
Meta-Album: Multi-domain Meta-Dataset for Few-Shot Image Classification

Ihsan Ullah^{*}, Dustin Carrión-Ojeda^{*¶}, Sergio Escalera^{#%}, Isabelle Guyon^{*%}, Mike Huisman⁺,
Felix Mohr[‡], Jan N. van Rijn⁺, Haozhe Sun^{*}, Joaquin Vanschoren[§], Phan Anh Vu^{*}

% ChaLearn, USA

¶ hessian.AI, Germany

Universitat de Barcelona, Spain

‡ Universidad de La Sabana, Colombia

¶ Technische Universität Darmstadt, Germany

* LISN/CNRS/INRIA, Université Paris-Saclay, France

§ TU/e Eindhoven University of Technology, The Netherlands

+ Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands

<https://meta-album.github.io/>

Abstract

We introduce Meta-Album, an image classification meta-dataset designed to facilitate few-shot learning, transfer learning, meta-learning, among other tasks. It includes 40 open datasets, each having at least 20 classes with 40 examples per class, with verified licences. They stem from diverse domains, such as ecology (fauna and flora), manufacturing (textures, vehicles), human actions, and optical character recognition, featuring various image scales (microscopic, human scales, remote sensing). All datasets are preprocessed, annotated, and formatted uniformly, and come in 3 versions (Micro \subset Mini \subset Extended) to match users' computational resources. We showcase the utility of the first 30 datasets on few-shot learning problems. The other 10 will be released shortly after. Meta-Album is already more diverse and larger (in number of datasets) than similar efforts, and we are committed to keep enlarging it via a series of competitions. As competitions terminate, their test data are released, thus creating a rolling benchmark, available through OpenML.org. Our website <https://meta-album.github.io/> contains the source code of challenge winning methods, baseline methods, data loaders, and instructions for contributing either new datasets or algorithms to our expandable meta-dataset.¹

fig/thumbnail/thumbnail_new.png

Figure 1: Meta-Album sample images. Each column represents one domain and each row one set. Domains are arranged in the same order as in Table 2.

¹All authors except for the first two authors (equal contributions) are in alphabetical order of last name.

1 Introduction

1.1 Background

Machine learning has progressed rapidly in recent years and has enabled breakthroughs in various domains. The success of most machine learning techniques hinges on the availability of large amounts of data [34, 63], limiting their applicability in domains where only little data is available. Enabling machine learning algorithms to learn new tasks from only a few examples is studied within the field of *few-shot learning* [43, 57, 65]. Novel meta-learning algorithms have recently been proposed targeting few-shot learning, triggering a surge of popularity for such problems [4, 22, 26, 75]. Despite the popularity of the field, progress is held back by a lack of good, challenging, and computationally feasible meta-datasets, that enable us to accurately assess the generalization abilities of few-shot learning algorithms. To remedy this, we introduce **Meta-Album** (Figure 1), an extensible **multi-domain meta-dataset**, including (so far) 40 image classification datasets from 10 different domains: 30 of these are currently available through our website, and the remaining 10 will be released in spring 2023. This is part of a long-term effort to create a publicly available and growing meta-dataset, in conjunction with a meta-learning challenge series (the 2021 and 2022 editions are both parts of the NeurIPS competition track [6, 12, 13]). As competitions terminate, older datasets get released, thus refreshing a rolling benchmark, made available through OpenML.org [71]. We check that all datasets are free for use in academic research and provide their original licenses.

Meta-Album was specifically designed to facilitate meta-learning research in the cross-domain few-shot setting, which is more realistic than commonly used evaluation protocols. Traditionally, few-shot learning algorithms (*e.g.*, [14, 24, 61]) have been evaluated by taking an existing benchmark dataset from a particular “domain” (*e.g.*, handwriting recognition) with a large number of classes, and then breaking it down into smaller classification **tasks**, each including a *random* subset of classes (*e.g.*, a few specific characters). Algorithms are then tested for their ability to solve such tasks “quickly” from a small number of examples, after being trained on many other tasks. Typically the number of classes N and examples per class k are both *fixed* in what is known as an N -way k -shot learning problem. While this setting has served research well, it is not very representative of practical real-world applications where tasks may come from various domains, include classes not drawn at random, but stemming from a class hierarchy, and include any number of classes and/or examples per class. By providing data from a wide variety of domains, including datasets with many classes and a minimum number of examples per class, and retaining class hierarchy annotations, Meta-Album enables benchmarking according to a variety of more realistic settings.

1.2 Related work

In this section, we review meta-datasets previously proposed to benchmark few-shot learning and meta-learning, as well as large-scale multi-class datasets, and then contrast them with Meta-Album.

Single dataset benchmarks: Omniglot [32] is often used as a starting benchmark for few-shot learning and meta-learning. MiniImageNet [73] and Tiered-ImageNet [54] are adapted for few-shot image classification from ImageNet [56]. CIFAR-FS [2] and FC100 [45] are remodeled from CIFAR-100 [30] for few-shot settings.

Multi-dataset benchmarks: A recent trend is to assemble numerous datasets from different domains in the same benchmark. Visual Decathlon [53] gathers 10 diverse datasets. The focus is on finding a model with a universal representation capacity for use in many tasks. VTAB (Visual Task Adaptation Benchmark) [82] assembles 19 image classification tasks across various domains. These tasks are grouped into 3 partitions: natural, specialized, and structured. Meta-Dataset [70] includes 10 image classification datasets from several application domains in one collection. Meta-Dataset also leverages the label hierarchy in ImageNet and Omniglot to organize the tasks.

Transfer learning and meta-learning benchmarks: VTAB + MD [10] attempts to unify common transfer and meta-learning datasets in a single benchmark. The authors also provide a comparison of popular meta- and transfer learning methods. BSCD-FSL (Broader Study of Cross-Domain Few-Shot Learning) [20] gathers 4 real-world tasks to compare few-shot, meta-learning, and transfer learning methods. WILDS [28] is a benchmark of 10 datasets of various modalities (images, graphs, and text; 6 of them are image datasets), reflecting a diverse range of distribution shifts that naturally arise in real-world applications, and hence useful to evaluate meta-learning and transfer learning techniques.

Table 1: Comparison between Meta-Album and other large-scale or (meta-) datasets

Dataset/ Meta-Dataset	# of domains	# of datasets	# of images	min/max classes per domain	min/max images per class	size on disk	multi-domain	lightweight (<20GB)	uniform # of images per class	uniform image size	repeated extensions
Meta-Dataset	7	10	53 068 000	43/1 696	3/140 000	210 GB	✓	✗	✗	✗	✗
VTAB	3	19	2 244 000	2/397	40/1 000	100 GB	✓	✗	✗	✗	✗
MS-COCO	1	1	328 000	80/80	9/10 777	44 GB	✗	✗	✗	✗	✗
Mini Imagenet	1	1	60 000	100/100	600/600	1 GB	✗	✓	✓	✓	✗
Omniglot	1	1	32 000	1 623/1 623	20/20	148 MB	✗	✓	✓	✓	✗
CUB-200	1	1	6 000	200/200	20/39	647 MB	✗	✓	✗	✗	✗
CIFAR-100	3	1	60 000	15/50	600/600	161 MB	✗	✓	✓	✓	✗
Meta-Album <i>Micro</i>	10	40	32 000	19/20	40/40	380 MB	✓	✓	✓	✓	✓
Meta-Album <i>Mini</i>	10	40	220 950	19/706	40/40	3.9 GB	✓	✓	✓	✓	✓
Meta-Album <i>Extended</i>	10	40	1 583 624	19/706	1/187 384	15 GB	✓	✓	✗	✓	✓

CTrL is a continual transfer-learning benchmark [72] including 7 commonly used datasets in image classification. In reinforcement learning, sets of simulation environments exist for meta- and transfer learning, such as Meta-World [81].

Outside few-shot and meta-learning: The AutoDL challenge [40] features a series of 66 datasets from numerous domains. These datasets cover a wide range of modalities: image, video, audio, text, and tabular. This competition focuses on finding a universal algorithm, which can solve many tasks without human supervision.

We compare Meta-Album with previous benchmarks/datasets in Table 1, and provide further details in Appendix G. Meta-Album covers a variety of domains, including ecology, manufacturing, textures, object classification, and character recognition, as well as a variety of scales: microscopic, macroscopic (human scale), or distant (remote sensing). While mostly re-purposing public datasets from heterogeneous sources to maximally vary recording conditions, we also introduce a few fresh datasets in OCR and ecology domains. Meta-Album comprises 3 different versions, *Micro* \subset *Mini* \subset *Extended*: **Micro** includes 20 classes and 40 images per class for ease of running sample code, **Mini** retains all original classes but also includes only 40 examples per class, while **Extended** includes all classes and examples. The variety of versions positions Meta-Album anywhere amongst small-scale datasets such as Omniglot [32], miniImageNet [73, 52] and CUB [74], which usually have at most 70 000 images in total and weigh at most a few GB, or very large-scale benchmarks such as Meta-dataset [70] and VTAB [10], which have more than 50 million images, weigh at least a few hundreds GB, and require high-end super-computer clusters. Its principal distinguishing feature is that it has, by far, the **largest number of domains and datasets**, collected in different conditions, and that it is designed to be **continually extended by either adding new domains or new datasets** in already existing domains, making it a tool of choice for cross-domain, domain-independent, and continual learning studies. Secondly, while other benchmarks usually provide only raw data, we **format all images uniformly as 128×128 pixel maps**, which has two benefits: reducing the storage/memory footprint and facilitating the benchmarking of methods independent of preprocessing steps. To that end, we optimized cropping and resizing to reduce dimensions as much as possible without degrading performance too much. In addition, Meta-Album includes datasets that have a **large number of classes** and class hierarchy annotations when available, with a **minimum number of classes and examples per class**: at least 20 classes (except one dataset having only 19 classes) with a minimum of 40 examples per class. This facilitates benchmark design, allowing us to vary the number of classes and the number of training examples per class over a large range of values. Finally and importantly, we selected datasets that are **not typically used in transfer-learning or meta-learning benchmarks**, *e.g.*, for pre-training backbone networks, such as ImageNet (which is included in *e.g.*, Meta-Dataset), or for conducting other meta-learning or transfer-learning experiments, such as Omniglot, CIFAR-100, SVHN, or MNIST (which are included in *e.g.*, VTAB and CTrL). This avoids giving an unfair advantage to methods that were developed using such commonly used datasets.

1.3 Contributions and recommended use

In summary, the contributions of our work are the following.

- We provide a **new meta-dataset for few-shot learning and meta-learning** consisting of 40 uniformly formatted datasets from 10 domains, which facilitates research in cross-domain meta-learning as well as practical and realistic evaluation of few-shot algorithms.
- We provide 3 versions of each dataset: Micro, Mini, and Extended to **facilitate usage by researchers with access to different amounts of computational power**.
- We **uniformly preprocessed and formatted data**, but also provide **instructions to retrieve the corresponding raw data** on our aforementioned website.
- We stimulate **community-driven benchmarking**, in conjunction with our challenge series, by welcoming new contributors and providing software and instructions to create additional datasets for Meta-Album, with strict quality control and review processes.
- We showcase our new meta-dataset by performing an **experimental evaluation** for several use cases, including transfer learning, few-shot meta-learning, and cross-domain few-shot meta-learning tasks, using a variety of algorithms, and we **open-source the code used**.

2 Meta-Album design and initial release

In this section, we explain the motivations behind the design of Meta-Album and present the 30 datasets included in the initial NeurIPS 2022 release. 10 more datasets are kept private, and will be released in spring 2023.

2.1 Motivation

Meta-Album emerged from a sequence of few-shot meta-learning benchmarks, following the problem formulations described in Section 3.1. The first of these was the 2020 MetaDL-mini challenge, which was run in conjunction with AAAI 2021 [12]. It followed the “within domain few-shot learning” protocol, and algorithms were evaluated with small-scale public datasets (Omniglot and CIFAR-100). Subsequently, we designed a first version of Meta-Album, including 15 datasets, for a larger-scale “within domain few-shot learning” challenge (MetaDL @ NeurIPS 2021 [13]). Here, algorithms were meta-trained and meta-tested on tasks extracted from a single dataset at a time, and performances were averaged over 5 datasets, both in the feedback phase and the final evaluation phase, to obtain a more robust evaluation. The 5 extra datasets were provided for practice purposes. The results of this challenge (further detailed in Section 3) indicated that these tasks were well within reach of state-of-the-art methods. This motivated us to move to the “cross-domain few-shot learning” setting. The design of this new challenge (part of an official NeurIPS 2022 challenge [6]) motivated us to grow Meta-Album to 30 datasets spanning multiple domains. We intend to continue growing Meta-Album and already have 10 more datasets lined up, in preparation for the next challenge. This will constitute a *rolling benchmark*: with each new challenge, previous feedback datasets are publicly released, previous final evaluation datasets become feedback datasets, and fresh datasets become final evaluation datasets.

Existing meta-datasets did not allow us to carry out our challenge program for several reasons: (1) they included datasets too familiar to the meta-learning community; (2) they did not include enough datasets to robustly evaluate participants (particularly in the cross-domain setting); (3) their datasets had a large variance in number of classes and examples per class, introducing bias in our experimental design. This required us to source new datasets. Furthermore, since these challenges include code submission, and providing the same resources to all participants, we needed to limit computational resources. Therefore, we had to downscale images while taking care that this does not significantly degrade performance.

2.2 Data search

Many people were involved in the sourcing of all datasets and their preparation, and they are gratefully acknowledged in our acknowledgements. This collaboration followed precise instructions to identify datasets that are: (i) from the same domain; (ii) freely available for academic research; (iii) having at

least 20 classes with at least 40 examples per class; (iv) with images of good enough quality by visual inspection and with no offensive material (we excluded “deprecated” datasets); (v) with baseline performance within a given range.

The last criterion was needed to ensure the success of our challenges, since tasks that are too easy or too hard do not allow us to separate challenge participants. For the purpose of designing Meta-Album, we defined a “domain” according to four characteristics: (1) application domain; (2) pattern recognition problem (texture or object classification); (3) scale: micro, human scale, or distant; (4) input channels. We ended up with 10 domains (see Table 2): Large animals, small animals, plants, plant diseases, microscopy, remote sensing, vehicles, manufacturing, human actions, and optical character recognition (OCR). Data sources were very varied, and mostly came from internet searches, but we also produced our own optical character recognition datasets and obtained novel donated data.

2.3 Data preparation

We performed several iterations of preprocessing, experiments, and analyses to prepare the datasets. This workflow included identifying and, when possible, correcting bias and artifacts (including artifacts we may have introduced by resizing and cropping images), and making sure that images are recognizable by human eye inspection.

As we work with datasets from diverse sources, each dataset requires a different preprocessing strategy, *e.g.*, the small animals’ datasets, plant-diseases datasets, manufacturing, and remote sensing datasets have images in different resolutions and orientations. However, usually, the object of interest lies in the middle of the image, which facilitated cropping images horizontally or vertically to get squared images. In some cases, *e.g.*, the plankton dataset, the image orientation depends on the shape of the plankton and the way it is photographed, *i.e.*, images have vertical or horizontal orientations based on the plankton in the image. Cropping images is not useful in this case because a big part of the plankton would be cropped. As such, we added a matching background to the images (either horizontally or vertically) by extending the top and bottom 3 rows or left and right columns respectively. In order to make sure that we do not introduce artifacts in the data, afterwards we applied a Gaussian kernel of size (29, 29) using open-cv [3] to the newly constructed background. In other cases, the area of interest was not necessarily centered, *e.g.*, human action datasets, and we had to use a human face detector to locate the subject, and then we cropped the upper body. For all datasets except for the optical character recognition datasets, we resized the images to a 128×128 resolution using an anti-aliasing filter [3]. The optical character recognition datasets are synthetically generated directly to the correct dimension by OmniPrint [64] (MIT license), and do not need further processing. The preprocessed data was formatted in a **data format** conserving as much meta-data as possible. For the micro and the mini version, only classes with at least 40 examples are kept for each dataset to maintain a balance between a large number of classes and sufficient examples per class while for the extended version, all classes and all images are kept. More details about data preparation and formatting can be found in Appendix C.

2.4 Initial Meta-Album release

The initial release of Meta-Album consists of 3 datasets for each of the 10 domains. Each dataset has 3 versions controlling the size, as explained in Section 1.2. All datasets are annotated with class labels and other meta-data. All 30 datasets were chosen after careful and critical analysis during the data preparation and quality control steps as described in Appendix C. Table 2 provides statistics on the various versions; Figure 1 shows sample images from each dataset. More details about datasets and their meta-data are listed in Appendix A. License information for all datasets can be found in Appendix B. Meta-Album datasets are being used in the **NeurIPS Cross-domain meta-learning Challenge 2022**. The first 30 datasets are available on **OpenML** [71], and later in spring 2023, 10 more datasets will be released, followed by other releases as our challenge program unfolds. Details about how to access Meta-Album datasets, contribute to the open meta-dataset, prepare new datasets with quality control, and submit these datasets for inclusion in Meta-Album can be found on the **Meta-Album Website**. This web page will also inform on software updates and revisions or new releases of our meta-dataset.

Table 2: Meta-Album: Datasets summary (*Mini versions*)

Domain ID	Domain Name	Set #	Dataset ID	Dataset Name	# Categories	# Images	Original source
LR_AM	Large Animals	0	BRD	Birds	315	12 600	Birds 400 [50]
		1	DOG	Dogs	120	4 800	Stanford Dogs [27]
		2	AWA	Animals with Attributes	50	2 000	AWA [78]
SM_AM	Small Animals	0	PLK	Plankton	86	3 440	WHOI [62]
		1	INS_2	Insects 2	102	4 080	Pest Insects [76]
		2	INS	Insects	104	4 160	SPIPOLL [59]
PLT	Plants	0	FLW	Flowers	102	4 080	Flowers [44]
		1	PLT_NET	PlantNet	25	1 000	PlantNet [18]
		2	FNG	Fungi	25	1 000	Danish Fungi [48]
PLT_DIS	Plant Diseases	0	PLT_VIL	PlantVillage	38	1 520	PlantVillage [23, 46]
		1	MED_LF	Medicinal Leaf	25	1 000	Medicinal Leaf [55]
		2	PLT_DOC	PlantDoc	27	1 080	Plant Doc [60]
MCR	Microscopy	0	BCT	Bacteria	33	1 320	DiBas [84]
		1	PNU	PanNuke	19	760	PanNuke [16, 17]
		2	PRT	Subcel. Human Protein	21	840	Protein Atlas [66]
REM_SEN	Remote Sensing	0	RESISC	RESISC	45	1 800	RESISC45 [8]
		1	RSICB	RSICB	45	1 800	RSICB128 [35]
		2	RSD	RSD	38	1 520	RSD46 [79, 41]
VCL	Vehicles	0	CRS	Cars	196	7 840	Cars [29]
		1	APL	Airplanes	21	840	Multi-type Aircraft [77]
		2	BTS	Boats	26	1 040	MARVEL [19]
MNF	Manufacturing	0	TEX	Textures	64	2 560	KTH-TIPS [15, 42] Kylberg [31] UIUC [33]
		1	TEX_DTD	Textures DTD	47	1 880	Texture DTD [9]
		2	TEX_ALOT	Textures ALOT	250	10 000	Texture ALOT [5]
HUM_ACT	Human Actions	0	SPT	100 Sports	73	2 920	100 Sports [49]
		1	ACT_40	Stanford 40 Actions	39	1 560	Stanford 40 Actions [80]
		2	ACT_410	MPH Human Pose	29	1 160	MPH Human Pose [1]
OCR	Optical Char. Recog.	0	MD_MIX	OmniPrint-MD-mix	706	28 240	
		1	MD_5_BIS	OmniPrint-MD-5-bis	706	28 240	OmniPrint [64]
		2	MD_6	OmniPrint-MD-6	703	28 120	

3 Use cases and baselines

This section illustrates how Meta-Album can be used for a variety of purposes. The code of all experiments is provided in our GitHub repository <https://github.com/ihsaan-ullah/meta-album>, and can serve as a basis to benchmark new algorithms against the baseline methods we investigate here. The problems investigated range from few-shot learning (for which Meta-Album was designed) to multi-class image classification, transfer learning, hierarchical classification, and continual learning. Because of lack of space, we only report few-shot learning experiments.

