# Task-Adaptive Negative Envision for Few-Shot Open-Set Recognition

Shiyuan Huang* Jiawei Ma* Guangxing Han Shih-Fu Chang
Columbia University
{shiyuan.h, jiawei.m, gh2561, sc250}@columbia.edu

## Abstract

*We study the problem of few-shot open-set recognition (FSOR), which learns a recognition system capable of both fast adaptation to new classes with limited labeled examples and rejection of unknown negative samples. Traditional large-scale open-set methods have been shown ineffective for FSOR problem due to data limitation. Current FSOR methods typically calibrate few-shot closed-set classifiers to be sensitive to negative samples so that they can be rejected via thresholding. However, threshold tuning is a challenging process as different FSOR tasks may require different rejection powers. In this paper, we instead propose task-adaptive negative class envision for FSOR to integrate threshold tuning into the learning process. Specifically, we augment the few-shot closed-set classifier with additional negative prototypes generated from few-shot examples. By incorporating few-shot class correlations in the negative generation process, we are able to learn dynamic rejection boundaries for FSOR tasks. Besides, we extend our method to generalized few-shot open-set recognition (GFSOR), which requires classification on both many-shot and few-shot classes as well as rejection of negative samples. Extensive experiments on public benchmarks validate our methods on both problems.* [1]

## 1. Introduction

With the emergence of large-scale image datasets [4, 5, 23], deep learning has achieved great success in various vision tasks [3, 15–17, 31, 34]. Current recognition systems usually assume a predefined set of classes with sufficient number of labeled data. Each testing sample is supposed to belong to these predefined classes so that the systems only need to perform closed-set classification.

In real-world applications, we face more challenging recognition scenarios. First, sufficient labeled training data are hardly guaranteed due to high cost of data collection and possibly limited access to sensitive or rare data. Few-shot learning [6, 37, 40, 43] (FSL) typically tackles data insufficiency scenario by fast adaptation of recognition system to new classes with access to very few (e.g., only one) labeled instances. But FSL still holds a closed-set assumption.

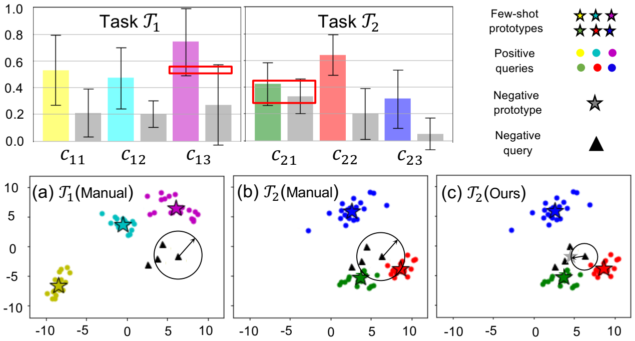

Figure 1. (Up): For each few-shot class in few-shot open-set recognition (FSOR) task $\mathcal{T}_1$ and $\mathcal{T}_2$, we calculate the detection scores (similarity) of both negative and positive queries and find their mean and standard deviation. (We use the same negative queries for both tasks.) However, a negative query may have similar detection score to the positive queries (highlighted by red box). (Down): In addition, to reject a negative query, existing FSOR method relies on a manually selected threshold. However, a rejection threshold working properly for $\mathcal{T}_1$ may fail in $\mathcal{T}_2$. Instead, we propose to learn a negative prototype that automatically estimates a task-adaptive threshold for negative detection.

On the other hand, there are efforts that endow a recognition system with the ability to handle out-of-distribution testing samples. Open-set recognition (OR) [1, 11, 25, 35, 36] considers the case where testing samples could come from other unknown source under a large-scale training setting. Without loss of capability of classifying closed-set queries (i.e., positive queries), it also needs to detect queries from unknown classes (i.e., negative queries). Current OR methods typically learns an open-set classifier by either calibrating prediction scores or synthesizing negative queries. They rely on large amount of data to avoid overfitting and estimate distribution properly. But with only a few labeled instances, it becomes hard to do so. Hence direct application

[1] Code available at https://github.com/shiyuanh/TANE

of OR methods under few-shot setting degrades the performance significantly [20, 24].

We aim to develop a model solving for both challenges, i.e., few-shot open-set recognition (FSOR). The goal of FSOR is to both 1) accept & recognize *positive queries* from few-shot classes with very few labeled samples and 2) detect *negative queries* from undisclosed (*negative*) classes. Previous FSOR methods [20, 24] provide meta-learning-based solutions for learning threshold-based negative detector. They calibrate few-shot close-set classifier and output a rejection score for each testing sample. A sample is rejected if the rejection score is above a certain rejection threshold, which has to be manually defined. However, as shown in Fig. 1, a good recognition performance relies heavily on a good choice of threshold: (a) a few-shot classifier may have similar detection score for a negative query and a positive query, where different thresholds need to be set separately; (b) to reject a negative query, a threshold works properly for one task may fail in other tasks. In summary, threshold tuning could be a challenging process as different FSOR tasks contain different few-shot classes that may need very different rejection powers to determine outliers.

In this paper, we instead propose to integrate threshold tuning into the learning process for FSOR. We extend the few-shot classifier with additional prototypes that represent the negative class. Specifically, a negative generator is applied on few-shot class prototypes and learns negative prototypes across tasks via meta-learning, so that negative prototypes can serve as task-adaptive rejection boundaries for different FSOR tasks. A testing query is then rejected if the prediction scores on all few-shot classes are lower than that on the negative prototype. We study the design of negative generator and experimentally demonstrate an optimal solution that involves task-level information into the negative prototype envision. We also introduce the concept of conjugate task of FSOR where two FSOR tasks are considered conjugate if the few-shot class in one task can be used to simulate unknown sources in the other. To this end, we propose a *conjugate training* strategy to facilitate the learning process. Moreover, we consider a new but more challenging problem, generalized FSOR (GFSOR), where the recognition system needs to classify on both many-shot and few-shot classes as well as reject negative samples. In this case, negative prototypes are generated from both many-shot and few-shot classes. We name our method of learning negative prototypes as *task-adaptive negative class envision*.

