LEARNING CLASSIFIER SYNTHESIS FOR GENERALIZED FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Object recognition in real-world requires handling long-tailed or even open-ended data. An ideal visual system needs to reliably recognize the populated visual concepts and meanwhile efficiently learn about emerging new categories with a few training instances. Class-balanced many-shot learning and few-shot learning tackle one side of this problem, via either learning strong classifiers for populated categories or learning to learn few-shot classifiers for the tail classes. In this paper, we investigate the problem of *generalized few-shot learning (GFSL)* —- a model during the deployment is required to not only learn about "tail" categories with few shots, but simultaneously classify the "head" and "tail" categories. We propose the Classifier Synthesis Learning (CASTLE), a learning framework that learns how to synthesize calibrated few-shot classifiers in addition to the multi-class classifiers of "head" classes, leveraging a shared neural dictionary. CASTLE sheds light upon the inductive GFSL through optimizing one clean and effective GFSL learning objective. It demonstrates superior performances than existing GFSL algorithms and strong baselines on *Mini*ImageNet and *Tiered*ImageNet data sets. More interestingly, it outperforms previous state-of-the-art methods when evaluated on standard few-shot learning.

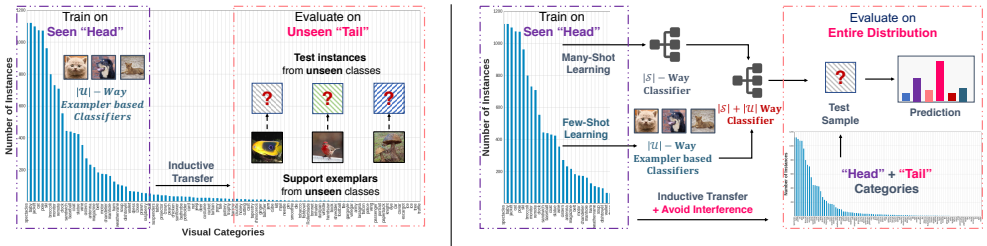
1 INTRODUCTION

Visual recognition for objects in the "long tail" has been an important challenge to address (Wang et al., 2017; Liu et al., 2019). 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 (*e.g* deep learning) can not be directly applied due to their notorious demand of a large number of annotated data (Krizhevsky et al., 2017; Simonyan & Zisserman, 2014; He et al., 2016).

Few-shot learning (FSL) (Vinyals et al., 2016; Snell et al., 2017; Finn et al., 2017) is mindful of the limited instances (*i.e*, shots) per "tail" concept, which attempts to address this challenging problem by distinguishing between the data-rich "head" categories as SEEN classes and data-scarce "tail" categories as UNSEEN classes. While it is difficult to build classifiers with data from UNSEEN classes, FSL leverages data from SEEN classes to extract inductive biases for effective classifiers acquisition on UNSEEN ones. We refer to (Larochelle, 2018) 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 study 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 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, contrasting the standard few-shot learning. In contrast to prior works (Hariharan & Girshick, 2017; Wang et al., 2017; Liu et al., 2019) that focus on learning "head" and "tail" concepts in a transductive manner, our learning setup requires inductive modeling of the"tail", which is therefore more challenging as we assume no knowledge about the UNSEEN "tail" categories is available during the model learning phase.



(a) Few-shot learning (FSL) **(b)** Generalized Few-shot Learning (GFSL)

Figure 1: A conceptual diagram comparing the Few-Shot Learning (FSL) and the Generalized Few-Shot Learning (GFSL). GFSL requires to extract inductive bias from SEEN categories to facilitate efficiently learning on few-shot UNSEEN "tail" categories, while maintaining discernability on "head" classes.

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 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 the remaining 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 be competitive when used in conjunction with many-shot classifiers of populated classes*. We argue that such highly contrastive learning can benefit few-shot classification with high discernibility in its learned visual embeddings (cf. Section 4.2 and Section 4.4).

We empirically validate our approach on two standard benchmark data sets — *Mini*ImageNet and *Tiered*ImageNet. The proposed approach retains competitive "head" concept recognition performances while outperforming existing approaches on few-shot learning and *generalized* few-shot learning. We highlight that CASTLE has learned a better calibration between many-shot SEEN classifiers and synthesized UNSEEN classifiers, which naturally addresses the confidence mismatch phenomena (Chao et al., 2016), *i.e*, SEEN and UNSEEN classifiers have different confidence ranges.

2 PROBLEM DESCRIPTION

We define a K-shot N-way classification task to be one with N classes to make prediction and K training examples per class for learning. The training set (*i.e*, support set) is represented as $\mathcal{D}_{\mathbf{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{K}$, where $\mathbf{x}_i \in \mathbb{R}^D$ is an instance and $\mathbf{y}_i \in \{0,1\}^N$ (*i.e*, one-hot vector) is its label. Similarly, the test set is $\mathcal{D}_{\mathbf{test}}$ and contains *i.i.d.* samples from the same distribution as $\mathcal{D}_{\mathbf{train}}$.

From few-shot learning to generalized few-shot learning. In *many-shot learning*, where K is large, a classification model $f : \mathbb{R}^D \to \{0,1\}^N$ is learned by optimizing $\mathbb{E}_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}_{\mathbf{train}}}\ell(f(\mathbf{x}_i), \mathbf{y}_i)$. Here f is often instantiated as an embedding function $\phi(\cdot)$ and a linear classifier $\mathbf{\Theta}$: $f(\mathbf{x}_i) = \phi(\mathbf{x}_i)^\top \mathbf{\Theta}$. The loss function $\ell(\cdot,\cdot)$ measures the discrepancy between the prediction and the true label. On the other hand, *Few-shot learning (FSL)* faces the challenge in transferring knowledge across learning visual concepts. It assumes two non-overlapping sets of SEEN ($\mathcal{S}$) and UNSEEN ($\mathcal{U}$) classes. During training, it has access to all SEEN classes for learning an inductive bias, which is then transferred to learn good classifiers on $\mathcal{U}$ rapidly with a small K. *Generalized Few-Shot Learning (GFSL)*, different from FSL which neglects classification of the $\mathcal{S}$ classes, aims at building models that simultaneously predicts over $\mathcal{S} \cup \mathcal{U}$ categories. As a result, such a model needs to deal with many-shot classification from $|\mathcal{S}|$ SEEN classes along side with learning $|\mathcal{U}|$ emerging UNSEEN classes [1].

Meta-learning for few-shot learning. Meta-learning has been an effective framework for FSL (Vinyals et al., 2016; Finn et al., 2017; Snell et al., 2017) in the recent years. The main idea is to *mimic* the future few-shot learning scenario by optimizing a shared f across K-shot N-way tasks drawn from the SEEN class sets $\mathcal{S}$.

$$\min_f \mathbb{E}_{(\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}, \mathcal{D}^{\mathcal{S}}_{\mathbf{test}})\sim\mathcal{S}} \mathbb{E}_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}} \left[\ell\left(f\left(\mathbf{x}_j; \mathcal{D}^{\mathcal{S}}_{\mathbf{train}}\right), \mathbf{y}_j\right)\right] \tag{1}$$

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

In particular, a K-shot N-way task $\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}$ sampled from $\mathcal{S}$ is constructed by randomly choosing N categories from $\mathcal{S}$ and K examples in each of them. A corresponding test set $\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}$ (a.k.a. query set) is sampled from $\mathcal{S}$ to evaluate the resulting few-shot classifier f. Therefore, we expect the learned classifier f "generalizes" well on the training few-shot tasks sampled from SEEN classes, to "generalize" well on few-shot tasks drawn from UNSEEN class set $\mathcal{U}$.

In this paper, we focus on the methods described in (Vinyals et al., 2016; Snell et al., 2017). 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 pushing dissimilar ones far away (Koch et al., 2015). For a test instance $\mathbf{x}_j$, the embedding function ϕ makes a prediction based on a soft nearest neighbor classifier:

$$\hat{y}_j = f\left(\mathbf{x}_j; \mathcal{D}_{\mathbf{train}}\right) = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\mathbf{train}}} \mathbf{sim}\left(\phi(\mathbf{x}_j), \phi(\mathbf{x}_i)\right) \cdot \mathbf{y}_i \tag{2}$$

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

3 METHOD

The main idea of CASTLE includes a classifier composition model for synthesizing classifiers with the few-shot training data, and an effective learning algorithm that learns many-shot classifiers and few-shot classifiers (together with its composition model end-to-end) at the same time. In Section 3.1, we introduce the classifier composition model uses a few-shot training data to query a common set of neural bases, and then assemble the target "synthesized classifiers". In Section 3.2, we propose a unified learning objective that directly contrasts many-shot classifiers with few-shot classifiers, via constructing classification tasks over $\mathcal{U} \cup \mathcal{S}$ categories. It enforces the few-shot classifiers to explicitly compete against the many-shot classifiers in the model learning, which leads to more discriminative few-shot classifiers in the GFSL setting.

