arXiv:1906.02944v1 [cs.CV] 7 Jun 2019

Learning Classifier Synthesis for Generalized Few-Shot Learning

Han-Jia Ye*
Nanjing University
yehj@lamda.nju.edu.cn

Hexiang Hu*
University of Southern California
hexiangh@usc.edu

De-Chuan Zhan
Nanjing University
zhandc@lamda.nju.edu.cn

Fei Sha†
Google Research
fsha@google.com

Abstract

Visual recognition in real-world requires handling long-tailed and even open-ended data. It is a practical utility of a visual system to reliably recognizing the populated "head" visual concepts and meanwhile to learn about "tail" categories of few instances. Class-balanced many-shot learning and few-shot learning tackle one side of this challenging problem, via either learning strong classifiers for populated categories or few-shot classifiers for the tail classes. In this paper, we investigate the problem of *generalized few-shot learning*, where recognition on the head and the tail are performed jointly. We propose a neural dictionary-based ClAssifier SynThesis LEarning (CASTLE) approach to synthesizes the calibrated "tail" classifiers in addition to the multi-class "head" classifiers, and simultaneously recognizes the head and tail visual categories in a global discerning framework. CASTLE has demonstrated superior performances across different learning scenarios, *i.e*, many-shot learning, few-shot learning, and generalized few-shot learning, on two standard benchmark datasets — *Mini*ImageNet and *Tiered*ImageNet.

1 Introduction

Visual recognition for objects in the "long tail" has been an important challenge to address [18, 36]. We often have a very limited amount of data on those objects as they are infrequently observed and/or visual exemplars of them are hard to collect. As such, state-of-the-art methods such as deep learning do not directly apply due to their notorious demand of a large number of annotated data [11, 13, 28].

Few-shot learning (FSL) [8, 29, 34] attempt to address this challenging problem, by being mindful that there are only a few instances (*i.e*, shots) per concept and distinguishing between the data-rich "head" categories as SEEN classes and data-scarce categories "tail" as UNSEEN classes. While it is difficult to build classifiers with data from UNSEEN classes, FSL leverage data from SEEN classes and design inductive biases such that classifiers would perform well on UNSEEN ones. We refer to Larochelle [15] for an up-to-date survey in few-shot learning.

This type of learning, however, creates a chasm in object recognition. Classifiers from many-shot learning for SEEN classes and those from few-shot learning for UNSEEN classes do not mix – they cannot be combined directly to recognize *all* object categories at the same time.

In this paper, we introduce the problem of **Generalized Few-Shot Learning (GFSL)**, which focuses on the joint classification of both data-rich and data-poor categories. In particular, our goal is for the

*Equal Contribution.
†On leave from University of Southern California (feisha@usc.edu)

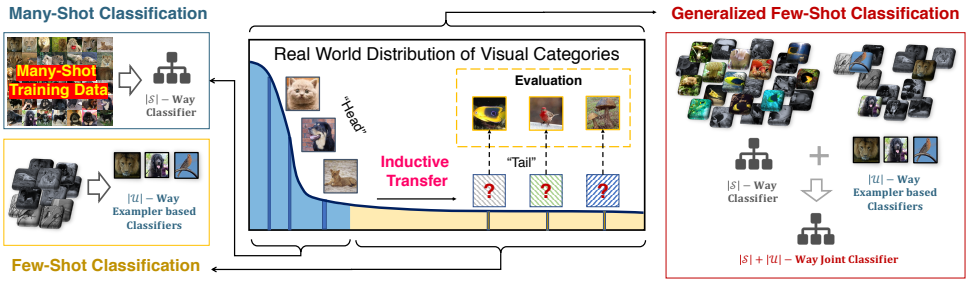


Figure 1: A conceptual diagram of generalized few-shot learning. It requires extracting inductive bias about UNSEEN "tail" categories from the SEEN categories. To achieve this, CASTLE tries to bridge between predicting the instances from data-rich (many-shot) "head" classes and those from data-poor (few-shot) "tail" classes.

model trained on the SEEN categories to be capable of incorporating limited UNSEEN class instances, and make predictions for test instances in both the "head" and "tail" of the entire distribution of categories. Figure 1 illustrates the high-level idea of our proposal. In contrast to prior works [18**?**] that focus on the transductive setup of learning "head" and "tail" concepts, our learning setup requires inductive modeling of the"tail" and is therefore more challenging as we assume no knowledge about the UNSEEN "tail" categories during the model learning phase.

To this end, we propose **ClAssifier SynThesis LEarning (CASTLE)**, where the few-shot classifiers are synthesized based on a shared neural dictionary across classes. Such synthesized few-shot classifiers *are then used together* with the regular many-shot classifiers. To this purpose, we create a scenario, via sampling a set of instances from SEEN categories and pretend that they come from UNSEEN, and apply the synthesized classifiers (based on the instances) as if they are many-shot classifiers to optimize multi-class classification together with regular many-shot SEEN classifiers. In other words, *we construct few-shot classifiers to not only perform well on the few-shot classes but also to perform competitively when used in conjunction with many-shot classifiers for classifying a bigger set of classes*. We argue that such high contrastive learning can benefit few-shot classification with high discernibility in its learned visual embeddings (cf. Section 5.2 for an empirical study).

We empirically validate our approach on two standard benchmark data sets — *Mini*ImageNet and *Tiered*ImageNet. The proposed approach retains competitive many-shot learning performances while demonstrating superior performances on few-shot learning and *generalized* few-shot learning.

2 Related work

Visual categories in real-world data are class-imbalanced, long-tailed, and open-ended [18, 36]. Building a high-quality visual system usually requires to have a large scale annotated training set with many shots per categories. Many large-scale datasets such as ImageNet have an ample number of instances for popular classes [13, 24]. However, the data-scarce "tail" of the category distribution matters. For example, a visual search engine needs to deal with the rare object of interests (*e.g* endangered species) or newly defined items (*e.g* new smartphone models), which only possess a few data instances. Directly training a system over all classes is prone to over-fit and can be biased towards the data-rich categories. cost-sensitive methods can provide some remedies [40], though they do not work well with extremely imbalanced scenarios with a large number of classes in the tail.

Zero-shot learning (ZSL) [1, 6, 14, 38] is a popular idea for addressing learning without labeled data. By aligning the visual and semantic definitions of objects, ZSL transfer the relationship between images and attributes learned from SEEN classes to UNSEEN ones, so as to recognize a novel instance with only its category-wise attributes [4, 5]. Generalized ZSL [7] extends this by calibrating a prediction bias to jointly predict between SEEN and UNSEEN classes.