Our method is validated by extensive experiments on public benchmarks for both FSOR and GFSOR problems. In summary, our contributions are as follows:

1. We provide a threshold-free solution for few-shot open-set recognition (FSOR), where we extend classifier with negative prototypes that compute task-adaptive rejection boundaries.
2. We provide a study of the negative prototype generator design and experimentally demonstrate an optimal solution involving task-level knowledge for negative envision. In addition, we propose an efficient and novel training strategy, *conjugate training*, to facilitate the learning process.
3. Extensively evaluated on public benchmarks, our approach is able to achieve SOTA performance. We further formulate the problem of generalized FSOR where our method is also shown to be effective.

In the following sections, we will discuss related literature in FSL, OR and FSOR (Sec. 2); In Sec. 3 we formally define FSOR and GFSOR tasks and go over existing threshold-based meta-learning solutions. In Sec. 4 we present our approach of task-adaptive negative envision. Finally in Sec. 5 we demonstrate the experimental analysis and results of our approach.

## 2. Related Works

**Few-Shot Learning.** FSL aims for fast adaptation to new recognition task with very few labeled examples. Meta-learning is widely used to learn transferable knowledge upon a set of tasks using episodic training. There are mainly two types of meta-learning approaches: 1). Optimization-based method [6, 7, 28, 38] modifies the gradient back-propagation so that the parameter updates can be more sensitive to the few training examples; 2). Metric-based methods [12–14, 29, 37, 40, 43, 48] learns to obtain an optimal metric space so that a class with the highest similarity is assigned to the query. As an extension on FSL, generalized FSL (GFSL) [10] learns to expand many-shot classifier with novel classes using a few training data. Both (G)FSL hold a closed-set assumption where testing queries belong to novel classes (or many-shot classes in GFSL). Our work instead extends (G)FSL to open-set setting.

**Large-Scale Open-Set Recognition.** OR aims to learn a classifier sensitive to negative queries that come from unknown classes. OR methods typically include class probability re-calibration [1, 21, 36] and negative sample synthesis with generative methods [8, 27]. Those methods typically assume large number of training data. A most relevant work to us [49] also proposes to augment classifier to learn adaptive rejection thresholds. But it relies on large-scale data to train the augmented classifier from scratch, while ours generates negative prototypes based on few-shot classes. Direct application of OR methods to few-shot setting fails or degrades the performance [20, 24] mainly due to over-fitting. Our work instead provides a few-shot-specific OR solution to deal with limited data.

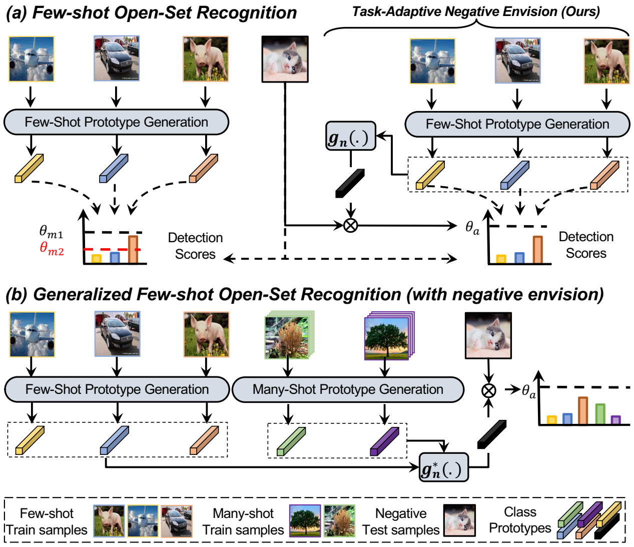
(a) Few-shot Open-Set Recognition
Task-Adaptive Negative Envision (Ours)
Few-Shot Prototype Generation
Few-Shot Prototype Generation
$g_n(.)$
$\theta_{m1}$
$\theta_{m2}$
Detection Scores
$\theta_a$
Detection Scores
(b) Generalized Few-shot Open-Set Recognition (with negative envision)
Few-Shot Prototype Generation
Many-Shot Prototype Generation
$\theta_a$
$g_n^*(.)$
Few-shot Train samples
Many-shot Train samples
Negative Test samples
Class Prototypes

Figure 2. (a) Comparison between conventional threshold-based FSOR methods and ours with negative envision. Conventional methods reject a sample if the detection scores of all classes are below a carefully selected threshold ($\theta_{m1}$). An improperly selected threshold ($\theta_{m2}$), on the other hand, would result in recognition failure. Instead, we propose to envision a negative prototype to *learn to* estimate threshold ($\theta_a$) for each instance dynamically within the task. (b) For a GFSOR task, the negative prototype is generated from both many-shot and few-shot prototypes.

**Few-Shot Open-Set Recognition.** To bridge FSL and OR, recently [24] provides a meta-learning-based solution for FSOR that introduces an open-set loss in the meta-training process to calibrate few-shot prototype-based classifier. [20] improves the limitation of negative sampling in [24] by imposing a transformation consistency regularization on few-shot samples. However, their methods are threshold-based, which require careful selection of thresholds to perform good recognition. Instead, we propose a threshold-free solution to overcome the challenge.

## 3. Problem Formulation

With only a few labeled training samples, few-shot open-set recognition (FSOR) aims to 1) detect negative queries that come from *unknown* sources and 2) correctly classify positive queries. Formally, a FSOR task can be denoted as $\mathcal{T} = (\mathcal{S}, \mathcal{Q}^f, \mathcal{Q}^n|\mathcal{C}^f)$ where $\mathcal{C}^f$ refer to the few-shot classes that have few labeled training samples (also called supports): $\mathcal{S} = \cup_{c\in\mathcal{C}^f}\mathcal{S}_c$. The goal is to learn a recognition model with the supports so that during testing time, it can successfully classify positive queries $\mathcal{Q}^f$ and detect negative queries $\mathcal{Q}^n$. We denote $\mathcal{Q} = \mathcal{Q}^f \cup \mathcal{Q}^n$ as the entire query set. We call a FSOR task $N$*-way* $K$*-shot* if we have $|\mathcal{C}^f| = N$ and $|\mathcal{S}_c| = K$ for all $c \in \mathcal{C}^f$. Briefly speaking, the only difference between FSOR and conventional FSL tasks is that FSOR has additional negative queries that need to be rejected.

Existing FSOR approaches [20, 24] are built upon the popular metric-based FSL method ProtoNet [37], and our approach also follows the same fashion. Below we provide more context on ProtoNet.