3.1 CLASSIFIER COMPOSITION WITH A NEURAL DICTIONARY

We base our classifier composition model on (Changpinyo et al., 2016; 2018). Different from their approach with a pre-fixed feature embedding, we use a *learned* embedding function and a neural dictionary. Here we define a dictionary as pairs of "key" and "value" embeddings, where each "key" and "value" is associated with a neural 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 $\mathbf{b}_k \in \mathcal{B} \in \mathbb{R}^d$. The key and value for the dictionary are generated based on two linear projections U and V of elements in $\mathcal{B}$. For instance, $\mathrm{U}\mathbf{b}_i$ and $\mathrm{V}\mathbf{b}_i$ represent the generated key and value embeddings. Denote $\mathbb{I}[\,\mathbf{y}_i = c\,]$ as an indicator that selects instances in the class c. 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):[2]

$$\mathbf{p}_c = \frac{1}{K} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\mathbf{train}}} \phi\left(\mathbf{x}_i\right) \cdot \mathbb{I}[\,\mathbf{y}_i = c\,] \tag{3}$$

We then compute the coefficients α_c for assembling the classifier of class c, via measuring the compatibility score between the class signature and the key embeddings of the neural dictionary,

$$\alpha_c^k \propto \exp\left(\mathbf{p}_c^\top \mathrm{U}\mathbf{b}_k\right), \text{where } \; k = 1, \cdots, |\mathcal{B}| \tag{4}$$

The coefficient α_c^k 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,

$$\mathbf{w}_c = \mathbf{p}_c + \sum_{k=1}^{|\mathcal{B}|} \alpha_c^k \cdot \mathrm{V}\mathbf{b}_k \tag{5}$$

[2]More choices of Eq. 3 are investigated in Section C.1 in the supplementary.

We formulate the classifier composition as a summation of the initial prototype embedding $\mathbf{p}_c$ and the residual component $\sum_{k=1}^{|\mathcal{B}|} \alpha_c^k \cdot \mathrm{U}_V \mathbf{b}_k$. 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 compositions of emerging visual categories.

3.2 Unified Learning of Few-Shot and Many-Shot Classifiers

In addition to transferring knowledge from SEEN to UNSEEN classes as in FSL, in generalized few-shot learning, the few-shot classifiers is required to do well when used in conjunction with many-shot classifiers. Therefore, a GFSL classifier f should have a low expected error as what follows:

$$\mathbb{E}_{\mathcal{D}^{\mathcal{U}}_{\mathbf{train}}} \mathbb{E}_{(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}^{\mathcal{S}\cup\mathcal{U}}_{\mathbf{test}}} \Big[\ell \left(f \left(\mathbf{x}_j; \mathcal{D}^{\mathcal{U}}_{\mathbf{train}}, \mathbf{\Theta}_{\mathcal{S}} \right), \mathbf{y}_j \right) \Big] \tag{6}$$

Suppose we have sampled a K-shot N-way few-shot learning task $\mathcal{D}^{\mathcal{U}}_{\mathbf{train}}$, which contains $|\mathcal{U}|$ visual UNSEEN categories. For each task, the classifier f predicts a test instance in $\mathcal{D}^{\mathcal{S}\cup\mathcal{U}}_{\mathbf{test}}$ towards *both* tail classes $\mathcal{U}$ and head classes $\mathcal{S}$. In other words, based on $\mathcal{D}^{\mathcal{U}}_{\mathbf{train}}$ and the many-shot classifiers $\mathbf{\Theta}_{\mathcal{S}}$, a randomly sampled instance in $\mathcal{S} \cup \mathcal{U}$ should be effectively predicted. In summary, a GFSL classifier generalizes its joint prediction ability to $\mathcal{S} \cup \mathcal{U}$ given $\mathcal{D}^{\mathcal{U}}_{\mathbf{train}}$ and $\mathbf{\Theta}_{\mathcal{S}}$ during inference.

Unified learning objective. CASTLE learns a generalizable GFSL classifier via training on the SEEN class set $\mathcal{S}$. For each class in $s \in \mathcal{S}$, it keeps many-shot classifiers (*i.e*, liner classifier over the embedding function $\phi(\cdot)$) $\mathbf{\Theta}_s$. Next, we sample a "fake" K-shot N-way few-shot task from $\mathcal{S}$, which contains $\mathcal{C}$ categories. For each classes in $\mathcal{C}$, we synthesize their classifiers by $\mathbf{W}_{\mathcal{C}} = \{ \mathbf{w}_c \mid c \in \mathcal{C} \}$ as in Eq. 5. We treat the remaining $\mathcal{S} - \mathcal{C}$ classes as the "fake" head classes, and use their corresponding many-shot classifiers $\mathbf{\Theta}_{\mathcal{S}-\mathcal{C}}$. They are combined with the synthesized classifiers $\mathbf{W}_{\mathcal{C}}$ (from the few-shot classes $\mathcal{C}$) to form the set of joint classifiers $\hat{\mathbf{W}} = \mathbf{W}_{\mathcal{C}} \cup \mathbf{\Theta}_{\mathcal{S}-\mathcal{C}}$, over *all* classes in $\mathcal{S}$. Finally, we optimize the learning objective as what follows:

$$\min_{\{\phi, \mathcal{B}, \{\mathbf{\Theta}_s\}, \mathrm{U}, \mathrm{V}\}} \sum_{\mathcal{C} \subset \mathcal{S}} \sum_{(\mathbf{x}_j, \mathbf{y}_j) \sim \mathcal{S}} \ell \Big(\hat{\mathbf{W}}^\top \phi(\mathbf{x}_j), \mathbf{y}_j \Big) \tag{7}$$

Despite that few-shot classifiers $\mathbf{W}_{\mathcal{C}}$ are synthesized using with K training instances (cf. Eq. 3), they are optimized to jointly classify instances from all SEEN categories $\mathcal{S}$. After minimizing the accumulated loss in Eq. 7 over multiple GFSL tasks, the learned model extends its discerning ability to UNSEEN classes so as has low error in Eq. 6. During inference, CASTLE synthesizes the classifiers for UNSEEN classes based on the neural dictionary with their few-shot training examples, and makes a joint prediction over $\mathcal{S} \cup \mathcal{U}$ with the help of many-shot classifier $\mathbf{\Theta}_{\mathcal{S}}$.

Multi-classifier learning. A natural way to minimize Eq. 7 implements a stochastic gradient descent step in each mini-batch by sampling one GFSL task, which contains a K-shot N-way training set together with a set of test instances $(\mathbf{x}_j, \mathbf{y}_j)$ from $\mathcal{S}$. It is clear that increasing the number of GFSL tasks per gradient step can improve the optimization stability. Therefore, we propose an efficient implementation that utilizes *a large number of* GFSL tasks to compute gradients. Specifically, we sample two sets of instances from *all* SEEN classes, *i.e*, $\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}$ and $\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}$. Then we construct a large number of joint classifiers $\{\hat{\mathbf{W}}^z = \mathbf{W}^z_{\mathcal{C}} \cup \mathbf{\Theta}^z_{\mathcal{S}-\mathcal{C}} \mid z = 1, \cdots, Z\}$ with different sets of $\mathcal{C}$, which is then applied to compute the averaged loss over z using Eq. 7. In the scope of this paper, CASTLE always uses *multi-classifier learning* unless it is explicitly mentioned. With this, we observed a significant speed-up in terms of convergence (cf. Section C.1 in the appendix for an ablation study).

4 Experiments

In this section, we design experiments to validate the effectiveness of the CASTLE in GFSL (cf. Section 4.2). We first introduce the training and evaluation protocol of Ren et al. (2018a) and compare CASTLE with existing methods. Next, we provide an analysis over algorithms with alternative protocols that measures different aspects of GFSL (cf. Section 4.3). We verify that CASTLE is

advantageous as it learns a better calibration between SEEN and UNSEEN classifiers. Finally, we show that CASTLE also benefit standard FSL performances (cf. Section 4.4).

4.1 EXPERIMENTAL SETUPS

Data sets. We consider two benchmark data sets derived from ILSVRC-12 dataset (Russakovsky et al., 2015). The *mini*ImageNet dataset (Vinyals et al., 2016) has 100 classes and 600 examples per class. For evaluation, we follow the split of (Ravi & Larochelle, 2017) and use 64 of 100 classes as SEEN, 16 for UNSEEN validation, and 20 for UNSEEN test. The *Tiered*ImageNet (Ren et al., 2018b) contains 34 super-categories in total, where 20 of them are SEEN, 6 and 8 are UNSEEN validation and test. This challenging setup results in 351, 97, and 160 fine-grained classes for SEEN, UNSEEN validation and test, respectively. To evaluate performance on SEEN classes, we use 400 non-overlapping images per category from ILSVRC-12 as validation and test images for SEEN classes. Figure A5 of the Appendix provides an illustration of how data are split.

Baselines and prior methods. We explore several (strong) choices in deriving classifiers for the SEEN and UNSEEN classes: *(1) Multiclass Classifier (MC) + kNN.* A multi-class classifier is trained on the SEEN classes as standard many-shot classification (He et al., 2016). 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 (Snell et al., 2017) (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, and use the final embedding function for UNSEEN classes. The prediction is straightforward as both sets of classes are generated with ProtoNet. *(3) MC + ProtoNet.* We combine the learning objective of (1) and (2) to jointly learn the MC classifier and feature embedding, which trades off between few-shot and many-shot learning.

