Submodular Meta Data Compiling for Meta Optimization

Fengguang Su, Yu Zhu, Ou Wu*, and Yingjun Deng

National Center for Applied Mathematics, Tianjin University, China
{fengguangsu,yuzhu,wuou,yingjun.deng}@tju.edu.cn

Abstract. The search for good hyper-parameters is crucial for various deep learning methods. In addition to the hyper-parameter tuning on validation data, meta-learning provides a promising manner for optimizing the hyper-parameters, referred to as meta optimization. In all existing meta optimization methods, the meta data set is directly given or constructed from training data based on simple selection criteria. This study investigates the automatic compiling of a high-quality meta set from training data with more well-designed criteria and the submodular optimization strategy. First, a theoretical analysis is conducted for the generalization gap of meta optimization with a general meta data compiling method. Illuminated by the theoretical analysis, four criteria are presented to reduce the gap's upper bound. Second, the four criteria are cooperated to construct an optimization problem for the automatic meta data selection from training data. The optimization problem is proven to be submodular, and the submodular optimization strategy is employed to optimize the selection process. An extensive experimental study is conducted, and results indicate that our compiled meta data can yield better or comparable performances than the data compiled with existing methods.

Keywords: Hyper-parameter optimization · Meta optimization · Generalization gap · Submodular optimization · Selection criteria

1 Introduction

Hyper-parameters have a considerable effect on the final performance of a model in machine learning. In shallow learning, cross-validation is usually leveraged to search (near) optimal hyper-parameters; in deep learning, due to the high time consumption of cross-validation, an independent validation set is constructed, and the hyper-parameters with the best performance are selected as the final hyper-parameters. In both strategies, the hyper-parameters are searched in a pre-defined grid. Recently, meta-learning has provided an effective manner to directly optimize the hyper-parameters instead of the grid search in existing strategies. Various hyper-parameters, such as learning rates [1], weights of noisy or imbalanced samples [3–5], pseudo labels [6–8], and others inside particular

* Corresponding author.

methods [9, 10], have been optimized via meta-learning on an additional small meta data set. Meta-learning based hyper-parameter optimization is called meta optimization.

In meta optimization, an independent meta data set is required, and ideally the meta set is unbiased. For example, in meta semantic data augmentation [10], which applies meta optimization for the covariance matrix, the meta data set in an experimental run contains a certain number of images independent of the training set. Although the leveraged meta set is claimed to be unbiased, no "unbiased" standard is provided. Most existing studies directly assume that an independent and high-quality meta set is ready for training. However, independent meta data do not usually exist. Recently, Zhang and Pfister [11] combine two criteria to compile meta data from training data with a simple greedy selection strategy. Initial promising results are reported in their study. However, their utilized criteria are still simple and may be insufficient in meta data compiling.

This study proposes a new effective method for compiling meta data only from the corresponding training data[1]. First, the generalization gap is analyzed for compiled meta data. Based on the upper bound of the gap, we analyze the characteristics that meta data should meet. Four selection criteria are then obtained: cleanness, balance, diversity, and uncertainty. The submodular optimization strategy [12] is leveraged to optimize the selection process with the criteria. Experiments on the two typical meta optimization scenarios, namely, imbalance learning and noisy label learning, are performed to verify the effectiveness of our method. The main contributions are summarized as follows:

- The expected generalization gap of the meta optimization is inferred when the ideal (i.e., not unbiased) meta set is not given, and the employed meta set is constructed through a meta data compiling method. This gap facilitates the understanding and explanation of the performances of meta optimization with different meta data compiling methods. Moreover, the gap provides theoretical guidance for automatic meta data construction.
- A new meta data compiling method is proposed to select meta data from training data for meta optimization. In our method, four sophisticated criteria are considered illuminated by the gap, and the submodular optimization strategy is introduced to solve the optimal subset selection with the fused criteria. Extensive experiments indicate our compiled meta data yield better accuracies in typical meta optimization scenarios than existing strategies.

2 Related Work

This section briefly introduces meta optimization, meta data compiling, and submodular optimization in machine learning.

[1] Although our method can be used to compile meta data directly from validation data, this study limits it to training data because validation data do not exist in many learning tasks.

2.1 Meta Optimization

Meta optimization is the instantiation of meta-learning [13, 14], which optimizes the target hyper-parameters by minimizing the learning error on meta data. Compared with the grid search, meta optimization is more efficient and has theoretical advantages over traditional cross-validation [15]. Let T and S be the training and (unbiased) meta sets, respectively. Let Θ and μ be the model parameters and hyper-parameters, respectively. Given μ, an optimal $\Theta^*(\mu)$ can be subsequently obtained as follows:

$$\Theta^*(\mu) = \arg\min_{\Theta} \mathcal{L}_T(\Theta, \mu), \tag{1}$$

where $\mathcal{L}$ is the loss. The optimal hyper-parameters μ^* can thus be obtained by minimizing the loss on the meta set S:

$$\mu^* = \arg\min_{\mu} \mathcal{L}_S(\Theta^*(\mu)). \tag{2}$$

Meta optimization has been widely used in various scenarios, such as imbalance learning and noisy label learning.

2.2 Meta Data Compiling

In existing studies, meta data are assumed to be given in advance, and no standard for selecting meta data is provided and discussed. Take meta optimization as an example in imbalance to illustrate how the meta data are compiled in nearly all existing studies. The benchmark data set CIFAR10 [16] contains 50,000 training samples on ten balanced categories. In the experiments, a balanced subset of 50,000 images is used as the independent meta set. Then, the rest of the images are used to build imbalanced training set by different category-wise probabilities.

Unfortunately, the above simulation process is infeasible in real applications. A promising solution is to define a set of "unbiased" criteria and then select meta data from training data. So far, only one recent study [11] has investigated this technical line. However, only the "cleanness" criterion and the "balance" criterion are considered. For the balance criterion in [11], if the number of samples for a certain class is not enough, the authors simply repeat the samples to attain balance. Theoretical guidance for how to compile meta data is still lacking up till now. This study attempts to construct guidance with a theoretical basis.

2.3 Submodular Optimization

Submodular optimization provides an efficient framework to solve the NP-hard combination problem with fast greedy optimization. A submodular optimization instance LtLG [17] can achieve linear time complexity in the data size, which is independent of the cardinality constraint in expectation. Submodular optimization has been widely used in text summarization, sensor placement, and speech recognition [18]. Joseph et al. [18] proposed an effective submodular

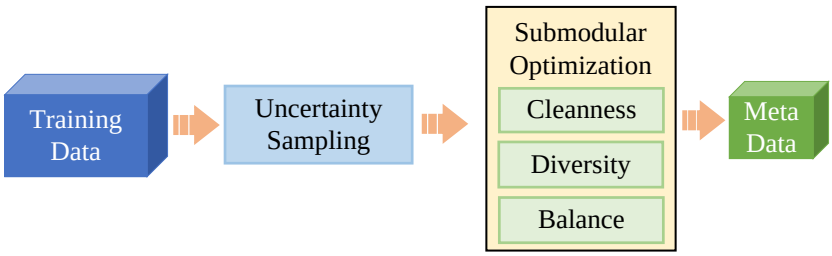

Fig. 1. Overview of our submodular meta data compiling. Our method is called submodular optimization-based meta data compiling (denoted as SOMC for briefly).

optimization-based method to construct a mini-batch in DNN training. Significant improvements in convergence and accuracy with submodular mini-batches have been observed.

When more sophisticated criteria are considered in automatic meta data compiling, the optimizing is very likely to become NP-hard, and simple greedy strategies are ineffective. Submodular optimization provides an effective solution.

3 Methodology

Fig. 1 illustrates the proposed submodular compiling process for the meta data set. The theoretical analysis for meta data construction is conducted firstly. And then the meta data selection method is described.

3.1 Theoretical analysis for meta data construction

Ideally, the distribution of samples in a compiled meta set equals that of testing samples. Bao et al. [15] infer a generalization gap for the meta optimization associated with independent ideal (i.e., unbiased) meta data. Let X be the sample space. Let p^{tr} and p^{me} be the distributions of training and meta data, respectively. Let T be a set of n training samples, and S_m^{me} be a set of m meta samples. Let $R(\mathcal{A}(T, S_m^{me}), p^{me}) = E_{x\sim p^{me}}[l(\mathcal{A}(T, S_m^{me}), x)]$ be the expected risk for the learning on the meta set S^{me}, where $l(\cdot, x)$ is the loss on x, $\mathcal{A}$ is a meta optimization method and $\mathcal{A}(T, S_m^{me})$ is the learned hyper-parameters and model with the training set T and meta set S_m^{me}. Let $\hat{R}(\mathcal{A}(T, S_m^{me}), S_m^{me}) = \frac{1}{m}\sum_{x\in S_m^{me}} l(\mathcal{A}(T, S_m^{me}), x)$ be the empirical risk for the learning on the meta set S_m^{me}.

The involved meta optimization method is assumed to be β-uniformly stable [15]. That is, for a randomized meta optimization algorithm $\mathcal{A}$, if for two arbitrary compiled meta sets S_m^{me} and $S_m^{'me}$ such that they differ in at most one sample, then $\forall\ T \in X^n, \forall\ x \in X$, we have

$$|E_{\mathcal{A}}[l(\mathcal{A}(T, S_m^{me}), x) - l(\mathcal{A}(T, S_m^{'me}), x)]| \leq \beta. \tag{3}$$

The generalization gap is defined as

$$gap(T, S_m^{me}) = R(\mathcal{A}(T, S_m^{me}), p^{me}) - \hat{R}(\mathcal{A}(T, S_m^{me}), S_m^{me}). \tag{4}$$

The expected generalization gap satisfies [15]

$$|E_{\mathcal{A},T,S_m^{me}}[gap(T, S_m^{me})]| \leq \beta. \tag{5}$$

We infer the expected generalization gap when a meta set is not ideal and constructed from training data, including ours method. As the involved meta optimization method is not changed in our study, the β-uniform stability is still assumed. Let S_m^{sme} be the compiled meta set consisting of m samples. Let P_m^{me} and P_m^{sme} be the distributions of S_m^{me} and S_m^{sme}, respectively.