ProtoNet [37] learns a prototype-based few-shot classifier. In detail, each few-shot class $c \in \mathcal{C}^f$ is represented by a prototype $\mathbf{p}_c$, computed by the average of $K$ support features: $\mathbf{p}_c = \frac{1}{K}\sum_{s\in\mathcal{S}_c} f(s)$, where $f$ is a feature extractor and $f(s) \in \mathcal{R}^d$. Then, all prototypes $\mathbf{P}^f = \{\mathbf{p}_c\}_{c\in\mathcal{C}^f}$ build up a closed-set classifier where a positive query $q \in \mathcal{Q}^f$ can be classified by nearest neighborhood search, *i.e.*,

$$\text{argmax}_c\big(\{f_s(f(q), \mathbf{p}_c)\}_{c\in\mathcal{C}^f}\big), \tag{1}$$

where $f_s(\cdot,\cdot)$ is a function to measure the closeness between two inputs, *e.g.*, cosine similarity.

In order to learn an open-set classifier, existing FSOR approaches [20, 24] calibrate the few-shot close-set classifier to get per-class detection scores and reject via thresholding. As illustrated in Fig. 2(a), for threshold-based FSOR methods, a threshold $\theta_m$ needs to be manually set and a negative query $q \in \mathcal{Q}^n$ will be rejected if all of the detection scores are below $\theta_m$, *i.e.*, $\max\big(\{f_s(f(q), \mathbf{p}_c)\}_{c\in\mathcal{C}^f}\big) < \theta_m$.

In addition to FSOR tasks, we further consider a more realistic situation where both few-shot classes $\mathcal{C}^f$ and many-shot classes $\mathcal{C}^*$ (i.e., containing large amount of labeled data) exist, resulting an imbalanced distribution. To this end, we formulate the generalized few-shot open-set recognition (GFSOR) task $\mathcal{T}^* = (\mathcal{S}, \mathcal{Q}^*, \mathcal{Q}^f, \mathcal{Q}^n|\mathcal{C}^*, \mathcal{C}^f)$ where $\mathcal{Q}^*$ are queries from $\mathcal{C}^*$. And the goal is to correctly classify both $\mathcal{Q}^* \cup \mathcal{Q}^f$ and reject negative query $\mathcal{Q}^n$. Similarly, we call a GFSOR task $N$*-way* $K$*-shot* if we have $|\mathcal{C}^f| = N$ and $|\mathcal{S}_c| = K$ for all $c \in \mathcal{C}^f$.

## 4. Approach

Here we present our threshold-free approach towards (G)FSOR. We first provide an overview of how to use negative envision to estimate task-adaptive rejection boundaries; then we provide a list of negative generators used in practice; finally we introduce conjugate training which encourages the learning process from task mutual supervision.

### 4.1. Overview

Fig. 2 provides an overview of our *Task-Adaptive Negative Envision* approach and how it compares to threshold-based methods. Threshold-based methods [20,24] calculate per-class detection scores and manually define a threshold for rejection. Without a carefully cherry-picked a threshold for each task, it's hard to successfully detect $q \in \mathcal{Q}^n$

across different tasks (Fig. 2(a)). Instead, we expand classifier with negative prototype $\mathbf{p}^-$ that are computed from few-shot class prototypes $\mathbf{P}^f$ via a negative generator $g_n(\cdot)$. When a query comes in, it's able to automatically calculate a task-specific threshold from the negative prototype:

$$\theta_a = f_s(f(q), \mathbf{p}^-). \tag{2}$$

Then, a negative query $q \in \mathcal{Q}^n$ will be rejected if $\max\big(\{f_s(f(q), \mathbf{p}_c)\}_{c \in \mathcal{C}^f}\big) < \theta_a$. As such, rejection boundaries are dynamically estimated with respect to few-shot classes $\mathcal{C}^f$ and support instances $\mathcal{S}$. Our approach can be also applied to GFSOR tasks where the negative prototype are generated from both few-shot and many-shot class prototypes to get task-adaptive threshold with respect to both few-shot and many-shot classes.

## 4.2. Negative Generator

To find the best negative generator $g_n(\cdot)$, we explore different choices which we describe below in detail.

### 4.2.1 MLP

We start with a simple generator that consists of a single MLP layer applied on averaged class prototypes, i.e.,

$$\mathbf{p}^- = f_n(\mathbf{p}^-_{avg}), \;\; \mathbf{p}^-_{avg} = \frac{1}{N}\sum\nolimits_{c \in \mathcal{C}^f} \mathbf{p}_c \tag{3}$$

where $f_n$ is a MLP that takes $\mathbf{p}^-_{avg}$, the mean of $\mathbf{P}^f$, as input so that $\mathbf{p}^-$ is independent from the prototype order. Meanwhile, we set $\mathbf{p}^-_{avg}$ as a naive baseline (**AVG**) as $\mathbf{p}^-_{avg}$ is also order-independent.

### 4.2.2 ATT

Transformers [42] are proved effective in exploiting relations, which is also independent from the input order (without positional encoding). We apply a standard Transformer attention block over few-shot class prototypes to generate the negative prototype. Specifically, we calculate the self-attention weight between class prototypes, i.e.,

$$\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^f)} = \frac{1}{\sqrt{d}}(\mathbf{P}^f\mathbf{K}_q(\mathbf{P}^f\mathbf{K}_k)^T),$$

where $\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^f)} \in \mathcal{R}^{|\mathcal{C}^f| \times |\mathcal{C}^f|}$ is the attention weights matrix, and $\mathbf{K}_q, \mathbf{K}_k \in \mathcal{R}^{d \times d}$ are trainable linear projection kernels. Then, we normalize the weights and output

$$\mathbf{P}^{'} = \mathbf{P}^f + \sigma(\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^f)})(\mathbf{P}^f\mathbf{K}_v),$$

where $\sigma(\cdot)$ is a softmax function for each row in $\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^f)}$ and $\mathbf{K}_v \in \mathcal{R}^{d \times d}$ is another trainable linear projection kernel. Then, we feed the average of $\mathbf{P}^{'}$ to a MLP function $f_n$ to get the negative prototype $\mathbf{p}^-$.

