Suppressing Mislabeled Data via Grouping and Self-Attention

Xiaojiang Peng[⋆1,2], Kai Wang[⋆1,2], Zhaoyang Zeng[⋆3], Qing Li[4], Jianfei Yang[5], and Yu Qiao[†1,2]

[1] Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 518055, China
[2] SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society
[3] Sun Yat-sen University
[4] Southwest Jiaotong University
[5] Nanyang Technological University, Singapore

Abstract. Deep networks achieve excellent results on large-scale clean data but degrade significantly when learning from noisy labels. To suppressing the impact of mislabeled data, this paper proposes a conceptually simple yet efficient training block, termed as Attentive Feature Mixup (AFM), which allows paying more attention to clean samples and less to mislabeled ones via sample interactions in small groups. Specifically, this plug-and-play AFM first leverages a *group-to-attend* module to construct groups and assign attention weights for group-wise samples, and then uses a *mixup* module with the attention weights to interpolate massive noisy-suppressed samples. The AFM has several appealing benefits for noise-robust deep learning. (i) It does not rely on any assumptions and extra clean subset. (ii) With massive interpolations, the ratio of useless samples is reduced dramatically compared to the original noisy ratio. (iii) It jointly optimizes the interpolation weights with classifiers, suppressing the influence of mislabeled data via low attention weights. (iv) It partially inherits the vicinal risk minimization of mixup to alleviate over-fitting while improves it by sampling fewer feature-target vectors around mislabeled data from the mixup vicinal distribution. Extensive experiments demonstrate that AFM yields state-of-the-art results on two challenging real-world noisy datasets: Food101N and Clothing1M.

Keywords: Noisy-labeled data, mixup, noisy-robust learning

1 Introduction

In recent years, deep neural networks (DNNs) have achieved great success in various tasks, particularly in supervised learning tasks on large-scale image recognition challenges, such as ImageNet [6] and COCO [21]. One key factor that drives impressive results is the large amount of well-labeled images. However,

⋆ Equally-contributed first authors.[†]Corresponding author (yu.qiao@siat.ac.cn)

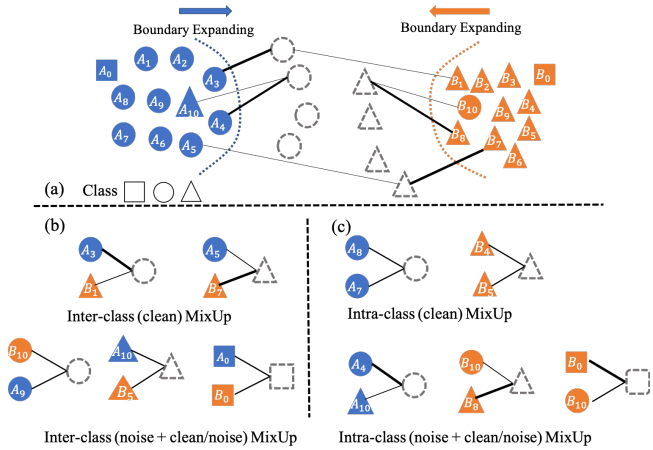

Fig. 1: Suppressing mislabeled samples by grouping and self-attention mixup. *Different colors and shapes denote given labels and ground truths. Thick and thin lines denote high and low attention weights, respectively.* A_0, A_{10}, B_0, and B_{10} are supposed to be mislabeled samples, and can be suppressed by assigning low interpolation weights in mixup operation.

high-quality annotations are laborious and expensive, even not always available in some domains. To address this issue, an alternative solution is to crawl a large number of web images with tags or keywords as annotations [8, 19]. These annotations provide weak supervision, which are noisy but easy to obtain.

In general, noisy labeled examples hurt generalization because DNNs easily overfit to noisy labels [7, 30, 2]. To address this problem, it is intuitive to develop noise-cleaning methods which aim to correct the mislabeled samples either by joint optimization of classification and relabeling [31] or by iterative self-learning [11]. However, the noise-cleaning methods often suffer from the main difficulty in distinguishing mislabeled samples from hard samples. Another solution is to develop noise-robust methods which aims to reduce the contributions of mislabeled samples for model optimization. Along this solution, some methods estimate a matrix for label noise modeling and use it to adapt output probabilities and loss values [30, 35, 26]. Some others resort to curriculum learning [4] by either designing a step-wise easy-to-hard strategy for training [10] or introducing an extra MentorNet [12] for sample weighting. However, these methods independently estimate the importance weights for individuals which ignore the comparisons among different samples while they have been proven to be the key of humans to perceive and learn novel concepts from noisy input images [29]. Some other solutions follow semi-supervised configuration where they assume a small manually-verified set can be used [20, 32, 15, 17]. However, this assumption may be not supported in real-world applications. With the Vicinal Risk Minimization(VRM) principle, mixup [36, 33] exploits a vicinal distribution for

sampling virtual sample-target vectors, and proves its robustness for synthetic noisy data. But its effectiveness is limited in real-world noisy data [1].

