PLATINUM: Semi-Supervised Model Agnostic Meta-Learning using Submodular Mutual Information

Changbin Li [*1] **Suraj Kothawade** [*1] **Feng Chen** [1] **Rishabh Iyer** [1]

Abstract

Few-shot classification (FSC) requires training models using a few (typically one to five) data points per class. Meta-learning has proven to be able to learn a parametrized model for FSC by training on various other classification tasks. In this work, we propose PLATINUM (semi-suPervised modeL Agnostic meTa learnIng usiNg sUbmodular Mutual information), a novel semi-supervised model agnostic meta learning framework that uses the submodular mutual information (SMI) functions to boost the performance of FSC. PLATINUM leverages unlabeled data in the inner and outer loop using SMI functions during meta-training and obtains richer meta-learned parameterizations. We study the performance of PLATINUM in two scenarios - 1) where the unlabeled data points belong to the same set of classes as the labeled set of a certain episode, and 2) where there exist out-of-distribution classes that do not belong to the labeled set. We evaluate our method on various settings on the *mini*ImageNet, *tiered*ImageNet and CIFAR-FS datasets. Our experiments show that PLATINUM outperforms MAML and semi-supervised approaches like pseduo-labeling for semi-supervised FSC, especially for small ratio of labeled to unlabeled samples.

1. Introduction

Deep neural networks (DNNs) have proven to be successful in a variety of domains. However, they require large amounts of data, which might not be available for all desired tasks. In such low data regimes, they struggle to perform well. A well known approach to mitigate this problem is meta-learning, which aims to learn from multiple smaller tasks that are related to the target task. The most promising meta-learning techniques that improve the performance of DNNs are gradient based meta-learning methods (Finn et al., 2017; Rusu et al., 2018; Sun et al., 2019). Typically, these methods are designed to operate for few-shot learning. A natural way to improve the performance of meta-learning techniques is by using additional unlabeled data. Semi-supervised techniques are known to use unlabeled data to improve the performance on tasks with relatively small number of labeled data (Oliver et al., 2018; Chapelle et al., 2009).

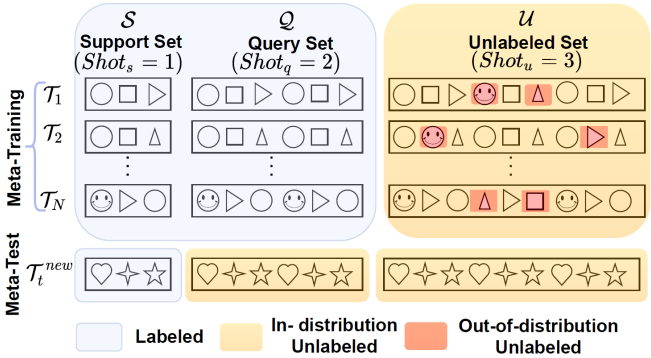

Figure 1: Semi-supervised few-shot learning setup. During meta-training, the goal is to iterate overs tasks $\mathcal{T}_1 \cdots \mathcal{T}_N$ and meta-learn a parametrization using the support set $\mathcal{S}$, query set $\mathcal{Q}$, and the unlabeled set $\mathcal{U}$. During meta-testing, the learned parametrization is used as an initialization and a model is trained using the $\mathcal{S}$ and $\mathcal{U}$ to perform well on $\mathcal{Q}$. In any task, $\mathcal{U}$ may contain data points that are out-of-distribution, *i.e.*not pertinent to the classes of data points in $\mathcal{S}$.

In this paper, we focus on few-shot classification using Model Agnostic Meta-Learning (MAML) and improve it via semi-supervised learning (see Fig. 1). In many realistic classification tasks, although the labeled data is scarce, there is plenty of unlabeled data available for training a classifier. Towards this goal, we propose PLATINUM, a novel framework that embeds semi-supervision in the MAML framework by using **submodular mutual information (SMI)** (Iyer et al., 2021) functions as per-class acquisition functions. We observe that embedding semi-supervision in the MAML framework is non-trivial, since simply using a pseudo-labeling approach in the inner loop does not improve the performance. This lack of improvement occurs due to either noisy pseudo-

[*]Equal contribution [1]University of Texas, Dallas, USA. Correspondence to: Suraj Kothawade <suraj.kothawade@utdallas.edu>, Changbin Li <changbin.li@utdallas.edu>.

labels or class imbalance caused due to pseudo-labels being confident only for certain classes. To overcome these issues, PLATINUM uses a class-wise unique instantiations of SMI functions to provide per-class semi-supervision. Furthermore, these per-class acquired subsets are diverse, leading to a richer meta-learned parameterization (see Fig. 2).

1.1. Related Work

Few Shot Learning. There has been an extensive amount of work in few-shot learning, which has mainly revolved around supervised learning. Although our framework is embedding semi-supervision into MAML which belongs to the gradient descent family of methods, the few-shot learning literature can be broadly divided into the following categories: 1) Metric learning methods (Vinyals et al., 2016; Snell et al., 2017; Sun et al., 2019) which deal with learning a similarity space where the task can be efficiently done with a few labeled data points. 2) Memory networks (Munkhdalai & Yu, 2017; Santoro et al., 2016; Oreshkin et al., 2018; Mishra et al., 2017), which focus on learning to store "experience" from previously observed tasks in the interest of generalizing to newer tasks. 3) Gradient based meta-learning methods (Finn et al., 2017; 2018; Antoniou et al., 2018; Ravi & Larochelle, 2017; Grant et al., 2018; Zhang et al., 2018; Sun et al., 2019) which aim to meta-learn a model in the outer loop that is used as a starting point in the inner loop for a new few-shot task. The PLATINUM framework embeds semi-supervision for gradient descent based methods that use an outer-inner bi-level optimization.

Semi-supervised learning (SSL). The goal of SSL methods is to leverage unlabeled data alongside the labeled data to obtain a better representation of the dataset than supervised learning (Oliver et al., 2018). The most basic SSL method, Pseudo-labeling (Lee et al., 2013) uses model predictions as target labels as a regularizer, and a standard supervised loss function for the unlabeled dataset. Some SSL methods like Π-Model (Laine & Aila, 2016; Sajjadi et al., 2016) and Mean Teacher (Tarvainen & Valpola, 2017) use consistency regularization, by using data augmentation and dropout techniques. Mean Teacher obtains a more stable target output by using an exponential moving average of parameters across previous epochs. Virtual Adversarial Training (VAT) (Miyato et al., 2018) uses an effective regularization technique that uses slight perturbations such that the prediction of the unlabeled samples is affected the most. More recent techniques like FixMatch (Sohn et al., 2020), MixMatch (Berthelot et al., 2019) and UDA (Xie et al., 2019) use data augmentations like flip, rotation, and crops to predict pseudo-labels. In this paper, we propose a new SSL technique that uses class-wise instantiations of SMI functions that mitigates the issue of class-imbalance in selected subsets and is comparatively robust to OOD classes in the unlabeled set.

Semi-supervised few-shot learning. There are two categories for semi-supervised few-shot learning: meta-learning based and transfer learning based. For meta-learning based approaches, (Ren et al., 2018) propose new extensions of Prototypical networks (Snell et al., 2017) by viewing each prototype as a cluster center and tuning the cluster locations using soft K-means such that the data points in support and unlabeled sets fit better. More recently, (Li et al., 2019) proposes learning to self-train (LST) which aims to meta-learn how to cherry-pick and label data points from the unlabeled set and optimizes weights of these pseudo-labels. However, their method uses meta-transfer learning (MTL) (Sun et al., 2019) which requires a pre-trained model. Furthermore, MTL requires the classes across all training tasks to be known beforehand for scaling and shifting the parameters of the pre-trained network. Unfortunately, such meta-data about the dataset may not be available in most realistic scenarios. On the other hand, our PLATINUM framework does not require a pre-trained network or any meta-data for embedding semi-supervision in gradient descent based methods. For transfer learning based approaches (Yu et al., 2020; Wang et al., 2020; Huang et al., 2021), they assume all examples of all training classes are labeled so that a feature extractor could be pretrained based on them. In contrast, we assume there are few examples per class are labeled during both meta-training and meta-test, which is different from transfer learning based approaches. In addition, transfer learning based approaches do not leverage episodes training strategy, which is different from ours.

Data subset selection (DSS). DSS using submodular functions has been studied in the context of various applications like video summarization (Kaushal et al., 2020; 2019b), image-collection summarization (Tschiatschek et al., 2014; Kothawade et al., 2020), efficient learning (Kaushal et al., 2019a; Killamsetty et al., 2021b;a; Liu et al., 2017), targeted learning (Kothawade et al., 2021c;b), *etc.* Recently, (Kothawade et al., 2021c) used the SMI functions for improving the performance of rare classes in the context of image classification, and (Kothawade et al., 2021b) used them for mining rare objects and slices for improving object detectors. (Kothawade et al., 2021a) used the submodular information measures as acquisition functions for active learning in scenarios with class imbalance, redundancy and OOD data. (Killamsetty et al., 2021b;a) studied the role of submodular functions and coresets for compute-efficient training of deep models.

