Appendix

In this Appendix, we provide proofs of proposition 1 (§A.1) and proposition 2 (§A.2), implementation details (§B), and more experiment results (§C).

A Proof of Proposition

A.1 Proof of Proposition 1

Proposition 1. Assuming $x_{(a,b)}$ is generated from two different classes, minimizing $\mathcal{L}_{MCE}$ is equivalent to regress corresponding λ in the gradient of $\mathcal{L}_{MCE}$:

$$(\nabla_{z_{(a,b)}}\mathcal{L}_{MCE})^i = \begin{cases} -\lambda + \frac{\exp(z^i_{(a,b)})}{\sum_c \exp(z^c_{(a,b)})}, & l = i \\ -(1-\lambda) + \frac{\exp(z^j_{(a,b)})}{\sum_c \exp(z^c_{(a,b)})}, & l = j \\ \frac{\exp(z^i_{(a,b)})}{\sum_c \exp(z^c_{(a,b)})}, & l \neq i, j \end{cases} \tag{7}$$

Proof. For the mixed sample $(x_{(a,b)}, y_{(a,b)})$, $z_{(a,b)}$ is derived from a feature extractor f_θ (i.e $z_{(a,b)} = f_\theta(x_{(a,b)})$). According to the definition of the mixup cross-entropy loss $\mathcal{L}_{MCE}$, we have:

$$\begin{aligned} \big(\nabla_{z(a,b)}\mathcal{L}_{MCE}\big)^l &= \frac{\partial \mathcal{L}_{MCE}}{\partial z^l_{(a,b)}} - \frac{\partial}{\partial z^l_{(a,b)}}\Big(y^T_{(a,b)} \log\big(\sigma(z_{(a,b)})\big)\Big) \\ &= -\sum_{i=1}^{C}\Big(y^i_{(a,b)} \frac{\partial}{\partial z^l_{(a,b)}}\big(\log(\frac{\exp(z^i_{(a,b)})}{\sum_{j=1}^{C}\exp(z^j_{(a,b)})})\big)\Big) \\ &= -\sum_{i=1}^{C}\Big(y^i_{(a,b)} \frac{\sum_{j=1}^{C}\exp(z^j_{(a,b)})}{\exp(z^i_{(a,b)})} \frac{\partial}{\partial z^l_{(a,b)}}\big(\frac{\exp(z^i_{(a,b)})}{\sum_{j=1}^{C}\exp(z^j_{(a,b)})}\big)\Big) \\ &= -\sum_{i=1}^{C}\Big(y^i_{(a,b)}\big(\delta^l_i - \frac{\exp(z^l_{(a,b)})}{\sum_{j=1}^{C}\exp(z^j_{(a,b)})}\big)\Big) \\ &= \frac{\exp(z^l_{(a,b)})}{\sum_{j=1}^{C}\exp(z^j_{(a,b)})} - y^l_{(a,b)}. \quad \square \end{aligned}$$

A.2 Proof of Proposition 2

Proposition 2. With the decoupled Softmax defined above, decoupled mixup cross-entropy $\mathcal{L}_{DM(CE)}$ can boost the prediction confidence of the interested classes mutually and escape from the λ-constraint:

$$\mathcal{L}_{DM(CE)} = \sum_{i=1}^{c}\sum_{j=1}^{c} y_a^i y_b^j \Big(\log\big(\frac{p^i_{(a,b)}}{1-p^j_{(a,b)}}\big) + \log\big(\frac{p^j_{(a,b)}}{1-p^i_{(a,b)}}\big)\Big).$$

Proof. For the mixed sample $(x_{(a,b)}, y_{(a,b)})$, $z_{(a,b)}$ is derived from a feature extractor f_θ (i.e $z_{(a,b)=f_\theta(x_{(a,b)})}$). According to the definition of the mixup cross-entropy loss $\mathcal{L}_{DM(CE)}$, we have:

$$\begin{aligned}
\mathcal{L}_{DM(CE)} &= y_{[a,b]}^T \log\big(H(Z_{(a,b)})\big) y_{[a,b]} \\
&\triangleq y_a^T \log\big(H(Z_{(a,b)})\big) y_b + y_b^T \log\big(H(Z_{(a,b)})\big) y_a \\
&= \sum_{i,j=1}^{C} y_a^i \log\big(\frac{\exp(z^i_{(a,b)})}{\sum_{k\neq j}^{C}\exp(z^j_{(a,b)})}\big) y_b^j + \sum_{i,j=1}^{C} y_a^j \log\big(\frac{\exp(z^j_{(a,b)})}{\sum_{k\neq i}^{C}\exp(z^i_{(a,b)})}\big) y_b^i \\
&= \sum_{i,j=1}^{C} y_a^i y_b^j \big(\log\big(\frac{\exp(z^i_{(a,b)})}{\sum_{k\neq j}^{C}\exp(z^j_{(a,b)})}\big) + \log\big(\frac{\exp(z^j_{(a,b)})}{\sum_{k\neq i}^{C}\exp(z^i_{(a,b)})}\big)\big) \\
&= \sum_{i,j=1}^{C} y_a^i y_b^j \big(\log\big(\frac{\frac{\exp(z^i_{(a,b)})}{\sum_{k=1}^{C}\exp(z^k_{(a,b)})}}{\frac{\sum_{k\neq j}^{C}\exp(z^j_{(a,b)})}{\sum_{k=1}^{C}\exp(z^k_{(a,b)})}}\big) + \log\big(\frac{\frac{\exp(z^j_{(a,b)})}{\sum_{k=1}^{C}\exp(z^k_{(a,b)})}}{\frac{\sum_{k\neq i}^{C}\exp(z^i_{(a,b)})}{\sum_{k=1}^{C}\exp(z^k_{(a,b)})}}\big)\big) \\
&= \sum_{i,j=1}^{C} y_a^i y_b^j \big(\log\big(\frac{p^i_{(a,b)}}{1-p^j_{(a,b)}}\big) + \log\big(\frac{p^j_{(a,b)}}{1-p^i_{(a,b)}}\big)\big),
\end{aligned}$$

where $p_{(a,b)} = \sigma(z_{(a,b)})$. □

B Implementation Details

B.1 Dataset

We briefly introduce used image datasets. (1) Small scale classification benchmarks: CIFAR-10/100 [19] contains 50,000 training images and 10,000 test images in 32×32 resolutions, with 10 and 100 classes settings. Tiny-ImageNet [5] is a rescaled version of ImageNet-1k, which has 10,000 training images and 10,000 validation images of 200 classes in 64×64 resolutions. (2) Large scale classification benchmarks: ImageNet-1k [20] contrains 1,281,167 training images and 50,000 validation images of 1000 classes in 224×224 resolutions. (3) Small-scale fine-grained classification scenarios: CUB-200-2011 [48] contains 11,788 images from 200 wild bird species for fine-grained classification. FGVC-Aircraft [33] contains 10,000 images of 100 classes of aircraft. Standford-Cars [18].

B.2 Training Settings

Small-scale image classification. As for small-scale classification benchmarks on CIFAR-100 and Tiny-ImageNet datasets, we adopt the CIFAR version of ResNet variants, *i.e.*, using a 3×3 convolution instead of the 7×7 convolution and MaxPooling in the stem, and follow the common training settings [17, 29]: the basic data augmentation includes `RandomFlip` and `RandomCrop` with 4 pixels padding; SGD optimizer and Cosine learning rate Scheduler [30] are used with the SGD weight decay of 0.0001, the momentum of 0.9, and the Batch size of 100; all methods train 800 epochs with the basic learning rate $lr = 0.1$ on CIFAR-100 and 400 epochs with $lr = 0.2$ on Tiny-ImageNet.

Fine-grained image classification. As for fine-grained classification experiments on CUB-200 and Aircraft datasets, all mixup methods are trained 200 epochs by SGD optimizer with the initial learning rate $lr = 0.001$, the weight decay of 0.0005, and the batch size of 16. We use the standard augmentations `RandomFlip` and `RandomResizedCrop`, and load the official PyTorch pre-trained models on ImageNet-1k as initialization.