In this paper, we propose a conceptually simple yet efficient training block, termed as Attentive Feature Mixup (AFM), to suppress mislabeled data thus to make training robust to noisy labels. The AFM is a plug-and-play block for training any networks and is comprised of two crucial parts: 1) a *Group-to-Attend* (GA) module that first randomly groups a minibatch images into small subsets and then estimates sample weights within those subsets by an attention mechanism, and 2) a *mixup* module that interpolates new features and soft labels according to self-attention weights. Particularly, for the GA module, we evaluate three feature interactions to estimate group-wise attention weights, namely concatenation, summary, and element-wise multiplication. The interpolated samples and original samples are respectively fed into an interpolation classifier and a normal classifier. Figure 1 illustrates how AFM suppress mislabeled data. Generally, there exists two main types of mixup: intra-class mixup (Figure 1 (c)) and inter-class mixup (Figure 1 (b)). For both types, the interpolations between mislabeled samples and clean samples may become useful for training with adaptive attention weights, *i.e.* low weights for the mislabeled samples and high weights for the clean samples.In other words, our AFM hallucinates numerous useful noisy-reduced samples to guide deep networks learn better representations from noisy labels. Overall, as a noisy-robust training method, our AFM is promising in the following aspects.

- It does not rely on any assumptions and extra clean subset.
- With AFM, the ratio of harmful noisy interpolations (*i.e.* between noisy samples) over all interpolations is largely less than the original noisy ratio.
- It jointly optimizes the mixup interpolation weights and classifier, suppressing the influence of mislabeled data via low attention weights.
- It partially inherits the vicinal risk minimization of mixup to alleviate overfitting while improves it by sampling less feature-target vectors around mislabeled data from the mixup vicinal distribution.

We validate our AFM on two popular real-world noisy-labeled datasets: Food101N [15] and Clothing1M [35]. Experiments show that our AFM outperforms recent state-of-the-art methods significantly with accuracies of **87.23**% on Food101N and **82.09**% on Clothing1M.

2 Related Work

2.1 Learning with Noisy Labeled Data

Learning with noisy data has been vastly studied on the literature of machine learning and computer vision. Methods on learning with label noise can be roughly grouped into three categories: noise-cleaning methods, semi-supervised methods and noise-robust methods.

First, noise-cleansing methods aim to identify and remove or relabel noisy samples with filter approaches [3, 24]. Brodley *et al.* [5] propose to filter noisy

samples using ensemble classifiers with majority and consensus voting. Sukhbaatar *et al.* [30] introduce an extra noise layer into a standard CNN which adapts the network outputs to match the noisy label distribution. Daiki *et al.* [31] propose a joint optimization framework to train deep CNNs with label noise, which updates the network parameters and labels alternatively. Based on the consistency of the noisy groundtruth and the current prediction of the model, Reed *et al.* [27] present a 'Soft' and a 'Hard' bootstrapping approach to relabel noisy data. Li *et al.* [20] relabel noisy data using the noisy groundtruth and the current prediction adjusted by a knowledge graph constructed from DBpedia-Wikipedia.

Second, semi-supervised methods aim to improve performance using a small manually-verified clean set. Lee *et al.* [15] train an auxiliary CleanNet to detect label noise and adjust the final sample loss weights. In the training process, the CleanNet needs to access both the original noisy labels and the manually-verified labels of the clean set. Veit *et al.* [32] use the clean set to train a label cleaning network but with a different architecture. These methods assume there exists such a label mapping from noisy labels to clean labels. Xiao *et al.* [35] mix the clean set and noisy set, and train an extra CNN and a classification CNN to estimate the posterior distribution of the true label. Li *et al.* [18] first train a teacher model on clean and noisy data, and then distill it into a student model trained on clean data.

Third, the noise-robust learning methods are assumed to be not too sensitive to the presence of label noise, which directly learn models from the noisy labeled data [13, 14, 25, 26, 34]. Manwani *et al.* [23] present a noise-tolerance algorithm under the assumption that the corrupted probability of an example is a function of the feature vector of the example. With synthetic noisy labeled data, Rolnick *et al.* [28] demonstrate that deep learning is robust to noise when training data is sufficiently large with large batch size and proper learning rate. Guo *et al.* [10] develop a curriculum training scheme to learn noisy data from easy to hard. Han *et al.* [11] propose a Self-Learning with Multi-Prototype (SMP) method to learn robust features via alternatively training and clustering which is time-consuming. Wang *et al.* [34] propose to suppress uncertain samples with self-attention, ranking loss, and relabeling. Our method is most related to MetaCleaner [37], which hallucinates a clean (precisely noise-reduced) representation by mixing samples (the ratio of the noisy images need to be small) from the same category. Our work differs from it in that i) we formulate the insight as attentive mixup, and ii) hallucinate noisy-reduced samples not only within class but also between classes which significantly increases the number of interpolations and expands the decision boundaries. Moreover, we introduce more sample interactions rather than the concatenation in [37], and find a better one.

2.2 Mixup and Variations

Mixup [36] regularizes the neural network to favor simple linear behavior in-between training examples. Manifold Mixup [33] leverages semantic interpolations in random layers as additional training signal to optimize neural networks. The interpolation weights of those two methods are drawn from a β distribution.