Definition 1. *The distance between two distributions P_m^{me} and P_m^{sme} is defined as follow:*

$$d(P_m^{me} \| P_m^{sme}) = \int_{S \in X^m} |P_m^{me}(S) - P_m^{sme}(S)| dS. \tag{6}$$

To brevity, $d(P_m^{me} \| P_m^{sme})$ is denoted as d_m. If the two distributions are identical, then d_m is zero. Let $\hat{R}(\mathcal{A}(T, S_m^{sme}), S_m^{sme}) = \frac{1}{m} \sum_{x \in S_m^{sme}} l(\mathcal{A}(T, S_m^{sme}), x)$ be the empirical risk for the learning on our compiled meta set S_m^{sme}. We first define the generalization gap for S_m^{sme} as follows:

$$gap(T, S_m^{me}, S_m^{sme}) = R(\mathcal{A}(T, S_m^{me}), p^{me}) - \hat{R}(\mathcal{A}(T, S_m^{sme}), S_m^{sme}). \tag{7}$$

We obtain the theorem for the expectation of the above generalization gap as follows:

Theorem 1. *Suppose a randomized meta optimization algorithm $\mathcal{A}$ is β-uniformly stable on meta data in expectation, then we have*

$$|E_{\mathcal{A},T,S_m^{me},S_m^{sme}}[gap(T, S_m^{me}, S_m^{sme})]| \leq \beta + bd_m, \tag{8}$$

where b is the upper bound of the losses of samples in the whole space (following the assumption in [15]), and $d_m = d(P_m^{me} \| P_m^{sme})$.

Compared with the expected generalization gap for independent (ideal) meta sets given in Eq. (5), our expected generalization gap for automatically compiled meta sets contains an additional term bd_m. Naturally, an ideal criterion for S_m^{sme} should make sure both β and bd_m as small as possible. Note that $\beta = \frac{2cL^2}{m}[\frac{1}{\kappa}((\frac{Ns(l)}{2cL^2})^\kappa - 1) + 1]$ (Theorem 2 in [15]), where m, c, L, γ, κ, and N remain unchanged and only $s(l) = b - a$ (the range of the loss) may change in terms of different meta data selection criteria. As $a \to 0$ when the cross-entropy loss is used, only b and d_m affect the upper bound of the gap (i.e., the value of the right-side of (8)). Consequently, we explore the selection criteria according to the minimization of both d_m and b, separately[2]. First, we have the following conclusion.

[2] The value of b affects both β and bd_m, while d_m only affects bd_m.

Corollary 1. *The optimal selected meta data distribution* $P_m^{sme}(S)$ *should satisfy that* $d_m = 0$*, i.e.,* $P_m^{sme}(S) = P_m^{me}(S)$*.*

Accordingly, it is inappropriate to select training data uniformly at random as meta data as the training data in many scenarios (e.g., imbalance learning) is biased against the true meta data. In practice, the true distribution of meta data is unknown. However, two requirements [3–6, 8–10] are usually assumed to be met for an arbitrary meta data set:

- **Cleanness**. As meta data are assumed to be drawn from the true distribution without observation noises, meta data should be as clean as possible.
- **Balance**. The balance over categories is usually taken as a prior in previous studies utilizing meta optimization. This study also inherits this assumption.

According to Corollary 1, to reduce the value of d_m, cleanness and balance should be leveraged as two selection criteria in our meta data compiling. We will show that cleanness and balance may also reduce the value of b in the succeeding discussion.

As d_m cannot be guaranteed to be zero only with the two criteria mentioned above, b should also be as small as possible. The value of b is determined by both the ideal yet unknown meta set (actually the true distribution of meta data) and the compiled meta set (actually the underlying distribution of our compiled meta data). Considering that the ideal meta set is not given and our selection criteria do not affect the distribution of the ideal meta set, the ideal meta set can be ignored in the discussion for the reduction of b. To reduce the value of b, the following selection criteria are beneficial:

- **Cleanness**. If there are noisy samples in the compiled meta set, then the losses of clean samples will be larger as noisy samples usually damage the generalization ability [38]. Therefore, keeping the compiled meta data as clean as possible will also reduce the value of b in a high probability.
- **Balance**. Even though the balance prior does not hold in a specific learning task, the balance over categories may reduce the maximum loss of the samples of tail categories [5]. For this consideration, balance is still useful.
- **Uncertainty**. Pagliardini et al. [37] show that adding more samples with high uncertainty will increase the classification margin. Accordingly, the maximum loss may also be reduced if the meta data are noisy-free. Indeed, uncertainty sampling [30–32] is prevalent in sample selection in active learning. It is proven to be more data-efficient than random sampling [34].
- **Diversity**. Diversity can be seen as the balance prior for the samples within a category. This balance prior may also reduce the maximum loss of each category. The maximum loss may subsequently be reduced. Indeed, diversity-aware selection has other merits. Madan et al. [36] find that using the same amount of training data, increasing the number of in-distribution combinations (i.e., data diversity) also significantly improves the generalization ability to out-of-distribution data.

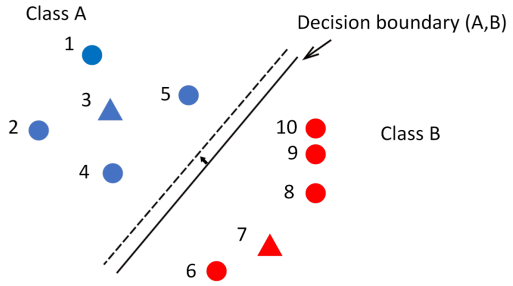

Fig. 2. An illustrative example of the four selection criteria. There are two classes and five samples per class with a decision boundary. The samples $\{3, 7\}$ are those with noisy labels. The cleanness criterion prefers the samples $\{4, 6\}$ to $\{3, 7\}$. The balance criterion promotes to select the samples $\{4, 5, 6, 10\}$ instead of $\{2, 4, 5, 6\}$. Diversity prefers samples $\{6, 8, 10\}$ to $\{8, 9, 10\}$. The uncertainty criterion promotes the selection of samples $\{4, 5\}$ instead of $\{1, 2\}$.

According to the above considerations, two more criteria, namely, uncertainty and diversity[3], are also considered in addition to the cleanness and balance criteria. Fig. 2 illustrates the roles of each of the four summarized selection criteria in terms of the reduction of b. There are two classes of points with a decision boundary between them. There are two noisy samples, 3 and 7. First, if cleanness is not considered, then the noisy samples $\{3, 7\}$ may appear in the meta set. It is highly possible that the losses of clean samples near $\{3\}$ and $\{7\}$ in the whole space are relatively high. Second, if the balance criterion is not considered, the losses of the samples in the tail categories are high to a certain extent. For example, if we choose the samples $\{1, 2, 4, 5, 6\}$ as meta data, then the losses of the samples near $\{9, 10\}$ in Class B will become high with a high probability. Third, if the diversity criterion is not considered, then the samples $\{8, 9, 10\}$ may be chosen. The samples around the sample $\{6\}$ may have higher loss values. Finally, if the uncertainty criterion is not considered (e.g., if $\{4, 5\}$ are not selected, the decision boundary will move in the direction of the dotted line.), then the classification margin will decrease [37]. Consequently, the losses of the samples near $\{4, 5\}$ will increase. Further, uncertainty can avoid selecting too many clean samples with small losses through Eq. (10), and thus can improve the update efficiency of meta optimization. Based on the above analysis, if any of the four summarized criteria are ignored, the losses of samples in specific local regions of the whole space will increase. As a result, b will increase.

3.2 Details of The Four Selection Criteria

This subsection describes how the four selection criteria are applied in the meta data compiling from a given training set. Considering that the training sizes in deep learning tasks are usually large, it is inappropriate to run all four selection

[3] Indeed, Ren et al. [29] revealed that the uncertainty and the diversity criteria are usually used together to improve the model performance in deep active learning.

criteria on each training sample. Therefore, sampling will firstly be performed to reduce the size of candidate samples fed to other criteria.

Uncertainty Criterion. Let Θ be the current model parameter. The output entropy of a training sample x_i is used to measure the uncertainty of x_i. Let C be the set of all classes. The calculation for the out entropy of x_i is as follows:

$$u(x_i) = -\sum\nolimits_{c \in C} P(c|x_i, \Theta) \log P(c|x_i, \Theta), \tag{9}$$

where $P(c|x_i, \Theta)$ refers to the probability that the current model predicts the sample x_i as the c-th category. Generally, a sample near the decision boundary has a high uncertainty score. In our implementation, we sample the data based on the normalized uncertainty score for each class, respectively. That is, the sampling probability of x_i is $u(x_i)/\sum_{j:y_j=y_i} u(x_j)$. More details about uncertainty sampling can be found in Algorithm 1 and the experimental implementation details in supplementary materials.

Cleanness Criterion. This criterion aims to select data with clean labels or clean features. Many metrics can be used to judge the noisy degree of a sample, including loss (prediction) [11], loss variance [2], gradient norm [20], etc. Considering that the loss metric is the most widely used, this study also adopts it. The cleanness degree of a set is defined as follows:

$$\mathcal{C}(S) = \sum\nolimits_{x_i \in S} c(x_i) = \sum\nolimits_{x_i \in S} P(y_i|x_i, \Theta), \tag{10}$$

where y_i is the label of x_i, and Θ is the model parameter(s). If y_i is a noisy label or x_i has non-trivial noisy features, then $P(y_i|x_i, \Theta)$ is usually small during training.

Balance Criterion. Imbalance can cause the model to have a good performance on the head categories but poor performance on the tail ones. Let n_c^s be the number of meta samples of the c-th category, and m be the total number of meta samples. The balance score of a subset is formulated as follows:

$$\mathcal{B}(S) = \prod\nolimits_{c \in C} I(\lfloor \frac{m}{|C|} \rfloor \leq n_c^s \leq \lceil \frac{m}{|C|} \rceil), \tag{11}$$

where C is the category set. When $\mathcal{B}(S) = 1$, the subset is balanced.

Diversity Criterion. The criterion selects samples with different features by considering the relationship among samples. The following approach is utilized to measure the diversity of a subset. Given $\phi(\cdot, \cdot)$ to be any distance metric between the two data points, a larger value of the minimum distance among points would imply more diversity in the subset.

$$\mathcal{D}(S) = \sum_{x_i \in S} \min_{x_j \in S: i \neq j} \phi(\tilde{x}_i, \tilde{x}_j), \tag{12}$$

where $\tilde{x}_i$ is the output of the final feature encoding layer of x_i. This score is dependent on the choice of distance metric. In our implementation, Euclidean distance ($\|\tilde{x}_i - \tilde{x}_j\|_2$) is employed according to the performances of different distance metrics reported in [18].