ImageNet image classification. For large-scale classification tasks on ImageNet-1k, we evaluate mixup methods on three popular training procedures, and Tab. A1 shows the full training settings of the three settings. Notice that DeiT [44] and RSB A3 [51] settings employ Mixup and CutMix with a switching probability of 0.5 during training. (a) PyTorch-style setting. Without any advanced training strategies, a PyTorch-style setting is used to study the performance gains of mixup methods: SGD optimizer is used to train 100 epochs with the SGD weight decay of 0.0001, a momentum of 0.9, a batch size of 256, and the basic learning rate of 0.1 adjusted by Cosine Scheduler. Notice that we replace the step learning rate decay with Cosine Scheduler [30] for better performances following [60]. (b) DeiT [44] setting. We use the DeiT setting to verify the DM(CE) effectiveness in training Transformer-based networks: AdamW optimizer [32] is used to train 300 epochs with a batch size of 1024, the basic learning rate of 0.001, and the weight decay of 0.05. (c) RSB A3 [51] setting. This setting adopts similar training techniques as DeiT to ConvNets, *especially using MBCE instead of MCE*: LAMB optimizer [58] is used to train 100 epochs with the batch size of 2048, the basic learning rate of 0.008, and the weight decay of 0.02. Notice that DeiT and RSB A3 settings use the combination of Mixup and CutMix (50% random switching probabilities) as the baseline.

Semi-supervised transfer learning. For semi-supervised transfer learning benchmarks, we use the same hyper-parameters and augmentations as Self-Tuning[2]: all methods are initialized by PyTorch pre-trained models on ImageNet-1k and trained 27k steps in total by SGD optimizer with the basic learning rate of 0.001, the momentum of 0.9, and the weight decay of 0.0005. We reproduced Self-Tuning and conducted all experiments in OpenMixup [25].

Semi-supervised learning. For semi-supervised learning benchmarks (training from scratch), we adopt the most commonly used CIFAR-10/100 datasets among the famous SSL benchmarks based on WRN-28-2 and WRN-28-8 following [42, 62]. For a fair comparison, we use the same hyperparameters and training settings as the original papers and adopt the open-source codebase TorchSSL [62] for all methods. Concretely, we use an SGD optimizer with a basic learning rate of

[2] `https://github.com/thuml/Self-Tuning`

Table A1: Ingredients and hyper-parameters used for ImageNet-1k training settings.

Procedure	PyTorch	DeiT	RSB A3
Train Res	224^2	224^2	224^2
Test Res	224^2	224^2	224^2
Test crop ratio	0.875	0.875	0.95
Epochs	100/300	300	100
Batch size	256	1024	2048
Optimizer	SGD	AdamW	LAMB
LR	0.1	1×10^{-3}	8×10^{-3}
LR decay	cosine	cosine	cosine
Weight decay	10^{-4}	0.05	0.02
optimizer momentum	0.9	$\beta_1, \beta_2 = 0.9, 0.999$	✗
Warmup epochs	✗	5	5
Label smoothing ϵ	✗	0.1	✗
Dropout	✗	✗	✗
Stoch. Depth	✗	0.1	0.05
Repeated Aug	✗	✓	✓
Gradient Clip.	✗	1.0	✗
H. flip	✓	✓	✓
RRC	✓	✓	✓
Rand Augment	✗	9/0.5	6/0.5
Auto Augment	✗	✗	✗
Mixup alpha	✗	0.8	0.1
Cutmix alpha	✗	1.0	1.0
Erasing prob.	✗	0.25	✗
ColorJitter	✗	✗	✗
EMA	✗	0.99996	✗
CE loss	✓	✓	✗
BCE loss	✗	✗	✓

$lr = 0.03$ adjusted by Cosine Scheduler, the total 2^{20} steps, the batch size of 64 for labeled data, and the confidence threshold $\tau = 0.95$.

B.3 Hyper-parameter Settings

We follow the basic hyper-parameter settings (*e.g.,* α) for mixup variants in OpenMixup [25], where we reproduce most comparison methods. Notice that *static* methods denote Mixup [63], CutMix [60], ManifoldMix [46], SaliencyMix [45], FMix [11], ResizeMix [37], and *dynamic* methods denote PuzzleMix [17], AutoMix [29], and SAMix [24]). Similarly, *interpolation-based* methods denote Mixup and ManifoldMix while *cutting-based* methods denote the rest mixup variants mentioned above. We set the hyper-parameters of DM(CE) as follows: For CIFAR-100 and ImageNet-1k, *static* methods use $\eta = 0.1$, and *dynamic* methods use $\eta = 1$. For Tiny-ImageNet and fine-grained datasets, *static* methods use $\eta = 1$ based on ResNet-18 while $\eta = 0.1$ based on ResNeXt-50; *dynamic* methods use $\eta = 1$. As for the hyper-parameters of DM(BCE) on ImageNet-1k, *cutting-based* methods use $t = 1$ and $\xi = 0.8$, while *interpolation-based* methods use $t = 0.5$ and $\xi = 1$. Note that we use $\alpha = 0.2$ and $\alpha = 2$ for the *static* and *dynamic* methods when using the proposed DM.