### 4.2.3 ATT-G

The above generators is suitable for FSOR problem. Now we consider the more challenging GFSOR task. Directly employing the above methods may introduce bias towards $\mathcal{C}^*$ as $\mathcal{C}^*$ has plenty of training samples and the prototype $\mathbf{P}^*$ can be better-estimated compared against the few-shot prototypes $\mathbf{P}^f$. Hence we need another negative generator compatible with GFSOR , which should take care of both $\mathcal{C}^*$ and $\mathcal{C}^f$. We build our ATT-G generator on top of a popular GFSL method [10], which uses an attention mechanism to calibrate few-shot prototypes $\mathbf{P}^f$ with $\mathbf{P}^*$. Specifically, we follow [9, 10, 41] to first train a network under large-scale classification task using the labeled samples of $\mathcal{C}^*$ (*i.e.*, pre-training) and use the weight in the last linear layer as many-shot class prototypes $\mathbf{P}^*$. Then we apply the attention block between $\mathbf{P}^f$ and $\mathbf{P}^*$ to generate the negative prototype $\mathbf{p}^-$, i.e.,

$$\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^*)} = \frac{1}{\sqrt{d}}(\mathbf{P}^f\mathbf{K}_q(\mathbf{P}^*\mathbf{K}_k)^T), \tag{4}$$

$$\mathbf{P}^{'} = \mathbf{P}^f + \sigma(\mathbf{A}_{(\mathbf{P}^f, \mathbf{P}^*)})(\mathbf{P}^*\mathbf{K}_v), \tag{5}$$

and $\mathbf{p}^-$ is similarly computed by feeding the average of $\mathbf{P}^{'}$ into a MLP $f_n$. Furthermore, we'd like to filter out task-irrelevant information by applying a channel-wise gating mechanism on top of $\mathbf{P}^f$:

$$\mathbf{p}_c^{'} = \mathbf{p}_c \odot \phi(f_g(\frac{1}{N-1}\sum\nolimits_{i \in \mathcal{C}^f \setminus \{c\}} \mathbf{p}_i)), \tag{6}$$

for $c \in \mathcal{C}^f$ where $\odot$ and $\phi$ denote element-wise multiplication and sigmoid operation, and $f_g$ is a fully-connected layer. Then, we use the updated $\mathbf{P}^{f'}$ to replace the input $\mathbf{P}^f$ in Eq. 4. Finally, we follow the order of many-shot prototypes $\mathbf{P}^*$, few-shot prototypes $\mathbf{P}^f$, and negative prototype $\mathbf{p}^-$ to build the open-set classifier for a GFSOR task, where $\mathbf{P}^*$ are the weights in the last linear layer after pre-training.

### 4.2.4 SEMAN-G

Inspired by recent cross-modal FSL works [22, 45], we further explore how class semantics could help model negative class. Specifically, we use a cross-modal attention mechanism on top of **ATT-G**. For each class $c \in \mathcal{C}^f \cup \mathcal{C}^*$, we concatenate $\mathbf{p}_c$ with its word embedding $\mathbf{e}_c \in \mathcal{R}^w$ along the channel to have $\mathbf{z}_c = [\mathbf{p}_c, \mathbf{e}_c] \in \mathcal{R}^{w+d}$. Then we use $\mathbf{Z}^f$ and $\mathbf{Z}^*$ instead of $\mathbf{P}^f, \mathbf{P}^*$ in Eq. 4 to calculate the attention. And stick to Eq. 5 to take $\mathbf{P}^*$ as input since we are still comparing visual features for recognition.

### 4.2.5 Multiple Negative Prototypes

In addition, we can easily extend from single negative generation to multiple negative generation. Specifically, we can learn a set of generators $\{g_{n,i}\}_{i=1}^{M}$ to generate multiple negative prototypes for each task. For ATT, ATT-G, and SEMAN-G, to reduce the number of trainable parameters, we choose to share the linear projection kernels in attention mechanism used to calculate $\mathbf{P}^{'}$, but just train separate MLPs $\{f_{n,i}\}_{i=1}^{M}$ to synthesis multiple negative prototypes. In this way, we get multiple thresholds $\{\theta_{a,i}\}_{i=1}^{M}$. Then the maximum threshold $\theta_a = \max\big(\{\theta_{a,i}\}_{i=1}^{M}\big)$ is used as the final threshold for open-set recognition.

## 4.3. Conjugate Training

Here we present our conjugate training strategy towards (G)FSOR. Conjugate training is built upon the standard FSOR meta-training approach [24, 37]. We first go over the standard FSOR meta-training then introduce our method.

### 4.3.1 Standard FSOR Meta-Training

Standard FSOR meta-training strategy [24, 37] trains the model by simulating FSOR tasks from the given *base* dataset $\mathcal{D}^B$. Specifically, it trains on a set of tasks sampled from the *base* dataset $\mathcal{D}^B$ where the images are from *base* classes $\mathcal{C}^B$. Within an $N$-way $K$-shot FSOR task $\mathcal{T}$, unknown sources are simulated using a different set of $N$ classes $\mathcal{C}^n$, *i.e.*, $\mathcal{C}^n \subset \mathcal{C}^B - \mathcal{C}^f$ where $|\mathcal{C}^n| = N$, and then randomly sample images belonging to $\mathcal{C}^n$ from $\mathcal{D}^B$ for $\mathcal{Q}^n$. Then the model is trained using an objective, typically an open recognition loss within the sampled $\mathcal{T}$. The standard FSOR meta-training can be generalized to GFSOR. For a GFSOR task $\mathcal{T}^*$, we can sample $\mathcal{C}^f$ and $\mathcal{C}^*$ from $\mathcal{C}^B$ and then simulate unknown sources as $\mathcal{C}^n \subset \mathcal{C}^B - (\mathcal{C}^f \cup \mathcal{C}^*)$ where $|\mathcal{C}^n| = N$. And a GFSOR training objective may be specified to learn the model for GFSOR. Note that, during inference time (i.e., meta-testing), tasks are sampled from *novel* dataset $\mathcal{D}^N$ where the images are from classes $\mathcal{C}^N$ and no sample from $\mathcal{C}^N$ is seen during meta-training, *i.e.*, $\mathcal{C}^B \cup \mathcal{C}^N = \varnothing$.