Meta-Mixup [22] introduces a meta-learning based online optimization approach to dynamically learn the interpolation policy from a reference set. AdaMixup [9] also learns the interpolation policy from dataset with an additional network to infer the policy and an intrusion discriminator. Our work differs from these variations in that i) we design a Group-to-Attend mechanism to learn attention weights for interpolating in a group-wise manner which is the key to reduce the influence of noises and ii) we address the noisy-robust problem on real-world noisy data and achieve state-of-the-art performance.

3 Attentive Feature Mixup

As proven in cognitive studies, we human mainly perceive and learn novel concepts from noisy input images by comparing and summary [29]. Based on this motivation, we propose a simple yet efficient model, called Attentive Feature Mixup (AFM), which aims to learn better features by making clean and noisy samples interact with each other in small groups.

3.1 Overview

Our AFM works on traditional CNN backbones and includes two modules: i) Group-to-Attend (GA) module and ii) mixup module, as shown in Figure 2.

Let $\mathcal{B} = \{(\mathbf{I}_1, y_1), (\mathbf{I}_2, y_2), \cdots, (\mathbf{I}_n, y_n)\}$ be the mini-batch set of a noisy labeled dataset, which contains n samples, and $y_i \in \mathcal{R}^C$ is the noisy one-hot label vector of image $\mathbf{I}_i$. The AFM works as the following procedure. First, a CNN backbone $\phi(\cdot;\theta)$ with parameter θ is used to extract image features $\{x_1, x_2, \cdots, x_n\}$. Then, the Group-to-Attend (GA) module is used to divide the mini-batch images into small groups and learn attention weights for each samples within each group. Subsequently, with the group-wise attention weights, a mixup module is used to interpolate new samples and soft labels. Finally, these interpolations along with the original image features are fed into an interpolation classifier f_{c1} (*i.e.* FC layer) and a normal classifier f_{c2} (*i.e.* FC layer), respectively. Particularly, the interpolation classifier is supervised by the soft labels from the mixup module and the normal classifier by the original given labels which are noisy. Our AFM partially inherits the vicinal risk minimization of mixup to alleviate over-fitting with massive interpolations. Further, with jointly optimizing the mixup interpolation weights and classifier, AFM improves mixup by sampling less feature-target vectors around mislabeled data from the mixup vicinal distribution.

3.2 Group-to-Attend Module

In order to obtain meaningful attention weights, *i.e.* high weights for clean samples and low weights for mislabeled samples, we elaborately design a Group-to-Attend module, which consists of four crucial steps. First, we randomly and repeatedly selecting K samples to construct groups as many as possible (the

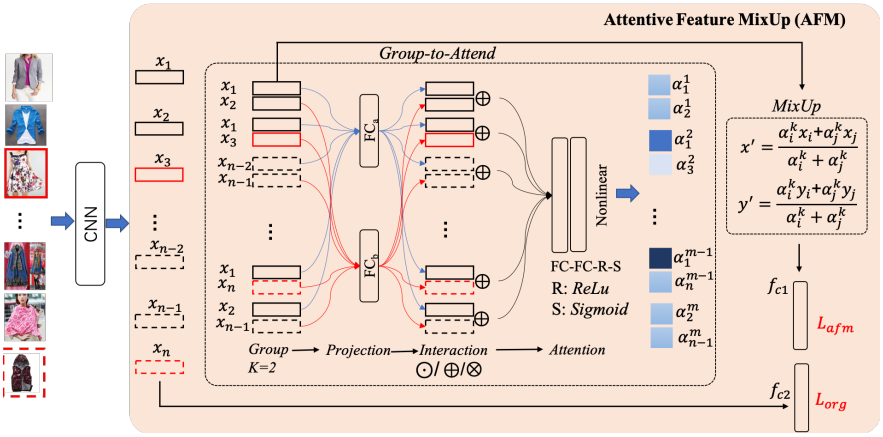

Fig. 2: The pipeline of Attentive Feature Mixup (AFM). Given a mini-batch of n images, a backbone CNN is first applied for feature extraction. Then, a Group-to-Attend (GA) module randomly composites massive groups with the group size K and linearly projects each element within a group with a separated FC layer, and then combines each group with an interaction (*i.e.* concatenation ($\odot$), sum ($\oplus$), and element-wise multiplication ($\otimes$)), and finally outputs K attention weights for each group. With the group-wise attention weights, a mixup module is used to interpolate new samples and soft labels.

number of groups depends on the input batch size and the GPU memory). Second, we use K fully-connected (FC) layers to map the ordered samples of each group into new feature embeddings for sample interactions. As an example of $K = 2$ in Figure 2, x_i and x_j are linearly projected as,

$$\tilde{x}_i = f_a(x_i; w_a), \quad \tilde{x}_j = f_b(x_j; w_b), \tag{1}$$

where w_a and w_b are the parameters of FC layer f_a and f_b, respectively. Third, we further make $\tilde{x}_i$ and $\tilde{x}_j$ interact for group-wise weight learning. Specifically, we experimentally explore three kinds of interactions: concatenation ($\odot$), sum ($\oplus$), and element-wise multiplication ($\otimes$). Last, we apply a light-weight self-attention network to estimate group-wise attention weights. Formally, for $K = 2$ and the sum interaction, this step can be defined as follows,