Table A2: Top-1 Acc (%)↑ of small-scale image classification on CIFAR-100 and Tiny-ImageNet datasets based on ResNet variants.

Datasets	CIFAR-100						Tiny-ImageNet			
	R-18		RX-50		WRN-28-8		R-18		RX-50	
Methods	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**
SaliencyMix	79.12	**79.28**	81.53	**82.61**	84.35	**84.41**	64.60	**66.56**	66.55	**67.52**
PuzzleMix	81.13	**81.34**	82.85	**82.97**	85.02	**85.25**	65.81	**66.52**	67.83	**68.04**
AutoMix	82.04	**82.32**	83.64	**83.94**	85.18	**85.38**	67.33	**68.18**	70.72	**71.56**
SAMix	82.30	**82.40**	84.42	**84.53**	85.50	**85.59**	68.89	**69.16**	72.18	**72.39**
Avg. Gain		**+0.19**		**+0.40**		**+0.15**		**+0.95**		**+0.56**

Table A3: Top-1 Acc (%)↑ of image classification on ImageNet-1k with ResNet variants using PyTorch-style 100-epoch training recipe.

Methods	R-18		R-34		R-50	
	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**
SaliencyMix	69.16	**69.57**	73.56	**73.92**	77.14	**77.42**
PuzzleMix	70.12	**70.32**	74.26	**74.51**	77.54	**77.71**
AutoMix	70.51	**70.64**	74.52	**74.77**	77.91	**78.15**
SAMix	70.85	**70.90**	74.96	**75.10**	78.11	**78.36**
Avg. Gain		**+0.20**		**+0.25**		**+0.23**

Table A4: Top-1 Acc (%)↑ of image classification on ImageNet-1k based on ResNet-50 using RSB A3 100-epoch training recipe.

Methods	MCE	**DM(CE)**	MBCE (one)	MBCE (two)	**DM(BCE) (one)**
SaliencyMix	76.85	**77.25**	77.93	72.74	**78.24**
PuzzleMix	77.27	**77.60**	78.02	77.19	**78.15**
AutoMix	77.45	**77.82**	78.33	77.46	**78.62**
SAMix	78.33	**78.45**	78.64	77.58	**78.75**
Avg. Gain		**+0.30**		-1.99	**+0.04**

Table A5: Top-1 Acc (%)↑ of classification on ImageNet-1k with ViTs.

Methods	DeiT-S		Swin-T	
	MCE	**DM(CE)**	MCE	**DM(CE)**
DeiT	79.80	**80.37**	81.28	**81.49**
SaliencyMix	79.32	**79.86**	80.68	**80.83**
PuzzleMix	79.84	**80.25**	81.03	**81.16**
AutoMix	80.78	**80.91**	81.80	**81.92**
SAMix	80.94	**81.12**	81.87	**81.97**
Avg. Gain		**+0.32**		**+0.13**

Table A6: Top-1 Acc (%)↑ of fine-grained image classification on CUB-200 and FGVC-Aircrafts with ResNet variants.

Datasets	CUB-200				FGVC-Aircrafts			
	R-18		RX-50		R-18		RX-50	
Methods	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**
SaliencyMix	77.95	**78.28**	83.29	**84.51**	80.02	**81.31**	84.31	**85.07**
PuzzleMix	78.63	**78.74**	84.51	**84.67**	80.76	**80.89**	86.23	**86.36**
AutoMix	79.87	**81.08**	86.56	**86.74**	81.37	**82.18**	86.69	**86.82**
SAMix	81.11	**81.27**	86.83	**86.95**	82.15	**83.68**	86.80	**87.22**
Avg. Gain		**+0.45**		**+0.42**		**+0.94**		**+0.36**

C More Experiment Results

C.1 Image Classification Benchmarks

Small-scale classification benchmarks. For small-scale classification benchmarks on CIFAR-100 and Tiny-ImageNet, we also conduct experiments of applying the proposed DM(CE) to *dynamic* mixup methods even though these algorithms have achieved high performance in Table A2: DM(CE) brings 0.23%∼0.36% on CIFAR-100 for the previous state-of-the-art PuzzleMix and brings 0.21%∼0.27% on Tiny-ImageNet for the current state-of-the-art method SAMix. Overall, the proposed DM(CE) produces +0.15∼0.4% and 0.56∼0.95% average gains on CIFAR-100 and Tiny-ImageNet, demonstrating its generalizability to advanced mixup augmentations.