ZSL is limited to recognizing objects with well-defined semantic descriptions. Few-shot learning (FSL) propose a more realistic setup, where we have access to a very limited number (instead of zero) of visual exemplars from the "tail" classes [3, 33]. Such approaches usually learn with a meta-learning objective that transfer the task-level inductive bias from SEEN classes to the new task composed by UNSEEN classes. For example, one line of works use meta-learned discriminative feature embeddings [16, 20, 25, 27, 29, 34, 39] together with non-parametric nearest neighbor classifiers, to

recognize novel classes given a few exemplars. Another line of works [2, 8, 17, 19, 35] choose to learn a common initialization to a pre-specified model configuration and adapt rapidly using fixed steps of gradient descents over the few-shot training data from UNSEEN categories.

FSL emphasizes on building models of the UNSEEN classes and *ignore its real-world use case of assisting the many-shot recognition of the "'head' categories*. A more realistic setting, *i.e*, low-shot learning, has been studied before [9, 10, 18, 37, 39]. The main aim is to recognize the entire set of concepts in a transductive learning framework — during the training of the target model, you have access to both the SEEN and UNSEEN categories. The key difference to our proposed GFSL is that we assume no access to UNSEEN classes in the learning phase, which requires the model to inductively transfer knowledge from SEEN classes to UNSEEN ones during the evaluation.

Previous approaches [9, 10, 18, 26, 37] often try to resolve the transductive learning setup of GFSL. Some of them [9, 10, 37] apply the exemplar-based classification paradigms on both SEEN and UNSEEN categories to resolve the transductive learning problem, requiring recomputing the centroid for SEEN categories after model updates. Others [18, 26] choose to ignore the explicit relationship and learn separate classifiers for SEEN and UNSEEN categories. In this paper, we propose the ClAssifier SynThesis LEarning (CASTLE) to solve the more challenging inductive modeling of UNSEEN categories in conjunction with seen ones. Our approach takes advantage of the neural dictionary to learn shared bases of composing many-shot and few-shot classifiers, which can better transfer the knowledge from SEEN to UNSEEN classifiers.

3 Few-shot and generalized few-shot learning

We provide the background of the problem settings in few-shot learning (FSL) and generalized FSL (Section 3.1), and then describe the meta-learning paradigm for tackling them (Section 3.2).

3.1 The Few-Shot and Generalized Few-Shot Learning Problems

A K-shot N-way classification task has N classes in total and K training examples in each class. In the setting of FSL, the training set $\mathcal{D}_{\mathbf{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{K}$, is also referred as the support set, where $\mathbf{x}_i \in \mathbb{R}^D$ is an instance with $\mathbf{y}_i \in \{0,1\}^N$ as its label.

Many-shot learning (MSL). When K is large enough, a classification model $f : \mathbb{R}^D \rightarrow \{0,1\}^N$ can be obtained by optimizing the following objective:

$$\min_f \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\mathbf{train}}} \ell(f(\mathbf{x}_i), \mathbf{y}_i) \ . \tag{1}$$

where the loss function $\ell(\cdot, \cdot)$ measures the discrepancy between the prediction and true label.

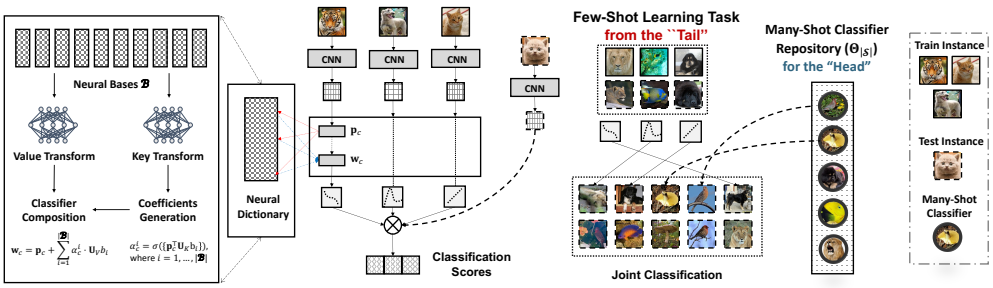
Few-shot learning (FSL). We assume there are two non-overlap sets with SEEN ($\mathcal{S}$) and UNSEEN classes ($\mathcal{U}$). The $\mathcal{U}$ is our target task but every class in it has a very small K. The main idea of FSL is to use $\mathcal{S}$, where each class could have a large K, to extract inductive bias that can be incorporated into solving the target task. We describe one way to obtain such bias in the next section.

Generalized FSL (GFSL). While FSL does not concern classification of the helper classes in $\mathcal{S}$, the aim of GFSL is to build models models are required to make a prediction over the collected $|\mathcal{S}| + |\mathcal{U}|$ categories [3]. The model needs to deal with many-shot classification from $|\mathcal{S}|$ SEEN classes and $|\mathcal{U}|$ emerging UNSEEN classes.

3.2 Meta-Learning for Few-Shot Learning

Meta-learning has been an effective framework for FSL [8, 29, 34]. The main idea is to *mimic* the future few-shot learning scenario by optimizing over K-shot N-way tasks drawn from the SEEN class sets $\mathcal{S}$. In particular, the K-shot N-way task $\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}$ sampled from $\mathcal{S}$ are constructed by randomly choosing N categories from $\mathcal{S}$ and K examples in each of them. Following this split use of $\mathcal{S}$, tasks and classes related to $\mathcal{S}$ are denoted as "meta-train", and called "meta-val/test" when they are related to $\mathcal{U}$. By sampling a corresponding test set $\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}$ (a.k.a. query set) from $\mathcal{S}$ for evaluating the resulting

[3] $|\mathcal{S}|$ and $|\mathcal{U}|$ denote the total number of classes from the SEEN and UNSEEN class sets respectively.



(a) **Classifier Synthesis** with Few-Shot Learning (b) **Inference** with Generalized Few-Shot Learning

Figure 2: Illustration of ClAssifier SynThesize LEarning (CASTLE).

few-shot classifier's inductive bias, meta-learning optimizes a shared f across few-shot tasks by minimizing the cumulated loss:

$$\min_{f} \sum_{(\mathbf{x}_j^{\mathcal{S}}, \mathbf{y}_j^{\mathcal{S}}) \in \mathcal{D}_{\mathbf{test}}^{\mathcal{S}}} \ell\left(f\left(\mathbf{x}_j^{\mathcal{S}}; \mathcal{D}_{\mathbf{train}}^{\mathcal{S}}\right), \mathbf{y}_j^{\mathcal{S}}\right) \tag{2}$$

Therefore, a classifier f has low loss value on average when predicting instance $\mathbf{x}_j^{\mathcal{S}}$ given the training set $\mathcal{D}_{\mathbf{train}}^{\mathcal{S}}$ (meta-training phase), which can also be applied to few-shot tasks drawn from UNSEEN class set $\mathcal{U}$ (meta-val/test phase).