$$\begin{aligned} [\alpha_i^k, \alpha_j^k] &= \psi_t(\tilde{x}_i \oplus \tilde{x}_j; \theta_t) \\ &= \psi_t(f_a(x_i; w_a) \oplus f_b(x_j; w_b); \theta_t), \end{aligned} \tag{2}$$

where ψ_t is the attention network, θ_t denotes its parameters, and k denotes the $k-$th group. For the architecture of ψ_t, we follow the best one of [37], *i.e.* FC-FC-ReLu-Sigmoid. It is worth noting that feature interaction is crucial for learning meaningful attention weights since the relationship between noisy and clean samples within a group can be learned efficiently while not the case of non-interaction (*i.e.* learning weights for each other separately).

Proposition 1 *The attention weights are meaningful with sum interaction if and only if* $f_a \neq f_b$.

Proof. Assume we remove the projection layers f_a and f_b or share them as the same function f, then Eq. (2) is rewritten as,

$$\begin{aligned}[\alpha_i^k, \alpha_j^k] &= \psi_t(f(x_i) \oplus f(x_j); \theta_t) \\ &= \psi_t(f(x_j) \oplus f(x_i); \theta_t). \end{aligned} \tag{3}$$

As can be seen, removing or sharing the projection makes the attention network ψ_t confirm the commutative law of addition. This corrupts the attention weights to be random since an attention weight can correspond to both samples for the following MixUp module.

The effect of GA. An appealing benefit of our GA is that it reduces the impact of noisy-labeled samples significantly. Let N_{noisy} and N_{total} represent the number of the noisy images and total images in a noisy dataset, respectively. The noise ratio is $\frac{N_{noisy}}{N_{total}}$ in the image-wise case. Nevertheless, the number of pure noisy groups (*i.e.* all the images are mislabeled in these groups) in the group-wise case becomes $A^K_{N_{noisy}}$. With $K = 2$, we have,

$$\frac{N_{noisy}}{N_{total}} > \frac{A^2_{N_{noisy}}}{A^2_{N_{total}}} = \frac{N_{noisy}(N_{noisy}-1)}{N_{total}(N_{total}-1)} \approx \frac{N^2_{noisy}}{N^2_{total}}. \tag{4}$$

We argue that GA can reduce the pure noisy ratio dramatically and partial noisy groups (*i.e.* some images within these groups are corrected-labeled) may provides useful supervision by the well-trained attention network. However, though the ratio is smaller when K becomes larger, large K may lead to over-smooth features for the new interpolations which are harmful for discriminative feature learning.

3.3 Mixup Module

The mixup module interpolates virtual feature-target vectors for training. Specifically, following classic mixup vicinal distribution, we normalize the attention weights into range [0, 1]. Formally, for $K = 2$ and group members $\{x_i, x_j\}$, the mixup can be written as follows,

$$x' = \frac{1}{\sum_\alpha}(\alpha_i x_i + \alpha_j x_j), \tag{5}$$

$$y' = \frac{1}{\sum_\alpha}(\alpha_i y_i + \alpha_j y_j), \tag{6}$$

where x' and y' are the interpolated feature and soft label.

3.4 Training and Inference

Training. Our AFM along with the CNN backbone can be trained in an end-to-end manner. Specifically, we conduct a multi-task training scheme to separate the contributions of original training samples and new interpolations. Let f_{c1} and f_{c2} respectively denote the classifiers (include the Softmax or Sigmoid operations) of interpolations and original samples, we can formulate the training loss in a mini batch as follows,

$$\begin{aligned}\mathcal{L}_{total} &= \lambda \mathcal{L}_{afm} + (1-\lambda)\mathcal{L}_{org} \\ &= \frac{\lambda}{m}\sum_{i=1}^{m}\mathcal{L}(f_{c1}(x_i'), y_i') + \frac{(1-\lambda)}{n}\sum_{i=1}^{n}\mathcal{L}(f_{c2}(x_i), y_i),\end{aligned} \tag{7}$$

where n is the batch size, m is the number of interpolations, and λ is a trade-off weight. We use the Cross-Entropy loss function for both L_{afm} and L_{org}. In this way, our AFM can be viewed as a regularizer over the training data by massive interpolations. As proven in [36, 33], this regularizer can largely improve the generalization of deep networks. In addition, the parameters of f_{c1} and f_{c2} can be shared since both original features and interpolations are in same dimensions.

Inference. After training, both the GA module and mixup module can be simply removed since we do not need to compose new samples at test stage. We keep the classifiers f_{c1} and f_{c2} for inference. Particularly, they are identical and we can conduct inference as traditional CNNs if the parameters are shared.

4 Experiments

In this section, we first introduce datasets and implementation details, and then compare our AFM with the state-of-the-art methods. Finally, we conduct ablation studies with qualitative and quantitative results.