ImageNet and fine-grained classification benchmarks. For experiments on ImageNet-1k, we also employ the proposed DM(CE) to *dynamic* mixup approaches on ImageNet-1k with PyTorch-style [13], DeiT [44], and RSB A3 [51] training settings to further evaluate the generalizability of decoupled mixup. As shown in Table A3 and Table A4, DM(CE) gains +0.2∼0.3% top-1 accuracy over MCE in average for four *dynamic* mixup methods based on ResNet variants on ImageNet-1k; Table A5 show DM(CE) also improves *dynamic* methods based on popular DeiT-S and Swin-T backbones with modern training recipes. These results indicate that the proposed decoupled mixup can also boost these *dynamic* mixup augmentations with high performances on ImageNet-1k. Moreover, the proposed DM(CE) can improve *dynamic* mixup variants on fine-grained classification benchmarks, as shown in Table A6, with around +0.4∼0.9% average gains over MCE based on ResNet variants.

Table A7: Top-1 Acc (%)↑ and FGSM error (%)↓ on CIFAR-100 and Tiny-ImageNet based on ResNet-18 training 400 epochs.

Datasets	CIFAR-100				Tiny-ImageNet			
	Acc(%)↑		Error(%)↓		Acc(%)↑		Error(%)↓	
Methods	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**	MCE	**DM(CE)**
Mixup	79.34	**79.70**	70.28	**70.05**	63.86	**65.07**	89.06	**88.91**
CutMix	79.58	**79.77**	87.43	**86.84**	65.53	**66.45**	89.14	**88.79**
ManifoldMix	80.18	**81.06**	72.50	**72.19**	64.15	**65.45**	88.78	**88.52**
PuzzleMix	80.22	**80.58**	79.76	**79.53**	65.81	**66.13**	**91.83**	92.05
AutoMix*	81.78	**81.96**	69.94	**69.80**	67.33	**68.18**	88.37	**88.34**

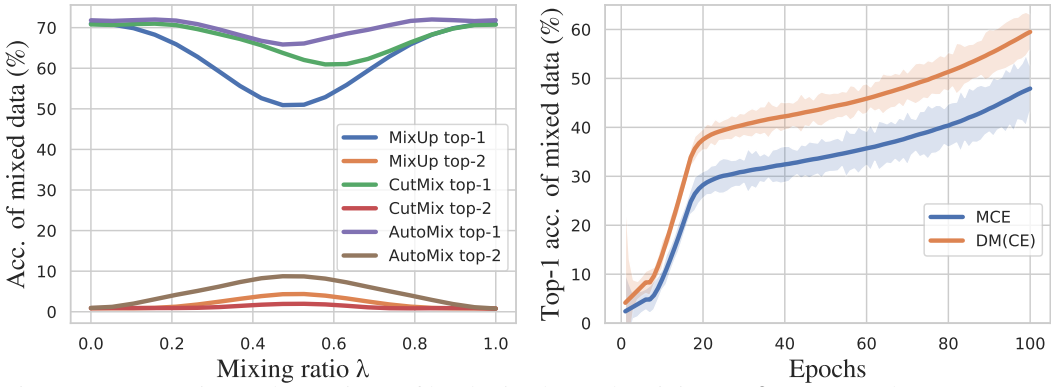

Figure A1: Experimental overviews of hard mixed sample mining. **Left**: Top-1 and top-2 accuracy of mixed data based on ResNet-50 trained 100 epochs on ImageNet-1k. Prediction is counted as correct if the top-1 prediction belongs to $\{y_a, y_b\}$; prediction is counted as correct if the top-2 predictions are equal to $\{y_a, y_b\}$. Compared with *static* policies like Mixup [63] and CutMix [60], the *dynamic* method AutoMix [29] significantly reduces the difficulty of mixup classification and alleviates the label mismatch issue [17] by providing more reliable mixed samples but also requires a large computational overhead. **Right**: Taking Mixup as an example, our proposed decoupled mixup cross-entropy, DM(CE), significantly improves training efficiency by exploring hard mixed samples and alleviates the label mismatch issue.