[32] provides a comprehensive survey of different implementation of this idea. In this paper, we focus on the methods described in [29, 34]. Specifically, the classifier f is based on an embedding function, $f = \phi : \mathbb{R}^D \to \mathbb{R}^d$, which transforms input examples into a latent space with d dimensions. ϕ is learned to pull similar objects close while dissimilar ones far away [12]. For a test instance $\mathbf{x}_j$, the embedding function ϕ makes a prediction based on a soft nearest neighbor classifier:

$$\hat{y}_j = f\left(\mathbf{x}_j; \mathcal{D}_{\mathbf{train}}\right) = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\mathbf{train}}} \mathbf{sim}\left(\phi(\mathbf{x}_j), \phi(\mathbf{x}_i)\right) \cdot \mathbf{y}_i \tag{3}$$

The $\mathbf{sim}(\phi(\mathbf{x}_j), \phi(\mathbf{x}_i))$ measures the similarity between the test instance embedding $\phi(\mathbf{x}_j)$ and each training instance embedding $\phi(\mathbf{x}_i)$ [29, 34]. When there is more than one instance in each class, i.e., $K > 1$, instances in the same class can also be averaged to assist make final decision [29]. By learning a good embedding, the important visual features for few-shot classification is distilled, which will be used for few-shot tasks from the UNSEEN classes.

4 Learning classifier synthesis for generalized few-shot learning

We introduce the main idea of our approach first, followed by a detailed description of two key ingredients: (1) synthesize classifiers with a neural dictionary (2) joint learning many-shot and (synthesized) few-shot classifiers.

Main Idea Prior studies have shown that there is a prediction bias towards SEEN classes, due to the overconfident classifiers trained on the many more data from the SEEN classes [7, 39]. We address such problem from two perspectives with the proposed ClAssifier SynThesis LEarning (CASTLE). Figure 2 illustrates the ideas. First, we parameterize the classifiers for all classes from $\mathcal{S} \cup \mathcal{U}$, by synthesizing them from a common set of bases. Secondly, we propose a unified objective optimized to bridge classification of many-shot classes and few-shot ones. This objective forces the few-shot classifiers to compete against the many-shot classifiers to fight back the prediction bias. This leads to more discriminative few-shot classifiers that perform well in the setting of GFSL.

4.1 Classifier Composition with a Neural Dictionary

Our model is based on classifier synthesis [4, 6] with a learned neural dictionary. Here we define a dictionary as pairs of "key" and "value" embeddings, where each "key" and "value" is associated with a base, which is designed to encode shared primitives for composing classifiers of $\mathcal{S} \cup \mathcal{U}$. Formally, the neural dictionary contains a set of $|\mathcal{B}|$ learnable bases $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{|\mathcal{B}|}\}$, and $\mathbf{b}_i \in \mathcal{B} \in \mathcal{R}^d$.

The key and value for the dictionary are generated based on two linear projections of elements in $\mathcal{B}$. For instance, $\mathrm{U}_K\mathbf{b}_i$ and $\mathrm{U}_V\mathbf{b}_i$ represent the generated key and value embeddings.

To synthesize a classifier for a class c, we first compute the class signature as the embedding prototype, defined as the average embedding of all K shots of instances (in a K-shot N-way task):

$$\mathbf{p}_c = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{D}_{\mathbf{train}}} \phi(\mathbf{x}_i) \cdot \mathbb{I}[\mathbf{y}_i = c] \tag{4}$$

Here, $\mathbb{I}[\mathbf{y}_i = c]$ denotes an indicator function that selects instances in the class c. We then compute the coefficients α_c for the classifier synthesis of class c, via measuring the compatibility score between the class signature and the key embeddings of the neural dictionary,

$$\alpha_c^i \propto \exp\left(\mathbf{p}_c^\top \mathrm{U}_K \mathbf{b}_i\right), \text{where } \; i = 1, \cdots, |\mathcal{B}| \tag{5}$$

The coefficient α_c^i is then *normalized* with the sum of compatibility scores over all $|\mathcal{B}|$ bases, which then is used to convexly combine the value embeddings and synthesize the classifier,

$$\mathbf{w}_c = \mathbf{p}_c + \sum_{i=1}^{|\mathcal{B}|} \alpha_c^i \cdot \mathrm{U}_V \mathbf{b}_i \tag{6}$$

We formulate the classifier composition as a summation of the initial prototype embedding $\mathbf{p}_c$ and the residual component $\sum_{i=1}^{|\mathcal{B}|} \alpha_c^i \cdot \mathrm{U}_V \mathbf{b}_i$. Such a composed classifier is then ℓ_2-normalized and used for (generalized) few-shot classification. Since both the embedding "key" and classifier "value" are generated based on the same set of neural bases, it encodes a compact set of latent features for a wide range of classes. We hope the learned neural bases contain a rich set of classifier primitives to be transferred to novel composition of emerging visual categories.

4.2 Joint Learning of Few-Shot and Many-Shot Classifiers

Our learning has two purposes. The first is to enable transferring knowledge from SEEN classes to UNSEEN classes. The second is to enable the few-shot classifiers to do well when used in conjunction with classifiers for many-shot classes. We achieve them by designing a unified objective.

Meta learning loss for "tail" concepts. Suppose we have sampled a K-shot N-way few-shot learning task $\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}$, which contains a set $\mathcal{C}$ of visual categories, we synthesize the classifiers for $\mathcal{C}$ as $\mathbf{W}_{\mathcal{C}} = \{\, \mathbf{w}_c \mid c \in \mathcal{C} \,\}$. Then we can rewrite Eq 2 as,

$$\ell^{(\mathcal{C})}_{\mathbf{meta}} = \sum_{(\mathbf{x}_j^{\mathcal{S}}, \mathbf{y}_j^{\mathcal{S}}) \in \mathcal{D}^{\mathcal{S}}_{\mathbf{test}}} \ell\Big(\mathbf{W}_{\mathcal{C}}^\top \cdot \phi\big(\mathbf{x}_j^{\mathcal{S}}\big), \mathbf{y}_j^{\mathcal{S}}\Big) \tag{7}$$

which uses the synthesized classifiers, rather than the soft nearest neighbor classifiers [34], to minimize losses for the test data of SEEN categories $\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}$.

Unified learning objective. For each class in $s \in \mathcal{S}$, we have a model (*i.e*, liner classifier over some features to be learnt) $\mathbf{\Theta}_s$. We then use those models $\mathbf{\Theta}_{\mathcal{S}-\mathcal{C}}$ for $(\mathcal{S} - \mathcal{C})$ and combine them with the models $\mathbf{W}_{\mathcal{C}}$ from the few-shot classes $\mathcal{C}$ to form a set of classifiers $\hat{\mathbf{W}} = \mathbf{W}_{\mathcal{C}} \cup \mathbf{\Theta}_{\mathcal{S}-\mathcal{C}}$, now over all classes in $\mathcal{S}$. We desire them to do well in the labeling space. This is achieved by taking $\hat{\mathbf{W}}$ into Eq 1, *i.e* the standard multi-class loss:

$$\ell^{(\mathcal{C})}_{\mathbf{multi\text{-}class}} = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{S}} \ell\Big(\hat{\mathbf{W}}^\top \phi\big(\mathbf{x}_i\big), \mathbf{y}_i\Big) \tag{8}$$