### 4.3.2 Conjugate Tasks

The idea of conjugate training is to sample task pairs whose few-shot examples of one task are used as the negative source of the other. Formally, we define two tasks $\mathcal{T}_1 = (\mathcal{S}_1, \mathcal{Q}_1^f, \mathcal{Q}_1^n | \mathcal{C}_1^f)$ and $\mathcal{T}_2 = (\mathcal{S}_2, \mathcal{Q}_2^f, \mathcal{Q}_2^n | \mathcal{C}_2^f)$ as a *conjugate task pair* when $\mathcal{Q}_1^n = \mathcal{Q}_2^f$ and $\mathcal{Q}_1^f = \mathcal{Q}_2^n$, *i.e.*, the few-shot classes $\mathcal{C}_1^f$ ($\mathcal{C}_2^f$) in $\mathcal{T}_1$($\mathcal{T}_2$) is used as the negative source in task $\mathcal{T}_2$($\mathcal{T}_1$). For a conjugate GFSOR task pair ($\mathcal{T}_1^*, \mathcal{T}_2^*$), in addition, $\mathcal{T}_1^*$ and $\mathcal{T}_2^*$ share the same many-shot class $\mathcal{C}^*$ and its queries $\mathcal{Q}^*$.

### 4.3.3 Conjugate Training Loss

We use a standard cross-entropy loss $\mathcal{L}_{CE}(\cdot, \cdot)$ [26]. For an FSOR task $\mathcal{T}$, we use cosine similarity as $f_s$ and use $\mathcal{Q}^f \cup \mathcal{Q}^n$ to perform $(N+1)$-way classification. For each positive query $q \in \mathcal{Q}^f$, we learn to maximize the class score of its label category by minimizing $\mathcal{L}_{CE}(y_q, q)$ where $y_q \in \{1, ..., N\}$ is the class label of $q$. For each negative query $q \in \mathcal{Q}^n$, we set its ground truth label as $N+1$ and maximize the threshold $\theta_a$ by minimizing $\mathcal{L}_{CE}(N+1, q)$. During conjugate training, we consider the dependency of negative sampling mentioned in [20]. Without loss of generality, for a positive query $q \in \mathcal{Q}_2^f$ belonging to class $c_n \in \mathcal{C}_2^f$ in $\mathcal{T}_2$, it is used as the negative query and is trained to have high similarity with the negative prototype in $\mathcal{T}_1$.

With a simple classification loss, the negative prototypes are optimized to learn a tight rejection boundary for a specific task. Besides, for the attention-based generators, we also regularize the intermediate variables $\mathbf{P}^{'}$ as class-specific negative prototypes. For each prototype $\mathbf{p}_c^{\prime} \in \mathbf{P}^{'}$ generated from a positive prototype $\mathbf{p}_c$, we can think of $\mathbf{p}_c^{\prime}$ as the negative prototype for class $c$. Then, for each $\mathbf{p}_c^{\prime}$, we minimize its similarity with queries of class $c$ and maximize its similarity of negative queries with a binary cross-entropy loss $\mathcal{L}_{BCE}$

$$\mathcal{L}_{neg}(c) = \frac{1}{|\mathcal{Q}_{c,1}^f|} \sum\nolimits_{q \in \mathcal{Q}_{c,1}^f} \mathcal{L}_{BCE}(0, f_s(f(q), \mathbf{p}_c^{\prime})) + \frac{1}{|\mathcal{Q}_1^n|} \sum\nolimits_{q \in \mathcal{Q}_1^n} \mathcal{L}_{BCE}(1, f_s(f(q), \mathbf{p}_c^{\prime}))$$

where $\mathcal{Q}_{c,1}^f = \{q | q \in \mathcal{Q}_1^f, y_q = c\}$ and $y_q$ denotes the class label of $q$. Finally, without loss of generality, for $\mathcal{T}_1$ in the conjugate task pair $(\mathcal{T}_1, \mathcal{T}_2)$, we have

$$\mathcal{L}_{\mathcal{T}_1} = \mathcal{L}_{CE}(\mathcal{Q}_1^f \cup \mathcal{Q}_1^n) + \frac{1}{|\mathcal{C}_1^f|} \sum\nolimits_{c \in \mathcal{C}_1^f} \mathcal{L}_{neg}(c), \qquad (7)$$

and the total conjugate training loss is $\mathcal{L} = \mathcal{L}_{\mathcal{T}_1} + \mathcal{L}_{\mathcal{T}_2}$.

Similarly, for the network trained on GFSOR tasks, given a conjugate task pair $(\mathcal{T}_1^*, \mathcal{T}_2^*)$, we have

$$\mathcal{L}_{\mathcal{T}_1^*}^* = \mathcal{L}_{CE}(\mathcal{Q}_1^* \cup \mathcal{Q}_1^f \cup \mathcal{Q}_1^n) + \frac{1}{|\mathcal{C}_1^f|} \sum\nolimits_{c \in \mathcal{C}_1^f} \mathcal{L}_{neg}(c),$$

where the class label for a negative query is $N + 1 + |\mathcal{C}^*|$, and the total loss being $\mathcal{L}^* = \mathcal{L}_{\mathcal{T}_1^*}^* + \mathcal{L}_{\mathcal{T}_2^*}^*$. In this way, our *conjugate training* involves the class-correlation during network training.

## 5. Experiments and Analysis

### 5.0.1 Datasets

For FSOR tasks, we evaluate on two widely used public benchmarks: MiniImageNet [43], TieredImageNet [33]. MiniImageNet [43] contains 100 classes and the class split for (meta-training, meta validation,meta-testing) is (64,16,20). Each class has 600 images. TieredImageNet

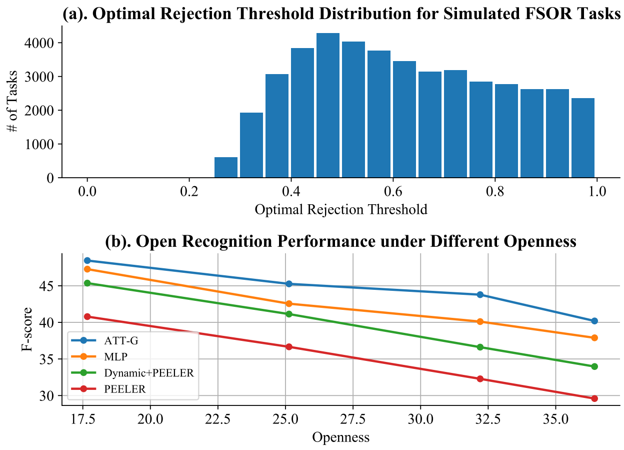

Figure 3. (a) Distribution of optimal rejection thresholds with Dynamic+PEELER on Mini-ImageNet. (b) Comparison of open recognition performance under different openness.