C.2 Adversarial Robustness

Since mixup variants are proven to enhance the robustness of DNNs against adversarial samples [63], we compare the robustness of the original MCE and the proposed DM(CE) by performing the FGSM [9] white-box attack of 8/255 ℓ_∞ epsilon ball following [17]. Table A7 shows that DM(CE) improves top-1 Acc of MCE while maintaining the competitive FGSM error rates for five popular mixup algorithms, which indicates that DM(CE) can *boost discrimination without disturbing the smoothness properties* of mixup variants.

C.3 Data-efficient Mixup with Limited Training Labels

To further DM whether data-efficient mixup training can be truly achieved, we conducted supervised experiments on CIFAR-100 with different sizes of training data. 15%, 30%, and 50% of the CIFAR-100 data are randomly selected as training data, and the test data are unchanged. The proposed decoupled mixup uses DM(CE) as the loss function by default. From Table A8, we can see that DM improves performance consistently without any computational overhead. Especially when using only 15% of the data, DM can improve accuracy by 2%. Therefore, combined with the experimental results of semi-supervised learning in Sec. 5.3 and Sec. 5.2, we can say that mixup training with DM is more data-efficient with limited data.

Table A8: Top-1 Acc (%)↑ of image classification on CIFAR-100 with ResNet-18 using 15%, 30%, and 50% labeled training sets.

	15%		30%		50%	
Methods	MCE	DM(CE)	MCE	DM(CE)	MCE	DM(CE)
Vanilla	42.48	-	56.41	-	64.32	-
Mixup	42.23	**44.39**	55.61	**56.78**	64.55	**65.92**
CutMix	43.81	**44.85**	55.99	**57.14**	64.38	**65.87**
SaliencyMix	42.95	**44.01**	55.42	**56.51**	64.56	**66.10**
PuzzleMix	42.67	**43.87**	56.19	**57.36**	64.74	**66.26**
Avg. Gain		**+1.36**		**+1.14**		**+1.48**

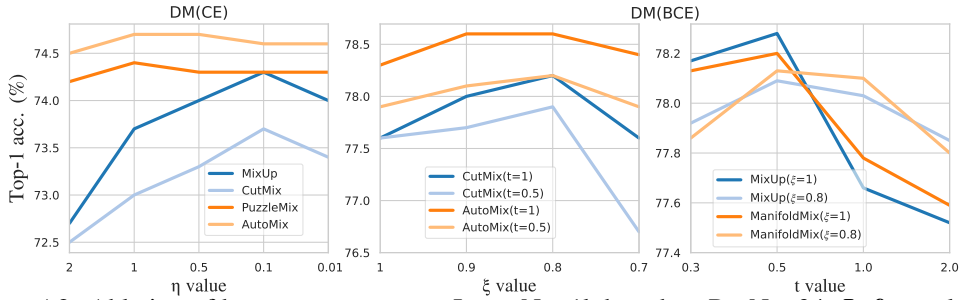

Figure A2: Ablation of hyper-parameters on ImageNet-1k based on ResNet-34. **Left**: analyzing the balancing weight η in DM(CE); **Middle**: analyzing ξ in DM(BCE) when t is fixed to 1 and 0.5; **Right**: analyzing t in DM(BCE) when ξ is fixed to 1 and 0.8.

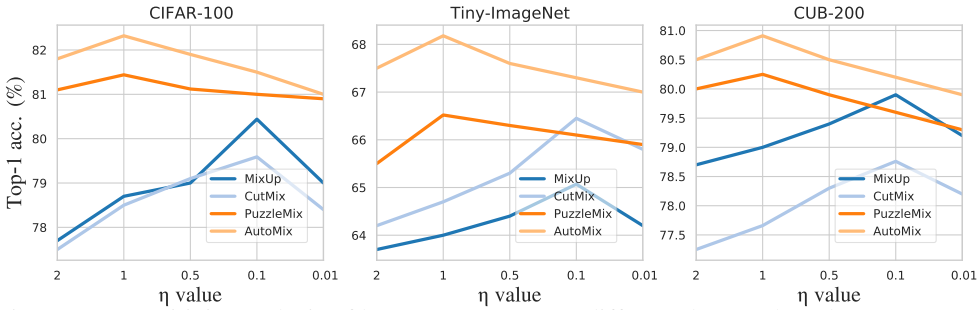

Figure A3: Sensitivity analysis of hyper-parameters on different datasets based on ResNet-18.