Note that, despite the few-shot classifiers $\mathbf{W}_{\mathcal{C}}$ are synthesized using with only K instances (cf. eq. (4)), they are optimized to perform well on all the instances from $\mathcal{C}$ and moreover, to perform well against all the instances from other classes. Finally, this is balanced with the need to perform well only on the $\mathcal{C}$:

$$\min_{\{\phi, \mathcal{B}, \{\mathbf{\Theta}_s\}, \mathrm{U}_K, \mathrm{U}_V\}} \sum_{\mathcal{C} \subset \mathcal{S}} \ell^{(\mathcal{C})}_{\mathbf{meta}} + \lambda \ell^{(\mathcal{C})}_{\mathbf{multi\text{-}class}} \tag{9}$$

where λ is the tradeoff hyper-parameter that balances two objectives and we have summed over random sampling of $\mathcal{C}$ ("fake" few-shot classes from $\mathcal{S}$). Note that $\mathbf{W}_{\mathcal{C}}$ serves as the **bridge** to connect $\mathcal{C}$ (few-shot) and $(\mathcal{S} - \mathcal{C})$ (many-shot).

Table 1: Few-shot classification accuracy on *mini*ImageNet with the ResNet-12 backbone.

Setups →	1-Shot 5-Way	5-Shot 5-Way
ProtoNet [29]	61.40 ± 0.02	76.56 ± 0.02
LEO [25]	61.76 ± 0.08	77.59 ± 0.12
OptNet [16]	62.64 ± 0.61	78.63 ± 0.46
FEAT [39]	62.96 ± 0.02	78.49 ± 0.02
Ours: CASTLE	**63.51** ± 0.02	**79.52** ± 0.02

Table 2: Few-shot classification accuracy on *Tiered*ImageNet with the ResNet-12 backbone.

Setups →	1-Shot 5-Way	5-Shot 5-Way
ProtoNet [29]	53.31 ± 0.89	72.69 ± 0.74
RelationNet [30]	54.48 ± 0.93	71.32 ± 0.78
LEO [25]	66.33 ± 0.05	81.44 ± 0.09
OptNet [16]	65.99 ± 0.72	81.56 ± 0.63
Ours: CASTLE	**68.79** ± 0.02	**83.92** ± 0.02

5 Experiments

We validate the effectiveness of the CASTLE in this section. We begin by introducing the setups, and then show the results of standard few-shot learning (Section 5.2), as well as the generalized few-shot/low-shot learning (Section 5.3). Finally, we perform an in-depth analysis in Section 5.4.

5.1 Experimental setups

Here we describe the general setups for (generalized) few-shot classification in our experiments. We provide complete details in supplementary material and will release our codes.

Data Sets. We introduce the two benchmark data sets used in our experiments briefly as below.

- **The *mini*ImageNet dataset** [34] is a subset of the ILSVRC-12 dataset [24]. There are totally 100 classes and 600 examples per class. For evaluation, we follow the split of [22] and use 64 of 100 classes for meta-training, 16 for validation, and 20 for meta-test (model evaluation).
- **The *Tiered*ImageNet [23]** contains 34 super-categories in total, with 20 for meta-training, 6 for validation, and 8 for meta-test. This results in 351, 97, and 160 classes for meta-training, meta-validation, and meta-test, respectively. The diversity of the super-concepts leads to a more challenging few-shot classification problem.

Implementation Details. Following the recent methods [21, 25, 39], we use 12 layer residual network [11] (ResNet-12) to implement the embedding backbone ϕ. We first pre-train this backbone network (also explored by [16, 21, 25, 39]) and perform model selection strategy similar to [39]. To learn our methods as well as baseline systems, we then use Momentum SGD with an initial learning rate 1e-4. The learning rate will be halved after optimizing every 2,000 iterations.

5.2 Standard few-Shot learning

Setups. We follow [8, 29, 31, 34] and use the same evaluation protocol, *i.e*, the performance of 1-shot 5-way and 5-shot 5-way classification. We keep the same configuration of tasks between meta-training and meta-test. We test models over 10,000 sampled few-shot tasks for a stable measure of performances [25, 39]. The mean accuracy and 95% confidence interval are reported.

Results. The few-shot classification results on *Mini*ImageNet and *Tiered*ImageNet are shown in Table 1 and Table 2. We listed some recent few-shot learning approaches for a comparison. We observe that our CASTLE approach outperforms previous *state-of-the-art* methods on both 1-shot 5-way and 5-shot 5 way classification settings, across two data sets. This supports our hypothesis that jointly learning with many-shot classification forces few-shot classifiers to be discriminative. Please refer to the supplementary material for visualizations of the learned visual embeddings.

5.3 Generalized few-shot learning

Setups. We compare CASTLE with baseline approaches on the previous two data sets and evaluate classification accuracy on both SEEN and UNSEEN categories. We use non-overlap test images from the ILSVRC-12 dataset [24] to evaluate models on SEEN classes to avoid information leak. And use few-shot evaluation protocols to evaluate models on UNSEEN tail classes.

GFSL Baselines. We explore several (strong) choices in deriving classifiers for the SEEN classes and UNSEEN: **(1) Multiclass Classifier (MC) + kNN** A multi-class classifier is trained on the SEEN

Table 3: Generalized Few-shot classification accuracies on *Mini*ImageNet. We denote the X/Y in "Many-Shot" column as the performances of one-shot trained model ($X\%$) and five-shot trained model ($Y\%$), respectfully.

Classification on →	**64 HEAD Categories**	**20 TAIL Categories**		**All Categories**	
Setups →	Many-Shot	1-Shot	5-Shot	TAIL w/ 1-Shot	TAIL w/ 5-Shot
Perf. Measures →	*Mean Accuracy*	*Mean Accuracy*		*Harmonic Mean Accuracy*	
MC + kNN	90.99	27.91	50.98	2.04	6.71
ProtoNet + ProtoNet	42.77 / 55.46	30.54	51.64	5.15	6.48
MC + ProtoNet	90.39 / 90.27	30.89	51.76	4.82	6.56
Ours: CASTLE	91.23 / 91.28	**33.11**	**52.95**	**8.62**	**9.60**

Table 4: Generalized Few-shot classification accuracy on *Tiered*ImageNet. We denote the X/Y in "Many-Shot" column as the performances of one-shot trained model ($X\%$) and five-shot trained model ($Y\%$), respectfully.

