IPMix: Label-Preserving Data Augmentation Method for Training Robust Classifiers

Zhenglin Huang[1], **Xianan Bao**[1], **Na Zhang**[1]*, **Qingqi Zhang**[2],
Xiaomei Tu[3], **Biao Wu**[1], and **Xi Yang**[4]

[1]School of Artificial Intellienge, Zhejiang Sci-Tech University
[2]Yamaguchi University [3]ZGUC
[4]University of Science and Technology of China, Hefei, China

Abstract

Data augmentation has been proven effective for training high-accuracy convolutional neural network classifiers by preventing overfitting. However, building deep neural networks in real-world scenarios requires not only high accuracy on clean data but also robustness when data distributions shift. While prior methods have proposed that there is a trade-off between accuracy and robustness, we propose IPMix, a simple data augmentation approach to improve robustness without hurting clean accuracy. IPMix integrates three levels of data augmentation (image-level, patch-level, and pixel-level) into a coherent and label-preserving technique to increase the diversity of training data with limited computational overhead. To further improve the robustness, IPMix introduces structural complexity at different levels to generate more diverse images and adopts the random mixing method for multi-scale information fusion. Experiments demonstrate that IPMix outperforms state-of-the-art corruption robustness on CIFAR-C and ImageNet-C. In addition, we show that IPMix also significantly improves the other safety measures, including robustness to adversarial perturbations, calibration, prediction consistency, and anomaly detection, achieving state-of-the-art or comparable results on several benchmarks, including ImageNet-R, ImageNet-A, and ImageNet-O.

1 Introduction

Deep neural network models have recently achieved remarkable performance on various computer vision tasks, such as zero-shot image classification [79, 88, 22], 3D object detection [52, 85, 51], and face recognition [75, 76]. In real-world scenarios, models can achieve impressive accuracy when training and test distributions are identical, but challenges appear when confronted with out-of-distribution examples [61, 9, 27], such as natural corruptions [54], adversarial perturbations [53], and anomaly patterns [17], necessitating robustness across distribution shifts. Data augmentation (DA) has been proposed to partially alleviate this issue, which applies diverse transformations on clean images to generate new training examples [21, 6]. Furthermore, a high diversity of augmented images enables neural networks to resist data distribution shifts and improve robustness [68]. DA approaches generally fall into three subgroups: image-level, patch-level, and pixel-level augmentations.

Image-level augmentation techniques [45, 10, 11] apply transformations on the whole image, such as brightness, sharpness, and solarization, to increase the total amount of training data. Patch-level augmentation techniques [70, 38] typically mask or replace a region of an image, compelling classifiers to focus on less discriminative portions. Meanwhile, pixel-level augmentation techniques [90, 33] mix images using pixel-wise weighted averages to increase diversity within the training dataset.

*Corresponding author

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

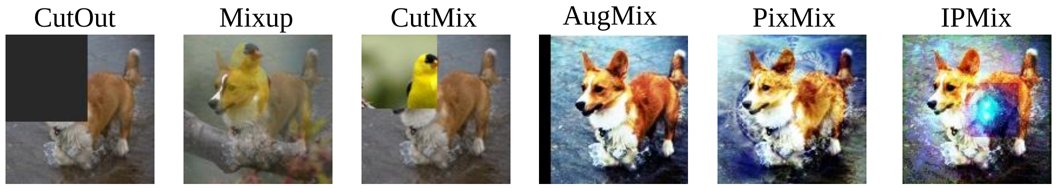

Figure 1: Visual comparison of various data augmentation methods. IPMix utilizes the structural complexity of fractals and multi-scale information to generate more diverse examples.

Previous studies have focused on either pixel-level or patch-level information to improve model performance. However, most of these techniques are label-variant, which may lead to manifold intrusion [24, 3] and decrease performance on unseen data. Simultaneously, a limitation of image-level data augmentation techniques is the computationally expensive search for an optimal augmentation policy, often exceeding the training process's complexity [45, 10]. Given these considerations and the potential for enhancing data augmentation strategies, we mainly discuss one question in this paper: ***How to take advantage of the strengths of the three methods while avoiding their drawbacks?***

Our contributions are as follows:

- We propose **IPMix**, a label-preserving data augmentation approach, which integrates three levels of data augmentation into a single framework with limited computational overhead, demonstrating that these approaches are complementary and that a unification among them is necessary to achieve robustness.
- To further enhance model performance, IPMix incorporates structural complexity from synthetic data at various levels to produce more diverse images. Additionally, we employ random mixing methods and scar-like image patches for multi-scale information fusion.
- Extensive experiments demonstrate that IPMix achieves state-of-the-art corruption robustness and improves numerous safety metrics compared with other data augmentation approaches.

IPMix integrates the three data augmentation techniques in a label-preserving fashion, effectively circumventing potential manifold intrusion and maintaining label consistency[67]. Furthermore, inspired by prior work, IPMix eliminates the need to search for an optimal data augmentation policy, thus reducing computational costs. By addressing these challenges, IPMix has achieved significant improvements, as depicted in Figure 2. In comparison to other methods that focus on leveraging one of these categories for enhancement, IPMix achieves state-of-the-art results in accuracy and robustness.

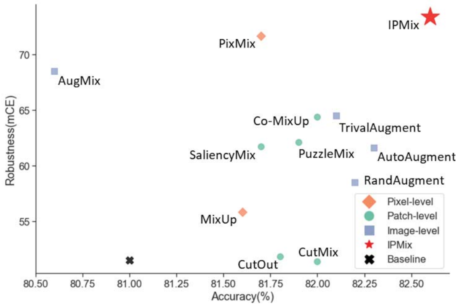

Figure 2: The performance of different levels of data augmentation methods on CIFAR-100. Compared to other approaches which focus on utilizing only one category, IPMix achieves state-of-the-art accuracy and robustness.

Since IPMix involves different levels of data augmentation techniques, it naturally motivates us to design a novel mixing method for better information fusion. Previous research has demonstrated that enhancing training data diversity [90, 13, 78] and image structural complexity [48, 4] is crucial for improving model robustness. The structural complexity of synthetic data, such as fractals and statistical information, can bolster model performance through pre-training [37] or blending with clean images [33]. For better data integration, IPMix mixes clean images with synthetic pictures at different scales by random mixing to improve structural complexity, which can generate more diverse images to improve robustness.

Building on the enhancement of corruption robustness, we further extend IPMix's capabilities to enhance various safety metrics to fulfill the demands of constructing secure and reliable systems in real-world situations [27]. We demonstrate that IPMix improves numerous safety metrics, including corruption robustness, calibrated uncertainty estimates, adversarial robustness, anomaly detection, and prediction consistency. On CIFAR-10-C and CIFAR-100-C, IPMix achieves the best results across different architectures. On ImageNet, IPMix outperforms previous methods and gains a substantial improvement on various safety measure benchmarks, achieving state-of-the-art or comparable results on ImageNet-R, ImageNet-A, and ImageNet-O [26, 32].

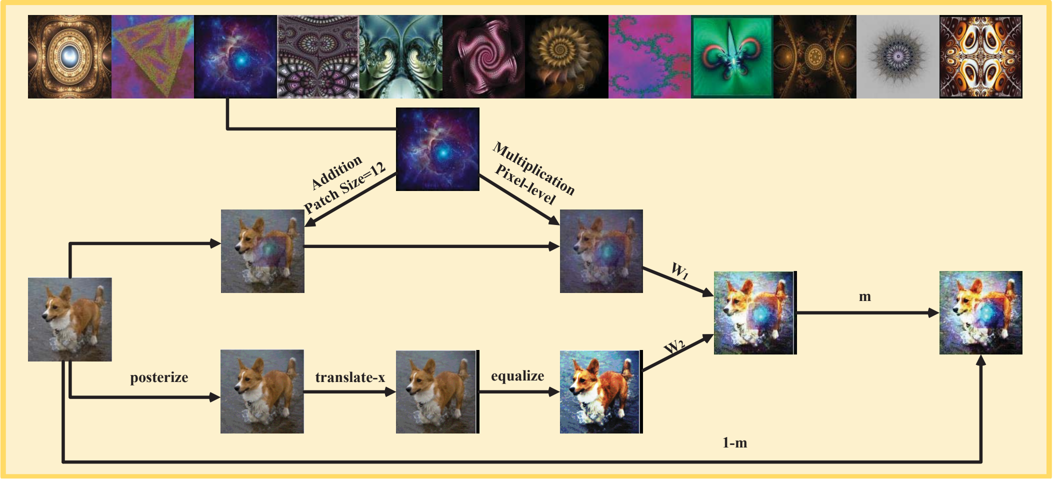

Figure 3: Top: Sample fractals from IPMix set. Bottom: An example of IPMix applied on a dog image, k = 2, t = 3. We randomly select P (pixel and patch) data augmentation methods and image-level data augmentation methods to generate a highly diverse set of augmented images. We sample w_k (k = 2, in this case) from Dirichlet distribution and use skip connection (m sample from a Beta distribution) to maintain semantic consistency.

2 Related Works

2.1 Data Augmentation

Data augmentation is crucial to the success of modern neural networks, contributing significantly to the improvement of model generalization performance. The presented data augmentation approaches can be classified into three high-level categories: image-level, pixel-level, and patch-level augmentations.

Image-level data augmentation. Image-level data augmentation methods are commonly label-preserving, applying transformations on the whole image to improve data diversity. AutoAugment [10] utilizes reinforcement learning to automatically search optimal compositions of transformations. Adversarial AutoAugment [91] generates adversarial images to extend data and produces a dynamic policy during training. TrivialAugment [57] randomly selects an operation and the magnitude to reduce search space and improve performance. AugMix [31] uses multiple transformations to create high diversity of augmented images, achieving state-of-the-art results on corruption robustness and calibration. AugMax [78] unifies diversity and hardness to search for the worst-case mixing strategy. PRIME [55] uses max-entropy image transformations to boost model corruption robustness.

Pixel-level data augmentation. Pixel-level data augmentation methods mix images using pixel-wise weighted averages. MixUp [90] generates augmented images by linearly interpolating between two randomly selected images and their corresponding labels. Manifold MixUp [74] encourages neural networks to learn smooth interpolations between data points in the hidden layers, improving accuracy by comparison with MixUp. PixMix [33] utilizes structural complexity synthetic pictures, such as fractals and feature visualizations, to improve model performance. Our work shared similarities with PixMix, but we use multi-scale information and better information fusion methods to train robust models by leveraging more diverse examples.