3.1 Problem setting

In this paper, we focus on **few-shot image classification**, where the goal is to **learn to perform new classification tasks from a limited number of examples**. Here, every task $\mathcal{T}_j = (\mathcal{D}_{\mathcal{T}_j}^{train}, \mathcal{D}_{\mathcal{T}_j}^{test})$ consists of a *support set* $\mathcal{D}_{\mathcal{T}_j}^{train}$ with training examples and a *query set* $\mathcal{D}_{\mathcal{T}_j}^{test}$ with test examples.² In N -way k -shot classification, we require that every *support set* contain exactly N classes with k examples per class ($kN = |\mathcal{D}_{\mathcal{T}_j}^{train}|$). Another requirement is that the classes in the query set must occur in the support set.

Few-shot learning does not necessarily require meta-learning. As in other “regular” learning problems, a *learner*, having available a set of training examples $\mathcal{D}_{\mathcal{T}_j}^{train}$ for a given task, can just return a *trained model* (classifier). But meta-learning is frequently used to enhance few-shot learning.

In a meta-learning problem, a *meta-learner*, having available a set of m training tasks $\mathcal{M}_{\mathcal{D}}^{train} = \{\mathcal{T}_j\}_{j=1}^m$, returns a meta-trained *learner*. In order to develop a meta-trained few-shot *learner*, available data organized in tasks $\mathcal{M}_{\mathcal{D}}$ (coming either from one or multiple datasets) are split into three “meta-splits” containing *disjoint sets of classes*: *meta-training* split $\mathcal{M}_{\mathcal{D}}^{train}$, *meta-validation* split $\mathcal{M}_{\mathcal{D}}^{valid}$, and *meta-testing* split $\mathcal{M}_{\mathcal{D}}^{test}$. The *learner* is meta-trained with $\mathcal{M}_{\mathcal{D}}^{train}$. During meta-training, the *learner* is evaluated with $\mathcal{M}_{\mathcal{D}}^{valid}$ every few meta-training cycles, to monitor progress. The

²The nomenclature *support set* instead of *training set*, and *query set* instead of *test set* is common in the meta-learning literature. It highlights the fact that, while meta-training is done on *tasks* = {*support set*, *query set*}, no actual test-data is presented to the classifier. The meta-test data also includes pairs of support and query sets, from which the ground truth of query set samples is hidden from the classifier.

Table 3: Datasets used in the NeurIPS 2021 MetaDL challenge [13].

Phase	Datasets according to Table 2
Feedback Phase	SM_AM.PLK, MDN.MLD, MNF.TEX_DTD, REM_SEN.RSICB, OCR.MD_MIX
Final test phase	SM_AM.INS, PLT_DIS.PLT_VIL, MNF.TEX, REM_SEN.RESISC, OCR.MD_5_BIS

final product of meta-training when the time budget has elapsed, is the *learner* with the highest performance on $\mathcal{M}_{\mathcal{D}}^{valid}$ tasks. It is then evaluated on tasks from $\mathcal{M}_{\mathcal{D}}^{test}$.

Within the realm of few-shot learning, we distinguish two cases. **Within domain few-shot learning** refers to the problem where data from the meta-validation and meta-test splits come from the same domain as meta-training data. Here, domain refers to one single dataset of Meta-Album \mathcal{D}_i , $i \in \{1, \dots, 30\}$. We enforce that \mathcal{D}_i is partitioned into $\mathcal{M}_{\mathcal{D}_i}^{train}$, $\mathcal{M}_{\mathcal{D}_i}^{valid}$, and $\mathcal{M}_{\mathcal{D}_i}^{test}$, using three disjoint sets of classes. In this setting, the goal of *learners* is to learn tasks including classes coming from the same original domain/dataset. If the *learner* has been meta-trained, **test tasks include new classes unseen during meta-training**. **Cross-domain few-shot learning**, in contrast, is a setting for which meta-split is performed at *dataset level* instead of *class level*. Once the *learner* has been meta-trained, **test tasks come from new datasets unseen during meta-training**. Note that as a consequence, there is still a slight domain overlap between the meta-train, meta-validation and meta-test dataset. For example, the meta-train dataset can contain observations from ten different datasets, including the ‘Fungi’ dataset, whereas the learner will be evaluated on a meta-test dataset constructed from ten different datasets, including the ‘Flowers’ dataset. This introduces two important challenges for the meta-learning algorithms whenever confronted with a given task in the meta-test set: 1) it has to deal with several classes in the meta-train set that are not related to the concepts from the task at hand. 2) while there are indeed observations in the meta-train set that are related to concept of the current task, these come from a different dataset, and might be sampled according to different conditions (different camera, lightning, geographical area, etc.). This aligns with the cross-domain setting introduced in the NeurIPS’22 meta-learning challenge [6]. Beyond the cross-domain setting, one can imagine a ‘domain independent’ setting, where each of the meta-train, meta-validation and meta-test datasets contain classes from different domains, and therefore no domain knowledge from the meta-train phase can be exploited.

We also distinguish between **fixed N-way k-shot** evaluations and **any-way any-shot** evaluations. The former requires fixing the value of N and k for the entire benchmark. The latter requires randomly choosing N and k for each task, within pre-defined ranges. Meta-Album allows us to choose $N \in [2, 20]$ and $k \in [1, 20]$.

3.2 Experiments

The first motivational use of Meta-Album has been the NeurIPS 2021 MetaDL challenge [13]. This was a meta-learning challenge with code submission, aiming at evaluating **few-shot learning methods in the within domain setting**, as described in Section 3.1. The evaluation was carried out with 600 tasks in the **5-way 5-shot setting**, using a subset of Meta-Album (see Table 3).

The solutions of the top participants have been **open-sourced**. In a paper, authored collaboratively between the competition organizers and the top-ranked participants [13], we analyse the results of the competition. The lessons learned include that learning good representations is essential for effective transfer learning. The winner’s solution MetaDelta++ [7], based on a combination of pre-trained backbone networks, performed best on all final 5 test phase datasets, with high accuracy scores (0.98, 0.94, 0.99, 0.92, 0.94). This indicates that, in future challenges, we are ready to tackle harder tasks, and motivated us to move to **cross-domain few-shot learning**, in the **any-way any-shot setting** for the NeurIPS 2022 challenge [6]. Fine-tuning backbones on meta-training data turned out to be important, though there are indications that off-the-shelf backbones pre-trained with self-supervised learning on massive datasets might become the way of the future, essentially making meta-learning unnecessary for image classification problems. Thus, meta-learning should be benchmarked in **de novo training conditions**, in the future, to prepare for scenarios (in other domains) in which such backbones are not available. The NeurIPS 2022 challenge encourages *de novo* training in a dedicated

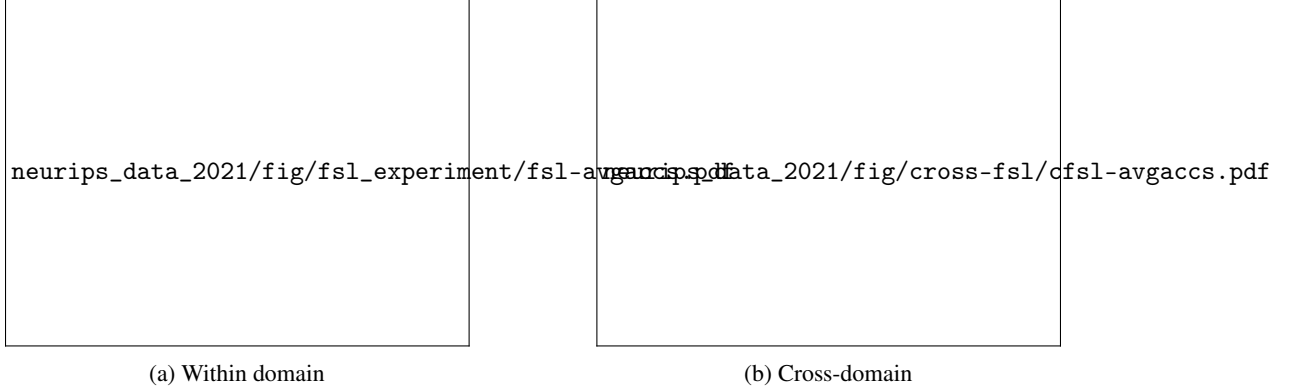


Figure 2: **Comparison of “within domain” and “cross-domain” few-shot learning.** We plot 5-way [1, 5, 10, 20]-shot learning meta-test mean task accuracy, averaged over 1 800 tasks drawn from the 30 released Meta-Album datasets. Corresponding 95% confidence intervals are within the size of the symbols.

league. Appendix D contains a detailed analysis of the difficulty of all Meta-Album datasets following the NeurIPS 2021 MetaDL challenge protocol.

Difficulty of cross-domain few-shot learning

Although several methods have been proposed in the state-of-the-art to tackle the cross-domain few-shot learning problem [11, 36, 37, 38, 39, 51, 69], they require too much time or are not compatible with our fully supervised setting. Therefore, we investigated the few-shot learning performance of popular meta-learning methods: MAML [14], Matching networks [73], and Prototypical networks [61]. We compared them against two baseline methods: TrainFromScratch (learning every task starting from a random initialization at meta-test time, *i.e.*, no meta-learning) and FineTuning, which is pre-trained on the classification problem arising from concatenating all meta-training classes and corresponding data and only fine-tunes the last layer at meta-test time [7]. It is worth mentioning that the purpose of our baseline methods is to give a set of “classical” and “representative” techniques, not to be exhaustive.

For a given dataset, all meta-learning techniques are meta-trained on some tasks. However, the backbone used for FineTuning is meta-trained (pre-trained) on some number of randomly sampled batches of fixed size. The performance of trainers is validated periodically (or batches in case of pre-training the FineTuning backbone). The query set for every task contains max possible number examples per class, following [7]. The learning algorithm with the best validation performance is evaluated on some number of meta-test tasks randomly sampled from the meta-testing split, which has information from unseen classes during training and validation. We average the results over 3 runs with different random seeds.

Results are shown in Figure 2. A first observation is that Prototypical Networks (ProtoNet) dominate other algorithms (both within domain and cross-domain) and that the ranking of algorithms does not significantly change with the number of shots. However, the exception is FineTuning for 1-shot learning in the cross-domain configuration, which outperforms ProtoNet by a small margin. Moreover, we observe that FineTuning outperforms MAML and Matching Networks (the other episodic meta-learning algorithm we tried), corroborating findings showing that finetuning yields excellent few-shot learning performance without using episodic meta-learning [7, 25, 67, 68]. We also see that the naive baseline TrainFromScratch yields the worst performance, indicating that meta-learning actually helps transfer knowledge to new tasks. Furthermore, we observe that the performances improve with the number of shots (training examples per class). Lastly, the details provided in Appendix E and Appendix F show that FineTuning is the fastest method at training time while ProtoNet and MatchingNet are the fastest methods at inference time with less than 1 second per task.

For cross-domain few-shot learning, as can be expected, the accuracy is lower since the problem is more complex. However, it does not dramatically decrease compared to within domain few-shot learning, which leads us to speculate that such a new problem is within reach of the current state-



Figure 3: **Comparison of “cross-domain” few-shot learning using fixed and variable number of ways and shots, and influence of number of ways on performance.** We plot few-shot learning meta-test mean task accuracy, averaged over test tasks drawn from the 30 released Meta-Album datasets. Corresponding 95% confidence intervals are almost imperceptible as they are around ± 0.15 .

of-the-art. This gives rise to new opportunities for improvement in this more complicated and more realistic setting.

Difficulty of “any”-way “any”-shot learning

Moving to yet more realistic and harder tasks, we also investigated the performance in the “any”-way “any”-shot setting, where tasks at meta-test time include a varying number of classes between 2 to 20 and a varying number of examples per class between 1 to 20. For example, at meta-test and meta-validation time, some carved out tasks might be as follows: **Test task 1:** 5-way 1-shot task from Dataset 9; **Test task 2:** 3-way 15-shot task from Dataset 3; **Test task 3:** 12-way 4-shot task from Dataset 8; etc. However, during the meta-training phase, we kept the number of classes constant (specifically, we used 5-way any-shot tasks). This facilitates using off-the-shelf meta-learning techniques. All other experimental conditions (hyper-parameters, computational resources) are the same as in the previous section.

In Figure 3a we can observe that the complexity of the any-way any-shot setting is similar to the 5-way 1-shot setting. Nevertheless, the meta-learning approaches (ProtoNet, MatchingNet, MAML) adapt better to this novel setting since their performance is better than the one achieved in the 5-way 1-shot setting, while the performance of FineTuning and TrainFromScratch is worse compared to the same setting. Additionally, the results presented in Figure 3b and Appendix F show that the dominant difficulty factor in any-way any-shot learning is the variability in the number of ways since as it can be seen, the performance of the evaluated methods is highly affected by the increment in this number. This is supported by the fact that the absolute Pearson correlation between the number of ways and the test accuracy is larger ($r=-0.55$, $p<0.05$) than the correlation between the number of shots and the accuracy ($r=0.1$, $p<0.05$). Therefore, we anticipate that this new setting of any-way any-shot learning will deliver new interesting results in the upcoming challenge.

4 Discussion and conclusion

We introduce Meta-Album, a new meta-dataset for few-shot image classification, which is both practical and extensive: it includes many datasets from a wide variety of domains, all preprocessed to allow training according to different settings on commodity GPUs. It is especially amenable to evaluating meta-learning and transfer learning techniques. It can also be used for hierarchical classification as well as domain adaptation, due to the presence of overlapping classes between datasets, and continual learning, where algorithms are progressively trained across datasets.

We evaluate the utility of Meta-Album using a range of few-shot learning experiments. Our findings include that Prototypical Networks and the FineTuning baseline perform quite well. This corroborates the results of the NeurIPS’21 challenge, in which the winners capitalized on the use of pre-trained backbones, obtaining results in the high 90% classification accuracy in the “within domain” 5-way

5-shot setting [13]. Meta-Album will further challenge the research community by being considerably larger and by mixing tasks from multiple domains, in [2-20]-way [1-20]-shot settings. Furthermore, Meta-Album allows *de novo* training. We tested and compared this new framework to that of previous challenges and demonstrated an increased difficulty on all our baseline methods.

In future work, we want to make tasks more challenging by ensuring that every task consists of related concepts, belonging to a same super-class. For example, insects from the Coleoptera order have more resemblance with one another than with insects coming from another order, *e.g.*, butterflies. As such, it would be more challenging to have a task where the goal is to classify various insects from the Coleoptera order, rather than a task where insects from various orders are combined into one classification task. This requires datasets in which class hierarchies are provided, and we currently have only a few of those. Further work also includes introducing the even more difficult “domain independent” settings, in which meta-training and meta-testing are performed on entirely different domains. Indeed, a problem setting where the goal is to learn how to learn on tasks that are not related to the meta-test data, would truly challenge a meta-learning system.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the data owners/creators:

LR_AM.BRD: Gerald Piosenka, *Scottsdale, Arizona, United States*; **LR_AM.DOG:** Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei from *Stanford University*; **LR_AM.AWA:** Christoph H. Lampert, Bernt Schiele and Zeynep Akata.

SM_AM.PLK: Heidi M. Sosik, Emily E. Peacock, Emily F. Brownlee and Eric Orenstein from *Woods Hole Oceanographic Institution, United States*; **SM_AM.INS:** Grégoire Lois, Colin Fontaine and Jean-Francois Julien from *National Museum of Natural History Paris, France* and *SPIOLL Science project*; **SM_AM.INS_2:** Xiaoping Wu, Chi Zhan, Yukun Lai, Ming-Ming Cheng and Jufeng Yang.

PLT.FLW: Maria-Elena Nilsback and Andrew Zisserman from *University of Oxford, England*; **PLT.PLT_NET:** Garcin Camille, Joly Alexis, Bonnet Pierre, Lombardo Jean-Christophe, Affouard Antoine, Chouet Mathias, Servajean Maximilien, Salmon Joseph and Lorieul Titouan; **PLT.FNG:** Lukáš Pícek, Milan Šulc, Jiří Matas, Jacob Heilmann-Clausen, Thomas S. Jeppesen, Thomas Læssøe and Tobias Frøslev.

PLT_DIS.PLT_VIL: Sharada Mohanty, David Hughes, and Marcel Salathé, from *EPFL Switzerland* and *Penn State University*, J. Arun Pandian and G. Geetharamani, from *Department of Mathematics, University College of Engineering, Anna University - BIT Campus* and *Department of Computer Science and Engineering, M.A.M. College of Engineering and Technology, Tiruchirappalli, India*; **PLT_DIS.MED_LF:** S Roopashree, J Anitha, from *Visvesvaraya Technological University, R V Institute of Management, Dayananda Sagar University, India*; **PLT_DIS.PLT_DOC:** Sharada Mohanty, David Hughes, and Marcel Salathé.

MCR.BCT: Bartosz Zieliński, Anna Plichta, Krzysztof Misztal, Przemysław Spurek, Monika Brzywczy-Włoch and Dorota Ochońska from *Uniwersytet Jagielloński*; **MCR.PRT:** Peter J Thul, Lovisa Akesson, Mikaela Wiking, Diana Mahdessian, Aikaterini Geladaki, Hammou Ait Blal, Tove Alm, Anna Asplund, Lars Björk, Lisa Breckels, and others from *Protein Atlas*; **MCR.PNU:** Gamper Jevgenij, Koohbanani Navid Alemi, Benet Ksenija, Khuram Ali and Rajpoot Nasir from *University of Warwick*.

REM_SEN.RESISC: Gong Cheng, Junwei Han, and Xiaoqiang Lu from *Northwestern Polytechnical University, Xi'an, China*; **REM_SEN.RSICB:** Haifeng Li, Xin Dou, Chao Tao, Zhixiang Hou, Jie Chen, Jian Peng, Min Deng, Ling Zhao from *Central South University, Changsha, China*; **REM_SEN.RSD:** Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu, Deren Li, Chunshan Wei, Gefu Tang and Junyi Liu from *State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China*.

VCL.CRS: Jonathan Krause, Michael Stark, Jia Deng and Li Fei-Fei from *Stanford University*; **VCL.APL:** Wu Zhize; **VCL.BTS:** Gundogdu E., Solmaz B, Yucesoy V., Koc A.

MNF.TEX: Eric Hayman, Barbara Caputo, Mario Fritz, P. Mallikarjuna and Alireza Tavakoli Targhi from *KTH Royal Institute of Technology in Stockholm* (for KTH TIPS and KTH TIPS 2);

Gustaf Kylberg from *Uppsala University, Sweden* (for Kylberg Texture); Jean Ponce, Svetlana Lazebnik and Cordelia Schmid from *University of Illinois Urbana-Champaign* (for UIUC Textures); **MNF.TEX_DTD**: Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed and Andrea Vedaldi, the Authors of Describable Textures Dataset (DTD); **MNF.TEX_ALOT**: Gertjan Burghouts and Jan-Mark Geusebroek from *University of Amsterdam, Netherlands*.

HUM_ACT.SPT: Gerald Piosenka, *Scottsdale, Arizona, United States*; **HUM_ACT.ACT_40**: B. Yao, X. Jiang, A. Khosla, A.L. Lin, L.J. Guibas, and L. Fei-Fei from *Stanford University*; **HUM_ACT.ACT_410**: Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele Bernt.

OCR.MD_MIX, OCR.MD_5_BIS, OCR.MD_6: Generated by Haozhe Sun (co-author).

We acknowledge the efforts of Philip Boser, Maria Belen Guaranda Cabezas, Jilin He, Felix Heron, Gabriel Lauzzana, Romain Mussard, and Manh Hung Nguyen for datasets and datasheets preparation. We also received useful input from many members of the TAU team of the LISN laboratory, Wei Wei Tu from 4Paradigm Inc, China, and the MetaDL technical crew: Adrian El Baz, Zhengying Liu, Adrien Pavao, Jennifer (Yuxuan) He, Yui Man Lui, Sébastien Treguer, Benjia Zhou, and Jun Wan, who participated in identifying datasets and contributed to discussions. We also would like to thank the hundreds of volunteers involved in the SPIOLL citizen science program who pictured and identified insects.