Classification on →	**351 HEAD Categories**	**160 TAIL Categories**		**All Categories**	
Setups →	Many-Shot	1-Shot	5-Shot	TAIL w/ 1-Shot	TAIL w/ 5-Shot
Perf. Measures →	*Mean Accuracy*	*Mean Accuracy*		*Harmonic Mean Accuracy*	
MC + kNN	63.92	12.37	25.70	0.51	0.87
ProtoNet + ProtoNet	15.95 / 22.47	12.98	27.00	0.83	1.07
MC + ProtoNet	57.74 / 60.95	12.84	26.89	0.77	0.89
Ours: CASTLE	59.96 / 61.85	**14.82**	**27.55**	**1.09**	**1.21**

classes as standard many-shot classification [11]. When evaluated on UNSEEN classes for few-shot tasks, we apply the learned feature embedding with a nearest neighbor classifier. **(2) ProtoNet + ProtoNet** We train Prototypical Network [29] (a.k.a ProtoNet) on SEEN classes, pretending they were few-shot. When evaluated on the SEEN categories, we randomly sample 100 training instances per category to compute the class prototypes. We use the MC classifier's feature mapping to initialize the embedding function. We use the final embedding function for UNSEEN classes.**(3) MC+ ProtoNet** We combine the learning objective of (1) and (2) to jointly learn the MC classifier and feature embedding. It is a very strong baseline as it trades off between few-shot and many-shot learning. The supplementary material details the training, inference, and hyper-parameters for each method.

To evaluate each method's performances on the joint space of SEEN and UNSEEN classes, for ProtoNet+ProtoNet, the prediction is straightforward as both sets of classes are generated with ProtoNet. For CASTLE, we use the $\{\mathbf{\Theta}_s\}$ (*i.e*, the multiclass classifiers, defined in section 4.2) for the SEEN and the synthesized classifiers for the UNSEEN classes to classify an instance into all classes and then select the prediction with the highest confidence score. This is also done for MC+kNN and MC+ProtoNet. Following practices in *generalized* zero-shot learning [38], we compute the top-1 accuracy for each SEEN and UNSEEN classes, and take their harmonic mean as the performance measure. Since the $\mathcal{D}^{\mathcal{U}}_{\textbf{train}}$ here is randomly sampled, we re-run this evaluation repeatedly (1000 times on both datasets) to obtain stable measures.

Results. The *generalized* few-shot classification results on *Mini*ImageNet and *Tiered*ImageNet are shown in Table 3 and Table 4. We compare our approach with baseline approaches. We observe that CASTLE outperforms all other approaches in UNSEEN and ALL categories, across two data sets. On the SEEN categories, CASTLE remains competitive against MC classifier. We conduct ablation study and analysis in Section 5.4 to study the many different factors' influences.

5.4 Ablation study and analysis

In this section, we aim to study the ablated variant of our approach and perform in-depth analyses. Due to the limited space, we omit some implementation details (or additional results) in the main text. Please see supplementary materials for the complete version of them.

Effects on the neural dictionary size $|\mathcal{B}|$. We show the effects of the dictionary size (as the ratio of SEEN class size) for standard few-shot learning in Figure 3. We observe that the neural dictionary with a ratio of 2 works best amongst all other dictionary sizes. Therefore, we use it across all experiments. Note that when the $|\mathcal{B}| = 0$, our method degenerates to a variant of ProtoNet [29]

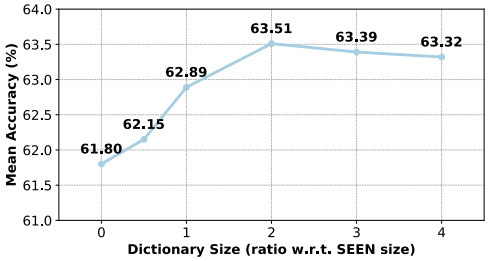


Figure 3: The 1-shot 5-way accuracy on UNSEEN of *Mini*ImageNet with different size of dictionaries.

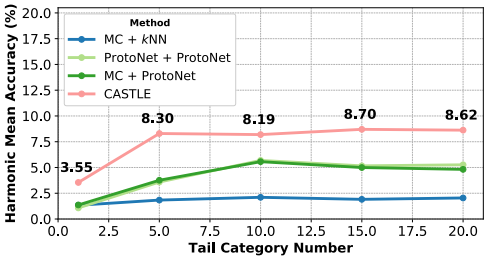


Figure 4: The 1-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

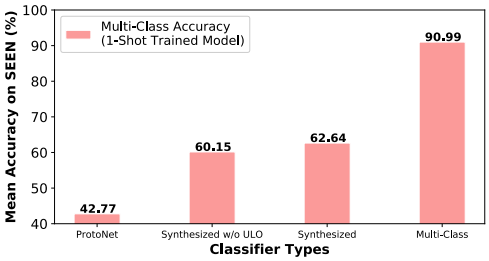


Figure 5: The 64-way multi-class accuracy on SEEN of *Mini*ImageNet with 1-shot trained model.

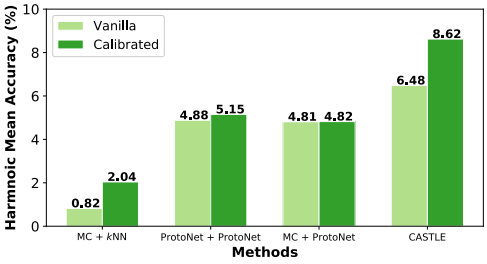


Figure 6: Calibration's effect to the 1-shot GFSL performance of different classifiers on *Mini*ImageNet.

Incremental evaluation of generalized few-shot learning. Other than the accuracy of all UNSEEN categories shown in GFSL (cf. Table 3 and 4), we study the dynamic of how harmonic mean accuracy changes with an incremental number of UNSEEN "tail' concepts emerging. In other words, we show the GFSL performances w.r.t different numbers of "tail" concepts. We use this as a robust evaluation of each system's GFSL capability. The one-shot learning result is shown as Figure 4. We observe that CASTLE consistently outperforms other baselines by a clear margin. Please refer to the supplementary material for five-shot learning results.

How well is synthesized classifiers comparing multi-class classifiers? To assess the quality of synthesized classifier, we made a comparison against ProtoNet and also the Multi-class Classifier on the "head" SEEN concepts. To do so, we sample few-shot training instances on each SEEN category to synthesize classifiers (or compute class prototypes for ProtoNet), and then use solely the synthesized classifiers/class prototypes to evaluate multi-class accuracy. The results are shown in the Figure 5. We observe that the learned synthesized classifier outperforms over ProtoNet by a large margin. Also, the model trained with unified learning objective (ULO) improves over the vanilla synthesized classifiers. Note that there is still a significant gap left against multi-class classifiers trained on the entire dataset. It suggests that the classifier synthesis we learned is effective against using sole instance embeddings while still far from the many-shot multi-class classifiers.

Confidence calibration makes a difference in GFSL evaluation. As suggested by Chao et al. [7], there exists a prediction bias in the classifiers learned for SEEN and UNSEEN categories in generalized zero-shot learning. *Therefore, we aim to study whether similar observation exists with few-shot learning*. To do so, we compute a calibration factor based on the validation set of UNSEEN categories such that the logits of prediction can be calibrated by subtracting this factor out from the confidence of SEEN categories' predictions. We observe that this results in a consistent improvement over the harmonic mean of accuracy for all SEEN and UNSEEN categories (see Figure 6). This suggests that similar prediction bias also exists in generalized few-shot learning.