3.3 Submodular Optimization

The four criteria are cooperated to construct an optimization problem for the final meta data compiling. As previously described, the uncertainty criterion is first utilized to reduce the candidate training data. The diversity and cleanness criteria are then combined as follows:

$$\mathcal{F}(S) = \lambda\mathcal{D}(S) + (1-\lambda)\mathcal{C}(S), \tag{13}$$

where λ is a hyper-parameter. Let T be the candidate training data which is passed through the uncertainty criterion. Consequently, an optimal meta set of size m is selected by solving the following optimization problem:

$$\begin{aligned} S^* &= \arg\max_{S\subseteq T} \mathcal{F}(S) \\ s.t. \quad & |S| \leq m; \quad \mathcal{B}(S) = 1 \end{aligned}. \tag{14}$$

The maximization of Eq. (14) is a NP-hard problem as the total diversity score in Eq. (12) cannot be factorized into the sum of diversity scores of each sample. The simple greedy method leveraged in [11] is inapplicable. Hence, to conduct an efficient and effective maximization, the submodular optimization manner is leveraged.

Submodular optimization guarantees a solution for a submodular objective function which is at least (in the worst case) $1-1/e$ of the optimal solution [21], where e is the base of the natural logarithm. Further, some fast submodular optimization algorithms such as LtLG [17] have been put forward. An optimization problem can be solved with submodular optimization if its objective function is submodular and monotonically non-decreasing. Therefore, to apply submodular optimization, we have two lemmas for the objective function.

Definition 2. *Let X be a finite set. A set function $\mathcal{F}(S): 2^X \longrightarrow \mathbb{R}$ is submodular if $\forall A, B \subset X$ with $A \subset B$ and an element $a \in X \setminus B$, we have*

$$\mathcal{F}(\{a\} \cup A) - \mathcal{F}(A) \geq \mathcal{F}(\{a\} \cup B) - \mathcal{F}(B).$$

Definition 2 indicates that the gain diminishes as we add elements [21].

Lemma 1. *$\mathcal{F}(S)$ in Eq. (13) is submodular.*

Lemma 2. *$\mathcal{F}(S)$ in Eq. (13) is monotonically non-decreasing.*

According to Lemmas 1 and 2, the submodular optimization technique can be used directly to solve Eq. (14). Inspired by the general submodular optimization framework SMDL [18], our method consists of three main processes shown in Algorithm 1. First, a training subset T is obtained based on uncertainty sampling and is randomly partitioned into K disjoint subsets. Secondly, a subset is further generated from each of the K subset by maximizing the marginal gain $\mathcal{F}(a|S) = \mathcal{F}(\{a\}\cup S) - \mathcal{F}(S)$. Lastly, the subsets are merged to generate the final meta data set by considering the margin gain maximization and the balance constraint.

Algorithm 1 SOMC

Input: Training set T, $u(x_i)$, $i = 1, \cdots, |T|$, m, K, λ, and $\mathcal{F}(\cdot)$ in Eq. (13).
Output: Meta data set S

$S \longleftarrow \emptyset$;
Obtain a subset (still marked as T) of size $\frac{|T|}{2}$ based on uncertainty sampling;
Partition T into K disjoint sets $T_1, T_2, ..., T_K$;
Generate a subset S_k (m samples) from T_k using LtLG [17], $k = 1, \cdots, K$;
$\tilde{S} \longleftarrow \bigcup_{k=1}^{K} S_k$;
While $|S| < m$
    Select a sample $(x^*, y^*) \in \tilde{S} \setminus S$ using LtLG;
    If $n_{y^*} \leq \lceil \frac{m}{|C|} \rceil - 1$
        $S \longleftarrow \{(x^*, y^*)\} \cup S$;
Return S.

The time complexity of our proposed submodular optimization-based meta data compiling (SOMC) is $O((|T| + Km)md)$, where d is the feature dimension. In practice, Algorithm 1 can be implemented in parallel and the time complexity becomes approximately $O((|T|/K + Km)md)$. When m is large, the time consumption can be significantly reduced by first compiling a batch of small meta sets and then merging them as the final meta set S. The entire algorithmic steps and more details are presented in the supplementary material.

4 Experiments

This section evaluates the performance of SOMC in benchmark image classification corpora, including CIFAR [16], ImageNet-LT [22], iNaturelist [23], and Clothing1M [24]. Details of these corpora and the source code are provided in the supplementary material.

4.1 Evaluation on CIFAR10 and CIFAR100

Nearly all existing meta optimization studies utilize independent meta sets, and thus they should be compared. In this part, the independent meta data used in existing studies are replaced by the data compiled by our SOMC. In addition, the only existing automatic meta data selection method FSR [11] is also compared. FSR only uses cleanness and balance to select meta data in the training set. Indeed, FSR also contains multiple data augmentation tricks and a novel meta optimization method. For a fair comparison, only the module of meta data compiling of FSR (denoted as "FSRC") is compared in this experiment.

Results on Imbalance Classification Following [25], we use CIFAR10 and CIFAR100 to build imbalance training sets by varying imbalance factors $\mu \in \{200, 100, 50, 20, 10\}$, namely, CIFAR10-LT and CIFAR100-LT. The original balanced test sets are still used. The concrete hyper-parameters setting is described

Table 1. Test top-1 accuracy (%) of ResNet-32 on CIFAR10-LT and CIFAR100-LT under different imbalance settings.

Data set	CIFAR10-LT					CIFAR100-LT				
Imbalance factor	200	100	50	20	10	200	100	50	20	10
Base model (CE)	65.87	70.14	74.94	82.44	86.18	34.70	38.46	44.02	51.06	55.73
MCW+**100/1000 meta images** (CE)	70.66	76.41	80.51	86.46	**88.85**	39.31	43.35	48.53	55.62	59.58
MCW+**FSRC** (CE)	72.34	77.65	81.31	86.25	88.02	38.53	44.21	49.72	55.98	60.17
MCW+**SOMC** (CE)	**73.71**	**79.24**	**82.34**	**86.98**	88.67	**39.95**	**45.97**	**51.28**	**57.32**	**61.11**
MetaSAug +**100/1000 meta images** (CE)	76.16	**80.48**	83.52	87.20	88.89	42.27	46.97	51.98	57.75	61.75
MetaSAug+**FSRC** (CE)	75.41	79.28	82.87	86.81	88.37	42.53	47.02	51.61	57.87	61.35
MetaSAug+**SOMC** (CE)	**76.25**	80.25	**83.61**	**87.43**	**89.02**	**43.32**	**48.03**	**52.36**	**58.52**	**61.88**
MCW+**100/1000 meta images** (FL)	74.43	78.90	82.88	86.10	88.37	39.34	44.70	50.08	55.73	59.59
MCW+**FSRC** (FL)	74.57	79.23	83.06	86.22	88.59	39.67	44.85	50.35	55.89	59.87
MCW+**SOMC** (FL)	**75.26**	**80.17**	**83.65**	**86.52**	**88.84**	**40.26**	**45.96**	**51.13**	**56.67**	**60.35**
MetaSAug+**100/1000 meta images** (FL)	75.73	80.25	83.04	**86.95**	88.61	40.42	45.95	51.57	57.65	61.17
MetaSAug+**FSRC** (FL)	75.12	79.87	82.52	85.99	88.21	39.77	45.86	51.22	57.25	60.84
MetaSAug+**SOMC** (FL)	**76.01**	**80.44**	**83.41**	86.77	**88.87**	**40.69**	**46.90**	**51.99**	**57.81**	**61.65**
MCW+**100/1000 meta images** (LDAM)	77.23	80.00	82.23	84.37	87.40	39.53	44.08	49.16	52.38	58.00
MCW+**FSRC** (LDAM)	76.85	79.97	82.04	85.12	88.03	40.25	44.83	49.79	53.34	59.46
MCW+**SOMC** (LDAM)	**77.69**	**80.43**	**82.86**	**85.74**	**88.51**	**41.37**	**45.73**	**50.62**	**54.29**	**60.30**
MetaSAug+**100/1000 meta images** (LDAM)	76.42	80.43	83.72	87.32	**88.77**	42.87	**48.29**	52.18	57.65	61.37
MetaSAug+**FSRC** (LDAM)	75.89	79.93	83.21	86.72	87.93	42.69	47.43	51.65	57.54	61.35
MetaSAug+**SOMC** (LDAM)	**76.56**	**80.61**	**83.96**	**87.45**	88.57	**43.48**	48.17	**52.56**	**58.43**	**61.93**

Table 2. Test top-1 accuracy (%) on CIFAR10 and CIFAR100 of WRN-28-10 with varying noise rates under uniform noise.

Data set	CIFAR10			CIFAR100		
Noise rate	0%	40%	60%	0%	40%	60%
Base model (CE)	95.60±0.22	68.07±1.23	53.12±3.03	79.95±1.26	51.11±0.42	30.92±0.33
MSLC+**1000 meta images**	95.42±0.07	**91.54±0.15**	**87.27±0.27**	80.75±0.11	**71.83±0.24**	**65.37±0.53**
MSLC+**FSRC**	95.23±0.17	88.15±0.31	81.84±0.33	80.49±0.23	67.86±0.14	59.63±0.42
MSLC+**SOMC**	**95.65±0.05**	89.38±0.13	83.56±0.27	**81.36±0.31**	68.75±0.29	61.03±0.17
MWNet+**1000 meta images**	94.52±0.25	89.27±0.28	84.07±0.33	78.76±0.24	67.73±0.26	58.75±0.11
MWNet+**FSRC**	95.03±0.23	88.78±0.16	84.26±0.17	79.95±0.08	67.88±0.25	59.37±0.28
MWNet+**SOMC**	**95.69±0.09**	**89.81±0.13**	**85.16±0.12**	**80.68±0.32**	**68.63±0.14**	**60.65±0.19**

in the supplementary material. The average accuracy of the three repeated runs is recorded for each method. The meta set for all existing studies contains ten images for each category. However, the numbers of images in some tail categories in CIFAR10-LT and CIFAR100-LT are less than ten. Thus, data augmentation techniques are used to generate more candidates for the successive meta image selection for these categories for our SOMC and FSRC. ResNet-32 [26] is used as the base network. The parameter λ of our SOMC is searched in $\{0.3, 0.5, 0.7\}$, and K is searched in $\{2, 5\}$. More details are described in the supplementary file.

Two meta optimization methods, namely, MCW [5] and MetaSAug [10], are leveraged. Partial results on the early representative method MWNet [4] are shown in the supplementary file. The original study of both methods provides source codes and meta sets on the above data sets. Our experimental results are obtained directly on these codes and hyper-parameter settings.

The classification accuracies of the three meta optimization methods with independent meta sets, FSRC, and our proposed SOMC on CIFAR10-LT and CIFAR100-LT are shown in Table 1. The results are organized into three distinct groups according to the adopted loss functions (i.e., Cross-entropy (CE), Focal

Table 3. Test top-1 accuracy (%) of ResNet-32 on CIFAR10 and CIFAR100 with varying noise rates under flip noise.