4.1 Datasets and Implementation Details

In this paper, we conduct experiments on two popular real-world noisy datasets: Food101N [16] and Clothing1M [35]. **Food101N** consists of 365k images that are crawled from Google, Bing, Yelp, and TripAdvisor using the Food-101 taxonomy. The annotation accuracy is about 80%. The clean dataset Food-101 is collected from *foodspotting.com* which contains 101 food categories with 101,000 real-world food images totally. For each class, 750 images are used for training, the other 250 images for testing. In our experiments, following the common setting, we use all images of Food-101N as the noisy dataset, and report the overall accuracy on the Food-101 test set. **Clothing1M** contains 1 million images of clothes with 14 categories. Since most of the labels are generated by the surrounding text of the images on the Web, a large amount of annotation noises exist, leading to a low annotation accuracy of 61.54% [35]. The human-annotated set of Clothing1M is used as the clean set which is officially divided into training, validation and

Table 1: Comparison with the state-of-the-art methods on Food101N dataset. VF(55k) is the noise-verification set used in CleanNet [16].

Method	Training Data	Training time	Acc
Softmax [16]	Food101	–	81.67
Softmax [16]	Food101N	–	81.44
Weakly Supervised [38]	Food101N	–	83.43
CleanNet(w_{hard}) [16]	Food101N + VF(55K)	–	83.47
CleanNet(w_{soft}) [16]	Food101N + VF(55K)	–	83.95
MetaCleaner [37]	Food101N	–	85.05
SMP [11]	Food101N	–	85.11
ResNet50 (baseline)	Food101N	4h16min40s	84.51
AFM (Ours)	Food101N	4h17min4s	**87.23**

testing sets, containing 50k, 14k and 10k images respectively. We report the overall accuracy on the clean test set of Clothing1M.

Implementation Details As widely used in existing works, ResNet50 is used as our CNN backbone and initialized by the official ImageNet pre-trained model. For each image, we resize the image with a short edge of 256 and random crop 224×224 patch for training. We use SGD optimizer with a momentum of 0.9. The weight decay is 5×10^{-3}, and the batch size is 128. For Food101N, the initial learning rate is 0.001 and divided by 10 every 10 epochs. We stop training after 30 epochs. For Clothing1M, the initial learning rate is 0.001 and divided by 10 every 5 epochs. We stop training after 15 epochs. All the experiments are implemented by Pytorch with 4 NVIDIA V100 GPUs. The default λ and K are 0.75 and 2, respectively. By default, the classifiers f_{c1} and f_{c2} are shared.

4.2 Comparison on Food101N

We compare AFM to the baseline model and existing state-of-the-art methods in Table 1. AFM improves our strong baseline from 84.51% to 87.23%, and consistently outperforms recent state-of-the-art methods with large margins. Moreover, our AFM is almost free since it only increases training time by 24s. Specifically, AFM outperforms [38] by 3.80%, CleanNet(w_{soft}) by 3.28%, and SMP [11] by 2.12%. We notice that, CleanNet(w_{hard}) and CleanNet(w_{soft}) use extra 55k manually-verified images, while we do not use any extra images. In particular, MetaCleaner [37] uses a similar scheme but limited in intra-class mixup and its single feature interaction type, which leads to 2.18% worse than our AFM. An ablation study will further discuss these issues in the following section.

4.3 Comparison on Clothing1M

For the comparison on Clothing1M, we evaluate our AFM in three different settings following [16, 26, 37, 11]: (1) only the noisy set are used for training,

Table 2: Comparison with the state-of-the-art methods on Clothing1M. VF(25k) is the noise-verification set used in CleanNet [16].

Method	Training Data	Acc. (%)
Softmax [16]	1M noisy	68.94
Weakly Supervised [38]	1M noisy	71.36
JointOptim [37]	1M noisy	72.23
MetaCleaner [37]	1M noisy	72.50
SMP (Final)[11]	1M noisy	**74.45**
SMP (Initial) [11]	1M noisy	72.09
AFM (Ours)	1M noisy	**74.12**
CleanNet(w_{hard}) [16]	1M noisy + VF(25K)	74.15
CleanNet(w_{soft}) [16]	1M noisy + VF(25K)	74.69
MetaCleaner [37]	1M noisy + VF(25K)	76.00
SMP [11]	1M noisy + VF(25K)	76.44
AFM (Ours)	1M noisy + VF(25K)	**77.21**
CleanNet(w_{soft}) [16]	1M noisy + Clean(50K)	79.90
MetaCleaner [37]	1M noisy + Clean(50K)	80.78
SMP [11]	1M noisy + Clean(50K)	81.16
CurriculumNet [10]	1M noisy + Clean(50K)	81.50
AFM (Ours)	1M noisy + Clean(50K)	**82.09**

(2) the 25K extra manually-verified images [16] are added into the noisy set for training, and (3) the 50K clean training images are added into the noisy set.

The comparison results are shown in Table 2. For the first setting, our AFM improves the baseline method from 68.94% to 74.12%, and consistently outperforms MetaCleaner, JointOptim, and SMP (Initial) by about 2%. Although SMP (Final) performs on par with AFM in this setting, it needs several training-and-correction loops and careful parameter tuning. Compared to SMP (Final), our AFM is simpler and almost free in computational cost.