C.4 Empirical Analysis

In addition to occlusion robustness in Figure 6, we analyze the top-1 and top-2 mixup classification accuracy and visualize validation accuracy curves during training to empirically demonstrate the effectiveness of DM in Figure A1.

C.5 Ablation Study and Analysis

Ablation of hyper-parameters We first provide ablation experiments of the shared hyper-parameter η in DM(CE) and DM(BCE). In Figure A2 *left*, the *static* (Mixup and CutMix) and the *dynamic* methods (PuzzleMix and AutoMix) prefer $\eta = 0.1$ and $\eta = 1$, respectively, which might be because the *dynamic* variants generate more discriminative and reliable mixed samples than the *static* methods. Then, Figure A2 *middle* and *right* show that ablation studies of hyper-parameters ξ and t in DM(BCE), where cutting-based methods (CutMix and AutoMix) prefer $\xi = 0.8$ and $t = 1$, while the interpolation-based policies (Mixup and ManifoldMix) use $\xi = 1.0$ and $t = 0.5$.

Sensitivity Analysis To verify the robustness of hyper-parameter η, extra experiments are conducted on CIFAR-100, Tiny-ImageNet, and CUB-200 datasets. Figure A3 shows the results consistent with our ablation study in Sec. 5.4. *Dynamic* mixup methods prefer the large value of η (*e.g.,* 1.0), while *static* ones are more like a small value (*e.g.,* 0.1). The main reason for this is the *dynamic* methods generate mixed samples where label mismatch is relatively rare, relying on larger weights to achieve better results, while the opposite is true in *static* methods.

References

[1] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] Jie-Neng Chen, Shuyang Sun, Ju He, Philip Torr, Alan Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[6] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M Nasrabadi. Supermix: Supervising the mixing data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13794–13803, 2021.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 3714–3722, 2019.

[11] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, and Adam Prügel-Bennett Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2(3):4, 2020.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[15] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[16] Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations (ICLR)*, 2021.

[17] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning (ICML)*, pages 5275–5285. PMLR, 2020.

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NeurIPS)*, pages 1097–1105, 2012.

[21] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896, 2013.

[22] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. I-mix: A domain-agnostic strategy for contrastive representation learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[23] Junnan Li, Caiming Xiong, and Steven Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[24] Siyuan Li, Zicheng Liu, Di Wu, Zihan Liu, and Stan Z. Li. Boosting discriminative visual representation learning with scenario-agnostic mixup. *arXiv preprint arXiv:2111.15454*, 2021.

[25] Siyuan Li, Zedong Wang, Zicheng Liu, Di Wu, and Stan Z. Li. Openmixup: Open mixup toolbox and benchmark for visual representation learning. `https://github.com/Westlake-AI/openmixup`, 2022.

[26] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[27] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021.

[29] Zicheng Liu, Siyuan Li, Di Wu, Zhiyuan Chen, Lirong Wu, Liu Zihan, and Stan Z Li. Automix: Unveiling the power of mixup for stronger classifier. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022.

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[34] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[35] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[36] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. *arXiv preprint arXiv:2206.14502*, 2022.

[37] Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. Resizemix: Mixing data with preserved object information and true labels. *arXiv preprint arXiv:2012.11101*, 2020.

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

[39] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2019.

[40] Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Unmix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[41] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[42] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research (JMLR)*, 15(1):1929–1958, 2014.

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pages 10347–10357, 2021.

[45] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020.

[46] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, pages 6438–6447. PMLR, 2019.

[47] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning (ICML)*, pages 10530–10541. PMLR, 2021.

[48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[49] Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3642–3646, 2020.

[50] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning (ICML)*, pages 1058–1066. PMLR, 2013.

[51] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm, 2021.

[52] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[53] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1492–1500, 2017.

[54] Wang Ximei, Gao Jinghan, Long Mingsheng, and Wang Jianmin. Self-tuning for data-efficient deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[55] Chen Xinyang, Wang Sinan, Fu Bo, Long Mingsheng, and Wang Jianmin. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[56] Huaxiu Yao, Yiping Wang, Linjun Zhang, James Y Zou, and Chelsea Finn. C-mixup: Improving generalization in regression. *Advances in Neural Information Processing Systems*, 35:3361–3376, 2022.

[57] Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020.

[59] Hao Yu, Huanyu Wang, and Jianxin Wu. Mixup without hesitation. In *International Conference on Image and Graphics (ICIG)*, pages 143–154. Springer, 2021.

[60] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.

[61] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[62] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[63] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.