Data set	CIFAR10			CIFAR100		
Noise rate	0%	20%	40%	0%	20%	40%
Base model (CE)	**92.89±0.32**	76.83±2.30	70.77±2.31	70.50±0.12	50.86±0.27	43.01±1.16
MSLC+**1000 meta images**	92.75±0.15	**91.67±0.19**	**90.23±0.13**	70.37±0.31	**67.59±0.06**	**65.02±0.21**
MSLC+**FSRC**	92.46±0.13	89.78±0.32	88.61±0.27	70.29±0.21	64.97±0.19	61.15±0.46
MSLC+**SOMC**	92.83±0.09	91.13±0.21	89.55±0.25	**70.82±0.15**	66.33±0.11	62.58±0.28
MWNet+**1000 meta images**	92.04±0.15	90.33±0.61	87.54±0.23	70.11±0.33	64.22±0.28	58.64±0.47
MWNet+**FSRC**	92.42±0.12	90.65±0.36	87.25±0.41	70.52±0.11	65.26±0.12	59.47±0.22
MWNet+**SOMC**	**93.06±0.06**	**91.37±0.11**	**88.65±0.26**	**71.39±0.31**	**66.69±0.11**	**60.34±0.19**

loss (FL), and LDAM). SOMC can construct more effective meta data only from training data than both independent meta sets and FSRC in nearly all the cases.

Results on Noisy Labels Learning Two typical types of corrupted training labels are constructed: 1) **Uniform noise.** The label of each sample is independently changed to a random class with probability p. 2) **Flip noise.** The label of each sample is independently flipped to similar classes with total probability p. Details are described in the supplementary file. Two typical meta optimization methods, MSLC [6] and MWNet [4], are used. In previous studies, the meta data for these two methods consist of absolutely clean images. These clean images will be replaced by the compiled images with our SOMC. ResNet-32 [26] and WRN-28-10 [27] are used as the base network. The hyper-parameters setting is presented in the supplementary file.

The classification results under different noise rates are shown in Tables 2 and 3. Our method outperforms the independent meta data in MWNet. As the noise ratio increases, SOMC degrades more than the independent meta data in MSLC. It is reasonable to require independent clean meta data in the case of a high noise rate. Our method SOMC consistently outperforms FSRC under different noise rates on both sets.

4.2 Evaluation of Large Data Sets

Four large data sets, iNaturalist2017 (iNat2017), iNaturalist2018 (iNat2018), ImageNet-LT, and Clothing1M are used. The former three are leveraged for imbalance learning, while the last is for noisy label learning. MCW and MetaSAug are utilized for ImageNet-LT, INat 2017 and 2018. MSLC and MWNet are utilized for Clothing1M. The experimental settings, including the hyper-parameters, are presented in the supplementary material.

Tables 4 and 5 show the results of the competing methods on iNaturalist data sets and ImageNet-LT. Although 25445 (for iNat2017), 16284 (for iNat2018), and 10000 (for ImageNet-LT) independent meta data are used for MCW and MetaSAug, their performances are worse than those of meta data compiled by our SOMC. FSRC yields the lowest accuracies among the three meta data construction methods for iNat2017 and 2018. In addition, MetaSAug+SOMC with the pre-trained BBN [28] yields the highest top-1 accuracy for iNat2017 and 2018. For ImageNet-LT, SOMC still yields the best results.

Table 6 shows the results on Clothing1M. It can be seen that SOMC achieves better results than 7000 independent meta data and compiled meta data by FSRC on MWNet. For MSLC, compared with 7000 independent meta data, SOMC still achieves comparable results. However, FSRC yields the worst results.

Table 4. Test top-1 accuracy (%) on iNaturalist 2017 and 2018.

Method	iNat2017	iNat2018
Base model (CE)	56.79	65.76
MCW+**25445/16284 meta images**	59.38	67.55
MCW+**FSRC**	58.76	67.52
MCW+**SOMC**	**60.47**	**68.89**
MetaSAug+**25445/16284 meta images**	63.28	68.75
MetaSAug+**FSRC**	62.59	68.28
MetaSAug+**SOMC**	63.53	69.05
MetaSAug+**SOMC** with BBN model	**65.34**	**70.66**

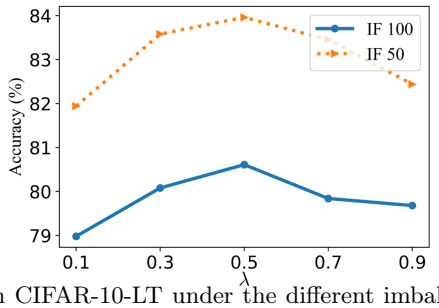

Fig. 3. Effect of λ on CIFAR-10-LT under the different imbalance factors (IF) based on MetaSAug+**SOMC** (LDAM).

4.3 Discussion

The supplementary file provides more details (including results and analysis) on the issues discussed in this part.

Table 5. Test top-1 accuracy (%) on ImageNet-LT.

Method	ImageNet-LT
Base model (CE)	38.88
MCW+**10000 meta images**	44.92
MCW+**FSRC**	45.05
MCW+**SOMC**	**45.97**
MetaSAug+**10000 meta images**	46.21
MetaSAug+**FSRC**	45.77
MetaSAug+**SOMC**	**46.68**

Table 6. Test top-1 accuracy (%) on on Clothing1M.

Method	Clothing1M
Base model (CE)	68.94
MWNet+**7000 meta images**	73.72
MWNet+**FSRC**	73.01
MWNet+**SOMC**	**73.89**
MSLC+**7000 meta images**	**74.02**
MSLC+**FSRC**	73.23
MSLC+**SOMC**	73.67

The above comparisons suggest that the meta data compiled by our SOMC are more effective than the independent meta data in most cases (except the cases of high noise rate when MSLC is used) and those compiled by FSRC in nearly all cases. This conclusion can be explained by Theorem 1. First, let P^{ime} and P^{fme} be the distributions of independent meta data and the FSCR meta data, respectively. It is very likely that $d(P^{me}||P^{sme})$ (i.e., d_m) is smaller than both $d(P^{me}||P^{ime})$ and $d(P^{me}||P^{fme})$ because our selection criteria are more elaborately designed. Second, we calculate the upper bounds of the test losses of the models corresponding to the three meta data compiling methods, namely, FSRC, IDMD (independent meta data), and our method SOMC, respectively. Fig. 4 shows the recorded values. SOMC does achieve the minimum upper bound of test losses (i.e., b) among the three methods. This is consistent with the theoretical analysis in Section 3.1 that the four criteria mainly aim to reduce d_m and b. More comparisons of the upper bounds of test losses are presented in the supplementary file.

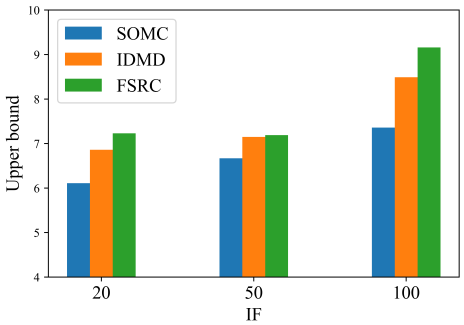

Fig. 4. The upper bounds of test losses on CIFAR10-LT for the three meta data compiling methods under different imbalance factors (IF) based on MetaSAug (LDAM).

There are two important hyper-parameters, namely, λ and K, in SOMC. They are tuned with grid search in the experiments. Nevertheless, the performances are usually satisfactory when $\lambda \in \{0.3, 0.5\}$ (shown in Fig. 3) and $K = 5$. In all the experiments, the parameter m in our SOMC equals the size of independent meta data used in existing studies for a fair comparison. In addition, the time cost of SOMC is recorded.

An ablation study is conducted for the importance of each criterion in SOMC. The results on imbalance learning (ResNet-32) are shown in Table 7. Removing each criterion causes a performance drop. This result indicates that each of the four criteria is useful in SOMC.

Table 7. Ablation study of MetaSAug+**SOMC** using CE loss on CIFAR-100-LT.

Imbalance factor	200	100	50
SOMC w/o Uncertainty	42.19	47.14	51.29
SOMC w/o Diversity	41.21	46.23	50.21
SOMC w/o Cleanness	41.37	46.42	50.13
SOMC w/o Balance	40.09	45.59	49.52
SOMC	**43.32**	**48.03**	**52.36**

We use different backbone networks (i.e., ResNet-50, ResNet-101, and ResNet-152 [26]). The results indicate that our method still achieves competitive performances. Comparisons with more competing methods and settings are conducted in the supplementary file.

5 Conclusions

This study has investigated the automatic compiling of meta data from training data for meta optimization. A theoretical analysis is firstly conducted for the generalization gap for automatic meta data compiling methods, and theoretical guidance for the construction of meta data is obtained. Four sophisticated selection criteria, namely, cleanness, balance, diversity, and uncertainty, are summarized to reduce the upper bound of the generalization gap. These criteria are cooperated to construct an objective function for optimal subset selection from training data. The submodular optimization technique is leveraged to search for the optimal subset. Extensive experiments on six benchmark data sets verify the effectiveness and competitive performance of the proposed method compared with SOTA competing methods.

References

1. Saxena, S., Vyas, N., DeCoste, D.: Training with data dependent dynamic learning rates. arXiv preprint arXiv:2105.13464 (2021)
2. Shin, W., Ha, J. W., Li, S., Cho, Y., Song, H.,s Kwon, S.: Which strategies matter for noisy label classification? insight into loss and uncertainty. arXiv preprint arXiv:2008.06218 (2020)
3. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML, pp. 4334–4343. PMLR, Stockholm (2018)
4. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. In: NeurIPS, Vol. 32. Vancouver (2019)