For the second setting, other methods except for MetaCleaner mainly apply the 25K verified images to train an accessorial network [16, 26] or to select the class prototypes [11]. Following [37], we train our AFM on 1M noisy training set, and then fine-tune it on the 25K verified images. As shown in Table 2, AFM obtains 77.21% which sets new record in this setting. Specifically, our AFM is better than MetaCleaner and SMP by 1.21% and 0.77%, respectively.

For the third setting, all the methods first train models on the noisy set and then fine-tune them on the clean set. CurriculumNet [10] uses a deeper CNN backbone and obtains accuracy 81.5%, which is slightly better than SMP and other methods. Our AFM outperforms CurriculumNet by 0.59%, and is better than MetaCleaner by 1.31%. It is worth emphasizing that both CurriculumNet and SMP need to train repeatedly after model convergence which are complicated and time-consuming, while AFM is much simpler and almost free.

Table 3: Results of different feature interactions in Group-to-Attend module. *It removes FC_a and FC_b in GA module.

#	Interaction type	Training Data	Acc. (%)
1	Concatenation	Food101N	86.95
2	Concatenation*	Food101N	86.51
3	Sum	Food101N	**87.23**
4	Sum*	Food101N	86.12
5	Multiplication	Food101N	86.64

Table 4: Evaluation of trade-off λ.

λ	0.00	0.25	0.50	0.75	1.00
Acc. (%)	84.51	86.75	86.97	**87.23**	86.47

Table 5: Evaluation of group size.

Size	2	3	4	5	6
Acc. (%)	**87.23**	86.46	86.01	85.92	85.46

4.4 Ablation Study

Evaluation of feature interaction types. *Concatenation*, *sum* and *element-wise multiplication* are three popular feature fusion or interaction methods. MetaCleaner [37] simply takes the *concatenation*, and ignores the impact of the interaction types. We conduct an ablation study for them along with the projection in Group-to-Attend module. Specifically, the group size is set to 2 for this study. Table 3 presents the results on Food101N. Two observations can be concluded as following. First, with FC_a and FC_b, the *sum* interaction consistently performs better than the others. Second, for both *concatenation* and *sum*, it is better to use the projection process. As mentioned in Section 3.2, removing FC_a and FC_b leads to random attention weights for *sum* interaction, which may degrade our AFM to standard Manifold mixup [36]. Nevertheless, it still improves the baseline (*i.e.* 84.51%) slightly.

Evaluation of the trade-off weight λ. In training phase, λ is used to trade-off the loss ratio between $\mathcal{L}_{afm}$ and $\mathcal{L}_{org}$. We evaluate it by increasing λ from 0 to 1 on Food101N, and present the results in Tabel 4. We achieve the best accuracy with default λ (*i.e.* 0.75). Decreasing λ means to use less interpolations from AFM, which gradually degrades the final performance. Particularly, $\lambda = 0$ is our baseline that only uses original noisy training data. In the other extreme case, using only the interpolations from AFM is better than the baseline but slight worse than the default one. This may be explained by that the massive interpolations are more or less smoothed by our AFM since the interpolation weights cannot be zeros due to the GA module. Hence, adding original features can be better since these features fill this gap naturally.

Evaluation of the group size. Our previous experiments fix the group size as 2 which construct pairwise samples for generating virtual feature-target

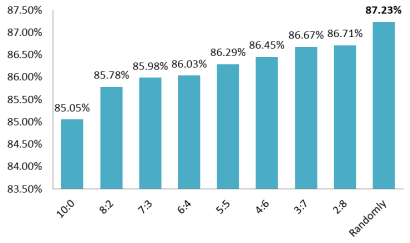

Fig. 3: Evaluation of the ratios of Intra- and Inter-class mixup.

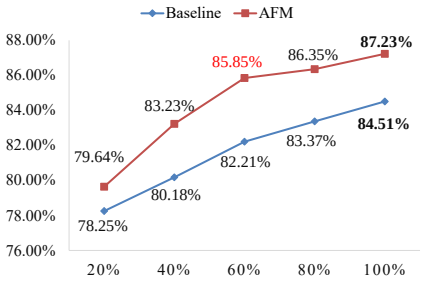

Fig. 4: Evaluation of AFM on synthetic small datasets.

vectors. Here we explore different group sizes for our attentive feature mixup, Specifically, we increase the group size from 2 to 6, and present the results in Table 5. As can be seen, enlarging the group size gradually degrades the final performance. This may be explained by that large group size interpolates over-smoothed features which are not discriminative for any classes.

Intra-class mixup vs. Inter-class mixup. To investigate the contributions of intra-class mixup and inter-class mixup, we conduct an evaluation by exploring different ratios between intra- and inter-class interpolations with group size 2. Specifically, we constrain the number of interpolations for both mixup types in each minibatch with 8 varied ratios from 10:0 to 2:8 on the Food101N dataset. The results are shown in Figure 3. Several observations can be concluded as following. First, removing the inter-class mixup (*i.e.* 10:0) degrades the performance (it is similar with MetaCleaner [37]) while adding a small ratio (*e.g.* 8:2) of inter-class mixup significantly improves the final result. This indicates that the inter-class mixup is more useful for better feature learning. Second, increasing the ratio of inter-class mixup further boosts performance but the performance gaps are small. Third, we get the best result by random selecting group-wise samples. We argue that putting constraints on the ratio of mixup types may result in different data distribution compared to the original dataset while random choice avoids this problem.