Patch-level data augmentation. Patch-level data augmentation methods mask or replace parts of the original image with different information. CutOut [13] randomly masks out regions of a clean image to learn less discriminative portions, thereby improving accuracy. CutMix [86] replaces a patch of an original image with another randomly picked image to improve performance. Patch Gaussian [49], which inputs a patch of Gaussian noise into the clean picture, combines the improved accuracy of CutOut with the noise robustness of Gaussian. SaliencyMix [73], based on the maximum intensity pixel local in the saliency map, replaces a square patch of the original image with salient information from another image. TokenMix [46] improves the performance of vision transformers by partitioning the mixing region into multiple separated parts and mixing two images at the token level. AutoMix [47] optimizes both the mixed sample generation task and the mixup classification task in a momentum training pipeline with corresponding sub-networks in a bi-level optimization framework.

2.2 Safety Measures

When deploying network models in real-world scenarios, it is crucial to consider comprehensive security measures beyond standard accuracy. Implementing unsafe machine learning systems in high-stakes environments [18, 60, 66] can lead to incalculable losses. With the rise of multimodal large language models (MLLMs) [34, 58, 65], safety issues are receiving increasing attention because their superior performance still makes mistakes. For example, GPT-4 [58] may be confidently wrong in its predictions and disturbed by adversarial questions. Previous research has proposed various safety measures, including but not limited to robustness and calibration.

Robustness. Corruption robustness considers how to improve the model resistance to unseen natural perturbations under data distribution shifts. As a variant of the original ImageNet, ImageNet-C [28] consists of 15 diverse commonplace corruptions belonging to different categories with five levels of severity, regarded as a general benchmark for corruption robustness. In addition to natural corruption, Hendrycks et al. [26] demonstrate that models should measure generalization to various abstract visual renditions. The robustness of adversarial attacks focuses on defending against imperceptible perturbations to images [14]. Prior works have proposed that there is a trade-off between the robustness of adversarial perturbations and clean image accuracy [81, 82]. ImageNet-O and ImageNet-A [32], widely regarded as benchmarks for evaluating image classifier performance under shifts in both input data and label distributions, are utilized for anomaly detection.

Calibration. Calibrated prediction confidences, which indicate whether a model's output should be trusted, are valuable for classification models in real-world settings. Bayesian approaches [23] are widely used to deal with uncertainty estimation. Kuleshov et al. [42] utilize recalibration methods to solve the miscalibration of credible intervals. Ovadia et al. [59] provide a benchmark for evaluating the accuracy and uncertainty of models under data distributional shifts.

2.3 Training with Synthetic Data

Previous works have proved that training with synthetic data can improve performance on real datasets. Debidatta et al. [16] discover that combining synthetic annotated datasets with real data can significantly improve the performance of instance detection. Baradad et al. [4] generate synthetic data by utilizing various procedural noise models. In addition, they find that naturalism and diversity are two important properties for synthetic data to achieve comparable results with real datasets. Kataoka et al. [36, 35] propose a suite of datasets generated by formula-driven supervised learning.

3 An Attempt to Integrate Existing Approaches

Some prior studies [33, 86] have suggested that combining different data augmentation techniques with existing methods can improve accuracy on standard datasets. However, these works merely employed simple combinations without considering the compatibility between methods at different levels. Simultaneously, these studies chose the clean accuracy as the sole evaluation metric and have not taken the model's safety performance into account. In this section, we select MixUp [90], CutMix [86], and AugMix [31] as representative data augmentation approaches for pixel-level, patch-level, and image-level, respectively, to conduct combination experiments of these approaches on CIFAR-100. Please refer to Appendix F for more details about the combination experiments.

Table 1: The combination of different levels of data augmentation. M, C and A are abbreviations for MixUp, CutMix, and AugMix, respectively.

	Classification	Robustness	Calibration
	Error(↓)	mCE(↓)	RMS(↓)
Vanilla	21.3	50	14.6
+M	20.5 (-0.8)	45.9(-4.1)	10.5(-4.1)
+M+C	20.2 (-1.1)	46.1(-3.9)	22.7(+8.1)
+M+C+A	23.4 (+2.1)	50.1(+0.1)	25.6(+11)

Results on Table 1 demonstrate that simply combining different data augmentation methods may significantly impair model performance. This could be attributed to the excessive perturbation of training data caused by the combination of these methods, making the newly generated samples more challenging to identify and impacting the model's ability to learn useful features, leading to performance degradation. When multiple label-variant methods are combined, manifold intrusion

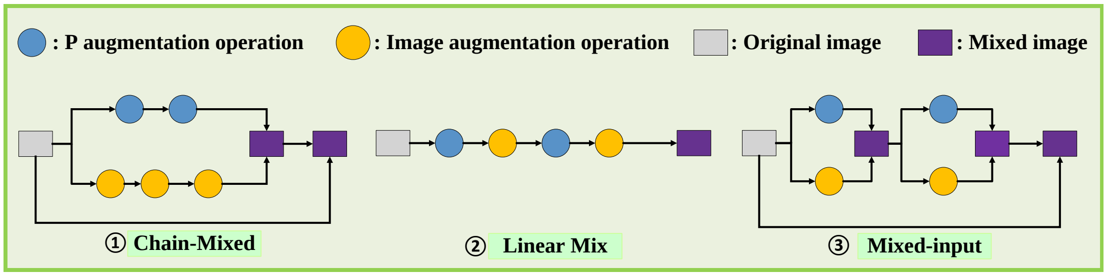

Figure 4: Different mixing framework of IPMix. P augmentation operation represents pixel-level and patch-level augmentation operations. ① Utilizing P operations and image-level operations in different chains and mixing the results. ② A clean image is randomly carried out by P operations or image-level operations in linear combinations to generate an IPMix image. ③ leveraging the mixed image as a new input.

issues may be more likely to arise. One possible solution for better information integration is to incorporate approaches (*e.g.*, MixUp) into search-based data augmentation techniques [11, 57]. However, searching the space for an optimal DA policy will bring expensive computation. Furthermore, this approach aims at improving clean accuracy and does not consider the overall safety performance.

4 IPMix: A Simple Method for Training Robust Classifiers

In this section, we propose IPMix, which integrates three levels of data augmentation methods into a label-preserving approach, comprehensively improving safety metrics without sacrificing clean accuracy. We first demonstrate how to merge various techniques into a coherent framework and then propose novel approaches to achieve superior information fusion.

4.1 Integrates Different Levels into A Coherent Approach

Pixel-level & Patch-level. As a label-preserving data augmentation approach, IPMix uses the equation below to mix two input images:

$$\tilde{x} = B \odot x_1 + (I - B) \odot x_2 \tag{1}$$

Where x_1 is the input image and x_2 represents an unlabeled synthetic image (*e.g.*, fractals, spectrum, or auto-generated contours). B is a mask matrix suitable for both patch-level and pixel-level data augmentation methods, and I is a binary mask filled with ones, having the same dimensions as B. $\odot$ represents the element-wise product. When performing mixing operations at the patch level, we choose a patch of random size and position from B, with a value of λ (sample from Beta distribution) in this range and a value of 1 in other areas, which ensures that except for the mixing patch, the rest of the generated image comes from x_1. When performing mixing operations at the pixel level, we treat the entire image as a patch, with a value of λ. To make it efficient, we adopt fractals as representatives of synthetic data. However, IPMix is insensitive to mixing sets change, as shown in Table 8.

Fractals are geometric shapes with structural complexities and natural geometries. While previous works [37, 1] merely use iterated function systems (IFS) to create fractal data, we employ the Escape-time Algorithm for generating "orbit trap" complex fractals to enhance dataset complexity and diversity. Please refer to Appendix E for details about generating fractal images.

The above-described method provides two key advantages: (1) We utilize a simple approach to combine operations of two levels, facilitating better information fusion. (2) Our method is label-preserving, ensuring it is not affected by manifold intrusion while eliminating the need for label smoothing [56]. In the following sections, we refer to the method used in Eq. (1) as **P-level** data augmentation, signifying the employment of both patch-level and pixel-level methods.

Image-level. IPMix leverages various augmentation techniques and compositions to create a new image that does not deviate significantly from the original. Drawing inspiration from previous works [57, 31], we randomly sample operations from PIL (*e.g.*, brightness, sharpness) and randomly sample strengths to enhance the diversity of training data without expensive searching. Notably, these operations are disjoint from ImageNet-C corruptions, ensuring the robustness test's validity.

Table 2: Results are reported on CIFAR-100 and CIFAR-100-C with ResNeXt-29. The Chain-Mixed achieves the most balanced result on these metrics. Bold is best.

	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
Chain-Mixed	18.3	28.1	3.8
Linear Mix	**18.2**	**27.4**	13.5
Mixed Input	19.8	29.6	**3.6**

The IPMix framework. To determine the most effective methods for combining P-level and image-level, we conducted experiments using different mixing structures to generate a diverse set of IPMix images, as illustrated in the Figure 4 and Table 2. While Linear Mix achieves excellent results in clean accuracy and corruption robustness, it performs poorly in calibrated prediction confidence. Mixed Input performs better in calibration but is inferior in accuracy and corruption robustness compared to Chain-Mixed. Consequently, we chose Chain-Mixed as the default framework for IPMix. Furthermore, the experimental results highlight the potential of establishing a general framework for integrating various data augmentation methods.

4.2 Multi-scale Information Fusion

IPMix can enhance the diversity and the structural complexity of training data to improve model performance. However, we found that simple mixing methods restrict the model's capabilities. To overcome this issue, we use random mixing and scar-like image patches for achieving more effective information fusion.

Random mixing. In previous data augmentation works, it is typical to either linearly mix two images or extract specific image features, such as saliency [38, 73], which requires additional computations, for image mixing. As IPMix incorporates various levels of operations, its objective is to enhance the mixing of images, ultimately increasing data diversity. To accomplish this objective, IPMix employs four mixing operations: addition, multiplication, random pixel mixing, and random element mixing [69]. Random pixel mixing creates a binary mask of size $H \times W \times 1$ that operates on each channel sequentially, while random element mixing generates a binary mask of size $H \times W \times 3$ (RGB) that applies to all channels simultaneously. An example is shown in Figure 5. The experiments in Appendix B.1 show that both operations are beneficial to better information mixing between images and fractals.

