmarks.** Unlike the ID scenario, in the OOD FSL benchmarks the similarity of training and test distributions can vary a lot. Ideally, we want a meta-learning method to be consistently better on multiple OOD benchmarks with different shifts. Since Ren et al. [33] originally use the Zappos dataset for OOD evaluation, we also perform a similar evaluation on new attribute pairs based on their setup. At test time, we use an attribute set $\mathcal{A}'$ disjoint from the one used in the Zappos-ID setup $\mathcal{A}$, and sample attribute pairs from $\mathcal{A}'$ only. This induces a test task distribution $\mathbb{P}_{\mathcal{A}'}$ different from the training task distribution $\mathbb{P}_{\mathcal{A}}$. We evaluate different meta-learning methods on these Zappos-OOD tasks to see if the performance order is consistent with other OOD FSL benchmarks (Table 3(c)). Here we see that despite SVM outperforming RR and PN on *mini* NovelGen, the performance order of these three methods are completely flipped on Zappos-OOD. Similar observations can be made from TADAM underperforming PN in [33] (Table 2). This is in stark contrast to the consistency of performance rankings observed among the Zappos-ID dataset and all the BaseGen results of OOD FSL benchmarks (Section 4). Based on these findings, **we caution meta-learning researchers to be aware of such conclusion inconsistencies over different OOD FSL scenarios and reason carefully about the generality of their empirical findings when using only specific types of OOD datasets.**

6 Conclusion

In this paper, we categorize meta few-shot learning evaluation into two settings: in-distribution (ID) and out-of-distribution (OOD). After explaining why common FSL benchmarks reflect OOD evaluation, we identify realistic needs for ID FSL evaluation and provide new benchmarks as well as suggestions for modifying existing OOD FSL benchmarks to allow for ID evaluation. Through experiments performed on these ID benchmarks, we demonstrate a surprising phenomenon that many meta-learning methods/training techniques improve OOD performance while sacrificing ID performance. Beyond this, through quantitative analyses we show that even in the OOD scenario current FSL benchmarks may present subtle challenges with both model selection for a given meta-learning method and reliable performance comparisons of different methods. For these concerns, we provide initial suggestions and alternatives with the hope of alleviating these issues. Overall, we aim to raise awareness about the dichotomy of FSL evaluation and motivate the meta-learning community to collectively reason about ways to improve both ID and OOD evaluation.

References

[1] M. Al-Shedivat, L. Li, E. Xing, and A. Talwalkar. On data efficiency of meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1369–1377. PMLR, 2021.

[2] Y. Bai, M. Chen, P. Zhou, T. Zhao, J. D. Lee, S. Kakade, H. Wang, and C. Xiong. How important is the train-validation split in meta-learning? *arXiv preprint arXiv:2010.05843*, 2020.

[3] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

[4] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24:2178–2186, 2011.

[5] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[6] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[7] J. Chen, X.-M. Wu, Y. Li, Q. LI, L.-M. Zhan, and F.-l. Chung. A closer look at the training strategy for modern meta-learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[8] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. *International Conference on Learning Representations*, 2019.

[9] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[10] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

[11] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33, 2020.

[12] A. Fallah, A. Mokhtari, and A. Ozdaglar. Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *arXiv preprint arXiv:2102.03832*, 2021.

[13] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[14] C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[15] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[16] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[17] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[18] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[19] M. G. Kendall. *Rank correlation methods*. Griffin, London, 3d ed. edition, 1962.

[20] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar. Adaptive gradient-based meta-learning methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f4aa0dd960521e045ae2f20621fb4ee9-Paper.pdf.

[21] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

[22] H. B. Lee, H. Lee, D. Na, S. Kim, M. Park, E. Yang, and S. J. Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. *International Conference on Learning Representations*, 2020.

[23] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[24] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[25] S. Lin, G. Yang, and J. Zhang. A collaborative learning framework via federated meta-learning. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 289–299, Los Alamitos, CA, USA, dec 2020. IEEE Computer Society. doi: 10.1109/ICDCS47774.2020.00032. URL https://doi.ieeecomputersociety.org/10.1109/ICDCS47774.2020.00032.

[26] H. B. McMahan et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021.

[27] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[28] B. N. Oreshkin, P. Rodriguez, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018.

[29] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

[30] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[31] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[32] M. Ren, R. Liao, E. Fetaya, and R. Zemel. Incremental few-shot learning with attention attractor networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e833e042f509c996b1b25324d56659fb-Paper.pdf.

[33] M. Ren, E. Triantafillou, K.-C. Wang, J. Lucas, J. Snell, X. Pitkow, A. S. Tolias, and R. Zemel. Flexible few-shot learning with contextual similarity. *arXiv preprint arXiv:2012.05895*, 2020.

[34] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[35] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[36] A. Setlur, O. Li, and V. Smith. Is support set diversity necessary for meta-learning? *arXiv preprint arXiv:2011.14048*, 2020.

[37] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[38] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

[39] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[40] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014.

Checklist

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? [N/A]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
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)?
   [Yes] We include the code and instructions to setup the dataset and run experiments in the supplementary material. We also provide link to download the anonymously hosted data.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We document these in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   [Yes] We provide error bars for the estimated performance over each task distribution.
   (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] Details in Appendix.
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?
   [Yes] We use public datasets and cite the creators.
   (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? [Yes]
   (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]