6 Discussion

Both the populated "head" and data-scarce "tail" visual concepts are essential to a visual recognition system. Different from the previous studies on class-balanced and few-shot learning, we propose a *generalized few-shot learning* algorithm stressing both sides simultaneously. Our neural dictionary-based ClAssifier SynThesis LEarning (CASTLE) approach synthesizes the "tail" classifiers based on the few-shot instances. A unified learning objective enforces the few-shot classifiers to work as

competitive as the many-shot ones, and produce better calibrated prediction for the generalized "head" and "tail" classification. Besides achieving state-of-the-art performance w.r.t. few-shot tasks on two popular benchmarks, CASTLE demonstrates high discriminative ability in the generalized few-shot scenario as well.

Acknowledgment. This work is partially supported by NSF Awards IIS-1513966/ 1632803/1833137, CCF-1139148, DARPA Award#: FA8750-18-2-0117, DARPA-D3M - Award UCB-00009528, Google Research Awards, gifts from Facebook and Netflix, and ARO# W911NF-12-1-0241 and W911NF-15-1-0484. We thank Bowen, Zhang, Jiacheng Chen, and Soravit Changpinyo for feedbacks on an earlier version of this draft.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, Portland, OR, 2013.

[2] Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. How to train your MAML. *CoRR*, abs/1810.09502, 2018.

[3] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, Las Vegas, NV, 2016.

[5] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3496–3505, Venice,Italy, 2017.

[6] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Classifier and exemplar synthesis for zero-shot learning. *CoRR*, abs/1812.06423, 2018.

[7] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Proceedings of the 14th European Conference on Computer Vision*, pages 52–68, Amsterdam, The Netherlands, 2016.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, Sydney, Australia, 2017.

[9] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems 31*, pages 983–993. 2018.

[10] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision*, pages 3037–3046, Venice, Italy, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[14] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3): 453–465, 2014.

[15] Hugo Larochelle. Few-shot learning with meta-learning: Progress made and challenges ahead. 2018.

[16] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019.

[17] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2933–2942, Stockholm, Sweden, 2018.

[18] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. *CoRR*, abs/1904.05160, 2019.

[19] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[20] Boris N. Oreshkin, Pau Rodríguez, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. *CoRR*, abs/1805.10123, 2018.

[21] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. *CoRR*, abs/1706.03466, 2017.

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

[23] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. *CoRR*, abs/1803.00676, 2018.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[25] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *CoRR*, abs/1807.05960, 2018.

[26] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. *CoRR*, abs/1812.01784, 2018.

[27] Tyler R. Scott, Karl Ridgeway, and Michael C. Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Advances in Neural Information Processing Systems 31*, pages 76–85. 2018.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, pages 4080–4090. Curran Associates, Inc., 2017.

[30] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, Salt Lake City, UT, 2018.

[31] Eleni Triantafillou, Richard S. Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems 30*, pages 2252–2262. Curran Associates, Inc., 2017.

[32] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

[33] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

[34] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.

[35] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*, 2018.

[36] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems 30*, pages 7032–7042. Curran Associates, Inc., 2017.

[37] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, Salt Lake City, UT., 2018.

[38] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3077–3086, Honolulu, HI, 2017.

[39] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *CoRR*, abs/1812.03664, 2018.

[40] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge Data Engineering*, 18(1):63–77, 2006.

A Implementation details

Our implementation of CASTLE and baseline approaches is made publicly available at https://github.com/Sha-Lab/CASTLE.

A.1 Data set details.

Two benchmark data sets are used in our experiments. The *mini*ImageNet dataset [34] is a subset of the ILSVRC-12 dataset [24]. There are totally 100 classes and 600 examples in each class. For evaluation, we follow the split of [22] and use 64 of 100 classes for meta-training, 16 for validation, and 20 for meta-test (model evaluation). In other words, a model is trained on few-shot tasks sampled from the 64 SEEN classes set during meta-training, and the best model is selected based on the few-shot classification performance over the 16 class set. The final model is evaluated based on few-shot tasks sampled from the 20 UNSEEN classes.

The *Tiered*ImageNet [23] is a more complicated version compared with the *mini*ImageNet. It contains 34 super-categories in total, with 20 for meta-training, 6 for validation, and 8 for model testing (meta-test). Each of the super-category has 10 to 30 classes. In detail, there are 351, 97, and 160 classes for meta-training, meta-validation, and meta-test, respectively. The divergence of the super-concept leads to a more difficult few-shot classification problem.

A.2 Feature network specification.

Following the setting of most recent methods [21, 25, 39], we use the residual network [11] to implement the embedding backbone ϕ. Different from the standard configuration, the literature [21, 25, 39] resize the input image to $80 \times 80 \times 3$ for *MiniImageNet* (while $84 \times 84 \times 3$ for *Tiered*ImageNet) and remove the first two down-sampling layers in the network. In concrete words, three residual blocks are used after an initial convolutional layer (with stride 1 and padding 1) over the image, which have channels $160/320/640$, stride 2, and padding 2. After a global average pooling layer, it leads to a 640 dimensional embedding. The concrete architecture is visualized as Figure 11. Please refer to Pytorch documentation [4] for complete references of each building blocks.

A.3 Pre-training strategy.

Before the meta-training stage, we try to find a good initialization for the embedding ϕ. In particular, on *MiniImageNet* we add a linear layer on the backbone output and optimize a 64-way (while 351-way for *Tiered*ImageNet) classification problem on the meta-training set with the cross-entropy loss function. Stochastic gradient descent with Momentum 0.9 is used to complete such optimization. The 16 classes in *MiniImageNet* (resp. 97 classes in *Tiered*ImageNet) for model selection also assist the choice of the pre-trained model. After each epoch, we use the current embedding and measures the nearest neighbor based few-shot classification performance on the sampled few-shot tasks from these 16 (resp. 97) classes. The most suitable embedding function is recorded. After that, such learned backbone is used to initialize the embedding part ϕ of the whole model. For the meta-learning phase, SGD with Momentum is used with an initial learning rate 1e-4. The learning rate will be halved after optimizing 2,000 iterations.

A.4 Complete Details on GFSL baselines

A.4.1 Multiclass Classifier (MC) + kNN.

Setup. We train a multi-class classifier on the populated SEEN classes following practices of training Residual Networks [11]. Here a ResNet-12 backbone network is used, identical to the ones described in Section A.2. During the training $|\mathcal{S}|$-way classifiers are trained in a supervised learning manner.

Training and Inference. During the inference, test examples of $\mathcal{S}$ categories are evaluated based on the $|\mathcal{S}|$-way classifiers and $|\mathcal{U}|$ categories are evaluated using the support embeddings from $\mathcal{D}^{\mathcal{U}}_{\mathbf{train}}$ with a nearest neighbor classifier. To evaluate the generalized few-shot classification task, we take

[4] See https://pytorch.org/docs/stable/index.html for references.

the union of multi-class classifiers' confidence and ProtoNet confidence as joint classification scores on $\mathcal{S} \cup \mathcal{U}$. A calibration is applied to balance the confidence scores.