This work was supported by ChaLearn, the ANR (Agence Nationale de la Recherche, National Agency for Research) under AI chair of excellence HUMANIA, grant number ANR-19-CHIA-0022 and Labex Digicosme project ANR11LABEX0045DIGICOSME operated by ANR as part of the program Investissement d’Avenir Idex Paris Saclay (ANR11IDEX000302). In addition, some experiments were performed using the compute resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by Leiden University. This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

- [1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3686–3693.
- [2] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi. “Meta-learning with differentiable closed-form solvers”. In: *7th International Conference on Learning Representations*. 2019.
- [3] G. Bradski. “The openCV library.” In: *Dr. Dobb’s Journal: Software Tools for the Professional Programmer* 25.11 (2000), pp. 120–123.
- [4] P. Brazdil, J. N. van Rijn, C. Soares, and J. Vanschoren. *Metalearning: Applications to Automated Machine Learning and Data Mining*. 2nd. Springer, 2022.
- [5] G. J. Burghouts and J.-M. Geusebroek. “Material-Specific Adaptation of Color Invariant Features”. In: *Pattern Recognition Letters* 30.3 (2009), pp. 306–313.
- [6] D. Carrión-Ojeda, H. Chen, A. El Baz, S. Escalera, C. Guan, I. Guyon, I. Ullah, X. Wang, and W. Zhu. “NeurIPS’22 Cross-Domain MetaDL competition: Design and baseline results”. In: *ECMLPKDD Workshop on Meta-Knowledge Transfer*. Vol. 191. Proceedings of Machine Learning Research. 2022.
- [7] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang. “A Closer Look at Few-shot Classification”. In: *7th International Conference on Learning Representations*. 2019.
- [8] G. Cheng, J. Han, and X. Lu. “Remote Sensing Image Scene Classification: Benchmark and State of the Art”. In: *Proceedings of the IEEE* 105.10 (2017), pp. 1865–1883.
- [9] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. “Describing Textures in the Wild”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3606–3613.
- [10] V. Dumoulin, N. Houlsby, U. Evci, X. Zhai, R. Goroshin, S. Gelly, and H. Larochelle. *Comparing Transfer and Meta Learning Approaches on a Unified Few-Shot Classification Benchmark*. 2021. arXiv: [2104.02638](#).
- [11] N. Dvornik, C. Schmid, and J. Mairal. *Selecting Relevant Features from a Universal Representation for Few-shot Classification*. 2020. arXiv: [2003.09338](#).

- [12] A. El Baz, I. Guyon, Z. Liu, J. N. van Rijn, S. Treguer, and J. Vanschoren. “MetaDL challenge design and baseline results”. In: *AAAI Workshop on Meta-Learning and MetaDL Challenge*. Vol. 140. Proceedings of Machine Learning Research. 2021, pp. 1–16.
- [13] A. El Baz et al. “Lessons learned from the NeurIPS 2021 MetaDL challenge: Backbone fine-tuning without episodic meta-learning dominates for few-shot learning image classification”. In: *NeurIPS 2021 Competitions and Demonstrations Track*. Vol. 176. Proceedings of Machine Learning Research. 2022, pp. 80–96.
- [14] C. Finn, P. Abbeel, and S. Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. 2017, pp. 1126–1135.
- [15] M. Fritz, E. Hayman, B. Caputo, and J. Eklundh. *THE KTH-TIPS database*. <https://www.csc.kth.se/cvap/databases/kth-tips/index.html>. 2004.
- [16] J. Gamper, N. A. Koohbanani, K. Benet, A. Khuram, and N. Rajpoot. “PanNuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification”. In: *European Congress on Digital Pathology*. Springer. 2019, pp. 11–19.
- [17] J. Gamper, N. A. Koohbanani, S. Graham, M. Jahanifar, S. A. Khurram, A. Azam, K. Hewitt, and N. M. Rajpoot. *PanNuke Dataset Extension, Insights and Baselines*. 2020. arXiv: [2003.10778](#).
- [18] C. Garcin, alexis joly, P. Bonnet, A. Affouard, J.-C. Lombardo, M. Chouet, M. Servajean, T. Lorieul, and J. Salmon. “Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. 2021.
- [19] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koç. “MARVEL: A Large-Scale Image Dataset for Maritime Vessels”. In: *Computer Vision – ACCV 2016*. Vol. 10115. Lecture Notes in Computer Science. 2017, pp. 165–180.
- [20] Y. Guo, N. C. Codella, L. Karlinsky, J. V. Codella, J. R. Smith, K. Saenko, T. Rosing, and R. Feris. *A Broader Study of Cross-Domain Few-Shot Learning*. 2020. arXiv: [1912.07200 \[cs.CV\]](#).
- [21] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.
- [22] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. “Meta-learning in neural networks: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), pp. 5149–5169.
- [23] D. P. Hughes and M. Salathé. *An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing*. 2015. arXiv: [1511.08060](#).
- [24] M. Huisman, A. Plaat, and J. N. van Rijn. “Stateless neural meta-learning using second-order gradients”. In: *Machine Learning* 111.9 (2022), pp. 3227–3244.
- [25] M. Huisman, J. N. van Rijn, and A. Plaat. “A Preliminary Study on the Feature Representations of Transfer Learning and Gradient-Based Meta-Learning Techniques”. In: *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*. 2021.
- [26] M. Huisman, J. N. van Rijn, and A. Plaat. “A survey of deep meta-learning”. In: *Artificial Intelligence Review* (2021).
- [27] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. “Novel Dataset for Fine-Grained Image Categorization”. In: *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO, 2011.
- [28] P. W. Koh et al. “WILDS: A Benchmark of in-the-Wild Distribution Shifts”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 5637–5664.
- [29] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. “3D Object Representations for Fine-Grained Categorization”. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 554–561.
- [30] A. Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009.

- [31] G. Kylberg. *The Kylberg Texture Dataset v. 1.0*. External report (Blue series) 35. Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, 2011.
- [32] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015), pp. 1332–1338.
- [33] S. Lazebnik, C. Schmid, and J. Ponce. “A sparse texture representation using local affine regions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1265–1278.
- [34] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.
- [35] H. Li, X. Dou, C. Tao, Z. Wu, J. Chen, J. Peng, M. Deng, and L. Zhao. “RSI-CB: A Large-Scale Remote Sensing Image Classification Benchmark Using Crowdsourced Data”. In: *Sensors* 20.6 (2020).
- [36] P. Li, S. Gong, C. Wang, and Y. Fu. “Ranking Distance Calibration for Cross-Domain Few-Shot Learning”. In: *2022 IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9099–9108.
- [37] W.-H. Li, X. Liu, and H. Bilen. “Cross-domain Few-shot Learning with Task-specific Adapters”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7161–7170.
- [38] W.-H. Li, X. Liu, and H. Bilen. “Universal Representation Learning from Multiple Domains for Few-shot Classification”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021, pp. 9506–9515.
- [39] L. Liu, W. L. Hamilton, G. Long, J. Jiang, and H. Larochelle. “A Universal Representation Transformer Layer for Few-Shot Image Classification”. In: *9th International Conference on Learning Representations*. 2021.
- [40] Z. Liu et al. “Winning solutions and post-challenge analyses of the ChaLearn AutoDL challenge 2019”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.9 (2021), pp. 3108–3125.
- [41] Y. Long, Y. Gong, Z. Xiao, and Q. Liu. “Accurate object localization in remote sensing images based on convolutional neural networks”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.5 (2017), pp. 2486–2498.
- [42] P. Mallikarjuna, A. T. Targhi, M. Fritz, E. Hayman, B. Caputo, and J. Eklundh. *THE KTH-TIPS 2 database*. <https://www.csc.kth.se/cvap/databases/kth-tips/index.html>. 2006.
- [43] D. K. Naik and R. J. Mammone. “Meta-neural networks that learn by learning”. In: *Proceedings of the International Joint Conference on Neural Networks*. IEEE. 1992, pp. 437–442.
- [44] M. Nilsback and A. Zisserman. “Automated Flower Classification over a Large Number of Classes”. In: *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. 2008, pp. 722–729.
- [45] B. N. Oreshkin, P. R. Lopez, and A. Lacoste. “TADAM: Task dependent adaptive metric for improved few-shot learning”. In: *Advances in Neural Information Processing Systems* 31. 2018, pp. 719–729.
- [46] J. A. Pandian and G. Geetharamani. “Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network”. In: *Mendeley Data*, V1 (2019).
- [47] P. Pérez, M. Gangnet, and A. Blake. “Poisson Image Editing”. In: *ACM Transactions on Graphics* 22.3 (2003), pp. 313–318.
- [48] L. Picek, M. Šulc, J. Matas, T. S. Jeppesen, J. Heilmann-Clausen, T. Læssøe, and T. Frøslev. “Danish Fungi 2020 - Not Just Another Image Recognition Dataset”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1525–1535.
- [49] G. Piosenka. *100 Sports Image Classification*. <https://www.kaggle.com/datasets/gpiosenka/sports-classification>.
- [50] G. Piosenka. *BIRDS 450 - SPECIES IMAGE CLASSIFICATION*. <https://www.kaggle.com/datasets/gpiosenka/100-bird-species>.
- [51] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine. “Meta-Learning with Implicit Gradients”. In: *Advances in neural information processing systems* 32. 2019.

- [52] S. Ravi and H. Larochelle. “Optimization as a Model for Few-Shot Learning”. In: *5th International Conference on Learning Representations*. 2017.
- [53] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. “Learning multiple visual domains with residual adapters”. In: *Advances in neural information processing systems 30*. 2017.
- [54] 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”. In: *6th International Conference on Learning Representations*. 2019.
- [55] S. Roopashree and J. Anitha. *Medicinal Leaf Dataset*. <https://data.mendeley.com/datasets/nnytj2v3n5/1>. Version 1. 2020.
- [56] O. Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.
- [57] J. Schmidhuber. “Evolutionary Principles in Self-Referential Learning”. Diploma Thesis. Technische Universität München, 1987.
- [58] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.
- [59] H. Serret, N. Deguines, Y. Jang, G. Lois, and R. Julliard. “Data quality and participant engagement in citizen science: comparing two approaches for monitoring pollinators in France and South Korea”. In: *Citizen Science: Theory and Practice* 4.1 (2019).
- [60] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra. “PlantDoc: A Dataset for Visual Plant Disease Detection”. In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. 2020, pp. 249–253.
- [61] J. Snell, K. Swersky, and R. Zemel. “Prototypical Networks for Few-shot Learning”. In: *Advances in Neural Information Processing Systems 30*. 2017.
- [62] H. M. Sosik, E. E. Peacock, and E. F. Brownlee. *Annotated Plankton Images - Data Set for Developing and Evaluating Classification Methods*. <https://hdl.handle.net/10.1575/1912/7341>. 2015.
- [63] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 843–852.
- [64] H. Sun, W.-W. Tu, and I. M. Guyon. “OmniPrint: A Configurable Printed Character Synthesizer”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. 2021.
- [65] S. Thrun. “Lifelong Learning Algorithms”. In: *Learning to learn*. 1998, pp. 181–209.
- [66] P. J. Thul, L. Åkesson, M. Wiking, D. Mahdessian, A. Geladaki, H. Ait Blal, T. Alm, A. Asplund, L. Björk, L. M. Breckels, et al. “A subcellular map of the human proteome”. In: *Science* 356.6340 (2017).
- [67] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. “Rethinking few-shot image classification: a good embedding is all you need?”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 266–282.
- [68] E. Triantafillou, V. Dumoulin, H. Larochelle, and R. Zemel. *Learning Flexible Classifiers with Shot-CONditional Episodic (SCONE) Training*. <https://openreview.net/forum?id=OMjC3uMthAb>. 2021.
- [69] E. Triantafillou, H. Larochelle, R. Zemel, and V. Dumoulin. “Learning a Universal Template for Few-shot Dataset Generalization”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 10424–10433.
- [70] E. Triantafillou et al. “Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples”. In: *8th International Conference on Learning Representations*. 2020.
- [71] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. “OpenML: networked science in machine learning”. In: *SIGKDD Explorations* 15.2 (2013), pp. 49–60.
- [72] T. Veniat, L. Denoyer, and M. Ranzato. “Efficient Continual Learning with Modular Networks and Task-Driven Priors”. In: *9th International Conference on Learning Representations*. 2021.
- [73] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. “Matching networks for one shot learning”. In: *Advances in Neural Information Processing Systems 29* (2016), pp. 3630–3638.

- [74] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.
- [75] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. “Generalizing from a Few Examples: A Survey on Few-Shot Learning”. In: *ACM Computing Surveys* 53.3 (2020).
- [76] X. Wu, C. Zhan, Y. Lai, M.-M. Cheng, and J. Yang. “IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition”. In: *2019 IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8787–8796.
- [77] Z. Wu. *Multi-type Aircraft of Remote Sensing Images: MTARSI*. <https://zenodo.org/record/3464319#.Y0FAAJBByrc>. 2019.
- [78] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. “Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.9 (2019), pp. 2251–2265.
- [79] Z. Xiao, Y. Long, D. Li, C. Wei, G. Tang, and J. Liu. “High-resolution remote sensing image retrieval based on CNNs from a dimensional perspective”. In: *Remote Sensing* 9.7 (2017).
- [80] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. “Human action recognition by learning bases of action attributes and parts”. In: *Proceedings of the IEEE international conference on computer vision*. 2011, pp. 1331–1338.
- [81] T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman, C. Finn, and S. Levine. “Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning”. In: *Proceedings of the 3rd Conference on Robotic Learning*. 2021.
- [82] X. Zhai et al. *A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark*. 2020. arXiv: [1910.04867](https://arxiv.org/abs/1910.04867) [cs.CV].
- [83] Q.-s. Zhang and S.-C. Zhu. “Visual interpretability for deep learning: a survey”. In: *Frontiers of Information Technology & Electronic Engineering* 19.1 (2018), pp. 27–39.
- [84] B. Zieliński, A. Plichta, K. Misztal, P. Spurek, M. Brzychczy-Włoch, and D. Ochońska. “Deep learning approach to bacterial colony classification”. In: *PLOS ONE* 12.9 (2017), pp. 1–14.

A Datasets and meta-data

Meta-Album datasets, their original sources and how the datasets are curated are explained separately in the following sections. All these datasets come from 10 domains:

- Large Animals
- Small Animals
- Plants
- Plant Diseases
- Microscopy
- Remote Sensing
- Vehicles
- Manufacturing
- Human Actions
- Optical Character Recognition

We have 3 versions of the datasets:

1. **Micro**: a minimal version with 20 randomly selected classes and 40 images per class;
2. **Mini**: a medium version with all classes having at least 40 images per class, including 40 randomly selected images per class;
3. **Extended**: a full version that consists of all classes and all images per class.

Sample images from Meta-Album are provided in Figure 4. The formatted Meta-Album datasets are referred to as “preprocessed” versions. Such preprocessed images are used in all Meta-Album versions: Micro, Mini, or Extended.

A.1 Large Animals (*LR_AM*)

BRD

When Meta-Album was created, the **Birds dataset** [50] contained images of 315 bird species, but now it has increased the number of species to 450. It has more than 49 000 images, each with a resolution of 224x224 px. All the images have their natural background, which can lead to bias since, for example, some birds are frequently found in water backgrounds. Additionally, the dataset is imbalanced regarding the ratio of male species images to female species images. The preprocessed version distributed in Meta-Album is made from the original dataset by resizing all the images to a resolution of 128x128 px using an anti-aliasing filter.

DOG

Researchers from Stanford University created the original **Dogs dataset** [27]. It contains more than 20 000 images belonging to 120 breeds of dogs worldwide. The images and annotations came from ImageNet for the task of fine-grained image categorization. The number of images per class and the resolution is not balanced. Each class can have 148 to 252 images with a resolution from 100x105 to 2 448x3 264 px. This dataset has a little inter-class variation and a large intra-class variation due to color, pose, and occlusion. Most of the images in this dataset are taken in man-made environments leading to a significant background variation. The preprocessed version of this dataset is prepared from the original dataset by cropping the images from either side to make squared images. In case an image has a resolution lower than 128 px, the squared images are done by either duplicating the top and bottom-most 3 rows or the left and right most 3 columns based on the orientation of the original image. These square images are then resized into 128x128 px using an anti-aliasing filter.

fig/sample/BRD.png	fig/sample/DOG.png	fig/sample/AWA.png	fig/sample/PLK.png	fig/sample/INS_2.png
LR_AM.BRD	LR_AM.DOG	LR_AM.AWA	SM_AM.PLK	SM_AM.INS_2
fig/sample/INS.png	fig/sample/FLW.png	fig/sample/PLT.png	fig/sample/FNG.png	fig/sample/PLT_VIL.png
SM_AM.INS	PLT.FLW	PLT.PLT_NET	PLT.FNG	PLT_DIS.PLT_VIL
fig/sample/MED.png	fig/sample/PLT_DOC.png	fig/sample/BCT.png	fig/sample/PNU.png	fig/sample/PRT.png
PLT_DIS.MED_LF	PLT_DIS.PLT_DOC	MCR.BCT	MCR.PNU	MCR.PRT
fig/sample/RESISC.png	fig/sample/RSICB.png	fig/sample/RSD.png	fig/sample/CRS.png	fig/sample/APL.png
REM_SEN.RESISC	REM_SEN.RSICB	REM_SEN.RSD	VCL.CRS	VCL.APL
fig/sample/BTS.png	fig/sample/TEX.png	fig/sample/TEX_DTD.png	fig/sample/TEX_ALOT.png	fig/sample/SPT.png
VCL.BTS	MNF.TEX	MNF.TEX_DTD	MNF.TEX_ALOT	HUM_ACT.SPT
fig/sample/ACT_40.png	fig/sample/ACT_410.png	fig/sample/MD_MIX.png	fig/sample/MD_5_BIS.png	fig/sample/MD_6.png
HUM_ACT.ACT_40	HUM_ACT.ACT_410	OCR.MD_MIX	OCR.MD_5_BIS	OCR.MD_6

Figure 4: Sample images from Meta-Album datasets.

AWA

The original **Animals with Attributes 2** (AWA) dataset [78] was designed to benchmark transfer-learning algorithms, in particular attribute base classification and zero-shot learning. It has more

than 37 000 images from 50 animals, where each animal corresponds to a class. The images of this dataset were collected from public sources, such as Flickr, in 2016, considering only images licensed for free use and redistribution. Each class can have 100 to 1 645 images with a resolution from 100x100 to 1 893x1 920 px. To preprocess this dataset, we cropped the images from either side to make them square. In case an image has a resolution lower than 128 px, the squared images are done by either duplicating the top and bottom-most 3 rows or the left and right most 3 columns based on the orientation of the original image. Lastly, the square images are resized into 128x128 px using an anti-aliasing filter.

A.2 Small Animals (*SM_AM*)

PLK

The Plankton dataset is created by researchers at the [Woods Hole Oceanographic Institution](#). Imaging FlowCytobot (IFCB) was used for the data collection. The Complete process and mechanism are described in the paper [62]. Each image in the dataset contains one or multiple planktons. The images are captured in a controlled environment and have different orientations based on the flow of the fluid in which the images are captured and the size and shape of the planktons. The preprocessed plankton dataset is prepared from the original WHOI Plankton dataset. The preprocessing of the images is done by creating a background squared image by either duplicating the top and bottom-most 3 rows or the left and right most 3 columns based on the orientation of the original image to match the width or height of the image respectively. A Gaussian kernel of size 29x29 is applied to the background image to blur the image. Finally, the original plankton image is pasted on the background image at the center of the image. The squared background image with the original plankton image on top of it as one image is then resized into 128x128 with anti-aliasing.

INS_2

The pest insects dataset [76] was originally created as a large scale benchmark dataset for [Insect Pest Recognition](#). It contains more than 75 000 images belongs to 102 categories. It also has a hierarchical taxonomy and the insect pests which mainly affect one specific agricultural product are grouped into the same upper-level category. The preprocessed version is made from the original dataset by cropping the images in perfect squares and then resizing them into the required images size of 128x128.