Besides IFSL (Ren et al., 2018a), we also re-implemented existing approaches (or adapted the original release if available), *i.e*, L2ML′ (Wang et al., 2017) and DFSL′ (Gidaris & Komodakis, 2018) to compare with CASTLE. Note that L2ML is originally designed in the transductive setting, which we made some adaption for inductive prediction. Please refer to original papers for details. For CASTLE, we use the $\{\Theta_{\mathcal{S}}\}$ (*i.e*, the multiclass classifiers, cf. Section 3.2) for the SEEN classes 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.

Evaluation measures. Mean accuracy over all SEEN and 5 sampled UNSEEN classes is the main measurement to evaluate a GFSL method (Gidaris & Komodakis, 2018; Wang et al., 2018). We sample 10,000 1-shot or 5-shot GFSL tasks to evaluate this for the sake of reliability. Besides the few-shot training examples, an equal number of test instances sampled from all head and 5 tail categories are used during the evaluation. The mean and 95% confidence interval are reported. In addition to accuracy, Ren et al. (2018a) also use Δ-value, a measure of average accuracy drop between predicting specific (SEEN or UNSEEN) class and predicting all categories jointly. Methods balance the prediction of SEEN and UNSEEN classes well can receive a low accuracy drop. In the later sections, we introduce two other GFSL measures —- the harmonic mean accuracy and the area under SEEN-UNSEEN curve (AUSUC).

Please refer to the Section A of the Appendix for more details about experimental setups, implementation details, model optimization, and evaluation measures [3].

4.2 MAIN RESULTS

The main results of all methods on *mini*ImageNet is shown in Table 1. We found that CASTLE outperforms all the existing methods as well as our proposed baseline systems in terms of the mean accuracy. Meanwhile, when looked at the Δ-value, CASTLE is least affected between predicting for SEEN/USSEEN classes separately and predicting over all classes jointly. However, we argue that either mean accuracy or Δ-value is not informative enough to tell about a GFSL algorithm's performances. For example, a baseline system, *i.e*, ProtoNet + ProtoNet perform better than IFSL in terms of 5-shot mean accuracy but not Δ-value. *In this case, how shall we rank these two systems?* To answer this

[3] Our implementation is publicly available on https://www.anonymous.com

Table 1: Generalized Few-shot classification performance (accuracy, Δ-value, and harmonic mean accuracy) on *Mini*ImageNet when there are **64 Head and 5 Tail categories**.

Setups→	1-Shot		5-Shot		1-Shot	5-Shot
Perf. Measures→	*Mean Acc* ↑	Δ ↓	*Mean Acc* ↑	Δ ↓	*Harmonic Mean Acc* ↑	
IFSL (Ren et al., 2018a)	54.95±0.30	11.84	63.04±0.30	10.66	-	-
L2ML′	46.25±0.04	27.49	45.81±0.03	35.53	2.98±0.06	1.12±0.04
DFSL′	61.00±0.11	13.28	72.84±0.09	10.58	59.96±0.13	72.42±0.09
MC + kNN	46.96±0.03	27.19	45.50±0.03	38.45	0.00±0.00	0.00±0.00
MC + ProtoNet	45.21±0.03	30.72	45.52±0.03	38.94	0.00±0.00	0.00±0.00
ProtoNet + ProtoNet	53.93±0.08	22.09	72.64±0.08	11.41	27.73±0.19	68.99±0.11
Ours: CASTLE	**66.48**±0.11	**9.94**	**76.25**±0.09	**8.14**	**64.29**±0.14	**75.79**±0.10

Table 2: 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 84 Categories**	
Setups →	Many-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Perf. Measures →	*Mean Acc.*	*Mean Acc.*		*Harmonic Mean Acc.*	
L2ML′	90.99	27.79± 0.73	43.42± 0.63	1.27± 0.09	2.38± 0.02
DFSL′	90.35 / 90.76	30.03± 0.75	49.10± 0.63	38.07± 0.06	55.54± 0.05
MC + kNN	90.99	27.91± 0.73	50.98± 0.64	0.00± 0.00	0.00± 0.00
MC + ProtoNet	90.39 / 90.27	30.89± 0.62	51.76± 0.62	0.00± 0.00	0.00± 0.00
ProtoNet + ProtoNet	87.03 / 88.47	30.54± 0.77	51.64± 0.62	24.81± 0.08	55.90± 0.06
Ours: CASTLE	91.23 / 91.28	**33.42**± 0.75	**52.95**± 0.62	**40.73**± 0.07	**57.78**± 0.07

question, we propose to use another evaluation measure, harmonic mean of the mean accuracy for each SEEN and UNSEEN category, when they are classified jointly.

Harmonic mean is a better GFSL performance measure. Since the number of SEEN and UNSEEN classes are most likely to be not equal, *e.g*, 64 vs. 5 in our cases, directly computing the mean accuracy over all classes is almost always biased. For example, a many-shot classifier that only classifies samples into SEEN classes can receive a good performance than one that recognizes both SEEN and UNSEEN. Therefore, we argue that *harmonic mean* over the mean accuracy can better assess a classifier's performance, as now the performances are negatively affected when a classifier ignores classes (*e.g*, MC classifier get 0% harmonic mean). Specifically, we compute the top-1 accuracy for instances from SEEN and UNSEEN classes, and take their harmonic mean as the performance measure. The results are included in the right side of the Table 1. Now we observe that the many-shot baseline MC+kNN has extremely low performance as it tends to ignore UNSEEN categories. Meanwhile, CASTLE remains the best when ranked by the harmonic mean accuracy against others.

Evaluate GFSL beyond 5 UNSEEN categories. Besides using harmonic mean accuracy, we argue that another important aspect in evaluating GFSL is to go beyond the 5 sampled UNSEEN categories, as it is never the case in real-world. On the contrary, we care most about the GFSL with a large number of UNSEEN classes. To this end, we evaluate GFSL with all available SEEN and UNSEEN categories over both *Mini*ImageNet and *Tiered*ImageNet, and report their results in Table 2 and Table 3. We report the mean accuracy over SEEN and UNSEEN categories, as well as the harmonic mean accuracy of *all* categories. We observe that CASTLE outperforms all approaches in the UNSEEN and more importantly, the ALL categories section, across two data sets. On the SEEN categories, CASTLE remains competitive against the ad hoc many-shot classifier (MC).

4.3 ANALYSIS

In this section, we do analyses to show (1) tuning a great confidence calibration factor significantly improves GFSL performance of baseline models, (2) CASTLE has balanced the confidence score of SEEN and UNSEEN predictions, requiring no explicit calibration, and (3) CASTLE is consistently better than other approaches across an increasing number of "tail" categories. For more ablation studies about CASTLE, we refer readers to the Appendix (cf. Section C.1).

Table 3: 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 511 Categories**	
Setups →	Many-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Perf. Measures →	*Mean Acc.*	*Mean Acc.*		*Harmonic Mean Acc.*	
DFSL′	62.23 / 63.92	14.56± 0.42	28.35± 0.40	12.60± 0.11	19.29± 0.05
MC + kNN	63.92	12.37± 0.41	25.70± 0.40	0.01± 0.00	0.01± 0.00
MC + ProtoNet	57.74 / 60.95	12.84± 0.41	26.89± 0.42	0.00± 0.00	0.00± 0.00
ProtoNet + ProtoNet	54.60 / 58.13	12.98± 0.42	27.00± 0.41	6.84± 0.05	28.66± 0.07
Ours: CASTLE	59.96 / 61.85	**14.86**± 0.20	**28.55**± 0.41	**17.89**± 0.51	**29.96**± 0.05

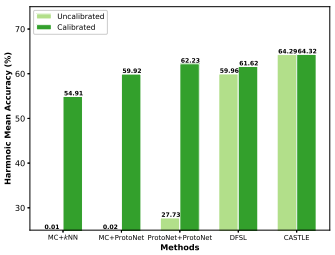


Figure 2: Calibration's effect to the 1-shot harmonic mean accuracy on *Mini*ImageNet.

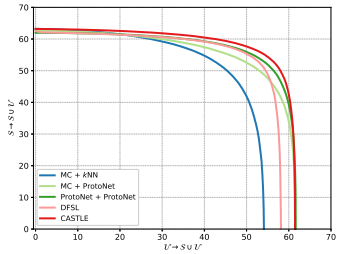


Figure 3: The 1-shot AUSUC performance with two configurations of UNSEEN classes on *Mini*ImageNet.