Table 1. F1-score comparison on 5-way 1-shot FSOR tasks on Mini-ImageNet. $^*$: our implementation.

| Neg. Gen. | 0.3 | 0.5 | 0.7 | 0.9 |
|---|---|---|---|---|
| PEELER | 34.01 | 40.71±0.59 | 41.44 | 35.30 |
| Dynamic+PEELER | 38.19 | 45.34±0.64 | 44.10 | 30.95 |

| Neg. Gen. | Single Neg. | Multi Neg. |
|---|---|---|
| AVG | 45.6±0.71 | - |
| MLP | 46.12±0.74 | 47.21±0.72 |
| ATT | 46.38±0.73 | 47.29±0.70 |
| ATT-G | 47.03±0.74 | 48.19±0.71 |
| SEMAN-G | 47.95±0.72 | 50.10±0.69 |

[33] contains 608 classes and the class split is (351, 97, 160) while the *base* dataset contains around 450K images. We evaluate GFSOR on MiniImageNet [43] and set the base classes during meta-training as the many-shot classes during meta-testing. We follow [10] and use another 300 images for each *base* class for the GFSOR simulation. All images for the two datasets are sized to 84×84. For **SEMAN-G**, we extract word embedding using GloVe [30]. More details of the datasets can be found in the supp. material.

### 5.0.2 Implementation Details

We use ResNet12 [19] network as the feature backbone. Following [9,41], we pre-train the ResNet12 and a classifier (a linear layer) with cross-entropy loss and a self-supervised rotation loss on the base set under fully-supervised classification task for 90 epochs using a SGD optimizer with learning rate $0.05$ decayed by 10 at epoch 60. The weights in the linear layer are used as base-class many-shot prototypes $\mathbf{P}^*$ for **ATT-G** and **SEMAN-G**. Through the experiment, we interchangeably use the term *base* and *many-shot*. During meta-training, the learning rate is set to $0.0001$ for the ResNet12 feature extractor, and $0.05$ for all other layers in the negative prototype generator. The entire network is trained for $18k$ tasks with a SGD optimizer, where the learning rate is decayed when the validation accuracy saturates. During meta-testing, we follow [20, 24] to randomly sample 600 tasks, and report the average value with $95\%$ confidence interval for all the metrics. We use cosine similarity [10] as the similarity function to compute per-class prediction scores. For FSOR evaluation, we follow [24] to sample training and testing tasks, where we set $N = 5$ and $K = 1, 5$. For each task, we sample 15 positive queries from each few-shot class. For negative detection, we sample 5 negative classes with each containing 15 negative queries. For each GFSOR task, in addition to query samples from few-shot and negative classes, we select 75 query samples for the *base* classes (each class has at least one sample). Following the setup in [43], we randomly sample 1000 5-way GFSOR tasks to learn to generate an open-set classifier for the union of 64 base classes and 5 novel classes.

### 5.0.3 Metrics

To measure the standard closed-set classification performance, we report top-1 accuracy for FSOR tasks over few-shot classes. For GFSOR, we follow the protocol defined in [32, 46], and report both arithmetic mean and harmonic mean between mean accuracy of base samples and mean accuracy of novel samples. In addition, we report $\Delta$-value to measure the accuracy drop between prediction among specific classes (base or novel classes) and prediction among all combined classes, where a better classifier is supposed to balance the prediction and have low $\Delta$-value. To measure the negative detection performance, we follow the protocol in [20, 24] to report AUROC (area Under ROC Curve). To measure the overall open-set recognition performance, we follow the protocol in [49] to report *macro-averaged F1-scores* on all many-shot/few-shot and negative classes.

## 5.1. FSOR Results

### 5.1.1 Comparison of Negative Generator

We first compare different choices of negative generators on FSOR tasks in Tab. 1. Note that **ATT-G** and **SEMAN-G** can also be applied for FSOR and compared to other approaches since all models are trained using base set only (including base prototype) and doesn't use any extra data. We can see that attention-based methods are effective in negative generation as they are good at modeling inter-class relations. Adding class semantic information is also beneficial for discrimination. Meanwhile, by enabling multiple negative prototypes, *i.e.*, $M = 5$, we can automatically estimate the threshold $\theta_a$ with more flexibility, which then achieve consistent performance gain in F1-score, when compared generating a single negative prototype $M = 1$. For the following experiment results, we set $M = 5$ for our methods.

Table 2. 5-way 1-shot and 5-shot FSOR results. $^*$: our implementation.

| Algorithm | MiniImageNet,5-way | | | | TieredImageNet,5-way | | | |
|---|---|---|---|---|---|---|---|---|
| | 1-shot | | 5-shot | | 1-shot | | 5-shot | |
| | Acc | AUROC | Acc | AUROC | Acc | AUROC | Acc | AUROC |
| ProtoNet [37] | 64.01±0.88 | 51.81±0.93 | 80.09 | 60.39 | 68.26 | 60.73 | 83.40 | 64.96 |
| FEAT [47] | 67.02±0.85 | 57.01±0.84 | 82.02 | 63.18 | 70.52 | 63.54 | 84.74 | 70.74 |
| OpenMax [1] | 63.69±0.84 | 62.64±0.80 | 80.56 | 62.27 | 68.28 | 60.13 | 83.48 | 65.51 |
| CounterFactual* [27] | 63.7±0.83 | 64.17±0.88 | 81.44±0.54 | 71.58±0.76 | 70.08±0.94 | 71.04±0.80 | 85.36±0.60 | 78.66±0.62 |
| PEELER [24] | 65.86±0.85 | 60.57±0.83 | 80.61 | 67.35 | 69.51 | 65.20 | 84.10 | 73.27 |
| SnaTCHer-T [20] | 66.60±0.80 | 70.17±0.88 | 81.77 | 76.66 | 70.45 | 74.84 | 84.42 | 82.03 |
| SnaTCHer-L [20] | 67.60±0.83 | 69.40±0.92 | 82.36 | 76.15 | 70.85 | **74.95** | 85.23 | 80.81 |
| ATT (ours) | 67.64±0.81 | 71.35±0.68 | 82.31±0.49 | 79.85±0.58 | 69.34±0.95 | 72.74±0.78 | 83.82±0.63 | 78.66±0.65 |
| ATT-G (ours) | 68.11±0.81 | 72.41±0.72 | 83.12±0.48 | 79.85±0.57 | 70.58±0.93 | 73.43±0.78 | 85.38±0.61 | 81.64±0.63 |
| SEMAN-G (ours) | **68.24±0.82** | **72.85±0.69** | **83.48±0.48** | **82.07±0.58** | **71.06±0.92** | 74.27±0.77 | **86.02±0.58** | **82.59±0.57** |