INS

The original Insects dataset [59] is created by the [National Museum of Natural History, Paris](#). It has more than 290 000 images in different sizes and orientations. The dataset has hierarchical classes which are listed from top to bottom as Order, Super-Family, Family, and Texa. Each image contains an insect in its natural environment or habitat, *i.e.*, either on a flower or near to vegetation. The images are collected by the researchers and hundreds of volunteers from [SPIPOLL Science project](#). The images are uploaded to a centralized server either by using the [SPIPOLL website](#), [Android application](#) or [IOS application](#). The preprocessed insect dataset is prepared from the original Insects dataset by carefully preprocessing the images, *i.e.*, cropping the images from either side to make squared images. These cropped images are then resized into 128x128 using Open-CV with an anti-aliasing filter.

A.3 Plants (*PLT*)

FLW

The [Flowers dataset](#) [44] consists of a variety of flowers gathered from different websites and some are photographed by the original creators. These flowers are commonly found in the UK. The images generally have large scale, pose and light variations. Some categories of flowers in the dataset has large variations of flowers while other have similar flowers in a category. The dataset was created back in 2008 at Oxford University by Nilsback, M-E. and Zisserman, A.[44]. The Flowers dataset in the Meta-Album meta-dataset is a preprocessed version of the original flowers dataset. The images are first cropped and made into squared images which are then resized into 128x128 with anti-aliasing filter.

PLT_NET

Meta-Album PlantNet dataset is created by sampling the [PI@ntNet-300k dataset](#), itself a sampling of the [PI@ntNet Project](#)'s repository. The images and labels which enter this database are sourced by citizen botanists from around the world, then confirmed using a weighted reliability score from others users, such that each image has been reviewed by 2.03 citizen botanists on average. Of the 1 081 classes in the original PI@ntNet-300k dataset, PLT_NET retains the 25 most populous classes, belonging to 21 genera, for a total of 120 688 images total, with min 2 914, max 9 011 image distribution per class. Each image contains a colored 128x128 image of a plant or a piece of a plant from the corresponding class (or in some cases sketches of plants or plant cells on microscope slides), scaled from the initial variable width using the INTER_AREA anti-aliasing filter from [Open-CV](#) [3]. Almost all images were initially square; cropping by taking the largest possible square with center at the middle of the initial image was applied otherwise.

FNG

Meta-Album Fungi dataset is created by sampling the [Danish Fungi 2020 dataset](#), itself a sampling of the [Atlas of Danish Fungi](#)'s repository. The images and labels which enter this database are sourced by a group consisting of 3 300 citizen botanists, then verified by their peers using a ranking of each person's reliability, then finally verified by experts working at the Atlas. Of the 128 classes in the original Danish Fungi 2020 dataset, FNG retains the 25 most populous classes, belonging to six genera, for a total of 15 122 images total, with min 372, and max 1 221 images per class. Each image contains a colored 128x128 image of a fungus or a piece of a fungus from the corresponding class. Because the initial data were of widely varying sizes, we needed to crop a significant portion of the images, which we implemented by taking the largest possible square with center at the middle of the initial image. We then scaled each squared image to the 128x128 standard using the INTER_AREA anti-aliasing filter from [Open-CV](#) [3].

A.4 Plant Diseases (*PLT_DIS*)

PLT_VIL

The [Plant Village dataset](#) [23, 46] contains camera photos of 17 crop leaves. The original image resolution is 256x256 px. This collection covers 26 plant diseases and 12 healthy plants. The leaves are removed from the plant and placed on gray or black background, in various lighting conditions. All images are captured on a variety of gray backgrounds, except Corn Common rust which has a black background. For the curated version, we exclude the irrelevant Background and Corn Common Rust classes from the original collection. Plant Village has a 2-level label hierarchy, the supercategory is the crop type and the category is the disease type. We have preprocessed Plant Village for Meta-Album by resizing a subset from the original dataset to 128x128 image size.

MED_LF

The [Medicinal Leaf Database](#) [55] gathers 30 species of healthy and mature medicinal herbs. The leaves are plucked from different plants of the same species, then placed on a white uniform background. There are around 1 800 images in total, captured with a mobile phone camera. The original resolution is 1 600x1 200 px. We create Medleaf for Meta-Album by cropping them at the center and resize to 128x128 px.

PLT_DOC

The [PlantDoc dataset](#) [60] is made up of images of leaves of healthy and unhealthy plants. The images were downloaded from Google Images and Ecosia, and later cropped by the authors, so generally, one complete leaf fits in one image. The original, uncropped images are generally different in scale, light conditions, and pose. However, within one category, images of leaves that came from the same original image can be found. The images correspond to 27 classes, including plant disease names and plant species names, *e.g.*: Corn Leaf Blight and Cherry Leaf respectively. The dataset was created for a benchmarking classification model work, published in 2020 by Singh et al. The PlantDoc dataset in the Meta-Album benchmark is extracted from a preprocessed version of the original PlantDoc dataset. First, to get *i.i.d.* samples, only one leaf image per each original image is randomly picked. Then,

leaves images are cropped and made into squared images which are then resized into 128x128 with anti-aliasing filter.

A.5 Microscopy (*MCR*)

BCT

The **Digital Images of Bacteria Species dataset (DIBaS)** [84] is a dataset of 33 bacterial species with around 20 images for each species. For the Meta-Album, since the images were large (2 048x1 532) with very few samples in each class, we decided to split each image into several smaller images before resizing them to 128x128. We then obtained a preprocessed dataset of 4 060 images with at least 108 images for each class. This dataset was also preprocessed with blob normalization techniques, which is quite unusual for this type of image. The goal of this transformation was to reduce the importance of color in decision-making for a bias-aware challenge.

PRT

This dataset is a subset of the Subcellular dataset [66] in the **Protein Atlas project**. The original dataset, which stems from the **Human Protein Atlas Image Classification Kaggle competition**, comprises 31 072 RGBY images of size 512x512 px, each of which belongs to one or more out of 28 classes. The labels correspond to protein organelle localizations. For Meta-Album, we performed two modifications: (1), to turn the dataset into a multi-class dataset, we dropped all images belonging to more than a single class and also those images that belong to classes with less than 40 members; (2) we converted the remaining images into RGB simply by dropping the yellow channel; this was also a common practice in the competition. Finally, and as for all datasets in Meta-Album, the images from the original dataset were resized to 128x128 image size.

PNU

The **PanNuke dataset** [16, 17] is a semi-automatically generated segmentation and classification task of nuclei. The dataset contains 7 753 images of 19 different tissue types. For the Meta-Album meta-dataset, even though this dataset was designed as a segmentation task, we were able to transform it into a tissue classification task since we had the tissue type for each sample in the dataset. We also resized the images to 128x128 pixels and applied stain normalization to avoid bias and remove some spurious features.

A.6 Remote Sensing (*REM_SEN*)

RESISC

RESISC45 dataset [8] gathers 700 RGB images of size 256x256 px for each of 45 scene categories. The data authors strive to provide a challenging dataset by increasing both within-class diversity and between-class similarity, as well as integrating many image variations. Even though RESISC45 does not propose a label hierarchy, it can be created from other common aerial image label organization scheme. We have preprocessed RESISC for Meta-Album by resizing the dataset to 128x128 px.

RSICB

RSICB128 dataset [35] covers 45 scene categories, assembling in total 36 000 images of resolution 128x128 px. The data authors select various locations around the world, and follow China's land-use classification standard. This collection has 2-level label hierarchy with 6 super-categories: agricultural land, construction land and facilities, transportation and facilities, water and water conservancy facilities, woodland, and other lands. The preprocessed version of RSICB is created by resizing the images into 128x128 px using an anti-aliasing filter.

RSD

RSD46 dataset [41, 79] is collected from Google Earth and Tianditu. The collection contains 46 scene categories, with a total of 117 000 images. Each scene category has between 500-3 000 images. The original resolution are 256x256 px or 512x512 px. We have created preprocessed version of RSD for Meta-Album by resizing the original dataset to 128x128 px.

A.7 Vehicles (*VCL*)

APL

The original [Airplanes dataset](#) [77] comprises more than 9 000 remote sensing images acquired from Google Earth satellite imagery, including 21 different types of aircraft from around 36 airports. All the images were carefully labeled by seven specialists in the field of remote sensing image interpretation. Each class can have 230 to 846 images, where each image contains only one complete aircraft, and they have variable resolutions. To preprocess this dataset, we cropped the images from either side to make them square, and then we resized them into 128x128 px using an anti-aliasing filter.

CRS

The original [Cars dataset](#) [29] was collected in 2013, and it contains more than 16 000 images from 196 classes of cars. Most images are on the road, but some have different backgrounds, and each image has only one car. Each class can have 48 to 136 images of variable resolutions. The preprocess version for this dataset was obtained by creating square images either duplicating the top and bottom-most 3 rows or the left and right most 3 columns based on the orientation of the original image. In this case, cropping was not applied to create the square images since following this technique results in losing too much information from the cars. Then, the square images were resized into 128x128 px using an anti-aliasing filter.

BTS

The original version of the Meta-Album boats dataset is called [MARVEL dataset](#) [19]. It has more than 138 000 images of 26 different maritime vessels in their natural background. Each class can have 1 802 to 8 930 images of variable resolutions. To preprocess this dataset, we either duplicate the top and bottom-most 3 rows or the left and right most 3 columns based on the orientation of the original image to create square images. No cropping was applied because the boats occupy most of the image, and applying this technique will lead to incomplete images. Finally, the square images were resized into 128x128 px using an anti-aliasing filter.

A.8 Manufacturing (*MNF*)

TEX

The original Textures dataset is a combination of 4 texture datasets: [KTH-TIPS and KTH-TIPS 2](#) [15, 42], [Kylberg Textures Dataset](#) [31] and [UIUC Textures Dataset](#) [33]. The data in all four datasets is collected in laboratory conditions, *i.e.*, images were captured in a controlled environment with configurable brightness, luminosity, scale and angle. The *KTH-TIPS* dataset was collected by Mario Fritz and *KTH-TIPS 2* dataset was collected by P. Mallikarjuna and Alireza Tavakoli Targhi, created in 2004 and 2006 respectively. Both of these datasets were prepared under the supervision of Eric Hayman and Barbara Caputo. The data for *Kylberg Textures Dataset* and *UIUC Textures Dataset* data was collected by the original authors of these datasets in September 2010 and August 2005 respectively.

The Meta-Album Textures dataset is a preprocessed version of the original dataset (combination of 4 datasets). All the images are preprocessed by first cropping into perfect squared images and then resized into 128x128 with an anti-aliasing filter.

TEX_DTD

The [Textures DTD dataset](#) is a large textures dataset which consists of 5 640 images. The data is collected from [Google](#) and [Flicker](#) by the original authors of the paper “Describing Textures in the Wild” [9]. The data was annotated using [Amazon Mechanical Turk](#). The data collection process is mentioned on the [dataset overview page](#) For Meta-Album meta-dataset, this dataset is preprocessed by cropping the images to square images and then resizing them to 128x128 using Open-CV with an anti-aliasing filter. This dataset has 47 class labels.

TEX_ALOT

Textures ALOT dataset [5] consists of 27 500 images from 250 categories. The images in the dataset are captured in controlled environment by the creators of the dataset. The images have different viewing angle, illumination angle, and illumination color for each material of texture.

A preprocessed version of Textures-ALOT is used in the Meta-Album meta-dataset. The images are first cropped into square images and then resized to 128x128 with anti-aliasing filter.

A.9 Human Actions (*HUM_ACT*)

SPT

The **100-Sports dataset** [49] is a collection of sports images covering 73 different sports. Images are 224x224x3 in size and in .jpg format. Images were gathered from internet searches. The images were scanned with a duplicate image detector program and all duplicate images were removed. For Meta-Album, the dataset is preprocessed and images are resized into 128x128 pixels using Open-CV [3]

ACT_40

The **Stanford 40 Actions dataset** [80] contains images of humans performing 40 actions. There are 9 532 images in total with 180-300 images per action class. The dataset is designed for understanding human actions in still images. For Meta-Album, the dataset is preprocessed and images are resized into 128x128 pixels using Open-CV [3].

ACT_410

The **MPII Human Pose dataset** [1] is a state of the art benchmark for evaluation of articulated human pose estimation. It includes around 25 000 images containing over 40 000 people with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities. Overall the dataset covers 410 human activities and each image is provided with an activity label. Each image was extracted from a YouTube video. Like other Meta-Album datasets, this dataset is preprocessed and all images are resized into 128x128 pixels.

A.10 Optical Character Recognition (*OCR*)

MD_MIX

OmniPrint-MD-mix dataset consists of 28 240 images (128x128, RGB) from 706 categories. The images are synthesized with OmniPrint [64], and no further processing was done. The OmniPrint synthesis parameters are stated as follows: font size is 192, image size is 128, the strength of random perspective transformation is 0.04, left/right/top/bottom margins are all 20% of the image size, the strength of pre-rasterization elastic transformation is 0.035, random translation is activated both horizontally and vertically, rotation is within -60 and 60 degrees, horizontal shear is within -0.5 and 0.5 , brightness is within 0.8333 and 1.2, contrast is within 0.8333 and 1.2, color enhancement is within 0.8333 and 1.2. The other parameters vary between images. We designed 20 settings, each setting is used to synthesize 2 images. The 20 settings are described below:

1. plain white background, random color foreground, trivial image blending (pasting)
2. plain white background, random color foreground, trivial image blending (pasting), random color outline
3. plain white background, random color foreground, trivial image blending (pasting), morphological gradient operation with elliptical kernel (kernel size is 5)
4. plain white background, image/texture foreground, trivial image blending (pasting)
5. plain white background, image/texture foreground, image/texture outline, trivial image blending (pasting)
6. random color background, trivial image blending (pasting), random color foreground

7. random color background, trivial image blending (pasting), random color foreground, random color outline
8. random color background, trivial image blending (pasting), random color foreground, morphological gradient operation with elliptical kernel (kernel size is 5)
9. random color background, trivial image blending (pasting), image/texture foreground
10. random color background augmented with random polygons, trivial image blending (pasting), random color foreground
11. image/texture background, Poisson image blending [47], random color foreground
12. image/texture background, Poisson image blending [47], random color foreground, random color outline
13. image/texture background, Poisson image blending [47], random color foreground, morphological gradient operation with elliptical kernel (kernel size is 5)
14. image/texture background, Poisson image blending [47], image/texture foreground
15. image/texture background, Poisson image blending [47], image/texture foreground, morphological gradient operation with elliptical kernel (kernel size is 5)
16. image/texture background, Poisson image blending [47], image/texture foreground, random color outline
17. image/texture background, Poisson image blending [47], image/texture foreground, random color outline, morphological gradient operation with elliptical kernel (kernel size is 5)
18. image/texture background, Poisson image blending [47], image/texture foreground, image/texture outline
19. image/texture background, Poisson image blending [47], image/texture foreground, image/texture outline, morphological gradient operation with elliptical kernel (kernel size is 5)
20. image/texture background, Poisson image blending [47], image/texture foreground, image/texture outline, outline size is 10

All images/textures consists of photos taken by a personal mobile phone [64].

The 706 categories are characters from:

- Armenian
 - lowercase letters
- ASCII digits
- Balinese
 - consonants
 - digits
- Basic Latin
 - lowercase letters
- Devanagari
 - digits
- Georgian
- Gujarati
 - consonants
- Hebrew
- Hiragana
- khmer
 - consonants
- Mongolian

- basic letters
- Myanmar
 - digits
- N Ko
 - letters
 - digits
- Oriya
 - consonants
- Russian
- Sinhala
 - independent vowels
- Tamil
 - consonants
 - digits
- Telugu
 - digits
- Thai
 - consonants
- Tibetan
 - digits

MD_5_BIS

OmniPrint-MD-5-bis dataset consists of 28 240 images (128x128, RGB) from 706 categories. The images are synthesized with OmniPrint [64], and no further processing was done. The OmniPrint synthesis parameters are stated as follows: font size is 192, image size is 128, the strength of random perspective transformation is 0.04, left/right/top/bottom margins are all 20% of the image size, the strength of pre-rasterization elastic transformation is 0.035, random translation is activated both horizontally and vertically, image blending method is Poisson Image Editing [47], rotation is within -60 and 60 degrees, horizontal shear is within -0.5 and 0.5 , the foreground is filled with a random color, the background consists of images downloaded from Pexels.

The 706 categories are characters from:

- Armenian
 - lowercase letters
- ASCII digits
- Balinese
 - consonants
 - digits
- Basic Latin
 - lowercase letters
- Devanagari
 - digits
- Georgian
- Gujarati
 - consonants
- Hebrew
- Hiragana

- khmer
 - consonants
- Mongolian
 - basic letters
- Myanmar
 - digits
- N Ko
 - letters
 - digits
- Oriya
 - consonants
- Russian
- Sinhala
 - independent vowels
- Tamil
 - consonants
 - digits
- Telugu
 - digits
- Thai
 - consonants
- Tibetan
 - digits

MD_6

OmniPrint-MD-6 dataset consists of 28 120 images (128x128, RGB) from 703 categories. The images are synthesized with OmniPrint [64], no further processing was done. The OmniPrint synthesis parameters are stated as follows: font size is 192, image size is 128, the strength of random perspective transformation is 0.04, left/right/top/bottom margins are all 20% of the image size, the strength of pre-rasterization elastic transformation is 0.035, random translation is activated both horizontally and vertically, image blending method is Poisson Image Editing [47], rotation is within -60 and 60 degrees, horizontal shear is within -0.5 and 0.5 , both foreground and background are images taken from a personal mobile phone [64].

The 703 categories are characters from:

- Arabic
- Armenian
 - uppercase letters
 - lowercase letters
- Balinese
 - independent vowels
- Basic Latin
 - uppercase letters
- Bengali
 - consonants
 - digits
 - independent vowels

- punctuation symbols
- Devanagari
 - consonants
 - independent vowels
- Ethiopic
 - digits
- Greek
- Gujarati
 - digits
 - independent vowels
- Katakana
- Khmer
 - digits
 - independent vowels
- Lao
 - consonants
 - digits
- Mongolian
 - digits
- Myanmar
 - consonants
 - independent vowels
- Oriya
 - digits
 - independent vowels
- Sinhala
 - astrological digits
 - consonants
- Tamil
 - independent vowels
- Telugu
 - consonants
 - independent vowels
- Thai
 - digits
- Tibetan
 - consonants

B License information of Meta-Album datasets

In this section, we provide details about the licenses of all Meta-Album datasets.

We researched the ownership of all datasets to determine the type of usage permission. We contacted original owners by email to clarify permissions if needed, when no explicit statement of license or permission was found.

Table 4: Meta-Album: Datasets license information

Domain Name	Set #	Dataset ID	Original source
Large Animals	0	<i>BRD</i>	Birds 400 [50]
	1	<i>DOG</i>	Stanford Dogs [27]
	2	<i>AWA</i>	AWA [78]
Small Animals	0	<i>PLK</i>	WHOI [62]
	1	<i>INS_2</i>	Pest Insects [76]
	2	<i>INS</i>	SPIOLL [59]
Plants	0	<i>FLW</i>	Flowers [44]
	1	<i>PLT_NET</i>	PlantNet [18]
	2	<i>FNG</i>	Danish Fungi [48]
Plant Diseases	0	<i>PLT_VIL</i>	PlantVillage [23, 46]
	1	<i>MED_LF</i>	Medicinal Leaf [55]
	2	<i>PLT_DOC</i>	Plant Doc [60]
Microscopy	0	<i>BCT</i>	DiBas [84]
	1	<i>PNU</i>	PanNuke [16, 17]
	2	<i>PRT</i>	Protein Atlas [66]
Remote Sensing	0	<i>RESISC</i>	RESISC45 [8]
	1	<i>RSICB</i>	RSICB128 [35]
	2	<i>RSD</i>	RSD46 [79, 41]
Vehicles	0	<i>CRS</i>	Cars [29]
	1	<i>APL</i>	Multi-type Aircraft [77]
	2	<i>BTS</i>	MARVEL [19]
Manufacturing	0	<i>TEX</i>	KTH-TIPS [15, 42] Kylberg [31] UIUC [33]
	1	<i>TEX_DTD</i>	Texture DTD [9]
	2	<i>TEX_ALOT</i>	Texture ALOT [5]
Human Actions	0	<i>SPT</i>	100 Sports [49]
	1	<i>ACT_40</i>	Stanford 40 Actions [80]
	2	<i>ACT_410</i>	MPII Human Pose [1]
Optical Char. Recog.	0	<i>MD_MIX</i>	
	1	<i>MD_5_BIS</i>	OmniPrint [64]
	2	<i>MD_6</i>	

C Data preparation

Data preparation process is done in many steps which include, depending on the dataset:

- Visual inspection and data cleaning
- Cropping images to a square around the region of interest (eventually background padding)
- Resizing squared images to 128x128 size (with anti-aliasing filter)
- Sub-sampling (to obtain the Mini and Micro versions).