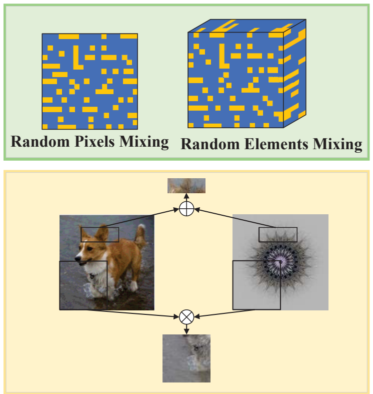

Figure 5: Top: Examples of random mixing operations. Bottom: Examples of IPMix-Scar mixing and IPMix-Square mixing.

Scar-like image patches. IPMix-Scar employs a long, thin rectangular box filled with an image patch to enhance dataset diversity, which has proven effective for anomaly detection [43]. An example of patch mixing is illustrated in Figure 5. First, IPMix randomly selects a point and a scar or square of the previously chosen size from the current image. Next, IPMix crops corresponding portions of the current image and the fractal picture and combine them.

Finally, we obtain IPMix, which employs various levels of data augmentation to create diverse transformations with image structural complexity and data diversity. Figure 3 displays an example of IPMix, where ***k*** denotes the number of augmented chains, and ***t*** represents the maximum number of times an image can be augmented. The algorithm of IPMix is summarized in Appendix D.

5 Experiments

In this section, we showcase the significant performance improvements brought by IPMix on clean datasets in multiple settings. We present the evaluation results of IPMix for image classification on three datasets—CIFAR-10, CIFAR-100 [40], and ImageNet [12]—across various models. Besides clean Classification, we assess IPMix on diverse safety tasks, including adversarial attack robustness, corruption robustness, prediction consistency, calibration, and anomaly detection. Please refer to Appendix C for details about the evaluation metrics. Lastly, we evaluate the properties of IPMix in thorough ablation studies and compare our approach with different levels of methods.

Table 3: Clean Error for IPMix on CIFAR-10 and CIFAR-100, lower is better. Top : CIFAR-10. Bottom : CIFAR-100. Mean and standard derivation over three random seeds is shown for each experiment. Bold is best.

	Vanilla	MixUp	CutOut	CutMix	AugMix	PixMix	IPMix
WideResNet40-4	4.4$_{(\pm 0.05)}$	3.8$_{(\pm 0.06)}$	**3.6**$_{(\pm 0.05)}$	4.0$_{(\pm 0.04)}$	4.3$_{(\pm 0.08)}$	4.1$_{(\pm 0.08)}$	4.0$_{(\pm 0.06)}$
WideResNet28-10	3.8$_{(\pm 0.07)}$	3.6$_{(\pm 0.08)}$	3.4$_{(\pm 0.06)}$	3.4$_{(\pm 0.05)}$	3.4$_{(\pm 0.07)}$	3.8$_{(\pm 0.13)}$	**3.3**$_{(\pm 0.08)}$
ResNeXt-29	4.3$_{(\pm 0.04)}$	**3.8**$_{(\pm 0.11)}$	4.2$_{(\pm 0.08)}$	**3.8**$_{(\pm 0.02)}$	4.2$_{(\pm 0.05)}$	**3.8**$_{(\pm 0.09)}$	**3.8**$_{(\pm 0.07)}$
ResNet-18	4.4$_{(\pm 0.05)}$	4.2$_{(\pm 0.04)}$	4.1$_{(\pm 0.05)}$	**4.0**$_{(\pm 0.04)}$	4.5$_{(\pm 0.03)}$	4.4$_{(\pm 0.05)}$	4.2$_{(\pm 0.07)}$
Mean	4.2	3.9	**3.8**	**3.8**	4.1	4.0	**3.8**
WideResNet40-4	21.3$_{(\pm 0.11)}$	20.5$_{(\pm 0.13)}$	19.9$_{(\pm 0.11)}$	20.3$_{(\pm 0.15)}$	20.6$_{(\pm 0.15)}$	20.4$_{(\pm 0.17)}$	**19.4**$_{(\pm 0.14)}$
WideResNet28-10	19.0$_{(\pm 0.13)}$	18.4$_{(\pm 0.12)}$	18.8$_{(\pm 0.15)}$	18.0$_{(\pm 0.11)}$	19.4$_{(\pm 0.11)}$	18.3$_{(\pm 0.13)}$	**17.4**$_{(\pm 0.15)}$
ResNeXt-29	20.4$_{(\pm 0.11)}$	20.3$_{(\pm 0.12)}$	19.6$_{(\pm 0.13)}$	19.5$_{(\pm 0.13)}$	20.4$_{(\pm 0.13)}$	20.1$_{(\pm 0.11)}$	**18.3**$_{(\pm 0.12)}$
ResNet-18	23.7$_{(\pm 0.09)}$	21.0$_{(\pm 0.07)}$	22.0$_{(\pm 0.11)}$	**20.8**$_{(\pm 0.12)}$	23.0$_{(\pm 0.14)}$	21.6$_{(\pm 0.15)}$	21.6$_{(\pm 0.13)}$
Mean	21.1	20.0	20.1	19.7	20.8	20.1	**19.2**

Table 4: Mean Corruption Error (mCE) for IPMix across architectures on CIFAR-10-C and CIFAR-100-C, lower is better. Top : CIFAR-10-C. Bottom : CIFAR-100-C. Bold is best.

	Vanilla	MixUp	CutOut	CutMix	AugMix	PixMix	IPMix
WideResNet40-4	26.4$_{(\pm 0.14)}$	21$_{(\pm 0.15)}$	25.9$_{(\pm 0.13)}$	26$_{(\pm 0.13)}$	10$_{(\pm 0.12)}$	9.5$_{(\pm 0.14)}$	**8.6**$_{(\pm 0.14)}$
WideResNet28-10	24.2$_{(\pm 0.15)}$	19.2$_{(\pm 0.17)}$	23.5$_{(\pm 0.17)}$	25.1$_{(\pm 0.13)}$	9.1$_{(\pm 0.14)}$	8.7$_{(\pm 0.14)}$	**7.5**$_{(\pm 0.17)}$
ResNeXt-29	27.5$_{(\pm 0.11)}$	23.6$_{(\pm 0.18)}$	27.3$_{(\pm 0.18)}$	28.5$_{(\pm 0.18)}$	11.3$_{(\pm 0.15)}$	9.2$_{(\pm 0.12)}$	**8.6**$_{(\pm 0.19)}$
ResNet-18	25$_{(\pm 0.09)}$	20$_{(\pm 0.15)}$	24.1$_{(\pm 0.13)}$	24.7$_{(\pm 0.19)}$	10.4$_{(\pm 0.13)}$	9$_{(\pm 0.11)}$	**8.4**$_{(\pm 0.17)}$
Mean	25.8	20.9	25.2	26	10	9.1	**8.2**
WideResNet40-4	50$_{(\pm 0.15)}$	45.9$_{(\pm 0.19)}$	51.5$_{(\pm 0.17)}$	50$_{(\pm 0.19)}$	33.3$_{(\pm 0.22)}$	31.1$_{(\pm 0.19)}$	**28.6**$_{(\pm 0.15)}$
WideResNet28-10	48.5$_{(\pm 0.21)}$	44.2$_{(\pm 0.18)}$	48.2$_{(\pm 0.15)}$	48.6$_{(\pm 0.21)}$	31.5$_{(\pm 0.21)}$	28.3$_{(\pm 0.21)}$	**26.6**$_{(\pm 0.19)}$
ResNeXt-29	51.4$_{(\pm 0.19)}$	47.9$_{(\pm 0.21)}$	51$_{(\pm 0.17)}$	52.4$_{(\pm 0.22)}$	34.1$_{(\pm 0.24)}$	30.6$_{(\pm 0.23)}$	**28.1**$_{(\pm 0.21)}$
ResNet-18	50$_{(\pm 0.18)}$	45.5$_{(\pm 0.21)}$	50.2$_{(\pm 0.19)}$	50.8$_{(\pm 0.24)}$	35$_{(\pm 0.25)}$	31.4$_{(\pm 0.21)}$	**29.9**$_{(\pm 0.19)}$
Mean	50	45.9	50.2	50.5	33.4	30.3	**28.3**

We evaluate IPMix on CIFAR-10-C, CIFAR-100-C, and ImageNet-C to measure its resistance to corruption data shifts. We test IPMix on CIFAR-10-P, CIFAR-100-P, and ImageNet-P to measure network prediction stability against minor perturbations. To thoroughly demonstrate our method's capabilities, we assess it on supplementary datasets, including ImageNet-R, ImageNet-O, and ImageNet-A. Experiments on these datasets validate our approach's robustness under real-world distribution shifts.

5.1 Evaluation on CIFAR

We experiment with different backbone architectures on CIFAR-10 and CIFAR-100, including 40-4 Wide ResNet [87], 28-10 Wide ResNet, ResNeXt-29 [83], and Resnet-18 [25]. We compare IPMix with various data augmentation methods, including CutOut, MixUp, CutMix, AugMix, and PixMix. Please refer to Appendix A for more details about the training configurations.

Accuracy. In Table 3, we demonstrate that IPMix improves standard accuracy across architectures. In comparison with other approaches, IPMix achieves the best or comparable accuracy, showing the improvement of safety measures is not at the cost of hurting clean accuracy.

Corruption robustness. Results show that IPMix substantially improves corruption robustness across architectures. Compared to AugMix on CIFAR-100-C, IPMix achieves **4.7%**(40-4) and **4.9%**(28-10) improvement on WideResNet, **6%** on ResNeXt, and **5.1%** on ResNet. Table 4 demonstrates that IPMix achieves state-of-the-art results on both CIFAR-10-C and CIFAR-100-C.

Calibration. We utilize RMS calibration error [30] to evaluate the empirical frequency of correctness. As depicted in Figure 6, IPMix surpasses other methods, achieving state-of-the-art results.

Prediction consistency. We leverage the mean flip rate (mFR) to evaluate prediction consistency on CIFAR-10-P and CIFAR-100-P [28]. IPMix achieves the lowest mFR, as shown in Figure 7.

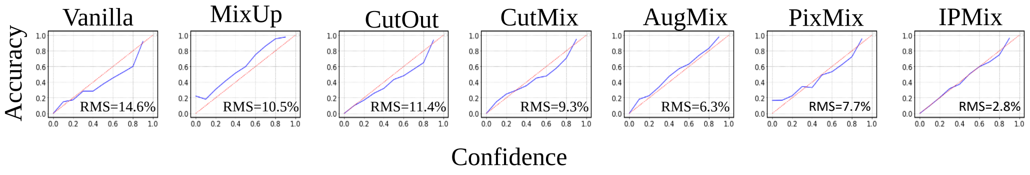

Figure 6: The results of calibration on CIFAR-100. IPMix achieves the lowest RMS error in all data augmentation methods, improving **11.8%** by comparing with Vanilla.