A.4.2 ProtoNet + ProtoNet.

Setup. We train a few-shot classifier (initialized by the MC classifier's feature mapping) using the Prototypical Network [29] (a.k.a ProtoNet) and evaluate its performances across three settings. The backbone network is the same ResNet-12 as before.

Training and Inference. During the inference, we compute the class prototypes of SEEN classes via using 100 training instances per category. The class prototypes of UNSEEN classes are computed based on the sampled few-shot training set. During the inference of *generalized* few-shot learning, we take the few-shot prototypes computed on SEEN class and UNSEEN class. A calibration is applied to balance the confidence scores.

A.4.3 MC + ProtoNet.

Setup. We combine the learning objective of (1) and (2) to jointly learn the MC classifier and feature embedding. It is a very strong baseline as it trades off between few-shot and many-shot learning. Therefore, this learned model can be used as multi-class linear classifiers on the "head" categories, and used as ProtoNet on the "tail" categories.

Training and Inference. During the inference, the model predicts instances from SEEN class $\mathcal{S}$ with the MC classifier, while takes advantage of the few-shot prototypes to discern UNSEEN class instances. To evaluate the generalized few-shot classification task, we take the union of multi-class classifiers' confidence and ProtoNet confidence as joint classification scores on $\mathcal{S} \cup \mathcal{U}$. A calibration is applied to balance the confidence scores.

As mentioned before, to obtain better generalized few-shot learning performances, a confidence calibration procedure between predictions for $\mathcal{S}$ and $\mathcal{U}$ is necessary. We therefore tune this factor based on the validation UNSEEN classes (*e.g* in the *Mini*ImageNet cases, we use 16 validation classes to compute this value) and then applied to the evaluation on test UNSEEN classes (*e.g* corresponding to the 20 test categories in *Mini*ImageNet).

B FSL Details and Additional Results

B.1 Details on Few-Shot Learning Setups

As mentioned in the main text, now we give the complete details for FSL setups. We follow [8, 29, 31, 34] and use the same evaluation protocol over all data sets, i.e., the performance of 1-shot 5-way and 5-shot 5-way classification. We keep the same configuration of tasks between meta-training and meta-test. In other words, for the 1-shot 5-way problem, we keep sampling the 1-shot 5-way training set (a.k.a. support set in the literature) from SEEN class set during meta-training. Besides, to evaluate the optimized classifier for a particular N-way problem, another 15 examples from each of the N classes are sampled as the test set (a.k.a. query set in the literature) to provide the loss and supervision. We test models over 10,000 sampled few-shot tasks for a stable measure of performances [25, 39]. The mean accuracy and 95% confidence interval are reported.

B.2 Visualization of learned embeddings on UNSEEN categories

To show the discriminative ability of the learned embedding, we visualize the embedding of 6 randomly selected UNSEEN classes with 50 instances per class from *Mini*ImageNet in Fig. 7. The embedding results of four baseline approaches, namely MC + kNN, ProtoNet + ProtoNet, MC + ProtoNet, and CASTLE are shown. It can be found that CASTLE grasps the instance relationship of UNSEEN classes better than others.

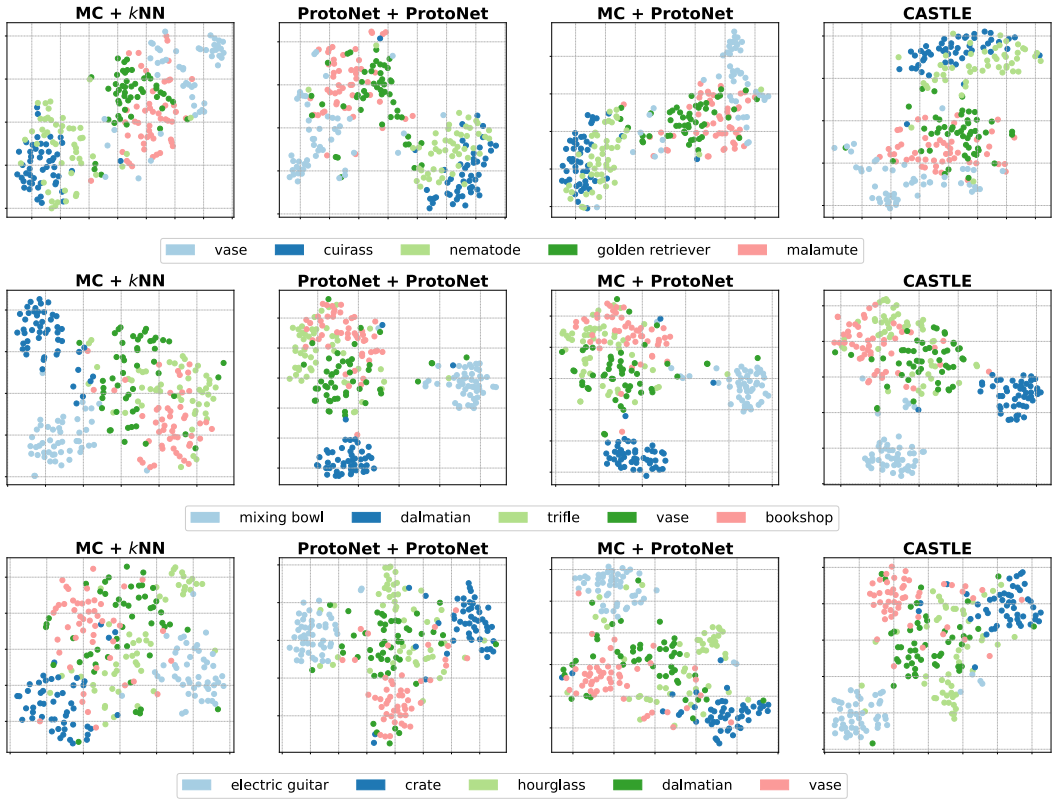


Figure 7: Three groups of embedding visualization results of 6 randomly selected UNSEEN classes. Four baselines are compared. Different colors denote the classes. Best viewed in color.

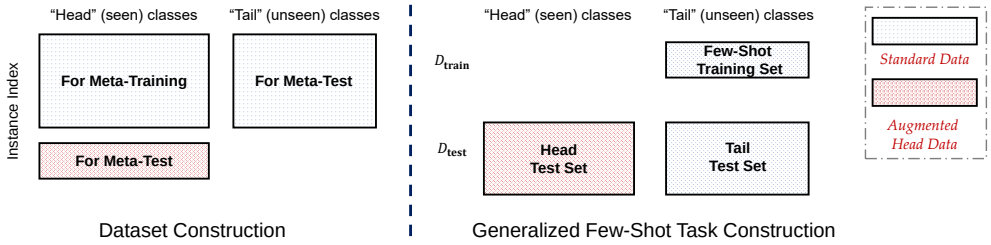


Figure 8: The split of data in the generalized few-shot classification scenario. In addition to the standard data set like *Mini*Imagetnet (blue part), we collect non-overlap augmented "head" class instances from the corresponding categories in the ImageNet (red part), to measure the classification ability on the seen classes. Then in the generalized few-shot classification task, few-shot instances are sampled from each of the unseen classes, while the model should have the ability to make prediction on instances from *both* the "head" and "tail" classes.