C.1 Visual inspection and data cleaning

The goal of this step is to remove non-suitable images.

Most Meta-Album datasets were downloaded from the Internet; a few were directly obtained from their original owners/creators. Cleaning steps included the removal of duplicate images, corrupted images, and images that included watermarks or identifiers. After cleaning, each dataset is saved in a folder including a sub-folder with the images, in a standard jpg format, and a meta-data file including file names, categories/classes/labels, super-categories, etc.

C.2 Image cropping

The goal of the step is to obtain square images.

Most of the datasets come with images of different sizes and resolutions. Usually, the object of interest (*e.g.*, bird, leaf, car, etc.) is already mostly centered in the middle of the image, with comfortable margins on all sides. In this case, the image is cropped to obtain a square by keeping the smallest dimension (width or height) and eliminating symmetrically spurious margins in the largest dimension.

Some datasets, *e.g.*, Plankton and Boats, are preprocessed in a different way because the rectangular images tightly fit the object of interest. In that case, the images are padded with extra background or either side in the smallest dimension, to obtain a square of the largest dimension. The padding is obtained from the 3 extreme rows (or columns) on either side, which is smoothed with a Gaussian kernel of size 29x29. We found that this technique minimally introduces image artifacts.

Some of the datasets, *e.g.*, RESISC and Textures datasets, are already square, hence the only preprocessing needed is resizing (see next sub-section). All OCR datasets are generated by the OmniPrint software [64]. The images are exactly in 128x128 size and hence no preprocessing is required at all.

Datasets in Human Actions domain are preprocessed in a different way as they contain humans performing some actions, *e.g.*, playing a sport, etc. For these datasets, a face detection software is used to detect a face in the image and then a reasonable area around the face is cropped in order to make sure that the human body along with the face is cropped. After cropping, like other datasets, the images are resized into 128x128 pixel sizes.

See Figure 5 for sample raw images from some of the datasets.

C.3 Image resizing

The goal of this step is to reduce the image size and obtain images of the same dimension for all datasets.

Except for the OCR datasets (images already in the correct size), all other dataset images are resized to 128x128 using Open-CV [3], with an anti-aliasing filter. This avoids the aliasing effect, often visible in the form of jagged edges, introduced by the loss of information during down-sampling.

The image resolution 128x128 is chosen based on preliminary experiments and analyses done for different resolutions of images on the insect dataset, presenting fine features necessary for recognition to the human eye: 64x64, 128x128 and 256x256. For this experiment (See Figure 6), a ResNet 152V2 architecture from Keras is used. It can be observed that 256x256 gives the highest scores among the observed resolutions but the dataset size also increases by a factor of 2. The images with resolution 128x128 are observed to be clearly recognizable to human eyes.

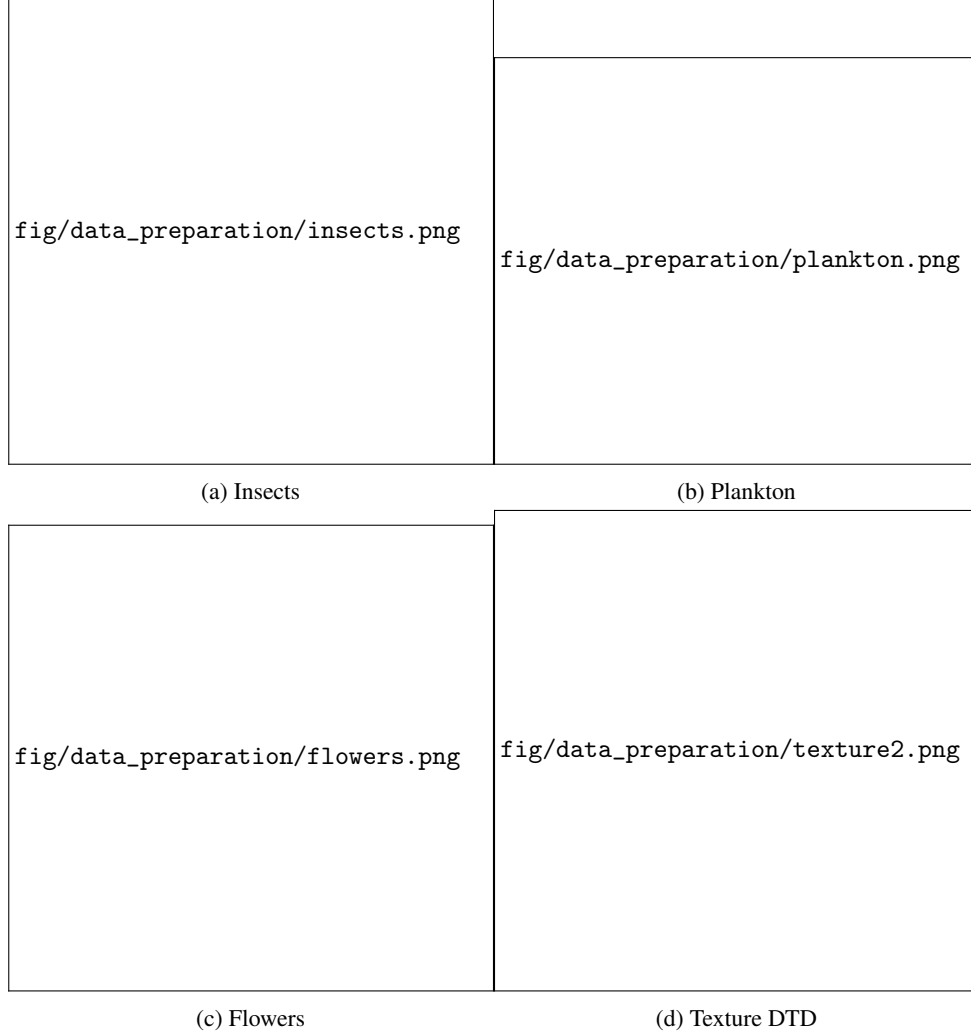


Figure 5: Meta-Album datasets raw sample images.


Another experiment on the insect dataset, comparing 64x64 and 128x128 resolution, is also carried out. The number of images used for the experiment is 180 000, 135 000 of which are used for training, and the remaining for validation. The accuracy plot in Figure 7 shows clearly that 128x128 produces much better results. The experimental settings for this experiment include: ResNet-18 architecture, SGD optimizer, learning rate of 0.001 and 10 epochs. The better quality of 128x128 images is also clearly visually observable (See Figure 8).

C.4 Sub-sampling

The goal of this step is to keep only classes sufficiently populated (more than 40 examples per class) and to balance the dataset (even number of examples per class).


After the previous steps, we obtain what we call the “Extended version” of Meta-Album. The preprocessed datasets in this version conserve the proportions of samples of the original data, with different number of classes and images per class.

To facilitate experimentation and reduce the need for large computational resources, we have prepared the Mini version, including all classes with at least 40 images/class, but only 40 randomly chosen images from each class. We have also prepared the Micro version, including only 20 randomly selected classes from the Mini version, with 40 images per class.



fig/data_preparation/image_resolution_compatison.png

Figure 6: Image resolution comparison (64x64 vs 128x128 vs 256x256).



fig/data_preparation/image_resolution_compatison_accuracy.png

Figure 7: Image resolution comparison (64x64 vs 128x128 vs 256x256).


The software to prepare the datasets, *i.e.*, add backgrounds, crop images, resize images and extract a subset from a dataset, is available in the Meta-Album GitHub Repository (<https://github.com/ihsaan-ullah/meta-album>).

C.5 Quality control

In order to ensure a high quality of datasets, we put a strong emphasis on quality control. We describe two ways to ensure this quality: factsheets and GradCam visualizations. We also define an easy to read Data format for all our datasets.


Data format

In order to generate **Factsheets** for quality control, the data is first formatted in the recommended **Data format**. This makes the data easy-to-read and understand. The data format has a root directory



`fig/data_preparation/image_resolution_compatison_128_A.png`

(a) Sample 128x128 images



`fig/data_preparation/image_resolution_compatison_64_A.png`

(b) Sample 64x64 images

Figure 8: Sample images in 128x128 and 64x64 resolutions.

(ideally with the dataset name and it consists of an *images* folder with all the required images for the fact-sheet generation, a *labels.csv* file with at least two columns, *i.e.*, *FILE_NAME* and *CATEGORY* (category is synonym of class/label) which shows the name of the image instance and the label of that image respectively. One more important file is a *info.json* file in the root directory, which has important information about the dataset and how the data should be read for factsheets. The Data format repository has **sample dataset** for reference. To verify the data format, a **python script** is also provided and complete details of the formatting and testing are given in the README of the data format repository.

Factsheets

For each dataset, we generate a factsheet which is an automatically generated report showing basic information and experimental results on the dataset. These experiments are run on the formatted datasets (Mini version). The experiments are flexible and the configuration can be changed easily by changing the parameters of the software. For each dataset, we train a randomly initialized ResNet-18

architecture [21] on various 5-way 20-shot classification tasks, *i.e.*, the network is presented with training sets consisting of 20 images for each of the 5 classes. The remaining 20 images per class are used as query set to measure the generalization ability of the network. Note that the network is trained “from scratch” (randomly initialized weights), on every task. From the experiments, we construct ROC curves for every task for all classes.

Classes that are too hard to separate (indicating non-learnable tasks) can be identified easily with this experiment. AUC score below 0.7 indicates the difficulty in separation and hence could be studied further to construct a fair task.

In addition to this automatic quality check, we also generate Imagesheets for the datasets, which are PDF files with images of every class. This way, we can also check the data quality visually.

GradCam visualization

In order to detect potential bias stemming from the data and our curating process, we inspect where the baseline model focuses with GradCAM [58]. GradCAM is a post-hoc attention method, showing the contribution of each input element to the output.

By displaying the activation map on top of the original input image, we can observe the impact of each pixel on the prediction output. As such, we can verify whether the model focuses on the relevant subject. In this way, it can also be used to detect whether artifacts are introduced by the image processing in the picture or on the background and other artifacts. For example, we use GradCAM for detecting any artifacts introduced by preprocessing the backgrounds of plankton images to make squared images using padding (Figure 9). Other methods should also be considered to diagnose data [83].

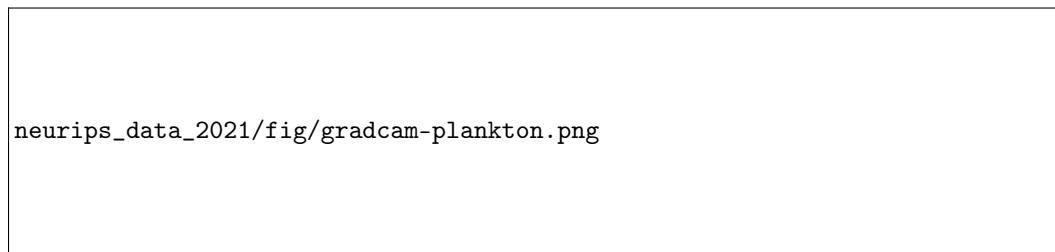


Figure 9: GradCAM activation maps to detect artifacts that could be introduced by image preprocessing.

D Dataset difficulty analysis

In this section, we analyze the difficulty per domain of each set of Meta-Album datasets following the NeurIPS 2021 MetaDL challenge protocol (within domain setting). In Figure 10a, Figure 10b, Figure 10c, and Figure 10d, the top of the blue bar indicates the worst baseline performance (TrainFromScratch). The top of the orange bar indicates the performance of the winners of the NeurIPS 2021 challenge (MetaDelta++) [13]. The top of the green bar indicates the maximum achievable performance. The larger the green bar, the larger the *intrinsic difficulty*. The larger the orange bar, the larger the *modeling difficulty*.

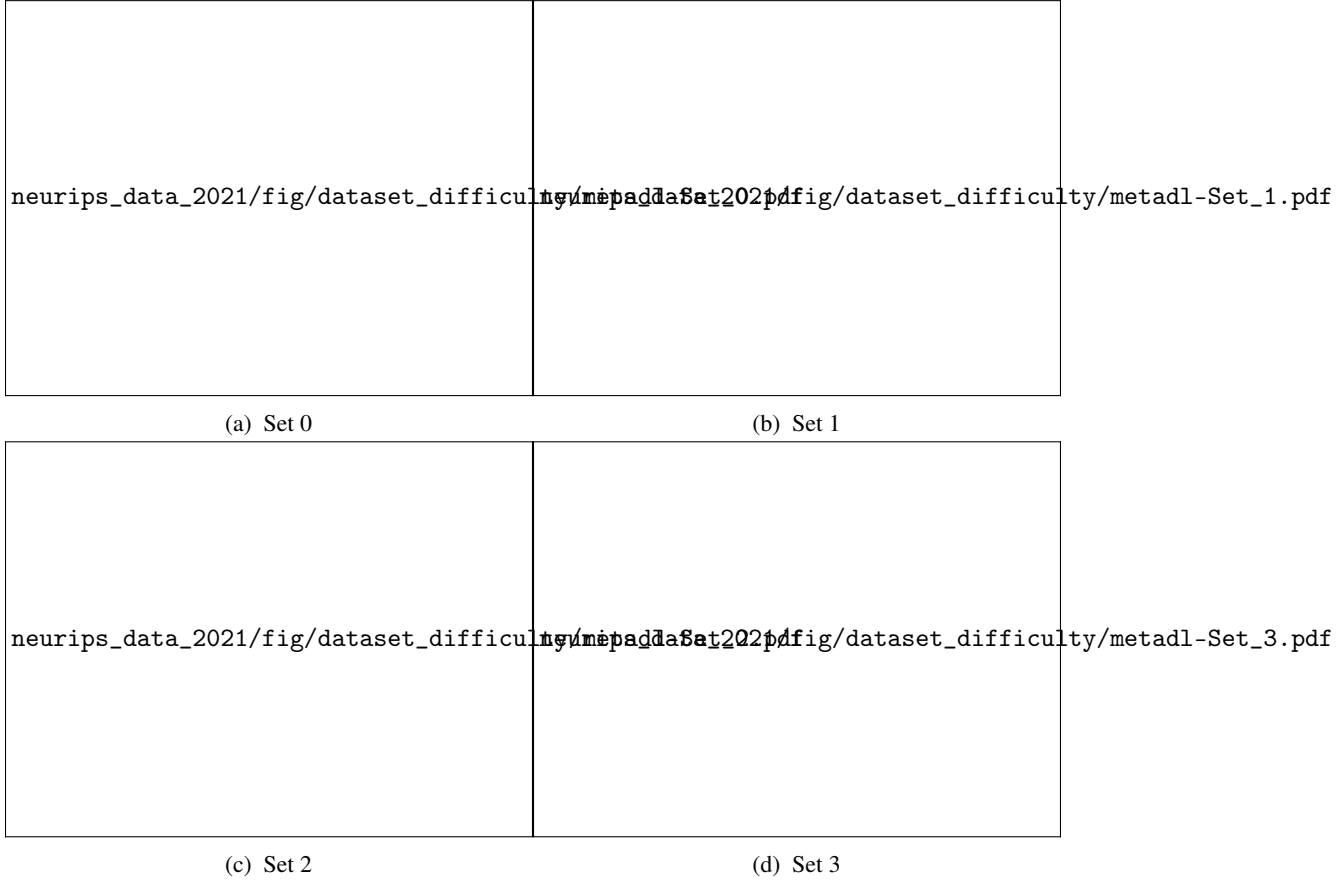


Figure 10: Difficulty analysis for 4 sets (40 datasets) of Meta-Album. Each set consists of 10 datasets from 10 domains (one dataset per domain).

Experimental protocol. Each dataset was evaluated independently by splitting it into meta-training, meta-validation, and meta-testing sets with non-overlapping classes, as explained in Section 3.1. Then, for each dataset, the MetaDelta++ method was meta-trained for 30 minutes with batches of size 64. The trainers produced by MetaDelta++ were meta-validated every 50 batches on 50 5-way 5-shot tasks drawn from the corresponding meta-validation split. On the other hand, as explained in Section 3.2, the TrainFromScratch method has no meta-training and meta-validation phases; instead, it learns each task at meta-test time. Lastly, the learning algorithm with the best meta-validation performance was meta-tested on 100 5-way 5-shot tasks randomly sampled from the meta-testing split. The query set of each meta-test task contained 20 images per class. As described in Section 3.2, we averaged the results over 3 runs with different random seeds.

The experimental results show that in some domains like OCR, there is always a large modeling difficulty, as evidenced by the large orange bar, which makes them desirable to be used in challenges. On the other hand, the microscopy datasets (MCR) in Sets 1 and 2 have a large intrinsic difficulty (large green bar): even the winner’s method does not get good results which may indicate that the

data of these datasets are of insufficient quality. Moreover, some datasets are relatively easy even for the TrainFromScratch method, such as the manufacturing datasets (MNF) of Set 0 or 2. Finally, on average, Set 0 is the easiest while the hardest is Set 2.

E Within Domain Few-shot learning: additional results

In this section, we show additional results for the within domain few-shot learning experiments that were presented in Section 3.2. Table 5, Table 6, Table 7, and Table 8 display the average accuracy per technique and dataset in the [1, 5, 10, 20]-shot settings, respectively. Note that the 1-shot performance of matching networks and prototypical networks is not the same as they use different distance measures. More specifically, matching networks use cosine similarity while prototypical networks use squared euclidean distance.

Table 9 displays the average ranks and running times per technique in different within domain settings. The rankings follow roughly the same pattern as in the right subplot of Figure 11. Note that the confidence intervals for the average ranks are larger than for the average accuracy as the former is computed over the 30 datasets, while the latter is computed at per-task level over all datasets. Figure 11 displays the average rank per method and average accuracy for within domain 5-way [1, 5, 10, 20]-shot classification.