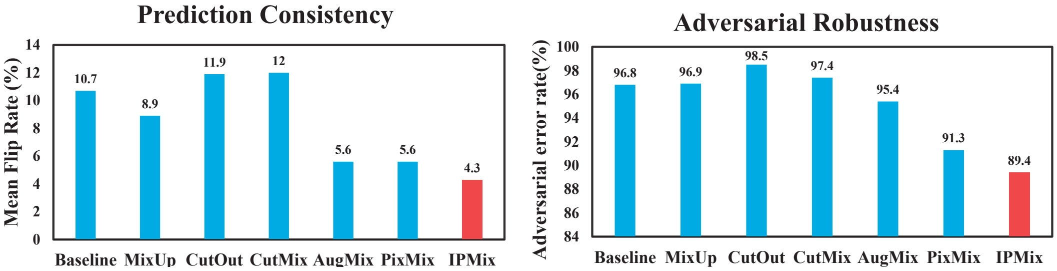

Figure 7: **Left: prediction consistency. Right: adversarial robustness.** IPMix achieves the best results on both metrics, demonstrating its ability to improve overall security performance.

Adversarial robustness. This measure evaluates the resistance of adversarially perturbed by projected gradient descent. We utilize PGD [50] to verify the adversarial robustness of image classifiers. The results in Figure 7 show that IPMix achieves the lowest error.

Table 5: The results of IPMix on ImageNet. For Anomaly Detection, we test the accuracy on ImageNet-A and AUPR on ImageNet-O, higher is better. IPMix achieves round improvement over various data augmentation methods. Bold is best, and underline is second.

	Classification	Robustness		Consistency	Calibration			Anomaly Detection	
	Clean Error(↓)	ImageNet-C mCE(↓)	ImageNet-R Error(↓)	ImageNet-P mFR(↓)	C RMS(↓)	R RMS(↓)	A RMS(↓)	ImageNet-A Classification(↑)	ImageNet-O AUPR(↑)
Vanilla	23.9	78.6	64	57.7	12	19.9	47	2.2	16.2
MixUp[90]	22.7	76.5	62.4	54.6	9.3	41.7	49.3	5.2	16.1
CutOut[13]	22.6	73.1	64.6	57.9	11.3	19.7	46.3	4.7	15.9
CutMix[86]	22.9	77.2	66.5	58.1	9.6	44.2	48	**7.2**	16.5
AugMix[31]	22.6	68.5	61.8	52.3	8.1	13.1	43.5	3.8	17.4
AugMax[78]	22.9	67.4	62.1	54.6	8.8	12.1	44.7	3.9	17.1
PixMix[33]	22.4	65.4	59.8	50.8	7.2	12.3	44	5.9	17.3
IPMix	**22.2**	**63**	**57.4**	**48.5**	**7.1**	**7**	**30**	6.6	**18.2**

5.2 Evaluation on ImageNet

For ImageNet experiments, we compare different data augmentation methods, including MixUp, CutOut, CutMix, AugMix, AugMax [78], and PixMix. We utilize SGD optimizer with an initial learning rate of 0.01 to train ResNet-50 for 180 epochs following a cosine decay schedule. Please refer to Appendix A for more details about the training configurations.

IPMix achieves state-of-the-art or comparable performances on a broad range of safety measures, as shown in Table 5. Compared with other methods, IPMix improves the resistance of out-of-distribution shifts without reducing clean accuracy. On corruption robustness, IPMix outperforms Vanilla by **15.6%** and AugMix by **5.5%**, achieving state-of-the-art results. On ImageNet-R, IPMix demonstrates the ability to improve rendition robustness, increasing by **6.6%** by comparison with Vanilla. On ImageNet-P, IPMix improves mFR by **9.2%** over Vanilla and **2.3%** over PixMix. On calibration tests, IPMix surpasses all methods on ImageNet-C, ImageNet-R, and ImageNet-A, improving RMS by **0.1%**, **5.1%**, and **13.5%** by comparison with the second-best approach. Furthermore, IPMix achieves convincing results on ImageNet-A and ImageNet-O, demonstrating its exceptional ability in anomaly detection. The results demonstrate that IPMix can roundly improve safety metrics.

Table 6: Ablation results of different components of IPMix on ImageNet with ResNet-50.

	Classification	Robustness		Consistency	Calibration			Anomaly Detection	
	Clean Error(↓)	ImageNet-C mCE(↓)	ImageNet-R Error(↓)	ImageNet-P mFR(↓)	C RMS(↓)	R RMS(↓)	A RMS(↓)	ImageNet-A Classification(↑)	ImageNet-O AUPR(↑)
IPMix	**22.2**	**63**	**57.4**	**48.5**	**7.1**	**7**	**30**	**6.6**	**18.2**
w/o patch	22.8$_{(\pm 0.11)}$	65.1$_{(\pm 0.16)}$	58.8$_{(\pm 0.11)}$	49.1$_{(\pm 0.15)}$	7.8$_{(\pm 0.11)}$	7.4$_{(\pm 0.08)}$	31.1$_{(\pm 0.01)}$	6$_{(\pm 0.01)}$	17.7$_{(\pm 0.02)}$
w/o pixel	23.1$_{(\pm 0.15)}$	65.6$_{(\pm 0.19)}$	59.3$_{(\pm 0.13)}$	49.5$_{(\pm 0.17)}$	8.2$_{(\pm 0.13)}$	7.4$_{(\pm 0.09)}$	32.4$_{(\pm 0.11)}$	5.6$_{(\pm 0.01)}$	17.2$_{(\pm 0.03)}$
w/o image	23.5$_{(\pm 0.16)}$	66.2$_{(\pm 0.21)}$	59.5$_{(\pm 0.17)}$	49.6$_{(\pm 0.14)}$	8.8$_{(\pm 0.13)}$	8.1$_{(\pm 0.13)}$	33.5$_{(\pm 0.13)}$	6.5$_{(\pm 0.02)}$	17.8$_{(\pm 0.03)}$

5.3 Ablation Study

In this paragraph, we evaluate the properties of our approach by ablation experiments. We first study the influence of different parts of IPMix on performance and then assess the stability of IPMix under various mixing sources. Please refer to more ablation experiments in Appendix B.1.

Table 7: Ablation results of different components of IPMix on CIFAR-100. Mean and standard derivation over three random seeds is shown for each experiment. Bold is the best.

	Classification	Robustness	Calibration
IPMix	**19.4**	**28.6**	**2.8**
w/o patch	19.7(± 0.13)	30 (± 0.21)	4.6 (± 0.07)
w/o pixel	19.6 (± 0.09)	33 (± 0.35)	8.2 (± 0.12)
w/o image	20.1 (± 0.27)	34 (± 0.65)	8.6 (± 0.21)

Components of IPMix. In this section, we evaluate the influence of different IPMix components on performance. We execute ablation experiments on the three primary IPMix constituents: image-level, patch-level, and pixel-level. The results show the indispensable contribution of each component to enhancing model performance, demonstrating that these approaches are complementary and that a unification among them is necessary to achieve robustness. The ablation experiment results are shown in Table 6 and Table 7. Please refer to thorough analysis in Appendix J.

Table 8: Ablation results on IPMix across different mixing sets.The results show that IPMix is insensitive to mixing sets change.

Mixing sets	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
Fractal + FVis	**19.4**	28.8	3.3
FractalDB	20	29	5.4
RCDB	19.5	**28.4**	3.2
Dead Leaves	**19.4**	29.1	3.1
Spectrum	19.8	29.2	4
fractals(ours)	**19.4**	28.6	**2.8**

Mixing sources. The excellent performance of IPMix is partly due to the structural complexity of fractal pictures. In this part, we examine the sensitivity of IPMix to different fractal sources on CIFAR-100. We report clean accuracy, corruption robustness, and calibration from different sources with WRN40-4. Fractal + FVis is the default setting of PixMix, which consists of fractals and feature visualization. FractalDB [36] consists of fractal images generated by Iterated Function System (IFS). RCDB [35] consists of auto-generated contours. Dead Leaves and Spectrum generated from generative image models [4]. The full results show in Table 8.

5.4 The Comparison with Different Levels of Method

In this section, we perform an extensive performance comparison between IPMix and a range of existing methods using multiple metrics. We consider AutoAugment, RandAugment, and TrivialAugment [57] as representative image-level techniques, while SaliencyMix, PuzzleMix [39], and Co-Mixup [38] serve as typical patch-level techniques. For pixel-level methods, Manifold Mixup stands as our representative choice. IPMix does not require searching for the optimal DA policy like image-level techniques. In contrast to patch-level approaches, IPMix eliminates the need for saliency computations. The results in Table 9 show that IPMix outperformed all other methods on all metrics.

6 Analysis of IPMix

IPMix combines three levels of data augmentation into a unified, label-preserving technique to improve model performance. We believe that IPMix's superior performance is due to the increased

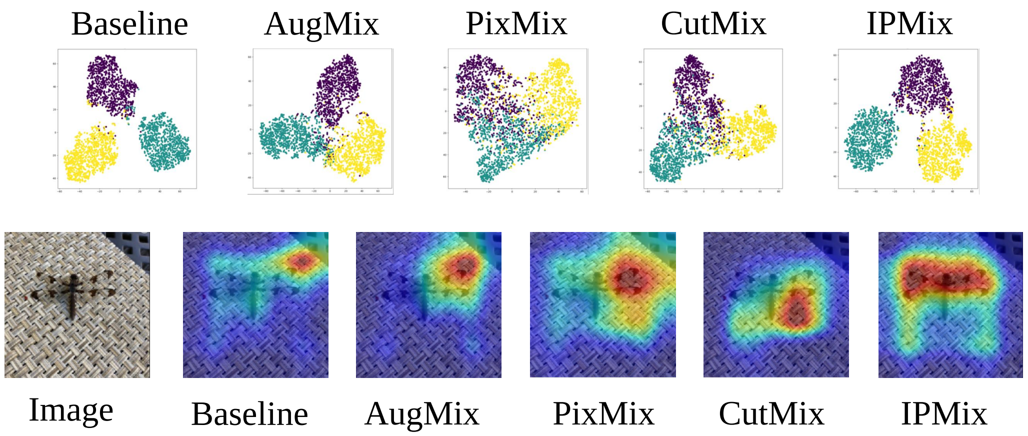

Figure 8: **Top: t-SNE visualization.** The features are from the penultimate layer of a WRN40-4 trained on CIFAR10. Compared with other approaches, IPMix has distinct boundaries between different category clusters and generates diverse samples to cover boundary areas, thereby improving the generalization ability. **Bottom: The Grad-CAM visualization**, with input images sourced from ImageNet-A, demonstrates that IPMix excels in identifying objects within complex scenarios.