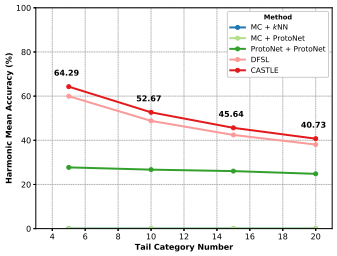


Figure 4: The 1-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

Confidence calibration matters in GFSL. In generalized zero-shot learning, Chao et al. (2016) has identified a significant prediction bias between classification confidence of SEEN and UNSEEN classifiers. We find a similar phenomena in GFSL. For instance, *ProtoNet + ProtoNet* baseline has a very confident classifier on SEEN categories than UNSEEN categories (The scale of confidence is on average 2.1 times higher). To address this issue, we compute a calibration factor based on the validation set of UNSEEN categories, such that the prediction logits are calibrated by subtracting this factor out from the confidence of SEEN categories' predictions. The results of all methods after calibration is shown in Figure 2. We observe a consistent improvement over the harmonic mean of accuracy for all methods, while CASTLE is the least affected. This suggests that CASTLE, learned with the unified GFSL objective, has a well-calibrated classification confidence and does not require additional data and extra learning phase to search this calibration factor.

Moreover, we use area under SEEN-UNSEEN curve (AUSUC) as a measure of different GFSL algorithms. Here, AUSUC is a performance measure that takes the effects of calibration factor out. To do so, we enumerate through a large range of calibration factors, and subtract it from the confidence score of SEEN classifiers. Through this process, the joint prediction performances over SEEN and UNSEEN categories, denoted as $\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$ and $\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$, shall vary as the calibration factor changes. For instance, when calibration factor is infinite large, we are measuring a classifier that only predicts UNSEEN categories. We denote this as the SEEN-UNSEEN curve. The results is shown in Figure 3. As a result, we observe that CASTLE archives the largest area under curve, which indicates that CASTLE is in general a better algorithm over others among different calibration factors.

Robust evaluation of GFSL. Other than the harmonic mean accuracy of all SEEN and UNSEEN categories shown in cf. Table 2 and 3, we study the dynamic of how harmonic mean accuracy changes with an incremental number of UNSEEN "tail" concepts. 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 1-shot learning result is shown as Figure 4. We observe that CASTLE consistently outperforms other baselines by a clear margin.

4.4 STANDARD FEW-SHOT LEARNING

Finally, we also evaluate our proposed approach's performance on two standard few-shot learning benchmarks, *i.e*, *mini*ImageNet and *Tiered*ImageNet data set. The results are shown in the Table 4 and Table 5. We compare our approach to previous state-of-the-art methods and found CASTLE

Table 4: Few-shot classification accuracy on *Mini*ImageNet with the ResNet-12 backbone.

Setups →	1-Shot 5-Way	5-Shot 5-Way
ProtoNet (Snell et al., 2017)	61.40 ± 0.02	76.56 ± 0.02
LEO (Rusu et al., 2018)	61.76 ± 0.08	77.59 ± 0.12
OptNet (Lee et al., 2019)	62.64 ± 0.61	78.63 ± 0.46
FEAT (Ye et al., 2018)	62.96 ± 0.02	78.49 ± 0.02
Ours: CASTLE	**63.06** ± 0.02	**79.33** ± 0.01

Table 5: Few-shot classification accuracy on *Tiered*ImageNet with the ResNet-12 backbone.

Setups →	1-Shot 5-Way	5-Shot 5-Way
ProtoNet (Snell et al., 2017)	53.31 ± 0.89	72.69 ± 0.74
RelationNet Sung et al. (2018)	54.48 ± 0.93	71.32 ± 0.78
LEO (Rusu et al., 2018)	66.33 ± 0.05	81.44 ± 0.09
OptNet (Lee et al., 2019)	65.99 ± 0.72	81.56 ± 0.63
Ours: CASTLE	**69.06** ± 0.02	**83.99** ± 0.02

outperforming all of them, in both 1-shot 5-way and 5-shot 5-way accuracy. This supports our hypothesis that jointly learning with many-shot classification forces few-shot classifiers to be discriminative. Please refer to the Appendix for details about task setups, performance measures, and visualizations.

5 Related Work and Discussion

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 (Russakovsky et al., 2015; Krizhevsky et al., 2017). 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.

Few-shot learning (FSL) is proposed to tackle this problem, via meta-learning an inductive bias from the SEEN classes, such that it transfers to the learning process of UNSEEN classes with few training data during the model deployment. For example, one line of works uses meta-learned discriminative feature embeddings (Snell et al., 2017; Oreshkin et al., 2018; Rusu et al., 2018; Scott et al., 2018; Ye et al., 2018; Lee et al., 2019) together with non-parametric nearest neighbor classifiers, to recognize novel classes given a few exemplars. Another line of works (Finn et al., 2017; Nichol et al., 2018; Lee & Choi, 2018; Antoniou et al., 2018; Vuorio et al., 2018) chooses 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 (Hariharan & Girshick, 2017; Wang et al., 2018; Gao et al., 2018; Ye et al., 2018; Liu et al., 2019). 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 mostly focus on the transductive setup of GFSL. Some of them (Hariharan & Girshick, 2017; Wang et al., 2018; Gao et al., 2018) apply the exemplar-based classification paradigms on both SEEN and UNSEEN categories to resolve the transductive learning problem. Others (Wang et al., 2017; Schönfeld et al., 2018; Liu et al., 2019) usually ignore the explicit relationship between SEEN and UNSEEN categories, and learn separate classifiers. Ren et al. (2018a); Gidaris & Komodakis (2018) propose to solve inductive GFSL via either composing UNSEEN with SEEN classifiers or meta-leaning with recurrent back-propagation procedure. Gidaris & Komodakis (2018) is the most related work to CASTLE, where we differ in how we compose classifiers and the unified learning objective, *i.e*, we used a learned neural dictionary instead of using MC classifiers as bases.

In summary, CASTLE learns both many-shot classifiers and synthesized classifiers via optimizing a single unified objective function, where a classifier composition model with a neural dictionary is leveraged for assembling few-shot classifiers. Our experiments highlight that CASTLE not only outperforms existing methods in terms of GFSL performances from many different aspects, but more interestingly, also improves the classifier's discernibility over standard FSL.

REFERENCES

Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. How to train your MAML. *CoRR*, abs/1810.09502, 2018. 8

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*, pp. 5327–5336, Las Vegas, NV, 2016. 3

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Classifier and exemplar synthesis for zero-shot learning. *CoRR*, abs/1812.06423, 2018. 3

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*, pp. 52–68, Amsterdam, The Netherlands, 2016. 2, 7, 15, 20

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*, pp. 1126–1135, Sydney, Australia, 2017. 1, 2, 8, 15

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*, pp. 983–993. 2018. 8

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, Salt Lake City, UT., 2018. 5, 8, 12, 14

Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision*, pp. 3037–3046, Venice, Italy, 2017. 1, 8

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*, pp. 770–778, 2016. 1, 5, 11, 12, 13

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 8

Hugo Larochelle. Few-shot learning with meta-learning: Progress made and challenges ahead. 2018. 1

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019. 8, 11

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*, pp. 2933–2942, Stockholm, Sweden, 2018. 8

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. 1, 8

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 8

Boris N. Oreshkin, Pau Rodríguez, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. *CoRR*, abs/1805.10123, 2018. 8

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. *CoRR*, abs/1706.03466, 2017. 11, 12

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations*, 2017. 5, 11

Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S. Zemel. Incremental few-shot learning with attention attractor networks. *CoRR*, abs/1810.07218, 2018a. 4, 5, 6, 8, 15

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, 2018b. 5, 11

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. 5, 8, 11

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. 8, 11, 12

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. 8

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*, pp. 76–85. 2018. 8

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, pp. 4080–4090. Curran Associates, Inc., 2017. 1, 2, 3, 5, 8, 13, 15

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*, pp. 1199–1208, Salt Lake City, UT, 2018. 8

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*, pp. 3630–3638. Curran Associates, Inc., 2016. 1, 2, 3, 5, 11, 15

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*, 2018. 8

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems 30*, pp. 7032–7042. Curran Associates, Inc., 2017. 1, 5, 8, 12, 14

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*, pp. 7278–7286, Salt Lake City, UT., 2018. 5, 8, 20

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*, pp. 3077–3086, Honolulu, HI, 2017. 15

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *CoRR*, abs/1812.03664, 2018. 8, 11, 12, 15

APPENDIX

A IMPLEMENTATION DETAILS

Following the recent methods (Qiao et al., 2017; Rusu et al., 2018; Ye et al., 2018), we use a residual network (He et al., 2016) (ResNet) to implement the embedding backbone ϕ. We first pre-train this backbone network (also explored by (Qiao et al., 2017; Rusu et al., 2018; Ye et al., 2018; Lee et al., 2019)) and perform model selection strategy similar to (Ye et al., 2018). To learn our methods as well as baseline systems, we then use Momentum SGD with an initial learning rate 1e-4. In the rest of this section, we explain each of the above with complete details.

A.1 DATA SET DETAILS.

Two benchmark data sets are used in our experiments.

The ***Mini*ImageNet** dataset (Vinyals et al., 2016) is a subset of the ILSVRC-12 dataset (Russakovsky et al., 2015). There are totally 100 classes and 600 examples in each class. For evaluation, we follow the split of (Ravi & Larochelle, 2017) 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** (Ren et al., 2018b) 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.