Table 5: Full performance results for 5-way 1-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	31.4 ± 0.37	62.7 ± 0.51	52.7 ± 0.59	67.7 ± 0.53	60.0 ± 0.58
	DOG	24.6 ± 0.27	31.7 ± 0.36	30.2 ± 0.42	33.6 ± 0.40	29.5 ± 0.38
	AWA	26.5 ± 0.31	32.5 ± 0.35	32.3 ± 0.42	31.7 ± 0.36	29.7 ± 0.42
Small animals	PLK	44.5 ± 0.53	60.0 ± 0.57	55.8 ± 0.60	66.7 ± 0.59	59.1 ± 0.58
	INS_2	23.8 ± 0.28	27.4 ± 0.31	26.8 ± 0.36	28.2 ± 0.35	24.7 ± 0.31
	INS	22.6 ± 0.25	34.5 ± 0.40	35.0 ± 0.49	36.1 ± 0.45	31.8 ± 0.51
Plants	FLW	35.4 ± 0.44	59.7 ± 0.51	48.3 ± 0.56	63.0 ± 0.49	55.0 ± 0.57
	PLT_NET	24.8 ± 0.29	31.5 ± 0.35	33.9 ± 0.49	30.5 ± 0.37	35.2 ± 0.41
	FNG	23.8 ± 0.25	23.8 ± 0.25	22.4 ± 0.24	23.7 ± 0.26	22.9 ± 0.26
Plant diseases	PLT_VIL	49.5 ± 0.47	69.0 ± 0.39	50.0 ± 0.46	64.1 ± 0.41	55.6 ± 0.42
	MED_LF	57.8 ± 0.46	63.0 ± 0.42	60.3 ± 0.39	68.4 ± 0.46	66.7 ± 0.43
	PLT_DOC	21.5 ± 0.23	23.1 ± 0.25	22.9 ± 0.26	24.1 ± 0.26	23.8 ± 0.27
Microscopy	BCT	35.3 ± 0.41	65.7 ± 0.45	70.0 ± 0.53	75.5 ± 0.43	58.7 ± 0.68
	PNU	24.2 ± 0.29	23.0 ± 0.24	23.3 ± 0.25	22.9 ± 0.25	24.2 ± 0.27
	PRT	22.0 ± 0.24	25.3 ± 0.27	27.9 ± 0.36	26.0 ± 0.30	28.0 ± 0.37
Remote sensing	RESISC	31.9 ± 0.39	44.0 ± 0.44	37.4 ± 0.46	42.2 ± 0.43	42.1 ± 0.50
	RSICB	47.0 ± 0.47	62.9 ± 0.46	50.2 ± 0.48	64.2 ± 0.46	60.2 ± 0.50
	RSD	33.7 ± 0.39	43.7 ± 0.44	45.1 ± 0.58	49.9 ± 0.46	51.6 ± 0.50
Vehicles	CRS	23.0 ± 0.25	48.9 ± 0.48	52.4 ± 0.61	63.5 ± 0.56	53.1 ± 0.64
	APL	29.9 ± 0.31	36.4 ± 0.39	34.0 ± 0.40	41.2 ± 0.43	33.1 ± 0.37
	BTS	23.6 ± 0.25	26.5 ± 0.28	23.8 ± 0.29	26.2 ± 0.29	23.7 ± 0.28
Manufacturing	TEX	59.8 ± 0.59	81.2 ± 0.46	72.8 ± 0.68	84.1 ± 0.50	84.5 ± 0.51
	TEX_DTD	24.4 ± 0.28	28.6 ± 0.34	26.3 ± 0.29	27.3 ± 0.32	27.6 ± 0.31
	TEX_ALOT	57.0 ± 0.59	89.8 ± 0.38	93.1 ± 0.38	96.0 ± 0.23	93.5 ± 0.35
Human actions	SPT	31.3 ± 0.41	45.4 ± 0.46	42.8 ± 0.57	48.2 ± 0.50	47.4 ± 0.55
	ACT_40	22.2 ± 0.23	24.8 ± 0.25	25.5 ± 0.30	24.7 ± 0.26	25.2 ± 0.30
	ACT_410	30.6 ± 0.36	33.2 ± 0.36	24.9 ± 0.29	27.6 ± 0.30	25.5 ± 0.30
OCR	MD_MIX	20.3 ± 0.19	40.8 ± 0.56	35.4 ± 0.78	89.0 ± 0.37	20.0 ± 0.15
	MD_5_BIS	20.2 ± 0.20	42.6 ± 0.44	69.6 ± 0.66	90.8 ± 0.35	20.2 ± 0.20
	MD_6	21.2 ± 0.22	79.7 ± 0.58	72.3 ± 0.65	92.1 ± 0.31	20.3 ± 0.21

Table 6: Full performance results for 5-way 5-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1\,800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	45.4 ± 0.43	80.5 ± 0.36	72.0 ± 0.41	82.4 ± 0.33	73.7 ± 0.42
	DOG	30.0 ± 0.31	40.1 ± 0.36	42.9 ± 0.39	41.8 ± 0.37	34.9 ± 0.38
	AWA	35.5 ± 0.32	42.6 ± 0.36	37.2 ± 0.33	45.2 ± 0.38	37.2 ± 0.36
Small animals	PLK	57.8 ± 0.50	73.9 ± 0.50	71.0 ± 0.50	76.7 ± 0.44	69.6 ± 0.49
	INS_2	28.3 ± 0.31	34.4 ± 0.32	29.1 ± 0.29	32.2 ± 0.31	31.4 ± 0.32
	INS	25.3 ± 0.25	43.6 ± 0.39	44.7 ± 0.42	45.0 ± 0.43	43.3 ± 0.43
Plants	FLW	50.8 ± 0.46	76.2 ± 0.42	68.0 ± 0.42	79.5 ± 0.33	72.8 ± 0.44
	PLT_NET	31.1 ± 0.30	40.5 ± 0.32	36.7 ± 0.29	41.1 ± 0.33	38.6 ± 0.39
	FNG	27.4 ± 0.26	27.6 ± 0.25	25.1 ± 0.25	29.1 ± 0.26	25.1 ± 0.25
Plant diseases	PLT_VIL	68.6 ± 0.41	85.8 ± 0.25	58.8 ± 0.33	80.9 ± 0.24	70.2 ± 0.35
	MED_LF	77.6 ± 0.35	79.9 ± 0.30	75.3 ± 0.30	83.6 ± 0.24	85.6 ± 0.22
	PLT_DOC	23.4 ± 0.23	27.5 ± 0.25	27.8 ± 0.26	26.3 ± 0.27	26.5 ± 0.26
Microscopy	BCT	51.1 ± 0.42	78.2 ± 0.36	78.6 ± 0.41	82.8 ± 0.34	80.4 ± 0.40
	PNU	28.5 ± 0.27	23.0 ± 0.21	26.5 ± 0.23	26.9 ± 0.24	31.0 ± 0.27
	PRT	25.1 ± 0.25	31.0 ± 0.25	31.5 ± 0.28	30.3 ± 0.28	33.3 ± 0.30
Remote sensing	RESISC	41.4 ± 0.38	57.5 ± 0.36	48.7 ± 0.33	62.7 ± 0.37	61.7 ± 0.35
	RSICB	60.5 ± 0.47	81.9 ± 0.33	77.2 ± 0.29	84.4 ± 0.26	79.9 ± 0.35
	RSD	39.9 ± 0.36	57.1 ± 0.41	56.2 ± 0.43	61.2 ± 0.39	49.4 ± 0.31
Vehicles	CRS	28.3 ± 0.28	65.3 ± 0.42	72.4 ± 0.45	70.2 ± 0.41	71.2 ± 0.48
	APL	40.1 ± 0.37	47.6 ± 0.37	42.7 ± 0.33	55.6 ± 0.38	45.0 ± 0.38
	BTS	27.8 ± 0.28	32.9 ± 0.25	29.2 ± 0.27	33.6 ± 0.27	28.9 ± 0.25
Manufacturing	TEX	71.4 ± 0.57	91.7 ± 0.27	91.5 ± 0.30	93.0 ± 0.28	92.5 ± 0.28
	TEX_DTD	27.2 ± 0.29	38.8 ± 0.34	36.4 ± 0.32	34.6 ± 0.32	35.6 ± 0.30
	TEX_ALOT	69.2 ± 0.57	96.9 ± 0.18	97.1 ± 0.18	98.2 ± 0.12	97.7 ± 0.16
Human actions	SPT	42.3 ± 0.43	60.8 ± 0.42	55.0 ± 0.43	62.0 ± 0.43	55.3 ± 0.44
	ACT_40	25.8 ± 0.24	29.6 ± 0.26	26.5 ± 0.27	32.3 ± 0.26	26.7 ± 0.25
	ACT_410	41.3 ± 0.34	43.6 ± 0.31	30.1 ± 0.27	40.6 ± 0.30	42.1 ± 0.48
OCR	MD_MIX	19.6 ± 0.18	53.1 ± 0.67	80.7 ± 0.42	95.7 ± 0.17	20.0 ± 0.18
	MD_5_BIS	20.3 ± 0.19	56.1 ± 0.49	90.0 ± 0.33	96.6 ± 0.17	20.5 ± 0.20
	MD_6	22.2 ± 0.23	90.2 ± 0.40	91.7 ± 0.27	97.0 ± 0.12	21.3 ± 0.21



Figure 11: **Within domain few-shot learning.** Averages over 30 released Meta-Album datasets of: (a) algorithm rank (smaller is better). The 95% confidence intervals are computed at dataset level over 30 datasets. (b) classification accuracy. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 \times 30 = 54\,000$).

Table 7: Full performance results for 5-way 10-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1\,800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	47.5 ± 0.44	83.2 ± 0.34	78.8 ± 0.36	85.6 ± 0.29	78.7 ± 0.37
	DOG	31.4 ± 0.31	43.3 ± 0.35	46.5 ± 0.36	47.6 ± 0.37	38.8 ± 0.37
	AWA	37.8 ± 0.34	46.5 ± 0.35	40.9 ± 0.31	52.0 ± 0.32	42.2 ± 0.37
Small animals	PLK	59.6 ± 0.50	77.0 ± 0.47	73.4 ± 0.45	78.7 ± 0.42	71.7 ± 0.50
	INS_2	28.2 ± 0.31	37.0 ± 0.32	34.1 ± 0.30	36.1 ± 0.33	35.3 ± 0.33
	INS	26.1 ± 0.26	47.0 ± 0.38	47.4 ± 0.41	52.5 ± 0.40	44.0 ± 0.43
Plants	FLW	53.3 ± 0.48	78.6 ± 0.39	72.3 ± 0.38	82.8 ± 0.30	75.9 ± 0.43
	PLT_NET	32.7 ± 0.31	43.1 ± 0.31	41.3 ± 0.28	47.3 ± 0.28	38.0 ± 0.31
	FNG	27.9 ± 0.25	28.9 ± 0.24	28.7 ± 0.23	31.6 ± 0.25	29.2 ± 0.24
Plant diseases	PLT_VIL	71.3 ± 0.40	87.3 ± 0.24	70.6 ± 0.29	87.6 ± 0.19	76.9 ± 0.30
	MED_LF	81.3 ± 0.33	82.4 ± 0.27	74.9 ± 0.39	81.4 ± 0.23	88.6 ± 0.21
	PLT_DOC	23.7 ± 0.22	29.4 ± 0.25	31.6 ± 0.27	30.6 ± 0.24	27.4 ± 0.25
Microscopy	BCT	53.5 ± 0.41	80.5 ± 0.35	82.4 ± 0.37	83.9 ± 0.32	82.8 ± 0.38
	PNU	29.5 ± 0.27	22.5 ± 0.19	27.9 ± 0.22	29.3 ± 0.23	34.8 ± 0.25
	PRT	25.8 ± 0.26	33.0 ± 0.25	35.0 ± 0.27	33.4 ± 0.28	32.2 ± 0.27
Remote sensing	RESISC	42.4 ± 0.37	61.3 ± 0.36	54.4 ± 0.29	66.4 ± 0.35	68.3 ± 0.35
	RSICB	61.1 ± 0.46	86.9 ± 0.25	85.5 ± 0.26	85.8 ± 0.26	87.6 ± 0.26
	RSD	40.2 ± 0.35	61.2 ± 0.38	53.3 ± 0.57	64.7 ± 0.37	55.0 ± 0.29
Vehicles	CRS	29.9 ± 0.30	69.4 ± 0.40	76.7 ± 0.39	72.4 ± 0.39	74.0 ± 0.43
	APL	41.9 ± 0.39	50.1 ± 0.35	51.5 ± 0.41	55.5 ± 0.31	49.9 ± 0.39
	BTS	29.1 ± 0.29	35.9 ± 0.25	33.1 ± 0.26	38.3 ± 0.26	31.2 ± 0.25
Manufacturing	TEX	71.6 ± 0.58	93.8 ± 0.22	90.9 ± 0.29	93.4 ± 0.26	93.3 ± 0.25
	TEX_DTD	27.2 ± 0.28	42.4 ± 0.32	36.6 ± 0.33	37.5 ± 0.31	40.1 ± 0.30
	TEX_ALOT	70.7 ± 0.57	97.6 ± 0.15	98.5 ± 0.11	98.7 ± 0.09	98.3 ± 0.12
Human actions	SPT	44.0 ± 0.42	64.8 ± 0.39	62.2 ± 0.40	65.7 ± 0.42	62.3 ± 0.46
	ACT_40	26.7 ± 0.24	31.6 ± 0.24	31.3 ± 0.24	35.3 ± 0.25	28.1 ± 0.23
	ACT_410	42.7 ± 0.35	47.3 ± 0.31	31.8 ± 0.31	47.1 ± 0.28	40.8 ± 0.50
OCR	MD_MIX	19.5 ± 0.16	55.6 ± 0.70	92.3 ± 0.24	96.5 ± 0.16	20.3 ± 0.17
	MD_5_BIS	20.2 ± 0.18	60.0 ± 0.49	95.1 ± 0.21	96.9 ± 0.16	21.0 ± 0.21
	MD_6	22.1 ± 0.22	91.8 ± 0.36	94.5 ± 0.19	97.9 ± 0.10	45.5 ± 1.58

Table 8: Full performance results for 5-way 20-shot image classification on all data sets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	47.8 ± 0.47	85.1 ± 0.32	82.2 ± 0.32	86.9 ± 0.27	81.0 ± 0.36
	DOG	31.7 ± 0.31	46.0 ± 0.36	48.1 ± 0.33	50.9 ± 0.36	41.7 ± 0.36
	AWA	37.8 ± 0.33	49.4 ± 0.34	46.7 ± 0.32	53.4 ± 0.33	48.9 ± 0.33
Small animals	PLK	59.9 ± 0.50	79.5 ± 0.44	77.3 ± 0.41	80.2 ± 0.40	75.8 ± 0.48
	INS_2	28.1 ± 0.31	39.3 ± 0.33	37.3 ± 0.32	41.3 ± 0.31	38.9 ± 0.33
	INS	25.8 ± 0.26	49.8 ± 0.39	51.2 ± 0.37	54.4 ± 0.36	47.2 ± 0.43
Plants	FLW	54.2 ± 0.50	80.5 ± 0.36	76.4 ± 0.33	85.7 ± 0.26	79.2 ± 0.35
	PLT_NET	32.6 ± 0.32	44.6 ± 0.37	47.7 ± 0.27	47.0 ± 0.32	41.6 ± 0.30
	FNG	27.8 ± 0.25	30.1 ± 0.24	32.2 ± 0.27	34.2 ± 0.24	31.3 ± 0.23
Plant diseases	PLT_VIL	72.0 ± 0.42	92.9 ± 0.17	80.7 ± 0.24	90.2 ± 0.17	85.3 ± 0.21
	MED_LF	82.6 ± 0.32	83.7 ± 0.30	81.5 ± 0.23	84.4 ± 0.25	93.1 ± 0.16
	PLT_DOC	23.5 ± 0.21	31.9 ± 0.25	31.8 ± 0.21	30.7 ± 0.23	32.2 ± 0.25
Microscopy	BCT	53.6 ± 0.42	80.8 ± 0.35	85.7 ± 0.28	87.0 ± 0.30	82.4 ± 0.40
	PNU	29.7 ± 0.26	23.5 ± 0.20	29.8 ± 0.21	27.2 ± 0.28	38.0 ± 0.26
	PRT	26.1 ± 0.25	34.2 ± 0.26	36.3 ± 0.26	36.8 ± 0.27	32.7 ± 0.25
Remote sensing	RESISC	42.8 ± 0.36	65.7 ± 0.34	64.7 ± 0.31	67.2 ± 0.31	65.0 ± 0.39
	RSICB	61.5 ± 0.49	89.2 ± 0.21	88.9 ± 0.22	90.9 ± 0.19	91.2 ± 0.22
	RSD	40.1 ± 0.35	65.9 ± 0.37	63.3 ± 0.36	67.6 ± 0.35	58.8 ± 0.28
Vehicles	CRS	30.1 ± 0.32	73.1 ± 0.39	76.4 ± 0.37	76.1 ± 0.35	75.8 ± 0.44
	APL	41.5 ± 0.38	52.7 ± 0.32	57.3 ± 0.27	57.9 ± 0.33	52.1 ± 0.32
	BTS	29.1 ± 0.30	36.9 ± 0.26	36.3 ± 0.24	40.1 ± 0.27	32.5 ± 0.24
Manufacturing	TEX	71.6 ± 0.59	95.4 ± 0.19	94.7 ± 0.21	95.1 ± 0.21	94.7 ± 0.22
	TEX_DTD	27.1 ± 0.28	46.2 ± 0.35	39.4 ± 0.31	41.7 ± 0.32	44.1 ± 0.29
	TEX_ALOT	70.4 ± 0.58	98.3 ± 0.11	98.9 ± 0.10	98.8 ± 0.08	98.6 ± 0.11
Human actions	SPT	44.4 ± 0.43	68.4 ± 0.39	66.1 ± 0.37	69.8 ± 0.38	63.5 ± 0.43
	ACT_40	26.8 ± 0.23	32.4 ± 0.23	33.6 ± 0.22	37.1 ± 0.24	29.5 ± 0.22
	ACT_410	42.5 ± 0.36	47.5 ± 0.30	40.7 ± 0.23	50.7 ± 0.29	55.9 ± 0.31
OCR	MD_MIX	19.5 ± 0.15	52.3 ± 0.77	94.6 ± 0.19	96.7 ± 0.15	20.4 ± 0.14
	MD_5_BIS	20.2 ± 0.17	71.8 ± 0.61	96.3 ± 0.17	96.7 ± 0.15	21.1 ± 0.21
	MD_6	21.4 ± 0.19	93.2 ± 0.34	96.5 ± 0.14	98.0 ± 0.09	45.9 ± 1.58

Table 9: The average rankings over all data sets and running times for all techniques on 5-way [1, 5, 10, 20]-shot image classification. The 95% confidence intervals are computed at dataset level over 30 datasets.

	1-shot		5-shot		10-shot		20-shot	
	AR	Time	AR	Time	AR	Time	AR	Time
TrainFromScratch	4.6 ± 0.35	1h29min	4.7 ± 0.28	1h29min	4.8 ± 0.25	1h29min	4.9 ± 0.16	1h29min
Finetuning	2.4 ± 0.39	3h31min	2.5 ± 0.36	3h37min	2.5 ± 0.36	3h41min	2.7 ± 0.36	4h18min
MatchingNet	3.2 ± 0.36	2h50min	3.3 ± 0.46	3h19min	3.1 ± 0.45	3h40min	2.8 ± 0.43	6h52min
ProtoNet	1.8 ± 0.43	2h52min	1.7 ± 0.41	3h15min	1.5 ± 0.29	3h40min	1.5 ± 0.32	6h58min
FO-MAML	2.9 ± 0.48	6h52min	2.9 ± 0.38	9h12min	3.1 ± 0.40	11h24min	3.1 ± 0.41	17h8min

F Cross-Domain Few-shot learning: additional results

Here, we show additional results for the cross-domain few-shot learning experiments from Section 3.2. Table 10, Table 11, Table 12, and Table 13 display the average accuracy per technique and dataset in the 5-way [1, 5, 10, 20]-shot settings. Moreover, Figure 12 displays the average rank per method and average accuracy for the same settings.

Table 14 displays the average accuracy per technique and dataset in the any-way any-shot setting. Table 15 and Table 16 display the results of the any-way any-shot setting but grouped by N -way and k -shot, respectively. Figure 13 shows a performance comparison of the different techniques when trained on each of the evaluated settings and tested on the novel any-way any-shot setting. From these results, we can observe that most of the techniques are not benefited from training with a variable number of shots; only MAML shows a performance improvement by this approach.

Table 17 displays the running times per technique in the different cross-domain settings. This time is separated in meta-training, meta-validation and meta-testing. Note that the meta-training and meta-validation time of TrainFromScratch is 0 because this baseline learns every task starting from a random initialization at meta-test time.