5. Jamal, M. A., Brown, M., Yang, M. H., Wang, L., Gong, B.: Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In: CVPR, pp. 7610–7619. IEEE (2020)
6. Wu, Y., Shu, J., Xie, Q., Zhao, Q., Meng, D.: Learning to purify noisy labels via meta soft label corrector. In: AAAI, pp. 10 388–10 396. AAAI Press, SlidesLive (2021)
7. Zhang, Z., Zhang, H., Arik, S. O., Lee, H., Pfister, T.: Distilling effective supervision from severe label noise. In: CVPR, pp. 9294–9303. IEEE (2020)
8. Zheng, G., Awadallah, A. H., Dumais, S.: Meta label correction for noisy label learning. In: AAAI, AAAI Press, SlidesLive (2021)
9. Mai, Z., Hu, G., Chen, D., Shen, F., Shen, H. T.: Metamixup: Learning adaptive interpolation policy of mixup with metalearning. IEEE Transactions on Neural Networks and Learning Systems (2021)
10. Li, S. et al.: Metasaug: Meta semantic augmentation for long-tailed visual recognition. In: CVPR, pp. 5212–5221. IEEE, Nashville (2021)
11. Zhang, Z., Pfister, T.: Learning fast sample re-weighting without reward data. In: ICCV, pp. 725–734. IEEE, Montreal (2021)
12. Mirzasoleiman, B., Karbasi, A., Sarkar, R., Krause, A.: Distributed submodular maximization: Identifying representative elements in massive data. In: NeurIPS, vol. 26. Lake Tahoe (2013)
13. Thrun, S., Pratt, L.: Learning to learn: Introduction and overview. In: Learning to learn, pp. 3–17. Springer, Boston, MA. (1998)
14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, pp. 1126–1135. PMLR, Sydney (2017)
15. Bao, F., Wu, G., Li, C., Zhu, J., Zhang, B.: Stability and Generalization of Bilevel Programming in Hyperparameter Optimization. In: NeurIPS, vol. 34 (2021)
16. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases, **1**(4) (2009)
17. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., Krause, A.: Lazier than lazy greedy. In: AAAI, pp. 1812–1818. AAAI Press, Austin Texas (2015)
18. Joseph, K. J., Singh, K., Balasubramanian, V. N.: Submodular batch selection for training deep neural networks. In: IJCAI, pp. 2677–2683 (2019)
19. Xiao, Y., Wang, W. Y.: Quantifying uncertainties in natural language processing tasks. In: AAAI, vol. 33, pp. 7322–7329. AAAI Press, Hawaii (2019)
20. Paul, M., Ganguli, S., Dziugaite, G. K.: Deep Learning on a Data Diet: Finding Important Examples Early in Training. In: NeurIPS, Vol, 34 (2021)
21. Nemhauser, G. L. et al.: An analysis of approximations for maximizing submodular set functions—I. Mathematical programming, **14**(1), pp. 265–294 (1978)
22. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S. X.: Large-scale long-tailed recognition in an open world. In: CVPR, pp. 2537–2546. IEEE, California (2019)
23. Van Horn, G. et al.: The inaturalist species classification and detection dataset. In: CVPR, pp. 8769–8778. IEEE, Salt Lake City (2018)
24. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: CVPR, pp. 2691–2699. IEEE (2015)
25. Cui, Y., Jia, M., Lin, T. Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR, pp. 9268–9277. IEEE, California (2019)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778. IEEE, Las Vegas (2016)
27. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC, pp. 87.1–87.12. BMVA Press, York (2016)
28. Zhou, B. et al.: BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR, pp. 9719–9728. IEEE (2020)

29. Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., Wang, X.: A survey of deep active learning. ACM Computing Surveys (CSUR), **54**(9), 1–40 (2021)
30. Beluch, W. H. et al.: The power of ensembles for active learning in image classification. In: CVPR, pp. 9368–9377. IEEE, Salt Lake City (2018)
31. Joshi, A. J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: CVPR, pp. 2372–2379. IEEE, Florida (2009)
32. Lewis, D. D., Gale, W. A.: A sequential algorithm for training text classifiers. In: SIGIR, pp. 3–12. Springer, London (1994)
33. Ranganathan, H., Venkateswara, H., Chakraborty, S., Panchanathan, S: Deep active learning for image classification. In: ICIP, pp. 3934–3938. IEEE, Beijing (2017)
34. Mussmann, S., Liang, P.: On the relationship between data efficiency and error for uncertainty sampling. In: ICML, pp. 3674–3682. PMLR, Stockholm (2018)
35. Lam, C. P., Stork, D. G.: Evaluating classifiers by means of test data with noisy labels. In: IJCAI, pp. 513–518. ACM, San Francisco (2003)
36. Madan, S., Henry, T., Dozier, J., Ho, H., Bhandari, N., Sasaki, T., Durand, F., Pfister H., Boix, X.: When and how do cnns generalize to out-of-distribution category-viewpoint combinations? Nature Machine Intelligence **4**(2), 146–153 (2022)
37. Pagliardini, M. et al.: Improving generalization via uncertainty driven perturbations. arXiv preprint arXiv:2202.05737 (2022)
38. Algan, G., I. Ulusoy.: Image classification with deep learning in the presence of noisy labels: a survey. Knowledge-Based Systems (2019)

Supplementary Materials

1 Supplementary Materials for Section 3.1

In this section, we provide the proofs of Theorem 1.

Proofs of Theorem 1 and its Corollary

Theorem 1. *Suppose a randomized meta optimization algorithm $\mathcal{A}$ is $\beta-$uniformly stable on meta data in expectation, then we have*

$$|E_{\mathcal{A},T,S_m^{me},S_m^{sme}}[gap(T,S_m^{me},S_m^{sme})]| \leq \beta + bd_m, \tag{S-1}$$

where b is the upper bound of the loss and $d_m = d(P_m^{me}||P_m^{sme})$.

Proof. The definition of $gap(T,S_m^{me},S_m^{sme})$ is as follows:

$$gap(T,S_m^{me},S_m^{sme}) = R(\mathcal{A}(T,S_m^{me}),p^{me}) - \hat{R}(\mathcal{A}(T,S_m^{sme}),S_m^{sme}) \tag{S-2}$$

Hence, by Eq. (S-2), we have

$$\begin{aligned}
&E_{S_m^{me},S_m^{sme}}[gap(T,S_m^{me},S_m^{sme})] \\
&= E_{S_m^{me}}[R(\mathcal{A}(T,S_m^{me}),p^{me})] - E_{S_m^{sme}}[\hat{R}(\mathcal{A}(T,S_m^{sme}),S_m^{sme})] \\
&= E_{S_m^{me}}[R(\mathcal{A}(T,S_m^{me}),p^{me})] - E_{S_m^{me}}[\hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me}))] \\
&+ E_{S_m^{me}}[\hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me}))] - E_{S_m^{sme}}[\hat{R}(\mathcal{A}(T,S_m^{sme}),S_m^{sme}))].
\end{aligned} \tag{S-3}$$

According to [6], the following Eq. (S-4) holds

$$|E_{\mathcal{A},T,S_m^{me}}[R(\mathcal{A}(T,S_m^{me}),p^{me}) - \hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me}))]| \leq \beta. \tag{S-4}$$

$$\begin{aligned}
&|E_{S_m^{me}}[\hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me}))] - E_{S_m^{sme}}[\hat{R}(\mathcal{A}(T,S_m^{sme}),S_m^{sme}))]| \\
&= |\int_S \frac{1}{m}\sum_{i=1}^{m} l(\mathcal{A}(T,S),z_i)P_m^{me}(S) - \frac{1}{m}\sum_{i=1}^{m} l(\mathcal{A}(T,S),z_i)P_m^{sme}(S)dS| \\
&\leq |\int_S \frac{1}{m}\sum_{i=1}^{m} l(\mathcal{A}(T,S),z_i)(P_m^{me}(S) - P_m^{sme}(S))dS| \\
&\leq b\int_S |P_m^{me}(S) - P_m^{sme}(S)|dS \leq bd_m,
\end{aligned} \tag{S-5}$$

where b is the upper bound of the loss (following the assumption in [6]) and $S = \{z_1, z_2, ..., z_m\}$.

Hence, according to the absolute value inequality, Eq. (S-4), and Eq. (S-5) we have

$$\begin{aligned}
&|E_{\mathcal{A},T,S_m^{me},S_m^{sme}}[gap(T,S_m^{me},S_m^{sme})]| \\
&\leq |E_{\mathcal{A},T,S_m^{me}}[R(\mathcal{A}(T,S_m^{me}),p^{me}) - \hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me}))]| \\
&+ |E_{\mathcal{A},T,S_m^{me},S_m^{sme}}[\hat{R}(\mathcal{A}(T,S_m^{me}),S_m^{me})) - \hat{R}(\mathcal{A}(T,S_m^{sme}),S_m^{sme}))]| \\
&\leq \beta + |E_{\mathcal{A},T}[bd_m]| = \beta + bd_m.
\end{aligned} \tag{S-6}$$

Theorem 1 is proved.

2 Supplementary Materials for Section 3.3

2.1 Proofs of Lemmas 1 and 2

Definition S-1. *Let X be a finite set. A set function $\mathcal{F}(S) : 2^X \longrightarrow R$ is submodular if $\forall A, B \subset X$ with $A \subset B$ and an element $a \in X \setminus B$, we have*

$$\mathcal{F}(\{a\} \cup A) - \mathcal{F}(A) \geq \mathcal{F}(\{a\} \cup B) - \mathcal{F}(B).$$

Definition S-1 indicates that the gain diminishes as we add elements [11].

Lemma 1. *$\mathcal{F}(\cdot)$ in Eq. (14) is submodular.*

Proof. We prove that the Cleanness criterion and the Diversity criterion are submodular, respectively.

Give two subsets S_1 and S_2 of a training set T such that $S_1 \subset S_2$, and a sample not selected so far: $(x', y') \in T \setminus S_2$, where y' is the label of x'. According to [8], we have

$$D((x', y')|S_1) = D(\{(x', y')\} \cup S_1) - D(S_1) = \min_{(x,y)\in S_1} \phi(\tilde{x}', \tilde{x}), \tag{S-7}$$

$$D((x', y')|S_2) = D(\{(x', y')\} \cup S_2) - D(S_2) = \min_{(x,y)\in S_2} \phi(\tilde{x}', \tilde{x}), \tag{S-8}$$

where $\tilde{x}$ is the output of the final feature encoding layer of x.

Since $S_1 \subset S_2$, according to the proof in [8], the following inequality holds:

$$D((x', y')|S_1) \geq D((x', y')|S_2). \tag{S-9}$$

Hence, according to Definition S-1, $D(\cdot)$ is submodular. For $C(S)$, the following equation holds:

$$C((x', y')|S_1) = C(\{(x', y')\} \cup S_1) - C(S_1) = P(y'|x', \Theta). \tag{S-10}$$

$$C((x', y')|S_2) = C(\{(x', y')\} \cup S_2) - C(S_2) = P(y'|x', \Theta). \tag{S-11}$$

Hence, according to Definition S-1, $C(\cdot)$ is submodular. Any conic combination of submodular functions is submodular [11], and thus $\mathcal{F}(\cdot)$ is submodular.

Lemma 2. *$\mathcal{F}(\cdot)$ in Eq. (14) is monotonically non-decreasing.*

Proof. Consider a subset S and an element $(x', y') \in T \setminus S$. According to [8], when (x', y') is added to S,

$$\mathcal{D}(\{(x', y')\} \cup S) = \mathcal{D}(S) + \min_{x\in S} \phi(\tilde{x}', \tilde{x}). \tag{S-12}$$

Hence, $\mathcal{D}(\cdot)$ is a monotonically non-decreasing function.