Table 9: Results of different augmentation methods on CIFAR-100 and CIFAR-100-C with 28-10 Wide ResNet. Bold is best.

Methods	Classification Error(↓)	Robustness mCE(↓)	Adversaries Error(↓)	Consistency mFR(↓)	Calibration RMS(↓)
AutoAugment [10]	17.7$_{(\pm 0.11)}$	38.4$_{(\pm 0.15)}$	97.8$_{(\pm 0.22)}$	8$_{(\pm 0.06)}$	7.9$_{(\pm 0.06)}$
RandAugment [11]	17.8$_{(\pm 0.14)}$	41.5$_{(\pm 0.13)}$	96.6$_{(\pm 0.25)}$	8.6$_{(\pm 0.10)}$	7.9$_{(\pm 0.04)}$
TrivialAugment [57]	17.9$_{(\pm 0.13)}$	96.3$_{(\pm 0.21)}$	35.4$_{(\pm 0.23)}$	7.3$_{(\pm 0.07)}$	8.7$_{(\pm 0.04)}$
SaliencyMix [73]	18.3$_{(\pm 0.14)}$	38.3$_{(\pm 0.24)}$	96.7$_{(\pm 0.21)}$	10.8$_{(\pm 0.07)}$	7.1$_{(\pm 0.07)}$
PuzzleMix [39]	18.1$_{(\pm 0.11)}$	37.9$_{(\pm 0.21)}$	96.1$_{(\pm 0.23)}$	10.5$_{(\pm 0.04)}$	7.5$_{(\pm 0.08)}$
Co-Mixup [38]	18.0$_{(\pm 0.19)}$	35.6$_{(\pm 0.25)}$	95.6$_{(\pm 0.21)}$	10.1$_{(\pm 0.05)}$	7.7$_{(\pm 0.04)}$
Manifold Mixup [74]	18.8$_{(\pm 0.21)}$	51.3$_{(\pm 0.23)}$	93.4$_{(\pm 0.17)}$	29.9$_{(\pm 0.28)}$	10.2$_{(\pm 0.09)}$
IPMix	**17.4**$_{(\pm 0.17)}$	**26.6**$_{(\pm 0.19)}$	**91.3**$_{(\pm 0.21)}$	**4.2**$_{(\pm 0.11)}$	**6.4**$_{(\pm 0.07)}$

data diversity and enhanced regularization effect. For a more intuitive demonstration of these effects, we utilize t-SNE and Class Activation Mapping (CAM) [64] for visualizations, as shown in Figure 8.

Increasing diversity. IPMix increases the diversity of training data by mixing data at multiple levels, enabling the model to learn a greater variety of feature combinations and patterns. Furthermore, the integration of synthetic data from distinct distributions (*e.g.*, fractals), further amplifies this diversity.

Enhanced regularization effect. The approach of mixing data also serves as a potent regularization technique. By randomly mixing samples, the model is compelled to learn more robust features rather than overly relying on specific sample or class characteristics, which reduces the risk of overfitting and enhances the model's performance in different environments.

7 Conclusion

We propose IPMix, which leverages different levels of augmentation techniques and image structural complexity to improve model performance. By employing random mixing methods, we facilitate more effective information fusion. The experimental results indicate that IPMix can significantly improve various safety metrics. We hope our work will attract attention to joining different methods into coherent and synergetic approaches to improve robustness and other safety measures. This adaptation is crucial given the growing importance of safety requirements in systems design.

Acknowledgement

This work was under the help of Xi Yang. We thank him for his selfless help and valuable suggestions.

References

[1] Connor Anderson and Ryan Farrell. Improving fractal pre-training. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2412–2421, 2021.

[2] Kyungjune Baek and Hyunjung Shim. Commonality in natural images rescues gans: Pretraining gans with generic and privacy-free synthetic data. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7844–7854, 2022.

[3] Raphael Baena, Lucas Drumetz, and Vincent Gripon. Preventing manifold intrusion with locality: Local mixup. *ArXiv*, abs/2201.04368, 2022.

[4] Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. In *Neural Information Processing Systems*, 2021.

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua B. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Neural Information Processing Systems*, 2019.

[6] Dan Andrei Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andr'as Gyorgy, Timothy A. Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *ArXiv*, abs/2104.01086, 2021.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311, 2022.

[8] Sanghyuk Chun, Seong Joon Oh, Sangdoo Yun, Dongyoon Han, Junsuk Choe, and Young Joon Yoo. An empirical evaluation on robustness and uncertainty of regularization methods. *ArXiv*, abs/2003.03879, 2020.

[9] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Edoardo Debenedetti, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *ArXiv*, abs/2010.09670, 2020.

[10] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

[11] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2019.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[13] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *ArXiv*, abs/1708.04552, 2017.

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2017.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[16] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1310–1319, 2017.

[17] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Network anomaly detection using lstm based autoencoder. *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2020.

[18] Samuel G. Finlayson, Isaac S. Kohane, and Andrew Beam. Adversarial attacks against medical deep learning systems. *ArXiv*, abs/1804.05296, 2018.

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ArXiv*, abs/1811.12231, 2018.

[20] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko Herbert Schütt, Matthias Bethge, and Felix Wichmann. Generalisation in humans and deep neural networks. *ArXiv*, abs/1808.08750, 2018.

[21] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1055–1064, 2020.

[22] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2021.

[23] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ArXiv*, abs/1706.04599, 2017.

[24] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *AAAI Conference on Artificial Intelligence*, 2018.

[25] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2020.

[27] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *ArXiv*, abs/2109.13916, 2021.

[28] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, abs/1903.12261, 2018.

[29] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136, 2016.

[30] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. *ArXiv*, abs/1812.04606, 2018.

[31] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *ArXiv*, abs/1912.02781, 2019.

[32] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. Natural adversarial examples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15257–15266, 2019.

[33] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Xiaodong Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16762–16771, 2021.

[34] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models. *ArXiv*, abs/2302.14045, 2023.

[35] Hirokatsu Kataoka, Ryo Hayamizu, Ryosuke Yamada, Kodai Nakashima, Sora Takashima, Xinyu Zhang, Edgar Josafat Martinez-Noriega, Nakamasa Inoue, and Rio Yokota. Replacing labeled real-image datasets with auto-generated contours. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21200–21209, 2022.

[36] Hirokatsu Kataoka, Asato Matsumoto, Ryosuke Yamada, Yutaka Satoh, Eisuke Yamagata, and Nakamasa Inoue. Formula-driven supervised learning with recursive tiling patterns. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 4081–4088, 2021.

[37] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. *International Journal of Computer Vision*, 130:990 – 1007, 2021.

[38] Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. *ArXiv*, abs/2102.03065, 2021.

[39] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. *ArXiv*, abs/2009.06962, 2020.

[40] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[42] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *ArXiv*, abs/1807.00263, 2018.

[43] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9659–9669, 2021.

[44] Xiaodan Li, Yuefeng Chen, Yao Zhu, Shuhui Wang, Rong Zhang, and Hui Xue. Imagenet-e: Benchmarking neural network robustness via attribute editing. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20371–20381, 2023.

[45] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *Neural Information Processing Systems*, 2019.

[46] Jihao Liu, B. Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *European Conference on Computer Vision*, 2022.

[47] Zicheng Liu, Siyuan Li, Di Wu, Zhiyuan Chen, Lirong Wu, Jianzhu Guo, and Stan Z. Li. Automix: Unveiling the power of mixup for stronger classifiers. In *European Conference on Computer Vision*, 2021.

[48] Raphael Gontijo Lopes, Sylvia J. Smullin, Ekin Dogus Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *ArXiv*, abs/2002.08973, 2020.

[49] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin Dogus Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *ArXiv*, abs/1906.02611, 2019.

[50] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017.

[51] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3144–3153, 2021.

[52] Daniel Maturana and Sebastian A. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[53] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *ArXiv*, abs/1702.04267, 2017.

[54] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. *ArXiv*, abs/2102.11273, 2021.

[55] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Prime: A few primitives can boost robustness to common corruptions. In *European Conference on Computer Vision*, 2021.

[56] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Neural Information Processing Systems*, 2019.

[57] Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 754–762, 2021.

[58] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[59] Yaniv Ovadia, Emily Fertig, Jie Jessie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *ArXiv*, abs/1906.02530, 2019.

[60] Nicolas Papernot, Patrick Mcdaniel, Arunesh Sinha, and Michael P. Wellman. Sok: Security and privacy in machine learning. *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414, 2018.

[61] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019.

[62] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14298–14308, 2021.

[63] Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. Increasing the robustness of dnns against image corruptions by playing the game of noise. *ArXiv*, abs/2001.06057, 2020.

[64] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2016.

[65] Yongliang Shen, Kaitao Song, Xu Tan, Dong Sheng Li, Weiming Lu, and Yue Ting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *ArXiv*, abs/2303.17580, 2023.

[66] Ludmila Skaf, Elvira Buonocore, Stefano Dumontet, Roberto Capone, and Pier Paolo Franzese. Applying network analysis to explore the global scientific literature on food security. *Ecol. Informatics*, 56:101062, 2020.

[67] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *ArXiv*, abs/2001.07685, 2020.

[68] Adarsh Subbaswamy, Roy J. Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[69] Cecilia Summers and Michael J. Dinneen. Improved mixed-example data augmentation. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270, 2018.

[70] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Asian Conference on Machine Learning*, 2018.

[71] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023.

[72] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv: Machine Learning*, 2018.

[73] A. F. M. Shahab Uddin, Mst Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. *ArXiv*, abs/2006.01791, 2020.

[74] 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*, 2018.

[75] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[76] H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[77] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. Learning robust global representations by penalizing local predictive power. In *Neural Information Processing Systems*, 2019.

[78] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In *Neural Information Processing Systems*, 2021.