1.2. Our Contributions

The following are our main contributions: 1) Given the limitations of existing approaches, we propose PLATINUM

(see Sec. 3) that uses per-class semi-supervision using SMI functions, thereby preventing class-imbalance in the selected subset for semi-supervision. 2) Our framework learns richer parameterizations by embedding semi-supervision in the inner and outer loop of MAML. 3) We conduct extensive experiments on *mini*ImageNet (Vinyals et al., 2016), *tiered*ImageNet (Ren et al., 2018), and CIFAR-FS (Bertinetto et al., 2018) datasets where the unlabeled set has in-distribution and out-of-distribution (OOD) classes. 4) We conduct various ablation experiments where we study the effect of varying the: i) ratio of labeled and unlabeled data points, ii) number of OOD classes, and iii) inner and outer loop selection for semi-supervision.

2. Preliminaries

2.1. Model Agnostic Meta Learning (MAML)

MAML (Finn et al., 2017) is a representative of gradient-based meta-learning approach, its goal is to obtain optimal initial model parameters for *unseen* tasks. Suppose there are a set of meta-training tasks sampled from a task distribution $p(\mathcal{T})$. Each task $\mathcal{T}_i$ is split into support (training) set and query (validation) set $\{\mathcal{S}_i, \mathcal{Q}_i\}$. As a bi-level optimization problem, in the *inner loop*, MAML adapts the task-specific model parameters ϕ_i from initialization parameters θ for task $\mathcal{T}_i$ based on its support set: $\phi_i = \text{argmin}_\theta \left[L(\theta; \mathcal{S}_i)\right]$ (L is the loss of model parameterized by θ on data $\mathcal{S}_i$). The loss of adapted model ϕ_i on the corresponding query set $L(\phi_i; \mathcal{Q}_i)$ is used to evaluate the performance. In the *outer loop*, the averaged query set loss is minimized to obtain the optimal initial parameters. Therefore, the objective function could be formulated as follows:

$$\theta^* = \underset{\theta \in \Theta}{\text{argmin}}\, \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \left[L(\text{Alg}(\theta; \mathcal{S}_i); \mathcal{Q}_i)\right] \quad (1)$$

where $\text{Alg}(\theta; \mathcal{S}_i)$ corresponds to single or multiple gradient descent steps in the inner-level task-specific adaptation. In case of single-step gradient update, $\text{Alg}(\theta; \mathcal{S}_i)$ can be specified as following:

$$\phi_i = \text{Alg}(\theta; \mathcal{S}_i) \approx \theta - \alpha \nabla_\theta L(\theta; \mathcal{S}_i) \quad (2)$$

where α is a learning rate. The learned meta-parameters θ^* from outer-level will be leveraged as model initialization for the *unseen* tasks during meta-test stage. A table of notations with corresponding explanations is given in Appendix A.

2.2. Submodular Mutual Information

Submodular functions. Submodular functions (Tohidi et al., 2020; Bach, 2011; 2019) have been widely used for data subset selection as they naturally model properties like coverage, representation, diversity, *etc.*. Given a ground-set of n data points $\mathcal{V} = \{1, 2, 3, \cdots, n\}$, and a set function $f : 2^{\mathcal{V}} \rightarrow \mathbb{R}$. The set function f is known to be submodular (Fujishige, 2005) if for $x \in \mathcal{V}$, $f(\mathcal{A} \cup x) - f(\mathcal{A}) \geq f(\mathcal{B} \cup x) - f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $x \notin \mathcal{B}$. We use two well known submodular functions in this work, facility location (FL) and graph-cut (GC) (see Tab. 1(a)) that can be instantiated using a similarity kernel containing pairwise similarities between all data points. In general, submodular functions admit a $1 - \frac{1}{e}$ constant factor approximation (Nemhauser et al., 1978) for cardinality constraint maximization. Furthermore, they can be optimized in *near-linear* time using greedy algorithms (Mirzasoleiman et al., 2015).

Submodular Mutual Information (SMI). In this work, we use the SMI instantiations of the above submodular functions to provide semi-supervision. Particularly, we use FLMI and GCMI where the underlying submodular function is FL and GC respectively (see Tab. 1(b)). The SMI functions can be used to select data points that are semantically *similar* to the data points in a given query set (Kothawade et al., 2021a;c;b). To obtain pseudo-labels, we use exemplars from a particular class in the query set used to instantiate an SMI function. The subset obtained by optimizing this SMI function is then assigned labels of the class of the data points used in the query set. Formally, the submodular mutual information (SMI) is defined as $I_f(\mathcal{A}; \mathcal{R}) = f(\mathcal{A}) + f(\mathcal{R}) - f(\mathcal{A} \cup \mathcal{R})$, where $\mathcal{R}$ is a query set. Note that (Iyer et al., 2021; Kothawade et al., 2021c) propose a few other SMI functions. However, we use only the FLMI and GCMI variants in the interest of scalability (see Sec. 3.2).

Table 1: Instantiations of different submodular functions.

(a) Instantiations of submodular functions.

SF	$f(\mathcal{A})$
FL	$\sum_{i \in \mathcal{U}} \max_{j \in \mathcal{A}} S_{ij}$
GC	$\sum_{i \in \mathcal{A}, j \in \mathcal{U}} S_{ij} - \sum_{i,j \in \mathcal{A}} S_{ij}$

(b) Instantiations of SMI functions.

SMI	$I_f(\mathcal{A}; \mathcal{R})$
FLMI	$\sum_{i \in \mathcal{R}} \max_{j \in \mathcal{A}} S_{ij} + \sum_{i \in \mathcal{A}} \max_{j \in \mathcal{R}} S_{ij}$
GCMI	$2 \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{R}} S_{ij}$

3. PLATINUM: Our Semi-Supervised Meta-Learning Framework

In this section, we define the semi-supervised few-shot classification setting considered in this work (see Fig. 1). We start with N meta-training tasks $\mathcal{T}_1 \cdots \mathcal{T}_N$. For each task $\mathcal{T}_i$, we have, $\{(\mathcal{S}_i, \mathcal{Q}_i, \mathcal{U}_i)\}_{i=1}^N$, where $\mathcal{S}$ is the labeled support set, $\mathcal{Q}$ is the query set with unseen data points for test, and $\mathcal{U}$ is the unlabeled set. In our experiments (Sec. 4), similar to (Ren et al., 2018), we consider both settings, where $\mathcal{U}$ *does* or *does not* consist of OOD classes.

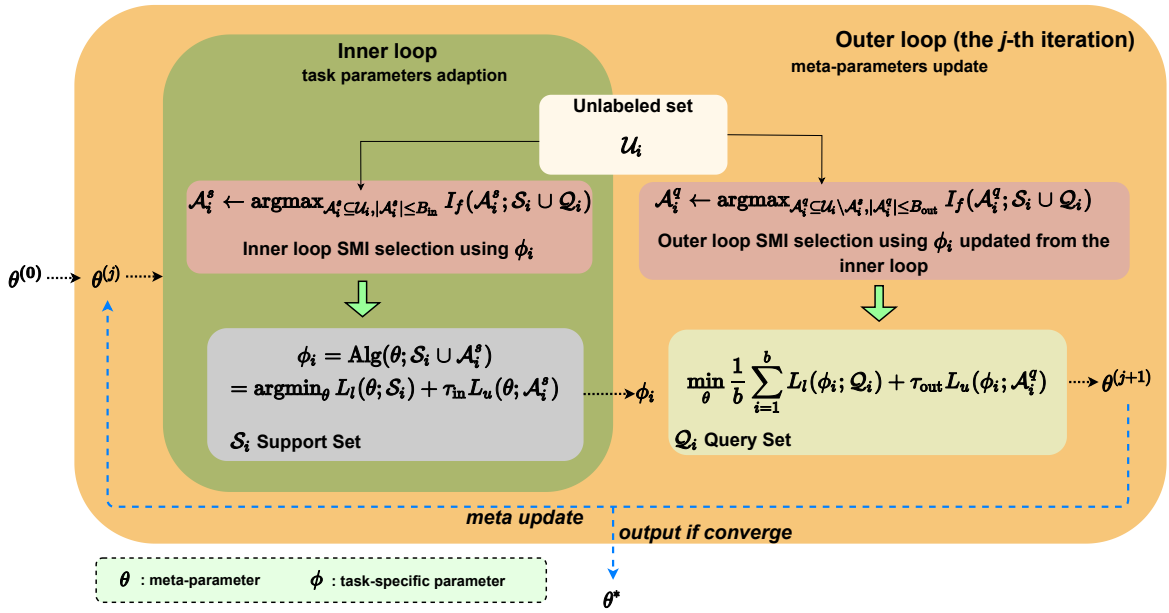

Figure 2: Overview of our PLATINUM framework that solves a semi-supervised few-shot learning problem. For a specific task $\mathcal{T}_i$, in each inner loop and outer loop gradient update step, we select a subset from the unlabeled set by maximizing the per-class SMI function (see Algorithm 1). In each *inner loop* step, the selected subset $\mathcal{A}_i^s$ and support set $\mathcal{S}_i$ will be used to update model parameters ϕ_i using Equ. (4). In the *outer-loop* of the meta-training stage, another subset $\mathcal{A}_i^q$ will be selected after inner loop selection according to the updated model parameters ϕ_i. Meta-parameters θ would be updated based on $\mathcal{A}_i^q$ and the query set $\mathcal{Q}_i$ using Equ. (3). It should be noted that, as temperature annealing coefficients, τ_{in} is a function of inner step t_{in}, and τ_{out} is a function of the global iteration index.