Since both data sets are constructed by images from ILSVRC-12, we augment the *meta-train* set of each data set by sampling non-overlapping images from the corresponding classes in ILSVRC-12. The auxiliary meta-train set is used to measure the generalized few-shot learning classification performance on the SEEN class set. For example, for each of the 64 SEEN classes in the *Mini*ImageNet, we collect 200 more non-overlapping images per class from ILSVRC-12 as the test set for many-shot classification. An illustration of the data set split is shown in Figure A5.

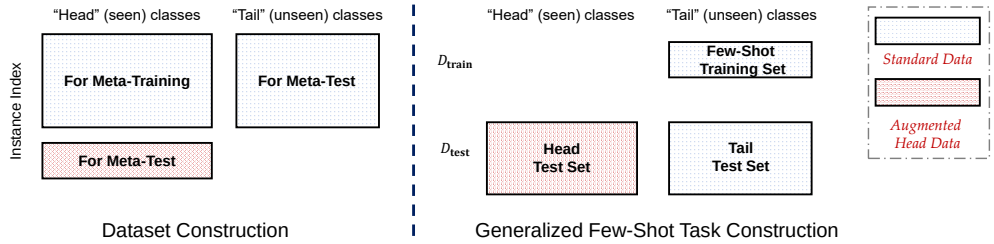


Figure A5: The split of data in the generalized few-shot classification scenario. In addition to the standard data set like *Mini*Imagetnet (blue part), we collect non-overlapping augmented "head" class instances from the corresponding categories in the ImageNet (red part), to measure the classification 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 predict instances from *both* the "head" and "tail" classes.

A.2 Feature network specification.

Following the setting of most recent methods (Qiao et al., 2017; Rusu et al., 2018; Ye et al., 2018), we use the residual network (He et al., 2016) to implement the embedding backbone ϕ. Different from the standard configuration, the literature (Qiao et al., 2017; Rusu et al., 2018; Ye et al., 2018) 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 A15. Please refer to Pytorch documentation [4] for complete references of each building blocks.

A.3 Pre-training strategy.

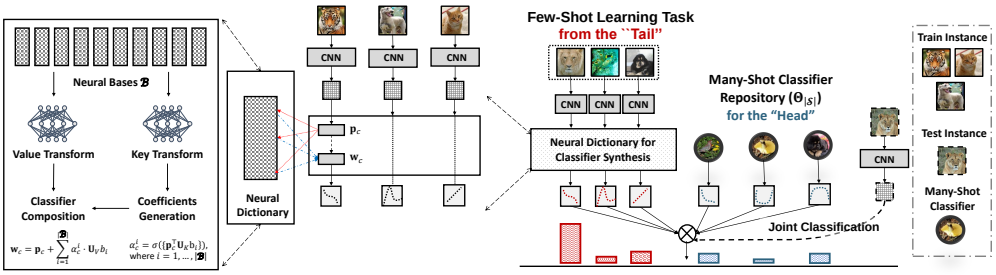
Before the meta-training stage, we try to find a good initialization for the embedding ϕ. In particular, on *MiniImageNet* we add a linear layer on the backbone output and optimize a 64-way (while 351-way for *Tiered*ImageNet) classification problem on the meta-training set with the cross-entropy loss function. Stochastic gradient descent with initial learning rate 0.1 and 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. In later sections, we will show the effect of pre-training strategy on both few-shot and generalized few-shot classification measures.

A.4 Default Training Strategy.

We use the pre-trained backbone to initialize the embedding part ϕ of a model for CASTLE and our re-implemented comparison methods such as MC+kNN, ProtoNet+ProtoNet, MC+ProtoNet, L2ML (Wang et al., 2017), and DFSL (Gidaris & Komodakis, 2018). When there exists a backbone initialization, we set the initial learning rate as 1e-4 and optimize the model with Momentum SGD. The learning rate will be halved after optimizing 2,000 mini-batches. During meta-learning, all methods are optimized over 5-way few-shot tasks, where the number of shots in a task is consistent with the inference (meta-test) stage. For example, if the goal is a 1-shot 5-way model, we sample 1-shot 5-way $\mathcal{D}^{\mathcal{S}}_{\mathbf{train}}$ during meta-training, together with 15 instances per class in $\mathcal{D}^{\mathcal{S}}_{\mathbf{test}}$.

An illustration of the architecture of CASTLE is shown in Figure A6. For CASTLE, we randomly sample a 24-way task from $\mathcal{S}$ in each mini-batch, and re-sample 64 5-way tasks from it. It is notable that all instances in the 24-way task are encoded by the ResNet backbone with same parameters

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



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

Figure A6: Illustration of ClAssifier SynThesize LEarning (CASTLE).

in advance. Therefore, by embedding the synthesized 5-way few-shot classifiers into the global many-shot classifier, it results in 64 different configurations of the generalized few-shot classifiers. To evaluate which we randomly sample instances with batch size 128 from $\mathcal{S}$ and compute the GFSL objective in Eq. 7.

B DETAILS AND SETUPS FOR GFSL METHODS

In this section, we provide details about the training and evaluation setups for the generalized few-shot learning, followed by concrete descriptions for comparison methods.

B.1 COMPLETE DETAILS ON GFSL BASELINES

B.1.1 MULTICLASS CLASSIFIER (MC) + kNN.

Setup. We train a multi-class classifier on the populated SEEN classes following practices of training Residual Networks (He et al., 2016). Here a ResNet 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 details. 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 the union of multi-class classifiers' confidence and ProtoNet confidence as joint classification scores on $\mathcal{S} \cup \mathcal{U}$.

B.1.2 PROTONET + PROTONET.

Setup. We train a few-shot classifier (initialized by the MC classifier's feature mapping) using the Prototypical Network (Snell et al., 2017) (a.k.a ProtoNet). The backbone network is the same ResNet 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, the confidence of a test instances is jointly determined by its (negative) distance to both SEEN and UNSEEN class prototypes.

B.1.3 MC + PROTONET.

Setup. We combine the learning objective of the previous two baselines to jointly learn the MC classifier and feature embedding. Since there are two objectives for many-shot (cross-entropy loss on all SEEN classes) and few-shot (ProtoNet meta-learning objective) classification respectively, it trades off between many-shot and few-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.

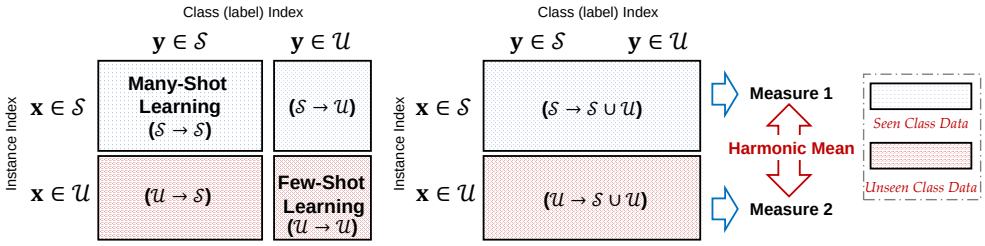


Figure A7: An illustration of the harmonic mean based GFSL evaluation. $\mathcal{S}$ and $\mathcal{U}$ denotes the SEEN and UNSEEN instances ($\mathbf{x}$) and labels ($\mathbf{y}$) respectively. $\mathcal{S} \cup \mathcal{U}$ is the joint set of $\mathcal{S}$ and $\mathcal{U}$. The notation $X \to Y, X, Y \in \{\mathcal{S}, \mathcal{U}, \mathcal{S} \cup \mathcal{U}\}$ means computing prediction results with instances from X to labels of Y. By computing a performance measure (like accuracy) on the joint label space prediction of SEEN and UNSEEN instances separately, a harmonic mean is computed to obtain the final measure.

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 $\mathcal{S} \cup \mathcal{U}$.

B.1.4 L2ML.

Setup. Wang et al. (2017) propose learning to model the "tail" (L2ML) by connecting a few-shot classifier with the corresponding many-shot classifier. The method is designed to learn classifier dynamics from data-poor "tail" classes to the data-rich "head" classes. Since L2ML is originally designed to learn with both SEEN and UNSEEN classes in a transductive manner, in our experiment, we adaptive it to out setting. Therefore, we learn a classifier mapping based on the sampled few-shot tasks from SEEN class set $\mathcal{S}$, which transforms a few-shot classifier in UNSEEN class set $\mathcal{U}$ inductively.

Training and inference. Following (Wang et al., 2017), we first train a many-shot classifier W upon the ResNet backbone on the SEEN class set $\mathcal{S}$. We use the same residual architecture as in (Wang et al., 2017) to implement the classifier mapping f, which transforms a few-shot classifier to a many-shot classifier. During the meta-learning stage, a $\mathcal{S}$-way few-shot task is sampled in each mini-batch, which produces a $\mathcal{S}$-way linear few-shot classifier $\hat{W}$ based on the fixed pre-trained embedding. The objective of L2ML not only regresses the mapped few-shot classifier $f(\hat{W})$ close to the many-shot one W measured by square loss, but also minimize the classification loss of $f(\hat{W})$ over a randomly sampled instances from $\mathcal{S}$. Therefore, this learned model uses a pre-trained multi-class classifier W for those "head" categories, and used the predicted few-shot classifiers with f for the "tail" categories.