[79] X. Wang, Chen Chen, Yuhu Cheng, and Z. Jane Wang. Zero-shot image classification based on deep feature extraction. *IEEE Transactions on Cognitive and Developmental Systems*, 10:432–444, 2018.

[80] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *ArXiv*, abs/2212.10560, 2022.

[81] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Loddon Yuille, and Quoc V. Le. Adversarial examples improve image recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 816–825, 2019.

[82] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Loddon Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, 2018.

[83] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2016.

[84] Ryosuke Yamada, Hirokatsu Kataoka, Naoya Chiba, Yukiyasu Domae, and Tetsuya Ogata. Point cloud pre-training with natural 3d structures. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21251–21261, 2022.

[85] Binh Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[86] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Young Joon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031, 2019.

[87] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

[88] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18102–18112, 2021.

[89] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, abs/1901.08573, 2019.

[90] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412, 2017.

[91] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhaobai Zhong. Adversarial autoaugment. *ArXiv*, abs/1912.11188, 2019.

[92] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, 2017.

Contents of the Appendices:

A Experimental Settings

A.1 CIFAR

In this section, we share the training settings of IPMix on CIFAR. We experiment with various backbone architectures on CIFAR-10 and CIFAR-100, including 40-4 Wide ResNet, 28-10 Wide ResNet [87], ResNeXt-29 [83], and Resnet-18 [25]. We train ResNet and RexNeXt for 200 epochs, and all Wide ResNets for 100 epochs. We employ the SGD optimizer with a weight decay of 0.0001 and a momentum of 0.9. We randomly crop training images to 32×32 resolution with zero padding and flip them horizontally. We compare IPMix with various data augmentation methods, including CutOut, MixUp, CutMix, AugMix, and PixMix. We select a CutOut size of 16×16 pixels on CIAFR-10, and 8×8 on CIFAR-100. For CutMix, we set CutMix probability as 0.5 and α = 1.0. We set k = 3 in AugMix, and k = 3, β = 4 in PixMix for the best results. For IPMix, we set k = 3, t = 3, and randomly select patch sizes from 4, 8, 16, and 32 (pixel-level). All experiments are conducted on a server with two NVIDIA GeForce RTX 3090 GPUs.

A.2 ImageNet-1K

For ImageNet experiments, we compare different data augmentation methods, including MixUp, CutOut, CutMix, AugMix, AugMax [78], and PixMix. Since regularization methods may require a greater number of training epochs to converge, we fine-tune a pre-trained ResNet-50 for 180 epochs. We utilize SGD optimizer with an initial learning rate of 0.01 following a cosine decay schedule, with a batch size of 256. For all approaches, we randomly crop training images to 224×224 resolution with zero padding and flip them horizontally. We adopt α = 0.2 for MixUp and CutMix and select a CutOut size of 56×56 pixels. For IPMix, we use k = 3, t = 3, and randomly select patch sizes from 4, 8, 16, 32, 64, and 256 (pixel-level). We set λ = 12 and n = 5 for AugMax-DuBIN, the same as the paper.

B Additional Experiments of IPMix

B.1 Ablation Exmperiments

IPMix hyperparameters. In this paragraph, we evaluate the hyperparameters sensitivity of IPMix. We examine two hyperparameters: the number of augmented chains k and the maximum image augmentation times t with clean accuracy and robustness. The results in Table 10 demonstrate that IPMix is not sensitive to hyperparameters, showing the performance of IPMix is stable under change.

Mixing operations ablation. In this paragraph, we test IPMix's mixing operation sensitivity. IPMix utilizes four different operations to improve model performance, including addition, multiplication, random pixels mixing, and random elements mixing. The results show in Table 11.

Patch mixing ablation. In this paragraph, we verify IPMix's patch variants, which can be divided into two categories, IPMix-Scar and IPMix-Square. The results in Table 12 show that PachtMix-Scar can improve model robustness.

Table 10: We evaluate clean accuracy on CIFAR-100 and Mean Corruption Error (mCE) on CIFAR-100-C with WRN40-4. The performance of IPMix is not strongly associated with hyperparmeters.

	$k = 2$	$k = 3$	$k = 4$
$t = 2$	19.5 29	19.3 28.9	19.3 29
$t = 3$	19.7 **28.5**	**19.4** 28.6	19.7 28.6

Table 11: Ablation results of IPMix on CIFAR-100 with WRN40-4. While the addition + multiplication achieves the highest accuracy, it compromises corruption and calibration. In contrast, random mixing operations bolster robustness and calibration. Experiment results demonstrate that combining all mixing operations achieves the most balanced performance.

Mixing operations	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
Addition + Multiplication	**19.2**	31	4.1
Random pixels mixing	19.6	28.7	3.7
Random elements mixing	19.9	28.8	**2.7**
IPMix	19.4	**28.6**	2.8

Table 12: The results of patch variants ablation on CIFAR-100 with ResNeXt-29.

Variants	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
IPMix-Square	**18.3**	28.5	3.9
IPMix-Scar	18.6	**28.0**	4.1
IPMix	**18.3**	28.1	**3.8**

B.2 Additional Robustness Experiments

Recent works propose that some data augmentation techniques are tailored to particular datasets when testing model robustness. To evaluate the generality of IPMix, we experiment with other types of distribution shifts beyond common corruptions. We examine IPMix on CIFAR-10-$\overline{\text{C}}$, CIAFR-100-$\overline{\text{C}}$, and ImageNet-$\overline{\text{C}}$ [54]. CIFAR-10-$\overline{\text{C}}$, CIFAR-100-$\overline{\text{C}}$, and ImageNet-$\overline{\text{C}}$ are similar to CIFAR-C and ImageNet-C but utilize a different set of corruptions.Results in Table 13 demonstrate that IPMix achieves SOTA or comparison results by comparing with other methods.

Table 13: Results of robustness resist other distribution shifts. Bold is best.

Methods	CIFAR-10-$\overline{\text{C}}$	CIFAR-100-$\overline{\text{C}}$	ImageNet-$\overline{\text{C}}$
Vanilla	26.4	52	60.2
MixUp [90]	22.4	50	54.1
CutOut [13]	24.2	50.1	58.4
CutMix [86]	25.1	49.9	57.8
AugMix [31]	19.3	41	54.3
PixMix [33]	13.6	36.7	**47.1**
IPMix	**13**	**36**	47.9

To better assess the performance of IPMix against natural distribution shifts, we extended our evaluation to various ImageNet benchmarks. We test IPMix on ObjectNet [5], ImageNet-E [44], ImageNet-Sketch [77], ImageNet-V2 [19], and Stylized-ImageNet [61]. The results presented in Table 14 indicate that IPMix consistently outperforms under diverse data shifts, underscoring its capability to enhance model robustness.

Table 14: Results of IPMix against natural distribution shifts. Higher is better.

	ObjectNet	ImageNet-E	ImageNet-Sketch	ImageNet-V2	Stylized-ImageNet
Vanilla	17.3	76.7	24.2	63.3	7.4
MixUp [90]	18.4	77.1	24.4	63.6	7.3
CutOut [13]	17.3	24.1	58.4	63.7	7.6
CutMix [86]	18.9	76.7	23.8	65.4	5.3
AugMix [31]	17.6	78.6	28.5	65.2	11.2
PixMix [33]	18.5	80	29.2	**65.8**	11.8
IPMix	**19.3**	**80.9**	**31.1**	65.6	**12.2**

C Evaluation Metrics

We evaluate various safety measures on CIFAR and ImageNet, including corruption robustness, calibration, adversarial robustness, consistency, and anomaly detection. Task evaluation metrics are shown below.

Corruption robustness. Following AugMix, we utilize the Mean Corruption Error (mCE) to test a model's resistance to corrupted data on CIFAR-10-C, CIFAR-100-C, and ImageNet-C. Mean Corruption Error is the mean error rate normalized by the corruption errors of a baseline model over 15 corruption types and 5 corruption severity. We train AlexNet [41] as the baseline for ImageNet experiments.

Calibration. The calibration task is to verify whether the predicted probability estimates are representative of the true correctness likelihood. We use RMS Calibration Error [30] as the metric, which can be computed as $\sqrt{\mathbb{E}_C[(\mathbb{P}(Y = \hat{Y}|C = c) - c)^2]}$, where C is the classifier's confidence that its prediction $\hat{Y}$ is correct. Lower is better.

Adversarial robustness. We utilize PGD to verify the adversarial robustness of image classifiers. We use 20 steps of optimization and an ℓ_∞ budget of 2/255 on CIFAR-10 and CIFAR-100. The metric is the classifier error rate. Lower is better.

Consistency. Following AugMix, we verify perturbation consistency on CIFAR-10-P, CIFAR-100-P, and ImageNet-P. The metric is the mean flip rate (mFR), which can be tested through video frame predictions normalized by a baseline model matched by 10 different perturbation types. We choose AlexNet as the baseline model.

Anomaly detection. We utilize two challenging datasets, ImageNet-A and ImageNet-O to evaluate model robustness under out-of-distribution shifts. The main metric on ImageNet-A is accuracy, and on ImageNet-O is the area under the precision-recall curve (AUPR). Higher is better. The anomaly score is the negative of the maximum softmax probabilities [29].

D The Algorithm of IPMix

The algorithm to generate IPMix images is summarized in Algorithm 1. The fractals we use are selected at random from the IPMix fractal set (for further details, please see Appendix E). On CIFAR, the patch sizes we employ are randomly chosen from a set including 4, 8, 16, and 32, whereas for ImageNet-1K, we opt for patch sizes from 4, 8, 16, 32, 64, and 256. We randomly mix the augmented original image to increase diversity. Across all our experiments, we consistently use $k = 3$ and $t = 3$.