The goal of our method is to obtain the optimal initial parameters that result into faster adaptation of the classifier to a new task. To do so, we minimize the following meta-training objective:

$$\theta^* = \underset{\theta \in \mathbf{\Theta}}{\text{argmin}}\, \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{J}(\theta)$$
$$\text{where } \mathcal{J}(\theta) = L(\text{Alg}(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s); \mathcal{Q}_i \cup \mathcal{A}_i^q) \tag{3}$$

Here, $\mathcal{A}_i^s \subseteq \mathcal{U}_i$ and $\mathcal{A}_i^q \subseteq \mathcal{U}_i$ are selected subsets with hypothesized labels in the *inner loop* and *outer loop*, respectively (see Sec. 3.1). $\text{Alg}(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s)$ corresponds to single or multiple updates on support set $\mathcal{S}_i$, and hypothesized labeled subset $\mathcal{A}_i^s$ for task $\mathcal{T}_i$ in the *inner loop*. We consider multiple steps in the inner loop in practice.

$$\begin{aligned} \phi_i &= \text{Alg}(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s) \\ &= \text{argmin}_\theta\, L\left(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s\right) \\ &= \phi_i - \nabla_\phi L\left(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s\right) \end{aligned} \tag{4}$$

In addition to the inner loop selection, PLATINUM embeds semi-supervision for the outer-loop selection. We do so since the outer-level also corresponds to the meta-training objective of generalizing well, especially when data is scarce to update meta-parameters. The model parameters updated from inner loop ϕ_i would be used to do outer loop selection. We perform the outer-loop update as follows:

$$\mathcal{J}(\theta) = L(\phi_i; (\mathcal{Q}_i \cup \mathcal{A}_i^q)) \tag{5}$$

For meta-testing, we sample a new *unseen* task $\mathcal{T}^{new}$. The *unseen* task for meta-testing is made of disjoint set of data points and classes from the tasks seen during meta-training. Next, we use the parameters from obtained from the meta-training stage and initialize a model and train it on $\{\mathcal{S}^{new}, \mathcal{U}^{new}\}$. Finally, we evaluate the model on $\mathcal{Q}^{new}$, and report the average accuracy across all unseen tasks.

3.1. Leveraging Full Potential of SSL during Meta-Training

In this section, we discuss the meta-training pipeline of PLATINUM. Particularly, we emphasis on *inner loop* and *outer loop* semi-supervision embeded using *class-wise* SMI instantiations. We detail our meta-training pipeline in Algorithm 1.

For any task $\mathcal{T}_i \sim p(\mathcal{T})$, we first initialize the model parameters $\phi_i \leftarrow \theta$, where θ is meta-learned by optimizing the outer loop on the previous tasks. Using parameters ϕ_i, we compute an embedding containing class probabilities for each data point belonging to the unlabeled set $\mathcal{U}_i$. Emprically, we found out that using the class probabilities based on the classes belonging to $\mathcal{T}_i$ was as effective as using a larger and computationally expensive embedding like last layer features or gradients. Since the support and

Algorithm 1 PLATINUM (Meta-Training)

Require: task distribution: $p(\mathcal{T})$, Base model with parameters θ, Batch size of tasks: b, Budge of selected samples from unlabeled set: B_{in}, B_{out}

1: Randomly initialize θ
2: **while** not converge **do**
3: Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{b} \sim p(\mathcal{T})$
4: **for** each task $\mathcal{T}_i = \{\mathcal{S}_i, \mathcal{Q}_i, \mathcal{U}_i\}, i \in [b]$ **do**
5: Initialize model parameters $\phi_i \leftarrow \theta$
6: **for** each inner step t **do**
7: $\mathcal{P}_{\mathcal{U}_i} \leftarrow \phi_i(\mathcal{U}_i)$
8: $\mathcal{X} \leftarrow$ COSINE_SIMILARITY$(\mathcal{P}_{\mathcal{U}_i}, \{\mathcal{P}_{\mathcal{S}_i} \cup \mathcal{P}_{\mathcal{Q}_i}\})$
9: Instantiate a submodular function f based on $\mathcal{X}$. */∗ inner loop selection ∗/*
10: $\mathcal{A}_{it}^{s} \leftarrow \text{argmax}_{\mathcal{A}_{it}^{s} \subseteq \mathcal{U}_i, |\mathcal{A}_{it}^{s}| \leq B_{\text{in}}} I_f(\mathcal{A}_{it}^{s}; \mathcal{S}_i \cup \mathcal{Q}_i)$ {Acquire subset with hypothesized labels using per-class greedy maximization, Equ. (6)}
11: $\phi_i \leftarrow \phi_i - \nabla_{\phi} L\left(\theta; \mathcal{S}_i \cup \mathcal{A}_i^{s}\right)$ Update ϕ_i by gradient descent, Equ. (4)
12: $\mathcal{A}_i^{s} \leftarrow \mathcal{A}_i^{s} \cup \mathcal{A}_{it}^{s}$
13: **end for**
14: $\mathcal{P}_{\mathcal{U}_i \setminus \mathcal{A}_i^{s}} \leftarrow \phi_i(\mathcal{U}_i \setminus \mathcal{A}_i^{s})$
15: $\mathcal{X} \leftarrow$ COSINE_SIMILARITY$(\mathcal{P}_{\mathcal{U}_i \setminus \mathcal{A}_i^{s}}, \{\mathcal{P}_{\mathcal{S}_i} \cup \mathcal{P}_{\mathcal{Q}_i}\})$ */∗outer loop selection∗/*
16: $\mathcal{A}_i^{q} \leftarrow \text{argmax}_{\mathcal{A}_i^{q} \subseteq \mathcal{U}_i \setminus \mathcal{A}_i^{s}, |\mathcal{A}_i^{q}| \leq B_{\text{out}}} I_f(\mathcal{A}_i^{q}; \mathcal{S}_i \cup \mathcal{Q}_i)$ {Acquire subset with hypothesized labels using per-class greedy maximization, Equ. (6)}
17: **end for** */∗meta update (outer loop) ∗/*
18: Obtain $\theta^{(t+1)}$ by Equ. (3) using $\{\mathcal{Q}_i \cup \mathcal{A}_i^{q}\}_{i=1}^{b}$
19: **end while**
20: Return the meta-learned parameters θ.

query set have labels during meta-training, we use a one-hot vector to represent the data points in $\mathcal{S}_i$ and $\mathcal{U}_i$, where the probabilty of the class corresponding to the label is set to one. Next, we compute a pairwise similarities $\mathcal{X}_{pq}$, where $p \in \{\mathcal{S}_i \cup \mathcal{Q}_i\}, q \in \mathcal{U}_i$. For each class c, we instantiate an SMI function I_f^c (Tab. 1) by using a sub-matrix $\mathcal{X}^c$ with pairwise similarities $\mathcal{X}_{pq}^c$ such that p belongs to class c, $\forall p \in \{\mathcal{S}_i \cup \mathcal{Q}_i\}$. We then maximize I_f^c with a budget of B/C as follows:

$$\mathcal{A}^c \leftarrow \underset{\mathcal{A}^c \subseteq \mathcal{U}_i, |\mathcal{A}^c| \leq B/C}{\text{argmax}} I_f^c(\mathcal{A}^c; \mathcal{S}_i \cup \mathcal{Q}_i) \tag{6}$$

