Supplementary Materials for 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

1 Supplementary Materials for Section 3.1

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].

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

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

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

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.

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 3.3. 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.

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

Table S-1. 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**

[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 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

Table S-2. 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-3. 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**

$\{0.3, 0.5\}$ 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-1 shows the results of MCW and MetaSAug with ResNet-50, ResNet-101, and ResNet-152. SOMC is run

Table S-4. 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

based on MetaSAug, and it can be observed that our method can achieve better results without independent meta data.

Table S-5. 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

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 re-

sults 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 [5], and FSRC [7]. Table 3.3 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 3.3 shows that our method achieves better results in almost all cases. Our method can further improve the accuracy of the model or achieve 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-4 and S-5 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-4 and S-5 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], FSRC [7]. For FSRC, we only compare its proposed meta data selection criteria for a fair comparison. Other results are from MetaSAug.

Table S-6. 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**

Table S-7. 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**

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-6 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-7 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 performances 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

Table S-8. 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

achieves the best results. Table S-8 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.

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.