C GFSL Details and Additional Results

C.1 Details on Generalized Few-Shot Learning Setups

As mentioned in the main text, now we give the complete details for Generalized FSL setups. To evaluate the ability classifying both SEEN and UNSEEN classes, we compare our CASTLE method with baseline approaches on the previous two data sets. During the model training, ProtoNet + ProtoNet, MC + ProtoNet, and our CASTLE approach are optimized over 5-way few-shot tasks, where the number of shots in a task is consistent with the inference stage.

To test model's discernibility on SEEN class set, we augment the SEEN class set with non-overlap instances drawn from the original ILSVRC-12 dataset [24]. For example, for each of the 64 SEEN classes in the *Mini*ImageNet, we collect 200 more non-overlapping images from ILSVRC-12 as the test set for many-shot classification. An illustration of the data set construction can be found in Fig. 8.

Next, we evaluate a model in three different scenarios. First, the model is used to predict all augmented instances from the head classes. For the embedding-based approaches, we store 100 instances from each SEEN class to compute the class prototype. Then, the few-shot classification results on *all* tail classes are evaluated, with both 1-shot and 5-shot configurations. Last, given few-shot training data from the UNSEEN classes, the model is asked to make prediction over instances from both head and tail classes. We sample 1000 generalized few-shot learning tasks, and there are 15 instances from each class for model evaluation. The top-1 accuracy for instances from SEEN and UNSEEN classes are calculated separately, and their harmonic mean is used as the final criterion, which balances the ability on the two disjoint sets well [38].

C.2 The influence of the Unified Learning Objective

We test the influence of the unified learning objective by test the 1-shot classification performance with different values of λ in Eq. 9 in the main text. By testing the few-shot classification performance on *Mini*ImageNet, we find the larger weights on the Unified Learning Objective (larger λ value) achieves better results.

Table 5: The influence of the Unified Learning Objective with *Mini*ImageNet.

Methods $\rightarrow$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$
1-Shot 5-Way	62.87	63.13	63.51

C.3 Inference with calibration factor on GFSL setups

Table 6: The effect of calibration factor on GFSL inference with *Mini*ImageNet.

Methods $\rightarrow$	MC+kNN	ProtoNet + ProtoNet	MC+ProtoNet	CASTLE
one shot learning				
vanilla inference	0.82	4.88	4.81	6.48
+ calibration	2.04	5.15	4.82	8.62
five shot learning				
vanilla inference	6.27	5.13	6.31	7.43
+ calibration	6.71	6.48	6.56	9.60

As mentioned in the main text, now we show the complete details and more results of the study with regard to the effects of calibration factors. The importance of the calibration factor has already been validated in [7, 37]. We exactly follow the strategy in [7] to complete the calibration by subtracting a bias on the prediction logits of all SEEN classes. In other words, different from the vanilla prediction, a calibration bias is subtracted from the SEEN confidence, to make it balanced with the unseen parts. In detail, we choose the range of the bias by sampling 200 generalized few-shot tasks composed by validation instances and record the difference between the maximum value of SEEN and UNSEEN logits. The averaged difference value is used as the range of the bias selection. 10 equally split

calibration bias values are used as candidates, and the best one is chosen based on 1000 generalized few-shot tasks sampled from the meta-validation set. As a result, we observe that calibrated methods can have a consistent improvement over the harmonic mean of accuracy for SEEN and UNSEEN categories (see Table 6).

C.4 Additional incremental evaluation of generalized few-shot learning.

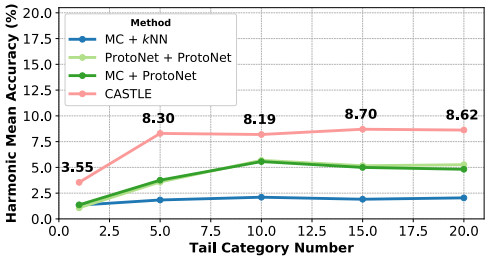


Figure 9: The 1-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

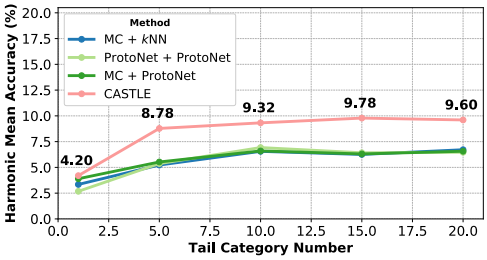


Figure 10: The 5-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

As mentioned in the ablation study of the main text, we now give additional five-shot learning results for the incremental evaluation of the generalized few-shot learning (together with one-shot learning results). In addition to the test instances from the "head" 64 classes in *Mini*ImageNet, 1 to 20 novel classes are included to compose the generalized few-shot tasks. Concretely, 1 or 5 instances per novel class are used to construct the "tail" classifier, combined with which the model is asked to do a *joint* classification of both SEEN and UNSEEN classes. Figure 9 and Figure 10 record the change of generalized few-shot learning performance (harmonic mean) when more UNSEEN classes emerge. We observe that CASTLE consistently outperforms all baseline approaches in each evaluation setup, with a clear margin.

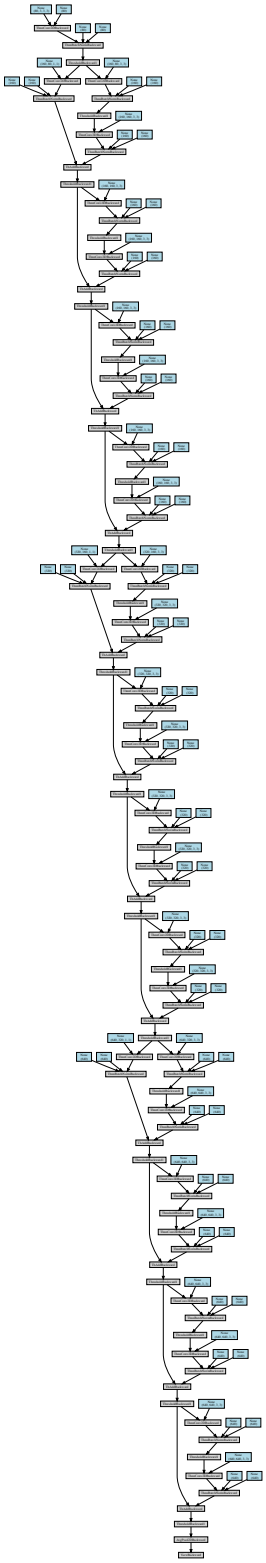


Figure 11: The detailed architecture of ResNet-12 backbone we used. Better perceived when zoomed in.