For $C(\cdot)$,

$$\mathcal{C}(\{(x', y')\} \cup S) = \mathcal{C}(S) + P(y'|x', \Theta). \tag{S-13}$$

Due to $P(y'|x', \Theta) \geq 0$ and $\lambda \geq 0$, $\mathcal{F}(\cdot)$ is a monotonically non-decreasing function.

2.2 More Details for SOMC

Details of the Algorithmic Steps Algorithm S-1 contains the entire algorithmic steps. SOMC first use uncertainty sampling to sample a subset of size $\frac{|T|}{2}$ and renote the subset as T. Second SOMC divides the data set T into some disjoint subsets, namely, $T_1, T_2, ..., T_K$. Then LtLG is run on these subsets [12]. LtLG starts with an empty set and an element from the random set R is added one by one by maximizing the marginal gain $\mathcal{F}(a|S) = \mathcal{F}(\{a\} \cup S) - \mathcal{F}(S)$. The above-mentioned set R is created by randomly sampling $s = \frac{|V|}{m} \log \frac{1}{\epsilon}$ samples from its superset V, where ϵ is a fixed user-defined tolerance level. ϵ is set as 0.2 in our experiments according to the default setting in [12].

Algorithm S-1 SOMC

Input: Training set T, $u(x_i)$, $i = 1, \cdots, |T|$, m, K, λ, and $\mathcal{F}(\cdot)$ in Eq. (8).
Output: Meta data set S

1: $S \longleftarrow \emptyset$;
2: Obtain a subset of size $\frac{|T|}{2}$ is based on uncertainty sampling and re-denoted as T;
3: Partition T into K disjoint sets $T_1, T_2, ..., T_K$;
4: **for** k = 1 to K **do**
5: $S_k = \emptyset$.
6: **for** j = 1 to m **do**
7: Randomly sample a subset R with size s from $T_k \setminus S_k$;
8: $(x_j^*, y_j^*) = \arg\max_{(x,y) \in R} \mathcal{F}((x, y)|S_k)$;
9: $S_k = \{x_j^*, y_j^*\} \cup S_k$;
10: **end for**
11: **end for**
12: $\tilde{S} \longleftarrow \bigcup_{k=1}^{K} S_k$;
13: **while** $|S| < m$ **do**
14: Randomly sample a subset R with size s from $\tilde{S} \setminus S$;
15: $(x_j^*, y_j^*) = \arg\max_{(x,y) \in R} \mathcal{F}((x, y)|S)$;
16: **if** $n_{y_j^*} < \frac{m}{|C|}$ **then**
17: $S = \{(x_j^*, y_j^*)\} \cup S$;
18: **end if**
19: **end while**
20: **Return** S.

Asymptotic Time Complexity of SOMC The main computational complexity of Algorithm S-1 is divided into two parts. The first part is comprised of steps 4 to 11, and the second part is comprised of steps 13 to 19. The calculation of the Uncertainty sampling and the Cleanness criterion is related linearly to the size of the data set. We calculate the time complexity of the feature balance criterion. The time complexity of the first part is $|T|md$. And the time complexity of the second part is Km^2d. Hence the total asymptotic time complexity of

SOMC is $O((|T| + Km)md)$. If we compute in parallel in the first part, then the time complexity is $O((|T|/K + Km)md)$.

Time Cost. We record the time cost of SOMC on a Linux platform with a 24Gb RTX 3090 GPU. We calculate the ratio of the time for selecting meta data to the total model training time. For MSLC+**SOMC** on CIFAR10 with a 40% flip noise rate, the ratio is 7.73% (639.08 seconds for selecting meta data and 8262.36 seconds for model training). For MetaSAug+**SOMC** on CIFAR100-LT with an imbalance factor of 200, the ratio is 6.47% (132.98 seconds for selecting meta data by SOMC and 2055.35 seconds for model training). We also test SOMC on Clothing1M. Because Clothing1M contains 1 million images from the real world, we first use the Cleanness criterion to filter out a balanced subset with 100,000 images and then build the meta data set by SOMC. ResNet-50 is the backbone network. For MWNet+**SOMC**, the model training time is about 120.25 hours, and the total time to select meta data is approximately 1.14 hours. Hence the ratio of the model training time to the time to select meta data is 0.95%. K is set to 5 for all time cost tests. Compared to model training, the time to select meta data is acceptable.

3 Supplementary Materials for Section 4

3.1 Details About the Benchmark Data Sets

CIFAR. CIFAR10 (CIFAR100) [9] contains 50,000 images uniformly sampled from 10 (100) classes and has 5,000 (500) images per class.
ImageNet-LT. ImageNet-LT is built by Liu et al. [14] from ImageNet [13], which contains 1,281,167 training images and 50,000 validation images. ImageNet-LT consists of 115,846 training samples in 1,000 classes. The imbalance factor is 1,280/5. Following [5], we adopt the original balanced validation to test methods.
INaturalist. The iNaturalist datasets are collected from the real world and thus have an extremely imbalanced class distribution. The iNaturalist 2017 [21] (iNaturalist 2018[35]) is composed of 579,184 (435,713) training images in 5,089 (8,142) classes with an imbalance factor of 3,919/9 (1,000/2). Following MetaSAug [5], we adopt the original validation set to test our method.
Clothing1M. Clothing1M [31] contains 1 million images of clothing obtained from online shopping in real world. It includes 14 categories, including Shirt, Sweater, and so on. The labels of the samples are generated from the description of the corresponding clothes and hence contain a large number of incorrect annotations.

3.2 Details and More Results in Section 4.1

Experiments on Imbalance Classification In this section, we show the hyper-parameters setting of imbalance classification and how to compile some data from the training set to build the meta data using our method.

Table S-1. Test top-1 accuracy (%) of varying K on CIFAR10-LT under the different imbalance factors based on MetaSAug+**SOMC** (CE).

K	1	2	4	5	8	10
200	76.32	76.25	76.22	76.23	76.11	76.02
100	80.38	80.24	80.22	80.17	80.09	79.94
50	83.52	83.49	83.37	83.32	83.17	83.05
20	87.54	87.42	87.41	87.38	87.35	87.27

Data Augmentation Method. In the imbalanced learning experiments, the number of images in some tail categories is too small to choose a balanced meta data set. Hence, we introduce data augmentation techniques to generate new samples for the tail categories. The data compiled from the training set do not participate in any training process except for meta optimization to ensure a fair comparison and highlight the effectiveness of our method. We use four simple data augmentation techniques (i.e., resize, crop, flip and color jittering, denoted as "RCFC") to generate candidate images for SOMC and FSRC.

Implementation details. To demonstrate the advantages of our method, we discard the original meta data set used in [5, 3, 2] from the training set, and they do not participate in any model training process. For MetaSAug [5], we reproduce them with the source code released by authors. Following MetaSAug [5], we train the ResNet-32 [19] on a single GPU with standard stochastic gradient descent (SGD) with momentum 0.9 and weight decay of 5×10^{-4} for all experiments for 200 epochs. The initial learning rate is 0.1 and is decayed by 0.01 at the 160-th and 180-th epochs. The batch size is set as 100 for all experiments. For SOMC, λ is searched in $\{0.3, 0.5, 0.7\}$ and K is searched in $\{2, 5\}$. To select the meta data, we use RCFC to make 1,000 images per class for CIFAR10-LT and 100 images per class for CIFAR100-LT (10,000 images in total for both CIFAR10-LT and CIFAR100-LT). Following MetaSAug, we select 10 images per class as meta data from the augmented images. We compile the meta data per 4 epochs for CIFAR10-LT and CIFAR100-LT. The meta data size m in our SOMC is the same as the competing methods in all experiments.

More Results About MWNet. More results about MWNet are presented in Table 3.4. It can be observed that SOMC is better than the independent meta data based on MWNet. And the test top-1 accuracy results of SOMC achieve an absolute advantage over FSRC.

Details for Hyper-parameter K We study the effect of hyper-parameter K on model performance. Table S-1 shows the accuracy variations on different K values. When K increases, the accuracy demonstrates a slightly downward trend. This is reasonable because when K increases, the quality of the selected meta data will decrease. When $K = 1$, we just use steps 4 to 11 in Algorithm S-1 to select meta data. For efficiency, K is searched in $\{2, 5\}$ in our experiments.

Tables S-2 shows the results on ImageNet-LT. The results obtained by SOMC are still better than those obtained with 10,000 images (10 independently annotated meta data per class) and meta data compiled with FSRC.

Experiments on Noisy Labels Learning Implementation Details. Following the strategy used in MWNet [2], we randomly select two classes as similar classes with equal probability in flip noise simulation. Wide ResNet-28-10 (WRN-28-10) [23] and ResNet-32 [19] are adopted as the base network in learning with uniform and flip noises, respectively. SGD is used with momentum 0.9, a weight decay of 5×10^{-4}, and an initial learning rate 0.1. Following MSLC [4], the max epoch is 120 for both ResNet-32 and WRN-28-10, and the learning rate is decayed with 0.1 at the 80-*th* epoch and the 100-*th* epoch. λ is searched in {0.3, 0.5, 0.7} and K is set to 5. The meta data is compiled per 10 epochs for ResNet-32 and WRN-28-10 when running our SOMC. Since the number of images per category is sufficient, we directly use SOMC to select meta data from the training set.

3.3 Details and More Results in Section 4.2

Hyper-parameter Settings of Large Data Sets Implementation Details on ImageNet-LT. Following MCW [3] and MetaSAug [5], ResNet-50 [19] is used as the backbone network. We reproduce the competing methods based on the code released by Li et al. [5]. The results of MCW are directly from the MetaSAug [5]. The batch size is 64, and the learning rate is decayed by 0.1 at 60-*th* and 80-*th* epoch (for a total epoch 90 as MetaSAug). In addition, we only finetune the last full-connected layer and fix the representations in the meta optimization stage for efficiency as MetaSAug. Except for our hyper-parameters, other hyper-parameters are the same as the baseline. We augment 50 images for each category by using RCFC to select the meta data and construct the meta data by SOMC per 10 epochs in training. The hyper-parameter λ is searched in $\{0.3, 0.5\}$ and K is set to 2.

Implementation Details on INaturalist 2017 and 2018. Following MCW [3] and MetaSAug [5], ResNet-50 [19] is used as the base network for both iNaturalist 2017 and 2018. We perform this part of the experiments on a Linux platform with 4 RTX 3090 GPUs, and each GPU has a capacity of 24Gb. Following MetaSAug and MCW, the networks are pre-trained on ImageNet for iNaturalist 2017 and ImageNet plus iNaturalist 2017 for iNaturalist 2018. We use stochastic gradient descent (SGD) with momentum to train models. The batch size is 64, and the initial learning rate is 0.01. The number of training epochs is the same as that of MetaSAug. Except for our hyper-parameters, other hyper-parameters are the same as the baseline. Using RCFC, we augment 15 images per class for iNaturalist 2017 and 10 for iNaturalist 2018 to select the meta data. The meta data are compiled per 10 epochs. The hyper-parameter λ is searched in $\{0.3, 0.5\}$ and K is set to 2.