Algorithm 1: Generate IPMix Images

```
input  : Origin image $x$, fractal $x_{\text{fractal}}$, augmentation methods $M$={image-level, P-level}, patch
         sizes $P_{\text{size}}$ , P operations $P$ = {random pixels mixing,...,add}, image operations $I$ =
         {invert,...,mirror} , width $k$, max depth $t$.
output : $x_{\text{IPMix}}$
Sample mixing weights $w_1$,...,$w_k \sim$ Dirichlet($\alpha$,...,$\alpha$)
Sample weights $m \sim$ Beta($\alpha$,$\alpha$)
Generate $x_{\text{mix}}$ = Zerolikes($x$)
for $i \leftarrow 1$ to $k$ do
    Generate $x_{\text{mixed}} = x$.copy()
    Randomly choose method 'meth' from $M$
    if 'meth' == 'P-level' then
        for $j = 1$ to random.choose([1,...,$t$]) do
            Random sample size $s$ from $P_{\text{size}}$          // $P_{\text{size}} = \text{x.size} \rightarrow$ Pixel-level op
            Sample operations $p_o$ from $P$
            if random.random() $>$ 0.5 then
                $x_{\text{mixed}}$ = patch mixing($x_{\text{mixed}}$, $x_{\text{fractal}}$, $s$, $p_o$)          // See Sec.4.2
            else
                Sample operations $i_o$ from $I$                  // For diversity increase
                $x_{\text{aug}}$ = $i_o(x)$
                $x_{\text{mixed}}$ = patch mixing($x_{\text{mixed}}$, $x_{\text{aug}}$, $s$, $p_o$)
    else
        for $j = 1$ to random.choose([1,...,$t$]) do
            Sample operations $i_o$ from $I$
            $x_{\text{mixed}}$ = $i_o(x_{\text{mixed}})$
    $x_{\text{mix}}$ += $w_i \cdot x_{\text{mixed}}$                          // $w_i$ from Dirichlet($\alpha$,...,$\alpha$)
return $x_{\text{IPMix}} = m \cdot x_{\text{mix}} + (1 - m) \cdot x$                // $m$ from Beta($\alpha$,$\alpha$)
```

E Generating Fractal Images

While prior works have exclusively utilized Iterated Function Systems (IFS) to generate fractal data [37, 1], various other fractal-generating programs can also be employed. To further enhance the structural complexity and diversity, we have ventured beyond IFS and incorporated the Escape-time Algorithm to generate 'orbit trap' complex fractals. The most common 'orbit trap' fractal images, Mandelbrot and Julia fractals, can be derived from Eq. (2):

$$F(z) = z^2 + c \tag{2}$$

In Eq. (2), z represents a complex number, and c is a constant value. In the case of the Mandelbrot set, we initialize z at 0, with c corresponding to the specific coordinate in the complex plane that is under examination. Conversely, when generating the Julia set, c remains constant throughout the set, and z is initiated as the particular coordinate that is currently being tested.

Moreover, guided by the approach of [1], we create an additional 3000 fractals, each rendered with a unique, randomly generated background and color scheme using IFS. Furthermore, we supplement our dataset with an additional 6000 fractals obtained from DeviantArt[2]. These images, exhibiting

[2] https://www.deviantart.com/

greater complexity than those generated via IFS or the Escape-time Algorithm, significantly enhance dataset diversity. Besides, we collect 3000 feature images to improve diversity. In total, we assemble a collection of 13000 images named IPMix set for increasing data diversity and structural complexity when mixed with clean images.

F The Details about Combination Experiments

In section 3, we show that simply combining different levels of approaches can degrade model performance across various metrics. Building upon these findings, in this part, we want to examine the impact of the order of operations on combination experiments.

In our experiments, we adopt MixUp [90], CutMix [86], and AugMix [31] as representative techniques for pixel-level, patch-level, and image-level augmentation, respectively. In all experiments, we apply AugMix first, followed by CutMix or MixUp. The rationale behind this order is that AugMix is commonly used in PIL images to enhance data diversity. In contrast, MixUp and CutMix interpolate and mix images after images conversion into tensors. Furthermore, applying Mixup/CutMix before AugMix could lead to unnatural transformations, as AugMix operations would distort the mixed images, counteracting the aim of preserving the individual image context during interpolation.

We have adopted several different combinations as follows.

- First, we apply AugMix, then MixUp, and finally CutMix.
- First, we apply AugMix, then CutMix, and finally MixUp.
- We apply AugMix first, followed by either CutMix or MixUp, chosen randomly.
- We apply AugMix first. Depending on the training epochs, we use either CutMix or MixUp.

Table 15: The combination experiments of different levels of data augmentation on CIFAR-100.

Methods	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
Vanilla	21.3	50	14.6
MixUp	20.5	45.9	10.5
CutMix	20.3	50	9.3
AugMix	20.6	33.3	6.3
AugMix→MixUp→CutMix	23.4	50.1	25.6
AugMix→CutMix→MixUp	27	51.4	26.7
Chosen Randomly ($p = 0.5$)	22.6	40.6	19
Epoch-Dependent	21.1	37.6	7.2

In all experiments, we use the optimal hyperparameters specified in the original papers. We set $k = 3$ for AugMix and $\alpha = 1$ for MixUp and CutMix. The results are demonstrated in Table 15.

We set the total number of training epochs to 100 on 40-4 Wide ResNet for all experiments. In our Epoch-Dependent combination experiments, we found that employing MixUp for the initial 50 epochs and transitioning to CutMix for the rest yielded the best performance. Nevertheless, it doesn't perform as well as the individual augmentation techniques. This underperformance might be due to the increased complexity in the synthesized training instances, possibly impeding the extraction of discriminative feature representations by models. Further experiments could explore different combinations of these techniques to improve their effectiveness.

In order to thoroughly analyze the influence of the augmentation strength of each method, we have conducted experiments considering various hyperparameter combinations. Specifically, we evaluated k = 1, 3, 5 (for AugMix) and α = 0.2, 0.5, 1 (for MixUp and CutMix). We opted to exclude k = 3 and α = 1, the original optimal hyperparameters in their papers, thereby reducing the total combinations from 27 to 8. From the experimental results in Table 16, combining different hyperparameters does not significantly improve the model performance. We set the total number of training epochs to 100 for all experiments with WRN40-4 on CIFAR-100.

Table 16: Could decreasing the augmentation strength of each method yield better performance?

Combination	Classification Error(↓)	Robustness mCE(↓)	Calibration RMS(↓)
$\alpha = 0.2$,$\alpha = 0.2$,$k = 1$	23.9	51.2	25.3
$\alpha = 0.2$,$\alpha = 0.2$,$k = 5$	24.5	51	25.3
$\alpha = 0.2$,$\alpha = 0.5$,$k = 1$	26	50.7	24.9
$\alpha = 0.2$,$\alpha = 0.5$,$k = 5$	24.4	50.6	25.7
$\alpha = 0.5$,$\alpha = 0.2$,$k = 1$	25.8	50.8	25.4
$\alpha = 0.5$,$\alpha = 0.2$,$k = 5$	25	49.1	24.8
$\alpha = 0.5$,$\alpha = 0.5$,$k = 1$	25.5	50.5	25.1
$\alpha = 0.5$,$\alpha = 0.5$,$k = 5$	26	51.2	25.9

G Training Time

In this section, we present a comparative analysis of the training time. The results in Table 17 show that IPMix adds only a modest training overhead over Vanilla, which is advantageous for its practical use in real-world scenarios.

Table 17: We test IPMix on two NVIDIA GeForce RTX 3090 GPUs with ResNet18 for 90 epochs. The training time of IPMix is acceptable by comparison with other data augmentation methods.

Method	Time(sec/epochs)
Vanilla	3764
MixUp [90]	3913
CutOut [13]	3870
CutMix [86]	4139
AugMix [31]	4762
PixMix [33]	4310
AugMax [78]	7564
IPMix	4380

H Full Results of IPMix across Architectures

In Table 18, we show the full results of IPMix across architectures on CIFAR-10 and CIFAR-100.

Table 18: Full results for IPMix on CIFAR-10 and CIFAR-100. We test the ability of IPMix on accuracy, robustness, consistency, adversaries, and calibration across different models. Top: CIFAR-10. Bottom : CIFAR-100.

	Classification Error(↓)	Robustness mCE(↓)	Consistency mFR(↓)	Adversaries Error(↓)	Calibration RMS(↓)
WideResNet40-4	4	8.6	1.3	74.4	2.3
WideResNet28-10	3.3	7.5	1.1	76.4	1.9
ResNeXt-29	3.8	8.6	1.4	93.2	2
ResNet-18	4.2	8.4	1.7	80	2.4
Mean	3.8	8.3	1.4	81	2.2
WideResNet40-4	19.4	28.6	4.3	89.4	2.8
WideResNet28-10	17.4	26.6	4.2	91.3	6.4
ResNeXt-29	18.3	28.1	5	96.9	3.8
ResNet-18	21.6	29.9	5.4	95.6	6.3
Mean	19.2	28.3	4.7	93.3	4.9

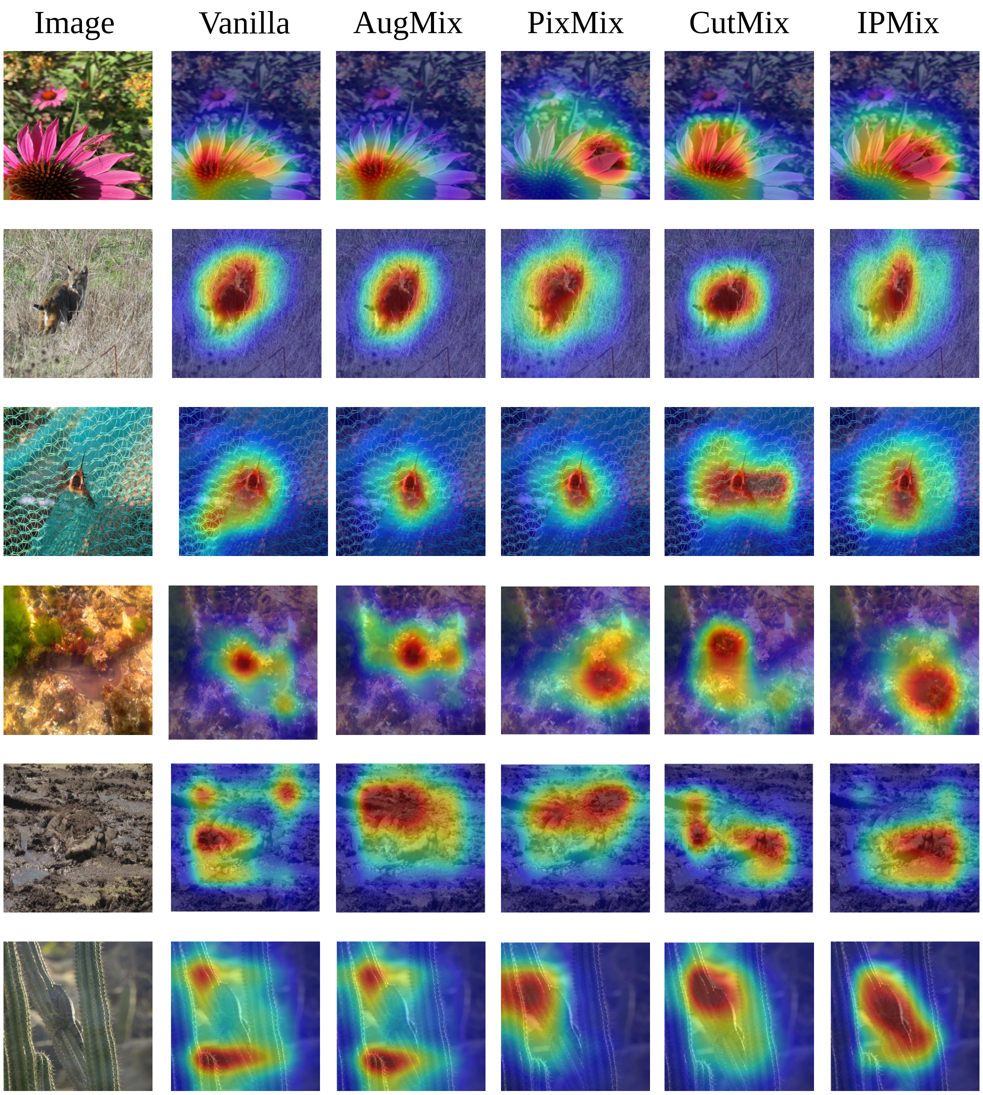

Figure 9: More CAM visualizations of IPMix. Input images come from ImageNet-A, the most challenging dataset to verify the performance of model classifiers against distribution shifts.