AFM for learning from small dataset. Since AFM can generate numerous of noisy-reduced interpolations in training stage, we intuitively check the power of AFM on small datasets. To this end, we construct sub-datasets from Food101N by randomly decreasing the size of Food101N to 80%, 60%, 40%, and 20%. The results of our default AFM on these synthetic datasets are shown in Figure 4. Several observations can be concluded as following. First, our AFM consistently improves the baseline significantly. Second, the improvements from data size 40% to 100% are larger than that of 20%. This may be because that small dataset leads to less diverse interpolations. Third, we interestingly find that our AFM already obtains the state-of-the-art performance with only 60% data on Food101N.

Table 6: Comparison of our AFM with mixup [36] and Manifold mixup [33]. We also evaluate f_{c1} and f_{c2} for them.

Method	$f_{c1} + f_{c2}$	$f_{c1} + f_{c2}$ (Shared)
mixup [36]	85.36%	85.63%
Manifold mixup [33]	85.85%	86.12%
AFM (Ours)	**86.97%**	**87.23%**

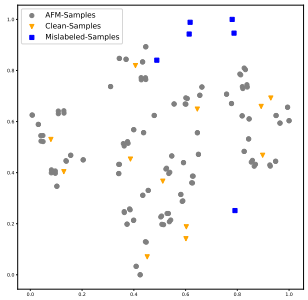

Fig. 5: AFM sample distribution.

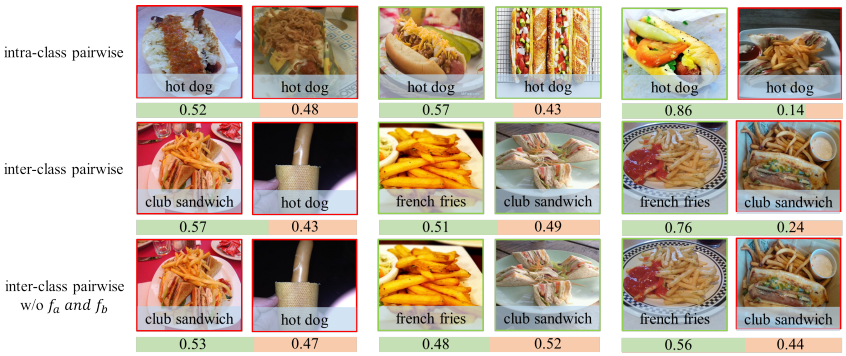

Fig. 6: Visualization of the attention weights in our AFM. The green and red boxes represent the clean and noisy samples.

AFM vs. classic mixup. Since our AFM is related to the mixup scheme, we compare it to the Standard mixup [36] and Manifold mixup [33]. The Standard mixup [36] interpolates samples in image level while Manifold mixup [33] in feature level. Both of them drawl the interpolation weights randomly from a β distribution. Our method introduce a Group-to-Attend (GA) module to generate meaningful weights for noise-robust training. As the new interpolations and the original samples can contribute differently, we separately apply classifiers for them, *i.e.* f_{c1} for interpolations and f_{c2} for original samples. Table 6 presents the comparison. Several observations are concluded as following. First, for both classifier setting, our AFM outperforms the others largely, *e.g.* AFM is better than standard mixup by 1.6% and the Manifold mixup by 1.11% in the shared classifier setting. Second, the shared classifiers are slightly better than the independent classifiers for all methods, which may be explained by that sharing parameters makes the classifier favor linear behavior over all samples thus reducing over-fitting and encouraging the model to discover useful features.

4.5 Visualizations

To better investigate the effectiveness of our AFM, we make two visualizations: i) attentive mixup sample distribution between clean and noisy samples in Figure 5 and ii) the normalized attention weights in Figure 6. For the former, we randomly select several noisy samples and clean samples on the VK(25) set of Food101N and apply our trained AFM model to generate virtual samples (*i.e.* AFM samples), and then use t-SNE to visualize all the real samples and attentive mixup samples. Figure 5 evidently shows that our AFM samples are mainly distributed around the clean samples, demonstrating our AFM suppresses noisy samples effectively. *It is worth noting that classical mixup samples are doomed to distribute around all the real samples rather than only clean samples.*

For the latter visualization, the first row of Figure 6 shows three types of pairs for the intra-class case, the second row for the inter-class case, and the third row for the inter-class case without projection in the Group-to-Attend module. The first column denotes the "noisy+noisy" interpolations, the second column denotes "clean+clean", and the third column denotes "clean+noisy". Several finds can be observed as following. First, for both intra- and inter-class cases, the weights of "noisy+noisy" and "clean+clean" interpolations trend to be equal since these interpolations may lie in the decision boundaries which make the network hard to identify which is better for training. Second, for the "clean+noisy" interpolations on the first two rows, our AFM assigns evidently low weights to these noisy samples which demonstrates the effectiveness of AFM. Last, without projection in the Group-to-Attend module, our default AFM loses the ability to identify noisy samples as shown in the last image pair.