B.1.5 DFSL.

Setup. Dynamic Few-Shot Learning without forgetting (DFSL) (Gidaris & Komodakis, 2018) also adopts a generalized few-shot learning objective. It decomposes the GFSL learning with two stages. A cosine classifier together with the backbone is learned at first. The pre-trained cosine classifier is regarded as bases. Based on the fixed backbone, another attention-based network constructs the classifier for a particular class by a linear combination of the elements in the bases.

Training and inference. We follow the strategy in (Gidaris & Komodakis, 2018) to train the DFSL model. Based on the pre-trained backbone and cosine classifier, we construct a dictionary with size $|\mathcal{S}|$ whose elements correspond to each category in $\mathcal{S}$. In each mini-batch of meta-training, we sample a few-shot task from the SEEN class set whose classes construct the set $\mathcal{C}$. Then, an attention model composes the classifier for the few-shot task by weighting the $|\mathcal{S}| - |\mathcal{C}|$ elements in the dictionary not corresponding to $\mathcal{C}$. To evaluate the composed classifier, DFSL samples an equal number of instances from $\mathcal{C}$ and $\mathcal{S} - \mathcal{C}$ for a test. For inference, we use the cosine classifier for "head" classes and composed few-shot classifier for "tail" classes.

B.2 DETAILS ON GENERALIZED FEW-SHOT LEARNING SETUPS

We take advantage of the auxiliary meta-train set from the benchmark data sets during GFSL evaluations, and an illustration of the data set construction can be found in Figure A5. The notation $X \to Y$ with $X, Y \in \{\mathcal{S}, \mathcal{U}, \mathcal{S} \cup \mathcal{U}\}$ means computing prediction results with instances from X to labels of Y. For example, $\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$ means we first filter instances come from the SEEN class set ($\mathbf{x} \in \mathcal{S}$), and predict them into the joint label space ($\mathbf{y} \in \mathcal{S} \cup \mathcal{U}$). For a GFSL model, we consider its performance with different measurements. An illustration of some criteria is shown in Figure A7.

Many-shot accuracy. A model is required to predict the auxiliary SEEN class instances towards all SEEN classes ($\mathcal{S} \to \mathcal{S}$). This is the same criterion with the standard supervised learning.

Few-shot accuracy. Following the standard protocol (Vinyals et al., 2016; Finn et al., 2017; Snell et al., 2017; Ye et al., 2018), we sample 10,000 K-shot N-way tasks from $\mathcal{U}$ during inference. In detail, we first sample N classes from $\mathcal{U}$, and then sample $K + 15$ instances for each class. The first NK labeled instances (K instances from each of the N classes) are used to build the few-shot classifier, and the remaining $15N$ (15 instances from each of the N classes) are used to evaluate the quality of such few-shot classifier. During our test, we consider $K = 1$ and $K = 5$ as in the literature, and change N ranges from $\{5, 10, 15, \ldots, |\mathcal{U}|\}$ as a more robust measure. It is noteworthy that in this test stage, all the instances come from $\mathcal{U}$ and are predicted to classes in $\mathcal{U}$ ($\mathcal{U} \to \mathcal{U}$).

Generalized few-shot accuracy. Different from many-shot and few-shot evaluations, the generalized few-shot learning takes the joint instance and label spaces into consideration. In other words, the instances come from $\mathcal{S} \cup \mathcal{U}$ and their predicted labels also in $\mathcal{S} \cup \mathcal{U}$ ($\mathcal{S} \cup \mathcal{U} \to \mathcal{S} \cup \mathcal{U}$). This is obviously more difficult than the previous many-shot ($\mathcal{S} \to \mathcal{S}$) and few-shot ($\mathcal{U} \to \mathcal{U}$) tasks. During the test, with a bit abuse of notations, we sample K-shot $\mathcal{S} + N$-way tasks from $\mathcal{S} \cup \mathcal{U}$. Concretely, we first sample a K-shot N-way task from $\mathcal{U}$, with NK training and $15N$ test instances respectively. Then, we *randomly* sample $15N$ instances from $\mathcal{S}$. Thus in a GFSL evaluation task, there are NK labeled instances from $\mathcal{U}$, and $30N$ test instances from $\mathcal{S} \cup \mathcal{U}$. We compute the accuracy of $\mathcal{S} \cup \mathcal{U}$ as the final measure.

Generalized few-shot Δ-value. Since the problem becomes difficult when the predicted label space expands from $\mathcal{S} \to \mathcal{S}$ to $\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$ (and also $\mathcal{U} \to \mathcal{U}$ to $\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$), the accuracy of a model will have a drop. To measure how the classification ability of a GFSL model changes when working in a GFSL scenario, Ren et al. (2018a) propose the Δ-Value to measure the average accuracy drop. In detail, for each sampled GFSL task, we first compute its many-shot accuracy ($\mathcal{S} \to \mathcal{S}$) and few-shot accuracy ($\mathcal{U} \to \mathcal{U}$). Then we calculate the corresponding accuracy of SEEN and UNSEEN instances in the joint label space, *i.e*, $\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$ and $\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$. The Δ-Value is the average decrease of accuracy in these two cases.

Generalized few-shot harmonic mean. Directly computing the accuracy still gets biased towards the populated classes, so we also consider the harmonic mean as a more balanced measure (Xian et al., 2017). By computing performance measurement such as top-1 accuracy and sample-wise Mean Average Precision (MAP) for $\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$ and $\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$, the harmonic mean is used to average the performance in these two cases as the final measure. An illustration is in Figure A7.

Generalized few-shot AUSUC. Chao et al. (2016) propose a calibration-agnostic criterion for generalized zero-shot learning. To avoid evaluating a model influenced by a calibration factor between SEEN and UNSEEN classes, they propose to determine the range of the calibration factor for all instances at first, and then plot the SEEN-UNSEEN accuracy curve based on different configurations of the calibration values. Finally, the area under the SEEN-UNSEEN curve is used as a more robust criterion. We follow (Chao et al., 2016) to compute the AUSUC value for sampled GFSL tasks.

C ADDITIONAL RESULTS FOR GFSL APPROACHES

In this section, we first do ablation studies on the proposed CASTLE approach, and then provide additional results for comparison methods in the GFSL evaluations.

C.1 DESIGN CHOICES OF CASTLE.

In this section, we aim to study the ablated variant of our approach and perform in-depth analyses.

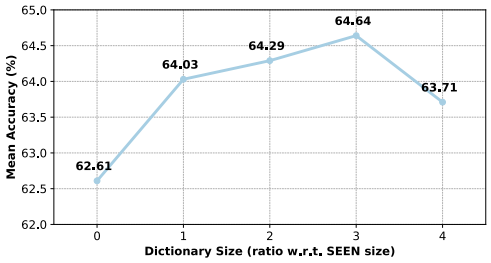


Figure A8: The 1-shot 5-way accuracy on UNSEEN of *Mini*ImageNet with different size of dictionaries.

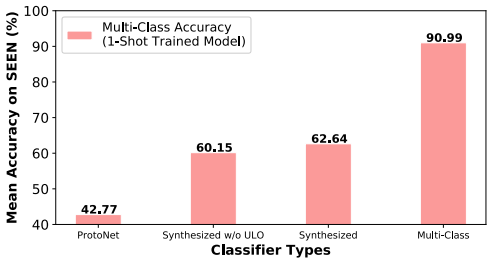


Figure A9: The 64-way multi-class accuracy on SEEN of *Mini*ImageNet with 1-shot trained model.

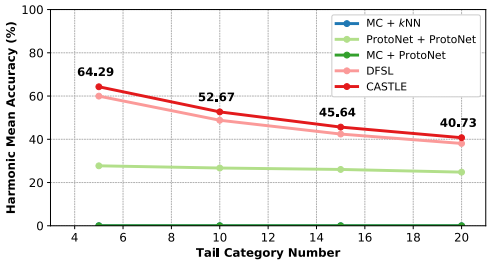


Figure A10: The 1-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

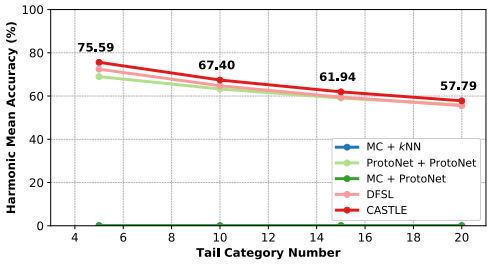


Figure A11: The 5-shot GFSL performance with incremental number of UNSEEN classes on *Mini*ImageNet.