where, C is the number of classes in the current task $\mathcal{T}$. Since $\mathcal{A}^c$ is obtained by optimizing the SMI function $I_f^c(\mathcal{A}^c; \mathcal{R})$, where $\mathcal{R}$ contains data points *only* from class c, we assigned the hypothesized label c to all data points in $\mathcal{A}^c$. We obtain $\mathcal{A}^s = \{\mathcal{A}^1 \cup \mathcal{A}^2 \cdots \cup \mathcal{A}^C\}$ by repeating the selection for each class. Finally, we update ϕ_i in the inner loop by using gradient descent on $\mathcal{S}_i$ and $\mathcal{A}_i^s$. Note that ϕ_i is updated for T_{in} steps in the inner loop. Similarly, we obtain $\mathcal{A}_i^q$ by using the SMI functions to embed semi-supervision in the outer loop and update θ by gradient descent on $\mathcal{Q}_i$ and $\mathcal{A}_i^q$. We summarize the inner loop and outer loop optimization problems in Equ. (7) and Equ. (8) respectively, and discuss them in more detail:

$$\mathcal{A}_i^s \leftarrow \underset{\mathcal{A}_i^s \subseteq \mathcal{U}_i, |\mathcal{A}_i^s| \leq B_{\text{in}}}{\text{argmax}} I_f(\mathcal{A}_i^s; \mathcal{S}_i \cup \mathcal{Q}_i) \tag{7}$$

$$\mathcal{A}_i^q \leftarrow \underset{\mathcal{A}_i^q \subseteq \mathcal{U}_i \setminus \mathcal{A}_i^s, |\mathcal{A}_i^q| \leq B_{\text{out}}}{\text{argmax}} I_f(\mathcal{A}_i^q; \mathcal{S}_i \cup \mathcal{Q}_i) \tag{8}$$

where B_{in} and B_{out} are selection budget per class.

Inner loop. Although MAML could achieve single step gradient update in the inner loop, it is not common to have good adaptation in practice especially considering the involving of additional unlabeled set. To illustrate this clearly, we assume there are T_{in} steps during the model adaptation in the inner loop. Inspired from (Lee et al., 2013), we add some unlabeled examples to update a task-specific model ϕ_i. Different from (Lee et al., 2013), we do not use all examples in the unlabeled set because efficiency matters in meta-learning training procedure. Therefore, the loss function in the inner loop is formulated as below:

$$L\left(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s\right) = L_l\left(\theta; \mathcal{S}_i\right) + \tau_{\text{in}} L_u\left(\theta; \mathcal{A}_i^s\right) \tag{9}$$

where $\mathcal{A}_i^s$ is the selected examples with pseudo labels.We define L_l as the loss based on examples with true labels, and L_u as the loss function based on hypothesized labeled examples. Similar formulations have been used in conventional semi-supervised learning approaches, such as Pseudo-Label (Lee et al., 2013) and VAT (Miyato et al., 2018).τ_{in} is a temperature annealing coefficient:

$$\tau_{\text{in}}(t) = \begin{cases} 0 & t < 2 \\ \exp\left(-5(1 - \frac{t}{T_{\text{in}}})^2\right) & 2 < t \leq T_{\text{in}} \end{cases} \tag{10}$$

Note that we consider multi inner step updates, and SMI subset selection happens in each step (line 6-13 in Algorithm 1).

Outer loop. Considering the meta-parameters are updated in the outer loop based on the labeled query set, and there are few labeled examples per class, it is beneficial to augment the query set aiming to generalize well for novel class in the meta-test stage. Considering the unlabeled examples, the loss function in the outer loop could be:

$$\begin{aligned} \mathcal{J}(\theta) &= L(\phi_i; \mathcal{Q}_i \cup \mathcal{A}_i^q) \\ &= L_l\left(\phi_i; \mathcal{Q}_i\right) + \tau_{\text{out}} L_u\left(\phi_i; \mathcal{A}_i^q\right) \end{aligned} \tag{11}$$

where $\mathcal{A}_i^q$ is the selected examples with pseudo labels, and τ_{out} is a temperature annealing coefficient:

$$\tau_{\text{out}}(j) = \begin{cases} \exp\left(-5(1 - \frac{t}{T_{\text{warm}}})^2\right) & 0 < j \leq T_{\text{warm}} \\ 1 & T_{\text{warm}} < j \leq T_{\text{out}} \end{cases} \tag{12}$$

T_{out} is the total number of epochs during meta-training procedure. T_{warm} is a warm starting epoch index. More detailed selection process explanation is given in Appendix B.

Table 2: Few-shot classification accuracies (%) on *mini*ImageNet. (†: only supervised setting is considered.)

Methods	*1-shot*		*5-shot*	
	w/o OOD	*w/ OOD*	*w/o OOD*	*w/ OOD*
Soft k-Means (Ren et al., 2018)	24.61±0.64	23.57±0.63	38.20±1.64	38.07±1.53
Soft k-Means+Cluster (Ren et al., 2018)	15.76±0.59	9.77±0.51	33.65±1.53	30.47±1.42
Masked Soft k-Means (Ren et al., 2018)	25.48±0.67	25.03±0.68	39.33±1.55	38.48±1.74
TPN-semi (Liu et al., 2019)	40.25±0.92	26.70±0.98	46.27±1.67	36.81±0.87
MAML† (Finn et al., 2017)	35.26±0.85	35.26±0.85	60.22±0.83	60.20±0.83
VAT (Miyato et al., 2018)	36.55±0.86	34.03±0.84	61.60±0.83	61.24±0.88
PL (Lee et al., 2013)	37.71±0.94	35.16±0.85	60.64±0.92	60.31±0.87
GCMI (ours)	41.94±0.96	**42.57**±0.93	63.62±0.95	**63.54**±0.94
FLMI (ours)	**42.27**±0.95	41.53±0.97	**63.80**±0.92	63.44±0.99

Table 3: Few-shot classification accuracies (%) on *tiered*ImageNet. (†: only supervised setting is considered.)

Methods	*1-shot*		*5-shot*	
	w/o OOD	*w/ OOD*	*w/o OOD*	*w/ OOD*
Soft k-Means (Ren et al., 2018)	27.53±0.74	27.04±0.76	44.63±1.19	44.78±1.05
Soft k-Means+Cluster (Ren et al., 2018)	30.48±0.84	31.30±0.86	46.93±1.18	49.33±1.17
Masked Soft k-Means (Ren et al., 2018)	33.85±0.84	32.99±0.87	47.63±1.12	47.35±1.08
TPN-semi (Liu et al., 2019)	44.13±1.04	31.83±1.09	58.53±1.57	56.92±1.67
MAML† (Finn et al., 2017)	41.96±0.84	41.96±0.84	61.30±0.85	61.30±0.85
VAT (Miyato et al., 2018)	41.52±0.82	41.51±0.79	59.98±0.83	60.01±0.87
PL (Lee et al., 2013)	41.22±0.89	40.87±0.83	61.70±0.77	60.57±0.87
GCMI (ours)	45.49±0.91	45.55±0.90	63.67±0.83	**62.59**±0.85
FLMI (ours)	**45.63**±0.86	**46.19**±0.94	**63.75**±0.87	62.19±0.91

Table 4: Few-shot classification accuracies (%) on CIFAR-FS. (†: only supervised setting is considered.)

Methods	*1-shot*		*5-shot*	
	w/o OOD	*w/ OOD*	*w/o OOD*	*w/ OOD*
MAML† (Finn et al., 2017)	37.90±0.91	37.90±0.91	52.60±0.89	52.60±0.89
VAT (Miyato et al., 2018)	39.48±0.83	38.91±0.88	53.20±0.80	52.44±0.83
PL (Lee et al., 2013)	38.11±0.87	37.29±0.92	52.83±0.82	52.42±0.91
GCMI (ours)	40.47±0.88	40.10±0.89	**55.01**±0.84	**54.42**±0.92
FLMI (ours)	**40.96**±0.86	**40.48**±0.87	54.94±0.80	54.16±0.91

3.2. Scalability of SMI Optimization

We chose to embed semi-supervision using FLMI and GCMI in our framework due to their scalability benefits (Kothawade et al., 2021a;b). Asymptotically, the time and space complexity of computing a similarity matrix $\mathcal{X}$ for FLMI and GCMI is only $|\mathcal{R}| \times |\mathcal{U}|$. Since in the few-shot learning setting, we set $\mathcal{R} \leftarrow \mathcal{S} \cup \mathcal{Q}$ which is comparatively much smaller than $\mathcal{U}$, the complexity of these SMI functions is only $|\mathcal{U}|$. Moreover, the SMI functions that we use are monotone and submodular which allows a $1 - \frac{1}{e}$ constant factor approximation (Nemhauser et al., 1978). Hence, for optimizing the SMI functions, we use a greedy algorithm (Nemhauser et al., 1978) using memoization (Iyer & Bilmes, 2019). This leads to an amortized cost of $|\mathcal{U}| \log |\mathcal{U}|$ which can be further reduced to $|\mathcal{U}|$ using the lazier than lazy greedy algorithm (Mirzasoleiman et al., 2015). Hence, FLMI and GCMI can be optimized in linear time, making it applicable to few-shot learning datasets with a large number of tasks and large unlabeled sets.