5 Conclusion

This paper proposed a conceptually simple yet efficient training block, termed as Attentive Feature Mixup (AFM), to address the problem of learning with noisy labeled data. Specifically, AFM is a plug-and-play training block, which mainly leverages grouping and self-attention to suppress mislabeled data and does not rely on any assumptions and extra clean subset. We conducted extensive experiments on two challenging real-world noisy datasets: Food101N and Clothing1M. Quantitative and qualitive results demonstrated that our AFM is superior to recent state-of-the-art methods. In addition, the grouping and self-attention is expected to extend in other topics, *e.g.* semi-supervised learning, where one may conduct this module for real annotations and pseudo labels to automatically suppress incorrect pseudo labels.

Acknowledge. This work is partially supported by National Key Research and Development Program of China (No. 2020YFC2004800), National Natural Science Foundation of China (U1813218, U1713208), Science and Technology Service Network Initiative of Chinese Academy of Sciences (KFJ-STS-QYZX-092), Guangdong Special Support Program (2016TX03X276), and Shenzhen Basic Research Program (JSGG20180507182100698, CXB201104220032A), Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Unsupervised label noise modeling and loss correction (2019)
2. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 233–242. JMLR. org (2017)
3. Barandela, R., Gasca, E.: Decontamination of training samples for supervised pattern recognition methods. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). pp. 621–630. Springer (2000)
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: ICML. pp. 41–48. ACM (2009)
5. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of artificial intelligence research **11**, 131–167 (1999)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
7. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. TNNLS **25**(5), 845–869 (2014)
8. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A multi-view embedding space for modeling internet images, tags, and their semantics. IJCV **106**(2), 210–233 (2014)
9. Guo, H., Mao, Y., Zhang, R.: Mixup as locally linear out-of-manifold regularization. In: AAAI. vol. 33, pp. 3714–3722 (2019)
10. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculumnet: Weakly supervised learning from large-scale web images. In: ECCV. pp. 135–150 (2018)
11. Han, J., Luo, P., Wang, X.: Deep self-learning from noisy labels. In: ICCV (2019)
12. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Regularizing very deep neural networks on corrupted labels. arXiv preprint arXiv:1712.05055 (2017)
13. Joulin, A., van der Maaten, L., Jabri, A., Vasilache, N.: Learning visual features from large weakly supervised data. In: ECCV. pp. 67–84. Springer (2016)
14. Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., Fei-Fei, L.: The unreasonable effectiveness of noisy data for fine-grained recognition. In: ECCV. pp. 301–320. Springer (2016)
15. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. arXiv preprint arXiv:1711.07131 (2017)
16. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In: CVPR. pp. 5447–5456 (2018)
17. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: CVPR (June 2019)
18. Li, Q., Peng, X., Cao, L., Du, W., Xing, H., Qiao, Y.: Product image recognition with guidance learning and noisy supervision. Comput. Vis. Image Underst. **196**, 102963 (2020)
19. Li, W., Wang, L., Li, W., Agustsson, E., Van Gool, L.: Webvision database: Visual learning and understanding from web data. arXiv preprint arXiv:1708.02862 (2017)
20. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., Li, L.J.: Learning from noisy labels with distillation. In: ICCV. pp. 1928–1936 (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)

22. Mai, Z., Hu, G., Chen, D., Shen, F., Shen, H.T.: Metamixup: Learning adaptive interpolation policy of mixup with meta-learning. arXiv preprint arXiv:1908.10059 (2019)
23. Manwani, N., Sastry, P.: Noise tolerance under risk minimization. IEEE transactions on cybernetics **43**(3), 1146–1151 (2013)
24. Miranda, A.L., Garcia, L.P.F., Carvalho, A.C., Lorena, A.C.: Use of classification algorithms in noise detection and elimination. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 417–424. Springer (2009)
25. Misra, I., Lawrence Zitnick, C., Mitchell, M., Girshick, R.: Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In: CVPR. pp. 2930–2939 (2016)
26. Patrini, G., Rozza, A., Menon, A.K., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: CVPR. pp. 2233–2241 (2017)
27. Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596 (2014)
28. Rolnick, D., Veit, A., Belongie, S., Shavit, N.: Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694 (2017)
29. Schmidt, R.A., Bjork, R.A.: New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. Psychological science **3**(4), 207–218 (1992)
30. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080 (2014)
31. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. arXiv preprint arXiv:1803.11364 (2018)
32. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.J.: Learning from noisy large-scale datasets with minimal supervision. In: CVPR. pp. 6575–6583 (2017)
33. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states pp. 6438–6447 (2019)
34. Wang, K., Peng, X., Yang, J., Lu, S., Qiao, Y.: Suppressing uncertainties for large-scale facial expression recognition. In: CVPR (June 2020)
35. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: CVPR. pp. 2691–2699 (2015)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
37. Zhang, W., Wang, Y., Qiao, Y.: Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In: CVPR. pp. 7373–7382 (2019)
38. Zhuang, B., Liu, L., Li, Y., Shen, C., Reid, I.: Attend in groups: a weakly-supervised deep learning framework for learning from web data. In: CVPR. pp. 1878–1887 (2017)