Table 3. 5-way generalized few-shot open-set recognition results on Mini-ImageNet.$^\dagger$: Implemented by CASTLE [18]. $^*$: our implementation.

| Algorithm | 1-shot | | 5-shots | | 1-shot | 5-shots | 1-shot | 5-shots |
|---|---|---|---|---|---|---|---|---|
| | Arith. Mean | Δ | Arith. Mean | Δ | Harmonic Mean | | AUROC | |
| IFSL [32] | 54.95±0.3 | 11.84 | 63.04±0.3 | 10.66 | - | - | - | - |
| L2ML [44] | 46.25±0.04 | 27.49 | 45.81±0.03 | 35.53 | 2.98±0.06 | 1.12±0.04 | - | - |
| ProtoNet2 † | 53.93±0.08 | 22.09 | 72.64±0.08 | 11.41 | 27.73±0.19 | 68.99±0.11 | - | - |
| CASTLE [18] | 66.48±0.11 | 9.94 | 76.25±0.09 | 8.14 | 64.29±0.14 | **75.79±0.1** | - | - |
| DynamicFSL* [10] | 60.85±0.14 | 12.97 | 73.1±0.13 | 10.92 | 60.13±0.13 | 69.8±0.09 | 67.56±0.17 | 72.86±0.11 |
| ATT-G(ours) | 65.49±0.13 | 11.25 | 75.51±0.10 | 10.97 | 63.94±0.12 | 73.89±0.12 | 73.12±0.16 | 77.22±0.13 |
| SEMAN-G(ours) | **66.83±0.11** | 10.24 | **77.02±0.08** | 9.78 | **64.77±0.12** | 75.62±0.09 | **73.55±0.14** | **78.22±0.11** |

### 5.1.2 Comparison with Threshold-based Classifier

For threshold-based methods, threshold-tuning is crucial to get good recognition performance. To evaluate the overall open-set recognition, we compare macro-weighted F1-score. For threshold-based approaches, we define different thresholds and compute the corresponding F1-score. We illustrate our result in both Tab. 1 and Fig. 3(a), where we consider two threshold-based classifiers PEELER [24] and a combination of PEELER and our ATT-G method baseline, Dynamic [10], which calibrates novel prototype with base-class prototypes. In detail, we apply PEELER's open-set training strategy on top of Dynamic.

In Fig. 3(a), we simulate 45k FSOR tasks and find their optimal rejection thresholds $\theta_m$ for the threshold-based approach Dynamic+PEELER. We plot the distribution of $\theta_m$, which shows that it covers a wide range between 0 and 1. It demonstrates that different FSOR tasks may need very different rejection threshold in practice with current threshold-based approach. And the overall recognition performance largely depends on threshold selection, as is shown in Tab. 1. Our method instead automatically learn a task-adaptive rejection boundary, and we can see from Tab. 1 that all our negative envision instantiations outperform threshold-based methods. Fig. 3(b) further analyze the recognition behaviour under different openness [39]:

$$\text{openness} = 1 - \sqrt{\frac{2|\mathcal{C}^f|}{2|\mathcal{C}^f| + |\mathcal{C}^n|}}$$

where we fix $|\mathcal{C}^f| = 5$ and vary $|\mathcal{C}^n|$ from 5 to 15. Similarly, we test for 600 randomly selected FSOR tasks and take the average. As is validated by Fig. 3, our method clearly outperforms threshold-based methods at all openness levels.

### 5.1.3 Comparison with SOTA Methods

We compare our method with other SOTA methods. The baselines we compare to include standard FSL methods (ProtoNet, FEAT), large-scale OR methods (OpenMax, CounterFactual), and existing FSOR methods (PEELER, SnaTCHer). We cite most of the baseline results from [20], and additionally compare to CounterFactual, a generative OR method which synthesize fake negative images and then train a $N + 1$ classifier. To apply in our FSOR setting, we first train its GAN network on base set and use the support set to synthesize fake images. The averaged fake image feature is used as the negative prototype for FSOR.

Tab. 2 demonstrates the results. Standard FSL methods perform poorly in negative detection due to its closed-set nature. Large-scale OR methods yields unsatisfactory performance especially on 1-shot classification. Interestingly,

Table 4. Ablate study on conjugate training. We report 5-way-1-shot FSOR result using ATT-G over three metrics on both datasets.

| Dataset | Metric | w/o Conjugate | w/ Conjugate |
|---|---|---|---|
| Mini ImageNet | ACC | 66.28±0.84 | 68.11±0.81 |
| | AUROC | 71.80±0.77 | 72.41±0.72 |
| | F1-score | 46.94±0.68 | 48.19±0.71 |
| Tiered ImageNet | ACC | 70.08±0.94 | 70.58±0.93 |
| | AUROC | 71.84±0.82 | 73.43±0.78 |
| | F1-score | 50.23±0.77 | 51.56±0.81 |

CounterFactual gives a relatively fair performance on negative detection, which also validates our concept of negative envision. But it's still much worse than our few-shot-specific negative generation strategy, which validates that our approach better suits for the limited data scenario. Both **ATT-G** and **SEMAN-G** outperform other methods on Mini-ImageNet and get comparable result on Tiered-ImageNet.

#### 5.1.4 Ablation Study on Conjugate Training

Tab. 4 shows the impact of conjugate training. We observe consistent improvement on all metrics and datasets, validating that conjugate training efficiently boosts the learning process by enabling mutual supervision from two tasks.