Implementation Details on Clothing1M. Following MSLC [4], the pre-trained ResNet-50 on ImageNet is used; SGD is used with a momentum 0.9, a

Table S-2. Test top-1 accuracy (%) on ImageNet-LT.

Method	ImageNet-LT
Base model (CE)	38.88
MCW+**10000 meta images**	44.92
MCW+**FSRC**	45.05
MCW+**SOMC**	**45.97**
MetaSAug+**10000 meta images**	46.21
MetaSAug+**FSRC**	45.77
MetaSAug+**SOMC**	**46.68**

Table S-3. Test top-1 accuracy (%) on ImageNet-LT of methods with different backbone networks.

Network	MCW	MetaSAug	MetaSAug+**SOMC**
ResNet-50	44.92	46.21	**46.68**
ResNet-101	46.24	49.05	**49.52**
ResNet-152	46.82	50.03	**50.38**

weight decay 10^{-3}, an initial learning rate 0.01, and batch size 32. The learning rate is divided by 10 after five epochs (for a total epochs 10). λ is searched in $\{0.3, 0.5\}$ and K is set to 5. Since Clothing1M contains one million pictures, for efficiency, we use Cleanness criterion to filter out a balanced subset with a size of 100,000, and then use SOMC to select meta data in this subset. We select meta data per 2 epochs.

Results of Deeper Backbone Networks Different deeper backbone networks are utilized to evaluate our method as [5]. Table S-3 shows the results of MCW and MetaSAug with ResNet-50, ResNet-101, and ResNet-152. SOMC is run based on MetaSAug, and it can be observed that our method can achieve better results without independent meta data.

3.4 Details and more comprehensive results in Section 4.3

Ablation Study under Flip Noise We also test the effectiveness of our method in the presence of corrupted labels. Table S-6 shows the results on CIFAR-10 with the different flip noise rates based on MSLC+**SOMC**. It can be observed that removing each criterion causes a performance drop. This results indicate that each of the three criteria and uncertainty sampling are useful in SOMC.

More Comprehensive Comparison Comprehensive Results on Imbalance Classification. We conduct a comprehensive comparison with the following methods: Base model (CE), Class-balanced CE [10], Class-balanced fine-tuning [15], BBN [18], Mixup [20], L2RW [1], MWNet [2], MCW [3], MetaSAug

Table S-4. Ablation study of MSLC+**SOMC** on CIFAR10 under flip noise.

Noise rate	20%	40%
SOMC w/o Uncertainty	89.62	87.99
SOMC w/o Diversity	89.48	87.84
SOMC w/o Cleanness	88.94	86.77
SOMC w/o Balance	89.87	88.07
SOMC	**91.13**	**89.55**

Table S-5. Test top-1 accuracy (%) of ResNet-32 on CIFAR10-LT and CIFAR100-LT under different imbalance settings. CE, FL and LDAM mean Cross-entropy loss, Focal loss and LDAM loss respectively.

Data set	CIFAR10-LT					CIFAR100-LT				
Imbalance factor	200	100	50	20	10	200	100	50	20	10
Base model (CE)	65.87	70.14	74.94	82.44	86.18	34.70	38.46	44.02	51.06	55.73
Class-balanced CE	68.77	72.68	78.13	84.56	86.90	35.56	38.77	44.79	51.94	57.57
Class-balanced fine-tuning	66.24	71.34	77.44	83.22	83.17	38.66	41.50	46.22	52.30	57.57
BBN	-	79.82	82.18	-	88.32	-	42.56	47.02	-	59.12
Mixup	-	73.06	77.82	-	87.10	-	39.54	44.99	-	58.02
L2RW	66.25	72.23	76.45	81.35	82.12	33.00	38.90	43.17	50.75	52.12
MWNet+**100/1000 meta images** (CE)	67.20	73.57	79.10	84.55	87.55	36.62	41.61	45.66	53.04	58.91
MWNet+**FSRC** (CE)	68.25	74.94	79.56	84.86	87.89	36.87	41.68	45.84	53.83	58.97
MWNet +**SOMC** (CE)	69.53	75.88	80.77	85.98	88.58	38.21	42.59	46.93	54.71	59.21
MCW +**100/1000 meta images** (CE)	70.66	76.41	80.51	86.46	88.85	39.31	43.35	48.53	55.62	59.58
MCW+**FSRC** (CE)	72.34	77.65	81.31	86.25	88.02	38.53	44.21	49.72	55.98	60.17
MCW+**SOMC** (CE)	73.71	79.24	82.34	86.98	88.67	39.95	45.97	51.28	57.32	61.11
MetaSAug+**100/1000 meta images** (CE)	76.16	**80.48**	83.52	87.20	88.89	42.27	46.97	51.98	57.75	61.75
MetaSAug+**FSRC** (CE)	75.41	79.28	82.87	86.81	88.37	42.53	47.02	51.61	57.87	61.35
MetaSAug+**SOMC** (CE)	**76.25**	80.25	**83.61**	**87.43**	**89.02**	**43.32**	**48.03**	**52.36**	**58.52**	**61.88**
FL	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
Class-balanced FL	68.15	74.57	79.22	83.78	87.48	36.23	39.60	45.21	52.59	57.99
MCW+**100/1000 meta images** (FL)	74.43	78.90	82.88	86.10	88.37	39.34	44.70	50.08	55.73	59.59
MCW+**FSRC** (FL)	74.57	79.23	83.06	86.22	88.59	39.67	44.85	50.35	55.89	59.87
MCW+**SOMC** (FL)	75.26	80.17	**83.65**	86.52	88.84	40.26	45.96	51.13	56.67	60.35
MetaSAug+**100/1000 meta images** (FL)	75.73	80.25	83.04	**86.95**	88.61	40.42	45.95	51.57	57.65	61.17
MetaSAug+**FSRC** (FL)	75.12	79.87	82.52	85.99	88.21	39.77	45.86	51.22	57.25	60.84
MetaSAug+**SOMC** (FL)	**76.01**	**80.44**	83.41	86.77	**88.87**	**40.69**	**46.90**	**51.99**	**57.81**	**61.65**
LDAM	66.75	73.55	78.83	83.89	87.32	36.53	40.60	46.16	51.59	57.29
LDAM-DRW	74.74	78.12	81.27	84.90	88.37	38.45	42.89	47.97	52.99	58.78
MCW+**100/1000 meta images** (LDAM)	77.23	80.00	82.23	84.37	87.40	39.53	44.08	49.16	52.38	58.00
MCW+**FSRC** (LDAM)	76.85	79.97	82.04	85.12	88.03	40.25	44.83	49.79	53.34	59.46
MCW+**SOMC** (LDAM)	**77.69**	80.43	82.86	85.74	88.51	41.37	45.73	50.62	54.29	60.30
MetaSAug+**100/1000 meta images** (LDAM)	76.42	80.43	83.72	87.32	**88.77**	42.87	**48.29**	52.18	57.65	61.37
MetaSAug+**FSRC** (LDAM)	75.89	79.93	83.21	86.72	87.93	42.69	47.43	51.65	57.54	61.35
MetaSAug+**SOMC** (LDAM)	76.56	**80.61**	**83.96**	**87.45**	88.57	**43.48**	48.17	**52.56**	**58.43**	**61.93**

[5], and FSRC [7]. Table 3.4 shows the comprehensive comparison. For FSRC, we only compare the proposed meta data selection criteria for a fair comparison. For MetaSAug, we reproduce the comparison method based on the code released by the authors. Other results are obtained directly from the study of MetaSAug.

Results. These results are divided into three groups according to different loss functions. Table 3.4 shows that our method achieves better results in almost all cases. Our method can further improve the accuracy of the model or achieve

Table S-6. Test top-1 accuracy (%) comparison on CIFAR10 and CIFAR100 of ResNet-32 with varying noise rates under flip noise.

Data set	CIFAR10			CIFAR100		
noise rate	0%	20%	40%	0%	20%	40%
Base model (CE)	92.89±0.32	76.83±2.30	70.77±2.31	70.50±0.12	50.86±0.27	43.01±1.16
Reed-Hard	92.31±0.25	88.28±0.36	81.06±0.76	69.02±0.32	60.27±0.76	50.40±1.01
S-Model	83.61±0.13	79.25±0.30	75.73±0.32	51.46±0.20	45.45±0.25	43.81±0.15
SPL	88.52±0.21	87.03±0.34	81.63±0.52	67.55±0.27	63.63±0.30	53.51±0.53
Focal Loss	93.03±0.16	86.45±0.19	80.45±0.97	70.02±0.53	61.87±0.30	54.13±0.40
Co-teaching	89.87±0.10	82.83±0.85	75.41±0.21	63.31±0.05	54.13±0.55	44.85±0.81
D2L	92.02±0.14	87.66±0.40	83.89±0.46	68.11±0.26	63.48±0.53	51.83±0.33
Fine-tining	**93.23±0.23**	82.47±3.64	74.07±1.56	70.72±0.22	56.98±0.50	46.37±0.25
MentorNet	92.13±0.30	86.36±0.31	81.76±0.28	70.24±0.21	61.97±0.47	52.66±0.56
L2RW	89.25±0.37	87.86±0.36	85.66±0.51	64.11±1.09	57.47±1.16	50.98±1.55
GLC	91.02±0.20	89.68±0.33	88.92±0.24	65.42±0.23	63.07±0.53	62.22±0.62
MWNet+**1000 meta images**	92.04±0.15	90.33±0.61	87.54±0.23	70.11±0.33	64.22±0.28	58.64±0.47
MWNet+**FSRC**	92.42±0.12	90.65±0.36	87.25±0.41	70.52±0.11	65.26±0.12	59.47±0.22
MWNet+**SOMC**	93.06±0.06	91.37±0.11	88.65±0.26	**71.39±0.31**	66.69±0.11	60.34±0.19
MSLC+**1000 meta images**	92.75±0.15	**91.67±0.19**	**90.23±0.13**	70.37±0.31	**67.59±0.06**	**65.02±0.21**
MSLC+**FSRC**	92.46±0.13	89.78±0.32	88.61±0.27	70.29±0.21	64.97±0.19	61.15±0.46
MSLC+**SOMC**	92.83±0.09	91.13±0.21	89.55±0.25	70.82±0.15	66.33±0.11	62.58±0.28

comparable performance without independent metadata. SOMC also achieves better performance than FSRC in all cases.