Table 10: Full performance results for cross-domain 5-way 1-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	31.45 ± 0.38	42.93 ± 0.43	37.53 ± 0.48	40.07 ± 0.44	31.59 ± 0.35
	DOG	23.79 ± 0.27	26.38 ± 0.28	25.17 ± 0.29	26.15 ± 0.29	24.41 ± 0.29
	AWA	25.75 ± 0.31	36.69 ± 0.44	32.16 ± 0.43	36.83 ± 0.46	32.12 ± 0.44
Small animals	PLK	42.15 ± 0.49	50.72 ± 0.52	34.28 ± 0.47	44.78 ± 0.52	42.08 ± 0.48
	INS_2	23.81 ± 0.28	29.10 ± 0.32	27.03 ± 0.34	28.22 ± 0.33	25.48 ± 0.31
	INS	23.00 ± 0.25	31.01 ± 0.34	27.07 ± 0.35	30.57 ± 0.34	27.10 ± 0.34
Plants	FLW	38.47 ± 0.48	49.48 ± 0.51	50.23 ± 0.59	53.57 ± 0.57	42.91 ± 0.50
	PLT_NET	24.57 ± 0.28	32.80 ± 0.36	29.88 ± 0.37	30.78 ± 0.35	27.87 ± 0.34
	FNG	22.04 ± 0.24	26.92 ± 0.29	25.93 ± 0.33	27.11 ± 0.31	26.73 ± 0.34
Plant diseases	PLT_VIL	37.49 ± 0.45	52.47 ± 0.51	42.18 ± 0.48	50.03 ± 0.53	44.61 ± 0.45
	MED_LF	54.54 ± 0.56	68.19 ± 0.54	56.02 ± 0.59	64.76 ± 0.54	50.27 ± 0.61
	PLT_DOC	23.61 ± 0.28	30.54 ± 0.32	26.92 ± 0.33	33.77 ± 0.38	27.61 ± 0.34
Microscopy	BCT	32.26 ± 0.41	40.16 ± 0.44	34.64 ± 0.46	37.24 ± 0.43	35.40 ± 0.42
	PNU	22.61 ± 0.26	23.71 ± 0.24	23.01 ± 0.26	23.83 ± 0.25	22.58 ± 0.25
	PRT	21.78 ± 0.24	23.56 ± 0.24	24.36 ± 0.27	25.08 ± 0.28	25.10 ± 0.29
Remote sensing	RESISC	33.61 ± 0.44	42.84 ± 0.45	41.76 ± 0.52	44.37 ± 0.48	38.83 ± 0.47
	RSICB	45.48 ± 0.55	66.90 ± 0.51	59.01 ± 0.62	62.77 ± 0.57	60.44 ± 0.65
	RSD	31.74 ± 0.40	46.82 ± 0.49	37.97 ± 0.47	48.21 ± 0.50	38.64 ± 0.47
Vehicles	CRS	24.03 ± 0.26	27.10 ± 0.31	25.75 ± 0.35	26.70 ± 0.33	23.73 ± 0.26
	APL	30.76 ± 0.36	34.67 ± 0.37	28.29 ± 0.34	32.81 ± 0.36	26.90 ± 0.33
	BTS	22.77 ± 0.26	26.85 ± 0.29	25.30 ± 0.32	25.80 ± 0.28	23.30 ± 0.26
Manufacturing	TEX	53.41 ± 0.65	72.43 ± 0.58	71.29 ± 0.63	70.04 ± 0.65	62.54 ± 0.64
	TEX_DTD	24.37 ± 0.30	33.19 ± 0.37	26.70 ± 0.35	30.25 ± 0.38	26.95 ± 0.35
	TEX_ALOT	55.62 ± 0.59	83.82 ± 0.44	74.34 ± 0.60	84.43 ± 0.44	77.78 ± 0.62
Human actions	SPT	32.08 ± 0.40	38.68 ± 0.39	32.18 ± 0.39	36.02 ± 0.41	33.96 ± 0.39
	ACT_40	23.23 ± 0.26	28.83 ± 0.30	26.31 ± 0.31	27.19 ± 0.30	25.69 ± 0.32
	ACT_410	28.08 ± 0.35	34.61 ± 0.37	31.00 ± 0.40	35.52 ± 0.40	32.05 ± 0.39
OCR	MD_MIX	19.93 ± 0.19	35.68 ± 0.40	19.63 ± 0.20	19.79 ± 0.19	19.68 ± 0.20
	MD_5_BIS	20.46 ± 0.21	26.66 ± 0.28	20.21 ± 0.21	21.35 ± 0.22	20.06 ± 0.21
	MD_6	21.09 ± 0.22	50.13 ± 0.48	20.94 ± 0.23	28.75 ± 0.47	20.52 ± 0.22

Table 11: Full performance results for cross-domain 5-way 5-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1\,800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	45.83 ± 0.43	57.30 ± 0.44	50.10 ± 0.44	59.77 ± 0.41	46.19 ± 0.42
	DOG	28.70 ± 0.29	32.15 ± 0.30	32.16 ± 0.30	32.04 ± 0.30	31.55 ± 0.32
	AWA	32.75 ± 0.37	47.31 ± 0.48	44.43 ± 0.46	49.63 ± 0.48	44.06 ± 0.47
Small animals	PLK	55.01 ± 0.52	62.53 ± 0.49	50.65 ± 0.43	63.35 ± 0.49	53.75 ± 0.54
	INS_2	27.91 ± 0.30	37.71 ± 0.33	34.64 ± 0.34	35.32 ± 0.33	34.73 ± 0.35
	INS	27.15 ± 0.28	40.09 ± 0.38	36.29 ± 0.35	39.90 ± 0.38	35.62 ± 0.36
Plants	FLW	53.29 ± 0.51	64.07 ± 0.53	63.37 ± 0.49	70.72 ± 0.47	60.74 ± 0.49
	PLT_NET	29.74 ± 0.31	42.85 ± 0.37	40.03 ± 0.37	38.92 ± 0.34	40.89 ± 0.36
	FNG	25.41 ± 0.26	32.64 ± 0.32	32.28 ± 0.35	33.56 ± 0.34	32.43 ± 0.36
Plant diseases	PLT_VIL	55.92 ± 0.49	67.84 ± 0.49	58.64 ± 0.43	69.99 ± 0.47	61.85 ± 0.47
	MED_LF	72.79 ± 0.47	81.77 ± 0.42	70.24 ± 0.46	81.89 ± 0.39	76.26 ± 0.45
	PLT_DOC	27.52 ± 0.28	39.41 ± 0.35	35.70 ± 0.34	46.88 ± 0.38	36.01 ± 0.34
Microscopy	BCT	43.48 ± 0.47	49.64 ± 0.45	49.37 ± 0.48	54.46 ± 0.49	46.42 ± 0.44
	PNU	25.12 ± 0.26	25.75 ± 0.23	25.55 ± 0.25	25.34 ± 0.24	25.42 ± 0.26
	PRT	25.71 ± 0.26	26.99 ± 0.26	29.81 ± 0.28	29.89 ± 0.28	29.23 ± 0.29
Remote sensing	RESISC	41.74 ± 0.43	56.74 ± 0.44	54.84 ± 0.44	57.62 ± 0.45	52.88 ± 0.47
	RSICB	57.07 ± 0.51	82.30 ± 0.39	75.38 ± 0.43	77.57 ± 0.42	77.56 ± 0.46
	RSD	39.09 ± 0.40	62.31 ± 0.46	53.84 ± 0.43	67.28 ± 0.46	56.12 ± 0.46
Vehicles	CRS	29.54 ± 0.32	32.82 ± 0.33	31.28 ± 0.35	34.66 ± 0.37	29.70 ± 0.33
	APL	41.97 ± 0.47	48.65 ± 0.42	36.68 ± 0.37	49.66 ± 0.40	39.36 ± 0.39
	BTS	26.62 ± 0.28	33.48 ± 0.31	30.02 ± 0.28	35.56 ± 0.32	30.62 ± 0.28
Manufacturing	TEX	69.12 ± 0.60	81.83 ± 0.50	81.44 ± 0.46	85.84 ± 0.40	78.84 ± 0.52
	TEX_DTD	27.09 ± 0.29	45.20 ± 0.39	36.66 ± 0.36	43.37 ± 0.39	36.16 ± 0.36
	TEX_ALOT	68.29 ± 0.55	93.65 ± 0.26	89.34 ± 0.34	95.36 ± 0.20	89.77 ± 0.39
Human actions	SPT	43.78 ± 0.43	51.71 ± 0.43	45.05 ± 0.41	50.85 ± 0.41	45.40 ± 0.43
	ACT_40	28.33 ± 0.29	35.41 ± 0.33	34.48 ± 0.33	31.62 ± 0.31	33.23 ± 0.33
	ACT_410	38.91 ± 0.40	47.89 ± 0.40	44.65 ± 0.41	51.08 ± 0.40	45.87 ± 0.41
OCR	MD_MIX	19.78 ± 0.18	48.39 ± 0.41	19.65 ± 0.20	20.92 ± 0.21	19.25 ± 0.20
	MD_5_BIS	20.21 ± 0.19	33.26 ± 0.33	21.40 ± 0.21	51.36 ± 0.48	22.12 ± 0.23
	MD_6	22.28 ± 0.22	65.88 ± 0.45	23.84 ± 0.25	52.77 ± 0.57	22.96 ± 0.23

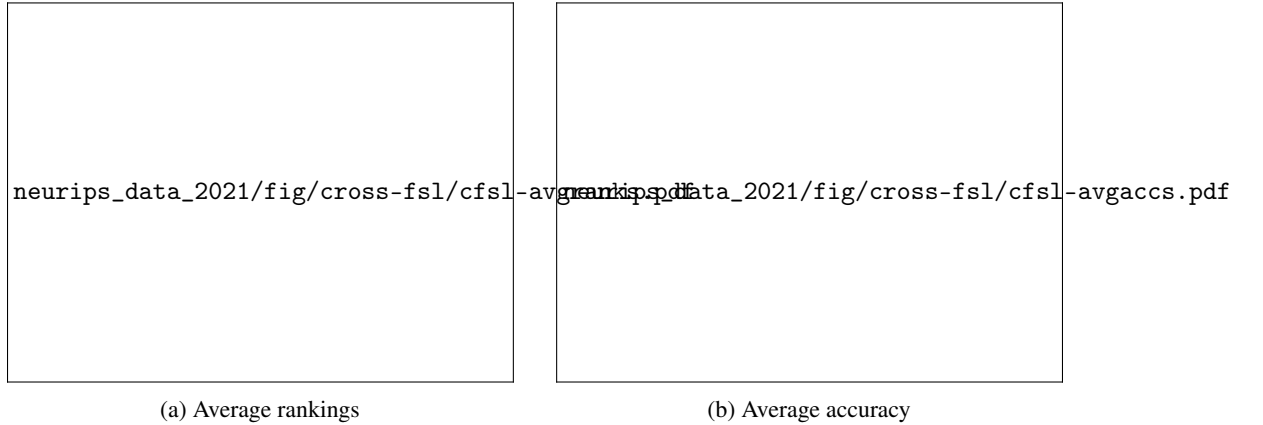


Figure 12: **Cross-domain few-shot learning.** Averages over 30 released Meta-Album datasets of: (a) algorithm rank (smaller is better). The 95% confidence intervals are computed at dataset level over 30 datasets. (b) classification accuracy. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 \times 30 = 54\,000$).

Table 12: Full performance results for cross-domain 5-way 10-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1\,800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	48.48 ± 0.46	61.79 ± 0.43	57.60 ± 0.40	62.38 ± 0.40	51.58 ± 0.40
	DOG	29.52 ± 0.29	34.00 ± 0.30	35.91 ± 0.32	36.10 ± 0.31	34.48 ± 0.32
	AWA	33.74 ± 0.38	50.16 ± 0.49	50.50 ± 0.45	54.60 ± 0.48	47.58 ± 0.48
Small animals	PLK	56.40 ± 0.52	64.48 ± 0.49	57.23 ± 0.43	70.63 ± 0.45	63.18 ± 0.44
	INS_2	28.03 ± 0.30	40.23 ± 0.34	38.74 ± 0.35	40.13 ± 0.35	38.89 ± 0.36
	INS	27.76 ± 0.30	41.61 ± 0.39	41.40 ± 0.35	45.36 ± 0.38	39.95 ± 0.38
Plants	FLW	55.70 ± 0.52	67.72 ± 0.50	68.24 ± 0.45	71.31 ± 0.46	66.97 ± 0.48
	PLT_NET	31.02 ± 0.32	45.71 ± 0.36	45.40 ± 0.37	43.16 ± 0.33	45.49 ± 0.35
	FNG	26.50 ± 0.27	34.09 ± 0.33	35.66 ± 0.35	37.73 ± 0.33	34.33 ± 0.33
Plant diseases	PLT_VIL	59.48 ± 0.51	70.68 ± 0.48	66.44 ± 0.44	74.39 ± 0.43	70.83 ± 0.43
	MED_LF	76.44 ± 0.47	83.83 ± 0.41	71.73 ± 0.40	86.76 ± 0.34	80.80 ± 0.40
	PLT_DOC	27.95 ± 0.29	41.36 ± 0.35	43.32 ± 0.34	50.87 ± 0.38	38.73 ± 0.34
Microscopy	BCT	46.23 ± 0.48	51.23 ± 0.46	57.60 ± 0.43	54.61 ± 0.45	51.40 ± 0.46
	PNU	26.17 ± 0.25	26.03 ± 0.23	28.53 ± 0.26	27.12 ± 0.25	28.64 ± 0.26
	PRT	26.65 ± 0.28	28.15 ± 0.26	31.40 ± 0.30	32.87 ± 0.27	32.55 ± 0.29
Remote sensing	RESISC	42.49 ± 0.42	60.34 ± 0.44	60.00 ± 0.41	61.27 ± 0.43	57.14 ± 0.46
	RSICB	57.65 ± 0.54	84.49 ± 0.36	79.37 ± 0.39	83.72 ± 0.36	81.53 ± 0.40
	RSD	39.90 ± 0.42	66.35 ± 0.46	64.29 ± 0.41	72.30 ± 0.42	61.07 ± 0.47
Vehicles	CRS	31.39 ± 0.35	34.76 ± 0.36	35.05 ± 0.36	37.47 ± 0.37	32.20 ± 0.31
	APL	43.64 ± 0.51	52.67 ± 0.43	41.73 ± 0.42	56.30 ± 0.39	42.43 ± 0.40
	BTS	27.44 ± 0.28	35.51 ± 0.31	31.82 ± 0.28	40.35 ± 0.33	33.93 ± 0.29
Manufacturing	TEX	71.33 ± 0.60	82.25 ± 0.50	85.79 ± 0.38	86.20 ± 0.41	84.93 ± 0.44
	TEX_DTD	27.09 ± 0.28	49.24 ± 0.37	40.48 ± 0.35	49.45 ± 0.38	39.60 ± 0.36
	TEX_ALOT	69.09 ± 0.56	94.96 ± 0.23	92.44 ± 0.26	97.04 ± 0.15	92.97 ± 0.32
Human actions	SPT	45.98 ± 0.44	55.94 ± 0.42	51.93 ± 0.39	54.67 ± 0.42	52.74 ± 0.42
	ACT_40	29.32 ± 0.28	37.45 ± 0.34	37.96 ± 0.34	35.27 ± 0.32	37.15 ± 0.35
	ACT_410	40.49 ± 0.40	51.72 ± 0.41	50.43 ± 0.39	58.72 ± 0.39	52.12 ± 0.41
OCR	MD_MIX	19.51 ± 0.17	50.45 ± 0.41	19.68 ± 0.21	35.54 ± 0.58	18.78 ± 0.20
	MD_5_BIS	20.37 ± 0.18	38.49 ± 0.34	22.84 ± 0.23	53.14 ± 0.39	22.75 ± 0.23
	MD_6	22.24 ± 0.22	69.41 ± 0.42	32.29 ± 0.31	51.61 ± 0.46	24.98 ± 0.25

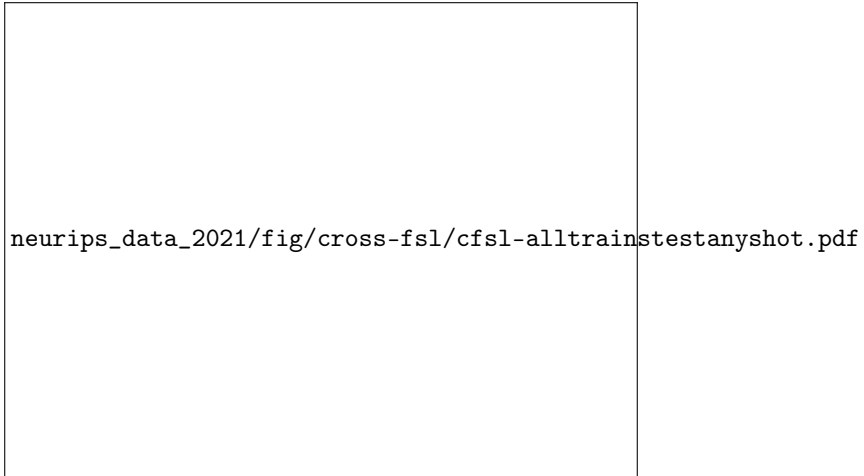


Figure 13: **Comparison of “cross-domain” few-shot learning trained on different settings but tested on the any-way any-shot setting.** We plot few-shot learning meta-test mean task accuracy, averaged over test tasks drawn from the 30 released Meta-Album datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 \times 30 = 54\,000$).

Table 13: Full performance results for cross-domain 5-way 20-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1\,800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	48.94 ± 0.48	63.79 ± 0.43	62.04 ± 0.37	67.99 ± 0.36	58.02 ± 0.41
	DOG	29.73 ± 0.31	35.63 ± 0.32	40.19 ± 0.32	40.05 ± 0.32	36.24 ± 0.31
	AWA	33.87 ± 0.38	51.84 ± 0.49	54.40 ± 0.43	58.49 ± 0.46	50.15 ± 0.48
Small animals	PLK	56.86 ± 0.54	65.72 ± 0.49	61.97 ± 0.41	76.19 ± 0.41	63.71 ± 0.52
	INS_2	27.98 ± 0.31	42.31 ± 0.35	43.40 ± 0.35	44.28 ± 0.36	40.70 ± 0.36
	INS	27.75 ± 0.30	43.22 ± 0.39	46.51 ± 0.37	49.40 ± 0.40	43.69 ± 0.38
Plants	FLW	56.41 ± 0.52	69.22 ± 0.51	68.95 ± 0.45	75.35 ± 0.43	71.06 ± 0.46
	PLT_NET	31.26 ± 0.31	47.83 ± 0.38	49.95 ± 0.34	47.25 ± 0.34	48.10 ± 0.35
	FNG	26.39 ± 0.27	35.44 ± 0.34	39.99 ± 0.34	42.02 ± 0.34	36.88 ± 0.34
Plant diseases	PLT_VIL	60.83 ± 0.52	72.18 ± 0.48	70.51 ± 0.39	78.77 ± 0.39	73.50 ± 0.43
	MED_LF	77.76 ± 0.46	85.50 ± 0.38	77.28 ± 0.38	88.53 ± 0.29	83.06 ± 0.38
	PLT_DOC	28.05 ± 0.28	43.92 ± 0.36	48.06 ± 0.34	54.50 ± 0.38	42.12 ± 0.33
Microscopy	BCT	46.74 ± 0.47	52.04 ± 0.46	58.27 ± 0.44	58.56 ± 0.47	58.92 ± 0.54
	PNU	26.19 ± 0.25	26.61 ± 0.24	29.93 ± 0.27	29.88 ± 0.25	30.87 ± 0.26
	PRT	26.93 ± 0.28	28.71 ± 0.26	34.81 ± 0.27	35.98 ± 0.28	33.15 ± 0.27
Remote sensing	RESISC	42.93 ± 0.44	62.09 ± 0.44	61.86 ± 0.42	64.37 ± 0.40	60.75 ± 0.44
	RSICB	57.97 ± 0.54	86.17 ± 0.35	82.72 ± 0.34	86.73 ± 0.32	83.35 ± 0.40
	RSD	39.29 ± 0.40	68.37 ± 0.45	68.91 ± 0.41	76.32 ± 0.40	65.66 ± 0.43
Vehicles	CRS	31.53 ± 0.36	35.88 ± 0.37	36.38 ± 0.35	42.31 ± 0.37	35.64 ± 0.36
	APL	43.64 ± 0.53	55.19 ± 0.45	48.82 ± 0.37	62.64 ± 0.40	45.55 ± 0.39
	BTS	27.31 ± 0.29	36.60 ± 0.32	36.96 ± 0.30	46.13 ± 0.33	35.53 ± 0.28
Manufacturing	TEX	70.89 ± 0.60	82.78 ± 0.51	86.10 ± 0.37	89.39 ± 0.31	88.69 ± 0.37
	TEX_DTD	26.78 ± 0.29	51.60 ± 0.38	44.25 ± 0.35	50.70 ± 0.37	42.53 ± 0.36
	TEX_ALOT	69.64 ± 0.56	95.71 ± 0.22	94.65 ± 0.22	97.73 ± 0.13	94.94 ± 0.24
Human actions	SPT	46.76 ± 0.44	58.44 ± 0.42	55.69 ± 0.37	59.35 ± 0.39	55.40 ± 0.41
	ACT_40	29.43 ± 0.30	39.43 ± 0.35	41.35 ± 0.36	39.34 ± 0.35	39.66 ± 0.35
	ACT_410	40.34 ± 0.42	54.19 ± 0.42	56.65 ± 0.38	64.87 ± 0.37	55.20 ± 0.41
OCR	MD_MIX	19.54 ± 0.16	52.72 ± 0.41	20.39 ± 0.22	46.29 ± 0.37	18.30 ± 0.19
	MD_5_BIS	20.32 ± 0.17	40.75 ± 0.37	25.81 ± 0.29	56.00 ± 0.37	22.34 ± 0.23
	MD_6	21.78 ± 0.20	70.69 ± 0.40	44.25 ± 0.37	64.42 ± 0.39	26.73 ± 0.26

Table 14: Full performance results for cross-domain any-way any-shot image classification on all datasets. The 95% confidence intervals are computed at per-task level over 3 runs per dataset with 600 tasks per run (total tasks: $3 \times 600 = 1800$).