Time complexity of PLATINUM. Since FLMI and GCMI can be optimized in linear time, asymptotically it does not change in terms of the worst case in MAML. Therefore, the iteration complexity is still $\mathcal{O}(1/\epsilon^2)$ (Fallah et al., 2020). Tab. 5 shows the empirical running time per epoch (100 iterations, one task per iteration) for MAML and our proposed GCMI and FLMI. Therefore, it is safe to say that our proposed framework PLATINUM would not slow down the original meta-learning framework (such as MAML).

Table 5: Running time (training time of 100 tasks) comparison on *mini*ImageNet domains for 1-shot (5-shot) 5-way experiments without OOD classes in unlabeled set.

METHODS	1-SHOT (S)	5-SHOT (S)
MAML	23.92	49.83
GCMI	27.33	58.91
FLMI	28.94	56.04

4. Experiments

In this section, we evaluate PLATINUM for semi-supervised few-shot image classification by comparing the accuracy attained at meta-testing. In Sec. 4.2, we compare our

Table 6: Few-shot classification accuracies (%) of different labeled ratios on *mini*ImageNet. (Due to the space limitation, we show results with 95% confidence interval in Appendix C.)

Methods	*1-shot* 1%	5%	10%	20%	30%	*5-shot* 1%	10%	20%	30%
Soft k-Means (Ren et al., 2018)	24.61	38.45	40.65	42.55	44.09	38.20	56.27	60.13	62.47
Soft k-Means+Cluster (Ren et al., 2018)	15.76	38.34	41.15	45.17	47.05	33.65	56.87	60.33	62.43
Masked Soft k-Means (Ren et al., 2018)	25.48	39.03	42.91	45.31	47.17	39.33	57.20	62.50	63.00
TPN-semi (Liu et al., 2019)	40.25	42.40	45.78	48.02	47.52	46.27	60.55	62.43	63.10
MAML (Finn et al., 2017)	35.26	42.51	44.29	45.10	45.26	60.22	61.06	63.18	65.60
PL (Lee et al., 2013)	37.71	44.04	46.58	45.13	44.37	60.64	61.17	63.06	65.14
GCMI (ours)	41.94	44.98	46.85	47.72	48.93	63.62	**62.72**	64.78	65.96
FLMI (ours)	**42.27**	**45.01**	**47.84**	**47.82**	**48.98**	**63.80**	62.60	**65.16**	**66.10**

method with the state-of-the-art techniques on a diverse set of datasets and settings. In Sec. 4.3, we discuss multiple ablation studies by varying the number OOD classes in the unlabeled set and studying the effect of the proposed semi-supervision in the inner *and* outer loop.

In order to demonstrate the effectiveness of PLATINUM, we aim to study two questions:

Q1: Can PLATINUM be successfully applied to semi-supervised few shot classification scenario with very few labeled examples on the top of MAML and boost the performance of MAML with the additional unlabeled set?
Q2: In realistic scenarios, the unlabeled set is bound to have OOD data. Can PLATINUM provide robust semi-supervision by ignoring the OOD data in the unlabeled set?

4.1. Datasets and Implementation details

Datasets. We conduct experiments on three datasets: *mini*ImageNet (Vinyals et al., 2016), *tiered*ImageNet (Ren et al., 2018), and CIFAR-FS (Bertinetto et al., 2018). Both *mini*ImageNet and *tiered*ImageNet are modified subsets of the ILSVRC-12 dataset (Russakovsky et al., 2015). ***mini*ImageNet** consists of 100 classes and each class has 600 images. Following the disjoint class split from (Ravi & Larochelle, 2017), we split it into 64 classes for training, 16 for validation, and 20 for test. Similarly, ***tiered*ImageNet** is a larger dataset, consisting of 608 classes and each class has 768∼1300 images. Classes are split into 351 for training, 97 for validation, and 160 for test (Ren et al., 2018). All images in these two datasets are of resolution 84×84×3. **CIFAR-FS** contains 60,000 images of size 32×32×3 from 100 classes. We use the same class split as *mini*ImageNet.

Implementation details. We follow the "K-shot, M-way" episode training setting in (Finn et al., 2017) to do few-shot classification experiments to evaluate PLATINUM. We implement image classification experiments in 5-way, 1-shot (5-shot) settings. Concretely, we sample each task to contain 1 (5) data points in the support set $\mathcal{S}$, and 15 (15) data points in the query set $\mathcal{Q}$. For the unlabeled set $\mathcal{U}$, we sample 50 (50) data points for each classes. To select a subset for semi-supervision using SMI functions, we use a budget $B_{in} =$ 25 (25) for the inner loop, and a budget $B_{out} = 50$ (50) for the outer loop. Note that we perform a per-class selection to assign pseduo-labels using the SMI functions, which leads to a budget of 5 and 10 data points for the inner and outer loop respectively. For our experiments in Tab. 2, Tab. 3 and Tab. 4, we use a labeled set ratio $\rho = 0.01$, where ρ is the ratio of the number of data points in the labeled set to the number of data points in the unlabeled set. However, we also compare with a number of other ρ values (see Tab. 6). For our experiments with OOD classes in the unlabeled set (Tab. 3), we use 5 distractor classes with 50 data points for each class. To make a fair comparison, we apply the same 4-layer CONV backbone architecture given in (Vinyals et al., 2016; Finn et al., 2017) for our model and all baselines. We provide detailed hyperparameters for our experiments in Appendix C. We use an NVIDIA RTX A6000 GPU for our experiments. The PyTorch implementation is available at `https://tinyurl.com/2p9y3ejm`.

Baselines. We consider meta-learning based semi-supervised few shot classification approaches, and compare with two methods, namely the extended prototypical network (Ren et al., 2018) (including Soft k-Means, Soft k-Means+Cluster, Masked Soft k-Means) and TPN-semi (Liu et al., 2019). We also compare with MAML which serves as the supervised classification baseline without the additional unlabeled set. In addition, we compare with two well known approaches from the semi-supervised learning literature and implement them in the inner and outer loop on the top of MAML. The first one is Pseudo-labeling (PL) (Lee et al., 2013) and the second one is a consistency regularization method, VAT(Miyato et al., 2018).

4.2. Results

In this section, we present extensive experiments that compare the performance of PLATINUM with other methods. We provide the results for 1-shot (5-shot), 5-way experiments for *mini*ImageNet in Tab. 2, *tiered*ImageNet in Tab. 3, and CIFAR-FS in Tab. 4. On all datasets, we conduct experiments with (*w/*) and without (*w/o*) OOD classes in the unlabeled set. Since these experiments use $\rho = 0.01$, we conduct experiments for ρ =0.1, 0.2 and 0.3 on *mini*ImageNet, and

present the results in Tab. 6.

Analysis across multiple datasets. We observe that FLMI outperforms other methods for the 1-shot setting on all datasets. The performance of FLMI is slightly better than GCMI due to the additional diversity that the FLMI function models (see Tab. 1). When compared to other methods, the SMI functions (FLMI and GCMI) improve the accuracy by $\approx 2-4\%$ over existing methods. Interestingly, in Tab. 4, we observe that GCMI outperforms FLMI and other baselines in the presence of OOD classes in the unlabeled set. This is expected since GCMI *only* models query-relevance (Kothawade et al., 2021c) as opposed to FLMI which also models diversity.

Varying the labeled set ratio ρ. In Tab. 6, we analyze different values of ρ for 1-shot and 5-shot on the *mini*ImageNet dataset. We observe that the gain using PLATINUM is higher when the number of labeled data points is lower than the number of unlabeled data points, *i.e.*, ρ is small. This further reinforces the need for a framework like PLATINUM which performs well in the low labeled data regime.

4.3. Ablation Study

Varying the number of distractor classes. To explore the effect of the number of OOD classes in the unlabeled set, we increased the number of OOD classes, while keeping the number of in-distribution classes to be 5. We keep using the same number of unlabeled images per class as previous experiment. In Fig. 3, we present the result for this ablation study on the 5-way 5-shot setting for the *mini*ImageNet dataset. As expected, we observe that the accuracy during meta-testing decreases as the complexity of in-distribution subset selection increases due to larger number of OOD classes. We observe that the semi-supervision provided by the SMI based methods (FLMI and GCMI) consistently aids MAML and outperform other methods as the number of OOD classes increase, while PL suffers and eventually performs slightly worse than MAML.

Inner and outer loop selection. One of the key components of PLATINUM is embedding semi-supervision in the outer-loop. We conduct an ablation study using the 5-way 1-shot and 5-shot setting on the *mini*ImageNet dataset to analyze the effect of outer-loop semi-supervision and present the results in Fig. 4. Particularly, we evaluate the meta-test accuracy of few-shot classification with (*w/*) and without (*w/o*) the outer-loop selection for three methods: PL, GCMI and FLMI. MAML is also included for comparison. We observe that providing semi-supervision in the outer loop consistently improves the performance across all experiments. Especially for 1-shot of PL, we observe an improvement of $\approx 4\%$. Interestingly, PL performs worse than MAML without the outer-loop semi-supervision, and outperforms MAML with it.