Comprehensive Results on Corrupted Labels Classification. We compare our method with the following methods: CE (Cross-Entropy), Reed-Hard [24], S-Model [25], SPL [26], Focal Loss [17], Co-teaching [27], D2L [28], Fine-tining, fine-tuning the result of Base model on the meta data with clean labels to further enhance its performance; MentorNet [29], L2RW [1], GLC [30], MWNet [2], MSLC [4], and FSRC [7]. Tables S-6 and S-7 show the competing results. For FSRC, we only compare its proposed meta data selection criteria for a fair comparison. For MSLC, because the base network and noise types in MWNet and MSLC are different, we reproduce their results through the author's open-source code.

Results. Tables S-6 and S-7 show that SOMC can achieve better results than the independent meta data based on MWNet. When the noise rate is 0%, SOMC can achieve better results than the independent meta data based on MSLC. When the noise rate increases, the performance of SOMC degrades more than the independent meta data based on MSLC, which indicates in the case of a high noise rate, independent meta data is required. SOMC achieves an absolute advantage over FSRC.

Comprehensive Results on Large Data Sets. For ImageNet-LT, we compare our method with CE, Class-balanced CE [10], OLTR [14], LDAM [16], LDAM-DRW [16], MCW [3], MetaSAug [5], and FSRC [7]. For FSRC, we only compare its proposed meta data selection criteria for a fair comparison. For MetaSAug, we reproduce the comparison method based on the code released by the authors. Other results are obtained from MetaSAug.

For iNaturalist 2017 and 2018, we compare SOMC with the following methods: CE (Cross-Entropy Loss), Class-balanced CE [10], Class-balanced focal [10], cRT [22], BBN [18], LDAM [16], LDAM-DRW [16], MCW [3], MetaSAug [5],

Table S-7. Test top-1 accuracy (%) comparison on CIFAR10 and CIFAR100 of WRN-28-10 with varying noise rates under uniform noise.

Data set	CIFAR10			CIFAR100		
noise rate	0%	40%	60%	0%	40%	60%
Base model (CE)	95.60±0.22	68.07±1.23	53.12±3.03	79.95±1.26	51.11±0.42	30.92±0.33
Reed-Hard	94.38±0.14	81.26±0.51	73.53±1.54	64.45±1.02	51.27±1.18	26.95±0.98
S-Model	83.79±0.11	79.58±0.33	-	52.86±0.99	42.12±0.99	-
SPL	90.81±0.34	86.41±0.29	53.10±1.78	59.79±0.46	46.31±2.45	19.08±0.57
Focal Loss	**95.70±0.15**	75.96±1.31	51.87±1.19	81.04±0.24	51.19±0.46	27.70±3.77
Co-teaching	88.67±0.25	74.81±0.34	73.06±0.25	61.80±0.25	46.20±0.15	35.67±1.25
D2L	94.64±0.33	85.60±0.13	68.02±0.41	66.17±1.42	52.10±0.97	41.11±0.30
Fine-tining	95.65±0.15	80.47±0.25	78.75±2.40	80.88±0.21	52.49±0.74	38.16±0.38
MentorNet	94.35±0.42	87.33±0.22	82.80±1.35	73.26±1.23	61.39±3.99	36.87±1.47
L2RW	92.38±0.10	86.92±0.19	82.24±0.36	72.99±0.58	60.79±0.91	48.15±0.34
GLC	94.30±0.19	88.28±0.03	83.49±0.24	73.75±0.51	61.31±0.22	50.81±1.00
MWNet+**1000 meta images**	94.52±0.25	89.27±0.28	84.07±0.33	78.76±0.24	67.73±0.26	58.75±0.11
MWNet+**FSRC**	95.03±0.23	88.78±0.16	84.26±0.17	79.95±0.08	67.88±0.25	59.37±0.28
MWNet+**SOMC**	95.69±0.09	89.81±0.13	85.16±0.12	80.68±0.32	68.63±0.14	60.65±0.19
MSLC+**1000 meta images**	95.42±0.07	**91.54±0.15**	**87.27±0.27**	80.75±0.11	**71.83±0.24**	**65.37±0.53**
MSLC+**FSRC**	95.23±0.17	88.15±0.31	81.84±0.33	80.49±0.23	67.86±0.14	59.63±0.42
MSLC+**SOMC**	95.65±0.05	89.38±0.13	83.56±0.27	**81.36±0.31**	68.75±0.29	61.03±0.17

FSRC [7]. For FSRC, we only compare its proposed meta data selection criteria for a fair comparison. Other results are from MetaSAug.

Table S-8. Test top-1 accuracy (%) on ImageNet-LT of different models.

Method	ImageNet-LT
Base model (CE)	38.88
Class-balanced CE	40.85
OLTR	40.36
LDAM	41.86
LDAM-DRW	45.74
MCW+**10000 meta images**	44.92
MCW+**FSRC**	45.05
MCW+**SOMC**	45.97
MetaSAug+**10000 meta images**	46.21
MetaSAug+**FSRC**	45.77
MetaSAug+**SOMC**	**46.68**

For Clothing1M, we compare our method with the following: Base model, Bootstrapping [24], U-correction [33], Joint Optimization [32], MWNet [2], FSRC [7], MSLC [4]. For FSRC, we only compare its proposed meta data selection criteria for a fair comparison. Other results are obtained from MSLC.

Results. Table S-8 shows the results of different models on ImageNet-LT. It can be observed that SOMC can achieve better results than the independent meta data used in baselines. And SOMC can select higher quality data than FSRC. Table S-9 shows the results on iNaturalist 2017 and 2018. From the results, we can see that although 25445 (for iNat2017) and 16284 (for iNat2018) independent annotated images are used for MCW and MetaSAug, their perfor-

Table S-9. Test top-1 accuracy (%) on iNaturalist (iNat) 2017 and 2018 of different models. * indicates that the results are from the original paper.

Method	iNat 2017	iNat 2018
Base model (CE)	56.79	65.76
Class-balanced CE	57.98	66.43
Class-balanced FL*	58.08	61.12
cRT*	-	67.60
BBN*	63.39	66.29
LDAM*	-	64.58
LDAM	60.85	65.87
LDAM-DRW*	-	68.00
LDAM-DRW	62.16	67.88
MCW+**25445/16284 meta images**	59.38	67.55
MCW+**FSRC**	58.76	67.52
MCW+**SOMC**	60.47	68.89
MetaSAug+**25445/16284 meta images**	63.28	68.75
MetaSAug+**FSRC**	62.59	68.28
MetaSAug+**SOMC**	63.53	69.05
MetaSAug+**SOMC** with BBn model	**65.34**	**70.66**

Table S-10. Test top-1 accuracy (%) on on Clothing1M.

Method	Clothing1M
Base model (CE)	68.94
Bootstrapping	69.12
U-correction	71.00
Joint Optimization	72.23
MWNet+**7000 meta images**	73.72
MWNet+**FSRC**	73.01
MWNet+**SOMC**	73.89
MSLC+**7000 meta images**	**74.02**
MSLC+**FSRC**	73.23
MSLC+**SOMC**	73.67

mances are inferior to those when the meta data are compiled by our SOMC, which indicates that the quality of meta data in meta-optimization is critical. When the BBN pre-train model [18] is used, the combination of MetaSAug+SOMC achieves the best results. Table S-10 shows the results of different models on Clothing1M. The table shows that SOMC can achieve better or comparable performance than the independent meta data and FSRC.

3.5 Study for the Robustness of Meta Optimization Methods

Ghosh and Lan [34] think that using the mean square error loss can improve the robustness of meta-optimization to noise. In this section, we take MSLC as an example to test its robustness to noise. Table S-11 shows the accuracy of injecting the same proportion of noise into meta data. The table shows that even if the

meta data are injected with the same ratio of noisy labels, it still achieves better performance than Base model (CE). The results show that meta-optimization can allow meta data to have noisy labels to a certain extent. However, higher quality metadata can achieve better performance.

Table S-11. Test top-1 accuracy (%) on CIFAR10 and CIFAR100 of ResNet-32 under flip noise. In this experiment, meta data are injected with the same proportion of noisy labels.

Data set	CIFAR10		CIFAR100	
Noise rate	20%	40%	20%	40%
Base model (CE)	76.83	70.77	50.86	43.01
MSLC+**1000 clean meta images**	91.67	90.23	67.59	65.02
MSLC+**1000 noisy meta images**	87.45	85.92	63.56	60.37

References

1. M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*. PMLR, 2018, pp. 4334–4343.
2. J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *NeurIPS*, 2019.
3. M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *CVPR*, 2020, pp. 7610–7619.
4. Y. Wu, J. Shu, Q. Xie, Q. Zhao, and D. Meng, "Learning to purify noisy labels via meta soft label corrector," *AAAI*, 2021.
5. S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta semantic augmentation for long-tailed visual recognition," in *CVPR*, 2021, pp. 5212–5221.
6. F. Bao, G. Wu, C. Li, J. Zhu, and B. Zhang, "Stability and generalization of bilevel programming in hyperparameter optimization," *NeurIPS*, 2021.
7. Z. Zhang and T. Pfister, "Learning fast sample re-weighting without reward data," in *ICCV*, 2021, pp. 725–734.
8. K. Joseph, V. Teja R, K. Singh, and V. N. Balasubramanian, "Submodular batch selection for training deep neural networks," in *IJCAI*, 2019, pp. 2677–2683.
9. A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
10. Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277.
11. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
12. B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *AAAI*, vol. 29, no. 1, 2015.
13. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

14. Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *CVPR*, 2019, pp. 2537–2546.
15. Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *CVPR*, 2018, pp. 4109–4118.
16. K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *NeurIPS*, 2019.
17. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.
18. B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *CVPR*, 2020, pp. 9719–9728.
19. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
20. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *ICLR*, 2018.
21. G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *CVPR*, 2018, pp. 8769–8778.
22. B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2020.
23. S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016, pp. 87.1–87.12.
24. S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," *ICLR*, 2015.
25. J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer." in *ICLR*, 2017.
26. M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models." in *NeurIPS*, vol. 1, 2010, p. 2.
27. B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *NeurIPS*, 2018.
28. X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," in *ICML*. PMLR, 2018, pp. 3355–3364.
29. L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *ICML*. PMLR, 2018, pp. 2304–2313.
30. D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," *NeurIPS*, 2018.
31. T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *CVPR*, 2015, pp. 2691–2699.
32. D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *CVPR*, 2018, pp. 5552–5560.
33. E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *ICML*. PMLR, 2019, pp. 312–321.
34. A. Ghosh and A. Lan, "Do we really need gold samples for sample weighting under label noise?" in *WACV*, 2021, pp. 3922–3931.
35. Inaturalist 2018 competition dataset. 2018, https://github.com/visipedia/inat_comp.