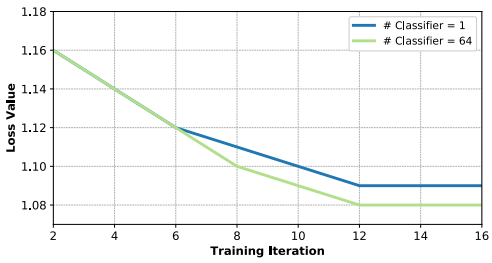


Figure A12: The 1-shot GFSL loss trained with different number of classifiers from pre-initialized backbone when there are 5 UNSEEN classes on *Mini*ImageNet.

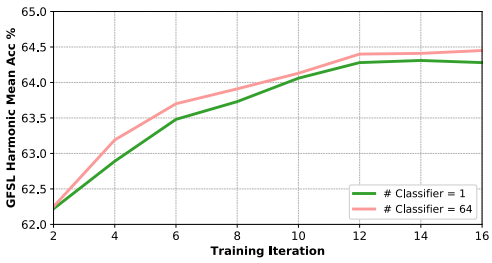


Figure A13: The 1-shot GFSL performance trained with different number of classifiers from pre-initialized backbone when there are 5 UNSEEN classes on *Mini*ImageNet.

Table A6: The performance with different choices of classifier synthesize strategies when tested with 5-Shot 5-Way UNSEN Tasks on *Mini*ImageNet. We denote the option compute embedding prototype and average synthesized classifiers as "Pre-AVG" and "Post-AVG" respectively.

Measurements →	FSL *Mean Accuracy*	GFSL *Harmonic Mean Accuracy*
CASTLE w/ Pre-AVG	79.34 ±0.01	75.59 ±0.10
CASTLE w/ Post-AVG	79.36 ±0.01	75.32 ±0.09

Table A7: The performance change with different number of classifiers when tested with 1-Shot 5-Way UNSEN Tasks on *Mini*ImageNet.

# of Classifiers →	1	64	128	256	512
CASTLE	62.81 ±0.14	64.02 ±0.14	64.29 ±0.14	64.88 ±0.14	63.99 ±0.14

Effects on the neural dictionary size $|\mathcal{B}|$. We show the effects of the dictionary size (as the ratio of SEEN class size) for the generalized few-shot learning (measured by harmonic mean accuracy when there are 5 UNSEEN classes) in Figure A8. We observe that the neural dictionary with a ratio of 2 or 3 works best amongst all other dictionary sizes. Therefore, we fix the dictionary size as 128 across all experiments. Note that when $|\mathcal{B}| = 0$, our method degenerates to case optimizing the unified objective in Eq. 7 without using the neural dictionary.

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 A9. 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 data set. 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.

Different choices of the classifier synthesis. As in Eq. 3, when there are more than one instance per class in a few-shot task (*i.e* $K > 1$), CASTLE compute the averaged embeddings first, and then use the prototype of each class as the input of the neural dictionary to synthesize their corresponding classifiers. Here we explore another choice to deal with multiple instances in each class. We synthesize classifiers based on each instance first, and then average the corresponding synthesized classifiers for each class. This option equals an ensemble strategy to average the prediction results of each instance's synthesized classifier. We denote the pre-average strategy (the one used in CASTLE) as "Pre-AVG", and the post-average strategy as "Post-AVG". The 5-Shot 5-way classification results on *Mini*ImageNet for these two strategies are shown in Table A6. From the results, "Post-AVG" does not improve the FSL and GFSL performance obviously. Since averaging the synthesized classifiers in a hindsight way costs more memory during meta-training, we choose the "Pre-AVG" option to synthesize classifiers when there are more than 1 shot in each class.

What is the performance when evaluated with more UNSEEN classes? As mentioned in the analysis 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, 5 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 A10 and Figure A11 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.

How is *multiple classifiers learning's* impact over the training? (cf. Section 3) CASTLE adopts a multi-classifier training strategy, *i.e* considering multiple GFSL tasks with different combinations of classifiers in a single mini-batch. Here we show the influence of the multi-classifier training method based on their FSL and GFSL performance. Figure A12 and Figure A13 show the change of loss and harmonic mean accuracy (with 5 UNSEEN tasks) when training CASTLE with different number of classifiers based on a pre-trained backbone, respectively. It is obvious that training with multiple classifiers converges faster and generalizes better than the vanilla model, without increasing the computational burden a lot. A more detailed comparison for training with different numbers of classifiers is listed in Table A7, which verifies the effectiveness of the multi-classifier training strategy.

C.2 EVALUATION ANALYSES OF GFSL METHODS.

In this subsection, we provide concrete values for the GFSL measurements on *Mini*ImageNet. To avoid repetition, only the results of 1-Shot GFSL tasks are listed. From Table A8 to Table A11, the number of ways of UNSEEN classes in an inference GFSL task varies from 5 to 20. In addition to the top-1 accuracy, the sample-wise mean average precision (MAP) is also calculated as a basic measure before harmonic mean. As shown in Figure A7, the harmonic mean is the harmonic average of the joint prediction performance of SEEN ($\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$) and UNSEEN ($\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$) instances. Although CASTLE cannot achieve high joint label space prediction on SEEN class instances ($\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$), its high harmonic mean performance results from its competitive discerning ability on the joint prediction of UNSEEN instances ($\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$).

Table A8: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 5-Way** UNSEEN class. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	90.95±0.07	1.54±0.03	2.98±0.06	93.75±0.05	21.14±0.06	34.37±0.08
DFSL′	66.19±0.11	55.81±0.19	59.96±0.13	74.96±0.13	73.12±0.13	73.72±0.10
MC+kNN	90.97±0.06	0.00±0.00	0.00±0.00	93.97±0.05	1.52±0.00	2.99±0.00
MC+ProtoNet	90.41±0.07	0.00±0.00	0.00±0.00	93.58±0.06	1.52±0.00	3.00±0.00
ProtoNet+ProtoNet	88.66±0.07	16.89±0.14	27.73±0.19	92.11±0.06	36.69±0.14	52.03±0.15
Ours: CASTLE	76.59±0.10	56.36±0.20	64.29±0.14	83.89±0.11	73.30±0.14	77.97±0.09

Table A9: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 10-Way** UNSEEN class. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \rightarrow \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	90.98±0.05	0.44±0.01	0.87±0.02	93.84±0.04	15.10±0.03	25.97±0.05
DFSL′	60.99±0.08	41.16±0.12	48.81±0.09	69.60±0.10	60.16±0.10	64.34±0.07
MC+kNN	91.00±0.05	0.00±0.00	0.00±0.00	93.89±0.04	1.49±0.00	2.94±0.00
MC+ProtoNet	90.42±0.05	0.00±0.00	0.00±0.00	93.57±0.04	1.51±0.00	2.97±0.00
ProtoNet+ProtoNet	88.14±0.05	15.94±0.09	26.72±0.13	91.70±0.04	34.38±0.10	49.81±0.10
Ours: CASTLE	69.99±0.08	42.68±0.13	52.67±0.10	78.93±0.09	61.75±0.10	69.12±0.07

C.3 EVALUATION ANALYSES OF GFSL METHODS WITH CALIBRATION.

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).

Table A10: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 15-Way** UNSEEN class. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	90.95±0.04	0.21±0.01	0.41±0.01	93.85±0.03	12.50±0.03	22.04±0.04
DFSL′	57.86±0.07	33.78±0.09	42.42±0.08	66.36±0.08	52.55±0.08	58.52±0.06
MC+kNN	91.02±0.04	0.00±0.00	0.00±0.00	93.95±0.03	1.47±0.00	2.90±0.00
MC+ProtoNet	90.42±0.04	0.00±0.00	0.00±0.00	93.60±0.03	1.49±0.00	2.94±0.00
ProtoNet+ProtoNet	87.94±0.04	15.41±0.07	26.06±0.10	91.39±0.03	32.58±0.07	47.91±0.08
Ours: CASTLE	65.27±0.07	35.38±0.10	45.65±0.08	75.31±0.08	54.49±0.08	63.10±0.06

Table A11: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 20-Way** UNSEEN class. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	90.99±0.03	0.13±0.00	0.27±0.01	93.85±0.03	10.96±0.02	19.61±0.03
DFSL′	55.48±0.06	29.18±0.07	38.07±0.06	64.03±0.07	47.33±0.07	54.32±0.05
MC+kNN	90.98±0.03	0.00±0.00	0.00±0.00	93.91±0.03	1.45±0.00	2.86±0.00
MC+ProtoNet	90.39±0.03	0.00±0.00	0.00±0.00	93.58±0.03	1.47±0.00	2.90±0.00
ProtoNet+ProtoNet	87.35±0.04	14.54±0.05	24.82±0.08	90.93±0.03	31.01±0.06	46.17±0.07
Ours: CASTLE	61.51±0.06	30.65±0.08	40.74±0.07	72.40±0.07	49.30±0.07	58.56±0.05