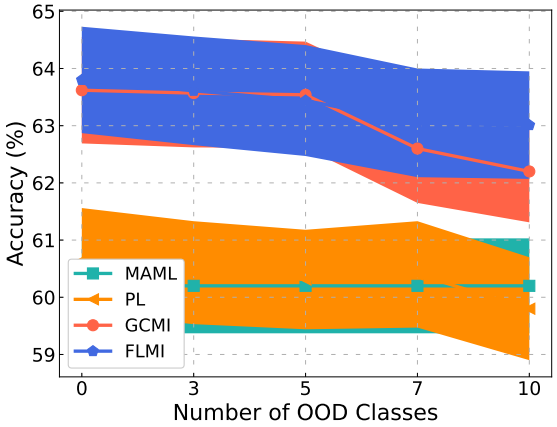

Figure 3: Comparison under different number of OOD classes in the Unlabeled Set for 5-shot case on *mini*ImageNet. TPN-semi is much worse than MAML by 20%, so we do not put it in this figure. (1-shot case is shown in Appendix C.)

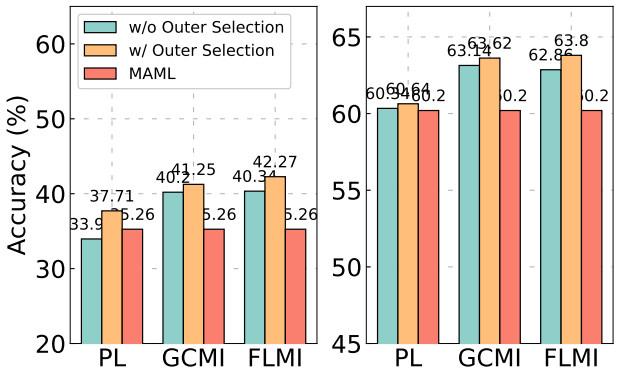

Figure 4: *w/* outer selection vs. *w/o* outer selection. **Left**: 1-shot, **Right**: 5-shot. Both of them are on *mini*ImageNet.

5. Conclusion

In this paper, we propose a novel semi-supervised model-agnostic meta-learning framework PLATINUM. It leverages submodular mutual information functions as per-class acquisition functions to select more balanced and diverse data from unlabeled data in the inner and outer loop of meta-learning. Meta-learning based semi-supervised few-shot learning experiments validates the effectiveness of embedding semi-supervision in the MAML by PLATINUM, especially for small ratio of labeled to unlabeled samples. We also notice that it might be useful to involve some diversity measurements for the selected subset to do quantitative analysis, we leave this as future work.

References

Antoniou, A., Edwards, H., and Storkey, A. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

Bach, F. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011.

Bach, F. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1):419–459, 2019.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2018.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Fallah, A., Mokhtari, A., and Ozdaglar, A. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092. PMLR, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Fujishige, S. *Submodular functions and optimization*. Elsevier, 2005.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Huang, K., Geng, J., Jiang, W., Deng, X., and Xu, Z. Pseudo-loss confidence metric for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8671–8680, October 2021.

Iyer, R. and Bilmes, J. A memoization framework for scaling submodular optimization to large scale problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2340–2349. PMLR, 2019.

Iyer, R., Khargoankar, N., Bilmes, J., and Asanani, H. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pp. 722–754. PMLR, 2021.

Kaushal, V., Iyer, R., Kothawade, S., Mahadev, R., Doctor, K., and Ramakrishnan, G. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1289–1299. IEEE, 2019a.

Kaushal, V., Subramanian, S., Kothawade, S., Iyer, R., and Ramakrishnan, G. A framework towards domain specific video summarization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 666–675. IEEE, 2019b.

Kaushal, V., Kothawade, S., Iyer, R., and Ramakrishnan, G. Realistic video summarization through visiocity: A new benchmark and evaluation framework. In *Proceedings of the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery*, pp. 37–44, 2020.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., De, A., and Iyer, R. Grad-match: A gradient matching based data subset selection for efficient learning. *In ICML*, 2021a.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. Glister: Generalization based data subset selection for efficient and robust learning. *In AAAI*, 2021b.

Kothawade, S., Girdhar, J., Lavania, C., and Iyer, R. Deep submodular networks for extractive data summarization. *arXiv preprint arXiv:2010.08593*, 2020.

Kothawade, S., Beck, N., Killamsetty, K., and Iyer, R. Similar: Submodular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems*, 34, 2021a.

Kothawade, S., Ghosh, S., Shekhar, S., Xiang, Y., and Iyer, R. Talisman: Targeted active learning for object detection with rare classes and slices using submodular mutual information. *arXiv preprint arXiv:2112.00166*, 2021b.

Kothawade, S., Kaushal, V., Ramakrishnan, G., Bilmes, J., and Iyer, R. Prism: A rich class of parameterized submodular information measures for guided subset selection. *arXiv preprint arXiv:2103.00128*, 2021c.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in LRtation learning, ICML*, volume 3, 2013.

Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S., and Schiele, B. Learning to self-train for semi-supervised few-shot classification. *Advances in Neural Information Processing Systems*, 32:10276–10286, 2019.

Liu, Y., Iyer, R., Kirchhoff, K., and Bilmes, J. Svitchboard-ii and fisver-i: Crafting high quality and low complexity conversational english speech corpora using submodular function optimization. *Computer Speech & Language*, 42:122–142, 2017.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. J., and Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. *International Conference on Learning Representations*, 2019.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Munkhdalai, T. and Yu, H. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563. PMLR, 2017.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. J. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv preprint arXiv:1804.09170*, 2018.

Oreshkin, B. N., Rodriguez, P., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

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

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016.

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

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

Tohidi, E., Amiri, R., Coutino, M., Gesbert, D., Leus, G., and Karbasi, A. Submodularity in action: From machine learning to signal processing applications. *IEEE Signal Processing Magazine*, 37(5):120–133, 2020.

Tschiatschek, S., Iyer, R. K., Wei, H., and Bilmes, J. A. Learning mixtures of submodular functions for image collection summarization. In *Advances in neural information processing systems*, pp. 1413–1421, 2014.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

Wang, Y., Xu, C., Liu, C., Zhang, L., and Fu, Y. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

Yu, Z., Chen, L., Cheng, Z., and Luo, J. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12856–12864, 2020.

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., and Song, Y. Metagan: An adversarial approach to few-shot learning. *NeurIPS*, 2:8, 2018.

A. Notation

For clear interpretation, we list the notations used in this paper and their corresponding explanation, as shown in Table 7.

Table 7: Important Notations and Descriptions

Notation	Description
$p(\mathcal{T})$	Probability distribution of meta-training tasks
N	The number of meta-training tasks
M-way, K-shot	The number of classes in one task M, and each class contains K examples in the support set
$\mathcal{T}_i$	The i-th meta-training task
$\{\mathcal{S}_i, \mathcal{Q}_i, \mathcal{U}_i\}$	Support set, query set, and unlabeled set of meta-training task $\mathcal{T}_i$
$\{\mathcal{S}^{new}, \mathcal{Q}^{new}, \mathcal{U}^{new}\}$	Support set, query set, and unlabeled set for task $\mathcal{T}_i'$ in meta-test
$\mathcal{A}_i^s$	Selected unlabeled examples from Unlabeled set for task $\mathcal{T}_i$ in the inner loop in meta-training
$\mathcal{A}_i^q$	Selected unlabeled examples from Unlabeled set for task $\mathcal{T}_i$ in the outer loop in meta-training
θ	Initial parameters of base learner
ϕ_i	Task-specific parameters for task $\mathcal{T}_i$
$L(\phi; \mathcal{D})$	Loss function on dataset $\mathcal{D}$ characterized by model parameter ϕ
L_l, L_u	Cross entropy loss on labeled data (or hypothesized labeled data)
$\text{Alg}(\theta; \mathcal{D})$	One or multiple steps of gradient descent initialized at θ on dataset $\mathcal{D}$
α, β	Learning rate in the inner loop and outer loop
τ_{in}, τ_{out}	Temperature annealing coefficient in the inner (or outer) loop
B_{in}, B_{out}	Budget in the inner (or outer) loop selection among all classes in the task
T_{in}, T_{out}, T_{warm}	Total number of steps in the inner loop; The number of epochs in the outer loop; Warm start epoch in the outer loop
f	A submodular function
S_{ij}	similarity between sample i and j
I_f	A submodular mutual information function instantiated using a submodular function f
$\mathcal{X}$	Pairwise similarity matrix used to instantiate a submodular function f

B. Details of Inner and Outer SMI Subset Selection

B.1. Inner loop

Task specific model parameters for task $\mathcal{T}_i$:

$$\begin{aligned}\phi_i =& \text{argmin}_\theta \, L_l\left(\theta; \mathcal{S}_i\right) + \tau_{\text{in}} L_u\left(\theta; \mathcal{A}_i^s\right) \\ =& \theta - \alpha \nabla_\theta L_l\left(\theta; \mathcal{S}_i\right) - \alpha \tau_{\text{in}} \nabla_\theta L_u\left(\theta; \mathcal{A}_i^s\right) \text{(one step gradient update example)}\end{aligned} \tag{13}$$

In which, α is learning rate. τ is the coefficient from the pseudo labeling approach. Since it is an increasing temperature variable, let $\tau^{(t)}$ denote the τ in step t.