Domain	Dataset ID	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
Large animals	BRD	29.95 ± 0.92	37.96 ± 0.91	43.08 ± 0.76	51.93 ± 0.72	34.40 ± 0.82
	DOG	19.58 ± 0.73	22.95 ± 0.76	25.76 ± 0.71	27.05 ± 0.72	22.69 ± 0.70
	AWA	21.53 ± 0.74	33.64 ± 0.83	38.01 ± 0.73	42.67 ± 0.76	35.19 ± 0.73
Small animals	PLK	36.18 ± 0.97	47.26 ± 0.90	42.89 ± 0.80	52.86 ± 0.73	45.29 ± 0.84
	INS_2	18.01 ± 0.70	27.07 ± 0.77	28.12 ± 0.71	29.37 ± 0.70	25.22 ± 0.73
	INS	17.88 ± 0.69	25.93 ± 0.78	29.86 ± 0.71	33.81 ± 0.72	27.91 ± 0.71
Plants	FLW	34.11 ± 0.98	42.86 ± 0.90	51.66 ± 0.75	60.11 ± 0.72	46.61 ± 0.85
	PLT_NET	20.30 ± 0.74	30.94 ± 0.82	33.53 ± 0.73	32.49 ± 0.74	30.26 ± 0.75
	FNG	17.58 ± 0.67	23.26 ± 0.75	26.94 ± 0.70	28.65 ± 0.71	25.05 ± 0.69
Plant diseases	PLT_VIL	38.25 ± 1.02	49.60 ± 0.97	51.78 ± 0.77	61.95 ± 0.72	50.69 ± 0.92
	MED_LF	54.99 ± 0.93	63.73 ± 0.84	62.36 ± 0.63	75.24 ± 0.55	58.56 ± 0.86
	PLT_DOC	18.06 ± 0.67	27.55 ± 0.77	30.39 ± 0.70	39.09 ± 0.73	28.13 ± 0.70
Microscopy	BCT	29.68 ± 0.95	35.47 ± 0.90	41.98 ± 0.80	42.60 ± 0.78	34.13 ± 0.87
	PNU	17.70 ± 0.64	18.50 ± 0.64	21.52 ± 0.62	19.42 ± 0.60	18.47 ± 0.61
	PRT	17.80 ± 0.70	18.10 ± 0.66	22.03 ± 0.67	22.10 ± 0.68	21.40 ± 0.67
Remote sensing	RESISC	27.05 ± 0.87	39.70 ± 0.86	43.55 ± 0.73	49.34 ± 0.71	39.34 ± 0.80
	RSICB	36.19 ± 1.03	67.64 ± 0.74	66.59 ± 0.65	70.96 ± 0.62	65.20 ± 0.76
	RSD	25.61 ± 0.88	46.74 ± 0.90	50.28 ± 0.75	59.66 ± 0.73	47.44 ± 0.77
Vehicles	CRS	20.10 ± 0.72	22.36 ± 0.74	24.12 ± 0.70	27.77 ± 0.70	21.02 ± 0.69
	APL	26.67 ± 0.91	37.08 ± 0.86	31.05 ± 0.70	40.28 ± 0.73	27.91 ± 0.78
	BTS	17.96 ± 0.68	24.09 ± 0.76	24.28 ± 0.69	30.36 ± 0.71	21.75 ± 0.67
Manufacturing	TEX	45.63 ± 1.11	66.63 ± 0.82	73.21 ± 0.60	79.37 ± 0.53	66.78 ± 0.82
	TEX_DTD	17.84 ± 0.65	34.59 ± 0.78	30.79 ± 0.69	32.95 ± 0.70	28.23 ± 0.68
	TEX_ALOT	44.15 ± 1.11	83.10 ± 0.56	85.45 ± 0.43	91.05 ± 0.39	81.62 ± 0.74
Human actions	SPT	28.81 ± 0.89	35.82 ± 0.89	37.01 ± 0.75	43.60 ± 0.74	35.15 ± 0.82
	ACT_40	19.11 ± 0.70	24.74 ± 0.77	27.81 ± 0.70	27.96 ± 0.70	23.55 ± 0.68
	ACT_410	25.03 ± 0.80	33.79 ± 0.83	40.50 ± 0.70	46.14 ± 0.70	36.83 ± 0.73
OCR	MD_MIX	13.42 ± 0.54	37.76 ± 0.79	12.99 ± 0.55	14.00 ± 0.56	12.88 ± 0.55
	MD_5_BIS	13.73 ± 0.54	28.78 ± 0.77	15.34 ± 0.58	24.55 ± 0.77	14.73 ± 0.57
	MD_6	15.16 ± 0.59	50.02 ± 0.92	20.00 ± 0.65	46.12 ± 0.82	16.85 ± 0.60

Table 15: Full performance results for cross-domain any-way any-shot image classification per number of ways. The 95% confidence intervals are computed at per-task level using the number of task shown in the table.

<i>N</i> -way	Evaluated Tasks	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
2	2925	0.70 ± 0.01	0.78 ± 0.01	0.73 ± 0.01	0.77 ± 0.01	0.72 ± 0.01
3	2708	0.55 ± 0.01	0.66 ± 0.01	0.61 ± 0.01	0.65 ± 0.01	0.58 ± 0.01
4	2910	0.46 ± 0.01	0.58 ± 0.01	0.54 ± 0.01	0.59 ± 0.01	0.52 ± 0.01
5	2704	0.38 ± 0.01	0.53 ± 0.01	0.48 ± 0.01	0.54 ± 0.01	0.46 ± 0.01
6	2856	0.33 ± 0.01	0.48 ± 0.01	0.44 ± 0.01	0.50 ± 0.01	0.43 ± 0.01
7	2683	0.30 ± 0.01	0.44 ± 0.01	0.42 ± 0.01	0.47 ± 0.01	0.40 ± 0.01
8	3025	0.26 ± 0.00	0.40 ± 0.01	0.39 ± 0.01	0.45 ± 0.01	0.37 ± 0.01
9	2864	0.24 ± 0.00	0.37 ± 0.01	0.37 ± 0.01	0.43 ± 0.01	0.35 ± 0.01
10	2943	0.22 ± 0.00	0.35 ± 0.01	0.35 ± 0.01	0.41 ± 0.01	0.33 ± 0.01
11	2892	0.19 ± 0.00	0.33 ± 0.01	0.33 ± 0.01	0.40 ± 0.01	0.31 ± 0.01
12	2983	0.18 ± 0.00	0.31 ± 0.01	0.32 ± 0.01	0.38 ± 0.01	0.29 ± 0.01
13	3076	0.17 ± 0.00	0.29 ± 0.01	0.31 ± 0.01	0.37 ± 0.01	0.28 ± 0.01
14	2952	0.15 ± 0.00	0.28 ± 0.01	0.30 ± 0.01	0.36 ± 0.01	0.26 ± 0.01
15	2811	0.14 ± 0.00	0.26 ± 0.01	0.29 ± 0.01	0.35 ± 0.01	0.25 ± 0.01
16	2682	0.13 ± 0.00	0.24 ± 0.01	0.27 ± 0.01	0.33 ± 0.01	0.23 ± 0.01
17	2842	0.12 ± 0.00	0.24 ± 0.01	0.27 ± 0.01	0.33 ± 0.01	0.22 ± 0.01
18	2779	0.12 ± 0.00	0.23 ± 0.01	0.26 ± 0.01	0.32 ± 0.01	0.21 ± 0.01
19	2711	0.11 ± 0.00	0.21 ± 0.01	0.25 ± 0.01	0.31 ± 0.01	0.20 ± 0.01
20	2654	0.10 ± 0.00	0.21 ± 0.01	0.25 ± 0.01	0.31 ± 0.01	0.19 ± 0.01

Table 16: Full performance results for cross-domain any-way any-shot image classification per number of shots. The 95% confidence intervals are computed at per-task level using the number of task shown in the table.

k -shot	Evaluated Tasks	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
1	2 635	0.21 ± 0.01	0.30 ± 0.01	0.29 ± 0.01	0.30 ± 0.01	0.25 ± 0.01
2	2 577	0.24 ± 0.01	0.35 ± 0.01	0.33 ± 0.01	0.35 ± 0.01	0.29 ± 0.01
3	2 571	0.25 ± 0.01	0.36 ± 0.01	0.34 ± 0.01	0.38 ± 0.01	0.31 ± 0.01
4	2 803	0.26 ± 0.01	0.37 ± 0.01	0.36 ± 0.01	0.40 ± 0.01	0.33 ± 0.01
5	2 737	0.26 ± 0.01	0.38 ± 0.01	0.37 ± 0.01	0.41 ± 0.01	0.34 ± 0.01
6	2 748	0.25 ± 0.01	0.37 ± 0.01	0.36 ± 0.01	0.41 ± 0.01	0.33 ± 0.01
7	2 761	0.26 ± 0.01	0.38 ± 0.01	0.37 ± 0.01	0.43 ± 0.01	0.34 ± 0.01
8	2 761	0.25 ± 0.01	0.38 ± 0.01	0.37 ± 0.01	0.43 ± 0.01	0.34 ± 0.01
9	2 718	0.26 ± 0.01	0.39 ± 0.01	0.38 ± 0.01	0.44 ± 0.01	0.35 ± 0.01
10	2 791	0.27 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.46 ± 0.01	0.37 ± 0.01
11	2 850	0.26 ± 0.01	0.39 ± 0.01	0.39 ± 0.01	0.45 ± 0.01	0.36 ± 0.01
12	2 747	0.26 ± 0.01	0.39 ± 0.01	0.39 ± 0.01	0.46 ± 0.01	0.36 ± 0.01
13	2 580	0.26 ± 0.01	0.39 ± 0.01	0.40 ± 0.01	0.46 ± 0.01	0.37 ± 0.01
14	2 607	0.26 ± 0.01	0.39 ± 0.01	0.40 ± 0.01	0.46 ± 0.01	0.37 ± 0.01
15	2 683	0.26 ± 0.01	0.39 ± 0.01	0.40 ± 0.01	0.47 ± 0.01	0.37 ± 0.01
16	2 616	0.27 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.47 ± 0.01	0.37 ± 0.01
17	2 632	0.27 ± 0.01	0.41 ± 0.01	0.41 ± 0.01	0.48 ± 0.01	0.38 ± 0.01
18	2 703	0.25 ± 0.01	0.38 ± 0.01	0.40 ± 0.01	0.47 ± 0.01	0.36 ± 0.01
19	2 753	0.25 ± 0.01	0.39 ± 0.01	0.40 ± 0.01	0.47 ± 0.01	0.37 ± 0.01
20	2 727	0.25 ± 0.01	0.39 ± 0.01	0.40 ± 0.01	0.47 ± 0.01	0.37 ± 0.01

Table 17: The running times for all techniques on all cross-domain settings over all datasets. The number of tasks in each phase is: 540 000 during meta-training, 129 600 during meta-validation, and 54 000 during meta-testing.

Setting	Phase	TrainFromScratch	Finetuning	MatchingNet	ProtoNet	FO-MAML
5-way 1-shot	Meta-train	0h0m	3h23m	6h33m	6h47m	17h40m
	Meta-validation	0h0m	24h53m	2h48m	2h45m	8h52m
	Meta-test	29h29m	9h21m	0h18m	0h19m	2h48m
5-way 5-shot	Meta-train	0h0m	3h30m	7h13m	7h0m	18h2m
	Meta-validation	0h0m	25h41m	3h24m	3h16m	9h32m
	Meta-test	29h24m	9h22m	0h19m	0h20m	2h58m
5-way 10-shot	Meta-train	0h0m	3h25m	7h25m	7h10m	19h28m
	Meta-validation	0h0m	25h45m	4h2m	3h55m	10h30m
	Meta-test	30h7m	9h26m	0h20m	0h21m	3h16m
5-way 20-shot	Meta-train	0h0m	3h24m	8h3m	8h20m	22h10m
	Meta-validation	0h0m	27h11m	5h21m	5h23m	13h18m
	Meta-test	29h56m	9h22m	0h24m	0h25m	3h39m
any-way any-shot	Meta-train	0h0m	3h20m	7h4m	7h59m	19h37m
	Meta-validation	0h0m	30h11m	8h27m	8h58m	17h21m
	Meta-test	30h2m	9h32m	0h32m	0h34m	4h19m

G Benchmarks/Datasets comparison

We compare the **Meta-Album meta-dataset** with other (meta-)datasets in Table 18. Each row shows a (meta-)dataset while each column shows features. The last three rows show versions of the newly proposed meta-dataset, *i.e.*, Meta-Album. The table describes the following features, divided in quantitative and qualitative features:

Quantitative features

- number of domains
- number of datasets
- number of total images
- min/max number of classes per domain
- min/max number of images per class
- size on disk (in GB)

Qualitative features

- Whether the (meta-)datasets has datasets from multiple domains
- Whether the (meta-)dataset is lightweight (easy to store on most computers, *i.e.*, < 20 GB)
- Whether the (meta-)dataset has a uniform number of images per class
- Whether the (meta-)dataset has images of uniform size
- Whether the (meta-)dataset is designed to be extended with more datasets or domains.

We discuss the number of domains and extensibility in more detail.

Dataset domains

We could identify the following domains in Meta-Dataset, VTAB, and CIFAR-100. For other datasets, we have not been able to find an adequate description of the domains, and as such we refrain from this.

Meta-Dataset The meta-dataset is divided into total 7 domains as listed below:

1. Organisms, *e.g.*, birds, fungi, flowers.
2. Common objects, *e.g.*, from ImageNet, ILSVRC, COCO.
3. Characters, *i.e.*, Omniglot.

Table 18: Feature comparison between Meta-Album and other large-scale or (meta-)datasets

Dataset/ Meta-Dataset	# of domains	# of datasets	# of images	min/max classes per domain	min/max images per class	size on disk	multi-domain	lightweight (<20GB)	uniform # of images per class	uniform image size	extendable
Meta-Dataset	7	10	53 068 000	43/1 696	3/140 000	210 GB	✓	✗	✗	✗	✗
VTAB	3	19	2 244 000	2/397	40/1 000	100 GB	✓	✗	✗	✗	✗
MS-COCO	1	1	328 000	80/80	9/10 777	44 GB	✗	✗	✗	✗	✗
Mini Imagenet	1	1	60 000	100/100	600/600	1 GB	✗	✓	✓	✓	✗
Omniglot	1	1	32 000	1 623/1 623	20/20	148 MB	✗	✓	✓	✓	✗
CUB-200	1	1	6 000	200/200	20/39	647 MB	✗	✓	✗	✗	✗
CIFAR-100	3	1	60 000	15/50	600/600	161 MB	✗	✓	✓	✓	✗
Meta-Album <i>Micro</i>	10	40	32 000	19/20	40/40	380 MB	✓	✓	✓	✓	✓
Meta-Album <i>Mini</i>	10	40	220 950	19/706	40/40	3.9 GB	✓	✓	✓	✓	✓
Meta-Album <i>Extended</i>	10	40	1 583 624	19/706	1/187 384	15 GB	✓	✓	✗	✓	✓

4. Textures, *i.e.*, DTD.
5. Vehicles, *i.e.*, FGVC-Aircraft.
6. Drawings, *i.e.*, Quick, Draw!
7. Traffic signs, *i.e.*, the German Traffic Sign Recognition Benchmark.

CIFAR-100 The CIFAR-100 is divided into 3 domains:

1. people, reptiles, small mammals, aquatic mammals, fish, insects, large carnivores, large omnivores and herbivores, medium-sized mammals, non-insect invertebrates
2. trees, flowers, fruit and vegetables
3. vehicles 1, vehicles 2, large natural outdoor scenes, large man-made outdoor things, household furniture, household electrical devices, food containers

VTAB VTAB is divided into 3 domains:

1. *Natural* image tasks include images of the natural world captured through standard cameras, representing generic objects, fine-grained classes, or abstract concepts.
2. *Specialized* tasks utilize images captured using specialist equipment, such as medical images or remote sensing.
3. *Structured* tasks often derive from artificial environments that target understanding of specific changes between images, such as predicting the distance to an object in a 3D scene, counting objects, or detecting orientation.

Extensibility

Meta-Album is the only one among these datasets that has been designed explicitly to be extended. We are inviting the community to contribute more domains and datasets. We provide open-source code to prepare more datasets following the Meta-Album format, and we have a review process so that newly added datasets are checked for quality and subsequently included.

Important Links

Meta-Album Website :

<https://meta-album.github.io/>

Meta-Album GitHub repo :

<https://github.com/ihsanullah2131/meta-album>

NeurIPS Cross-Domain MetaDL Competition 2022 :

<https://github.com/DustinCarrion/cd-metadl>

NeurIPS MetaDL Challenge 2021 :

<https://autodl.lri.fr/competitions/210>

MetaDL Self Service :

<https://competitions.codalab.org/competitions/31280>

Factsheets repo :

<https://github.com/ihsanullah2131/meta-album/tree/master/Factsheets>

Data Format :

<https://github.com/ihsanullah2131/meta-album/tree/master/DataFormat>

Data Format Check :

https://github.com/ihsanullah2131/meta-album/blob/master/DataFormat/check_data_format.py

Factsheet Report Script :

https://github.com/ihsanullah2131/meta-album/blob/master/Factsheets/generate_pdf_report.py

Factsheet Report Template :

<https://github.com/ihsanullah2131/meta-album/blob/master/Factsheets/template.html>

Contact

For any query about the Meta-Album meta-dataset, reach us out by email meta-album@chalearn.org.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [No]

Justification: We haven't mentioned the main claims in the abstract and introduction

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We have clearly explained the limitations of our work in Section: Discussion and conclusion

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We have detailed all the theoretical proofs in the main paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: **No**

Justification: We have tried to detail all the experiments details but we cannot make public our code and experiment details for reproducibility

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Our work is proprietary and we are not making the code and data available for public

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have explained the experimental settings or details in Section 3.2: Experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have reported error bars in Section 3.2: Experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided details about compute resources used in Section 3.2: Experiments, Appendix E: Within Domain Few-shot learning, and Appendix F: Cross-Domain Few-shot learning

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [No]

Justification: We are not bound to follow the NeurIPS code of ethics

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have detailed potential impacts and recommended use in Section 1.3

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: Indeed, we are releasing our datasets with proper license and terms of use. We have also provided recommended usage of these datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We are releasing our datasets with proper license, we are also mentioning the original licenses for the original datasets in Appendix B: License information of Meta-Album datasets

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We have properly documented our new datasets using meta-data files and datasheets for datasets (provided in the appendices).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: We have mentioned all the crowdsourcing details in the main paper

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[Yes\]](#)

Justification: We have received approvals from IRB and have mentioned in the main paper

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.