### 5.2. GFSOR Results

In Tab. 3, we compare **ATT-G** and **SEMAN-G** with other standard methods on GFSOR tasks. Under the more challenging GFSOR setting, we achieve comparable GFSL classification accuracy with SOTA method and significantly improve the AUROC score which measures negative query detection. In addition, as GFSL methods are not trained to envision negative prototype but has more classes to recognize during evaluation, it will be challenging to manually set a threshold to reject negative queries while maintaining high classification accuracy. Thus, it is necessary to learn to dynamically generate threshold for each query for GFSOR.

### 5.3. More Experiments

We further conduct FSOR experiments on two few-shot benchmark datasets: CIFAR-FS [2], FC100 [29]. CIFAR-FS [2] contains 100 classes with the class split for (64,16,20). FC100 [29] contains 100 classes with the class split (60,20,20). Each class has 600 images and all images for the two sets are of size is 32×32. As shown in Tab. 5 and 6, we compare our methods with the threshold-based methods and direct application of large-scale open-set recognition methods. Consistent with Tab. 2, for low-resolution dataset, our method achieves the best performance on both classification accuracy and negative query rejection, which again demonstrates the effectiveness of our approach.

Table 5. 5-way FSOR results CIFAR-FS.*: our implementation.

| Algorithm | 1-shot | | 5-shot | |
|---|---|---|---|---|
| | Acc | AUROC | Acc | AUROC |
| OpenMax [1] | 71.65 | 50.21 | 85.66 | 75.78 |
| CounterFactural [27] | 71.71 | 72.57 | 85.71 | 80.44 |
| PEELER [24]* | 71.47 | 71.28 | 85.46 | 75.97 |
| Dynamic [10] | 71.56 | 66.89 | 85.78 | 76.03 |
| ATT-G (ours) | 72.43 | 76.72 | 86.52 | 84.64 |
| SEMAN-G (ours) | 74.55 | 78.10 | 86.71 | 86.47 |

Table 6. 5-way FSOR results on FC100. *: our implementation.

| Algorithm | 1-shot | | 5-shot | |
|---|---|---|---|---|
| | Acc | AUROC | Acc | AUROC |
| OpenMax [1] | 44.70 | 50.10 | 60.11 | 57.78 |
| CounterFactural [27] | 44.53 | 57.20 | 61.12 | 62.35 |
| PEELER [24]* | 44.45 | 55.86 | 60.86 | 61.07 |
| Dynamic [10] | 44.88 | 55.62 | 60.45 | 59.01 |
| ATT-G (ours) | 45.11 | 59.55 | 61.18 | 63.34 |
| SEMAN-G (ours) | 46.01 | 59.73 | 62.18 | 64.46 |

## 6. Conclusion

In this work, we show the limitation of threshold-based approaches for few-shot open-set recognition where different tasks may need very different rejection threshold and hence the tuning process could be challenging. To this end, we propose our task-adaptive negative envision approach towards (G)FSOR, where negative prototypes are computed from few/many-shot class examples. We study the different design of negative generator, and find attention-based generator works the best; adding class semantics further improves the performance. We also introduce a new conjugate class training strategy to better facilitate the learning process. Extensive experiments demonstrate the effectiveness of our approach. We note the limitation where we assume the negative source only being the images from other categories. Other possible negative sources include, e.g., data from different domains, adversarial data, etc. We will leave those as future work and study they affect our approach.

## 7. Acknowledgement

This research is based upon work supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00345. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of IARPA, DOI/IBC or the U.S. Government.

# References

[1] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1, 2, 7, 8

[2] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *Int. Conf. Learn. Represent.*, 2019. 8

[3] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 1, 2

[7] Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*, 2019. 2

[8] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017. 2

[9] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8059–8068, 2019. 4, 6

[10] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 4, 6, 7, 8

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[12] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3263–3272, October 2021. 2

[13] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[14] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[15] Guangxing Han, Xuan Zhang, and Chongrong Li. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3360–3364, 2017. 1

[16] Guangxing Han, Xuan Zhang, and Chongrong Li. Revisiting faster r-cnn: A deeper look at region proposal network. In *International Conference on Neural Information Processing*, pages 14–24, 2017. 1

[17] Guangxing Han, Xuan Zhang, and Chongrong Li. Semi-supervised dff: Decoupling detection and feature flow for video object detectors. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1811–1819, 2018. 1

[18] De-Chuan Zhan Han-Jia Ye, Hexiang Hu and Fei Sha. Learning adaptive classifiers synthesis for generalized few-shot learning. *arXiv preprint arXiv:1906.02944*, 2019. 7

[19] 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. 6

[20] Minki Jeong, Seokeon Choi, and Changick Kim. Few-shot open-set recognition by transformation consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12566–12575, 2021. 2, 3, 5, 6, 7

[21] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 2

[22] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12576–12584, 2020. 4

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[24] Bo Liu, Hao Kang, Haoxiang Li, Gang Hua, and Nuno Vasconcelos. Few-shot open-set recognition using meta-learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 5, 6, 7, 8

[25] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[26] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. 5

[27] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 7, 8

[28] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2

[29] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In S. Bengio, H. Wallach, H.

Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 721–731. Curran Associates, Inc., 2018. 2, 8

[30] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 6

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[32] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S Zemel. Incremental few-shot learning with attention attractor networks. *arXiv preprint arXiv:1810.07218*, 2018. 6, 7

[33] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *iclr*, 2018. 5, 6

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

[35] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. In *IEEE Trans. Pattern Anal. Mach. Intell.*, number 7, pages 1757–1772, 2013. 1

[36] Patrick Schlachter, Yiwen Liao, and Bin Yang. Open-set recognition using intra-class splitting. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 1, 2

[37] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, 2017. 1, 2, 3, 5, 7

[38] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019. 2

[39] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13480–13489, 2020. 7

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 2

[41] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020. 4, 6

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[43] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Adv. Neural Inform. Process. Syst.*, 2016. 1, 2, 5, 6

[44] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 7032–7042, 2017. 7

[45] Chen Xing, Negar Rostamzadeh, Boris N Oreshkin, and Pedro O Pinheiro. Adaptive cross-modal few-shot learning. *arXiv preprint arXiv:1902.07104*, 2019. 4

[46] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning classifier synthesis for generalized few-shot learning. 2019. 6

[47] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 7

[48] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2

[49] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2021. 2, 6