Since there are several gradient update steps in inner loop.

$$\phi_i^{(t+1)} = \phi_i^{(t)} - \nabla L\left(\theta; \mathcal{S}_i \cup \mathcal{A}_i^s\right) \tag{14}$$

Let $\phi_i^{(t)}$ denote the model parameters for t-th step for task $\mathcal{T}_i$.

- initialization: $\phi_i^{(0)} = \theta$, $\mathcal{A}_i^s = \emptyset$
- inner step 1:

  $$\phi_i^{(1)} = \theta - \alpha \nabla_\theta L_l(\theta; \mathcal{S}_i) - \alpha \tau_{\text{in}}^{(1)} \nabla_\theta L_u\left(\theta; \mathcal{A}_i^s\right)$$

  Select subset for this step: $\mathcal{A}_{i1}^s \leftarrow \text{argmax}_{\mathcal{A}_{i1}^s \subseteq \mathcal{U}_i, |\mathcal{A}_{i1}^s| \leq B_{\text{in}}} I_f(\mathcal{A}_{i1}^s; \mathcal{S}_i \cup \mathcal{Q}_i)$

Set of selected examples: $\mathcal{A}_i^s = \mathcal{A}_i^s \cup \mathcal{A}_{i1}^s$

In this step, the CNN model used to calculate the class probabilities for SMI is parameterized by θ.

- inner step 2:

$$\phi_i^{(2)} = \theta - \alpha \nabla_{\phi^{(1)}} L_l(\phi^{(1)}; \mathcal{S}_i) - \alpha \tau_{\text{in}}^{(2)} \nabla_{\phi^{(1)}} L_u\left(\phi^{(1)}; \mathcal{A}_i^s\right)$$

Select subset for this step: $\mathcal{A}_{i2}^s \leftarrow \text{argmax}_{\mathcal{A}_{i2}^s \subseteq \mathcal{U}_i, |\mathcal{A}_{i2}^s| \leq B_{\text{in}}} I_f(\mathcal{A}_{i2}^s; \mathcal{S}_i \cup \mathcal{Q}_i)$

Set of selected examples: $\mathcal{A}_i^s = \mathcal{A}_i^s \cup \mathcal{A}_{i2}^s$

In this step, the CNN model used to calculate the class probabilities for SMI is parameterized by $\phi^{(1)}$.

- inner step 3:

$$\phi_i^{(3)} = \theta - \alpha \nabla_{\phi^{(2)}} L_l(\phi^{(2)}; \mathcal{S}_i) - \alpha \tau_{\text{in}}^{(3)} \nabla_{\phi^{(2)}} L_u\left(\phi^{(2)}; \mathcal{A}_i^s\right)$$

Select subset for this step: $\mathcal{A}_{i3}^s \leftarrow \text{argmax}_{\mathcal{A}_{i3}^s \subseteq \mathcal{U}_i, |\mathcal{A}_{i3}^s| \leq B_{\text{in}}} I_f(\mathcal{A}_{i3}^s; \mathcal{S}_i \cup \mathcal{Q}_i)$

Set of selected examples: $\mathcal{A}_i^s = \mathcal{A}_i^s \cup \mathcal{A}_{i3}^s$

In this step,the CNN model used to calculate the class probabilities for SMI is parameterized by $\phi^{(2)}$.

- continue repeat until the end of inner loop: step $T_{\text{in}} - 1$.

Model parameters for task T_i udpate process:

$$\phi^{(0)}(:= \theta) \rightarrow \phi^{(1)} \rightarrow \phi^{(2)} \rightarrow \phi^{(3)} ... \rightarrow \phi^{(T_{\text{in}}-1)}$$

B.2. Outer Loop

Meta-parameter update according to:

$$\theta = \underset{\theta}{\text{argmin}}\, \mathcal{J}(\theta) \tag{15}$$

The final loss function is:

$$\begin{aligned} \mathcal{J}(\theta) &= \frac{1}{M} \sum_{i=1}^{M} L_l\left(\phi_i; \mathcal{Q}_i\right) + \tau_{\text{out}} L_u\left(\phi_i; \mathcal{A}_i^q\right) \\ &= \frac{1}{M} \sum_{i=1}^{M} L_l\left(\text{argmin}_\theta\, L_l\left(\theta; \mathcal{S}_i\right) + \tau_{\text{in}} L_u\left(\theta; \mathcal{A}_i^s\right); \mathcal{Q}_i\right) + \tau_{\text{out}} L_u\left(\text{argmin}_\theta\, L_l\left(\theta; \mathcal{S}_i\right) + \tau_{\text{in}} L_u\left(\theta; \mathcal{A}_i^s\right); \mathcal{A}_i^q\right) \end{aligned} \tag{16}$$

in which, T is the set of meta-training tasks. τ_{out} is still a coefficient borrowed from the pseudo label formulation.

The second equal in the above equation is according to the inner loop update:

$$\phi_i = \text{argmin}_\theta\, L_l\left(\theta; \mathcal{S}_i\right) + \tau L_u\left(\theta; \mathcal{A}_i^s\right)$$

Since there is only one step in the outer loop, subset selection only happens one time.

$$\mathcal{A}_i^q \leftarrow \underset{\mathcal{A} \subseteq \mathcal{U}_i \setminus \mathcal{A}_i^s, |\mathcal{A}| \leq B_{\text{out}}}{\text{argmax}} I_f(\mathcal{A}; \mathcal{S}_i \cup \mathcal{Q}_i) \tag{17}$$

The CNN model used here should be the final model parameter after the inner loop: $\phi_i^{(T_{\text{in}}-1)}$.

The motivation here to use $\phi_i^{(T_{\text{in}}-1)}$ instead of meta-parameter θ is that SMI needs a CNN model which has powerful representation. For task $\mathcal{T}_i$, $\phi_i^{(T_{\text{in}}-1)}$ is more powerful than θ.

C. Additional Experimental Detail

As aforementioned, our backbone follows the same architecture as the embedding function used by (Finn et al., 2017). Concretely, the backbone structure is made of 4 modules, each of which contains a 3×3 convolutions and 64 filters, followed by batch normalization, a ReLU, and a 2×2 max-pooling with stride 2. To reduce overfitting, 32 filters per layer are considered. Cross entropy loss function is used for all experiment including the loss of selected unlabeled set between the predicted labels and the hypothesized labels.

C.1. Hyparameters tuning

All baseline approaches including three extended prototypical networks (Ren et al., 2018) and TPN-semi (Liu et al., 2019) are reimplemented via their official code following the original implementation including hyper-parameters. For our PLATINUM algorithm, all step sizes (α, β) are chosen from $\{0.0001, 0.001, 0.01, 0.1\}$. The Batch size (number of tasks per iteration) is chosen from $\{1, 2, 4\}$. The number of iterations are chosen from $\{10{,}000,\ 20{,}000,\ 30{,}000,\ 40{,}000,\ 60{,}000\}$. The selected best ones are: learning rate in the inner loop $\alpha = 0.01$, meta parameters step size (outer learning rate) $\beta = 0.0001$; the number of iterations for all experiments is set to be $60,000$ (600 epochs, each epoch has 100 iterations). We monitor the accuracy and loss from meta-validation stage and save the model which has the best validation accuracy and use that to evaluate the performance on unseen novel tasks in meta-test stage.

C.2. Additional Results

Table 8: Few-shot classification accuracies (%) of different labeled ratios on *mini*ImageNet.

Methods	*1-shot* 1%	5%	10%	20%	30%
Soft k-Means (Ren et al., 2018)	24.61±0.64	38.45±0.81	40.65±0.92	42.55±0.99	44.09±1.08
Soft k-Means+Cluster (Ren et al., 2018)	15.76±0.59	38.34±0.82	41.15±0.99	45.17±0.95	47.05±1.08
Masked Soft k-Means (Ren et al., 2018)	25.48±0.67	39.03±0.89	42.91±0.93	45.31±1.01	47.17±1.07
TPN-semi (Liu et al., 2019)	40.25±0.92	42.40±0.77	45.78±0.80	48.02±0.82	47.52±0.83
MAML (Finn et al., 2017)	35.26±0.85	42.51±0.78	44.29±0.78	45.10±0.75	45.26±0.78
PL	37.71±0.94	44.04±0.82	46.58±0.72	45.13±0.78	44.37±0.81
GCMI (ours)	41.94±0.96	44.98±0.80	46.85±0.74	47.72±0.76	48.93±0.70
FLMI (ours)	**42.27**±0.95	**45.01**±0.83	**47.84**±0.86	**47.82**±0.78	**48.98**±0.72

Table 9: Few-shot classification accuracies (%) of different labeled ratios on *mini*ImageNet.