I More CAM Visualizations

In this section, we demonstrate more CAM visualizations of IPMix, as shown in Figure 9.

J The Analysis of Ablation Experiments

In this section, we will detailed analyze the impact of each part on different safety metrics through ablation experiment results shown in Table 19.

Accuracy: The image-level augmentation has the most substantial effect on accuracy, aligning with current findings [10, 11] that image-level methods are commonly used to boost accuracy.

Robustness: Both pixel-level and image-level augmentations improve robustness. Since pixel-level introduces fine-grained variations for pattern recognition, while image-level increases dataset diversity, preventing the model from merely memorizing fixed augmentations.

Table 19: Ablation results of different components of IPMix on CIFAR-100. Mean and standard derivation over three random seeds is shown for each experiment. Bold is the best.

	Classification Error(↓)	Robustness mCE(↓)	Consistency mFR(↓)	Adversaries Error(↓)	Calibration RMS(↓)
IPMix	**19.4**$_{(\pm 0.17)}$	**28.6**$_{(\pm 0.2)}$	89.4$_{(\pm 0.18)}$	**4.3**$_{(\pm 0.09)}$	**2.8**$_{(\pm 0.07)}$
w/o patch	19.7$_{(\pm 0.13)}$	30 $_{(\pm 0.21)}$	91.7 $_{(\pm 0.15)}$	4.7 $_{(\pm 0.02)}$	4.6 $_{(\pm 0.07)}$
w/o pixel	19.6 $_{(\pm 0.09)}$	33 $_{(\pm 0.35)}$	92.6 $_{(\pm 0.20)}$	5.2 $_{(\pm 0.05)}$	8.2 $_{(\pm 0.12)}$
w/o image	20.1 $_{(\pm 0.27)}$	34 $_{(\pm 0.65)}$	**87.8** $_{(\pm 0.22)}$	5.5 $_{(\pm 0.11)}$	8.6 $_{(\pm 0.21)}$

Calibration and Consistency: The Image-level part significantly influences calibration and consistency, which increases diversity to improve the prediction calibration across scenarios and ensures consistency in responses to minor perturbations.

Adversarial Attacks: Without the image-level component, adversarial performance improves, implying diverse data might **weaken** defense against attacks. Conversely, removing pixel-level methods will degrade adversarial robustness, given their inherent resistance to perturbations.

K The Experiment Results on Transformer Architecture

In this section, we will evaluate the performance of IPMix on Vision Transformer. We trained a small ViT for 300 epochs on CIFAR-10 and CIFAR-100. This step aimed to confirm IPMix's potential on smaller datasets using Transformer architectures. In future work, we plan to expand our experiments with transformer architectures. The experiment results in Table 20 and Table 21 show that IPMix achieves the best performance on ViT.

Table 20: Experiments on CIFAR-10. Bold is the best.

	Classification Error(↓)	Robustness mCE(↓)	Consistency mFR(↓)	Adversaries Error(↓)	Calibration RMS(↓)
Vanilla	19.5$_{(\pm 0.07)}$	27.7$_{(\pm 0.14)}$	91.3$_{(\pm 0.13)}$	5.9$_{(\pm 0.02)}$	10
MixUp	1$_{(\pm 0.11)}$	34.7$_{(\pm 0.21)}$	89.3$_{(\pm 0.21)}$	6$_{(\pm 0.05)}$	9.9$_{(\pm 0.03)}$
CutMix	19.3$_{(\pm 0.08)}$	34.3 $_{(\pm 0.19}$	89.1$_{(\pm 0.14)}$	5.5$_{(\pm 0.05)}$	7.5 $_{(\pm 0.02)}$
PixMix	28.4$_{(\pm 0.14)}$	33.$_{(\pm 0.24)}$	91$_{(\pm 0.12)}$	6.5$_{(\pm 0.11)}$	4.4$_{(\pm 0.07)}$
AugMix	20.3$_{(\pm 0.14)}$	25.6$_{(\pm 0.2)}$	80.3$_{(\pm 0.16)}$	5.1$_{(\pm 0.09)}$	6$_{(\pm 0.08)}$
IPMix	**19.2**$_{(\pm 0.12)}$	**23.7**$_{(\pm 0.2)}$	**75.8**$_{(\pm 0.13)}$	**3.7**$_{(\pm 0.07)}$	**5.3**$_{(\pm 0.07)}$

Table 21: Experiments on CIFAR-100. Bold is the best.

	Classification Error(↓)	Robustness mCE(↓)	Consistency mFR(↓)	Adversaries Error(↓)	Calibration RMS(↓)
Vanilla	40.1$_{(\pm 0.12)}$	56.3$_{(\pm 0.1)}$	96.2$_{(\pm 0.14)}$	12.4$_{(\pm 0.04)}$	14.8$_{(\pm 0.02)}$
MixUp	40$_{(\pm 0.14)}$	56 $_{(\pm 0.18}$	92.5$_{(\pm 0.18)}$	9.8$_{(\pm 0.03)}$	9.5 $_{(\pm 0.02)}$
CutMix	39.5$_{(\pm 0.11)}$	56.3 $_{(\pm 0.15}$	96.2$_{(\pm 0.17)}$	10$_{(\pm 0.03)}$	9.8 $_{(\pm 0.03)}$
PixMix	48.7$_{(\pm 0.14)}$	54.3$_{(\pm 0.21)}$	93.2$_{(\pm 0.14)}$	10.9$_{(\pm 0.17)}$	4.9$_{(\pm 0.04)}$
AugMix	35.3$_{(\pm 0.17)}$	42.4$_{(\pm 0.21)}$	84.6$_{(\pm 0.16)}$	6.9$_{(\pm 0.03)}$	6.4$_{(\pm 0.07)}$
IPMix	**32.6**$_{(\pm 0.11)}$	**39.6**$_{(\pm 0.23)}$	**83.2**$_{(\pm 0.15)}$	**6.3**$_{(\pm 0.04)}$	**5.3**$_{(\pm 0.05)}$

L The Drawbacks of Different Levels of Methods

In this section, we will reveal the drawbacks of different levels of approaches and explain how IPMix solves these problems.

The drawbacks of label variant methods:

Pixel-level: Mixing images with distinct labels and linearly interpolating between them will impose certain “local linearity” constraints on the model’s input space beyond the data manifold, which may lead to "manifold intrusion". Consider one experiment on MNIST. If we use MixUp to linearly mix two numbers, such as "1" and "5", the generated image will show the characteristics of "8". When the generated "8" collides with a real "8" in the data manifold, there will be a problem of manifold intrusion. Since the two samples have similar characteristics, one is the real label and the other is a soft label ("1" and "5"). This will interfere with its ability to understand and classify categories and degrade model performance.

Patch-level: The problem of manifold intrusion also occurs in the patch-level method, termed "label mismatch." This occurs when the chosen source patch doesn’t accurately represent the source object, leading the interpolated label misleads the model to learn unexpected feature representation. For example, using CutMix to mix images of a cat and a dog. CutMix might select 20 % of the background area from the cat image without information about the object (cat). However, their interpolated labels encourage the model to learn both objects’ features (dog and cat) from that training image and degrade model performance.

The drawbacks of image-level methods:

Image-level data augmentation increases data diversity by applying label-preserving transformations to the whole image. Notable among these are search-based methods like AutoAugment, RandAugment, and FastAugment. While they improve performance effectively, the computationally expensive search for an optimal augmentation policy often exceeds the training process’s complexity. Thus, efforts to minimize the search space, optimize search parameters, and uncover potential universal pipelines are central to the effectiveness of these methods.

In conclusion, we solve these questions by:

- Incorporate structural complexity from synthetic data at various levels to produce more diverse images. Our method is **label-preserving**, ensuring it is not affected by manifold intrusion.
- Randomly sample operations from PIL (e.g., brightness, sharpness) and randomly sample strengths to enhance the diversity of training data **without expensive searching**.
- Integrate three levels of data augmentation into a single framework with limited computational overhead, demonstrating that these approaches are complementary and that a unification among them is necessary to achieve robustness.

M Limitation and Broader Impact

While IPMix has shown promising results, the theoretical foundation of IPMix requires further development to gain deeper insights into its underlying principles. Meanwhile, our approach primarily focuses on CNN, and its effectiveness on Visual Transformers requires additional experimental validation. Additionally, the experiments conducted on a limited set of safety metrics, and the performance of IPMix in real-world scenarios with more comprehensive safety measures warrants future investigation [27]. In continuing our efforts to refine and enhance the IPMix methodology, we will focus on addressing these limitations in future works.

Since IPMix improves various safety measures, it can generate many beneficial effects in real-world environments, improving the robustness against attacks and the calibrated prediction confidence of models. Moreover, IPMix integrates three levels of data augmentation into a single framework, demonstrating that these approaches are complementary and necessary to achieve better performance. We believe the improvements in safety metrics and the coherent framework of combining various techniques will shed light on this field.