Table A12: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 5-Way** UNSEEN class, and the harmonic mean is computed **with a calibration factor**. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	71.62±0.11	53.18±0.19	60.42±0.14	84.19±0.09	69.36±0.13	75.82±0.09
DFSL′	72.61±0.10	54.26±0.19	61.53±0.13	84.23±0.11	69.57±0.13	75.93±0.09
MC+kNN	65.14±0.14	48.65±0.19	54.91±0.13	78.76±0.13	59.07±0.18	66.93±0.12
MC+ProtoNet	72.49±0.11	52.20±0.21	59.92±0.15	73.89±0.16	70.39±0.17	71.58±0.12
ProtoNet+ProtoNet	76.79±0.10	53.29±0.20	62.23±0.15	84.54±0.10	69.26±0.15	75.81±0.10
Ours: CASTLE	73.35±0.10	57.92±0.20	64.72±0.14	85.84±0.10	71.82±0.14	77.94±0.09

Table A13: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 10-Way** UNSEEN class, and the harmonic mean is computed **with a calibration factor**. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	69.06±0.08	38.57±0.12	49.16±0.10	83.28±0.07	55.95±0.10	66.79±0.07
DFSL′	72.62±0.07	39.31±0.12	50.68±0.11	81.14±0.09	57.85±0.10	67.38±0.07
MC+kNN	72.28±0.09	32.00±0.12	43.97±0.11	68.04±0.11	51.74±0.11	58.49±0.08
MC+ProtoNet	64.95±0.09	40.45±0.13	49.47±0.10	77.63±0.10	52.60±0.12	62.43±0.09
ProtoNet+ProtoNet	71.71±0.08	40.73±0.13	51.59±0.11	80.92±0.08	58.67±0.11	67.83±0.08
Ours: CASTLE	73.68±0.07	41.61±0.13	52.84±0.11	81.53±0.08	60.86±0.10	69.53±0.07

Table A14: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 15-Way** UNSEEN class, and the harmonic mean is computed **with a calibration factor**. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	67.31±0.06	31.12±0.09	42.34±0.08	82.41±0.05	48.23±0.07	60.74±0.06
DFSL′	65.97±0.06	33.08±0.09	43.85±0.08	79.16±0.07	50.84±0.08	61.80±0.06
MC+kNN	66.62±0.08	27.76±0.09	38.94±0.09	77.88±0.07	38.78±0.09	51.58±0.08
MC+ProtoNet	59.48±0.08	33.94±0.09	42.97±0.08	73.10±0.08	47.41±0.09	57.33±0.07
ProtoNet+ProtoNet	67.88±0.07	33.95±0.10	45.02±0.09	77.96±0.07	52.23±0.08	62.42±0.06
Ours: CASTLE	69.46±0.07	34.70±0.10	46.04±0.09	83.58±0.06	51.83±0.08	63.87±0.06

Table A15: Concrete evaluation criteria for generalized few-shot classification measurements on *Mini*ImageNet. The GFSL tasks are composed by **1-shot 20-Way** UNSEEN class, and the harmonic mean is computed **with a calibration factor**. "HM" denotes the harmonic mean.

Performance Measure →	Top1-Acc			Sample-wise MAP		
Criteria →	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM	$\mathcal{S} \to \mathcal{S} \cup \mathcal{U}$	$\mathcal{U} \to \mathcal{S} \cup \mathcal{U}$	HM
L2ML′	67.26±0.06	26.54±0.07	37.89±0.07	82.36±0.05	42.92±0.06	56.35±0.06
DFSL′	69.21±0.05	28.04±0.07	39.76±0.07	80.60±0.06	45.19±0.06	57.82±0.05
MC+kNN	62.54±0.07	24.63±0.07	35.17±0.07	75.39±0.07	35.63±0.07	48.25±0.07
MC+ProtoNet	54.81±0.07	29.45±0.08	38.13±0.07	70.34±0.07	43.17±0.07	53.36±0.06
ProtoNet+ProtoNet	65.23±0.06	29.47±0.08	40.42±0.07	75.44±0.06	47.48±0.07	58.18±0.05
Ours: CASTLE	66.07±0.06	30.16±0.08	41.24±0.07	81.73±0.06	47.20±0.07	59.75±0.05

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 (Chao et al., 2016; Wang et al., 2018). We exactly follow the strategy in (Chao et al., 2016) 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 confidence for SEEN classes, to make it balanced with the predictions for 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. 30 equally split calibration bias values are used as candidates, and the best one is chosen based on 500 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. The results are listed from Table A12 to Table A15, and the number of UNSEEN classes in a GFSL task changes from 5 to 20. Comparing with the results without calibration factor in Table A8-A11, the additional calibration step increases the joint prediction ability of UNSEEN instances a lot, so as to improve the final harmonic mean measurement. Our CASTLE get similar results after using the calibration bias, especially when there are 5 UNSEEN classes. Therefore, CASTLE fits the generalized few-shot learning task, and does not require additional calibration step to balance the SEEN and UNSEEN predictions.

C.4 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 Figure A14. 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.

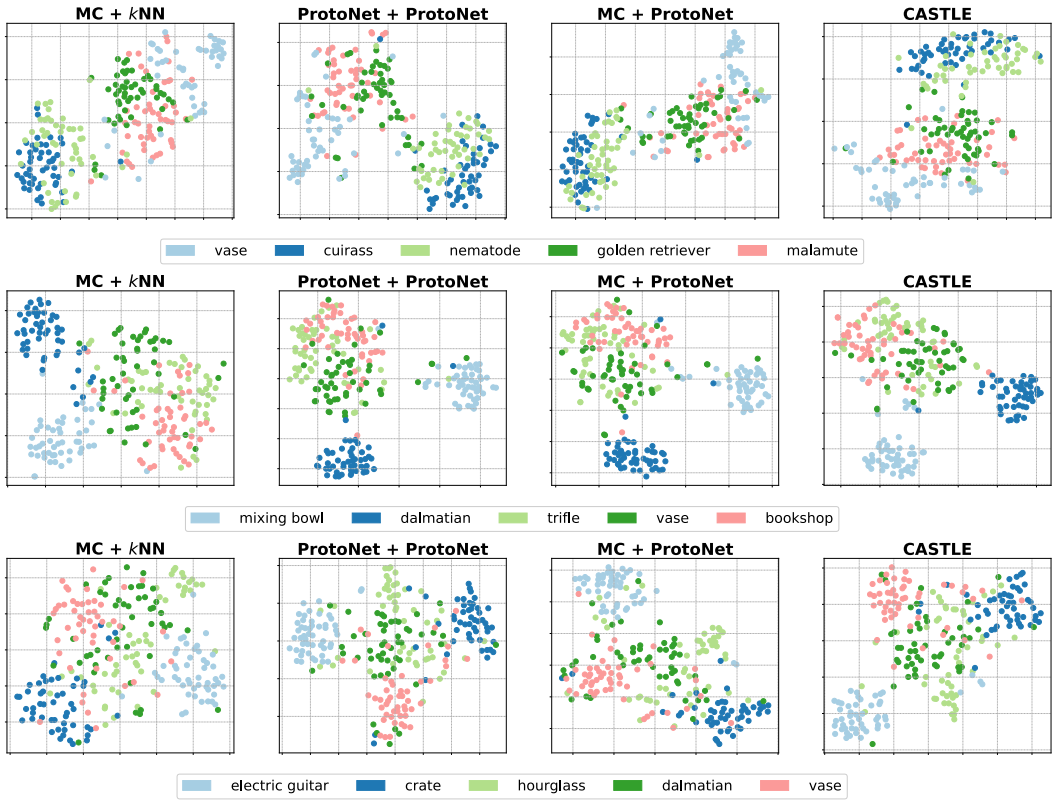


Figure A14: Three groups of embedding visualization results of 6 randomly selected UNSEEN classes. Four baselines are compared. Different colors denote the classes. Best viewed in color.

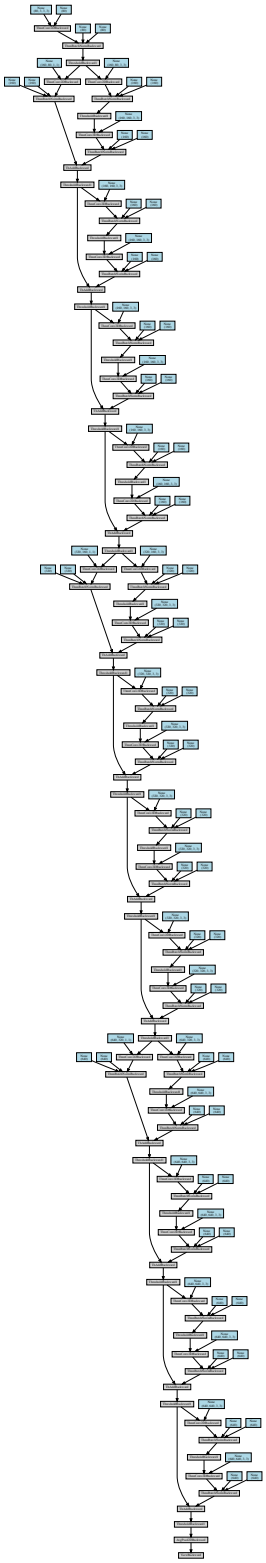


Figure A15: The detailed architecture of ResNet backbone we used. Better perceived when zoomed in.