Methods	*5-shot* 1%	10%	20%	30%
Soft k-Means (Ren et al., 2018)	38.20±1.64	56.27±1.75	60.13±1.79	62.47±1.65
Soft k-Means+Cluster (Ren et al., 2018)	33.65±1.53	56.87±1.77	60.33±1.81	62.43±1.79
Masked Soft k-Means (Ren et al., 2018)	39.33±1.55	57.20±1.64	62.50±1.78	63.00±1.77
TPN-semi (Liu et al., 2019)	46.27±1.67	60.55±0.72	62.43±0.69	63.10±0.69
MAML (Finn et al., 2017)	60.22±0.83	61.06±0.81	63.18±0.76	65.60±0.82
PL (Lee et al., 2013)	60.64±0.92	61.17±0.85	63.06±0.79	65.14±0.74
GCMI (ours)	63.62±0.95	**62.72**±0.88	64.78±0.76	65.96±0.74
FLMI (ours)	**63.80**±0.92	62.60±0.86	**65.16**±0.74	**66.10**±0.79

Selection accuracy. Fig. 6, Fig. 7, and Fig. 8 show the selection accuracy of three SSL algorithms: PL, GCMI and FLMI in the inner loop during meta-test. Although GCMI and FLMI has slightly low accuracy than PL, this verifies that our proposed PLATINUM is able to select more balanced and diverse data which are more important for model training.

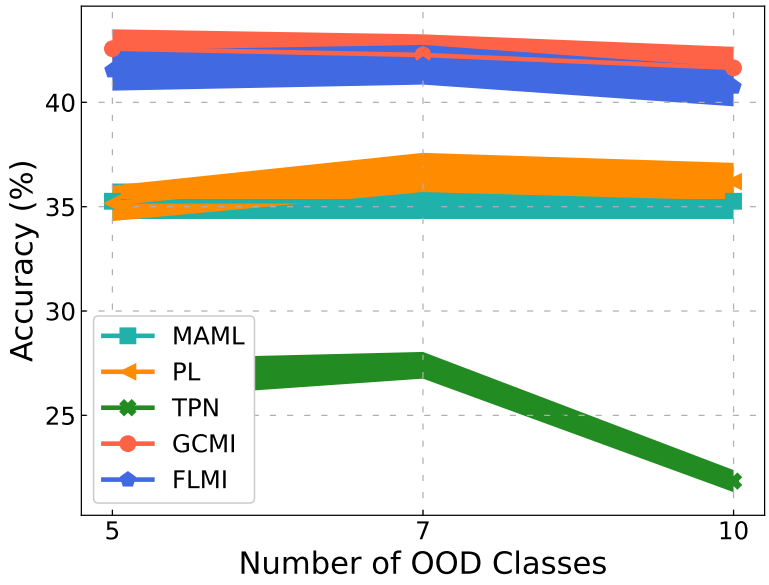

Figure 5: Comparison under different number of OOD classes in the unlabeled set for 1-shot case on *mini*ImageNet.

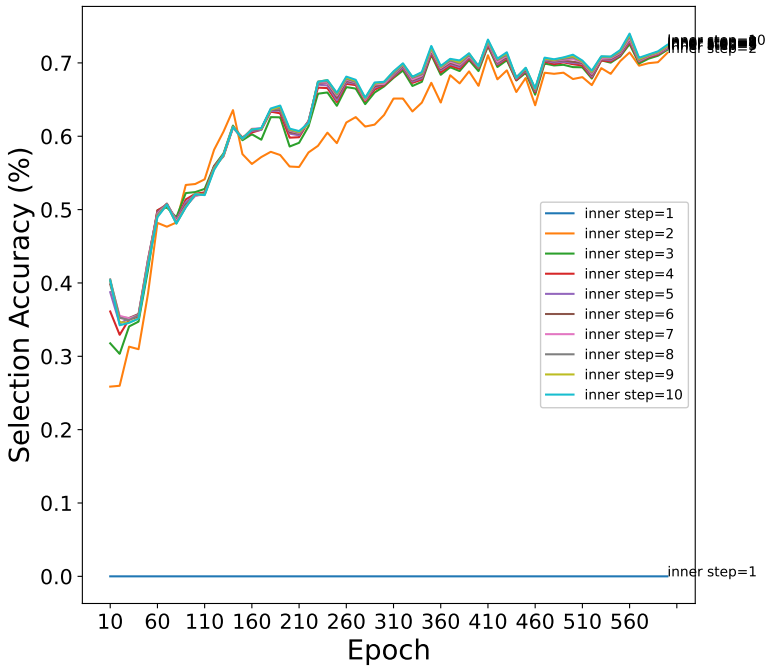

Figure 6: Selection accuracy in the unlabeled set for 1-shot case on *mini*ImageNet during meta-test for PL.

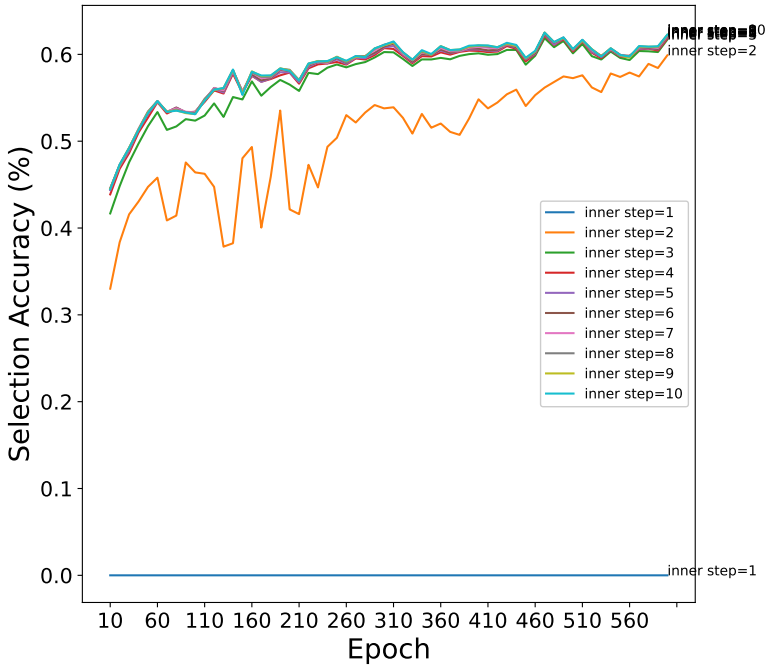

Figure 7: Selection accuracy in the unlabeled set for 1-shot case on *mini*ImageNet during meta-test for GCMI.

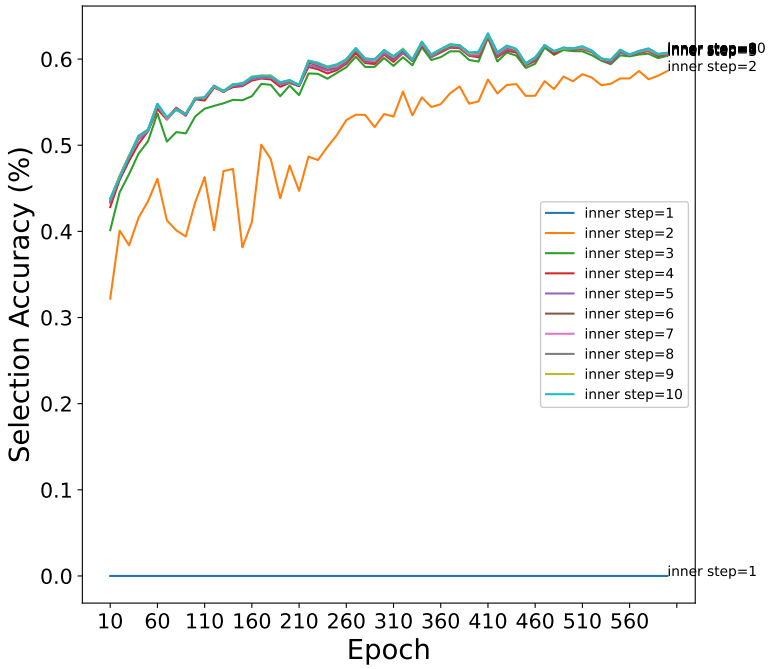

Figure 8: Selection accuracy in the unlabeled set for 1-shot case on *mini*ImageNet during meta-test for FLMI.